TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Nachrichtentechnik

Polar Coding with Complexity-Adaptive Decoding and Time-Varying Channels

Peihong Yuan

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Die Dissertation wurde am 16.06.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 14.10.2021 angenommen.

Acknowledgment

This thesis is based on work conducted at the Institute for Communications Engineering (LNT) at the Technical University of Munich (TUM) during the past years. Many people have contributed to this thesis and I am thankful to all of them.

First, I would like to thank my doctoral advisor Gerhard Kramer for accepting me as a doctoral student and giving me the opportunity to conduct research at his chair. I thank Ido Tal to be my second examiner, and Gerhard Rigoll for chairing the examination.

Then, I would like to express my deepest gratitude to Georg Böcherer who gave me direction already during my Master's.

Next, I want to thank my key collaborator, Mustafa Cemil Coşkun, with whom I worked on a major portion of this thesis. I also want to thank my close collaborators, Tobias Prinz, Patrick Schulte, Fabian Steiner and Thomas Wiegart. Moreover, I wish to thank all the colleagues at LNT for their valuable advices and ideas.

Finally, I would like to thank my family. The support of my family is one of the key drivers that helped me in pursuing this path comfortably. I have been extremely lucky to have you as my family.

München, June 2021 Peihong Yuan

Contents

Zusammenfassung

Polarcodes sind die ersten nachweislich kapazitätserreichbar Codes für binäre diskrete gedächtnislose Kanäle (B-DMCs) und haben aufgrund ihrer geringen Codierungs- und Decodierungskomplexität beträchtliche akademische und industrielle Aufmerksamkeit erhalten. Der 5G Mobilfunk Standard implementiert Polarcodes als eines seiner Kanalcodierungsschemata. Diese Arbeit untersucht Decodieralgorithmen, Ratenanpassung, Kanalschätzung und Modulationen höherer Ordnung für Polarcodes.

Für die Dekodierung wird ein komplexitätsadaptiver Suchalgorithmus vorgeschlagen, der als Successive-Cancelling-Ordered-Search (SCOS) bezeichnet wird und eine Maximum-Likelihood (ML)-Dekodierung implementiert. Vergleiche mit existierenden Dekodierern zeigen, dass SCOS eine geringere Komplexität und eine bessere Zuverlässigkeit aufweist und robuster gegenüber Änderungen der Kanalparameter ist. Einfache Modifikationen des Algorithmus begrenzen Komplexität des Worst Case mit geringem Leistungseinbußen. Für die Ratenanpassung wird eine Erweiterung der plarcodes mit variabler Länge (VLPE) eingeführt, welche auf dynamischen eingefrorenen Bits als ein hybrides automatisches Wiederholungsanfragesystem basiert. Numerische Ergebnisse zeigen, dass die von der VLPE erzeugten Polarcodes ähnlich wie klassische Polarcodes funktionieren. Schließlich wird für Block-Fading-Kanäle ein Kanalschätzungsschema vorgeschlagen, das die Codebeschränkungen der eingefrorenen Bits verwendet. Dieses Schema wird für die zweistufige polarcodierte Übertragung (PCT) ohne Pilotsymbole verwendet, um Kanal und Nachricht gemeinsam zu schätzen. Numerische Ergebnisse zeigen, dass PCT Übertragungsschemen mit Pilotsymbolen und ähnlicher Komplexität deutlich übertrifft und sich der Leistung eines kohärenten Empfängers annähert.

Abstract

Polar codes are the first provably capacity-achieving codes for binary-input discrete memoryless channels (B-DMCs) and have attracted considerable attention in academia and industry due to their low encoding and decoding complexity. The 5-th generation wireless system (5G) standardized polar codes as one of its channel coding schemes. This thesis investigates decoding algorithms, rate adaptation, channel estimation, and higher-order modulation for polar codes.

For decoding, a complexity-adaptive tree search algorithm called successive cancellation ordered search (SCOS) is proposed that implements maximum-likelihood (ML) decoding. Comparisons with existing decoders show that SCOS has lower complexity and better reliability, and is more robust to changes in the channel parameters. Simple modifications to the algorithm limit the worst-case complexity with only small degradations in performance. For rate adaptation, a variable-length polar extension (VLPE) based on dynamic frozen bits is introduced as a hybrid automatic repeat request scheme. Numerical results show that the polar codes generated by the VLPE perform similar to classic polar codes. Finally, for block fading channels a channel estimation scheme is proposed that uses the code constraints imposed by the frozen bits. This scheme is used for a pilot-free two-stage polar-coded transmission (PCT) to jointly estimate the channel and message. Numerical results show that PCT significantly outperforms pilot-assisted transmission with a similar complexity and approaches the performance of a coherent receiver.

1 Introduction

Shannon provided the blueprint for digital communications in his landmark work [99]. However, Shannon's proof that reliable communication is possible over noisy channels was based on random coding that does not specify a good code with low decoding complexity.

Algebraic coding dominated the first decades of coding theory. Research focused on linear codes with good algebraic properties, e.g., large minimum Hamming distance. Important examples include Golay [39], Hamming [40], Reed-Muller (RM) [77,84], and Reed-Solomon (RS) [85] codes. The year 1993 saw the announcement of turbo codes with iterative decoding [9,10] that achieve near-Shannon-limit performance with reasonable decoding complexity. Shortly thereafter, low-density parity-check (LDPC) codes [36,37] were rediscovered [63,100] that also have a low complexity iterative decoder.

Polar codes were proposed in [5, 101] and Arıkan proved in [5] that they achieve the capacity of binary-input discrete memoryless channels (B-DMCs) asymptotically in the code length with low encoding and decoding complexity. The origin of polar codes stems from the computational cutoff rate of channels [4]. Consider two uses of a B-DMC $p_{Y_1|X_1} = p_{Y_2|X_2} = p_{Y|X}$. The simple linear transform $X_1 = U_1 + U_2$, $X_2 = U_2$ converts the original channel to two (virtual) bit channels $p_{Y_1Y_2|U_1}$ and $p_{Y_1Y_2U_1|U_2}$ with a larger average cutoff rate than the original channel. By recursively repeating the transform, the capacity of the bit channels polarizes to either zero or one as the code length tends to infinity. Moreover, the proportion of virtual channels with capacity close to one converges to the capacity of the original channel. Arıkan called this phenomenon *channel polarization*.

This thesis focuses on three problems and applications of polar codes:

▷ Low complexity decoding of polar (and polar-like) codes.

- ▷ Retransmission protocols of polar coded communication systems.
- ▷ Channel estimation that exploits the imperfection of channel polarization.

The "core" of the thesis is Chapters 4-6 that introduce most of the findings and contributions. The thesis is structured as follows:

- ▷ Chapter 2 describes notation, reviews information theory, and provides basic tools for subsequent chapters.
- ▷ Chapter 3 reviews polar coding from both theoretical and practical points of view. We start with the basic linear transform and use it to develop the channel polarization phenomenon. We then consider practical decoding schemes, code design, rate adaptation and higher-order modulation.
- ▷ Chapter 4 introduces a complexity-adaptive tree search decoding algorithm for polar (and polar-like) codes that implements maximum-likelihood (ML) decoding by using a successive decoding schedule. We provide a complexity analysis and compare with existing decoders.
- ▷ Chapter 5 deals with retransmission protocols with polar coding. A variable-length retransmission scheme based on dynamic frozen bits is proposed. This scheme is extended to higher-order modulation.
- ▷ Chapter 6 discusses polar coding over block fading channels. We propose a channel estimation method by using the code constraints imposed by the frozen bits. We further introduce a pilot-free two-stage polar-coded transmission scheme to jointly estimate the channel state and message.
- ▷ Chapter 7 concludes the thesis by summarizing the main contributions of each chapter.
- ▷ Appendix A provides contributions on outer code design and non-binary kernels.

2 Preliminaries

This chapter introduces notation and information theory for channel coding. Section 2.1 specifies set, probability, and information theory notation. Section 2.2 describes a communication model for channel coding and modulation. Section 2.3 reviews the capacity of discrete and additive white Gaussian noise (AWGN) channels. Finally, Section 2.4 reviews log-likelihood ratio (LLR) computation.

2.1. Notation

Sets and Numbers

Sets are generally written as calligraphic letters, e.g., $\mathcal{X}$, $\mathcal{F}$ or $\mathcal{A}$. The cardinality of $\mathcal{X}$ is $|\mathcal{X}|$. $\mathcal{A} \cup \mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$ denote the union and intersection of $\mathcal{A}$ and $\mathcal{B}$, respectively. The set difference of $\mathcal{A}$ and $\mathcal{B}$ is written as $\mathcal{A} \setminus \mathcal{B}$.

The sets of natural, real, and complex numbers are written as $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{C}$, respectively. For $\mathbb{N}$, we write $[k] = \{i : i \in \mathbb{N}, 1 \leq i \leq k\}$. For $\mathbb{R}$, we write $x \in [a, b]$ and $x \in [a, b)$ for the intervals $a \leq x \leq b$ and $a \leq x < b$, respectively. The notation $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ refers to the ceiling and floor functions, respectively. For $\mathbb{C}$, the imaginary unit is j. The operators $\Re(\cdot)$, $\Im(\cdot)$, $|\cdot|$ and $(\cdot)^*$ return the real part, the imaginary part, the magnitude and the conjugate of a complex number, respectively.

Vectors and Matrices

Row vectors are written as $x^n = \boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. The all-zeros row vector of dimension n is $\mathbf{0} = 0^n$. The i-th entry of $\boldsymbol{x}$ is x_i and we write $x_i^j = (x_i, \ldots, x_j)$. If $j < i$ then x_i^j is void. We write $x_{\mathcal{A}}$ for the row vector formed by the ordered elements with indices in $\mathcal{A}$. A matrix is written as $\boldsymbol{X}$ and the all-zeros matrix as $\mathbf{0}$, where we reuse notation. The transpose and conjugate transpose of $\boldsymbol{X}$ are $\boldsymbol{X}^{\mathrm{T}}$ and $\boldsymbol{X}^{\mathrm{H}}$, respectively.

$\overline{x}$ denotes the bit-flipped version of a binary scalar x. The notation $\overline{x^i}$ refers to the element-wise bit-flipped version of a vector x^i with binary entries. $x \oplus y$ denotes the XOR of two binary scalars. The element-wise XOR of two binary vectors is written as $x^n \oplus y^n$.

Probability

$\Pr(\mathcal{E})$ is the probability of event $\mathcal{E}$. The probability of $\mathcal{E}_1$ conditioned on $\mathcal{E}_2$ with $\Pr(\mathcal{E}_2) > 0$ is $\Pr(\mathcal{E}_1|\mathcal{E}_2) = \Pr(\mathcal{E}_1 \cap \mathcal{E}_2)/\Pr(\mathcal{E}_2)$.

A random variable (RV) is written with an uppercase letter such as X. A realization of X is written with the corresponding lowercase letter x. A vector of RVs is written as $X^n = \underline{X} = (X_1, X_2, \ldots, X_n)$.[1] The probability mass function (PMF) of a discrete RV X evaluated at x is

$$P_X(x) \triangleq \Pr(X = x). \tag{2.1}$$

For continuous RVs with a density, the probability density function (PDF) evaluated at x is $p_X(x)$. The support sets are

$$\operatorname{supp}(P_X) = \{x : P_X(x) > 0\} \tag{2.2}$$
$$\operatorname{supp}(p_X) = \{x : p_X(x) > 0\}. \tag{2.3}$$

The expectation value of a real-valued function $f(\cdot)$ of a discrete RV X is

$$\mathbb{E}[f(X)] = \sum_{x \in \operatorname{supp}(P_X)} f(x) P_X(x). \tag{2.4}$$

For a continuous RV X, we have

$$\mathbb{E}[f(X)] = \int_{x \in \operatorname{supp}(p_X)} f(x) p_X(x)\, \mathrm{d}x. \tag{2.5}$$

[1]The dimension of a vector is denoted as a superscript, if it is important, i.e., $x^n = \boldsymbol{x}, X^n = \underline{X}$.

The variance of a real-valued X is

$$\mathrm{Var}\left[X\right] = \mathbb{E}\left[\left(X - \mathbb{E}\left[X\right]\right)^2\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2. \tag{2.6}$$

The continuous RV X is Gaussian if it has the density

$$p_X\left(x\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},\ x \in \mathbb{R} \tag{2.7}$$

where $\mu = \mathbb{E}\left[X\right]$ is the mean and $\sigma^2 = \mathrm{Var}\left[X\right]$ is the variance. We write real Gaussian RVs as $X \sim \mathcal{N}\left(\mu, \sigma^2\right)$. Similarly, we write $X \sim \mathcal{CN}\left(\mu, 2\sigma^2\right)$ for a complex-valued RV X whose real and imaginary parts are independent and Gaussian with means $\Re(\mu)$ and $\Im(\mu)$, respectively, and that each have variance σ^2, i.e., we have

$$\begin{cases} \Re\left(X\right) \sim \mathcal{N}\left(\Re(\mu), \sigma^2\right) \\ \Im\left(X\right) \sim \mathcal{N}\left(\Im(\mu), \sigma^2\right) \\ p_X(a + b\mathrm{j}) = p_{\Re(X)}(a) \cdot p_{\Im(X)}(b),\ \forall a, b \in \mathbb{R}. \end{cases} \tag{2.8}$$

We write $X \sim \mathcal{U}\left(\mathcal{X}\right)$ if X is uniformly distributed on $\mathcal{X}$. For example, $X \sim \mathcal{U}\left([a, b)\right)$ is a real-valued RV X that is uniformly distributed on $[a, b)$, i.e., we have

$$p_X\left(x\right) = \frac{1}{b - a},\ a \le x < b. \tag{2.9}$$

Similarly, for discrete and finite $\mathcal{X}$ the notation $X \sim \mathcal{U}\left(\mathcal{X}\right)$ means

$$P_X\left(x\right) = \frac{1}{|\mathcal{X}|},\ x \in \mathcal{X}. \tag{2.10}$$

Information Theory

Consider discrete X and Y. The *entropy* of X is

$$\mathbb{H}\left(X\right) = \mathbb{E}\left[-\log_2 P_X(X)\right] = \sum_{x \in \mathrm{supp}(P_X)} -P_X(x) \log_2 P_X(x). \tag{2.11}$$

The *joint entropy* of X and Y is

$$\mathbb{H}\left(XY\right) = \mathbb{E}\left[-\log_2 P_{XY}(XY)\right] = \sum_{xy \in \mathrm{supp}(P_{XY})} -P_{XY}(xy) \log_2 P_{XY}(xy). \tag{2.12}$$

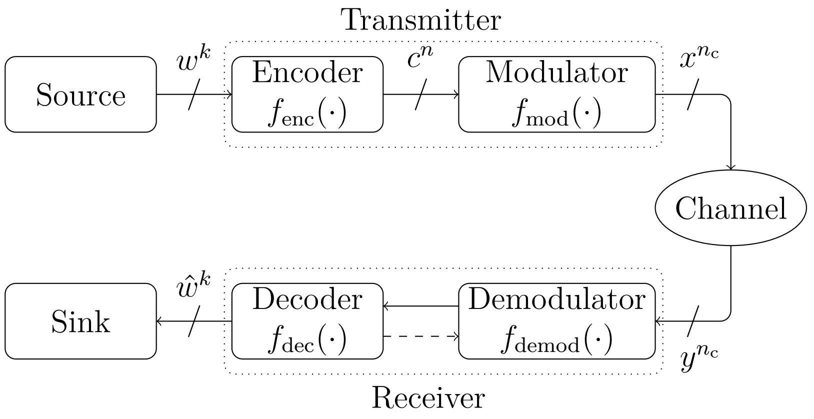

Transmitter
Source
w^k
Encoder $f_{\mathrm{enc}}(\cdot)$
c^n
Modulator $f_{\mathrm{mod}}(\cdot)$
x^{n_c}
Channel
Sink
$\hat{w}^k$
Decoder $f_{\mathrm{dec}}(\cdot)$
Demodulator $f_{\mathrm{demod}}(\cdot)$
y^{n_c}
Receiver

Figure 2.1.: A discrete-time coded modulation model.

The *conditional entropy* of X given Y is

$$\mathbb{H}(X|Y) = \mathbb{E}\left[-\log_2 P_{X|Y}(X|Y)\right] = \sum_{xy \in \mathrm{supp}(P_{XY})} -P_{XY}(xy)\log_2 P_{X|Y}(x|y). \tag{2.13}$$

The *mutual information* of X and Y is

$$\mathbb{I}(X;Y) = \mathbb{H}(X) - \mathbb{H}(X|Y) = \mathbb{H}(Y) - \mathbb{H}(Y|X). \tag{2.14}$$

For a continuous RV X, the *differential entropy* is

$$\mathrm{h}(X) = \mathbb{E}\left[-\log_2(p_X(X))\right] = -\int_{x\in\mathrm{supp}(P_X)} p_X(x)\log_2 p_X(x)\mathrm{d}x. \tag{2.15}$$

Other information-theoretic quantities for continuous RVs can be formulated in terms of the differential entropy, as done above for discrete RVs.

2.2. Communication Model

The coded modulation model described in [66] and shown in Figure 2.1 is widely recognized as a fundamental model of digital communications. We consider discrete-time block-based transmission, i.e., we treat continuous-time waveforms as part of a channel.

The model has five main parts:

▷ The *source* puts out a string w^k of symbols called the *message*. We consider a binary symmetric source (BSS), i.e., w^k has entries that are independent and identically distributed (i.i.d.) with a uniform distribution on $\{0,1\}$, i.e., we have

$$\begin{cases} P_{W^k}\left(w^k\right) = \prod_{i=1}^{k} P_{W_i}\left(w_i\right) \\ P_{W_i}\left(w_i\right) = P_W\left(w\right), \ i \in [k] \\ P_W\left(0\right) = P_W\left(1\right) = 0.5. \end{cases} \tag{2.16}$$

- ▷ The *transmitter* consists of an encoder and a modulator. We consider an (n,k) binary linear encoder, where n and k are the code length and dimension, respectively. The code rate is $R = k/n$. The *encoder* $f_{\text{enc}}(\cdot)$ associates each message w^k with a codeword c^n of the codebook $\mathcal{C}$, where $|\mathcal{C}| = 2^k$. The *modulator* $f_{\text{mod}}(\cdot)$ takes the codeword c^n as input and converts it to a sequence of length n_{c} with entries taken from a constellation $\mathcal{X}$.
- ▷ The string $x^{n_{\text{c}}}$ of modulated signal points is transmitted over a *channel* $p_{Y^{n_{\text{c}}}|X^{n_{\text{c}}}}$ and the receiver sees $y^{n_{\text{c}}}$.
- ▷ The *receiver* consists of a demodulator and a decoder. The *demodulator* $f_{\text{demod}}(\cdot)$ converts $y^{n_{\text{c}}}$ to soft (or hard) information. The *decoder* $f_{\text{dec}}(\cdot)$ has the task to estimate the message. Depending on the receiver criteria, the demodulator and the decoder can operate separately or jointly.
- ▷ The decoded message $\hat{w}^k$ is sent to the *sink*.

We are interested in three properties of a transmission system:

- ▷ *Transmission rate:* The rate is measured by a so-called spectral efficiency (SE)[2] in bits per channel use (bpcu):

$$\text{SE} \triangleq \frac{\text{number of transmitted bits}}{\text{number of channel uses}}. \tag{2.17}$$

For example, the SE of the model in Figure 2.1 is given by

$$\text{SE} = \frac{k}{n_{\text{c}}} \text{ bpcu}. \tag{2.18}$$

- ▷ *Reliability:* We measure reliability by the block error probability defined as

$$\text{block error probability} \triangleq \Pr\left(\hat{W}^k \neq W^k\right). \tag{2.19}$$

[2]The terminology is based on the assumption that we have a bandlimited linear channel with "sinc" pulses and Nyquist-rate signaling.

▷ *Power efficiency:* The average transmission power per channel use is

$$E_\mathrm{s} = \mathbb{E}\left[|X|^2\right]. \tag{2.20}$$

2.3. Channel Capacity

The *channel capacity* [99] specifies the supremum of rates for which one can communicate reliably over a communication channel. Shannon developed the simple-looking capacity formula

$$\mathrm{C} = \max_X \ \mathbb{I}\left(X;Y\right) \tag{2.21}$$

where X and Y are the channel input and output, respectively.

We consider only memoryless channels, i.e., for discrete inputs and outputs we have

$$P_{Y_i|X^iY^{i-1}}\left(y_i \left| x^i y^{i-1}\right.\right) = P_{Y_i|X_i}\left(y_i \left| x_i\right.\right), \quad i \in [n_\mathrm{c}] \tag{2.22}$$

and similarly for continuous outputs. The memoryless channel is time-invariant if $P_{Y_i|X_i} = P_{Y|X}$ for all i. On the other hand, if we repeatedly use length-L blocks of a discrete memoryless channel then the capacity is

$$\mathrm{C} = \frac{1}{L}\max_{P_{X^L}} \mathbb{I}\left(X^L;Y^L\right) = \frac{1}{L}\sum_{i=1}^{L}\max_{P_{X_i}} \mathbb{I}\left(X_i;Y_i\right). \tag{2.23}$$

For example, a binary erasure channel (BEC) has $\mathcal{X} = \{0,1\}$, $\mathcal{Y} = \{0,1,e\}$, and the transition probabilities

$$\begin{cases} P_{Y|X}\left(0|0\right) = P_{Y|X}\left(1|1\right) = 1-p \\ P_{Y|X}\left(e|0\right) = P_{Y|X}\left(e|1\right) = p \end{cases} \tag{2.24}$$

where p is the erasure probability. A uniform input distribution gives

$$\mathrm{C} = 1-p. \tag{2.25}$$

For continuous outputs, a symmetric binary-input channel with continuous outputs has

$$p_{Y|X}\left(y|0\right) = p_{Y|X}\left(-y|1\right), \ y \in \mathbb{R} \tag{2.26}$$

and a uniform input distribution again achieves the capacity.

Real-Alphabet AWGN Channel

The output of a real-alphabet AWGN channel is given by

$$Y = X + Z \tag{2.27}$$

where $\mathcal{X} = \mathcal{Y} = \mathbb{R}$ and $Z \sim \mathcal{N}\left(0, \sigma^2\right)$ is independent of X. We have

$$p_{Y|X}\left(y|x\right) = p_Z\left(y - x\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-x)^2}{2\sigma^2}}. \tag{2.28}$$

The noise power spectral density N_0 is defined as $N_0 \triangleq \sigma^2$ and the signal-to-noise ratio (SNR) is

$$\mathrm{SNR} = \frac{E_\mathrm{s}}{N_0} = \frac{E_\mathrm{s}}{\sigma^2} = \frac{\mathbb{E}\left[X^2\right]}{\sigma^2}. \tag{2.29}$$

The AWGN channel capacity is (see [23, Section 9.2])

$$\mathrm{C} = \frac{1}{2}\log_2\left(1 + \mathrm{SNR}\right). \tag{2.30}$$

For discrete inputs and continuous outputs, the capacity is

$$\begin{aligned}\mathrm{C} &= \max_{P_X}\left[\mathbb{H}\left(X\right) - \mathbb{H}\left(X|Y\right)\right] && (2.31)\\ &= \max_{P_X}\left[\mathrm{h}\left(Y\right) - \mathrm{h}\left(Y|X\right)\right] && (2.32)\\ &= \left[\max_{P_X}\mathrm{h}\left(Y\right)\right] - \frac{1}{2}\log_2\left(2\pi e\sigma^2\right). && (2.33)\end{aligned}$$

For example, the binary-input additive white Gaussian noise (biAWGN) channel has $\mathcal{X} = \left\{-\sqrt{E_\mathrm{s}}, +\sqrt{E_\mathrm{s}}\right\}$, the optimal X has $\mathbb{H}\left(X\right) = 1$, and the capacity is

$$\mathrm{C} = 1 - \int_{-\infty}^{\infty} p_{Y|X}\left(y \middle| +\sqrt{E_\mathrm{s}}\right)\log_2\left(1 + e^{-\frac{2}{\sigma^2}\sqrt{E_\mathrm{s}}y}\right)\mathrm{d}y. \tag{2.34}$$

We remark that real-alphabet channels can be treated as complex-alphabet channels by grouping the real symbols into pairs as complex symbols.

Complex-Alphabet Fading Channel

The output of a complex-alphabet fading AWGN channel is

$$Y = HX + Z \tag{2.35}$$

where $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \mathcal{H} = \mathbb{C}$, $Z \sim \mathcal{CN}(0, 2\sigma^2)$, and H, X, Z are mutually independent. The noise power spectral density N_0 is defined as $N_0 \triangleq 2\sigma^2$. Suppose the receiver knows $H = h$ and computes

$$\frac{1}{h} y = \frac{h^*}{|h|^2} y = x + \frac{h^*}{|h|^2} z. \tag{2.36}$$

Thus, the receiver can form the equivalent channel

$$\tilde{y} = x + \tilde{z} \tag{2.37}$$

where

$$\tilde{y} = \frac{h^* y}{|h|^2}, \quad \tilde{Z} \sim \mathcal{CN}\left(0, \frac{2\sigma^2}{|h|^2}\right). \tag{2.38}$$

Channel Degradation

Consider two channels $p_{Y_{\mathrm{I}}|X}$ and $p_{Y_{\mathrm{II}}|X}$. If $X - Y_{\mathrm{I}} - Y_{\mathrm{II}}$ forms a Markov chain, then we say that the channel $p_{Y_{\mathrm{II}}|X}$ is *physically degraded* (or simply *degraded*) with respect to the channel $p_{Y_{\mathrm{I}}|X}$. We write this as $p_{Y_{\mathrm{II}}|X} \preceq p_{Y_{\mathrm{I}}|X}$. For example, an AWGN with higher noise power is degraded with respect to an AWGN with lower noise power. The *data-processing inequality* [23, Theorem 2.8.1] gives

$$\mathbb{I}(X; Y_{\mathrm{II}}) \le \mathbb{I}(X; Y_{\mathrm{I}}). \tag{2.39}$$

2.4. Decoding

A block-wise maximum-a-posteriori (MAP) decoder puts out the codeword that maximizes the a-posteriori probability (APP) of the channel output, i.e., we have

$$\hat{\boldsymbol{c}} = \underset{\boldsymbol{c} \in \mathcal{C}}{\operatorname{argmax}}\ P_{\underline{C}|\underline{Y}}(\boldsymbol{c}|\boldsymbol{y}). \tag{2.40}$$

A block-wise ML decoder instead puts out the codeword

$$\hat{\boldsymbol{c}} = \operatorname*{argmax}_{\boldsymbol{c}\in\mathcal{C}} \, p_{\underline{Y}|\underline{C}}\left(\boldsymbol{y}|\boldsymbol{c}\right). \tag{2.41}$$

Of course, the likelihood function is $P_{\underline{Y}|\underline{C}}\left(\boldsymbol{y}|\boldsymbol{c}\right)$ for discrete Y.

Bayes' theorem gives

$$P_{\underline{C}|\underline{Y}}\left(\boldsymbol{c}|\boldsymbol{y}\right) = p_{\underline{Y}|\underline{C}}\left(\boldsymbol{y}|\boldsymbol{c}\right)\frac{P_{\underline{C}}\left(\boldsymbol{c}\right)}{p_{\underline{Y}}\left(\boldsymbol{y}\right)}. \tag{2.42}$$

Block-wise MAP decoding is thus the same as block-wise ML decoding if $P_{\underline{C}}\left(\boldsymbol{c}\right) = 1/|\mathcal{C}|$.

Log-Likelihood Ratios

Consider a binary codeword $\boldsymbol{c}$ and a B-DMC with transition probability $p_{Y_i|C_i}$. The LLR based on the channel output is defined as[3]

$$\ell\left(c_i\right) \triangleq \log\frac{P_{C_i|Y_i}(0|y_i)}{P_{C_i|Y_i}(1|y_i)}. \tag{2.43}$$

By using Bayes' theorem, we have

$$\log\frac{P_{C_i|Y_i}(0|y_i)}{P_{C_i|Y_i}(1|y_i)} = \log\frac{p_{Y_i|C_i}(y_i|0)P_{C_i}(0)}{p_{Y_i|C_i}(y_i|1)P_{C_i}(1)} \tag{2.44}$$

$$= \log\frac{p_{Y_i|C_i}(y_i|0)}{p_{Y_i|C_i}(y_i|1)} + \log\frac{P_{C_i}(0)}{P_{C_i}(1)}. \tag{2.45}$$

If C_i is uniformly distributed on $\{0,1\}$, then we have

$$\ell\left(c_i\right) = \log\frac{p_{Y_i|C_i}(y_i|0)}{p_{Y_i|C_i}(y_i|1)}. \tag{2.46}$$

Binary Message Passing

Binary message passing algorithms are discussed in [36, 37, 53, 87, 91, 104]. We consider messages output by decoders for the check nodes (CNs) and variable nodes (VNs) of a LDPC code.

[3]We use the common LLR notation that has a lowercase argument although the LLR is actually a property of the corresponding random variable.

We begin with the CNs. Consider a B-DMC $p_{Y_i|X_i}$ and suppose

$$S = X_1 \oplus \cdots \oplus X_n. \tag{2.47}$$

The LLR of X_i based on Y_i is

$$\ell\left(x_i\right) = \log \frac{P_{X_i|Y_i}(0|y_i)}{P_{X_i|Y_i}(1|y_i)}, \quad i \in [n]. \tag{2.48}$$

A *degree-n CN update* outputs the LLR of S based on Y^n, i.e., we have

$$\ell\left(s\right) = \log \frac{P_{S|Y^n}(0|y^n)}{P_{S|Y^n}(1|y^n)} = 2\tanh^{-1}\left(\prod_{i=1}^{n} \tanh \frac{\ell\left(x_i\right)}{2}\right) \tag{2.49}$$

where $\tanh\left(\cdot\right)$ denotes the hyperbolic tangent

$$\tanh\left(\frac{a}{2}\right) = \frac{e^a - 1}{e^a + 1}. \tag{2.50}$$

We now consider the VNs. Consider a B-DMC $p_{Y_i|X_i}$ and suppose

$$S = X_1 = \cdots = X_n. \tag{2.51}$$

The LLR of X_i based on Y_i is again (2.48). A *degree-n VN update* outputs the LLR of S based on Y^n:

$$\ell\left(s\right) = \log \frac{P_{S|Y^n}(0|y^n)}{P_{S|Y^n}(1|y^n)} = \sum_{i=1}^{n} \ell\left(x_i\right). \tag{2.52}$$

3 Polar Coding and Decoding

This chapter reviews binary polar codes and decoding as proposed in [5, 101]. Non-binary polar codes are developed in [18, 76, 95, 112, 124]. Polar codes with non-binary 2×2 kernels [124] are introduced in Section A.2.

3.1. Basic Transform

Consider a rate-one binary code of length $N = 2$ over a stationary *time-invariant* symmetric B-DMC $p_{Y_i|C_i} = p_{Y|C},\ i \in [N]$. The generator matrix is a size-2 transform

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.1}$$

and the encoding is represented by

$$(c_1, c_2) = (u_1, u_2) \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = (u_1 \oplus u_2, u_2) \tag{3.2}$$

as visualized in Figure 3.1, where U_i and C_i are uniformly distributed on $\{0, 1\}$. The decoder initializes by computing the LLRs

$$\ell\left(c_i\right) = \log \frac{P_{C_i|Y_i}(0|y_i)}{P_{C_i|Y_i}(1|y_i)},\ i = 1, 2. \tag{3.3}$$

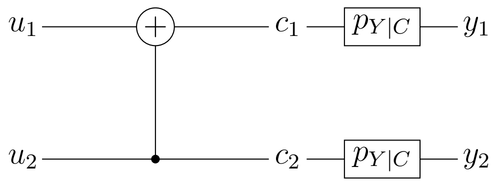


Figure 3.1.: A size-2 transform $\boldsymbol{F}$.

The message is estimated as follows via *successive cancellation (SC)* decoding as visualized in Figure 3.2.

Step 1. Compute the LLR of u_1 by a degree-2 CN update function

$$
\begin{aligned}
f^{-}\left(\ell\left(c_1\right), \ell\left(c_2\right)\right) \triangleq \ell\left(u_1\right) &= \log \frac{P_{U_1|Y_1Y_2}(0|y_1y_2)}{P_{U_1|Y_1Y_2}(1|y_1y_2)} && (3.4)\\
&= 2\tanh^{-1}\left(\tanh\frac{\ell\left(c_1\right)}{2}\cdot\tanh\frac{\ell\left(c_2\right)}{2}\right) && (3.5)\\
&= \operatorname{sign}\left(\ell\left(c_1\right)\right)\cdot\operatorname{sign}\left(\ell\left(c_2\right)\right)\cdot\min\left\{\left|\ell\left(c_1\right)\right|, \ell\left(c_2\right)\right|\right\} && (3.6)\\
&\quad + \log\left(1+e^{-|\ell(c_1)+\ell(c_2)|}\right) - \log\left(1+e^{-|\ell(c_1)-\ell(c_2)|}\right). && (3.7)
\end{aligned}
$$

where sign$(\cdot)$ is the signum function. With the approximation

$$
\log\left(1+x\right) \approx \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \tag{3.8}
$$

we have a hardware friendly version[1] [7]

$$
f^{-}\left(\ell\left(c_1\right), \ell\left(c_2\right)\right) \approx \operatorname{sign}\left(\ell\left(c_1\right)\right)\cdot\operatorname{sign}\left(\ell\left(c_2\right)\right)\cdot\min\left\{\left|\ell\left(c_1\right)\right|, \ell\left(c_2\right)\right|\right\}. \tag{3.9}
$$

Step 2. Perform a hard decision

$$
\hat{u}_1 = \begin{cases} 0, & \text{if } \ell\left(u_1\right) \geq 0 \\ 1, & \text{if } \ell\left(u_1\right) < 0. \end{cases} \tag{3.10}
$$

Step 3. Suppose the estimate $\hat{u}_1$ is correct. u_2 now has two independent constraints:

$$
u_2 = \begin{cases} c_2 \\ u_1 \oplus c_1 = \hat{u}_1 \oplus c_1 \end{cases} \tag{3.11}
$$

[1]The simulation results in this thesis are by default with the approximation. Note that the approximation (3.8) is the same as the rectified linear activation function (ReLU) of neural networks.

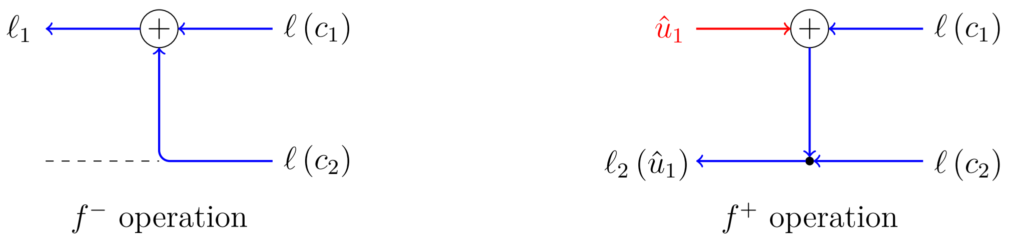


Figure 3.2.: An SC decoder of size-2.

Thus, the LLR of u_2 based on $\hat{u}_1$ is computed by a degree-2 VN update function

$$\begin{aligned}\ell(u_2|\hat{u}_1) &\triangleq \log\frac{P_{U_2|Y_1Y_2U_1}(0|y_1y_2\hat{u}_1)}{P_{U_2|Y_1Y_2U_1}(1|y_1y_2\hat{u}_1)} &(3.12)\\ &= \ell(c_2)+\ell(c_1\oplus\hat{u}_1) &(3.13)\\ &= \begin{cases}\ell(c_2)+\ell(c_1), & \text{if } \hat{u}_1=0\\ \ell(c_2)-\ell(c_1), & \text{if } \hat{u}_1=1.\end{cases} &(3.14)\end{aligned}$$

We define the function

$$f^+(\ell(c_1),\ell(c_2),\hat{u}_1)=(1-2\hat{u}_1)\,\ell(c_1)+\ell(c_2). \tag{3.15}$$

Step 4. Perform a hard decision

$$\hat{u}_2=\begin{cases}0, & \text{if } \ell(u_2|\hat{u}_1)\geq 0\\ 1, & \text{if } \ell(u_2|\hat{u}_1)<0\end{cases} \tag{3.16}$$

Effectively, the size-2 transform converts the B-DMC to two symmetric *bit channels*

$$p_{Y|C}\xrightarrow{\boldsymbol{F}}\begin{cases}p_{Y_1Y_2|U_1}\\ p_{Y_1Y_2U_1|U_2}.\end{cases} \tag{3.17}$$

We further have

$$\begin{aligned}2\cdot\mathbb{I}(C;Y) &= \mathbb{I}(C_1;Y_1)+\mathbb{I}(C_2;Y_2) &(3.18)\\ &= \mathbb{I}(C_1C_2;Y_1Y_2) &(3.19)\\ &\overset{\text{(a)}}{=} \mathbb{I}(U_1U_2;Y_1Y_2) &(3.20)\\ &= \mathbb{I}(U_1;Y_1Y_2)+\mathbb{I}(U_2;Y_1Y_2|U_1) &(3.21)\end{aligned}$$

where (a) follows because the transform $\boldsymbol{F}$ is bijective. We have [5, Proposition 4]

$$\mathbb{I}\left(U_1; Y_1 Y_2\right) \leq \mathbb{I}\left(C; Y\right) \leq \mathbb{I}\left(U_2; Y_1 Y_2 | U_1\right) \tag{3.22}$$

where the equalities hold if $\mathbb{I}\left(C; Y\right) = 0$ or 1.

Example 3.1. Consider a size-2 transform for a BEC with erasure probability p. We use the notation I for the channel capacity before the transform:

$$I \triangleq \mathbb{I}\left(C; Y\right) = 1 - p. \tag{3.23}$$

By using *density evolution (DE)* [87], the bit channel capacities after the transform are

$$\begin{aligned} I^- &\triangleq \mathbb{I}\left(U_1; Y_1 Y_2\right) = (1-p)^2 \\ I^+ &\triangleq \mathbb{I}\left(U_1; Y_1 Y_2 | U_2\right) = 1 - p^2. \end{aligned} \tag{3.24}$$

Obviously, we have $I^- \leq I \leq I^+$ and the equalities hold if and only if $p = 0$ or 1.

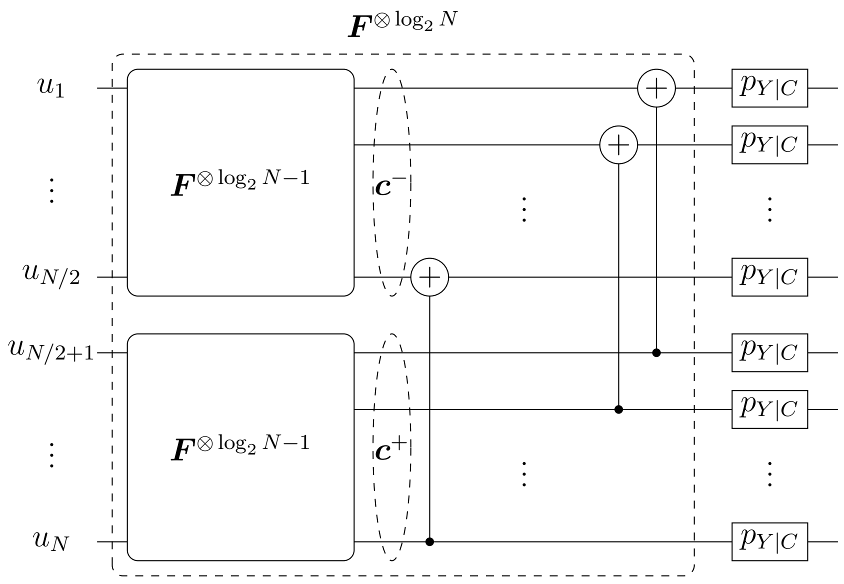
3.2. Channel Polarization

A transform with larger size is recursively constructed with the kernel $\boldsymbol{F}$:

$$\boldsymbol{F}^{\otimes t} = \begin{bmatrix} \boldsymbol{F}^{\otimes t-1} & \boldsymbol{0} \\ \boldsymbol{F}^{\otimes t-1} & \boldsymbol{F}^{\otimes t-1} \end{bmatrix} \tag{3.25}$$

where $\otimes$ denotes the Kronecker product of matrices and $(\cdot)^{\otimes}$ denotes the Kronecker power. $\boldsymbol{F}^{\otimes t}$ is a $2^t \times 2^t$ matrix and thus $\boldsymbol{F}^{\otimes \log_2 N}$ is a size-N transform for N a power of 2. For example, for $N = 8$ we have

$$\boldsymbol{F}^{\otimes \log_2 N} = \begin{bmatrix} \boldsymbol{F}^{\otimes \log_2 N-1} & \boldsymbol{0} \\ \boldsymbol{F}^{\otimes \log_2 N-1} & \boldsymbol{F}^{\otimes \log_2 N-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.26}$$



$\boldsymbol{F}^{\otimes \log_2 N}$
u_1
$u_{N/2}$
$u_{N/2+1}$
u_N
$\boldsymbol{F}^{\otimes \log_2 N-1}$
$\boldsymbol{F}^{\otimes \log_2 N-1}$
$\boldsymbol{c}^-$
$\boldsymbol{c}^+$
$p_{Y|C}$

Figure 3.3.: Recursive structure of a size-N transform $\boldsymbol{F}^{\otimes \log_2 N}$ $(N > 2)$, the output $c^N = (\boldsymbol{c}^- \oplus \boldsymbol{c}^+, \boldsymbol{c}^+)$

and the matrix has the lower-diagonal structure shown in Figure 3.4 where the shaded areas represent the locations where the 1's occur. This structure plays an important role, e.g., for shortening polar codes in Section 3.7 and for hybrid automatic repeat request (HARQ) in Chapter 5.

The encoding of u^N is represented by

$$c^N = u^N \boldsymbol{F}^{\otimes \log_2 N} \tag{3.27}$$

as visualized in Figure 3.3, where U_i and C_i are uniformly distributed on $\{0, 1\}$.

The decoder computes the LLRs $\ell(c_i)$, $i \in [N]$, based on the channel outputs. The message is estimated via recursive SC decoding using Algorithm 3.1 and visualized in Figure 3.5. We perform *Plotkin-decomposition* [82] recursively until we have length-one codes and perform hard decisions [101, Section 5.2.2] with the LLRs. Effectively, the size-N transform converts the B-DMC to N symmetric bit channels

$$p_{Y|C} \xrightarrow{\boldsymbol{F}^{\otimes \log_2 N}} p_{Y^N U^{i-1}|U_i}, \quad i \in [N]. \tag{3.28}$$

We have

$$N \cdot \mathbb{I}(C; Y) = \mathbb{I}\left(C^N; Y^N\right) \tag{3.29}$$

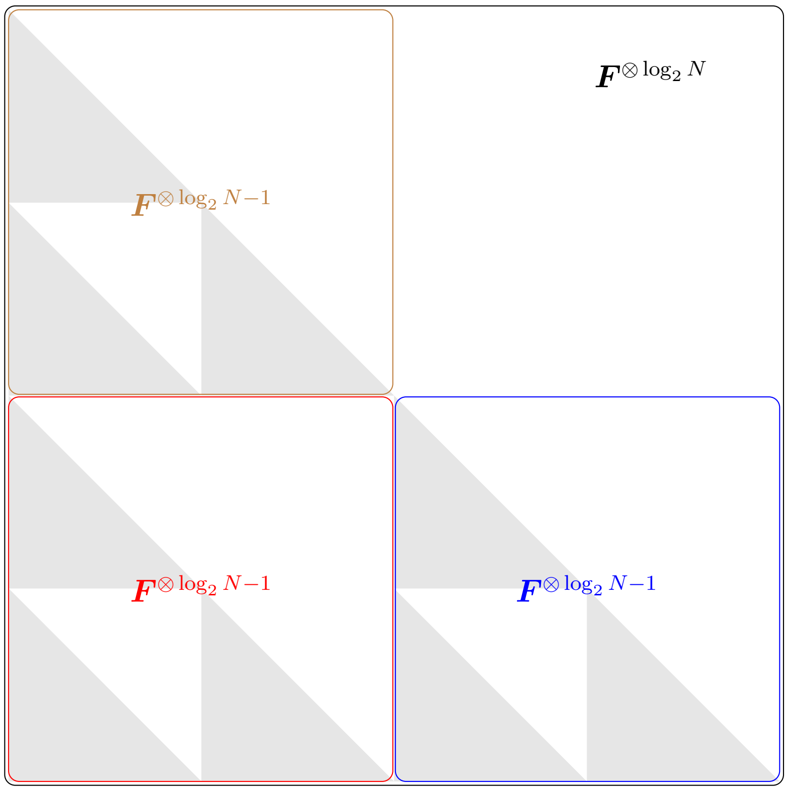

$\boldsymbol{F}^{\otimes \log_2 N}$
$\boldsymbol{F}^{\otimes \log_2 N-1}$
$\boldsymbol{F}^{\otimes \log_2 N-1}$
$\boldsymbol{F}^{\otimes \log_2 N-1}$

Figure 3.4.: Lower-diagonal matrix structure of $\boldsymbol{F}^{\otimes \log_2 N}$.

$$= \mathbb{I}\left(U^N; Y^N\right) \tag{3.30}$$

$$= \sum_{i=1}^{N} \mathbb{I}\left(U_i; Y^N \left| U^{i-1}\right.\right) \tag{3.31}$$

where $\mathbb{I}\left(U_i; Y^N \left| U^{i-1}\right.\right)$ is the *bit channel capacity* of the i-th bit channel $p_{Y^N U^{i-1}|U_i}$ with uniform input distribution.

Theorem 3.1. The transform $\boldsymbol{F}^{\otimes \log_2 N}$ converts a symmetric B-DMC to N *polarized* symmetric bit channels [5], i.e., for any $0 < a < b < 1$, we have

$$\begin{aligned}
&\lim_{N\to\infty} \frac{1}{N} \left|\left\{i : 0 \leq \mathbb{I}\left(U_i; Y^N \left| U^{i-1}\right.\right) < a\right\}\right| = 1 - \mathbb{I}\left(C; Y\right) \\
&\lim_{N\to\infty} \frac{1}{N} \left|\left\{i : a \leq \mathbb{I}\left(U_i; Y^N \left| U^{i-1}\right.\right) \leq b\right\}\right| = 0 \\
&\lim_{N\to\infty} \frac{1}{N} \left|\left\{i : b < \mathbb{I}\left(U_i; Y^N \left| U^{i-1}\right.\right) \leq 1\right\}\right| = \mathbb{I}\left(C; Y\right).
\end{aligned} \tag{3.32}$$

Example 3.2. Consider a BEC with $p = 0.5$, the bit channel capacities are recursively computed by DE with (3.24). A numerical example is shown in Figure 3.6. Observe that

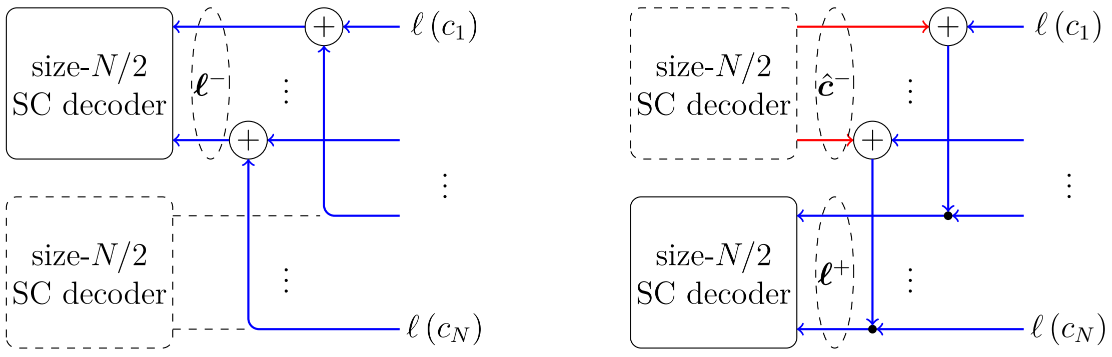


Figure 3.5.: Recursive structure of a size-N SC decoder.

almost all bit channel capacities are close to either 0 or 1 with growing code length N, i.e., almost every bit channel is either *completely noisy* or *noiseless*.

Definition 3.1. A polar code [5, 101] is defined by a 3-tuple $(N, k, \mathcal{A})$, where N and k denote the code length and the number of message bits. The elements of the information set $\mathcal{A}$ are the indices of the least noisy bit channels. The elements of the frozen set $\mathcal{F}$ are the indices of the noisiest bit channels. We have $|\mathcal{A}| = k$ and $[N] \setminus \mathcal{A} = \mathcal{F}$. The encoding procedure is given by

$$c^N = u^N \boldsymbol{F}^{\otimes \log_2 N} \tag{3.33}$$

where the vector u^N contains k message bits and $n-k$ frozen bits, i.e., we choose

$$u_{\mathcal{A}} = w^k, \quad u_{\mathcal{F}} = 0^{n-k}. \tag{3.34}$$

Algorithm 3.2 shows the SC decoding for a polar code, where

$$\ell\left(u_i \middle| \hat{u}^{i-1}\right) \triangleq \log \frac{P_{U_i|Y^N U^{i-1}}\left(0\middle|y^N, \hat{u}^{i-1}\right)}{P_{U_i|Y^N U^{i-1}}\left(1\middle|y^N, \hat{u}^{i-1}\right)} \tag{3.35}$$

is the LLR of u_i based on all channel outputs y^N and all previous estimates $\hat{u}^{i-1}$. The pseudo codes of an SC decoder are given in Section A.3. Note that the estimate $\hat{u}_{\mathcal{A}}$ is in a one-to-one correspondence with an estimate $\hat{c}^N$ because the transform $\boldsymbol{F}^{\otimes \log_2 N}$ is bijective.

Theorem 3.1 implies that polar codes achieve the capacity of symmetric[2] B-DMCs

[2]If the channel is *asymmetric*, then the polar coding introduced in Definition. 3.1 achieves only the mutual information with uniform input distributions. A modified polar code is introduced in [45, 73, 117] that achieves the capacity of asymmetric channels.

Algorithm 3.1: Size-N recursive SC decoding: $f_{\mathrm{sc},N}$

```
Input  : the LLRs of $c_i$ based on channel outputs $\ell(c_i)$, $i \in [N]$
Output: the estimated codeword $\hat{c}^N$
1  if $N = 1$ then
       /* perform hard decision for length-one code */
2      $\hat{u}_1 = \hat{c}_1 = \begin{cases} 0, & \text{if } \ell(c_1) \geq 0 \\ 1, & \text{if } \ell(c_1) < 0 \end{cases}$
3      return $\hat{c}_1$
4  else
       /* perform Plotkin-decomposition */
5      for $i = 1, 2, \ldots, N/2$ do
6          $\ell_i^- = f^-\left(\ell(c_i), \ell\left(c_{i+N/2}\right)\right)$
7      $\hat{\boldsymbol{c}}^- = f_{\mathrm{sc},N/2}(\boldsymbol{\ell}^-)$, where $\boldsymbol{\ell}^- = \left(\ell_1^-, \ldots, \ell_{N/2}^-\right)$
8      for $i = 1, 2, \ldots, N/2$ do
9          $\ell_i^+ = f^+\left(\ell(c_i), \ell\left(c_{i+N/2}\right), \hat{c}_i^-\right)$
10     $\hat{\boldsymbol{c}}^+ = f_{\mathrm{sc},N/2}(\boldsymbol{\ell}^+)$, where $\boldsymbol{\ell}^+ = \left(\ell_1^+, \ldots, \ell_{N/2}^+\right)$
11     return $\hat{c}^N = (\hat{\boldsymbol{c}}^- \oplus \hat{\boldsymbol{c}}^+, \hat{\boldsymbol{c}}^+)$
```

asymptotically under SC decoding. The complexity of encoding and SC decoding are both $\mathcal{O}(N \log N)$, where $\mathcal{O}(\cdot)$ is the Landau notation.

3.3. Code Constructions

3.3.1. Construction for a Fixed Channel

The polar code construction finds the most reliable positions of u^N under SC decoding. For BECs, simple DE can be used to compute the bit channel capacities as in Example 3.2. For general B-DMCs, a Monte Carlo (MC) based construction was introduced in [5, 101] but this requires extensive simulations. Quantized DE based constructions were proposed in [74, 75, 102]. One first computes the bit error probabilities of the bit channels assuming the previous bits were correctly decoded via MC or DE. Let

$$p_i \triangleq \Pr\left(\hat{U}_i \neq U_i \middle| \hat{U}^{i-1} = U^{i-1}\right), \quad i \in [N] \tag{3.36}$$

be the probability that the first bit error occurred for u_i in SC decoding. The information set $\mathcal{A}$ consists of the indices of the most reliable channels. The block error rate (BLER)

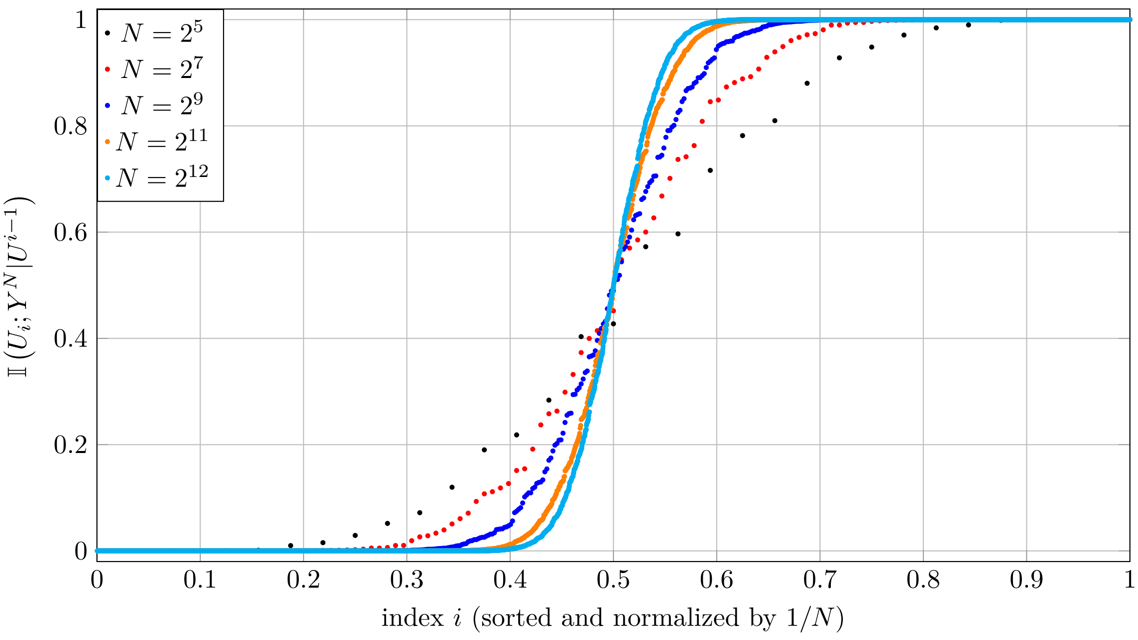


Figure 3.6.: Bit channel capacities on a BEC with $p = 0.5$.

of the polar code under SC decoding is estimated by

$$\begin{aligned}\text{BLER}_{\text{SC,est}} &= \Pr\left(\bigcup_{i\in\mathcal{A}} \left\{\hat{U}_i \neq U_i\right\}\right) \\ &= 1 - \Pr\left(\bigcap_{i\in\mathcal{A}} \left\{\hat{U}_i = U_i\right\}\right) \\ &\approx 1 - \prod_{i\in\mathcal{A}} (1 - p_i)\,. \end{aligned} \tag{3.37}$$

For biAWGN channels, DE with Gaussian approximation (GA) [108] has much lower complexity and performs similar to the MC construction. The update rule for mutual information of a single level transform is

$$\begin{aligned} I^- &= 1 - J\left(\sqrt{\left[J^{-1}(1-I_1)\right]^2 + \left[J^{-1}(1-I_2)\right]^2}\right) \\ I^+ &= J\left(\sqrt{\left[J^{-1}(I_1)\right]^2 + \left[J^{-1}(I_2)\right]^2}\right) \end{aligned} \tag{3.38}$$

where I^- and I^+ are the mutual information values after a single level evolution with inputs I_1 and I_2, see Figure 3.7. The J-function is given by

Algorithm 3.2: SC decoding for polar codes

```
  Input  : the LLRs of $c_i$ based on channel outputs $\ell(c_i),\ i \in [N]$
  Output: the estimated message bits $\hat{u}_i,\ i \in \mathcal{A}$
1 for $i = 1, 2, \ldots, N$ do
2     compute $\ell_i(\hat{u}^{i-1})$ // recursively computed via Algorithm 3.1
3     if $i \in \mathcal{A}$ then
4         $\hat{u}_i = \begin{cases} 0, & \text{if } \ell_i\left(\hat{u}^{i-1}\right) \geq 0 \\ 1, & \text{if } \ell_i\left(\hat{u}^{i-1}\right) < 0 \end{cases}$
5     else
6         $\hat{u}_i = 0$
7 return $\hat{u}_{\mathcal{A}}$
```

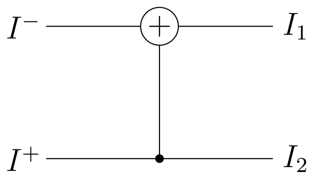


Figure 3.7.: The mutual information evolution of a single level transform.

$$J(\sigma) = 1 - \int_{-\infty}^{+\infty} \frac{e^{-\left(\xi - \sigma^2/2\right)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \cdot \log_2\left(1 + e^{-\xi}\right) \mathrm{d}\xi. \tag{3.39}$$

We use the numerical approximations in [15]:

$$\begin{aligned} J(\sigma) &\approx \left(1 - 2^{-H_1 \sigma^{2H_2}}\right)^{H_3} \\ J^{-1}(I) &\approx \left(-\frac{1}{H_1}\log_2\left(1 - I^{\frac{1}{H_3}}\right)\right)^{\frac{1}{2H_2}} \end{aligned} \tag{3.40}$$

where $H_1 = 0.3073$, $H_2 = 0.8935$ and $H_3 = 1.1064$, and we compute $\mathbb{I}(C;Y)$ via (2.34). By recursively applying (3.38), we obtain the bit channel capacity $\mathbb{I}\left(U_i; Y^N \left| U^{i-1}\right.\right)$. The corresponding probability can be derived from the bit channel capacity, i.e., we have

$$p_{i,\mathrm{GA}} = Q\left(\frac{1}{2}J^{-1}\left(\mathbb{I}\left(U_i; Y^N \middle| U^{i-1}\right)\right)\right) \approx p_i, \quad i \in [N] \tag{3.41}$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{\frac{u^2}{2}} \mathrm{d}u. \tag{3.42}$$

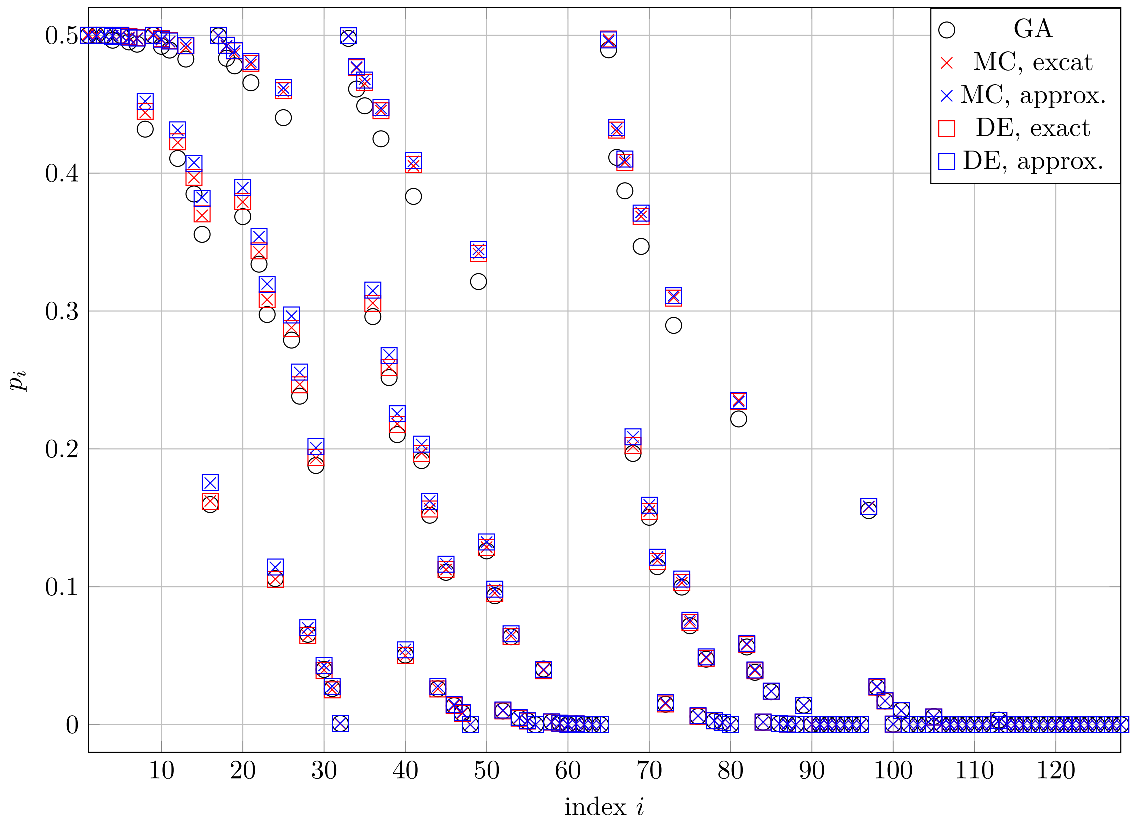


Figure 3.8.: Comparison of MC, DE and GA based constructions for $N = 128$ and a biAWGN channel at SNR = 2 dB.

Example 3.3. Consider a length-128 polar code transmitted over a biAWGN channel at SNR = 2 dB. The reliabilities of the bit channels under SC decoding are shown in Figure 3.8. We provide the results with the *exact* f^- operation (3.5) and the *approximated* f^- operation (3.9). We observe that

▷ The approximated f^- operations are conservative (give higher p_i). However, the *reliability order* is the same.

▷ Quantized DE closely approximates the reliabilities of the optimal DE.

▷ The GA construction gives the same reliability order as MC and DE, i.e., the GA construction outputs the same information set $\mathcal{A}$ as MC and DE.

3.3.2. Universal Construction

The structure of $\boldsymbol{F}^{\otimes \log_2 N}$ leads to two *universal* (channel quality independent) properties of the bit channels:

- ▷ nesting property [58] (asymptotic);
- ▷ universal partial order (UPO) [97] (non-asymptotic).

Theorem 3.2. Consider two B-DMCs $p_{Y_{\mathrm{I}}|C}$ and $p_{Y_{\mathrm{II}}|C}$ where $p_{Y_{\mathrm{II}}|C}$ is degraded with respect to $p_{Y_{\mathrm{I}}|C}$, i.e., we have $p_{Y_{\mathrm{II}}|C} \preceq p_{Y_{\mathrm{I}}|C}$. By applying a transform $\boldsymbol{F}^{\otimes \log_2 N}$ we have

$$p_{Y_{\mathrm{II}}^N U^{i-1}|U_i} \preceq p_{Y_{\mathrm{I}}^N U^{i-1}|U_i}, \quad i \in [N]. \tag{3.43}$$

With an infinite code length N, we have the sets $\mathcal{A}_{Y_{\mathrm{I}}}$ and $\mathcal{A}_{Y_{\mathrm{II}}}$ containing the indices of the noiseless bit channels for $p_{Y_{\mathrm{I}}|C}$ and $p_{Y_{\mathrm{II}}|C}$, respectively. For any index i, we have

- ▷ The bit channel $p_{Y_{\mathrm{I}}^N U^{i-1}|U_i}$ is noiseless if $p_{Y_{\mathrm{II}}^N U^{i-1}|U_i}$ is noiseless.
- ▷ The bit channel $p_{Y_{\mathrm{II}}^N U^{i-1}|U_i}$ is completely noisy if $p_{Y_{\mathrm{I}}^N U^{i-1}|U_i}$ is completely noisy.

Therefore, we have the nesting property [58], i.e., $\mathcal{A}_{Y_{\mathrm{II}}} \subseteq \mathcal{A}_{Y_{\mathrm{I}}}$.

Remark. This property leads to a universal reliability ordering of the polarized bit channels. However, the nesting property generally does not hold for finite code lengths.

Theorem 3.3. Consider a B-DMC and a size-N transform, and label the bit channel i with the binary representation of $i-1$:

$$i \mapsto \left(b_1, \ldots, b_{\log_2 N}\right). \tag{3.44}$$

Then we have the following properties called UPO [97]:

- ▷ the bit channel $(a_1, 1, a_2)$ is more reliable than $(a_1, 0, a_2)$
- ▷ the bit channel $(a_1, 1, a_2, 0, a_3)$ is more reliable than $(a_1, 0, a_2, 1, a_3)$

where a_1, a_2 and a_3 are any bit strings including empty strings.

Example 3.4. For a polar code with $N = 4$, we have

$$\begin{cases} i = 1 \mapsto (0,0) \\ i = 2 \mapsto (0,1) \\ i = 3 \mapsto (1,0) \\ i = 4 \mapsto (1,1) \end{cases} \tag{3.45}$$

Theorem 3.3 states that for any B-DMC we have

- ▷ The bit channel $2 \mapsto (0,1)$ is more reliable than $1 \mapsto (0,0)$.
- ▷ The bit channel $3 \mapsto (1,0)$ is more reliable than $2 \mapsto (0,1)$.
- ▷ The bit channel $4 \mapsto (1,1)$ is more reliable than $3 \mapsto (1,0)$.

There thus exists only one possible ascending reliability order:

$$(1,2,3,4)\,. \tag{3.46}$$

However, more generally the reliabilities are not determined by the UPO. For example, consider $N = 8$ and

$$4 \mapsto (0,1,1) \text{ and } 5 \mapsto (1,0,0)\,. \tag{3.47}$$

Definition 3.2. Consider a length-N polar code and label bit channel i by the binary representation of $i-1$:

$$i \mapsto \left(b_1, \ldots, b_{\log_2 N}\right). \tag{3.48}$$

The polarization weight (PW) [41] of bit channel i is defined as

$$\mathrm{PW}_i = \sum_{j=1}^{\log_2 N} b_j \beta^j, \quad i \in [N] \tag{3.49}$$

where β is a real number with $\beta > 1$.

This notation is called a β-expansion [81,86]. PWs fulfill the properties in Theorem 3.3 and we use them to find a universal reliability order by carefully choosing β. An asymptotically optimal β for SC decoding is [41]

$$\beta = 2^{\frac{1}{4}} \approx 1.1892. \tag{3.50}$$

Remark. The nesting property and the UPO generally do not hold for (non-stationary) *time-varying* channels.

3.4. Decoding Latency and Memory Requirements

The decoding latency describes the required number of time steps assuming that all parallelizable operations are completed at the same time step. The serial nature of SC decoding

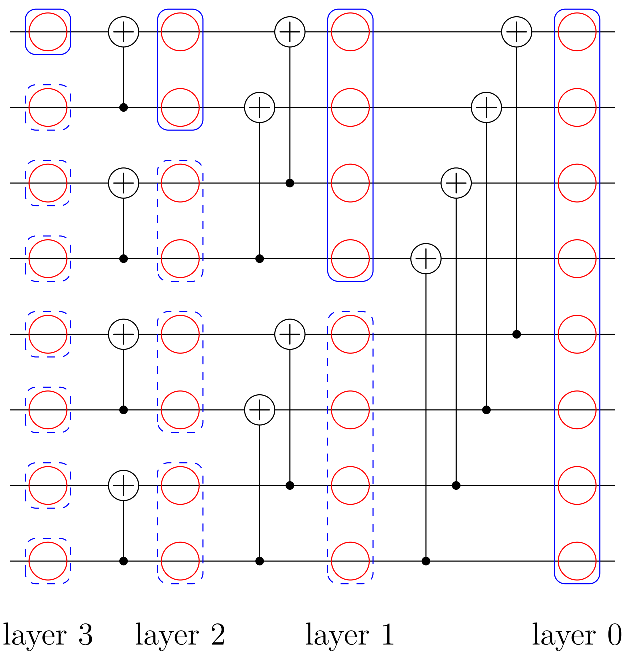


Figure 3.9.: Memory requirement for an SC decoder of length 8.

makes the latency scale as $\mathcal{O}(N)$. Simplified SC decoding is proposed in [3], where one performs Plotkin-decomposition (see Figure 3.5) recursively up to code length $N = 1$ and makes hard decisions. With simplified SC decoding, we make hard decisions for special codes whose code lengths are larger than one. Four such codes[3] are suggested in [3]: rate-0 code, rate-1 code, repetition code, and single parity-check code. The algorithm is discussed and improved in [72, 92–94]. The authors of [72] show that the latency of simplified SC decoding is $\mathcal{O}\left(N^{1-1/\mu}\right)$, where μ is the scaling exponent of the channel.

An SC decoder of length N must store computations at $N \cdot (\log_2 N + 1)$ nodes (the red circles in Figure 3.9). A memory-efficient SC decoding is proposed in [103]. Consider layer 1 in Figure 3.9. To compute the LLRs in the (lower) dashed blue block one does not need the LLRs of the (upper) solid blue block. Thus, only $N/2^j$ nodes are needed at layer j. The memory requirement of an SC decoder is thus reduced to $2N - 1$ (the solid blue blocks in Figure 3.9).

3.5. Improved Decoding Algorithms

Although polar codes achieve capacity asymptotically under SC decoding, the finite length performance of polar codes with SC decoding is not competitive. This section introduces

[3]The suggested codes have simple and optimal decoding algorithms.

three classes of improved decoding algorithms.

3.5.1. List Decoding

The authors of [28, 103] propose successive cancellation list (SCL) decoding that improves the performance of SC decoding by deploying L parallel SC decoding paths. At each decoding phase $i \in \mathcal{A}$, instead of performing a hard decision on u_i, we create two decoding paths and continue decoding in two parallel threads, i.e., we have 2^k decoding paths at the end of decoding. In order to avoid the exponential growth of the number of decoding paths, only the L most reliable paths survive at each step. At the end of decoding, the most reliable path is selected as the output. The reliability of a length-i decoding path v^i is described by $P_{U^i|Y^N}\left(v^i\left|y^N\right.\right)$ and we have the recursive update rule

$$P_{U^i|Y^N}\left(v^i\left|y^N\right.\right) = P_{U_i|Y^N U^{i-1}}\left(v_i\left|y^N, v^{i-1}\right.\right) P_{U^{i-1}|Y^N}\left(v^{i-1}\left|y^N\right.\right) \tag{3.51}$$

where the reliability is initialized to 1, i.e., $P_{U^0|Y^N}\left(v^0\left|y^N\right.\right) = 1$. An LLR-based path metric (PM) [7] is defined by

$$\begin{aligned} M\left(v^i\right) &\triangleq -\log P_{U^i|Y^N}\left(v^i\left|y^N\right.\right) && (3.52)\\ &= -\log P_{U_i|Y^N U^{i-1}}\left(v_i\left|y^N, v^{i-1}\right.\right) - \log P_{U^{i-1}|Y^N}\left(v^{i-1}\left|y^N\right.\right) && (3.53)\\ &= -\log P_{U_i|Y^N U^{i-1}}\left(v_i\left|y^N, v^{i-1}\right.\right) + M\left(v^{i-1}\right). && (3.54)\end{aligned}$$

From the definition of $\ell\left(u_i\left|v^{i-1}\right.\right)$ in (3.35), we have

$$\begin{aligned} -\log P_{U_i|Y^N U^{i-1}}\left(0\left|y^N, v^{i-1}\right.\right) &= \log\left(1 + e^{-\ell\left(u_i\left|v^{i-1}\right.\right)}\right)\\ -\log P_{U_i|Y^N U^{i-1}}\left(1\left|y^N, v^{i-1}\right.\right) &= \log\left(1 + e^{\ell\left(u_i\left|v^{i-1}\right.\right)}\right). \end{aligned} \tag{3.55}$$

Therefore, the recursive PM update rule is

$$M\left(v^i\right) = M\left(v^{i-1}\right) + \log\left(1 + e^{-(1-2v_i)\ell\left(u_i\left|v^{i-1}\right.\right)}\right) \tag{3.56}$$

where $M\left(v^0\right) = -\log\left(1\right) = 0$. By applying the approximation in (3.8), we have a hardware friendly version [7],

$$M\left(v^i\right) \approx \begin{cases} M\left(v^{i-1}\right), & \text{if } \operatorname{sign}\left(\ell\left(u_i \middle| v^{i-1}\right)\right) = 1 - 2v_i \\ M\left(v^{i-1}\right) + \left|\ell\left(u_i \middle| v^{i-1}\right)\right|, & \text{if } \operatorname{sign}\left(\ell\left(u_i \middle| v^{i-1}\right)\right) \neq 1 - 2v_i \end{cases}. \tag{3.57}$$

The complexity of a direct implementation of an SCL decoder is $\mathcal{O}\left(LN^2\right)$. In [103], a *lazy copy* technique based on memory sharing is proposed to reduce the number of copy operations. The complexity of SCL decoding is thereby reduced to $\mathcal{O}\left(LN\log N\right)$.

3.5.2. Flip Decoding

The serial nature of SC decoding may cause an erroneous bit decision through error propagation. The main idea of successive cancellation flip (SCF) decoding [2] is to try to correct the first erroneous bit decision by sequentially flipping unreliable decisions.

An error detection outer code, e.g., a cyclic redundancy check (CRC) code, checks whether the output is a valid codeword. The SCF decoder starts by performing SC decoding for the inner code to generate the first estimate v^N and stores the soft information $\ell\left(u_i \middle| v^{i-1}\right)$, $i \in [N]$. If v^N passes the CRC, it is declared as the output $\hat{u}^N$. In case the CRC fails, the SCF algorithm attempts to correct the first bit error at most T times. At the t-th attempt, $t \in [T]$, the decoder finds the index i_t of the t-th least reliable decision. The SCF algorithm restarts the SC decoder by flipping the estimate v_{i_t} to $\overline{v_{i_t}} = v_{i_t} \oplus 1$. The CRC is checked after each attempt. This decoding process continues until the CRC passes or T is reached.

In [2], the authors suggest to used metric

$$Q_1(i) = \left|\ell\left(u_i \middle| v^{i-1}\right)\right|, \quad i \in \mathcal{A} \tag{3.58}$$

to describe the reliability of the decision v_i. The flipping position is selected by

$$i^* = \operatorname*{argmin}_{i\in[N]} Q_1(i). \tag{3.59}$$

Because the target is to find the *first* erroneous decision, the reliabilities of the previous decisions should be taken into consideration. A new metric is proposed in [16] to find the first erroneous decision more efficiently

$$Q_2(i) = \left|\ell\left(u_i \left| v^{i-1}\right.\right)\right| + \sum_{\substack{j\in\mathcal{A}\\ j\le i}} \frac{1}{\alpha}\log\left(1+e^{-\alpha\left|\ell\left(u_j\left|v^{j-1}\right.\right)\right|}\right), \quad i\in\mathcal{A} \tag{3.60}$$

where $\alpha > 0$ is a scaling factor to be optimized by MC simulation.

In [16], dynamic successive cancellation flip (DSCF) decoding is introduced as a generalization of SCF decoding. DSCF decoding finds and flips multiple bit estimates simultaneously. The reliability of the decisions $\hat{u}_{\mathcal{E}}$ is described by

$$Q_2(\mathcal{E}) = \sum_{i\in\mathcal{E}}\left|\ell\left(u_i \left| v^{i-1}\right.\right)\right| + \sum_{\substack{j\in\mathcal{A}\\ j\le i_{\max}}} \frac{1}{\alpha}\log\left(1+e^{-\alpha\left|\ell\left(u_j\left|v^{j-1}\right.\right)\right|}\right), \quad \mathcal{E}\subseteq\mathcal{A} \tag{3.61}$$

where $i_{\max}$ is the largest element in $\mathcal{E}$. The flipping set is selected by

$$\mathcal{E}^* = \operatorname*{argmin}_{\mathcal{E}} Q_2(\mathcal{E}) \tag{3.62}$$

and is constructed progressively. A hardware friendly version of (3.61) is introduced in [26]:

$$Q_3(\mathcal{E}) = \sum_{i\in\mathcal{E}}\left|\ell\left(u_i \left| v^{i-1}\right.\right)\right| + \sum_{\substack{j\in\mathcal{A},\ j\le i_{\max}\\ \left|\ell\left(u_j\left|v^{j-1}\right.\right)\right|<\beta}} \left(\beta - \left|\ell\left(u_j\left|v^{j-1}\right.\right)\right|\right), \quad \mathcal{E}\subseteq\mathcal{A} \tag{3.63}$$

where $\beta > 0$ is a perturbation factor to be optimized by MC simulation. The complexity of DSCF decoding is adaptive and upper bounded by $\mathcal{O}\left(\left(1+T\right)N\log N\right)$. The latency of DSCF decoding is not stable.

3.5.3. Sequential Decoding

Sequential decoding does not deploy L parallel decoding paths and thereby avoids constructing many low probability paths in the decoding tree. Two prominent sequential decoding algorithms are successive cancellation sequential (SCS) decoding [71, 110] and successive cancellation Fano (SC-Fano) decoding [6, 50].

Sequential decoding compares paths with different lengths. However, the probabilities $P_{U^i|Y^N}\left(v^i\left|y^N\right.\right)$ cannot capture the effect of the path's length. This effect was taken into account first by [110]. Similar approach was used by [50] to account for the expected error rate of the future bits as

$$S\left(v^i\right) \triangleq -\log \frac{P_{U^i|Y^N}\left(v^i \middle| y^N\right)}{\prod_{j=1}^{i}\left(1-p_j\right)} \tag{3.64}$$

$$= M\left(v^i\right) + \sum_{j=1}^{i} \log\left(1-p_j\right) \tag{3.65}$$

where p_i is defined by (3.36) and $S\left(v^0\right) \triangleq 0$. The probabilities p_i can be computed via MC or approximated via DE offline.

SCS decoding stores the L most reliable paths of (possibly) different length and discards paths as needed. At each step, the decoder selects the most reliable path and creates two possible decoding paths based on it. The winning word is declared once a path length becomes N. SC-Fano decoding deploys a Fano search [33] that allows to move backward in the decoding tree. The decoder tries to find the most reliable path with the help of a dynamic threshold. The dynamic threshold is initialized to $T = 0$. During the Fano search, if one cannot find a path with score less than T then the dynamic threshold is updated to $T + \Delta$, where Δ is called the threshold spacing and controls the tradeoff between performance and complexity.

Both SCS and SC-Fano decoding are complexity-adaptive, i.e., their average decoding complexity is close to that of SC decoding for reliable channels. However, the single thread nature of SCS and SC-Fano decoding can make the latency unstable. Furthermore, SCS decoding needs a large list size to achieve the same performance as SCL decoding. Also, SC-Fano decoding has a relatively high worst case complexity.

3.6. Improved Code Designs

The polar code construction discussed in Section 3.3 finds the most reliable positions under SC decoding, i.e., the original code construction is optimal for SC decoding only. Moreover, for short and moderate code lengths, polar codes with improved decoding algorithms still perform worse than existing codes, e.g., Turbo and LDPC codes, because of their poor distance properties. This section introduces some improved code designs.

In [103], the distance property of polar codes is enhanced by serially concatenating a CRC outer code. We proposed an algorithm in [123] to find the good CRC generator polynomial for a given CRC length, the details are discussed in Section A.1.

RM-polar [57] codes combine the code constructions of RM codes and polar codes. Observe that RM and polar codes are obtained from the matrix $\boldsymbol{F}^{\otimes \log_2 N}$. While polar codes select information bits according to the bit reliability under SC decoding, RM codes [77,84] select the information bits according to the row weight. The bits with the largest weights of their corresponding rows are selected as information bits, and the other bits are chosen as frozen bits. The construction of RM-polar codes sacrifices some reliable bits under SC decoding in order to guarantee better distance properties. In [30, 31], a genetic algorithm is proposed to obtain a balance between RM and polar code constructions.

In [111, 112], *dynamic frozen bits* are proposed to improve the distance properties. Such bits are defined as linear combinations of previous message bits instead of predetermined values, and thus using dynamic frozen bits does not change the performance under SC decoding. A method to construct polar codes with guaranteed distance properties based on dynamic frozen bits is proposed in [112]. The main idea is to force polar codes to be subcodes of the higher rate codes with known distance properties. Dynamic frozen bits with random linear combinations are discussed in [21, 59, 60, 115, 123]. The polarization-adjusted convolutional (PAC) codes proposed in [6] are based on dynamic frozen bits with convolutional linear combinations. For instance, suppose we have a length-d vector $\boldsymbol{g}$. Then the (dynamic) frozen bit u_i is computed via

$$u_i = \bigoplus\!\!\!\!\!\sum_{j=1}^{d} g_j u_{i-j}, \text{ if } i \in \mathcal{F} \text{ and } i - j \geq 1 \tag{3.66}$$

where $\bigoplus\!\!\!\!\!\sum$ denotes an XOR sum.

To conclude, two methods are jointly used to improve the distance property of polar codes, namely changing the information set and using dynamic frozen bits. These methods tradeoff the performance under SC decoding (determined by the choice information set) and ML decoding (determined by the distance properties). The best design of polar codes strongly depends on the target decoding complexity [21, 22, 90, 123].

3.7. Rate Adaptation

The recursive structure of the transform $\boldsymbol{F}^{\otimes \log_2 N}$ gives polar codes a code length that is a power of two. The classic techniques of *puncturing* and *shortening* can be used to modify the code length and rate.

Punctured polar codes were introduced in [32]. Consider a length-N mother polar code and suppose the coded bits $c_{\mathcal{P}}$ are not transmitted, where $\mathcal{P}$ is the *puncturing pattern*. The effective code length is thus $N - |\mathcal{P}|$. The receiver uses the decoder of the mother polar code where the LLRs $\ell\,(c_i)$, $i \in \mathcal{P}$, are set to zero.

Shortened polar codes are proposed in [114]. Consider a length-N mother polar code and fix the coded bits $c_{\mathcal{S}}$ to zeros, where $\mathcal{S}$ is a *shortening pattern.* These coded bits no longer need to be transmitted and the effective code length becomes $N - |\mathcal{S}|$. The receiver uses the decoder of the mother polar code where the LLRs $\ell(c_i)$, $i \in \mathcal{S}$, are set to infinity because the decoder knows that $c_i = 0$, $i \in \mathcal{S}$.

Puncturing and shortening cause time-varying B-DMCs, i.e., the coded bits are transmitted over channels with various qualities. The universal properties introduced in Section 3.3.2 thus do not hold. If we consider punctured and shortened polar codes constructed by GA, then we set the channel mutual information as follows:

$$\begin{aligned} \mathbb{I}(C_i; Y_i) &= 0, \text{ if } i \in \mathcal{P} \\ \mathbb{I}(C_i; Y_i) &= 1, \text{ if } i \in \mathcal{S}. \end{aligned} \tag{3.67}$$

So far in literature, there is no method to find the optimal puncturing and shortening patterns efficiently. A low complexity suboptimal algorithm is proposed in [70]. The quasi-uniform puncturing (QUP) and reversal quasi-uniform puncturing (RQUP) are introduced in [78, 114] and have been adopted for enhanced mobile broadband (eMBB) in the 5-th generation wireless system (5G) standard [12].

Definition 3.3. An (n, k, N) QUP polar code with length-n and dimension k is obtained from a (N, k) mother polar code by using the puncturing pattern $\mathcal{P} = [N - n]$, i.e., the first $N - n$ coded bits are not transmitted.

Definition 3.4. A length-n RQUP polar code is obtained from a length-N mother polar code by using the shortening pattern $\mathcal{S} = \{n+1, \dots, N\}$, i.e., we set

$$u_i = 0, \;\; n+1 \le i \le N. \tag{3.68}$$

Because $\boldsymbol{F}^{\otimes \log_2 N}$ is a lower triangular matrix, the last $N - n$ coded bits c_{n+1}^N are fixed to zeros and not transmitted.

Practically, RQUP polar codes outperform QUP polar codes if the code rate is higher than 0.5, and vice versa.

Remark. The QUP and RQUP methods are generally suboptimal but perform close to the best patterns that we found. The algorithms are called quasi-uniform because the punctured/shortened positions in a bit-reversal representation [5] are almost uniformly distributed on $[N]$.

Algorithm 3.3: Encoding procedure of MLPC.

```
Input  : message bits $w^k$
         information set $\mathcal{A}$
Output: modulated signal points $x^N$
  /* Put $k$ message bits in a vector of length $mN$ */
1 $u_{\mathcal{A}} = w^k,\ u_{[mN]\setminus\mathcal{A}} = 0^{mN-k}$
  /* partition the vector into $m$ binary vectors of length $N$ */
2 for $j = 1, 2, \ldots, m$ do
3     $u[j]^N = u_{(j-1)N+1}^{jN}$
  /* perform $m$ parallel transforms */
4 for $j = 1, 2, \ldots, m$ do
5     $c[j]^N = u[j]^N \boldsymbol{F}^{\otimes \log_2 N}$
  /* map the coded bits to signal point with SP labeling */
6 for $i = 1, 2, \ldots, N$ do
7     $(c[1]_i, c[2]_i, \ldots, c[m]_i) \overset{\text{SP}}{\mapsto} x_i$
8 return $x^N$
```

3.8. Polar-Coded Modulation

Higher-order modulation increases the SE of coded modulation systems. Consider a 2^m-ary modulation with binary codes. Each signal point x in the constellation $\mathcal{X}$ is uniquely addressed by the label $(b_1, \ldots, b_m)$ in the manner

$$(b_1, \ldots, b_m) \mapsto x,\ x \in \mathcal{X},\ |\mathcal{X}| = 2^m. \tag{3.69}$$

Several polar-coded modulation (PCM) schemes are discussed in [64, 98, 107] and their performance is compared in [14]. Efficient design algorithms are presented in [13, 14]. In [98], multilevel polar coding (MLPC) with set partitioning (SP) labeling is proposed and provides the best performance. A MLPC system with output length N consists of m component polar codes of length N. We denote the input and output of the j-th transform by $u[j]^N$ and $c[j]^N$ for $j \in [m]$, respectively. Algorithm 3.3 shows the encoding procedure of MLPC.

The receiver applies multi-stage decoding [48, 113], see Figure 3.10. The codeword $c[j]^N$ is effectively transmitted over the binary-input channel $P_{B_j|YB^{j-1}}$. One successively performs demodulation and decoding as in Algorithm 3.4 where the level-j demodulating

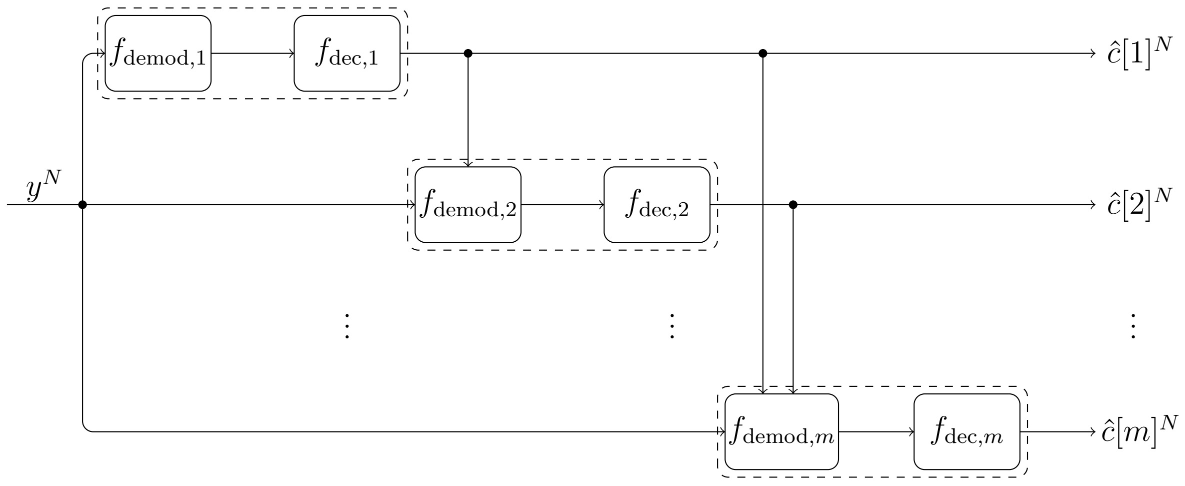


Figure 3.10.: Multi-stage decoding for MLPC.

Algorithm 3.4: Decoding procedure of MLPC.

```
Input  : channel outputs $y^N$
Output: decoded codewords $\hat{c}[j]^N,\ j \in [m]$
1 for $j = 1, 2, \dots, m$ do
      /* demodulation */
2     for $i = 1, 2, \dots, N$ do
3         $\ell(c[j]_i) = f_{\text{demod},j}(y_i, \hat{c}[1]_i, \dots, \hat{c}[j-1]_i)$
      /* SC decoding */
4     $\hat{c}[j]^N = f_{\text{dec},j}(\ell(c[j]_1), \ell(c[j]_2), \dots, \ell(c[j]_N))$
5 return $\hat{c}[j]^N,\ j \in [m]$
```

function is

$$f_{\text{demod},j}(y_i, \hat{c}[1]_i, \dots, \hat{c}[j-1]_i) = \log \frac{P_{B_j|YB^{j-1}}(0|y_i, \hat{c}[1]_i, \dots, \hat{c}[j-1]_i)}{P_{B_j|YB^{j-1}}(1|y_i, \hat{c}[1]_i, \dots, \hat{c}[j-1]_i)},\ i \in [N]. \quad (3.70)$$

We have

$$\mathbb{I}(X;Y) = \mathbb{I}(B_1 B_2 \cdots B_m; Y) = \sum_{j=1}^{m} \mathbb{I}\left(B_j \middle| Y B^{j-1}\right). \quad (3.71)$$

Thus, the SE of the system k/N achieves $\mathbb{I}(X;Y)$ for a uniform input distribution.

The MLPC is often design by MC methods [98]. In [14], we proposed a surrogate channel based GA to find the information set $\mathcal{A}$ for MLPC efficiently, see Algorithm 3.5.

SCF, SCS and SC-Fano decoding can be directly applied to a MLPC system. Figure 3.11

Algorithm 3.5: Surrogate channel based GA.

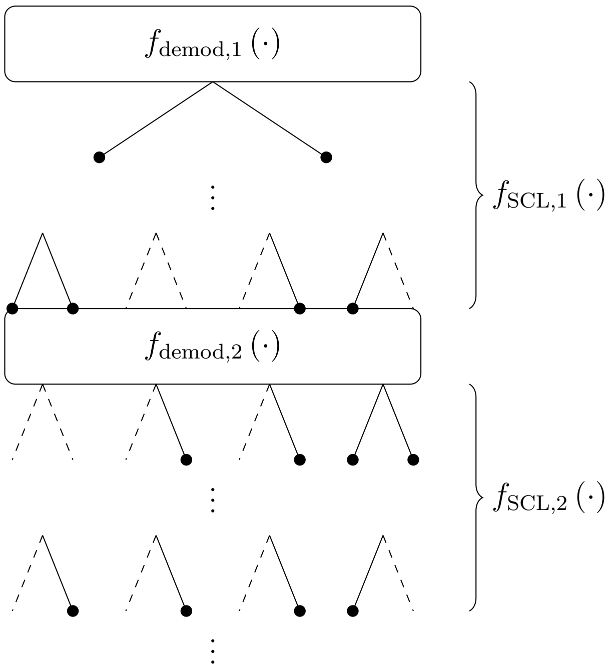
```
Input  : channel $p_{Y|X}$
         modulation order $m$ and constellation $\mathcal{X}$
         labeling $(b_1, \ldots, b_m) \mapsto x,\ x \in \mathcal{X}$
         component code length $N$
Output: information set $\mathcal{A}$
1 compute $\mathbb{I}\,(B_j|YB^{j-1})$
2 for $j = 1, 2, \ldots, m$ do
      /* GA for the j-th component code */
3     compute $\mathbb{I}\left(U_i; Y^N\,|U^{i-1}\right)$ via GA based on $\mathbb{I}\,(B_j\,|YB^{j-1})$ for $(j-1)N < i \le jN$
4 put the indices $i$ of the $k$ largest $\mathbb{I}\left(U_i; Y^N\,|U^{i-1}\right)$ in set $\mathcal{A}$
5 return $\mathcal{A}$
```

shows the SCL decoding for MLPC. The component decoders inherit all decoding paths and their PMs from the previous decdoing levels. One proceeds as follows.

▷ After SCL decoder $f_{\mathrm{SCL},j-1}$ completes its computations, all active paths are passed to the demodulator $f_{\mathrm{demod},j}$. The corresponding PMs of the paths are passed to the decoder $f_{\mathrm{SCL},j}$.

▷ The demodulator $f_{\mathrm{demod},j}$ computes the LLRs for all possible paths and gives them to the decoder $f_{\mathrm{SCL},j}$.

▷ Initialize the SCL decoder $f_{\mathrm{SCL},j}$ with the LLRs for all possible paths and the corresponding PMs from the previous decoder.



$f_{\mathrm{demod},1}(\cdot)$
$f_{\mathrm{SCL},1}(\cdot)$
$f_{\mathrm{demod},2}(\cdot)$
$f_{\mathrm{SCL},2}(\cdot)$

Figure 3.11.: SCL decoding tree for MLPC ($L = 4$).

4

Complexity-Adaptive Decoding of Polar Codes

This chapter proposes *successive cancellation ordered search (SCOS)* decoding [121] as a complexity-adaptive ML decoder for polar codes. The decoder is extended to limit the worst-case complexity while still achieving near-ML performance.

4.1. Successive Cancellation Ordered Search Decoding

SCOS decoding borrows ideas from SC-based flip [2,16], sequential [33,50,71,110] and list decoders [27,35,103,118]. It is a tree search algorithm that flips the bits of valid paths to find a leaf (i.e., a path of length-N) with higher likelihood than other leaves, if such a leaf exists, and repeats until the ML decision is found. The search stores a list of branches that is updated progressively by flipping the bits of the most likely leaf at each iteration. The order of the candidates is decided according to the probability that they provide the ML decision. SCOS does not require an outer code (as for flip-decoders) or parameter optimization to adjust the tradeoff between performance and complexity (as for sequential decoders).

In the proposed SCOS algorithm, metrics (3.52) and (3.64) are used. We first define

$$\overline{M}\left(v^i\right) \triangleq M\left(v^{i-1}\overline{v_i}\right) = -\log P_{U^i|Y^N}\left(v^{i-1}\overline{v_i}\middle|y^N\right) \tag{4.1}$$

$$\overline{S}\left(v^i\right) \triangleq S\left(v^{i-1}\overline{v_i}\right) = -\log P_{U^i|Y^N}\left(v^{i-1}\overline{v_i}\,\middle|y^N\right) + \sum_{j=1}^{i} \log\left(1-p_j\right) \tag{4.2}$$

An exemplary decoding process is illustrated in Figure 4.1 for $N = 4$ (also $K = 4$). SCOS decoding starts by SC decoding to provide an output v^N as the current most likely leaf, e.g., the black path (0111) in Figure 4.1. The initial SC decoding computes and stores the PM $\overline{M}\left(v^i\right)$ and the score $\overline{S}\left(v^i\right)$ associated with the flipped versions of the decisions v_i for all $i \in \mathcal{A}$, e.g., illustrated as the red paths in Figure 4.1. Every index $i \in \mathcal{A}$ with $\overline{M}\left(v^i\right) < M(v^N)$ is a flipping set.[1] The collection of all flipping sets forms a list $\mathcal{L}$. Each list member is visited in ascending order according to the score associated with it.

Upon deciding on a flipping set $\mathcal{E}$ in the list $\mathcal{L}$, let index $j \in [N]$ be the deepest common node of the current most likely leaf and the branch node defined by $\mathcal{E}$ in the decoding tree (see the brown dot in Figure 4.1(a)). Then, the decoder flips the decision v_j and SC decoding continues. The set $\mathcal{E}$ is popped from the list $\mathcal{L}$. The PMs (3.52) and scores (3.64) are calculated again for the flipped versions for decoding phases with $i > j$, $i \in \mathcal{A}$, and the list $\mathcal{L}$ is enhanced by new flipping sets progressively (similar to [16]). The branch node, including all of its child nodes, is discarded if at any decoding phase its PM exceeds that of the current most likely leaf, i.e., $M(v^N)$.[2] Such a branch cannot output the ML decision, since for any valid path v^i the PM (3.52) is non-decreasing for the next stage, i.e., we have

$$M\left(v^i\right) \leq M\left(v^{i+1}\right), \forall v_{i+1} \in \{0,1\}. \tag{4.3}$$

For instance, suppose that $M(11) > M(0111)$ in Figure 4.1(g). Then, any path $\tilde{v}^N$, with $\tilde{v}^2 = (1,1)$ cannot be the ML decision; hence, it is pruned. If a leaf with lower PM is found, then it replaces the current most likely leaf. The procedure is repeated until it is impossible to find a more reliable path by flipping decisions, i.e., until $\mathcal{L} = \varnothing$. Hence, SCOS decoding implements an ML decoder.

Remark. The score (3.64) dominates the search priority of SCOS, while the tree pruning and final decisions are based on the PM (3.52).

[1] Each set is a singleton at this stage.

[2] This pruning method is similar to the adaptive skipping rule proposed in [118] for ordered-statistics decoding [27, 35].

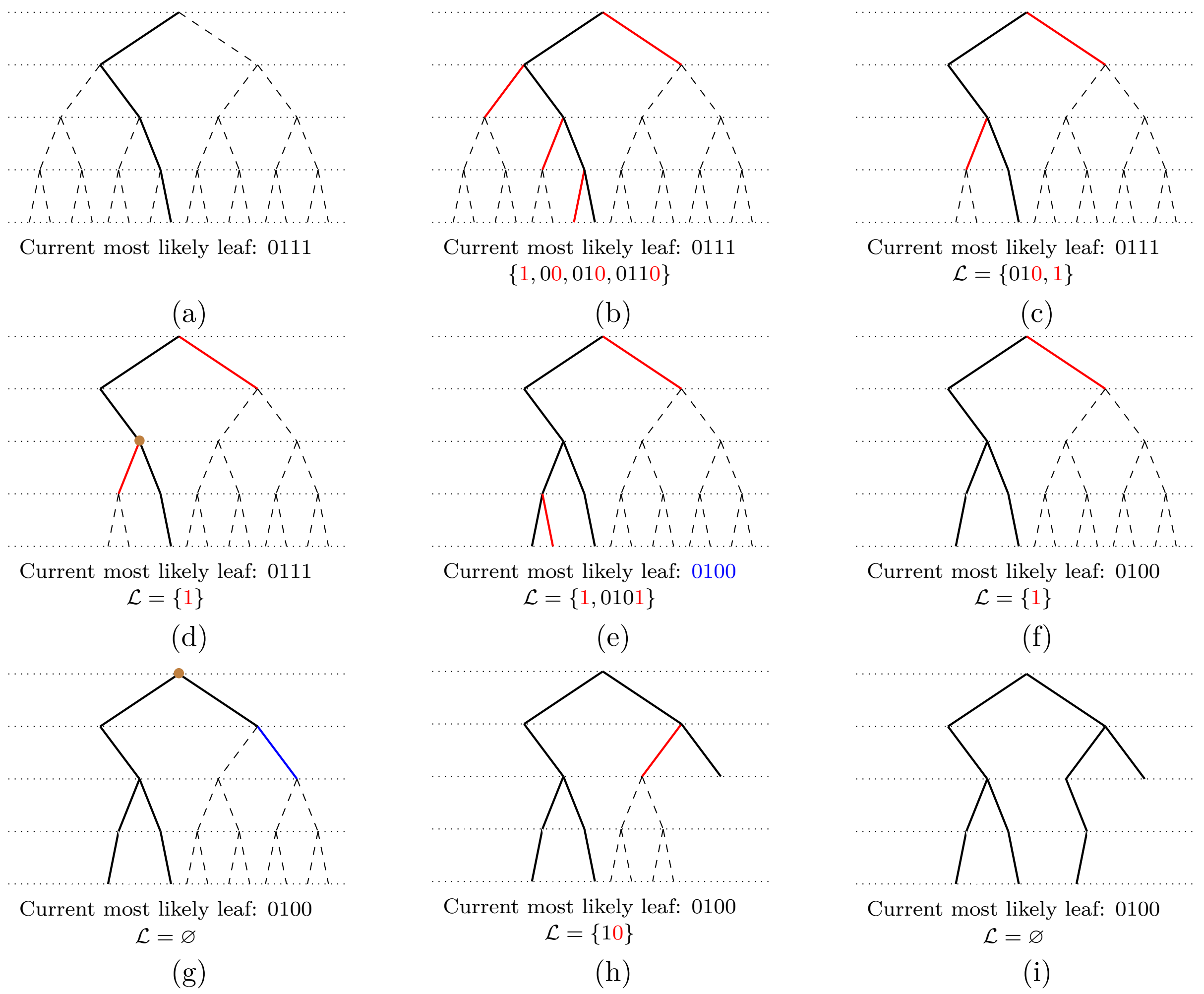

Current most likely leaf: 0111
(a)
Current most likely leaf: 0111
$\{1, 00, 010, 0110\}$
(b)
Current most likely leaf: 0111
$\mathcal{L} = \{010, 1\}$
(c)
Current most likely leaf: 0111
$\mathcal{L} = \{1\}$
(d)
Current most likely leaf: 0100
$\mathcal{L} = \{1, 0101\}$
(e)
Current most likely leaf: 0100
$\mathcal{L} = \{1\}$
(f)
Current most likely leaf: 0100
$\mathcal{L} = \varnothing$
(g)
Current most likely leaf: 0100
$\mathcal{L} = \{10\}$
(h)
Current most likely leaf: 0100
$\mathcal{L} = \varnothing$
(i)

Figure 4.1.: (a) Initial SC decoding outputs $v^N = 0111$ with the corresponding PM $M(v^N)$. (b) During the initial SC decoding, the PMs and scores are computed for branch nodes $\{1, 00, 010, 0110\}$. (c) The branch nodes with PMs larger than that of the current most likely leaf are pruned, e.g., we have $M(00), M(0110) > M(0111)$. Suppose now that $S(010) < S(1)$ and the members of $\mathcal{L}$ are ordered in the ascending order according to their scores. (d) The first candidate is popped from the list and the decoder returns to the deepest (or nearest) common node. (e) The decision is flipped and SC decoding continues. During the decoding, the list $\mathcal{L}$ and the current most likely leaf are updated. (f) The branch nodes with PMs larger than that of the current most likely leaf are pruned as in (c) (in this case, a leaf node is removed). (g) Repeat the procedure as in step (d), where we assume $M(11) > M(0100)$. (h) The list $\mathcal{L}$ is updated when the branch (11) was visited. (i) The decoder examines the last member of the list $\mathcal{L}$, pops it from $\mathcal{L}$. After reaching the N-th decoding phase, suppose that there is no branch node left, which has a smaller PM than that of the current most likely leaf, i.e., $\mathcal{L} = \varnothing$. The current most likely leaf is declared as the decision $\hat{u}^N$.

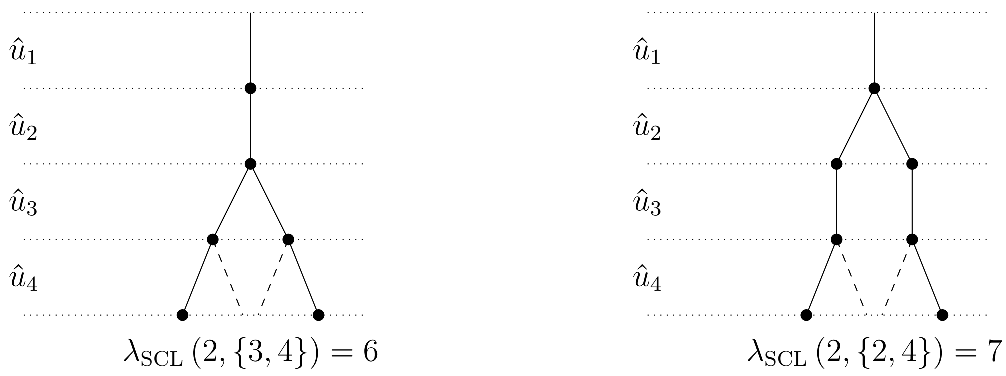


Figure 4.2.: SCL decoding trees for $(4,2)$ polar codes with $L = 2$. The black nodes are the visited nodes by decoding.

4.2. Complexity Analysis

The decoding complexity is measured by the *number of node-visits.* For instance, the number λ_{SC} of node-visits for SC decoding is simply the code length N independent of the channel output y^N. For a given code construction, the number of node-visits for SCL decoding with list size L, denoted as $\lambda_{\text{SCL}}(L, \mathcal{A})$, is also constant and upper bounded as $\lambda_{\text{SCL}}(L, \mathcal{A}) \leq LN$. An example is shown in Figure 4.2.

On the other hand, the complexity of SCOS decoding, denoted as $\lambda\left(y^N\right)$, depends on the channel output as for other sequential decoders [49]; hence, it is a RV defined as $\Lambda \triangleq \lambda\left(Y^N\right)$. In the following, we are interested in the average behaviour of Λ.

SCOS decoding may visit the same node more than once and these visits are included in the comparison. To understand the minimum required complexity for SCOS decoding, we consider the set of partial input sequences v^i with $i \in [N]$ with a smaller PM than the ML decision $\hat{u}^N_{\text{ML}}$.[3]

Definition 4.1. For a given channel output y^N and any binary sequence v^N, we define the string set

$$\mathcal{V}\left(v^N, y^N\right) \triangleq \bigcup_{i=1}^{N} \left\{u^i \in \{0,1\}^i : M\left(u^i\right) \leq M\left(v^N\right)\right\}. \tag{4.4}$$

[3]There are i node-visits for SC decoding for any decoding path v^i.

Lemma 4.1. For a particular realization y^N, we have

$$\lambda\left(y^N\right) \geq \left|\mathcal{V}\left(\hat{u}_{\mathrm{ML}}^N\left(y^N\right), y^N\right)\right| \tag{4.5}$$

and the expected complexity is lower bounded as

$$\frac{1}{\lambda_{\mathrm{SC}}}\mathbb{E}\left[\Lambda\right] \geq \frac{1}{N}\mathbb{E}\left[\left|\mathcal{V}\left(\hat{u}_{\mathrm{ML}}^N\left(Y^N\right), Y^N\right)\right|\right]. \tag{4.6}$$

Proof. The inequality (4.5) follows from the definition (4.4) and the description of SCOS decoding. Since (4.5) is valid for any y^N, the bound (4.6) follows by $\lambda_{\mathrm{SC}} = N$. ■

Remark. Recall that the PM (3.52) is calculated using the SC decoding schedule, i.e., it ignores the frozen bits coming after the current decoding phase i. This means the size of the set (4.4) tends to be smaller for codes more suited for SC decoding, e.g., polar codes, while it gets larger for others, e.g., RM codes.

The computational complexity of SCOS decoding can be limited by imposing a constraint on $\lambda\left(y^N\right)$, e.g., $\lambda\left(y^N\right) \leq \lambda_{\max}$, at the expense of suboptimality. To also limit the space complexity, one can impose a list size, e.g., we relate the space complexity to the number of node-visits heuristically as

$$|\mathcal{L}| \leq \eta \triangleq \log_2 N \times \frac{\lambda_{\max}}{N}. \tag{4.7}$$

4.3. Detailed Description

This section gives pseudo codes[4] and the details of the proposed SCOS decoder. The required data structures together with their size are listed in Table 4.1. Note that arrays `L` and `C` both store $(\log_2 N + 1) \times N$ elements in contrary to [103], where only $2N-1$ elements are stored, since we reuse some decoding paths to decrease the computational complexity (this is similar to the SC-Fano decoder). In the pseudo codes, the notation $\mathtt{v}[i]$ refers to the i-th entry of an array `v`. Similarly, the entry in position (i,j) of array `C` is denoted as $\mathtt{C}[i,j]$. The entries of bias vector `b` are computed offline via

$$\mathtt{b}[i] = \sum_{j=1}^{i} \log\left(1-p_j\right), \; i \in [N] \tag{4.8}$$

[4]In the pseudo codes, we use type-writer font for the data structures (with an exception for sets) and 1-based indexing arrays.

name	size	data type	description
$\mathtt{b}$	N	float	precomputed bias term
$\mathtt{L}$	$(\log_2 N + 1) \times N$	float	LLR
$\mathtt{C}$	$(\log_2 N + 1) \times N$	binary	hard decision
$\hat{\mathtt{u}}$, $\mathtt{v}$	N	binary	decoding path
$\mathtt{M}$, $\overline{\mathtt{M}}$, $\overline{\mathtt{S}}$	N	float	metric
$\mathtt{M}_{\text{cml}}$	1	float	PM of the current most likely leaf
$\mathcal{E}$, $\mathcal{E}_{\text{p}}$	1	set (of indices)	flipping set
$\mathtt{F}$	1	$\langle$set, float, float$\rangle$	structure of a flipping set
$\mathcal{L}$	$\leq \eta$	type of $\mathtt{F}$	list of flipping structures

Table 4.1.: Data structures for SCOS decoding.

Algorithm 4.1: InsertList ($\mathtt{F}$)

```
Input : structure F of a flipping set
1 $i = |\mathcal{L}| + 1$
2 while $i > 1$ and $\mathtt{F}.\overline{\mathtt{S}}_{\mathcal{E}} < \mathcal{L}[i-1].\overline{\mathtt{S}}_{\mathcal{E}}$ do
3     $i = i - 1$
4 insert F in list $\mathcal{L}$ at position $i$
5 if $|\mathcal{L}| > \eta$ then
6     pop the last element of $\mathcal{L}$
```

unless otherwise is stated explicitly. We implement the approximated f^- function (3.9) and PM update rule (3.57) to avoid non-linear operations.

A list $\mathcal{L}$ containing flipping structures $\mathtt{F} = \langle \mathcal{E}, \overline{\mathtt{M}}_{\mathcal{E}}, \overline{\mathtt{S}}_{\mathcal{E}} \rangle$ is constructed,[5] where $\mathcal{E}$ is the set of flipping indices. Then, $\overline{\mathtt{M}}_{\mathcal{E}}$ and $\overline{\mathtt{S}}_{\mathcal{E}}$ are the path metric and the score function associated to the flipping set $\mathcal{E}$, respectively, as defined in (4.1) and (4.2). The size of $\mathcal{L}$ is constrained by a parameter η, e.g., (4.7). Algorithm 4.1 is used to insert a new member into the list and the members are kept in ascending ordered by their score, i.e., for any pair of i and j such that $1 \leq i < j \in [\eta]$, it holds that $\mathcal{L}[i].\overline{\mathtt{S}}_{\mathcal{E}} \leq \mathcal{L}[j].\overline{\mathtt{S}}_{\mathcal{E}}$. Given two flipping sets, Algorithm 4.2 finds the decoding stage where the decoder should return, e.g., see Figure 4.1(d). The Algorithm 4.3 takes a real-valued PM, a binary decision and a real-valued LLR as inputs and updates the PM using (3.57).

[5]Observe that the list in Figure 4.1 is slightly different, which was needed for simplicity.

```
Algorithm 4.2: FindStartIndex $(\mathcal{E}, \mathcal{E}_\mathrm{p})$
  Input  : current flipping set $\mathcal{E}$ and previous flipping set $\mathcal{E}_\mathrm{p}$
  Output: first different index
1 for $i = 1, 2, \ldots, N$ do
2     if $(i \in \mathcal{E}) \oplus (i \in \mathcal{E}_\mathrm{p})$ then
3         return $i$
```

```
Algorithm 4.3: CalcPM $(m, v, \ell)$
  Input  : input PM $m$, hard decision $v$, LLR $\ell$
  Output: updated PM
1 if $v = \mathrm{HardDec}(\ell)$ then
2     return $m$
3 else
4     return $m + |\ell|$
```

Algorithm 4.4 is a modified SC decoder. Compared to the regular SC decoding (Algorithm A.2), it is modified as follows (highlighted as blue in the pseudo code).

- ▷ One can start at any decoding phase i_start. The starting index i_start is found by Algorithm 4.2, which returns the first index that differs between the current and previous flipping sets.
- ▷ It keeps updating the PM $\mathtt{M}$ (`line 13`).
- ▷ The decisions are flipped at the decoding phases in $\mathcal{E}$ (`line 6-8`).
- ▷ The values $\overline{\mathtt{M}}[i]$ and $\overline{\mathtt{S}}[i]$ for $i \in \mathcal{A}, i > \mathrm{maximum}(\mathcal{E})$ (`line 10-12`).
- ▷ If a more likely leaf is found, update $\hat{\mathbf{u}}$ and $\mathtt{M}_\mathrm{cml}$ (`line 19-23`).
- ▷ If the PM $\mathtt{M}[i]$ is larger than $\mathtt{M}_\mathrm{cml}$, stop SCDec function and return the current phase i as i_end (`line 14-15`).

Algorithm 4.5 is the main loop of the proposed SCOS decoder. The metric $\mathtt{M}_\mathrm{cml}$ is initialized to $+\infty$. After the initial SC decoding (`line 6`), $\mathtt{M}_\mathrm{cml}$ is updated to the PM of the SC estimate. Then, a tree search is performed, where the candidates are ordered by their score functions $\overline{\mathtt{S}}$. Many subtrees are pruned because of the threshold $\mathtt{M}_\mathrm{cml}$, i.e., PM of the current most likely leaf. The stopping condition of the "while loop" (`line 10`), i.e., $\mathcal{L} = \varnothing$, implies that the most likely codeword is found, i.e., there cannot be any other codeword with a smaller PM. The estimate corresponding to $\mathtt{M}_\mathrm{cml}$ is output as the decision.

Algorithm 4.4: SCDec $(i_{\text{start}}, \mathcal{E})$

```
Input  : start index $i_{\text{start}}$, flipping set $\mathcal{E}$
Output: end index $i_{\text{end}}$
for $i = i_{\text{start}}, \dots, N$ do
    recursivelyCalcL $(\log_2 N + 1, i - 1)$
    if $i \notin \mathcal{A}$ then
        $\mathtt{v}[i] = 0$ // compute $\mathtt{v}[i]$ if dynamically frozen
    else
        if $i \in \mathcal{E}$ then
            $\mathtt{v}[i] = \text{HardDec}(\mathtt{L}[\log_2 N + 1, i]) \oplus 1$
        else
            $\mathtt{v}[i] = \text{HardDec}(\mathtt{L}[\log_2 N + 1, i])$
        if $i > \text{maximum}(\mathcal{E})$ then
            $\bar{\mathtt{M}}[i] = \text{CalcPM}(\mathtt{M}[i-1], \mathtt{v}[i] \oplus 1, \mathtt{L}[\log_2 N + 1, i])$
            $\bar{\mathtt{S}}[i] = \bar{\mathtt{M}}[i] + \mathtt{b}[i]$
    $\mathtt{M}[i] = \text{CalcPM}(\mathtt{M}[i-1], \mathtt{v}[i], \mathtt{L}[\log_2 N + 1, i])$
    if $\mathtt{M}[i] \geq \mathtt{M}_{\text{cml}}$ then
        return $i$
    $\mathtt{C}[\log_2 N + 1, i] = \mathtt{v}[i]$
    if $i \mod 2 = 0$ then
        recursivelyCalcC $(\log_2 N + 1, i - 1)$
if $\mathtt{M}[N] < \mathtt{M}_{\text{cml}}$ then
    $\mathtt{M}_{\text{cml}} = \mathtt{M}[N]$
    for $i = 1, 2, \dots, N$ do
        $\hat{\mathtt{u}}[i] = \mathtt{v}[i]$
    return $N$
```

Remark. SCOS decoding stores only the flipping set and the corresponding metrics in $\mathcal{L}$. Alternatively, one may store the memory (see Section 3.4 and Figure 3.9) of all decoding paths in $\mathcal{L}$ as for SCS decoding [71, 110] to prevent node-revisits, which trades memory requirement for computational complexity. The size of the arrays L and C is then increased to $\eta \times (2N - 1)$.

Algorithm 4.5: SCOS $\left(\ell^N\right)$

```
Input  : LLRs $\ell^N$
Output: estimates $\hat{\mathtt{u}}$
1  $\mathcal{L} = \varnothing, \mathcal{E}_\mathrm{p} = \varnothing, \mathtt{M}_\mathrm{cml} = +\infty$
2  if non bit-reversed polar code then
3      $\ell^N = \mathrm{BitReverse}\left(\ell^N\right)$
4  for $i = 1, 2, \ldots, N$ do
5      $\mathtt{L}[1, i] = \ell_i$
6  SCDec$(1, \varnothing)$
7  for $i = 1, 2, \ldots, N$ do
8      if $i \in \mathcal{A}$ and $\overline{\mathtt{M}}[i] < \mathtt{M}_\mathrm{cml}$ then
9          InsertList$(\langle \{i\}, \overline{\mathtt{M}}[i], \overline{\mathtt{S}}[i] \rangle)$
10 while $\mathcal{L} \neq \varnothing$ do
11     $\langle \mathcal{E}, \overline{\mathtt{M}}_\mathcal{E}, \overline{\mathtt{S}}_\mathcal{E} \rangle = \mathrm{popfirst}(\mathcal{L})$ // pop the first element of $\mathcal{L}$
12     if $\overline{\mathtt{M}}_\mathcal{E} < \mathtt{M}_\mathrm{cml}$ then
13         $i_\mathrm{start} = \mathrm{FindStartIndex}(\mathcal{E}, \mathcal{E}_\mathrm{p})$
14         $i_\mathrm{end} = \mathrm{SCDec}(i_\mathrm{start}, \mathcal{E})$
15         for $i = \mathrm{maximum}(\mathcal{E}) + 1, \ldots, i_\mathrm{end}$ do
16             if $i \in \mathcal{A}$ and $\overline{\mathtt{M}}[i] < \mathtt{M}_\mathrm{cml}$ then
17                 InsertList$(\langle \mathcal{E} \cup \{i\}, \overline{\mathtt{M}}[i], \overline{\mathtt{S}}[i] \rangle)$
18         $\mathcal{E}_\mathrm{p} = \mathcal{E}$
19 return $\hat{\mathtt{u}}$
```

4.4. Numerical Results

4.4.1. Comparison with Existing Decoders

This section provides simulation results for biAWGN channels. The complexity is normalized by the complexity of SC decoding. The bias term $\mathtt{b}$ for SCOS decoding is precomputed by DE for every single SNR point. The random coding union bound (RCUB) and meta-converse bound [83] are plotted as benchmarks. The empirical *ML lower bounds* of [103] are also plotted: for SCOS decoding with the largest maximum complexity constraint, each time a decoding failure occurred the decision $\hat{u}^N$ was checked. If

$$M(\hat{u}^N) \leq M(u^N) \tag{4.9}$$

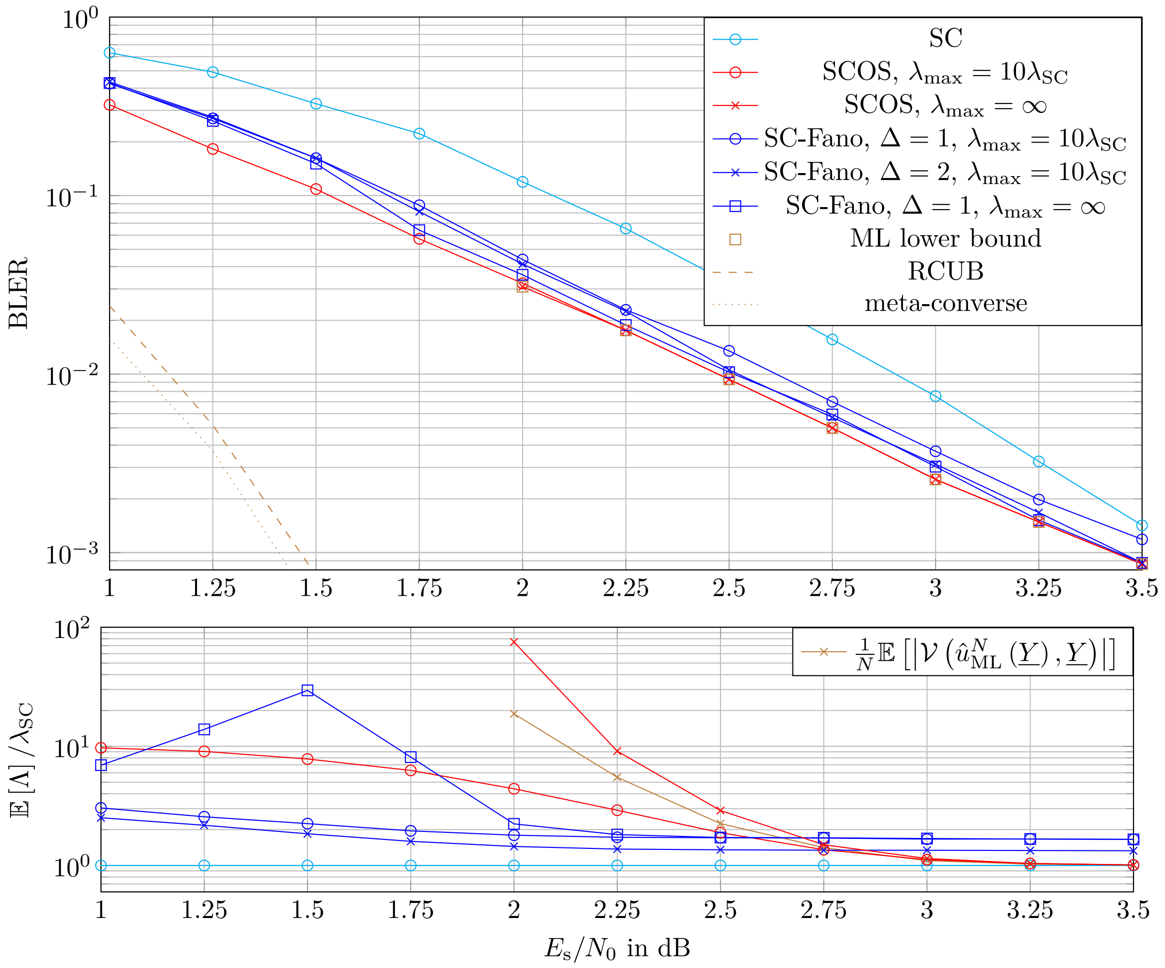


Figure 4.3.: SCOS decoding for a $(512, 256)$ polar code with PW construction over biAWGN channel.

then even an ML decoder would make an error.

The decoding performance of a $(512, 256)$ polar code under SCOS decoding is shown in Figure 4.3. The polar code is designed by PW (Definition 3.2). The decoding performance of a $(128, 64)$ RM under SCOS decoding is shown in Figure 4.4. Figure 4.5 shows the performance of a $(128, 64)$ PAC code [6] under SCOS decoding. The frozen set $\mathcal{F}$ is the same as for an RM code. The polynomial of the convolutional code is given by $\boldsymbol{g} = (0, 1, 1, 0, 1, 1)$, i.e., we have dynamic frozen bits

$$u_i = u_{i-2} \oplus u_{i-3} \oplus u_{i-5} \oplus u_{i-6}, \; i \in \mathcal{F} \text{ and } i > 6. \tag{4.10}$$

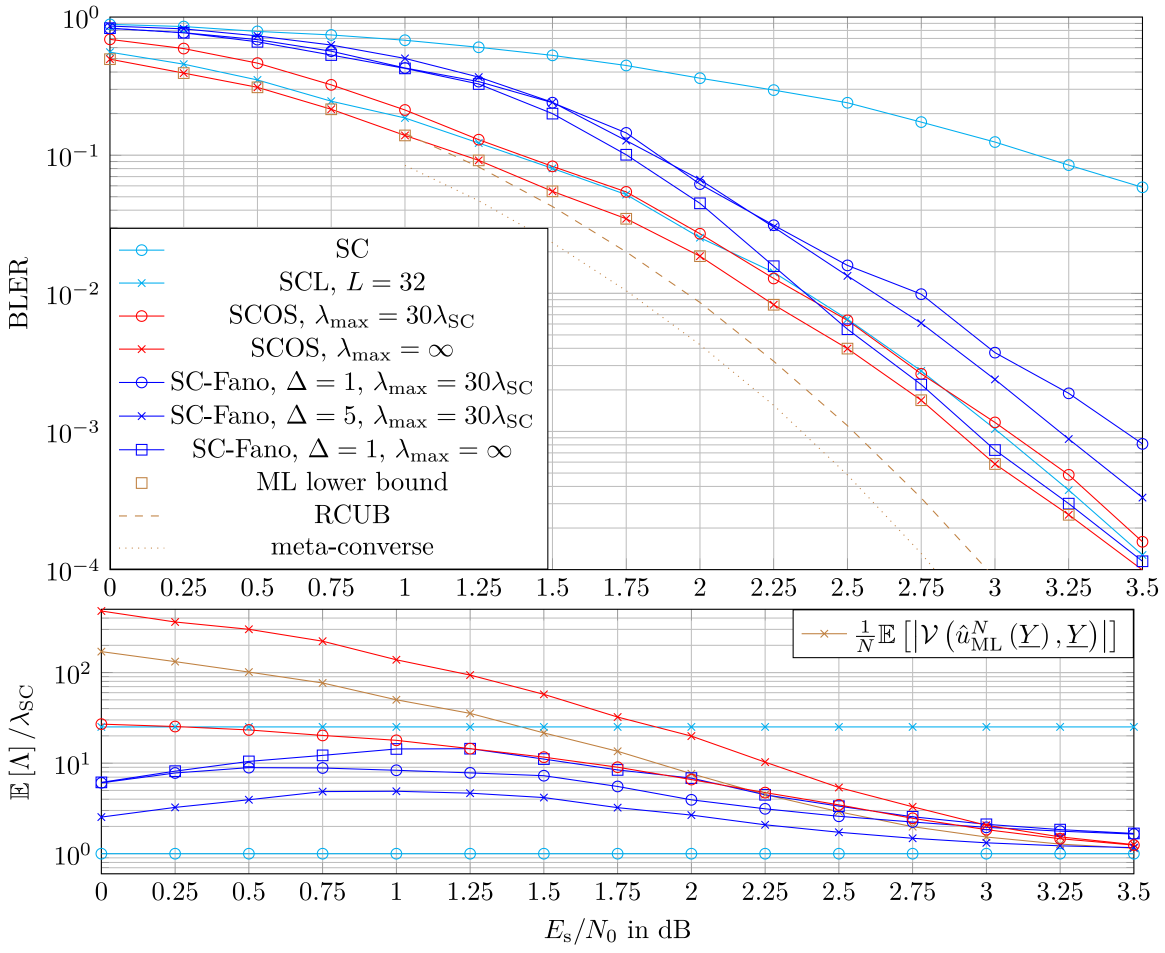


Figure 4.4.: SCOS decoding for a $(128, 64)$ RM code over biAWGN channel.

We compare SCOS decoding with SCL [7, 103] and SC-Fano [50] decoding. We observe the following behaviors,

- ▷ SCOS decoding with unbounded complexity matches the ML lower bound since it implements an ML decoder.
- ▷ The average complexity $\mathbb{E}\left[\Lambda\right]$ of an SCOS decoder approaches the complexity of an SC decoder at high SNR. Indeed, it reaches the ultimate limit given by Lemma 4.1, which is not the case for SC-Fano decoding. The difference to the RCUB bound [83] is at most 0.2 dB for the entire SNR regime for the PAC code in Figure 4.5.
- ▷ The lower bound on the average given by (4.6) is validated and is tight for high SNR. However, the bound appears to be loose at low SNR mainly for two reasons:
  (i) Usually the initial SC decoding estimate v^N is not the ML decision and some

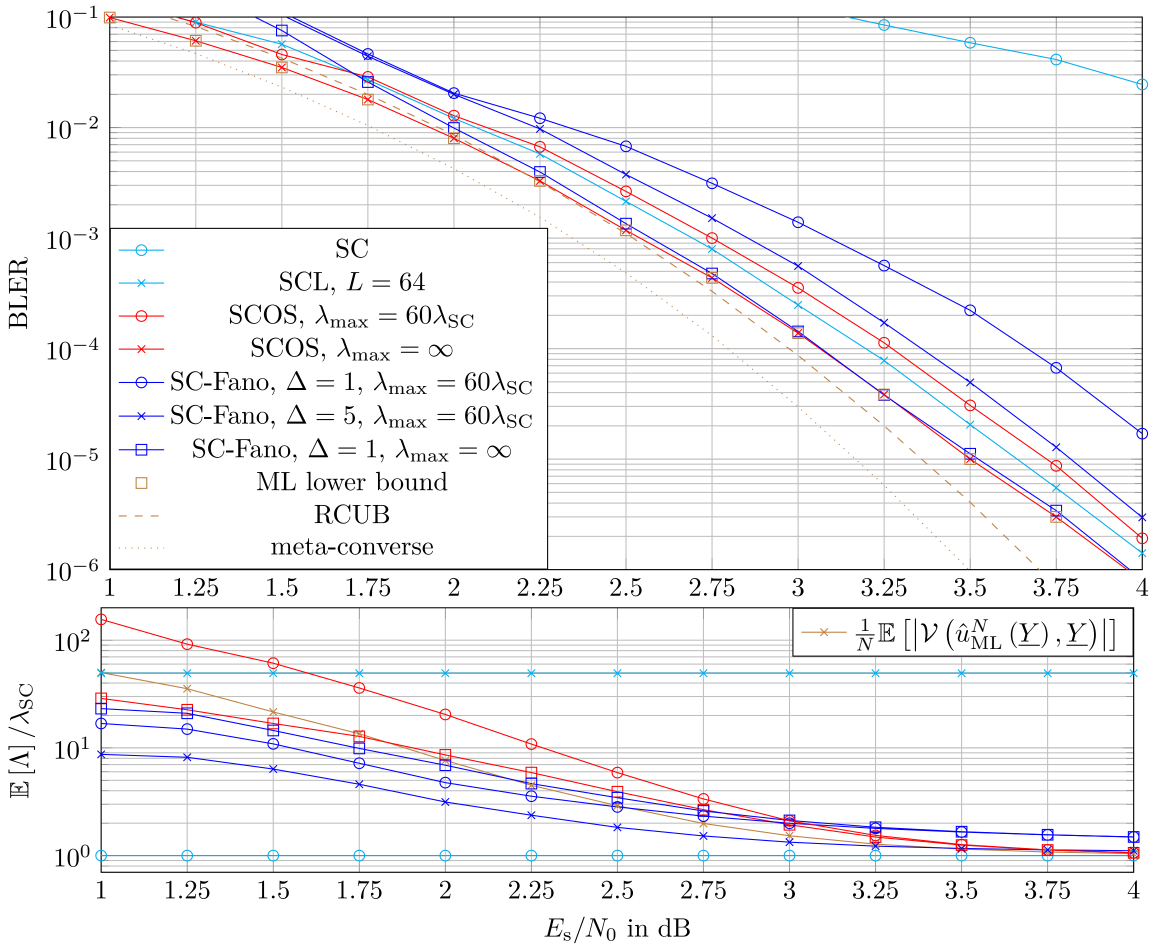


Figure 4.5.: SCOS decoding for a $(128, 64)$ PAC code with information set of an RM code and polynomial $\boldsymbol{g} = (0, 1, 1, 0, 1, 1)$.

extra nodes in the difference set $\mathcal{V}\left(v^N, y^N\right) \setminus \mathcal{V}\left(\hat{u}^N_{\mathrm{ML}}\left(y^N\right), y^N\right)$ are visited.

(ii) SCOS decoding visits the same node multiple times and this cannot be tracked by a set definition.

Considering (ii), it may be possible to reduce the number of revisits by improving the search schedule.

▷ A parameter Δ must be optimized carefully for SC-Fano decoding to achieve ML performance and this usually requires extensive simulations. Setting it small enough without any bound on the complexity would also practically achieve ML performance; however, the complexity then explodes for longer codes. As seen from Figure 4.5, $\Delta = 1$ matches the ML performance, but the average complexity is almost double that of SC decoding near BLERs of 10^{-5} or below. Moreover, under a maximum complexity constraint, the average complexity of SC-Fano decoding is not closer to that of SC decoding than SCOS decoding for similar performance.

▷ The parameter Δ must be optimized again for a good performance once a maximum complexity constraint is imposed. Otherwise, the performance degrades significantly. Even so, SCOS decoding outperforms SC-Fano decoding for the same maximum complexity constraint. However, SC-Fano decoding has a lower average complexity for high BLERs (if Δ is optimized) with a degradation in the performance. In contrast, SCOS decoding does not require such an optimization.

By applying an outer CRC code, we compare the proposed method with DSCF [16] and adaptive successive cancellation list (ASCL) [56] decoding in Figure 4.6. The $(256, 139)$ polar code is designed by PW and the CRC polynomial is $g(x) = x^{11} + x^{10} + x^9 + x^5 + 1$. The ASCL decoder starts by performing SCL decoding with list size $L = 1$. In case none of the candidates in the output list pass the CRC, the ASCL decoder restarts an SCL decoder with doubled list size. This decoding process continues until the CRC passes or the maximum list size $L_{\max}$ is reached. The worse-case complexity of an ASCL decoder with maximum list size $L_{\max}$ is close to an SCL decoder with list size $L = 2 \times L_{\max}$. The SCOS decoder treats the CRC-aided polar codes as polar codes with dynamic frozen bits, i.e., the last 11 bits in u^N are dynamically frozen with the CRC constraints.

In Table 4.2, we compare the following details of the decoders:

▷ *Worse-case complexity*: The worse-case complexity is given by the maximum number of node-visits in the decoding tree.

▷ *Space complexity*: We consider only the most critical matrix containing the soft information (2-D array `L`). By using the structure proposed in [103], the memory requirement of an SC decoder is reduced from $N \cdot (\log_2 N + 1)$ to $2N - 1$. An SCL decoder needs memory $L \cdot (2N - 1)$. Because the SCOS and SC-Fano decoders allow moving backward, the structure in [103] does not work. Thus, the memory requirement of an SCOS or SC-Fano decoder is approximately $\frac{1}{2} \log_2 N$ times higher than for an SC decoder.

▷ *Decoding latency*: The decoding latency describes the required number of time steps assuming all parallelizable operations are done at the same time step. The SC and SCL decoders have stable decoding latencies. The SCOS, DSCF, ASCL and SC-Fano decoders have variable latency since they restart the decoder multiple times or allow to move back to an earlier decoding phase.

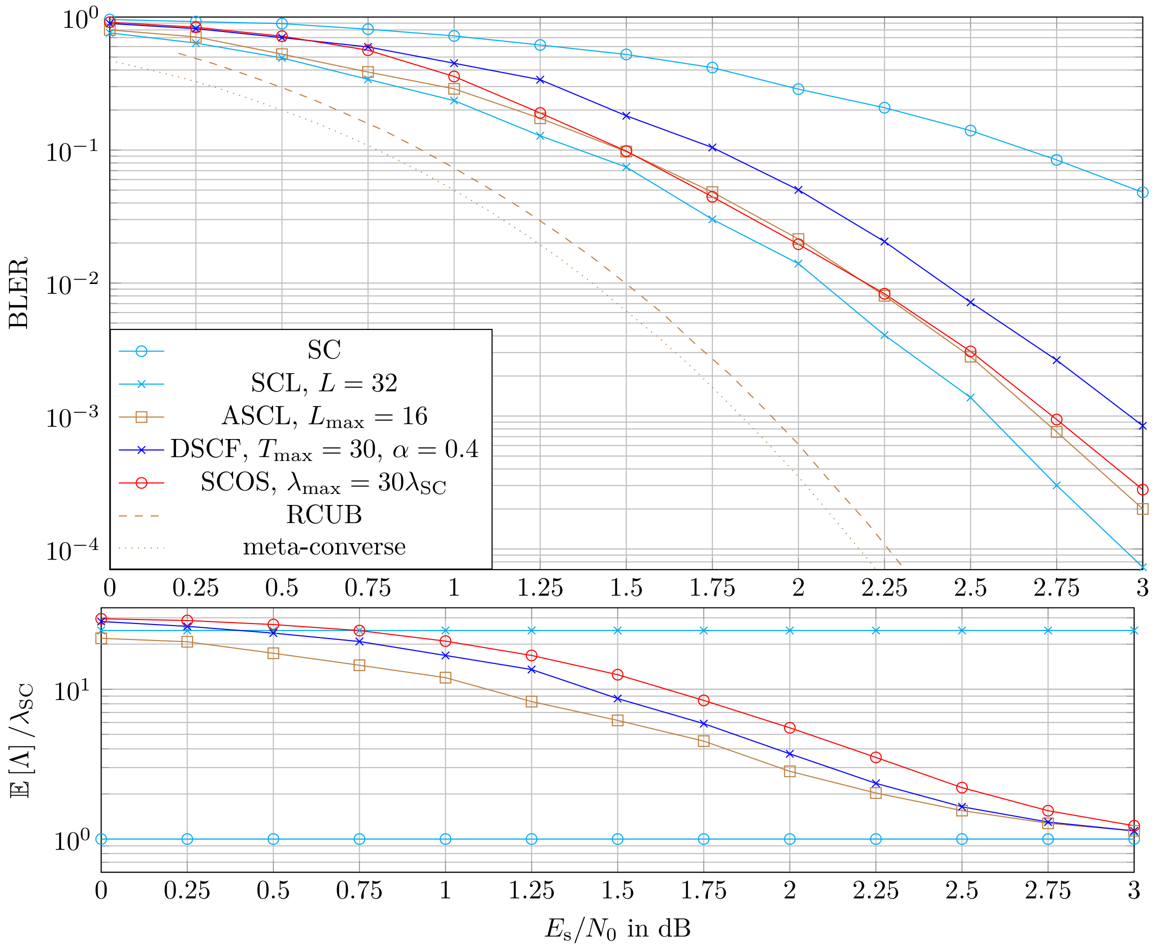


Figure 4.6.: SCOS decoding for a $(256, 128 + 11)$ polar code with 11 bits CRC (generator polynomial $g(x) = x^{11} + x^{10} + x^9 + x^5 + 1$) designed by PW over biAWGN channel.

4.4.2. Bias Term Robustness

Consider the bias terms $\mathtt{b}$ given in (4.8), which impacts the search priority but not the performance. This means that a *suboptimal* bias term does not change the performance of SCOS decoding with unbounded complexity (which is still ML decoding), but increases its complexity. Figure 4.7 illustrates the effect of various bias terms, outlined below, on the performance of SCOS decoding.

- ▷ The bias terms are computed via (4.8) using density evolution for each SNR point.
- ▷ The bias terms are set to zero, i.e., $\mathtt{b}[i] = 0$, $i \in [N]$.

As mentioned earlier, the changes in the bias terms do not affect the performance if there is no complexity constraint. The reduction in the complexity is also limited for the considered

	Worst-case complexity	Space complexity	Decoding latency
SC	N	$2N-1$	fixed
SCL (L)	$\lambda_{\text{SCL}}(L, \mathcal{A})$	$L \cdot (2N-1)$	fixed
DSCF (α, $T_{\max}$)	$(T_{\max}+1) \cdot N$	$2N-1$	varying
ASCL ($L_{\max}$)	$\sum_{i=0}^{\log_2 L_{\max}} \lambda_{\text{SCL}}(2^i, \mathcal{A})$	$L_{\max} \cdot (2N-1)$	varying
SC-Fano (Δ, $\lambda_{\max}$)	$\lambda_{\max}$	$N \cdot (\log_2 N + 1)$	varying
SCOS ($\lambda_{\max}$)	$\lambda_{\max}$	$N \cdot (\log_2 N + 1)$	varying

Table 4.2.: Comparison among polar decoders.

case when (4.8) is used instead of setting the bias terms to zero. Nevertheless, setting them to zero causes a small degradation in the performance (approximately 0.12 dB) when the maximum complexity is constraint to five times that of SC decoding with almost no savings in the average complexity. Hence, we conclude that SCOS decoding is not very sensitive to the choice of the bias terms.

4.5. Further Improvements

Via Monte Carlo simulation, one can approximate the PDF of the PM for the transmitted message at a given SNR by using genie-aided SC decoding [5].[6] Figure 4.8 provides the PDF for the case of the $(128, 64)$ PAC code at SNR $= 3.5$ dB. We have

$$\Pr\left(M\left(u^N\right) > \zeta\right) \approx \begin{cases} 0.0217, \text{ if } \zeta = 25 \\ 0.0003, \text{ if } \zeta = 35 \\ 0, \text{ if } \zeta = 50. \end{cases} \tag{4.11}$$

Observe, for instance, that we have $\Pr\left(M\left(u^N\right) > 50\right) \approx 0$. This means that, if the decoder discards the paths having PMs larger than 50, the performance degradation is negligible while providing savings in the computational complexity. Such a modification is relevant, in particular, for the case where a maximum complexity constraint is imposed to SCOS decoding. In this case, unnecessary node-visits drains the budget and leads to a suboptimal decision more often. In addition, such a threshold test enables the decoder to

[6]Note that $M\left(u^N\right)$ is a RV where the source of randomness is the channel output. Since we consider symmetric B-DMCs and a linear code with uniform distribution, the PDF of $M\left(u^N\right)$ could be approximated with an all zero codeword assumption.

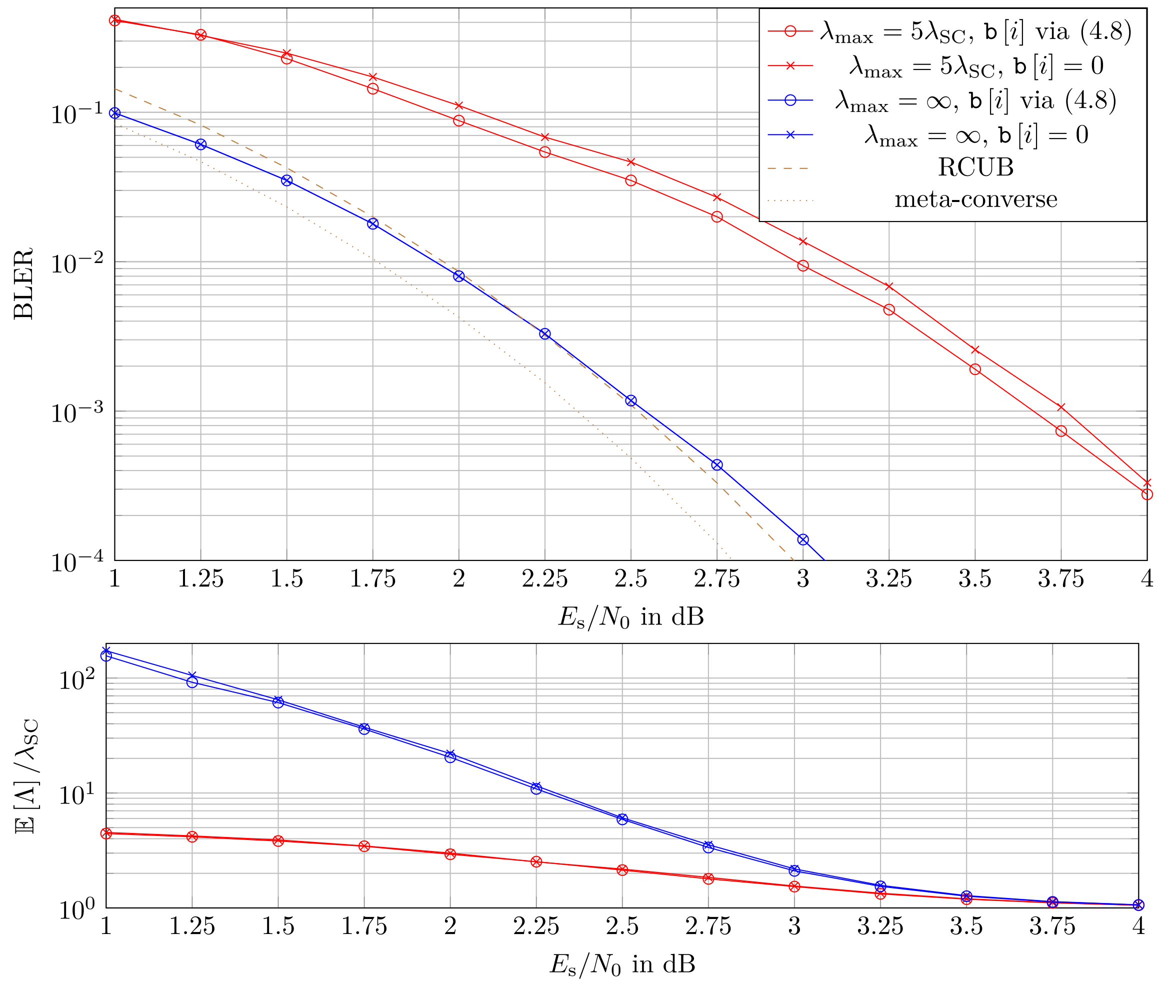


Figure 4.7.: SCOS decoding with various bias terms and maximum complexity constraints for a (128, 64) PAC code with RM construction and polynomial $\boldsymbol{g} = (0, 1, 1, 0, 1, 1)$ over biAWGN channel.

reject a decision when it is not reliable enough, which reduces the number of undetected errors if the threshold is carefully optimized (see, e.g., [34]).

In the following, we modify SCOS decoding by setting a maximum PM $M_{\max}$, which forces the decoder to consider paths with PMs lower than $M_{\max}$. This modification is provided as Algorithm 4.6. We choose $M_{\max}$ heuristically so that

$$\Pr\left(M\left(u^N\right) > M_{\max}\right) \approx \text{target BLER}. \tag{4.12}$$

Figure 4.9 compares the performance and the complexity of the modified algorithm to those of the conventional one, where the former outperforms the latter by approximately 0.25 dB with the same maximum complexity constraint $\lambda_{\max} = 5N$ if $M_{\max} = 35$ via (4.12).

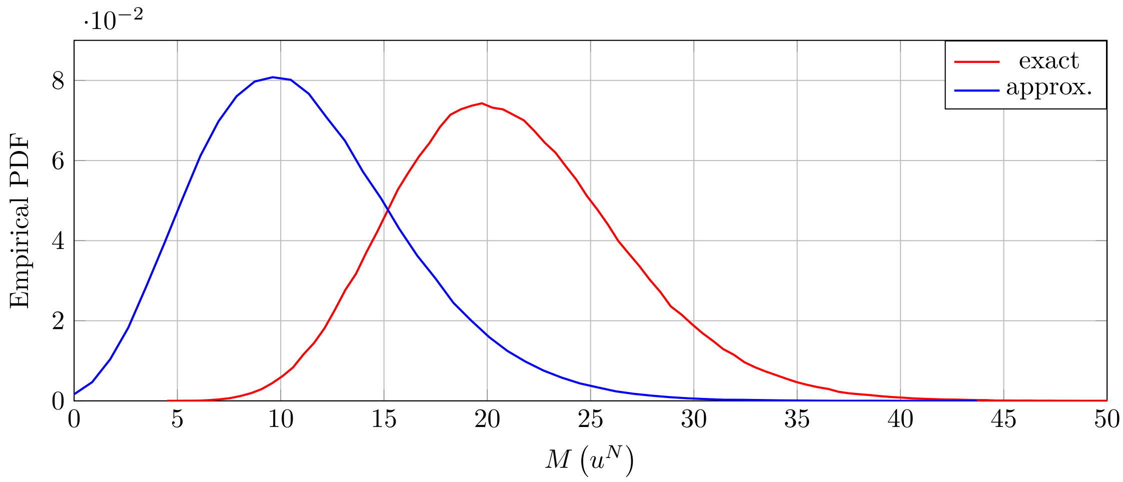


Figure 4.8.: Empirical PDF (from 10^7 samples) of the $M\left(u^N\right)$ for a $(128, 64)$ PAC code with RM construction and polynomial $\boldsymbol{g} = (0, 1, 1, 0, 1, 1)$ over a biAWGN channel at SNR $= 3.5$ dB. The blue line is with a genie-aided SC decoder [5] with the exact f^- operation (3.5) and PM update rule (3.56). The red line is with a genie-aided SC decoder with approximated f^- operation (3.9) and PM update rule (3.57).

Note also that their average complexity, which is provided here in the linear scale, is very similar.

For a given threshold $M_{\max}$, we define a binary RV Ω as

$$\Omega = \mathbb{1}\left\{M(\hat{u}^N) \leq M_{\max}\right\} \tag{4.13}$$

where the indicator function $\mathbb{1}\{\cdot\}$ equals 1 if the proposition is true and 0 otherwise. The proposition of the indicator function (4.13) reads as "the modified SCOS decoding finds an estimate $\hat{u}^N$ with a PM smaller than $M_{\max}$". Then, the undetected error probability of the algorithm is given as

$$\Pr\left(\hat{U}^N \neq U^N, \Omega = 1\right). \tag{4.14}$$

Observe that the overall error probability is equal to the summation of detected and undetected error probabilities, i.e., we have

$$\Pr\left(\hat{U}^N \neq U^N\right) = \sum_{\omega \in \{0,1\}} \Pr\left(\hat{U}^N \neq U^N, \Omega = \omega\right) \tag{4.15}$$

which simply follows from the law of total probability. The parameter $M_{\max}$ controls the BLER and undetected block error rate (uBLER) tradeoff [34, 44]. In particular, (4.14)

Algorithm 4.6: SCOS with maximum PM $\left(\ell^N, M_{\max}\right)$

```
Input  : input LLRs $\ell^N$, $M_{\max}$
Output: output vector $\hat{\mathtt{u}}$, decoding state $\omega$
1  $\mathcal{L} = \varnothing, \mathcal{E}_{\mathrm{p}} = \varnothing, \mathtt{M}_{\mathrm{cml}} = M_{\max}$, $\omega = 0$
2  if non bit-reversed polar code then
3      $\ell^N = \mathrm{BitReverse}\left(\ell^N\right)$
4  for $i = 1, 2, \ldots, N$ do
5      $L[1, i] = \ell_i$
6  $i_{\mathrm{end}} = \mathrm{SCDec}(1, \varnothing)$
7  if $i_{\mathrm{end}} = N$ then $\omega = 1$
8  for $i = 1, 2, \ldots, N$ do
9      if $i \in \mathcal{A}$ and $\overline{\mathtt{M}}[i] < \mathtt{M}_{\mathrm{cml}}$ then
10         $\mathrm{InsertList}\left(\langle\{i\}, \overline{\mathtt{M}}[i], \overline{\mathtt{S}}[i]\rangle\right)$
11 while $\mathcal{L} \neq \varnothing$ do
12     $\langle\mathcal{E}, \overline{\mathtt{M}}_{\mathcal{E}}, \overline{\mathtt{S}}_{\mathcal{E}}\rangle = \mathrm{popfirst}(\mathcal{L})$
13     if $\overline{\mathtt{M}}_{\mathcal{E}} < \mathtt{M}_{\mathrm{cml}}$ then
14         $i_{\mathrm{start}} = \mathrm{FindStartIndex}(\mathcal{E}, \mathcal{E}_{\mathrm{p}})$
15         $i_{\mathrm{end}} = \mathrm{SCDec}(i_{\mathrm{start}}, \mathcal{E})$
16         if $i_{\mathrm{end}} = N$ then $\omega = 1$
17         for $i = \mathrm{maximum}(\mathcal{E}) + 1, \ldots, i_{\mathrm{end}}$ do
18             if $i \in \mathcal{A}$ and $\overline{\mathtt{M}}[i] < \mathtt{M}_{\mathrm{cml}}$ then
19                 $\mathrm{InsertList}\left(\langle\mathcal{E} \cup \{i\}, \overline{\mathtt{M}}[i], \overline{\mathtt{S}}[i]\rangle\right)$
20         $\mathcal{E}_{\mathrm{p}} = \mathcal{E}$
21 return $\hat{\mathtt{u}}$, $\omega$
```

becomes equal to the left-hand side of (4.15) if $M_{\max} = \infty$. Figure 4.10 illustrates that the $(128, 64)$ PAC code under modified SCOS decoding provides simultaneous gains in overall BLER as well as in uBLER compared to a $(128, 71)$ polar code concatenated with a CRC-7, resulting in a $(128, 64)$ overall code, under SCL decoding with $L = 16$ at high SNR regime. Furthermore, it outperforms the DSCF decoding with the maximum $T_{\max} = 70$ bit flips although with a small increase in the average complexity.

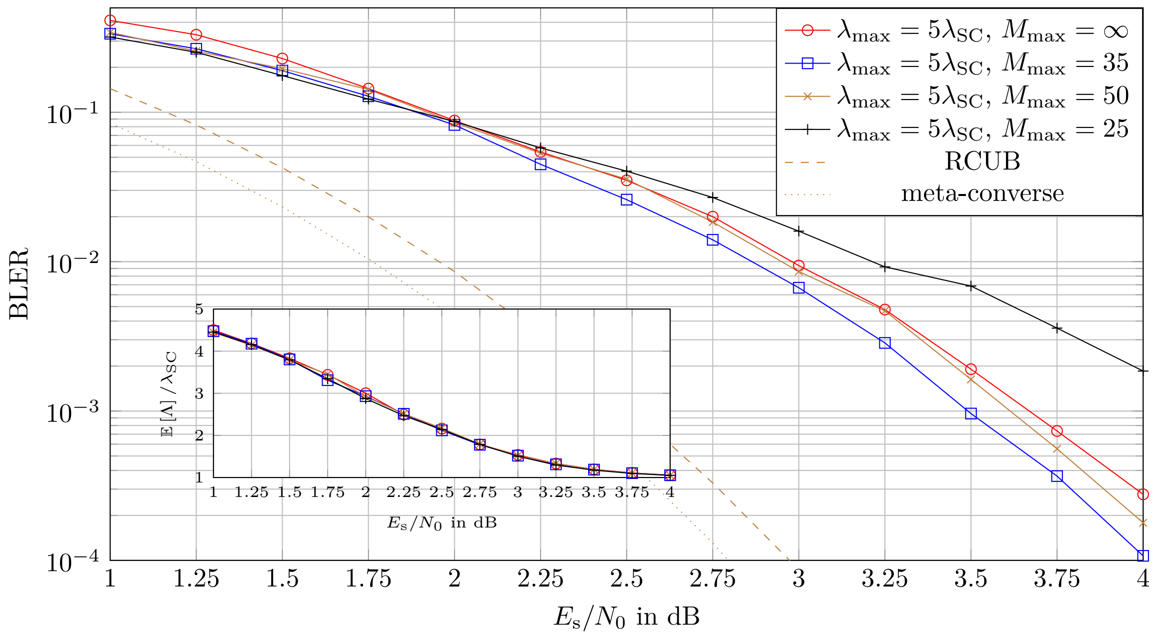


Figure 4.9.: Modified SCOS decoding with various maximum PMs and a fixed maximum complexity constraint for a $(128, 64)$ PAC code with RM construction and polynomial $\boldsymbol{g} = (0, 1, 1, 0, 1, 1)$ over biAWGN channel.

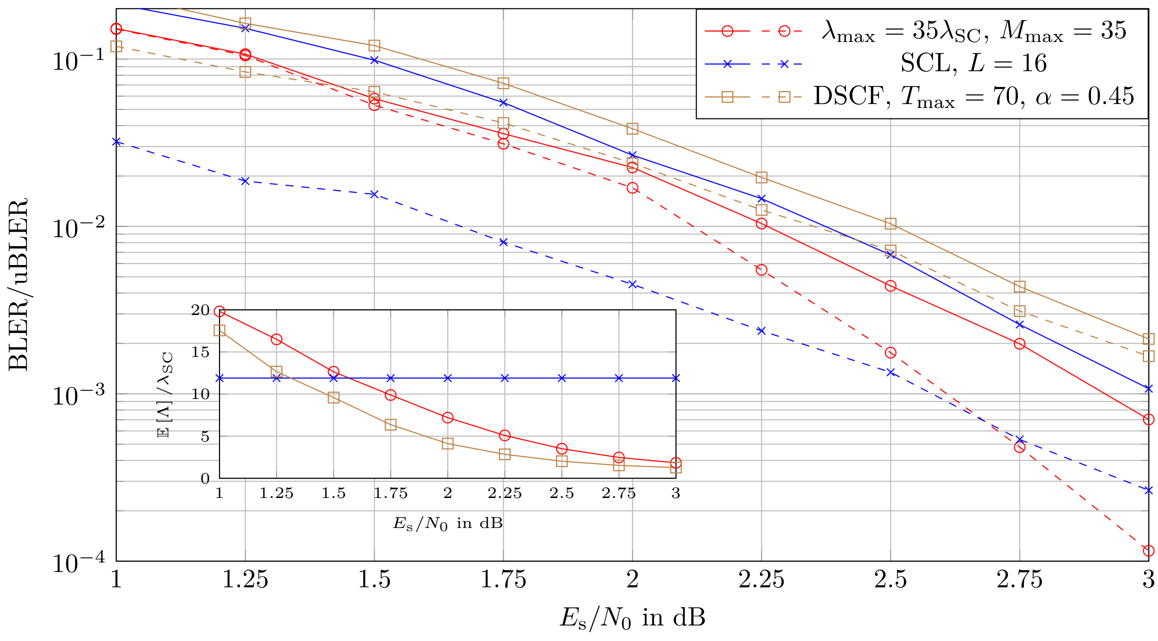


Figure 4.10.: BLER and uBLER of modified SCOS decoding for a $(128, 64)$ PAC code with RM construction and polynomial $\boldsymbol{g} = (0, 1, 1, 0, 1, 1)$ compared to SCL and DSCF decoding for a $(128, 64 + 7)$ polar code designed by PW with 7 bits CRC (generator polynomial $g(x) = x^7 + x^6 + x^5 + x^2 + 1$).

5 Polar-Coded IR-HARQ

5.1. Problem Statement

Many communication channels are time-varying and unknown to the transmitter. A HARQ method combines error correction and automatic repeat request (ARQ) error-control. HARQ is usually classified as Chase combining (CC) or incremental redundancy (IR), depending on whether the retransmitted bits are the same as the first transmission or not.

▷ CC-HARQ has each retransmission send the same coded bits as the first transmission. Received packets are combined before they are fed to the channel decoder. CC thus increases the received SNR but does not provide coding gain.

▷ IR-HARQ is shown in Figure 5.1 and transmits new redundancy (usually in the form of new redundant constraints) until the information bits can be reconstructed. The receiver combines its received symbols with previous transmissions.

IR generally achieves higher throughput than CC. For example, let R_{IR} and R_{CC} be the IR-HARQ and CC-HARQ rates, respectively, and let SNR_i be the SNR of the i-th transmission. For AWGN channels we have

$$\begin{aligned} R_{\mathrm{IR}} &= \sum_i \log_2\left(1+\mathrm{SNR}_i\right) = \log_2\left(\prod_i \left[1+\mathrm{SNR}_i\right]\right) \\ &\geq \log_2\left(1+\sum_i \mathrm{SNR}_i\right) = R_{\mathrm{CC}}. \end{aligned} \tag{5.1}$$

In a IR-HARQ system, the decoder receives n_t bits from the t-th transmission and then

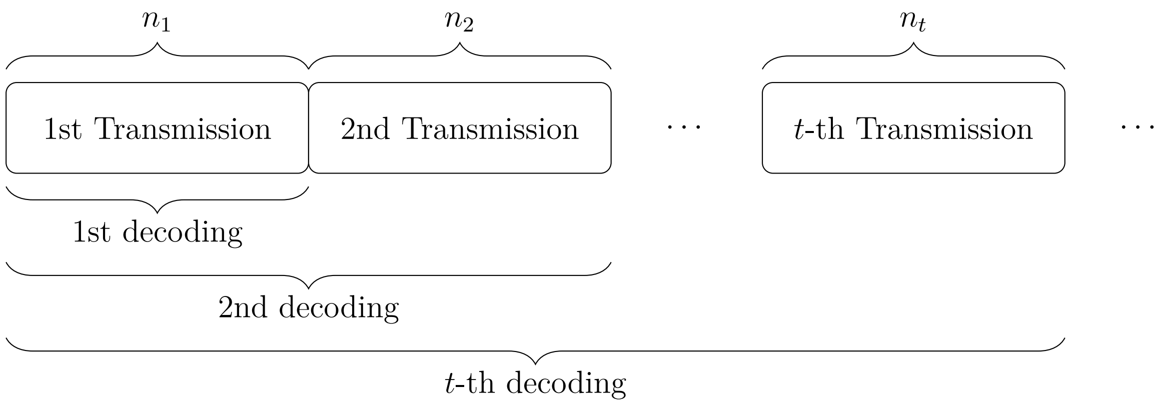


Figure 5.1.: IR-HARQ: Each transmission contains new redundancy.

decodes a $\left(\sum_{q=1}^{t} n_q, k\right)$ code. Using an error detection outer code (e.g., CRC code), the receiver may detect a decoding failure, in which case it requests the $(t+1)$-st transmission from the sender. IR-HARQ solutions are mostly based on turbo or LDPC codes. Turbo codes have a low rate mother code and are thus suitable for IR-HARQ when combined with puncturing. The coding scheme for eMBB in 5G uses protograph-based, Raptor-like LDPC codes [17] that allow flexible adaptation of the block length and code rate. The standard defines two base matrices for different operating regimes.

5.2. Existing Schemes of Polar-Coded IR-HARQ

5.2.1. Incremental Freezing

Polar codes with incremental freezing (IF) are proposed in [46,58]. The main idea is as follows. Transmit a (n_1, k) polar codeword $c[1]^{n_1}$. When retransmission occurs, transmit the k' *least reliable* message bits with an (n_2, k') polar codeword $c[2]^{n_2}$ in the next transmission. After the second transmission one decodes the (n_2, k') code and the first code successively. Note that the first code becomes a $(n_1, k-k')$ code with the estimate of the least reliable message bits. For the third transmission (if needed), transmit the k'' least reliable message bits of both the $(n_1, k-k')$ and the (n_2, k') codewords. The retransmissions are continued in this manner until decoding is successful. An example is shown in Figure 5.2.

The nesting property implies that for infinite code lengths we have

$$\mathcal{A}_t \subseteq \mathcal{A}_{t-1} \tag{5.2}$$

where $\mathcal{A}_t$ denotes the information set of the code $c[t]^{n_t}$. Thus, this scheme achieves capac-

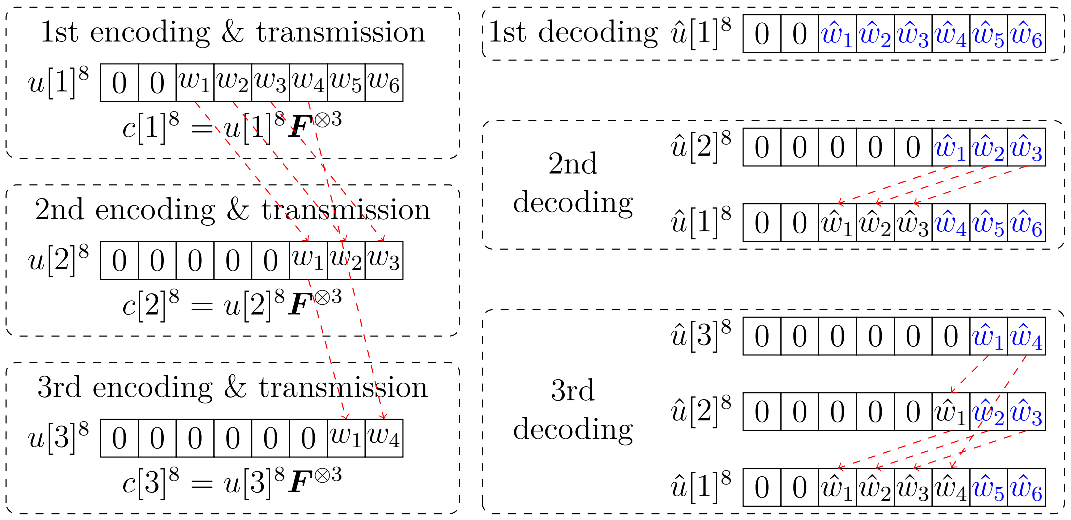

1st encoding & transmission
$u[1]^8$ 0 0 w_1 w_2 w_3 w_4 w_5 w_6
$c[1]^8 = u[1]^8 \boldsymbol{F}^{\otimes 3}$
2nd encoding & transmission
$u[2]^8$ 0 0 0 0 0 w_1 w_2 w_3
$c[2]^8 = u[2]^8 \boldsymbol{F}^{\otimes 3}$
3rd encoding & transmission
$u[3]^8$ 0 0 0 0 0 0 w_1 w_4
$c[3]^8 = u[3]^8 \boldsymbol{F}^{\otimes 3}$
1st decoding $\hat{u}[1]^8$ 0 0 $\hat{w}_1$ $\hat{w}_2$ $\hat{w}_3$ $\hat{w}_4$ $\hat{w}_5$ $\hat{w}_6$
2nd decoding
$\hat{u}[2]^8$ 0 0 0 0 0 $\hat{w}_1$ $\hat{w}_2$ $\hat{w}_3$
$\hat{u}[1]^8$ 0 0 $\hat{w}_1$ $\hat{w}_2$ $\hat{w}_3$ $\hat{w}_4$ $\hat{w}_5$ $\hat{w}_6$
3rd decoding
$\hat{u}[3]^8$ 0 0 0 0 0 0 $\hat{w}_1$ $\hat{w}_4$
$\hat{u}[2]^8$ 0 0 0 0 0 $\hat{w}_1$ $\hat{w}_2$ $\hat{w}_3$
$\hat{u}[1]^8$ 0 0 $\hat{w}_1$ $\hat{w}_2$ $\hat{w}_3$ $\hat{w}_4$ $\hat{w}_5$ $\hat{w}_6$

Figure 5.2.: Polar codes with IF, $k = 6$, $n_1 = n_2 = n_3 = 8$. Suppose the ascending reliability order of indices is $1, 2, 3, 5, 4, 6, 7, 8$. After the t-th transmission, t polar codes are decoded successively. Only the blue bits are treated as message bits at the decoder.

ity [58] asymptotically.

Note that this scheme is equivalent to dividing the $\left(\sum_{q=1}^{t} n_q, k\right)$ code into t separate polar codes and decoding them successively. This causes a large performance degradation for finite block lengths. For example, after the second transmission in Figure 5.2 the receiver decodes two separate $(8, 3)$ codes successively instead of one $(16, 6)$ code.

5.2.2. Polar Extension

The authors of [62] propose polar extension (PE). Consider the first transmission with an (n, k) polar codeword $c[1]^n$ with information set $\mathcal{A}_1$. For retransmission, one uses a $(2n, k)$ polar codeword $c[2]^{2n}$ with information set $\mathcal{A}_2$:

$$\begin{aligned} c[1]^n &= u[1]^n \boldsymbol{F}^{\otimes \log_2 n} \\ c[2]^{2n} &= u[2]^{2n} \boldsymbol{F}^{\otimes \log_2 n+1}. \end{aligned} \tag{5.3}$$

Let $\mathcal{A}_2^-$ and $\mathcal{A}_2^+$ be the information sets of $u[2]^n$ and $u[2]_{n+1}^{2n}$, respectively:

$$\begin{aligned} \mathcal{A}_2^- &= \{i : i \in [n], i \in \mathcal{A}_2\} \\ \mathcal{A}_2^+ &= \{i : i \in [n], i + n \in \mathcal{A}_2\}. \end{aligned} \tag{5.4}$$

We have

$$\left|\mathcal{A}_2^-\right| + \left|\mathcal{A}_2^+\right| = |\mathcal{A}_2| = |\mathcal{A}_1| \,. \tag{5.5}$$

Let $\mathcal{A}_2^+$ be a subset of $\mathcal{A}_1$ and set $u[2]_{n+1}^{2n} = u[1]^n$. Then copy the bits $u[1]_{\mathcal{A}_1 \setminus \mathcal{A}_2^+}$ to $u[2]_{\mathcal{A}_2^-}$:

$$\begin{aligned} &u[2]_{n+1}^{2n} = u[1]^n \\ &u[2]_{\mathcal{A}_2^-} = u[1]_{\mathcal{A}_1 \setminus \mathcal{A}_2^+} \\ &u[2]_i = 0, \ i \in [n] \setminus \mathcal{A}_2^- . \end{aligned} \tag{5.6}$$

For the structure in Figure 3.3, the second half of the codeword $c[2]^{2n}$ is

$$c[2]_{n+1}^{2n} = u[2]_{n+1}^{2n} \boldsymbol{F}^{\otimes \log_2 n} = u[1]^n \boldsymbol{F}^{\otimes \log_2 n} = c[1]^n \tag{5.7}$$

which was already transmitted. Thus, only the first half of the codeword $c[2]^{2n}$ is encoded and transmitted in the second transmission, i.e., we have

$$c[2]^n = u[2]^n \boldsymbol{F}^{\otimes \log_2 n} \oplus c[1]^n. \tag{5.8}$$

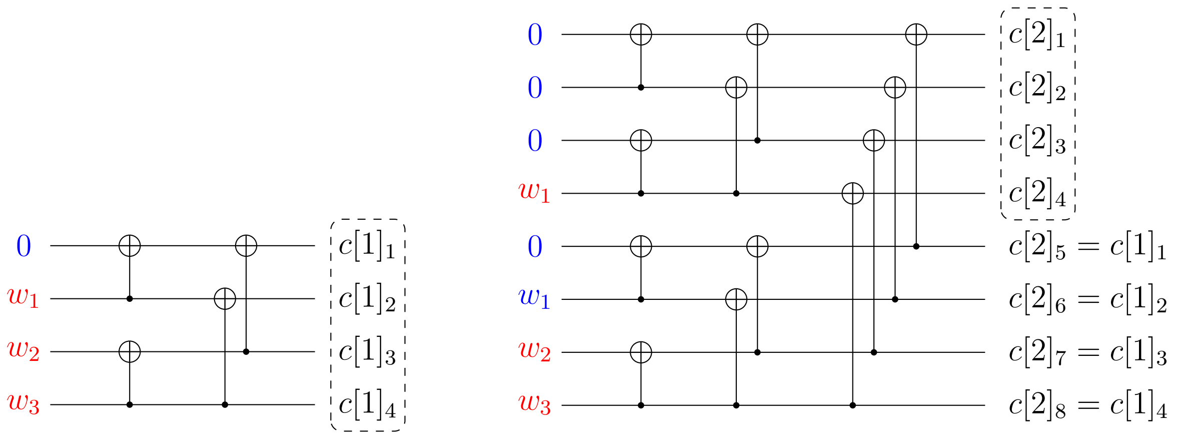
After the second transmission, the receiver concatenates the channel output from both transmissions to obtain the noisy version of the codeword $c[2]^{2n} = \left(c[2]^n, c[2]_{n+1}^{2n}\right)$. We decode $u[2]^{2n} = \left(u[2]^n, u[2]_{n+1}^{2n}\right) = (u[2]^n, u[1]^n)$ with dynamic frozen bits

$$u[1]_{\mathcal{A}_1 \setminus \mathcal{A}_2^+} = u[2]_{\mathcal{A}_2^-}. \tag{5.9}$$

This extension is repeated until the decoding is successful. An example is shown in Figure 5.3.

Compared with IF, PE has better performance by decoding a $\left(\sum_{q=1}^{t} n_q, k\right)$ code instead of t separate codes, see Figure 5.4. However, PE is not flexible because the packet length must be the same as the sum of all previous transmissions, i.e., one requires

$$n_t = \sum_{q=1}^{t-1} n_q = 2^{t-2} n_1, \ t = 2, 3, \ldots \tag{5.10}$$



(a) 1st Transmission (b) 2nd Transmission

Figure 5.3.: Polar codes with PE, $k = 3$, $n_1 = n_2 = 4$. We assume that $\mathcal{A}_1 = \{2, 3, 4\}$ and $\mathcal{A}_2 = \{4, 7, 8\}$. Only the bits in the dashed box are transmitted. The red bits are message bits, while the blue bits are statically or dynamically frozen. After the second transmission, the receiver concatenates the channel output to obtain the noisy version of $c[2]^8$. $c[2]^8$ is now a codeword of an $(8, 3)$ polar code with dynamic frozen bits $u_6 = u_4$.

Theorem 5.1. The PE scheme achieves capacity asymptotically in the following sense: for any integer $t \geq 1$, if the capacity of the channel satisfies

$$\frac{k}{2^t n_1} \leq \mathrm{C} < \frac{k}{2^{t-1} n_1} \tag{5.11}$$

then PE achieves a rate of $k/(2^t n_1)$ reliably, where n_1 is a power of two.

Remark. Similar to IF [58, Section 2.C], PE is not truly capacity-achieving in the sense of achieving arbitrary rate.

5.3. Variable-Length Polar Extension

We proposed variable-length polar extension (VLPE) based on QUP in [125]. The scheme generalizes PE and supports any packet length. We show that a QUP polar code punctured from long mother codes does not introduce more encoding and decoding complexity.

Theorem 5.2. Let $k \leq n$. An (n, k, N) QUP polar code has the first $N - n$ bits of u^N are frozen.

Proof. Consider the single level mutual information evolution in Figure 3.7. We have

$$I^- + I^+ = I_1 + I_2 \tag{5.12}$$

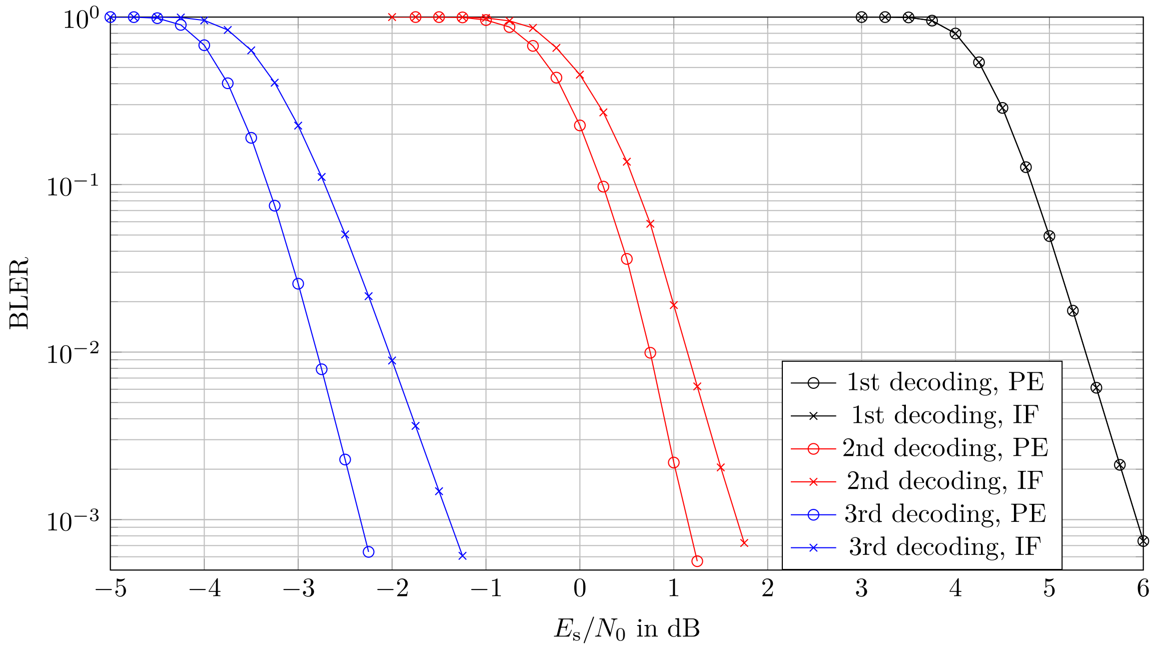


Figure 5.4.: Comparison of IF and PE over biAWGN channels with SC decoding and $k = 768$, $n_1 = n_2 = 1024$, $n_3 = 2048$. The polar codes are constructed via GA for $\{5, 1, -3\}$ dB for the 1st, 2nd and 3rd transmissions, respectively.

where $I^- \leq \min\{I_1, I_2\}$ and $I^+ \geq \max\{I_1, I_2\}$. As mutual information is non-negative, we have

$$I^- = 0, \text{ if } I_1 = 0 \text{ or } I_2 = 0 \tag{5.13}$$

i.e., the channels with zero capacity are propagated to lower indices. With QUP we have

$$\mathbb{I}\left(C_i; Y_i\right) = 0, \ i \in [N - n]. \tag{5.14}$$

The recursive structure of $\boldsymbol{F}^{\otimes \log_2 N}$ implies that the zeros will propagate through the transform which gives

$$\mathbb{I}\left(U_i; Y^N \middle| U^{i-1}\right) = 0, \ i \in [N - n]. \tag{5.15}$$

■

Corollary 5.3. All $(n, k, 2^j N)$ QUP polar codes have the same encoding and decoding complexity as (n, k, N) QUP polar codes, where $N = 2^{\lceil \log_2 n \rceil}$ and $j \in \mathbb{N}$.

Proof. We start with $j = 1$. We have $N > n > k$. More than N bits are punctured, i.e.,

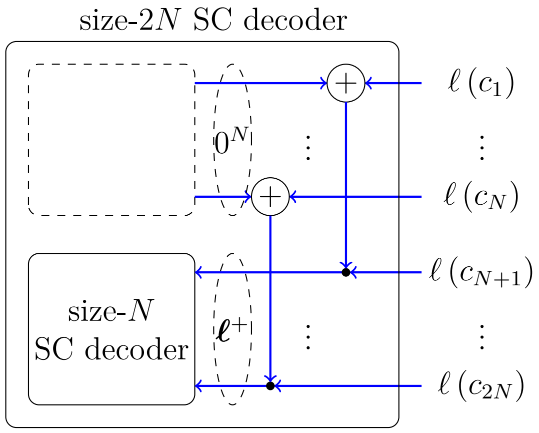


Figure 5.5.: Equivalence of (n, k) QUP polar codes punctured from mother code length N and $2N$.

$2N - n > N$. As a result of Theorem 5.2, the u^N are all frozen.

The encoder has $c^{2N} = (\boldsymbol{c}^- \oplus \boldsymbol{c}^+, \boldsymbol{c}^+)$, where

$$\begin{aligned} \boldsymbol{c}^- &= u^N \boldsymbol{F}^{\otimes \log_2 N} = 0^N \\ \boldsymbol{c}^+ &= u_{N+1}^{2N} \boldsymbol{F}^{\otimes \log_2 N}. \end{aligned} \tag{5.16}$$

We only need to compute the (n, k, N) QUP polar codeword $\boldsymbol{c}^+$ and transmit the last n bits of $\boldsymbol{c}^+$.

The decoder for $(n, k, 2N)$ QUP polar codes is shown in Figure 5.5. According to the SC decoding in Algorithm 3.1, we first decode $\boldsymbol{c}^-$ and then decode $\boldsymbol{c}^+$ based on $\hat{\boldsymbol{c}}^-$. We have

$$\begin{aligned} \ell(c_i) &\overset{\text{(a)}}{=} 0,\ i \in [N] \\ c_i^- &\overset{\text{(b)}}{=} 0,\ i \in [N] \end{aligned} \tag{5.17}$$

where (a) follows by puncturing and (b) follows because the u^N are all frozen. The input of the decoder for $\boldsymbol{c}^+$ is

$$\ell_i^+ = f^+(0, \ell(c_{N+i}), 0) = \ell(c_{N+i}),\ i \in [N]. \tag{5.18}$$

Thus for $(n, k, 2N)$ QUP polar codes, we only need to run the decoder for $\boldsymbol{c}^+$ with input $\ell(c_{N+i}),\ i \in [N]$.

The same idea extends the corollary to the mother code of length $2^j N,\ j > 1$. ■

Hence, we use (n, k) to describe a $\left(n, k, 2^{\lceil \log_2 n \rceil + j}\right)$ QUP polar code for any $j \in \{0, \mathbb{N}\}$.

Algorithm 5.1: VLPE: Design the t-th transmission via GA

```
Input : message length $k$, mother code length $N$,
        $t$-th packet length $n_t$,
        previous packet length $n'_t = \sum_{q=1}^{t-1} n_q$,
        design mutual information $I_t$,
        previous information set $\mathcal{A}'_t = \bigcup_{q=1}^{t-1} \mathcal{A}_q$ and frozen set $\mathcal{F}'_t = \bigcup_{q=1}^{t-1} \mathcal{F}_q$
Output: information set $\mathcal{A}_t$, frozen set $\mathcal{F}_t$,
        dynamic frozen constraint
1 Estimate probabilities $p_i$ (3.36) via GA.
2 Set $p_{\mathcal{F}'_t} = 1$. // the frozen bits from previous transmissions must be frozen
3 Find $k$ smallest $p_i$ and put their indices in $\mathcal{A}_t$. Frozen set is $\mathcal{F}_t = \bigcup_{q=1}^{t} \mathcal{I}_q \setminus \mathcal{A}_t$.
4 if $t \neq 1$ then
5     Dynamic frozen constraint is given by $u_{\mathcal{A}'_t \setminus \mathcal{A}_t} = u_{\mathcal{A}_t \setminus \mathcal{A}'_t}$.
6 return $\mathcal{A}_t$, $\mathcal{F}_t$, $u_{\mathcal{A}'_t \setminus \mathcal{A}_t} = u_{\mathcal{A}_t \setminus \mathcal{A}'_t}$
```

5.3.1. Detailed Description

Suppose the IR-HARQ system is designed for a maximum of $t_{\max}$ transmissions. Let the length of a mother polar code be

$$N = 2^{\left\lceil \log_2 \left(\sum_{q=1}^{t_{\max}} n_q \right) \right\rceil} \tag{5.19}$$

where n_q is the length of the q-th transmission. In the proposed scheme, a $\left(\sum_{q=1}^{t} n_q, k\right)$ QUP polar code is decoded after the t-th transmission. The main structure of the algorithm is shown in Figure 5.6 and Algorithm 5.1. The coded bits $c_{\mathcal{I}_t}$ are sent in the t-th transmission, where

$$\mathcal{I}_t = \left\{ N - \sum_{q=1}^{t} n_q + 1, \ldots, N - \sum_{q=1}^{t-1} n_q \right\} \tag{5.20}$$

and $\mathcal{A}_t$ and $\mathcal{F}_t$ are the information and frozen sets of the $\left(\sum_{q=1}^{t} n_q, k\right)$ QUP polar code after the t-th transmission, respectively. The dynamic frozen constraint is used for encoding and decoding:

- ▷ The encoder copies the information bits to the extended part, i.e., $u_{\mathcal{A}_t \setminus \mathcal{A}'_t} = u_{\mathcal{A}'_t \setminus \mathcal{A}_t}$.
- ▷ The decoder dynamically freezes the bits in $u_{\mathcal{A}'_t \setminus \mathcal{A}_t}$, i.e., $\hat{u}_{\mathcal{A}'_t \setminus \mathcal{A}_t} = \hat{u}_{\mathcal{A}_t \setminus \mathcal{A}'_t}$.

Note that $\mathcal{A}_t \cup \mathcal{F}_t = \bigcup_{q=1}^{t} \mathcal{I}_q$. The first $N - \sum_{q=1}^{t} n_q$ bits are frozen to zero (as a result of Theorem 5.2) but their indices are neither in $\mathcal{A}_t$ nor $\mathcal{F}_t$.

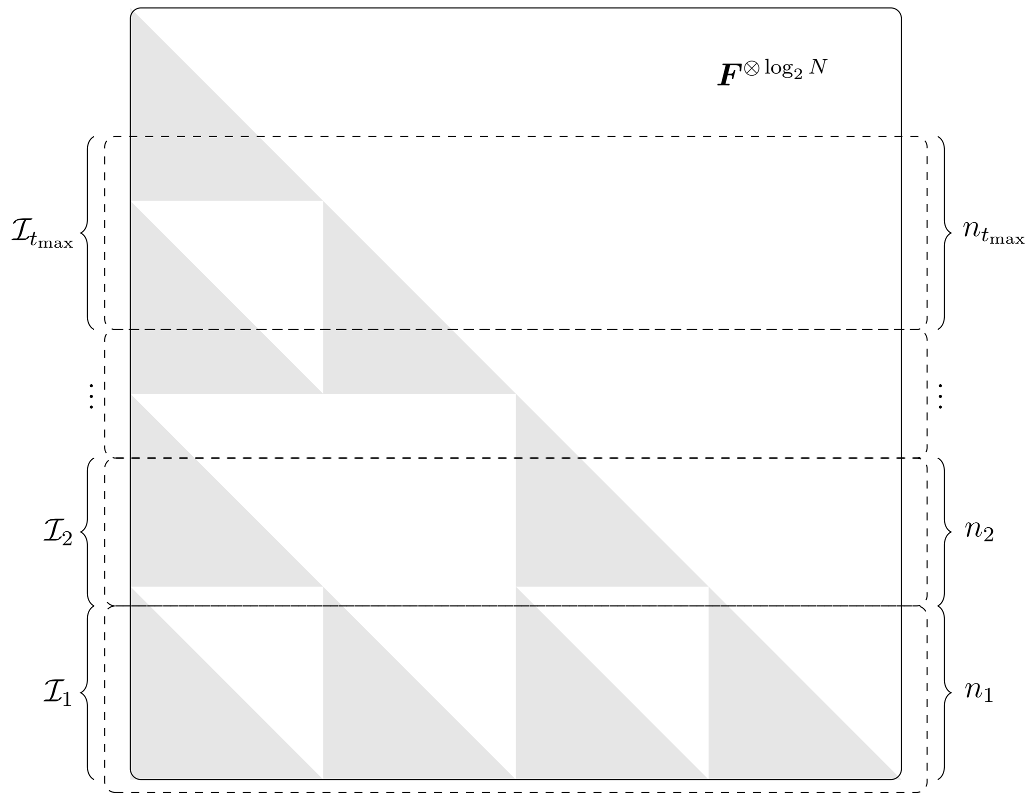


Figure 5.6.: Proposed VLPE scheme.

For the first transmission ($t = 1$), the code is simply a QUP polar code. For $t \geq 1$, the codes are extended from previous codes with dynamic frozen bits. The frozen bits must be frozen for all extensions, while the message bits could to converted to dynamic frozen bits in the extensions. Because $\boldsymbol{F}^{\otimes \log_2 N}$ is a lower triangular matrix, the bits in c_{i+1}^{N} are not changed by adjusting the bit u_i. The vector $c_{\mathcal{I}_t}$ is transmitted in the t-th transmission.

Example 5.1. Consider $k = 5$, $n_1 = 7$ and $n_2 = 5$. The message bits are w^5. The mother code length is $N = 2^{\lceil \log_2(n_1+n_2) \rceil} = 16$ and we have $\mathcal{I}_1 = \{10, 11, 12, 13, 14, 15, 16\}$, $\mathcal{I}_2 = \{5, 6, 7, 8, 9\}$.

- ▷ For the first transmission, we find $\mathcal{A}_1 = \{12, 13, 14, 15, 16\}$ for the $(7, 5, 16)$ QUP polar code. We precode u^{16} with

$$\begin{cases} u_{\mathcal{A}_1} = w^5 \\ u_i = 0, \text{ if } i \in [N],\ i \notin \mathcal{A}_1. \end{cases} \tag{5.21}$$

After the transform $c^{16} = u^{16} \boldsymbol{F}^{\otimes 4}$, the last 7 bits $c_{\mathcal{I}_1} = c_{10}^{16}$ are transmitted.

- ▷ For the second transmission, we find $\mathcal{A}_2 = \{8, 12, 14, 15, 16\}$ for the $(12, 5, 16)$ QUP polar code. We adjust u^{16} with

$$u_{\mathcal{A}_2 \setminus \mathcal{A}_1} = u_{\mathcal{A}_1 \setminus \mathcal{A}_2} \tag{5.22}$$

  which is $u_8 = u_{13}$ in this example. We perform the transform[1] $c^{16} = u^{16}\boldsymbol{F}^{\otimes 4}$ and transmit $c_{\mathcal{I}_2} = c_5^9$.

Because $\boldsymbol{F}^{\otimes 4}$ is a lower triangular matrix, adjusting $u_8 = 0$ to $u_8 = u_{13}$ does not change $c_{\mathcal{I}_1} = c_{10}^{16}$. The receiver decodes the $(12, 5, 16)$ QUP polar code from the noisy version of c_5^{16}. Note that u_{13} is now a dynamic frozen bit $\hat{u}_{13} = \hat{u}_8$. In this example, the QUP polar codes with optimal information set are decoded after every single transmission by using all the received information.

This scheme is equivalent to PE in [62] if n_1 is a power of two and $n_t = 2^{t-2}n_1,\ t = 2, 3, \ldots, t_{\max}$.

5.3.2. Design Examples and Numerical Results

As discussed in Section 5.3.1, the frozen bits must be frozen for the future extensions (`line 2` in Algorithm 5.1). We cannot guarantee that the polar code designed by VLPE is an optimal QUP polar code, since some of the reliable bits could be frozen in previous transmissions.

In Figure 5.7, Figure 5.8 and Figure 5.9, the performances of VLPE over biAWGN channels with short, moderate and long packet lengths are shown. The polar codes are designed via GA and are decoded with SC. The $\left(\sum_{q=1}^{t} n_q, k\right)$ QUP polar codes (dashed curves) serve as a reference and cannot be used for an IR-HARQ scheme. The simulation results show that the polar codes of VLPE are close to the optimal QUP polar codes.

Now consider 2^m-ary modulation over real valued AWGN channels $Y = X + Z$ with uniform distribution, i.e., we have

$$\begin{aligned} \mathcal{X} &= \{\pm 1\Delta, \pm 3\Delta, \pm (2^m - 1)\Delta\} \\ P_X(x) &= \frac{1}{2^m},\ \forall x \in \mathcal{X}, \end{aligned} \tag{5.23}$$

where $Z \sim \mathcal{N}(0, \sigma^2)$ and Δ is a scaling factor to adjust the transmission power. The SNR is $E_\mathrm{s}/N_0 = \mathbb{E}[X^2]/\sigma^2$.

[1] Practically, we do not have to perform the full transform $\boldsymbol{F}^{\otimes 4}$ because we know that $u_{\mathcal{I}_1}$ and $c_{\mathcal{I}_1}$ will not be changed.

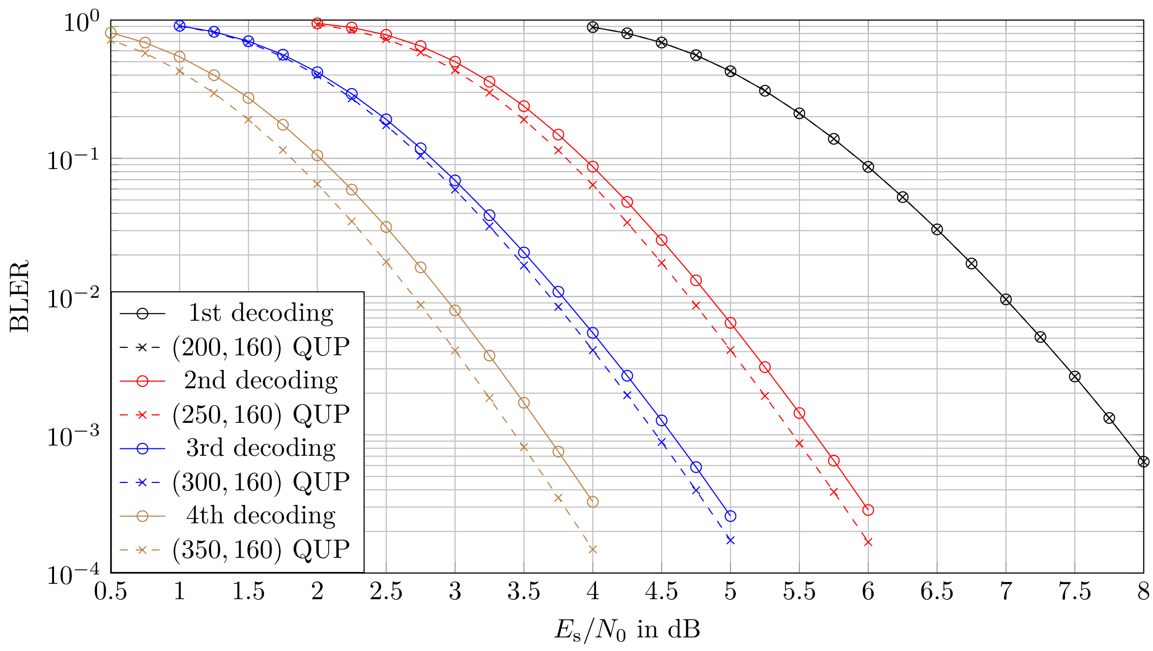


Figure 5.7.: VLPE over biAWGN channel with SC decoding, $k = 160$, $n_1 = 200$, $n_2 = n_3 = n_4 = 50$. The design SNR is $\{6, 4, 3, 2\}$ dB for the 1st to 4th transmission, respectively.

The (n_c, m, k) QUP-MLPC consists of m component QUP polar codes of length n_c. Multilevel VLPE is an extension of VLPE for higher-order modulation based on QUP-MLPC. In Figure 5.10, Figure 5.11 and Figure 5.12, the performances of multilevel VLPE with SP labeling over AWGN channels with short, moderate and long packet lengths are shown. The codes are designed via a surrogate channel based GA and are decoded with SC. The $\left(\sum_{q=1}^{t} n_{c,q}, m, k\right)$ QUP-MLPC curves (dashed) serve only as a reference and cannot be used for an IR-HARQ scheme. The simulation results show that the multilevel VLPE after every single transmission performs very close to optimal QUP-MLPC.

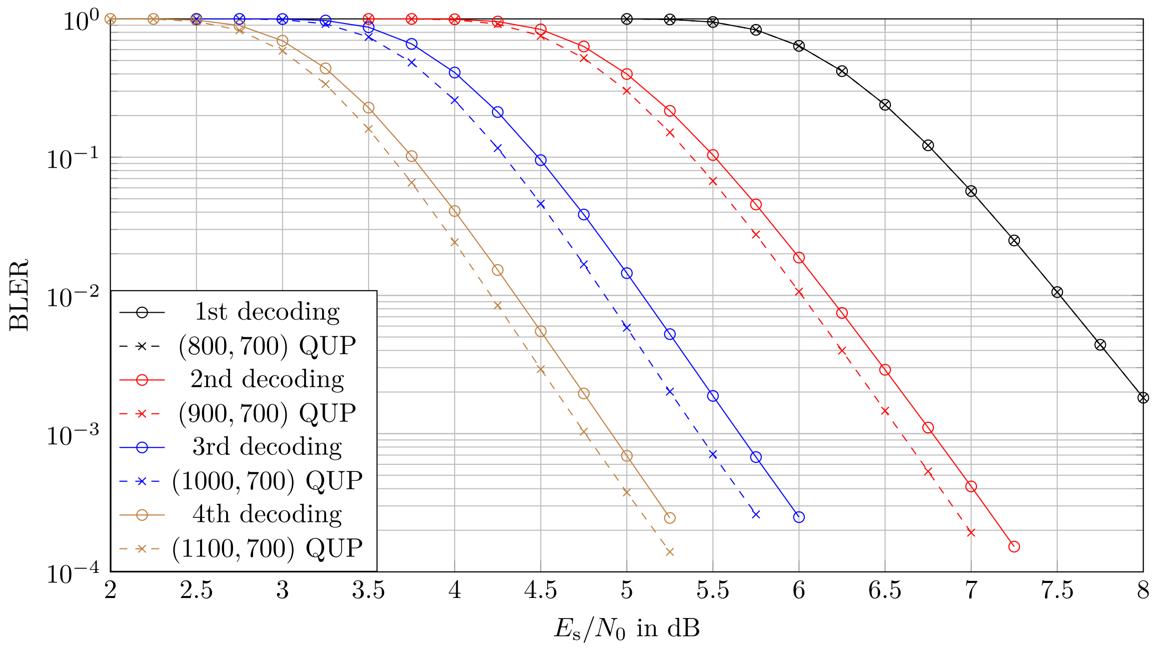


Figure 5.8.: VLPE over biAWGN channel with SC decoding, $k = 700$, $n_1 = 800$, $n_2 = n_3 = n_4 = 100$. The design SNR is $\{7, 5.5, 4.5, 4\}$ dB for the 1st to 4th transmissions, respectively.

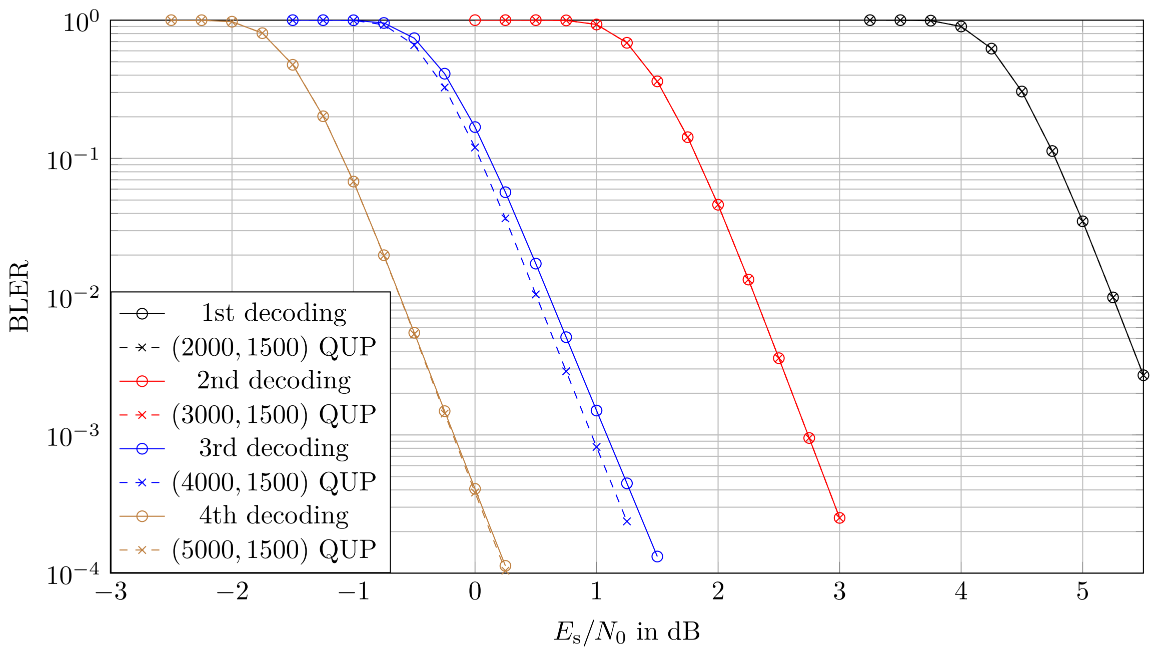


Figure 5.9.: VLPE over biAWGN channel with SC decoding, $k = 1500$, $n_1 = 2000$, $n_2 = n_3 = n_4 = 1000$. The design SNR is $\{5, 2, 0, -1\}$ dB for the 1st to 4th transmissions, respectively.

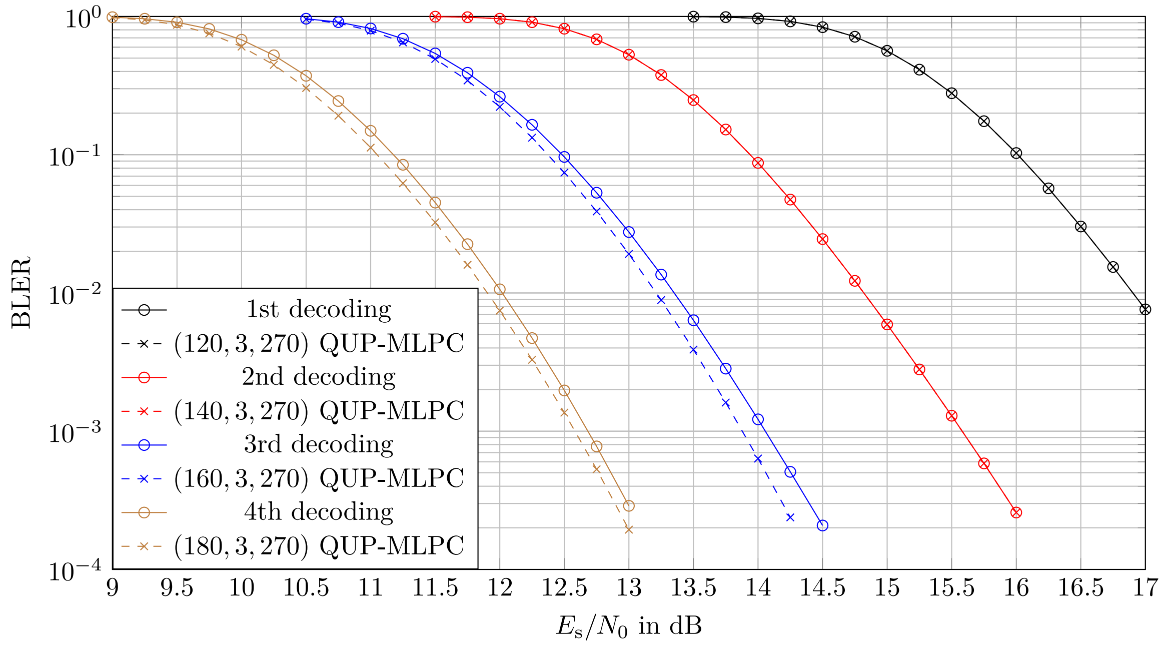


Figure 5.10.: VLPE over AWGN channel with 8-ASK modulation and SC decoding, $k = 270$, $n_{c,1} = 120$, $n_{c,2} = n_{c,3} = n_{c,4} = 20$. The design SNR is $\{16, 14, 13, 11\}$ dB for the 1st to 4th transmissions, respectively.

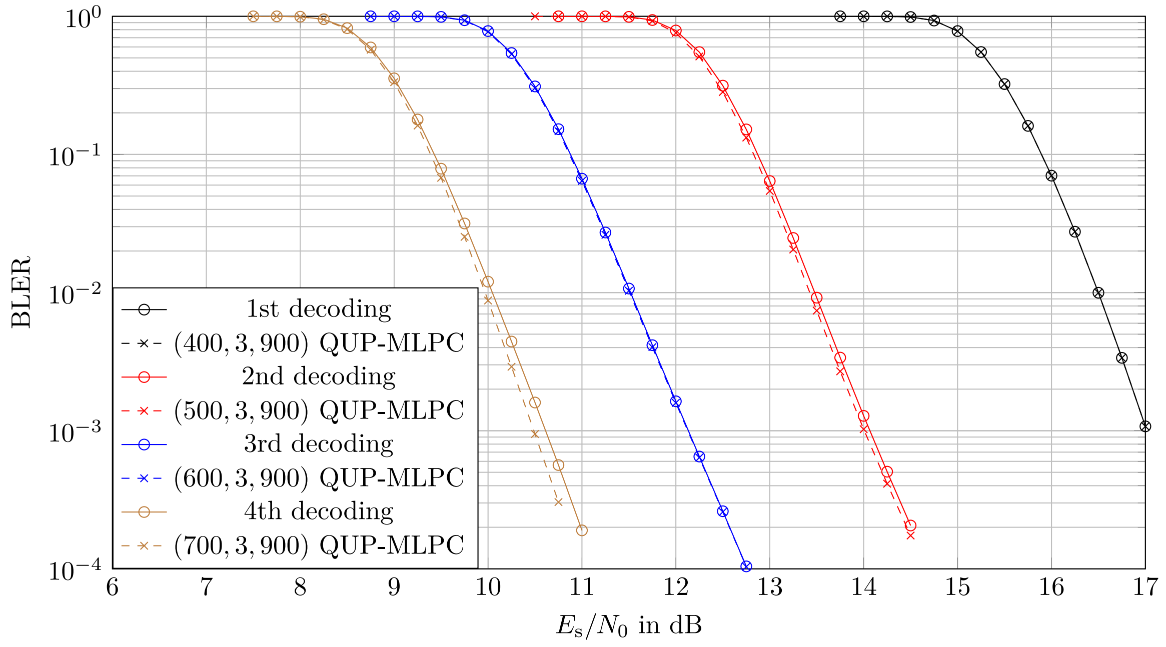


Figure 5.11.: VLPE over AWGN channel with 8-ASK modulation and SC decoding, $k = 900$, $n_{c,1} = 400$, $n_{c,2} = n_{c,3} = n_{c,4} = 100$. The design SNR is $\{16, 13, 11, 10\}$ dB for the 1st to 4th transmissions, respectively.

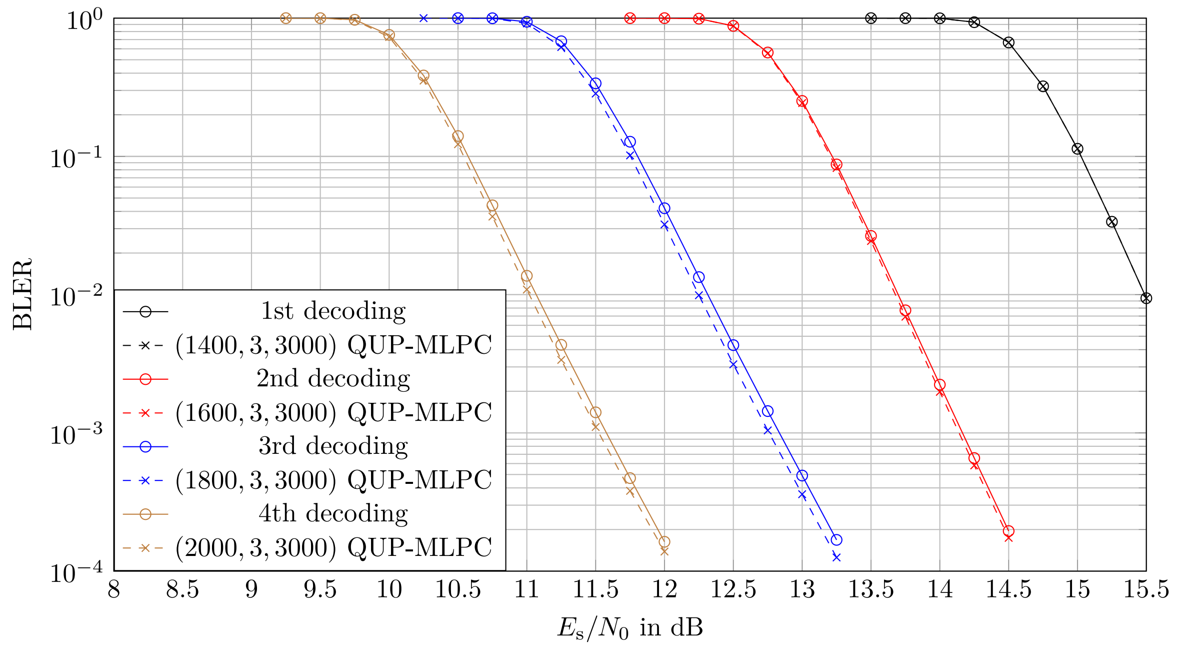


Figure 5.12.: VLPE over AWGN channel with 8-ASK modulation and SC decoding, $k = 3000$, $n_{c,1} = 1400$, $n_{c,2} = n_{c,3} = n_{c,4} = 200$. The design SNR is $\{15, 13, 11.5, 10.5\}$ dB for the 1st to 4th transmissions, respectively.

6 Polar-Coded Channel Estimation

The main results of this chapter appear in [122] and we add more details to help understand the performance of the proposed polar-coded channel estimation (PCCE) scheme.

The communication setting where channel state information (CSI) is not available at the transmitter or receiver is known as *non-coherent* communication [11, Section 10.7]. A common approach to address the lack of CSI is to embed pilot symbols in the transmitted symbol string, have the receiver estimate the CSI based on the pilots, and use the estimated CSI to decode. This approach is called pilot-assisted transmission (PAT) [109] with mismatched decoding [38, Exercise 5.22], [55, 68, 96, 105, 106].

PAT has two disadvantages for short block lengths: mismatched decoding reduces reliability and pilot symbols reduce rate significantly at low to moderate SNR [29, 61, 80, 105, 106]. Both problems can be partially mitigated with sophisticated signal processing. For instance, one may use iterative channel estimation and decoding [24, 42, 43, 51, 69, 79, 119], or two-stage algorithms that consider pilot symbols as part of the codebook [19, 120], or even ML decoding. Nevertheless, there is a fundamental performance degradation due to using pilot symbols [80].

We propose a PCCE scheme and then a pilot-free two-stage polar-coded transmission scheme [122] to jointly estimate the CSI and data with an adjustable complexity that can be made comparable to PAT. In the first stage, SCL decoding and the polar code constraints are used to estimate the CSI. In the second stage, mismatched SCL decoding proceeds with with this estimate. Gains of up to 2 dB are shown at a BLER of 10^{-4} as compared to classic PAT schemes for several non-coherent settings.

A related method to estimate CSI uses the parity-check constraints of a LDPC code [47].

However, SCL decoding of polar codes naturally provides soft estimates of frozen bits. Moreover, polar codes are usually used with a high rate outer code [1, 103] that can resolve CSI ambiguities, e.g., the phase ambiguity when using quadrature phase-shift keying (QPSK) and Gray labeling [47]. Of course, one may consider outer codes for LDPC codes as well. Other low-complexity methods for non-coherent channels are described in, e.g., [20, 47, 67, 89, 116]. We remark that our focus is on QPSK but the ideas extend to higher-order modulations. One may also combine PAT and PCCE to optimize performance.

6.1. System Model

Consider a scalar block fading channel, i.e., the fading coefficient H is constant for n_b channel uses and changes independently across t coherence blocks, resulting in a frame size of $n_\mathrm{c} = n_\mathrm{b}t$ symbols. The channel output of the i-th coherence block is

$$\boldsymbol{y}_i = h_i \boldsymbol{x}_i + \boldsymbol{z}_i, \ i \in [t] \tag{6.1}$$

where $\boldsymbol{x}_i \in \mathcal{X}^{n_\mathrm{b}}$ and $\boldsymbol{y}_i \in \mathbb{C}^{n_\mathrm{b}}$ are the transmitted and received vectors, $h_i \in \mathbb{C}$ is a realization of H, and $\boldsymbol{z}_i$ is an AWGN vector whose entries are i.i.d. as $\mathcal{CN}(0, 2\sigma^2)$. Neither the transmitter nor the receiver knows h_i or even the probability distribution of H. We assume that the noise variance $2\sigma^2$ is known to the receiver; this may be justified by the slow time scale of receiver device variations as compared to fading due to mobility. A vector without subscripts denotes a concatenation of vectors or scalars, e.g., $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$, $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ and $\boldsymbol{h} = (h_1, \ldots, h_t)$.

Consider QPSK with Gray labeling (see Figure 6.1). The input alphabet is $\mathcal{X} = \{\pm\Delta \pm \mathrm{j}\Delta\}$, $\Delta > 0$, and we map the binary vector c^{2m} to $x^m \in \mathcal{X}^m$ via $f_\mathrm{mod} : \{0,1\}^{2m} \mapsto \mathcal{X}^m$ as

$$f_\mathrm{mod}\left(c^{2m}\right) = (f_\mathrm{QPSK}(c_1, c_2), f_\mathrm{QPSK}(c_3, c_4), \ldots, f_\mathrm{QPSK}(c_{2m-1}, c_{2m})) \tag{6.2}$$

where $f_\mathrm{QPSK}(c_1, c_2) = (-1)^{c_1}\Delta + \mathrm{j}(-1)^{c_2}\Delta$. The mapping (6.2) is *symmetric*, i.e., if $f_\mathrm{mod}(c^{2m}) = \boldsymbol{x}$ then $f_\mathrm{mod}\left(\overline{c^{2m}}\right) = -\boldsymbol{x}$, where $\overline{c^m}$ denotes the bit-flipped version of c^m, i.e., $\overline{c^m} = (\overline{c}_1, \ldots, \overline{c}_m) = (c_1 \oplus 1, \ldots, c_m \oplus 1)$. The receiver of a block-wise non-coherent channel is described in Figure 6.2.

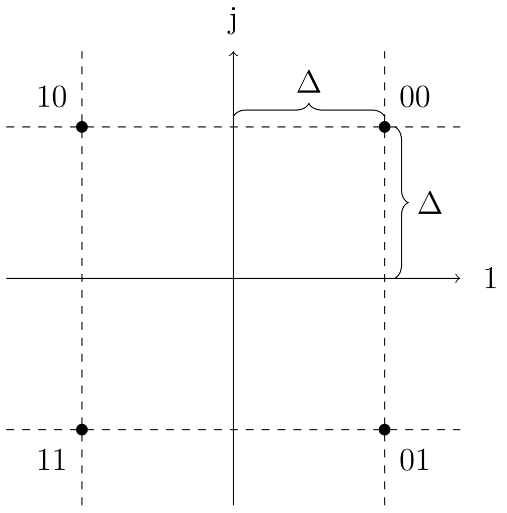


Figure 6.1.: QPSK with Gray labeling.

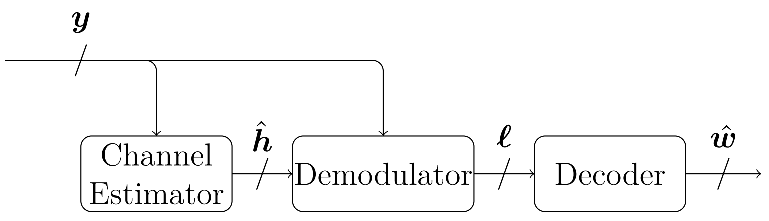


Figure 6.2.: Receiver structure.

6.1.1. Mismatched Log-Likelihood Ratios

Consider one symbol $x = f_{\text{QPSK}}\left(c_1, c_2\right)$ and the channel output y from the channel (6.1). Suppose first that the channel coefficient h is known to the receiver. The receiver can then form

$$\tilde{y} = \frac{1}{h} y = \frac{h^*}{|h|^2} y = x + \tilde{z} \tag{6.3}$$

where $\tilde{Z} \sim \mathcal{CN}\left(0, 2\sigma^2/|h|^2\right)$. The LLRs for bits c_1 and c_2 are

$$\begin{aligned} \ell\left(c_1\right) &= \log \frac{p_{Y|C_1,H}(y|0,h)}{p_{Y|C_1,H}(y|1,h)} = \frac{2\Delta\Re(y\,h^*)}{\sigma^2} \\ \ell\left(c_2\right) &= \log \frac{p_{Y|C_2,H}(y|0,h)}{p_{Y|C_2,H}(y|1,h)} = \frac{2\Delta\Im(y\,h^*)}{\sigma^2}. \end{aligned} \tag{6.4}$$

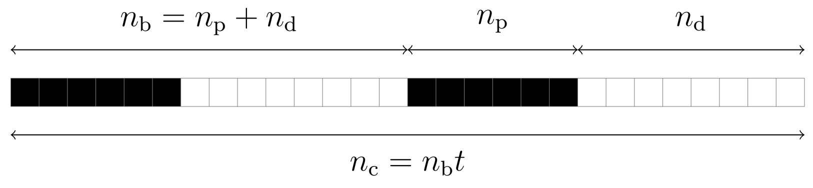

$n_\mathrm{b} = n_\mathrm{p} + n_\mathrm{d}$
n_p
n_d
$n_\mathrm{c} = n_\mathrm{b} t$

Figure 6.3.: PAT frame structure with $t = 2$ coherence blocks. Dark and white boxes represent pilot and coded symbols, respectively.

Suppose now that the channel coefficient is estimated as $\hat{h}$. The mismatched LLRs based on (6.4) are

$$\begin{aligned}\log \frac{p_{Y|C_1,H}(y|0,\hat{h})}{p_{Y|C_1,H}(y|1,\hat{h})} &= \frac{2\Delta\Re(y\,\hat{h}^*)}{\sigma^2}\\ \log \frac{p_{Y|C_2,H}(y|0,\hat{h})}{p_{Y|C_2,H}(y|1,\hat{h})} &= \frac{2\Delta\Im(y\,\hat{h}^*)}{\sigma^2}.\end{aligned} \tag{6.5}$$

6.1.2. Pilot-Assisted Transmission

Consider PAT as shown in Figure 6.3 where the first n_p symbols in coherence block i are pilot symbols $\boldsymbol{x}_{\mathrm{p},i}$ and the remaining $n_\mathrm{d} = n_\mathrm{b} - n_\mathrm{p}$ symbols $\boldsymbol{x}_{\mathrm{d},i}$ are coded, i.e., we have

$$\boldsymbol{x}_i = (\boldsymbol{x}_{\mathrm{p},i}, \boldsymbol{x}_{\mathrm{d},i}), \quad i \in [t]. \tag{6.6}$$

The pilot and coded symbols have the same energy. Upon observing the channel output $\boldsymbol{y}_i = (\boldsymbol{y}_{\mathrm{p},i}, \boldsymbol{y}_{\mathrm{d},i})$, the ML estimate of the CSI based on $\boldsymbol{y}_{\mathrm{p},i}$ is

$$\hat{h}_i = \operatorname*{argmax}_h p_{\underline{Y}|\underline{X}H}\left(\boldsymbol{y}_{\mathrm{p},i}\,|\boldsymbol{x}_{\mathrm{p},i}, h\right) = \frac{\boldsymbol{y}_{\mathrm{p},i}\cdot \boldsymbol{x}_{\mathrm{p},i}^\mathrm{H}}{\boldsymbol{x}_{\mathrm{p},i}\cdot \boldsymbol{x}_{\mathrm{p},i}^\mathrm{H}}. \tag{6.7}$$

A mismatched decoder uses $\hat{\boldsymbol{h}} = (\hat{h}_1, \ldots, \hat{h}_t)$ to compute the LLRs that are fed to the decoder that puts out a codeword estimate.

6.2. Joint Channel Estimation and Decoding

This section presents a low-complexity joint channel estimation and decoding scheme for polar codes. We do not use pilot symbols, i.e., we have $n_\mathrm{p} = 0$ and $\boldsymbol{x}_i = \boldsymbol{x}_{\mathrm{d},i}$. A random interleaver $\boldsymbol{\Pi}$ permutes the encoded bits $\boldsymbol{c}$ and is followed by the mapping (6.2). The channel model is (6.1). Let $h_i = r_i e^{\mathrm{j}\theta_i}$ where $r_i \in [0,\infty)$ and $\theta_i \in [0, 2\pi)$, $i \in [t]$.

We use the codebook for channel estimation, i.e., for polar codes with the code constraint $U_{\mathcal{F}} = \mathbf{0}$ we have

$$\hat{h} = \operatorname*{argmax}_{h} p_{\underline{Y}|U_{\mathcal{F}}H}\left(\boldsymbol{y}\,|\mathbf{0}, h\right). \tag{6.8}$$

However, for phase-shift keying (PSK) we simplify the estimation by separately estimating $r_i = |h_i|$. For a single coherence block, the received power is approximately the sum of the (received) signal power and the noise power, i.e., we have

$$\frac{1}{n_{\mathrm{b}}}\boldsymbol{y}_i \cdot \boldsymbol{y}_i^{\mathrm{H}} = \frac{1}{n_{\mathrm{b}}}\sum_{k=1}^{n_{\mathrm{b}}} |y_{i,k}|^2 \approx |h_i|^2 \left(2\Delta^2\right) + 2\sigma^2 \tag{6.9}$$

where the approximation becomes more accurate as n_{b} grows by the law of large numbers. Hence, we estimate the r_i as

$$\hat{r}_i = \frac{1}{\sqrt{2}\Delta} \cdot \sqrt{\frac{1}{n_{\mathrm{b}}}\left(\boldsymbol{y}_i \cdot \boldsymbol{y}_i^{\mathrm{H}}\right) - 2\sigma^2}, \quad i = 1, \ldots, t. \tag{6.10}$$

Let β be a number of input bits used for channel estimation, and let $\mathcal{A}_\beta = \mathcal{A} \cap [\beta]$ and $\mathcal{F}_\beta = \mathcal{F} \cap [\beta]$ be sets of information and frozen indices, respectively, among the first β input bits u^β. We use the polar code constraints to estimate the phase as

$$\begin{aligned}\left\{\hat{\theta}_1, \ldots, \hat{\theta}_t\right\} &= \operatorname*{argmax}_{\{\theta_1,\ldots,\theta_t\}} p_{\underline{Y}|U_{\mathcal{F}_\beta}\underline{H}}\left(\boldsymbol{y}\,\middle|\mathbf{0}, \hat{\boldsymbol{h}}\right)\\ &= \operatorname*{argmax}_{\{\theta_1,\ldots,\theta_t\}} \sum_{v_{\mathcal{A}_\beta}} p_{\underline{Y}U_{\mathcal{A}_\beta}|U_{\mathcal{F}_\beta}\underline{H}}\left(\boldsymbol{y}, v_{\mathcal{A}_\beta}\,\middle|\mathbf{0}, \hat{\boldsymbol{h}}\right)\end{aligned} \tag{6.11}$$

where $\hat{h}_i = \hat{r}_i e^{\mathrm{j}\theta_i}$, $i \in [t]$. The sum in (6.11) can be computed by SCL decoding up to decoding stage β with a list size $L_{\mathrm{e}} = 2^{|\mathcal{A}_\beta|}$. To reduce complexity at the expense of accuracy, one can approximate the calculation with SCL decoding and L_{e} satisfying $1 \le L_{\mathrm{e}} < 2^{|\mathcal{A}_\beta|}$. In fact, simulations in Section 6.3 show that small list sizes such as $L_{\mathrm{e}} = 8$ give BLER curves close to those of the coherent receiver.

Remark. The search space in (6.11) grows exponentially in the number of diversity branches t. There are several approaches to reduce complexity and we consider only the symmetry of the likelihood function due to the channel (6.1) and mapping (6.2) that halves the search space. We further adopt a coarse-fine search [47, 88] as an efficient optimizer.

Lemma 6.1. Polar-coded QPSK with the mapping (6.2) over the channel (6.1) has a sign ambiguity for the channel coefficients, i.e., for all $\boldsymbol{y}$, $\boldsymbol{h}$ and v^{N-1}, we have

$$p_{\underline{Y}|U\underline{H}}\left(\boldsymbol{y}\left|\left(v^{N-1},0\right),\boldsymbol{h}\right.\right)=p_{\underline{Y}|U\underline{H}}\left(\boldsymbol{y}\left|\left(v^{N-1},1\right),-\boldsymbol{h}\right.\right). \tag{6.12}$$

Proof. For all $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{h}$ and $\boldsymbol{s}\in\{-1,+1\}^t$, we have

$$p_{\underline{Y}|\underline{X}\underline{H}}\left(\boldsymbol{y}\left|\boldsymbol{x},\boldsymbol{h}\right.\right)=\prod_{i=1}^{t}p_{\underline{Y}_i|\underline{X}_iH_i}\left(\boldsymbol{y}_i\left|s_i\boldsymbol{x}_i,s_ih_i\right.\right) \tag{6.13}$$

as $s_i^2=1$. Recall that $c^N=\boldsymbol{\Pi}^{-1}\left(f_{\text{mod}}^{-1}(\boldsymbol{x})\right)$ so that $\overline{c^N}=\boldsymbol{\Pi}^{-1}\left(f_{\text{mod}}^{-1}(-\boldsymbol{x})\right)$. By choosing $s_i=-1,\ i\in[t]$, we have

$$p_{\underline{Y}|\underline{C}\underline{H}}\left(\boldsymbol{y}\left|\boldsymbol{c},\boldsymbol{h}\right.\right)=p_{\underline{Y}|\underline{C}\underline{H}}\left(\boldsymbol{y}\left|\overline{\boldsymbol{c}},-\boldsymbol{h}\right.\right). \tag{6.14}$$

Let v^N be the vector such that $c^N=v^N\boldsymbol{F}^{\otimes\log_2 N}$. We have

$$\overline{c^N}=\left(v^{N-1},v_N\oplus 1\right)\boldsymbol{F}^{\otimes\log_2 N} \tag{6.15}$$

because the last row of $\boldsymbol{F}^{\otimes\log_2 N}$ is all ones. ■

Lemma 6.1 implies that if a polar code is considered for (6.1) then the decoder cannot resolve the ambiguity on bit v_N. This ambiguity occurs for any binary linear block code that has a generator matrix with an all ones row, which is reflected in the bit v_N for polar codes.

Theorem 6.2. Polar-coded QPSK with the mapping (6.2) over the channel (6.1) satisfies

$$p_{\underline{Y}|U_{\mathcal{F}_\beta}\underline{H}}\left(\boldsymbol{y}\left|\boldsymbol{0},\boldsymbol{h}\right.\right)=p_{\underline{Y}|U_{\mathcal{F}_\beta}\underline{H}}\left(\boldsymbol{y}\left|\boldsymbol{0},-\boldsymbol{h}\right.\right) \tag{6.16}$$

$$p_{\underline{Y}|U^i\underline{H}}\left(\boldsymbol{y}\left|v^i,\boldsymbol{h}\right.\right)=p_{\underline{Y}|U^i\underline{H}}\left(\boldsymbol{y}\left|v^i,-\boldsymbol{h}\right.\right) \tag{6.17}$$

for all $\boldsymbol{y}$, $\boldsymbol{h}$, $\beta\in[N-1]$ and v^i, $i\in[N-1]$.

Proof. For $i\in[N-1]$, we have

$$p_{\underline{Y}|U^i\underline{H}}\left(\boldsymbol{y}\left|v^i,\boldsymbol{h}\right.\right)\overset{(a)}{=}\sum_{v_{i+1}^N}P_{U_{i+1}^N}\left(v_{i+1}^N\right)p_{\underline{Y}|U^N\underline{H}}\left(\boldsymbol{y}\left|v^N,\boldsymbol{h}\right.\right) \tag{6.18}$$

$$\overset{(b)}{=}\sum_{v_{i+1}^{N-1}}P_{U_{i+1}^{N-1}}\left(v_{i+1}^{N-1}\right)\left[\sum_{v_N}\frac{1}{2}p_{\underline{Y}|U^N\underline{H}}\left(\boldsymbol{y}\left|v^N,\boldsymbol{h}\right.\right)\right] \tag{6.19}$$

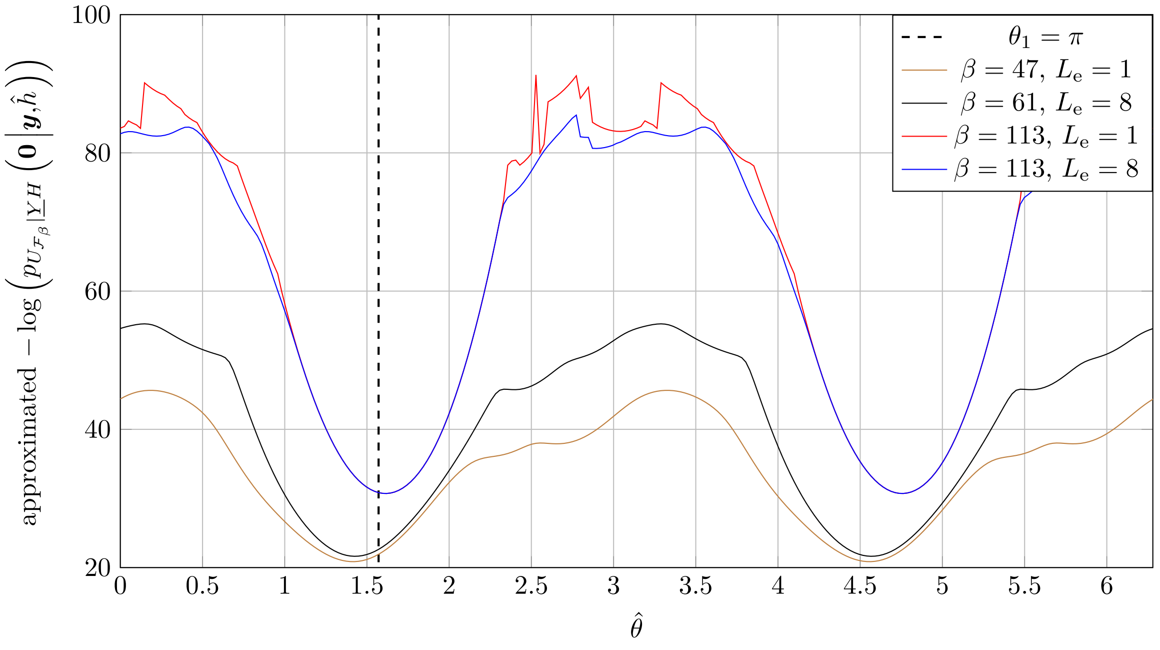


Figure 6.4.: PCCE for a $(128, 38)$ polar code with a PW construction over a single block fading channel ($t = 1$) at 2 dB. $\theta_1 = \pi$.

$$\overset{(c)}{=} \sum_{v_{i+1}^{N-1}} P_{U_{i+1}^{N-1}}\left(v_{i+1}^{N-1}\right)\left[\sum_{v_N}\frac{1}{2}p_{\underline{Y}|U^N\underline{H}}\left(\boldsymbol{y}\middle|v^N, -\boldsymbol{h}\right)\right] \tag{6.20}$$

$$\overset{(d)}{=} \sum_{v_{i+1}^{N}} P_{U_{i+1}^{N}}\left(v_{i+1}^{N}\right) p_{\underline{Y}|U^N\underline{H}}\left(\boldsymbol{y}\middle|v^N, -\boldsymbol{h}\right) \tag{6.21}$$

$$= p_{\underline{Y}|U^i\underline{H}}\left(\boldsymbol{y}\middle|v^i, -\boldsymbol{h}\right) \tag{6.22}$$

where step (a) follows by the law of total probability and the mutual independence of U^i, U_{i+1}^N and $\underline{H}$; steps (b) and (d) follow by rearranging the sums and noting that U_N is uniform; step (c) follows by Lemma 6.1. Next, expand

$$p_{\underline{Y}|U_{\mathcal{F}_\beta}\underline{H}}\left(\boldsymbol{y}\,|\mathbf{0}, \boldsymbol{h}\right) \overset{(a)}{=} \sum_{v_{\mathcal{A}_\beta}} P_{U_{\mathcal{A}_\beta}}\left(v_{\mathcal{A}_\beta}\right) p_{\underline{Y}|U^\beta\underline{H}}\left(\boldsymbol{y}\middle|v^\beta, \boldsymbol{h}\right) \tag{6.23}$$

$$\overset{(b)}{=} \sum_{v_{\mathcal{A}_\beta}} P_{U_{\mathcal{A}_\beta}}\left(v_{\mathcal{A}_\beta}\right) p_{\underline{Y}|U^\beta\underline{H}}\left(\boldsymbol{y}\middle|v^\beta, -\boldsymbol{h}\right) \tag{6.24}$$

$$= p_{\underline{Y}|U_{\mathcal{F}_\beta}\underline{H}}\left(\boldsymbol{y}\,|\mathbf{0}, -\boldsymbol{h}\right) \tag{6.25}$$

where step (a) follows by the law of total probability and mutually independent $U_{\mathcal{A}_\beta}$, $U_{\mathcal{F}_\beta}$ and $\underline{H}$; step (b) follows by (6.17). ■

Algorithm 6.1: Joint Channel Estimation and Decoding

```
Input  : the received vector $\boldsymbol{y}$
Output: the decoded word $\hat{u}^N$
1 estimate $\{\hat{r}_1, \dots, \hat{r}_t\}$ via (6.10)
2 estimate $\{\hat{\theta}_1, \dots, \hat{\theta}_t\} \in [0, 2\pi)^{t-1} \times [0, \pi)$ via (6.11)
3 run an SCL decoder with the LLRs obtained using $\hat{\boldsymbol{h}}$ and output the list $\mathcal{L}$ of $v^N$
4 obtain $\mathcal{L}'$ by flipping the last bit of all $v^N \in \mathcal{L}$
5 among all $v^N \in \mathcal{L} \cup \mathcal{L}'$ that pass the outer code test, choose the most likely one as $\hat{u}^N$
```

Theorem 6.2 (6.16) implies that the PCCE (6.11) outputs two solutions: $\{\hat{\theta}_1, \dots, \hat{\theta}_t\}$ and $\{\hat{\theta}_1 + \pi, \dots, \hat{\theta}_t + \pi\}$ where addition is modulo 2π. An example for $t = 1$ is shown in Figure 6.4. An outer code can resolve this ambiguity by optimizing over the set $[0, 2\pi)^{t-1} \times [0, \pi)$ to obtain $\{\hat{\theta}_1, \dots, \hat{\theta}_t\}$ by using the inner code constraints. The demodulator then feeds the SCL decoder with the LLR. Let $\mathcal{L}$ be the list of candidates v^N output by the decoder and define

$$\mathcal{L}' = \left\{ \left(v^{N-1}, \overline{v_N} \right) : v^N \in \mathcal{L} \right\}. \tag{6.26}$$

The outer code now eliminates invalid words in $\mathcal{L} \cup \mathcal{L}'$. Among the survivors, if any, the estimate $\hat{u}^N$ is chosen to maximize $p_{\underline{Y}|U^N\underline{H}}\left(\boldsymbol{y} \,\middle|\, v^N, \hat{\boldsymbol{h}}\right)$ if $v^N \in \mathcal{L}$ or $p_{\underline{Y}|U^N\underline{H}}\left(\boldsymbol{y} \,\middle|\, v^N, -\hat{\boldsymbol{h}}\right)$ if $v^N \in \mathcal{L}'$. An overview is given in Algorithm 6.1.

Remark. An outer code with a minimum distance of at least two can resolve the phase ambiguity.

Remark. The PCCE (6.11) is based on the imperfection of channel polarization. The estimation quality can be improved by freezing reliable bits.

6.3. Numerical Results

This section provides simulation results to compare the performance of PAT and PCCE. The SNR is expressed as E_s/N_0, where E_s is the energy per symbol and N_0 is the single-sided noise power spectral density. The inner code is a $(128, 38)$ polar code designed by PW and the outer code is a 6-bit CRC code with generator polynomial x^6+x^5+1, resulting in a $(128, 32)$ code. For the QPSK modulator (6.2) we have $n_\mathrm{c} = n_\mathrm{b}t = 64$ channel uses and an overall rate of $R = 0.5$ bpcu. For PAT, the $(128, 32)$ code is punctured to obtain $n_\mathrm{p}t$ pilot symbols in total, resulting in a $(128-2n_\mathrm{p}t, 32)$ QUP polar code. All curves shown in the figures below are for SCL decoding with a list size of $L = 8$ after estimating the CSI. The optimization (6.11) uses a coarse-fine search with 8 levels in both the coarse and fine search parts [88]. The performance is compared for various estimator parameters β and L_e and to the coherent receiver with perfect CSI. No puncturing is required for the coherent receiver. As discussed below, the gains of our scheme are similar for $t \in \{1, 2\}$ and with or without fading.

Single Coherence Block

Consider the channel (6.1) with $t = 1$, $r_1 = 1$, and the phase θ_1 is uniformly distributed on $[0, 2\pi)$, i.e., we have

$$y_i = e^{\mathrm{j}\theta_1} x_i + z_i,\ i \in [n_\mathrm{c}]\,,\ \Theta_1 \sim \mathcal{U}\,[0, 2\pi)\,. \tag{6.27}$$

Figure 6.5 compares PAT and PCCE. The best PAT performance for the BLER of interest is achieved with $n_\mathrm{p} = 14$, i.e., 14 pilot symbols gave the lowest SNR for BLER ranging from 10^{-2} to 10^{-4}. For smaller n_p the quality of the channel estimate limits performance, and for larger n_p the puncturing weakens the polar code and limits performance. PCCE performs within 0.3 dB of the receiver with perfect CSI if the estimator is run with $L_\mathrm{e} = 8$ and up to the last frozen bit with $\beta = 113$. It thereby outperforms PAT by about 1.5 dB at a BLER of 10^{-4}. Observe that if the estimator is run up to the last frozen bit before the first information bit, i.e., $\beta = 47$, then the performance is worse than for PAT. The parameters $\beta = 113$ and $L_e = 1$ provide a good tradeoff between complexity and performance when combined with a second-stage SCL decoding with a list size $L = 8$.

Table 6.1 compares the number of node-visits per codeword in the decoding tree along with the BLER at $E_\mathrm{s}/N_0 = 1$ dB. For PCCE, we state the sum of the number of node-visits by the estimator and the number of node-visits by the decoder. The number of visited nodes with PAT and perfect CSI is thus the same. Observe that PCCE with $\beta = 113$ and $L_\mathrm{c} = 1$ visits a similar number of nodes as PAT with a list size $L = 32$ (the difference is less than 10%) and it reduces the error probability by one order of magnitude. We remark that measuring the complexity by the number of node-visits is pessimistic for PCCE since most of the visited nodes are frozen bits. Hence, simplified SC decoders [3, 92–94] can significantly reduce complexity.

Figure 6.6 compares PCCE and PAT with *extra* pilot symbols. For instance, PCCE with $\beta = 113, L_\mathrm{e} = 1$ performs close to PAT with 15 pilot symbols. However, the PAT rate is now $R = 32/79 \approx 0.4$ bpcu rather than $R = 0.5$ bpcu.

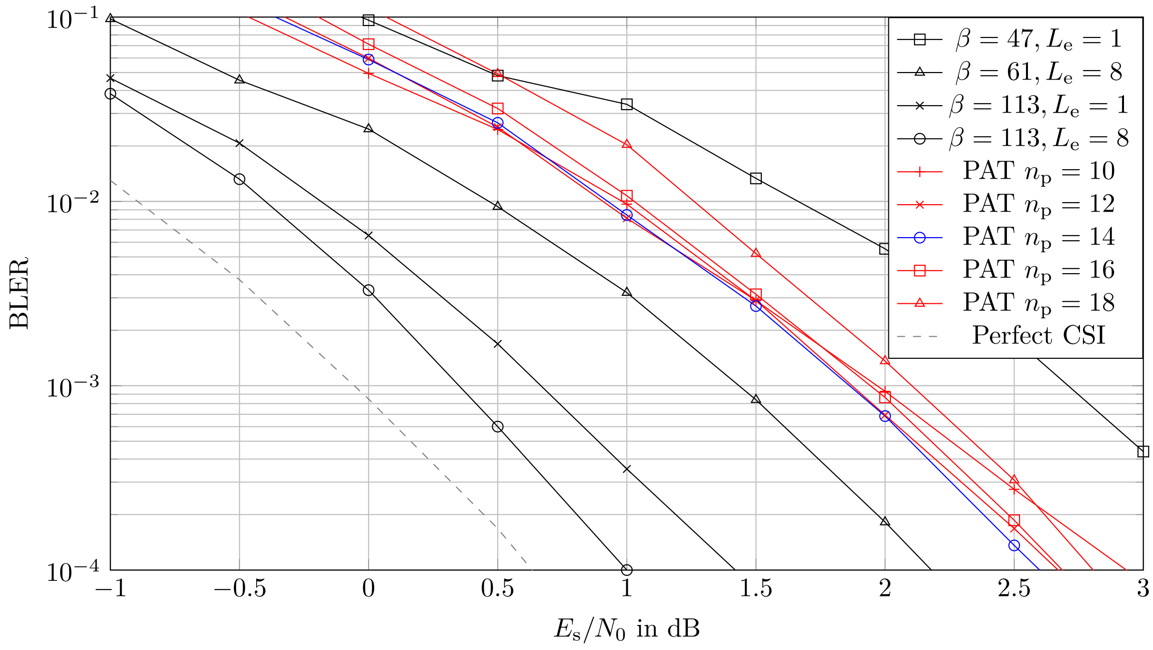


Figure 6.5.: Performance of PAT and PCCE for the channel (6.1) with $t = 1$, $r_1 = 1$, and $\Theta_1 \sim [0, 2\pi)$. SCL decoding uses a list size of $L = 8$ for all cases. A $(128, 32)$ polar code designed by PWs is used with QPSK for PCCE. PCCE uses a $8 + 8$ coarse-fine search for (6.11). $(128 - 2n_p, 32)$ QUP polar codes designed by PWs are used for PAT. The overall rate is 0.5 bpcu for all cases.

Two Coherence Blocks

We next consider $t = 2$ coherence blocks. Figure 6.7 shows the BLER for $r_i = 1$ and $\Theta_i \sim \mathcal{U}[0, 2\pi)$, $i \in \{1, 2\}$. Figure 6.8 shows the BLER for a Rayleigh block fading channel with $H_i \sim \mathcal{CN}(0, 1)$, $i \in \{1, 2\}$. The best performance for PAT is achieved with $n_p = 7$ pilot symbols per coherence block for both cases. Observe that, in both cases, PCCE outperforms PAT by about 2 dB at a BLER $\approx 10^{-4}$. Moreover, PCCE approaches the performance of a coherent receiver with perfect CSI.

Figure 6.8 also plots achievable BLERs based on random coding as the curve labeled “RCUB” [65, Theorem 1]. The curve labeled “meta-converse” is a lower bound on the BLERs [83, Theorem 28]. Both curves assume that there is a power constraint per coherence block rather than a codeword. Also, the input distribution is induced by unitary space-time modulation rather than QPSK. For more details, see [54].

Method	BLER	Number of node-visits
PAT ($n_\mathrm{p} = 14,\ L = 8$)	8.43×10^{-3}	631
PAT ($n_\mathrm{p} = 14,\ L = 32$)	3.16×10^{-3}	2223
PCCE ($\beta = 47,\ L_\mathrm{e} = 1,\ L = 8$)	3.36×10^{-2}	1383
PCCE ($\beta = 61,\ L_\mathrm{e} = 8,\ L = 8$)	3.20×10^{-3}	2151
PCCE ($\beta = 113,\ L_\mathrm{e} = 1,\ L = 8$)	3.50×10^{-4}	2439
PCCE ($\beta = 113,\ L_\mathrm{e} = 8,\ L = 8$)	1.00×10^{-4}	8807
Perfect CSI ($L = 8$)	2.40×10^{-5}	631

Table 6.1.: Performance and complexity comparison at SNR = 1 dB. The setting is the same as in Figure 6.5.

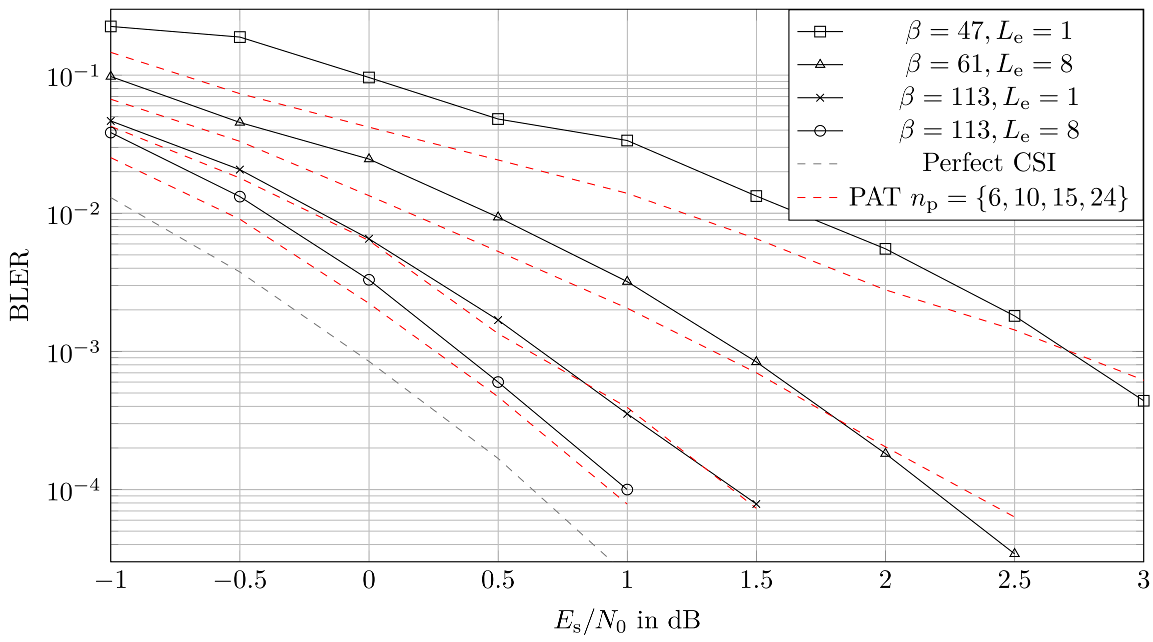


Figure 6.6.: Performance of PAT and PCCE for the channel (6.1) with $t = 1$, $r_1 = 1$, and $\Theta_1 \sim [0, 2\pi)$. A $(128, 32)$ polar code designed by PWs is used with QPSK for all cases. SCL decoding uses a list size of $L = 8$. PCCE uses a $8 + 8$ coarse-fine search for (6.11). Extra pilot symbols are used for PAT, i.e., the polar code is not punctured and the overall rate is $32/\left(64 + n_\mathrm{p}\right)$ bpcu.

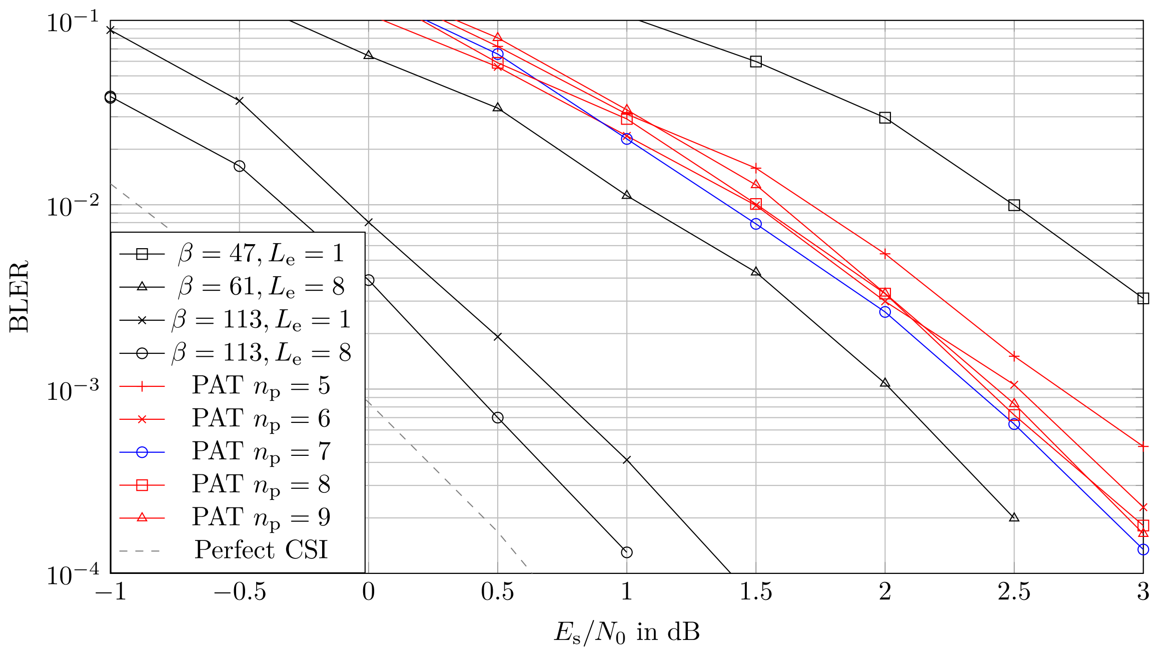


Figure 6.7.: Performance of PAT and PCCE for the channel (6.1) with $t = 2$, $r_i = 1$, and $\Theta_i \sim [0, 2\pi)$ for $i \in \{1, 2\}$. SCL decoding uses a list size of $L = 8$ for all cases. A $(128, 32)$ polar code designed by PWs is used with QPSK for PCCE. $(128 - 4n_p, 32)$ QUP polar codes designed by PWs are used for PAT. The overall rate is 0.5 bpcu for all cases.

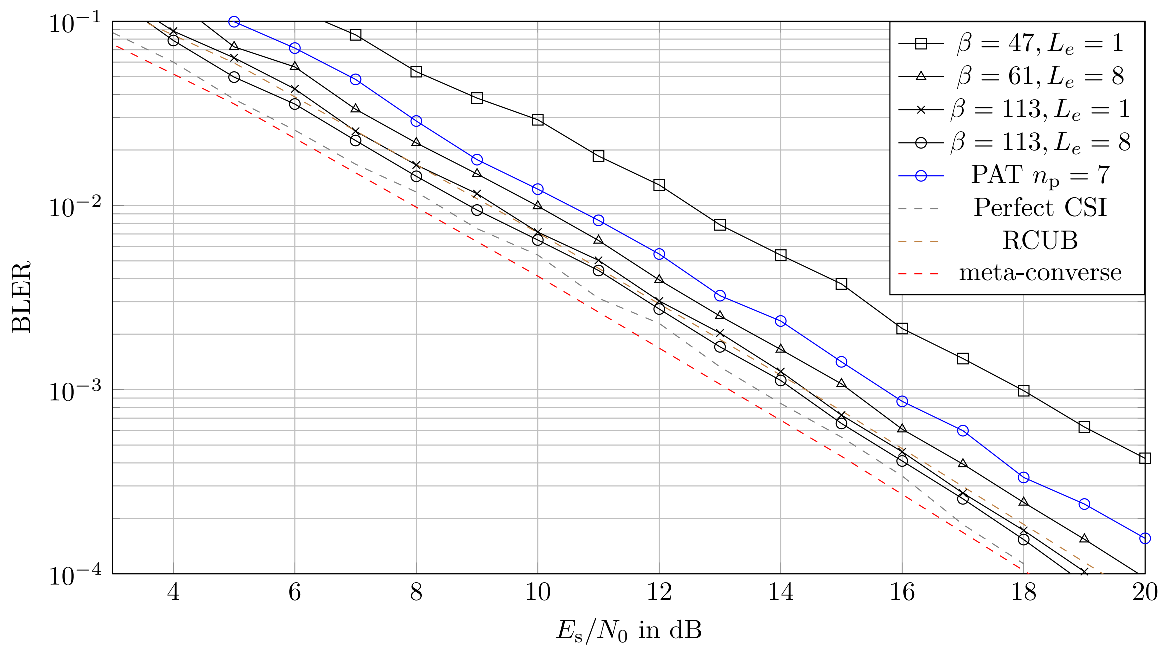


Figure 6.8.: Performance of PAT and PCCE for a Rayleigh block fading channel and $t = 2$, $H_i \sim \mathcal{CN}(0, 1)$ for $i \in \{1, 2\}$. SCL decoding uses a list size of $L = 8$ for all cases. A $(128, 32)$ polar code designed by PWs is used with QPSK for PCCE. $(128 - 4n_p, 32)$ QUP polar codes designed by PWs are used for PAT. The overall rate is 0.5 bpcu for all cases.

7 Conclusions and Outlook

We summarize the main contributions the thesis and suggest problems that deserve further attention.

- ▷ In Chapter 4, a complexity-adaptive tree search algorithm called SCOS was proposed for polar codes that implements ML decoding by using a SC schedule. The complexity is adapted to the channel quality and approaches the complexity of SC decoding for polar codes at high SNR. Furthermore, a lower bound on the complexity (4.4) was provided. By modifying the algorithm to limit the worst-case complexity, one still obtains near-ML performance. Unlike existing alternatives, the algorithm does not need an outer code (e.g., CRC codes for ASCL and DSCF, see Section 3.5) or a separate parameter optimization (e.g., parameter Δ for SC-Fano and α for DSCF). Future work could develop information theory to improve the complexity lower bound.
- ▷ Chapter 5 proposed a polar-coded IR-HARQ scheme based on QUP and dynamic frozen bits. The proposed VLPE was extended to PCM. Numerical results show that the rate-matched polar codes generated by VLPE perform almost as well as the basic QUP polar codes.
- ▷ Chapter 6 proposes PCCE by using the code constraints imposed by the frozen bits. Interestingly, PCCE takes advantage of the imperfection of channel polarization. We further introduced a pilot-free two-stage polar-coded transmission scheme to jointly estimate the channel state and message. An outer code improves reliability and resolves phase ambiguities. Numerical results show that the proposed scheme

significantly outperforms PAT with a similar complexity and approaches the performance of a coherent receiver. Future work should extend the results to higher-order modulation and multiple-input, multiple-output (MIMO) systems. Another interesting question for further research is to develop information theory to analyze the estimation quality.

A
Appendix

A.1. Selection of CRC Polynomials

This appendix introduces a scheme to find good generator polynomial of the CRC outer code for a given redundancy length r_{CRC} with polar SCL decoding.[1] We do this by targeting the ML performance; based on [123, Conjecture 1] the best ML polynomial should provide the best performance for any list size L.

The BLER under ML decoding is upper bounded the union bound (UB). For a biAWGN channel, the UB is given by

$$\text{BLER} \leq \text{UB} = \frac{1}{2}\sum_{w=1}^{N} A_w \text{erfc}\left(\sqrt{w\text{SNR}}\right) \tag{A.1}$$

where w and A_w denote the Hamming weight and code weight enumerator, respectively. However, the distance spectrum of polar codes with CRC outer codes is generally unknown.

In [56], the authors proposed a tool to analyze the distance spectrum by using list decoding. Suppose the list contains only the codewords with the least weights if the all zero codeword is transmitted over a channel with small noise variance. The algorithm works as follows.

Step 1. Transmit the all zero codeword at very high SNR.

[1]The SC performances are the same for any CRC generator polynomials for a $(N, k + r_{\text{CRC}})$ polar code with redundancy length r_{CRC}.

r_{CRC}	Polynomial	AUB	estimated SC BLER
3	`0x5`	1.9433×10^{-4}	4.2750×10^{-3}
4	`0xC`	1.6791×10^{-4}	5.1077×10^{-3}
5	`0x18`	8.9280×10^{-5}	6.4029×10^{-3}
6	`0x2D`	5.8441×10^{-6}	7.8608×10^{-3}
7	`0x72`	3.2828×10^{-6}	9.5868×10^{-3}
8	`0xA6`	2.4525×10^{-6}	1.1465×10^{-2}

Table A.1.: Best CRC polynomials for $(128, 64 + r_{\text{CRC}})$ polar codes with r_{CRC} bits CRC outer codes. The codes are designed via GA for 4 dB. The ML performance (evaluated by the AUB) and SC performance are estimated for a biAWGN channel at 4 dB.

Step 2. Perform list decoding with a very large list size on the received soft information for a $(N, k + r_{\text{CRC}})$ code and store the output list.

Step 3. Remove the candidates that do not satisfy the outer code check.

Step 4. Find all codewords with non-zero weight in the list and the corresponding multiplicities.

Step 5. Calculate the approximated union bound (AUB) with (A.1) by using the remaining candidates.

We perform Steps 1 and 2 only once, and repeat Steps 3-5 for $2^{r_{\text{CRC}}-1}$ polynomials.

An example is shown in Table A.1. The generator polynomials are described by a hexadecimal number (Koopman Notation [52]), e.g., `0x5b` denotes the generator polynomial $g(x) = x^7 + x^5 + x^4 + x^2 + x + 1$ ($r_{\text{CRC}} = 7$).

A.2. Polar Coding with Non-binary Kernels

We introduced polar codes based on 2×2 non-binary kernels in [124]. We consider only Galois fields (GFs) of order $q = 2^r$, $r \in \mathbb{N}$. Binary and decimal representations are used to describe elements over GFs, i.e., GF(q) has q elements and the elements can be represented by binary $\log_2 q$-tuples or integers[2] between 0 and $q - 1$. For the codes over GF(q), N_q denotes the code length in q-ary symbols, $N = \log_2 q \cdot N_q$ denotes the code length in bits, and k is the code dimension in bits.

[2]Without loss of generality, the left bit is most significant for the bit-to-integer conversion.

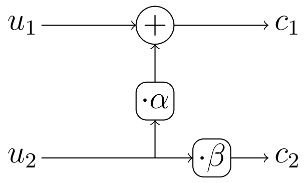


Figure A.1.: A 2×2 kernel over GF(q), $\alpha, \beta \in$ GF(q).

A non-binary polar code over GF(q) of length N_q and dimension k is defined by the q-ary transform $\boldsymbol{F}^{\otimes \log_2 N_q}$ and $N-k$ frozen (bit) positions, where $\boldsymbol{F}$ denotes the extended kernel

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 \\ \alpha & \beta \end{bmatrix}, \quad \alpha, \beta \in \mathrm{GF}(q). \tag{A.2}$$

Figure A.1 shows the q-ary transform of size $N_q = 2$.

The encoding procedure is represented by

$$c^{N_q} = u^{N_q} \boldsymbol{F}^{\otimes \log_2 N_q}. \tag{A.3}$$

The vectors u^{N_q}, c^{N_q} and all (addition, multiplication) operations are defined over GF(q). The vector c^{N_q} denotes the code symbols. The vector u^{N_q} can be represented by $N = \log_2 q \cdot N_q$ bits including the message bits and frozen bits. For a symbol $u_i \in$ GF(q), t_i bits are selected as message bits, where $t_i \in \{0, 1, \ldots, \log_2 q\}$. The set of all possibilities of symbol u_i is denoted $\mathcal{S}_i$. Obviously, we have

$$|\mathcal{S}_i| = 2^{t_i}, \quad \sum_{i=1}^{N_q} t_i = k. \tag{A.4}$$

To simplify the code design, we always freeze the last t_i bits for a symbol $u_i \in$ GF(q) in this work. The choice of symbol u_i is then restricted to

$$\mathcal{S}_i = \left\{0, \ldots, 2^{t_i} - 1\right\}. \tag{A.5}$$

A.2.1. Message Passing on Non-binary Graphs

Considering the decoder in the probability domain, each message is a vector with q probabilities

$$\boldsymbol{P}_A = \left(P_A(0), P_A(1) \ldots, P_A(q-1)\right). \tag{A.6}$$

We have three basic probability domain operations:

- ▷ Multiplication and addition: $B = A\alpha$ and

$$P_A(\mu) = P_B(\mu\alpha), \quad \mu \in \mathrm{GF}(q). \tag{A.7}$$

We can find a $q \times q$ permutation matrix $\mathbf{\Pi}_{\cdot\alpha}$ such that

$$\boldsymbol{P}_B = \boldsymbol{P}_A \mathbf{\Pi}_{\cdot\alpha}, \quad \boldsymbol{P}_A = \boldsymbol{P}_B \mathbf{\Pi}_{\cdot\alpha}^{-1}. \tag{A.8}$$

We can also find a permutation matrix $\mathbf{\Pi}_{+\alpha}$ for addition such that

$$\boldsymbol{P}_{A+\alpha} = \boldsymbol{P}_A \mathbf{\Pi}_{+\alpha}. \tag{A.9}$$

Note that $\mathbf{\Pi}_{\cdot\alpha}$ depends on the primitive polynomial.

- ▷ CN update: A degree-2 CN node computes $\boldsymbol{P}_{A+B}$ from $\boldsymbol{P}_A$ and $\boldsymbol{P}_B$.

$$P_{A+B}(\mu) = \sum_{\substack{\mu_1,\mu_2 \in \mathrm{GF}(q): \\ \mu_1+\mu_2=\mu}} P_A(\mu_1) P_B(\mu_2), \quad \mu \in \mathrm{GF}(q). \tag{A.10}$$

We have

$$\boldsymbol{P}_{A+B} = \boldsymbol{P}_A \circledast \boldsymbol{P}_B \tag{A.11}$$

where $\circledast$ denotes cyclic discrete convolution with complexity $\mathcal{O}(q^2)$ [25]. By applying the fast Hadamard transform (FHT), the discrete convolution is translated to element-wise multiplication [8] which reduces the complexity of the CN update to $\mathcal{O}(q \log_2 q)$. We have

$$\boldsymbol{P}_{A+B} = a\left(\mathcal{H}^{-1}\left(\mathcal{H}(\boldsymbol{P}_A) \odot \mathcal{H}(\boldsymbol{P}_B)\right)\right) \tag{A.12}$$

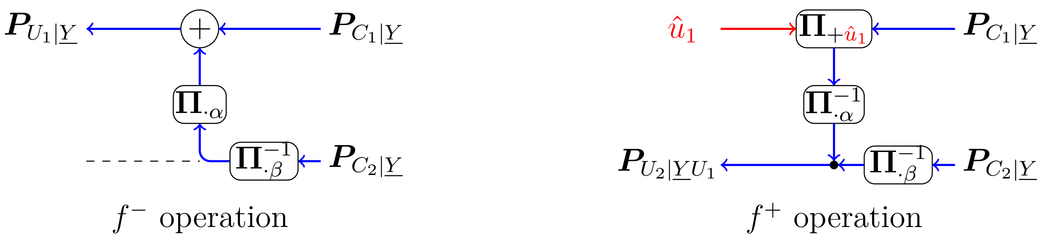
where the scalar a is a normalization factor that ensures that the probabilities in $\boldsymbol{P}_{A+B}$ sum to 1, $\mathcal{H}(\cdot)$ denotes the FHT operation and $\odot$ denotes element-wise multiplication. Note that the FHT is its own inverse, i.e., $\mathcal{H}(\cdot) = \mathcal{H}^{-1}(\cdot)$.

- ▷ VN update: A degree-2 CN node computes $\boldsymbol{P}_A$ from $\boldsymbol{P}_{A_1}$ and $\boldsymbol{P}_{A_2}$ for two independent observations A_1 and A_2, respectively. We have

$$P_A(\mu) = P_{A_1}(\mu) P_{A_2}(\mu),\ \mu \in \mathrm{GF}(q) \tag{A.13}$$

and

$$\boldsymbol{P}_A = a\left(\boldsymbol{P}_{A_1} \odot \boldsymbol{P}_{A_2}\right) \tag{A.14}$$



$\boldsymbol{P}_{U_1|\underline{Y}}$
$\boldsymbol{P}_{C_1|\underline{Y}}$
$\boldsymbol{\Pi}_{\cdot\alpha}$
$\boldsymbol{\Pi}_{\cdot\beta}^{-1}$
$\boldsymbol{P}_{C_2|\underline{Y}}$
f^- operation
$\hat{u}_1$
$\boldsymbol{\Pi}_{+\hat{u}_1}$
$\boldsymbol{P}_{C_1|\underline{Y}}$
$\boldsymbol{\Pi}_{\cdot\alpha}^{-1}$
$\boldsymbol{P}_{U_2|\underline{Y}U_1}$
$\boldsymbol{\Pi}_{\cdot\beta}^{-1}$
$\boldsymbol{P}_{C_2|\underline{Y}}$
f^+ operation

Figure A.2.: A q-ary SC decoder of size-2.

where the scalar a is a normalization factor.

A.2.2. q-ary SC decoding

The q-ary SC decoder follows mainly the implementation in [103, Algorithm 2-4]. Figure A.2 shows the extended message update functions for an example of size 2. We have

$$\begin{aligned}
\boldsymbol{P}_{U_1|\underline{Y}} &= f^-\left(\boldsymbol{P}_{C_1|\underline{Y}}, \boldsymbol{P}_{C_2|\underline{Y}}\right) \\
&= a_1\mathcal{H}\left(\mathcal{H}\left(\boldsymbol{P}_{C_1|\underline{Y}}\right) \odot \mathcal{H}\left(\boldsymbol{P}_{C_2|\underline{Y}}\boldsymbol{\Pi}_{\cdot\beta}^{-1}\boldsymbol{\Pi}_{\cdot\alpha}\right)\right) \\
&= a_1\mathcal{H}\left(\mathcal{H}\left(\boldsymbol{P}_{C_1|\underline{Y}}\right) \odot \mathcal{H}\left(\boldsymbol{P}_{C_2|\underline{Y}}\boldsymbol{\Pi}_{\cdot\frac{\alpha}{\beta}}\right)\right) \qquad \text{(A.15)} \\
\boldsymbol{P}_{U_2|\underline{Y}U_1} &= f^+\left(\boldsymbol{P}_{C_1|\underline{Y}}, \boldsymbol{P}_{C_2|\underline{Y}}, \hat{u}_1\right) \\
&= a_2\left(\boldsymbol{P}_{C_1|\underline{Y}}\boldsymbol{\Pi}_{+\hat{u}_1}\boldsymbol{\Pi}_{\cdot\alpha}^{-1}\right) \odot \left(\boldsymbol{P}_{C_2|\underline{Y}}\boldsymbol{\Pi}_{\cdot\beta}^{-1}\right). \qquad \text{(A.16)}
\end{aligned}$$

The decoder input is

$$\boldsymbol{P}_{C_i|\underline{Y}} = \left(P_{C_i|\underline{Y}}\left(0|\boldsymbol{y}\right), \ldots, P_{C_i|\underline{Y}}\left(q-1|\boldsymbol{y}\right)\right), \quad i \in [N_q]. \qquad \text{(A.17)}$$

At the decoding phase $i \in [N_q]$, we obtain a conditional PMF of U_i by recursively updating the messages

$$\boldsymbol{P}_{U_i|\underline{Y}U^{i-1}} = \left(P_{U_i|\underline{Y}U^{i-1}}\left(0|\boldsymbol{y}, \hat{u}^{i-1}\right), \ldots, P_{U_i|\underline{Y}U^{i-1}}\left(q-1|\boldsymbol{y}, \hat{u}^{i-1}\right)\right). \qquad \text{(A.18)}$$

The hard decision of u_i is

$$\hat{u}_i = \underset{\mu\in\mathcal{S}(u_i)}{\operatorname{argmax}}\, P_{U_i|\underline{Y}U^{i-1}}\left(\mu|\boldsymbol{y}, \hat{u}^{i-1}\right). \qquad \text{(A.19)}$$

The improved decoding algorithms in Section 3.5 can easily be extended to the q-ary case [124].

A.2.3. Kernel Selection and Code Construction

From (A.15) and (A.16), we observe that the $\boldsymbol{P}_{U_1|\underline{Y}}$ and $\boldsymbol{P}_{U_2|\underline{Y}U_1}$ depend on the ratio $\frac{\alpha}{\beta}$. We now use a MC approach to choose the best ratio $\frac{\alpha}{\beta}$:

Step 1. Set $u_1 = 0$ and select $u_2 \in \mathrm{GF}(q)$ randomly.

Step 2. q-ary encoding ($N_q = 2$): $c_1 = u_1 + u_2\alpha$ and $c_2 = u_2\beta$.

Step 3. Transmit c_1, c_2 over the channel $p_{Y|C}$.

Step 4. Compute $\boldsymbol{P}_{U_2|\underline{Y}U_1}$ via (A.15) and (A.16).

Note that $\boldsymbol{P}_{U_2|\underline{Y}U_1}$ is a random vector that depends on the channel model $p_{Y|C}$. MC simulation is used to find the optimal $\frac{\alpha}{\beta}$ that maximizes the "single-level" polarization effect, i.e., we choose

$$\frac{\alpha}{\beta} = \underset{\frac{\alpha}{\beta}\in\mathrm{GF}(q)}{\mathrm{argmax}}\,\mathbb{E}\left[\boldsymbol{P}_{U_2|\underline{Y}U_1}\left(u_2|\boldsymbol{y}, u_1\right)\right]. \tag{A.20}$$

The code construction of a non-binary kernel is to find a vector of length

$$t^{N_q} = \underset{\sum_{i=1}^{N_q} t_i = k}{\mathrm{argmax}} \prod_{i=1}^{N_q} \mathcal{R}\left(i, t_i\right) \tag{A.21}$$

where $\mathcal{R}\left(i, t_i\right)$ denotes the symbol reliability of u_i with t_i message bits, i.e.,

$$\mathcal{R}\left(i, t_i\right) = \Pr\left(\hat{U}_i = U_i \middle| \hat{U}^{i-1} = U^{i-1}, |\mathcal{S}_i| = 2^{t_i}\right),\ i \in [N_q],\quad t_i \in \{0, 1, \ldots, \log_2 q\}. \tag{A.22}$$

Obviously, we have

$$\mathcal{R}\left(i, t_i - 1\right) \geq \mathcal{R}\left(i, t_i\right),\quad t_i \in \{1, \ldots, \log_2 q\} \tag{A.23}$$

and $\mathcal{R}\left(i, 0\right) = 1$. We use the MC method to compute $\mathcal{R}\left(i, t_i\right)$ numerically. The relative reliability $\bar{\mathcal{R}}\left(i, t_i\right)$ is given by

$$\bar{\mathcal{R}}\left(i, t_i\right) = \frac{\mathcal{R}\left(i, t_i\right)}{\mathcal{R}\left(i, t_i - 1\right)},\ i \in [N_q],\ t_i \in \{1, \ldots, \log_2 q\}. \tag{A.24}$$

Algorithm A.1: Progressive construction of non-binary polar codes.

```
Input  : relative reliability $\bar{\mathcal{R}}(i, t_i)$, $i \in [N_q]$, $t_i \in [\log_2 q]$
Output: $t^{N_q}$
1 Set $t_i = 0, \ i \in [N_q]$
2 while $\sum_{i=1}^{N_q} t_i < k$ do
3     $i_{\max} = \operatorname{argmax}_i \bar{\mathcal{R}}(i, t_i + 1)$
4     $t_{i_{\max}} = t_{i_{\max}} + 1$
5 return $t^{N_q}$
```

name	size	data type	description
L	$(\log_2 N + 1) \times N$	float	LLRs
C	$(\log_2 N + 1) \times N$	binary	hard decisions
û	N	binary	output vectors

Table A.2.: Data structures for SC decoding.

We have

$$\mathcal{R}(i, t_i) = \prod_{j=0}^{t_i} \bar{\mathcal{R}}(i, j). \tag{A.25}$$

The vector t^{N_q} is chosen progressively by Algorithm A.1 to fulfill (A.21).

A.3. Pseudo Codes

The memory requirement and pseudo codes of an SC decoder are shown in Table A.2 and Algorithm A.2. The size of the 2-D arrays `L` and `C` could be reduced to $2N - 1$, see Section 3.4. Algorithm A.5 and Algorithm A.6 are the 1-based indexing versions of [103, Algorithm 3,4].

The components of the decoder are implemented for polar codes with bit-reversed order [5]. The decoder input must be permuted to bit-reversed order with Algorithm A.4 if non bit-reversed polar codes in Definition 3.1 are used (`line 1-2`).

Algorithm A.2: SCDec $\left(\ell^N\right)$

```
Input  : LLRs $\ell^N$
Output: estimates $\hat{\mathtt{u}}$
if non bit-reversed polar code then
    $\ell^N = \text{BitReverse}\left(\ell^N\right)$
for $i = 1, 2, \ldots, N$ do
    $\mathtt{L}[1, i] = \ell_i$
for $i = 1, \ldots, N$ do
    recursivelyCalcL $(\log_2 N + 1, i - 1)$
    if $i \notin \mathcal{A}$ then
        $\hat{\mathtt{u}}[i] = 0$ // compute $\hat{\mathtt{u}}[i]$ if dynamically frozen
    else
        $\hat{\mathtt{u}}[i] = \text{HardDec}(\mathtt{L}[\log_2 N + 1, i])$
    $\mathtt{C}[\log_2 N + 1, i] = \hat{\mathtt{u}}[i]$
    if $i \mod 2 = 0$ then
        recursivelyCalcC $(\log_2 N + 1, i - 1)$
return $\hat{\mathtt{u}}$
```

Algorithm A.3: HardDec (ℓ)

```
Input  : LLR $\ell$
Output: hard decision
if $\ell > 0$ then
    return 0
else
    return 1
```

Algorithm A.4: BitReverse $\left(a^N\right)$

```
Input  : vector $a^N$
Output: input vector with bit-reversed order $b^N$
if $N = 2$ then
    $b^N = a^N$
else
    $b^{N/2} = \text{BitReverse}\left(a_{\{1,3,\ldots,N-1\}}\right)$
    $b^N_{N/2+1} = \text{BitReverse}\left(a_{\{2,4,\ldots,N\}}\right)$
return $b^N$
```

Algorithm A.5: recursivelyCalcL (λ, ϕ)

```
Input : layer $\lambda$ and phase $\phi$
1  if $\lambda = 1$ then
2      return
3  $\psi = \lfloor \phi/2 \rfloor, t = 2^{\lambda-2}$
4  if $\phi \mod 2 = 0$ then
5      recursivelyCalcL $(\lambda - 1, \psi)$
6  for $\beta = 0, 1, \ldots, 2^{\log_2 N - \lambda + 1} - 1$ do
7      if $\phi \mod 2 = 0$ then
8          $\mathtt{L}[\lambda, \phi + 2\beta t + 1]$
               $= f^{-}(\mathtt{L}[\lambda - 1, \psi + 2\beta t + 1], \mathtt{L}[\lambda - 1, \psi + (2\beta + 1)t + 1])$
9      else
10         $\mathtt{L}[\lambda, \phi + 2\beta t + 1]$
               $= f^{+}(\mathtt{L}[\lambda - 1, \psi + 2\beta t + 1], \mathtt{L}[\lambda - 1, \psi + (2\beta + 1)t + 1], \mathtt{C}[\lambda, \phi + 2\beta t])$
```

Algorithm A.6: recursivelyCalcC (λ, ϕ)

```
Input : layer $\lambda$ and phase $\phi$
1  $\psi = \lfloor \phi/2 \rfloor, t = 2^{\lambda-2}$
2  for $\beta = 0, 1, \ldots, 2^{\log_2 N - \lambda + 1} - 1$ do
3      $\mathtt{C}[\lambda - 1, \psi + 2\beta t + 1] = \mathtt{C}[\lambda, \phi + 2\beta t] \oplus \mathtt{C}[\lambda, \phi + 2\beta t + 1]$
4      $\mathtt{C}[\lambda - 1, \psi + (2\beta + 1)t + 1] = \mathtt{C}[\lambda, \phi + 2\beta t + 1]$
5  if $\psi \mod 2 = 1$ then
6      recursivelyCalcC $(\lambda - 1, \psi)$
```

B

Acronyms

5G	5-th generation wireless system
APP	a-posteriori probability
ARQ	automatic repeat request
ASCL	adaptive successive cancellation list
AUB	approximated union bound
AWGN	additive white Gaussian noise
B-DMC	binary-input discrete memoryless channel
BEC	binary erasure channel
biAWGN	binary-input additive white Gaussian noise
BLER	block error rate
bpcu	bits per channel use
BSS	binary symmetric source
CC	Chase combining
CN	check node

CRC	cyclic redundancy check
CSI	channel state information
DE	density evolution
DSCF	dynamic successive cancellation flip
eMBB	enhanced mobile broadband
FHT	fast Hadamard transform
GA	Gaussian approximation
GF	Galois field
HARQ	hybrid automatic repeat request
i.i.d.	independent and identically distributed
IF	incremental freezing
IR	incremental redundancy
LDPC	low-density parity-check
LLR	log-likelihood ratio
MAP	maximum-a-posteriori
MC	Monte Carlo
MIMO	multiple-input, multiple-output
MLPC	multilevel polar coding
ML	maximum-likelihood
PAC	polarization-adjusted convolutional
PAT	pilot-assisted transmission
PCCE	polar-coded channel estimation
PCM	polar-coded modulation

PDF	probability density function
PE	polar extension
PMF	probability mass function
PM	path metric
PSK	phase-shift keying
PW	polarization weight
QPSK	quadrature phase-shift keying
QUP	quasi-uniform puncturing
RCUB	random coding union bound
RM	Reed-Muller
RQUP	reversal quasi-uniform puncturing
RS	Reed-Solomon
RV	random variable
SC	successive cancellation
SC-Fano	successive cancellation Fano
SCF	successive cancellation flip
SCL	successive cancellation list
SCOS	successive cancellation ordered search
SCS	successive cancellation sequential
SE	spectral efficiency
SNR	signal-to-noise ratio
SP	set partitioning
UB	union bound

uBLER	undetected block error rate
UPO	universal partial order
VLPE	variable-length polar extension
VN	variable node

Bibliography

[1] 3rd Generation Partnership Project (3GPP). Multiplexing and channel coding. 3GPP 38.212 V.15.3.0, 2018.

[2] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg. A low-complexity improved successive cancellation decoder for polar codes. *Asilomar Conf. Signals, Syst., Comput.*, 67(1):61–72, 2018.

[3] A. Alamdar-Yazdi and F. R. Kschischang. A simplified successive-cancellation decoder for polar codes. *IEEE Commun. Lett.*, 15(12):1378–1380, 2011.

[4] E. Arıkan. Channel combining and splitting for cutoff rate improvement. *IEEE Trans. Inf. Theory*, 52(2):628–639, 2006.

[5] E. Arıkan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inf. Theory*, 55(7):3051–3073, July 2009.

[6] E. Arıkan. From sequential decoding to channel polarization and back again. *arXiv preprint arXiv:1908.09594*, 2019.

[7] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg. LLR-based successive cancellation list decoding of polar codes. *IEEE Trans. Signal Process.*, 63(19):5165–5179, 2015.

[8] L. Barnault and D. Declercq. Fast decoding algorithm for LDPC over GF(2^q). *IEEE Inf. Theory Workshop (ITW)*, pages 70–73, Apr. 2003.

[9] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Trans. Commun.*, 44(10):1261–1271, 1996.

[10] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. *IEEE Int. Conf. Commun. (ICC)*, 2:1064–1070, 1993.

[11] E. Biglieri. *Coding for Wireless Channels.* Springer, 2005.

[12] V. Bioglio, C. Condo, and I. Land. Design of polar codes in 5G new radio. *IEEE Commun. Surveys & Tutorials*, 2020.

[13] G. Böcherer. Principles of coded modulation. Habilitation, TU München, 2018.

[14] G. Böcherer, T. Prinz, P. Yuan, and F. Steiner. Efficient polar code construction for higher-order modulation. *IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, pages 1–6, 2017.

[15] F. Brännström, L. K. Rasmussen, and A. J. Grant. Convergence analysis and optimal scheduling for multiple concatenated codes. *IEEE Trans. Inf. Theory*, 51(9):3354–3364, Aug. 2005.

[16] L. Chandesris, V. Savin, and D. Declercq. Dynamic-SCFlip decoding of polar codes. *IEEE Trans. Commun.*, 66(6):2333–2345, 2018.

[17] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel. Protograph-based raptor-like LDPC codes. *IEEE Trans. Commun.*, 63(5):1522–1532, 2015.

[18] M.-C. Chiu. Non-binary polar codes with channel symbol permutations. *Int. Symp. Inf. Theory and its Applicat. (ISITA)*, pages 433–437, Oct. 2014.

[19] M. C. Coşkun, G. Liva, J. Östman, and G. Durisi. Low-complexity joint channel estimation and list decoding of short codes. *ITG Int. Conf. Syst., Commun. and Coding*, Feb. 2019.

[20] G. Coluccia and G. Taricco. An optimum blind receiver for correlated Rician fading MIMO channels. *IEEE Commun. Lett.*, 11(9):738–739, 2007.

[21] M. C. Coşkun, J. Neu, and H. D. Pfister. Successive cancellation inactivation decoding for modified Reed-Muller and eBCH codes. *IEEE Int. Symp. Inf. Theory (ISIT)*, pages 437–442, 2020.

[22] M. C. Coşkun and H. D. Pfister. An information-theoretic perspective on successive cancellation list decoding and polar code design. *arXiv preprint arXiv:2103.16680*, 2021.

[23] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley and Sons, Inc., 2006.

[24] J. Dauwels and H. A. Loeliger. Phase estimation by message passing. *IEEE Int. Conf. Commun. (ICC)*, 1:523–527, 2004.

[25] M. C. Davey and D. MacKay. Low density parity check codes over GF(q). *IEEE Commun. Lett.*, 2(6):165–167, June 1998.

[26] N. Doan, S. A. Hashemi, F. Ercan, T. Tonnellier, and W. J. Gross. Neural dynamic successive cancellation flip decoding of polar codes. *IEEE Int. Workshop Signal Process. Syst. (SiPS)*, pages 272–277, 2019.

[27] B. Dorsch. A decoding algorithm for binary block codes and J-ary output channels (corresp.). *IEEE Trans. Inf. Theory*, 20(3):391–394, 1974.

[28] I. Dumer. Soft-decision decoding of Reed-Muller codes: a simplified algorithm. *IEEE Trans. Inf. Theory*, 52(3):954–963, Mar. 2006.

[29] G. Durisi, T. Koch, and P. Popovski. Towards massive, ultra-reliable, and low-latency wireless communications with short packets. *Proc. IEEE*, 104(9):1711–1726, Sept. 2016.

[30] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink. Decoder-tailored polar code design using the genetic algorithm. *IEEE Trans. Commun.*, 67(7):4521–4534, 2019.

[31] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink. Genetic algorithm-based polar code construction for the AWGN channel. *Int. ITG Conf. Syst. Commun. Coding (SCC)*, pages 1–6, 2019.

[32] A. Eslami and H. Pishro-Nik. A practical approach to polar codes. *IEEE Int. Symp. Inf. Theory (ISIT)*, pages 16–20, Aug. 2011.

[33] R. Fano. A heuristic discussion of probabilistic decoding. *IEEE Trans. Inf. Theory*, 9(2):64–74, 1963.

[34] G. Forney. Exponential error bounds for erasure, list, and decision feedback schemes. *IEEE Trans. Inf. Theory*, 14(2):206–220, 1968.

[35] M. Fossorier and S. Lin. Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Trans. Commun.*, 41(5):1379–1396, Sept. 1995.

[36] R. G. Gallager. Low-density parity-check codes. *IRE Trans. Inf. Theory*, 8(1):21–28, 1962.

[37] R. G. Gallager. *Low-density parity-check codes.* M.I.T. Press, Cambridge, MA, USA, 1963.

[38] R. G. Gallager. *Information Theory and Reliable Communication*, volume 2. Springer, 1968.

[39] M. J. Golay. Notes on digital coding. *Proc. IEEE*, 37:657, 1949.

[40] R. W. Hamming. Error detecting and error correcting codes. *Bell Labs Tech. J.*, 29(2):147–160, 1950.

[41] G. He, J.-C. Belfiore, I. Land, G. Yang, X. Liu, Y. Chen, R. Li, J. Wang, Y. Ge, R. Zhang, et al. Beta-expansion: A theoretical framework for fast and recursive construction of polar codes. *IEEE Global Telecommun. Conf. (GLOBECOM)*, pages 1–6, 2017.

[42] C. Herzet, N. Noels, V. Lottici, H. Wymeersch, M. Luise, M. Moeneclaey, and L. Vandendorpe. Code-aided turbo synchronization. *Proc. IEEE*, 95(6):1255–1271, 2007.

[43] C. Herzet, V. Ramon, and L. Vandendorpe. A theoretical framework for iterative synchronization based on the sum–product and the expectation-maximization algorithms. *IEEE Trans. Signal Process.*, 55(5):1644–1658, 2007.

[44] E. Hof, I. Sason, and S. Shamai. Performance bounds for erasure, list, and decision feedback schemes with linear block codes. *IEEE Trans. Inf. Theory*, 56(8):3754–3778, 2010.

[45] J. Honda and H. Yamamoto. Polar coding without alphabet extension for asymmetric models. *IEEE Trans. Inf. Theory*, 59(12):7829–7838, 2013.

[46] S.-N. Hong, D. Hui, and I. Marić. Capacity-achieving rate-compatible polar codes. *IEEE Int. Symp. Inf. Theory (ISIT)*, pages 41–45, July 2016.

[47] R. Imad, S. Houcke, and M. Ghogho. Blind estimation of the phase and carrier frequency offsets for LDPC-coded systems. *EURASIP J. Adv. Signal Process.*, 2010(1):1–13, 2010.

[48] H. Imai and S. Hirakawa. A new multilevel coding method using error-correcting codes. *IEEE Trans. Inf. Theory*, 23(3):371–377, 1977.

[49] I. Jacobs and E. Berlekamp. A lower bound to the distribution of computation for sequential decoding. *IEEE Trans. Inf. Theory*, 13(2):167–174, 1967.

[50] M.-O. Jeong and S.-N. Hong. SC-Fano decoding of polar codes. *IEEE Access*, 7:81682–81690, 2019.

[51] M. Khalighi and J. J. Boutros. Semi-blind channel estimation using the EM algorithm in iterative MIMO APP detectors. *IEEE Trans. Wireless Commun.*, 5(11):3165–3173, Nov. 2006.

[52] P. Koopman. Best CRC Polynomials. `https://users.ece.cmu.edu/~koopman/crc/`. Accessed: 2021-07-20.

[53] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory*, 47(2):498–519, 2001.

[54] A. Lancho, J. Östman, G. Durisi, T. Koch, and G. Vazquez-Vilar. Saddlepoint approximations for short-packet wireless communications. *IEEE Trans. Wireless Commun.*, 19(7):4831–4846, 2020.

[55] A. Lapidoth and P. Narayan. Reliable communication under channel uncertainty. *IEEE Trans. Inf. Theory*, 44(6):2148–2177, 1998.

[56] B. Li, H. Shen, and D. Tse. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check. *IEEE Commun. Lett.*, 16(12):2044–2047, Nov. 2012.

[57] B. Li, H. Shen, and D. Tse. A RM-polar codes. *arXiv preprint arXiv:1407.5483*, 2014.

[58] B. Li, D. Tse, K. Chen, and H. Shen. Capacity-achieving rateless polar codes. *IEEE Int. Symp. Inf. Theory (ISIT)*, pages 46–50, July 2016.

[59] B. Li, H. Zhang, and J. Gu. On pre-transformed polar codes. *arXiv preprint arXiv:1912.06359*, 2019.

[60] Y. Li, H. Zhang, R. Li, J. Wang, G. Yan, and Z. Ma. On the weight spectrum of pre-transformed polar codes. *arXiv preprint arXiv:2102.12625*, 2021.

[61] G. Liva, G. Durisi, M. Chiani, S. S. Ullah, and S. C. Liew. Short codes with mismatched channel state information: A case study. *IEEE Int. Workshop on Signal Process. Adv. in Wireless Commun.*, pages 1–5, Jul. 2017.

[62] L. Ma, J. Xiong, Y. Wei, and M. Jiang. An incremental redundancy HARQ scheme for polar code. *arXiv preprint arXiv:1708.09679*, 2017.

[63] D. J. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory*, 45(2):399–431, 1999.

[64] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang. Polar coding for bit-interleaved coded modulation. *IEEE Trans. Veh. Technol.*, 65(5):3115–3127, June 2015.

[65] A. Martinez and A. Guillén i Fàbregas. Saddlepoint approximation of random–coding bounds. *Inf. Theory Applic. Workshop (ITA)*, Feb. 2011.

[66] J. L. Massey. Coding and modulation in digital communications. *Rec. Int. Zurich Sem. Digital Commun., Zurich, Switzerland. Mar. 1974*, 1974.

[67] B. Matuz, G. Liva, E. Paolini, M. Chiani, and G. Bauch. Low-rate non-binary LDPC codes for coherent and blockwise non-coherent AWGN channels. *IEEE Trans. Commun.*, 61(10):4096–4107, 2013.

[68] N. Merhav, G. Kaplan, A. Lapidoth, and S. Shamai Shitz. On information rates for mismatched decoders. *IEEE Trans. Inf. Theory*, 40(6):1953–1967, 1994.

[69] H. Meyr, M. Moeneclaey, and S. Fechtel. *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing.* Wiley, 1997.

[70] V. Miloslavskaya. Shortened polar codes. *IEEE Trans. Inf. Theory*, 61(9):4852–4865, 2015.

[71] V. Miloslavskaya and P. Trifonov. Sequential decoding of polar codes. *IEEE Commun. Lett.*, 18(7):1127–1130, 2014.

[72] M. Mondelli, S. A. Hashemi, J. Cioffi, and A. Goldsmith. Sublinear latency for simplified successive cancellation decoding of polar codes. *IEEE Trans. Wireless Commun.*, 2020.

[73] M. Mondelli, S. H. Hassani, and R. L. Urbanke. How to achieve the capacity of asymmetric channels. *IEEE Trans. Inf. Theory*, 64(5):3371–3393, 2018.

[74] R. Mori. Properties and construction of polar codes. *arXiv preprint arXiv:1002.3521*, 2010.

[75] R. Mori and T. Tanaka. Performance of polar codes with the construction using density evolution. *IEEE Commun. Lett.*, 13(7):519–521, July 2009.

[76] R. Mori and T. Tanaka. Non-binary polar codes using reed-solomon codes and algebraic geometry codes. *IEEE Inf. Theory Workshop (ITW)*, pages 1–5, Sept. 2010.

[77] D. E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Trans. IRE Professional Group Electron. Comput.*, EC-3(3):6–12, 1954.

[78] K. Niu, K. Chen, and J.-R. Lin. Beyond turbo codes: Rate-compatible punctured polar codes. *IEEE Int. Conf. Commun. (ICC)*, pages 3423–3427, June 2013.

[79] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe. Turbo synchronization: an EM algorithm interpretation. *IEEE Int. Conf. Commun. (ICC)*, 4:2933–2937, 2003.

[80] J. Östman, G. Durisi, E. G. Ström, M. C. Coşkun, and G. Liva. Short packets over block-memoryless fading channels: Pilot-assisted or noncoherent transmission? *IEEE Trans. Commun.*, 67(2):1521–1536, Feb. 2019.

[81] W. Parry. On the β-expansions of real numbers. *Acta Mathematica Academiae Scientiarum Hungarica*, 11:401–416, 1960.

[82] M. Plotkin. Binary codes with specified minimum distance. *IRE Trans. Inf. Theory*, 6(4):445–450, 1960.

[83] Y. Polyanskiy, H. V. Poor, and S. Verdú. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory*, 56(5):2307–2359, May 2010.

[84] I. Reed. A class of multiple-error-correcting codes and the decoding scheme. *Trans. IRE Professional Group Inf. Theory*, 4(4):38–49, 1954.

[85] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *J. SIAM*, 8(2):300–304, 1960.

[86] A. Rényi. Representations for real numbers and their ergodic properties. *Acta Mathematica Academiae Scientiarum Hungarica*, 8:477–493, 1957.

[87] T. Richardson and R. Urbanke. *Modern Coding Theory.* Cambridge university press, 2008.

[88] D. Rife and R. Boorstyn. Single tone parameter estimation from discrete-time observations. *IEEE Trans. Inf. Theory*, 20(5):591–598, 1974.

[89] Rong-Rong Chen, R. Kötter, U. Madhow, and D. Agrawal. Joint noncoherent demodulation and decoding for the block fading channel: a practical framework for approaching Shannon capacity. *IEEE Trans. Commun.*, 51(10):1676–1689, 2003.

[90] M. Rowshan and E. Viterbo. How to modify polar codes for list decoding. *IEEE Int. Symp. Inf. Theory (ISIT)*, 2019.

[91] W. Ryan and S. Lin. *Channel Codes: Classical and Modern.* Cambridge university press, 2009.

[92] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross. Fast polar decoders: Algorithm and implementation. *IEEE J. Sel. Areas Commun.*, 32(5):946–957, 2014.

[93] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross. Fast list decoders for polar codes. *IEEE J. Sel. Areas Commun.*, 34(2):318–328, 2015.

[94] G. Sarkis and W. J. Gross. Increasing the throughput of polar decoders. *IEEE Commun. Lett.*, 17(4):725–728, 2013.

[95] E. Şaşoğlu, E. Telatar, and E. Arıkan. Polarization for arbitrary discrete memoryless channels. *IEEE Inf. Theory Workshop (ITW)*, pages 144–148, Oct. 2009.

[96] J. Scarlett, A. Martinez, and A. G. i. Fabregas. Mismatched decoding: Error exponents, second-order rates and saddlepoint approximations. *IEEE Trans. Inf. Theory*, 60(5):2647–2666, 2014.

[97] C. Schürch. A partial order for the synthesized channels of a polar code. *IEEE Int. Symp. Inf. Theory (ISIT)*, pages 220–224, July 2016.

[98] M. Seidl, A. Schenk, C. Stierstorfer, and J. B. Huber. Polar-coded modulation. *IEEE Trans. Commun.*, 61(10):4108–4119, Sept. 2013.

[99] C. E. Shannon. A mathematical theory of communication. *Bell Labs Tech. J.*, 27(3):379–423, 1948.

[100] D. A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Inf. Theory*, 42(6):1723–1731, 1996.

[101] N. Stolte. *Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung*. PhD thesis, TU Darmstadt, 2002.

[102] I. Tal and A. Vardy. How to construct polar codes. *IEEE Trans. Inf. Theory*, 59(10):6562–6582, 2013.

[103] I. Tal and A. Vardy. List decoding of polar codes. *IEEE Trans. Inf. Theory*, 61(5):2213–2226, 2015.

[104] R. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory*, 27(5):533–547, 1981.

[105] G. Taricco and E. Biglieri. Space-time decoding with imperfect channel estimation. *IEEE Trans. Wireless Commun.*, 4(4):1874–1888, 2005.

[106] G. Taricco and G. Coluccia. Optimum receiver design for correlated Rician fading MIMO channels with pilot-aided detection. *IEEE J. Sel. Areas Commun.*, 25(7):1311–1321, 2007.

[107] S. R. Tavildar. Bit-permuted coded modulation for polar codes. *IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017.

[108] S. ten Brink, G. Kramer, and A. Ashikhmin. Design of low-density parity-check codes for modulation and detection. *IEEE Trans. Commun.*, 52(4):670–678, May 2004.

[109] L. Tong, B. M. Sadler, and M. Dong. Pilot-assisted wireless transmissions: General model, design criteria, and signal processing. *IEEE Signal Process. Mag.*, 21(6):12–25, Nov. 2004.

[110] P. Trifonov. A score function for sequential decoding of polar codes. *IEEE Int. Symp. Inf. Theory (ISIT)*, pages 1470–1474, 2018.

[111] P. Trifonov and V. Miloslavskaya. Polar codes with dynamic frozen symbols and their decoding by directed search. *IEEE Inf. Theory Workshop (ITW)*, pages 1–5, 2013.

[112] P. Trifonov and V. Miloslavskaya. Polar subcodes. *IEEE J. Sel. Areas Commun.*, 34(2):254–266, Feb. 2016.

[113] U. Wachsmann, R. F. Fischer, and J. B. Huber. Multilevel codes: Theoretical concepts and practical design rules. *IEEE Trans. Inf. Theory*, 45(5):1361–1391, 1999.

[114] R. Wang and R. Liu. A novel puncturing scheme for polar codes. *IEEE Commun. Lett.*, 18(12):2081–2084, Oct. 2014.

[115] T. Wang, D. Qu, and T. Jiang. Parity-check-concatenated polar codes. *IEEE Commun. Lett.*, 20(12):2342–2345, 2016.

[116] D. Warrier and U. Madhow. Spectrally efficient noncoherent communication. *IEEE Trans. Inf. Theory*, 48(3):651–668, 2002.

[117] T. Wiegart, F. Steiner, P. Schulte, and P. Yuan. Shaped on–off keying using polar codes. *IEEE Commun. Lett.*, 23(11):1922–1926, 2019.

[118] Y. Wu and C. N. Hadjicostis. Soft-decision decoding using ordered recodings on the most reliable basis. *IEEE Trans. Inf. Theory*, 53(2):829–836, 2007.

[119] H. Wymeersch. *Iterative Receiver Design.* Cambridge, 2007.

[120] M. Xhemrishi, M. C. Coşkun, G. Liva, J. Östman, and G. Durisi. List decoding of short codes for communication over unknown fading channels. *Asilomar Conf. Signals, Systems, Computers*, pages 810–814, 2019.

[121] P. Yuan and M. C. Coşkun. Complexity-adaptive maximum-likelihood decoding of modified $\boldsymbol{G}_N$-coset codes. *IEEE Inf. Theory Workshop (ITW)*, pages 1–6, 2021.

[122] P. Yuan, M. C. Coşkun, and G. Kramer. Polar-coded non-coherent communication. *IEEE Commun. Lett.*, 25(6):1786–1790, 2021.

[123] P. Yuan, T. Prinz, G. Böcherer, O. İşcan, R. Böhnke, and W. Xu. Polar code construction for list decoding. *Int. ITG Conf. Syst. Commun. Coding (SCC)*, pages 1–6, 2019.

[124] P. Yuan and F. Steiner. Construction and decoding algorithms for polar codes based on 2×2 non-binary kernels. *IEEE Int. Symp. Turbo Codes & Iterative Inf. Process. (ISTC)*, pages 1–5, 2018.

[125] P. Yuan, F. Steiner, T. Prinz, and G. Böcherer. Flexible IR-HARQ scheme for polar-coded modulation. *IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, pages 49–54, 2018.