Throughput-based Design for Polar Coded-Modulation

Hossein Khoshnevis, *Student Member, IEEE,* Ian Marsland, *Member, IEEE,*
and Halim Yanikomeroglu, *Fellow, IEEE*

***Abstract*—Typically, forward error correction (FEC) codes are designed based on the minimization of the error rate for a given code rate. However, for applications that incorporate hybrid automatic repeat request (HARQ) protocol and adaptive modulation and coding, the throughput is a more important performance metric than the error rate. Polar codes, a new class of FEC codes with simple rate matching, can be optimized efficiently for maximization of the throughput. In this paper, we aim to design HARQ schemes using multilevel polar coded-modulation (MLPCM). Thus, we first develop a method to determine a set-partitioning based bit-to-symbol mapping for high order QAM constellations. We simplify the LLR estimation of set-partitioned QAM constellations for a multistage decoder, and we introduce a set of algorithms to design throughput-maximizing MLPCM for the successive cancellation decoding (SCD). These codes are specifically useful for non-combining (NC) and Chase-combining (CC) HARQ protocols. Furthermore, since optimized codes for SCD are not optimal for SC list decoders (SCLD), we propose a rate matching algorithm to find the best rate for SCLD while using the polar codes optimized for SCD. The resulting codes provide throughput close to the capacity with low decoding complexity when used with NC or CC HARQ.**

***Index Terms*—Throughput-based polar code design, HARQ, Chase-combining, multilevel coding, multistage decoding, low complexity LLR estimation.**

I. Introduction

The time-varying nature of wireless channels requires the use of adaptive modulation and coding (AMC) schemes to achieve high throughput communication. Hybrid automatic repeat request (HARQ) error control protocols can enhance the throughput of AMC especially when high order modulation schemes are used. Typically, the design objective of binary error correction codes for the additive white Gaussian noise (AWGN) channel is to minimize the error rate, for a given code rate and signal-to-noise ratio (SNR). This method of design has been widely used for convolutional codes [1], parallel concatenated (turbo) codes [2], low density parity check codes (LDPC) [3] and polar codes [4], [5]. However, to optimize the AMC and HARQ protocols, the throughput is a much more relevant performance metric than error rate, where throughput is defined as the average rate of successful message delivery and indicates how close the performance of a system is to the channel capacity. Designing throughput-optimal codes and coded-modulation typically involves an exhaustive search over a set of code rates and modulations and employs simulation to estimate the throughput [6]. However, as demonstrated in this paper, this process can be greatly simplified for polar codes.

Polar codes, introduced by Erdal Arikan in [4], are a new class of forward error correction (FEC) codes that can achieve the capacity of binary-input symmetric-output memoryless channels under successive cancellation decoding (SCD) as their length tends to infinity. However, finite length polar codes can be designed and decoded efficiently to perform close to the channel capacity. Polar codes work based on the concept of channel polarization which simplifies the rate matching. Therefore, they are a strong candidate for designing throughput-maximizing FEC codes. When designing polar codes, a set of information bit-channels must be chosen. To find this information-set, the distribution of the log-likelihood ratios (LLRs) of bit-channels should be determined using simulation [4] or density evolution. Trifonov in [7] showed that the density evolution based design of polar codes in an AWGN channel can be simplified using the Gaussian approximation (GA) of the distribution of LLRs.

When designing polar codes based on the throughput, the optimal set of information bits to maximize the throughput is chosen. The particular advantage of polar codes that facilitates their optimization for maximizing the throughput is this straightforward design method in comparison to most other modern codes. These polar codes are particularly useful for non-combining (NC)[1] and Chase-combining (CC) HARQ where, for the retransmission of a failed codeword, the whole codeword should be retransmitted. Furthermore, they can be used for incremental redundancy HARQ (IR HARQ) schemes. Although efficient IR HARQ schemes using polar codes have been proposed (e.g., [8]) their scheduling, especially for coded-modulation schemes, is difficult [9] since transmitting short length redundancy may need an overwhelming amount of control signals. To achieve simple scheduling, we set the retransmission redundancy sizes to be equal to the mother code length. In this case, limited feedback can be employed to achieve high throughput using AMC and HARQ protocols since the polar code graph is fixed and only the information-set should be modified when the SNR changes. Polar codes have also been designed to maximize the throughput of CC HARQ with SCD in [10] and with SCLD in [9] based on puncturing for a fixed message length and a BPSK constellation. These methods, given the message length K, require a full search over all puncturing lengths from 0 to $N-K$ and thus require

H. Khoshnevis, I. Marsland, and H. Yanikomeroglu are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada (e-mail: {khoshnevis, ianm, halim}@sce.carleton.ca). This work is supported by Huawei Canada Co., Ltd. This paper was presented in part in *IEEE VTC-Fall* 2017.

[1] Non-combining HARQ is also known as Type-I HARQ.

running the GA method for $N - K$ times. However, in our proposed code design method, we only need to run the GA method once for most protocols. Therefore, their design is at least $N - K$ times more complex than our method and, as we will show, there is no advantage in using their algorithms.

To construct polar coded-modulation, either multilevel coding with multistage decoding (MLC/ MSD) or bit-interleaved-coded-modulation (BICM), two well-known methods of concatenating binary codes with high order modulations, can be employed, but MLC/MSD with a set-partitioning based bit-to-symbol mapping (SPM), due to the clarity of design and the conceptual similarity to channel polarization, outperforms BICM [11]. In addition, the LLR estimation for MSD is cheaper than for BICM which make it a promising technique for future wireless communication systems. Similar to BPSK modulation, designing multilevel polar coded-modulation (MLPCM) using the puncturing-based search methods in [9] and [10] is hard since they require a full search over all levels of a MLC. To design MLPCM, Trifonov in [7] suggested using the GA, but the full steps of design have not been presented in the literature.

In this paper, first the accurate LLR estimation of quadrature amplitude modulation (QAM) with two dimensional (2D) SPM is simplified using a novel device constructed from two independent pulse amplitude modulation (PAM) constellations for the I-channel and Q-channel, and a linear bit mapping. Then, an approximation is proposed to simplify the LLR estimation of PAM constellations. Using the LLR simplifications, the average LLR of each binary channel of QAM constellations is measured and a set of algorithms for designing MLPCM using GA is proposed that is based on maximizing the throughput, instead of the well-studied objective of minimizing the FER. In these algorithms, we fix the code-length and find the optimal information-set using bounds on the throughput of the coded-modulation schemes. These codes are designed for NC, CC, and IR HARQ schemes with successive cancellation decoders (SCD). In addition, since polar codes optimized for SCD are suboptimal for SC list decoding (SCLD), introduced in [12], a fast rate matching algorithm is proposed to find the code rate corresponding to the maximum throughput for SCLD when used with polar codes designed for SCD.

In particular, first we show that by adapting the MLPCM to SNR, throughput very close to the capacity can be achieved. In this case, CC HARQ does not provide any advantage over NC HARQ. However, when the transmitter is restricted to use a smaller number of codes, CC HARQ outperforms NC HARQ and also provides higher throughput in comparison to BICM-based CC and IR HARQ, constructed in [13]. We also show that when the levels of MLPCM can be decoded independently using HARQ, the throughput is enhanced substantially.

The rest of the paper is organized as follows: The system model and HARQ protocols are described in Section II, and the simplified LLR approximation method for QAM and PAM constellations with SPM along with a simple method of constructing set-partitioning for QAM constellations is described in Section III. The throughput of HARQ protocols as a design metric is explained in Section IV. The polar code design methods based on the throughput for a BPSK constellation with SCD are introduced in Section V, and the MLPCM design procedure for QAM constellations is described in Section VI. In addition, a rate matching algorithm for SCLD is proposed in Section VII. Finally, numerical results and discussions are provided in Section VIII, and conclusions are presented in Section IX.

II. System Model

The communication system includes a single user transmitter and receiver that use a NC or CC HARQ error control protocol. At the transmitter, a cyclic redundancy check (CRC) sequence of length L_{CRC} is added to data of length K bits and each $K' = K + L_{\text{CRC}}$ bits of the CRC and data are coded using a multilevel polar code of total length N_{tot} and a total code rate of $R = K'/N_{\text{tot}}$. The MLC scheme employs a set of B independent encoders, each with a code length $N = N_{\text{tot}}/B$ and code rate R_n, with one encoder for each binary channel corresponding to an address bit of a constellation with a cardinality of 2^B. After encoding of all levels, each set of code-bits $\{c_1^i, c_2^i, ..., c_B^i\}$, for $i = 1, ..., N$, are modulated using BPSK or QAM and transmitted through an AWGN channel with a noise variance of $\frac{N_0}{2}$ per dimension. Throughout this paper, due to the power inefficiency of rectangular QAM when B is an odd number, we limit the discussions to even values of B. The system model can be written as

$$y_{i,l} = x_{i,l} + w_{i,l}, \tag{1}$$

where $y_{i,l}$ is the i^{th} received sample in the l^{th} retransmission and $x_{i,l}$ and $w_{i,l}$ represent the transmitted symbol and the noise, respectively. The system has an average SNR of $\gamma = E_s/N_0$ where $E_s = E[|x_{i,l}|^2]$.

In a MSD, the LLR for each level is calculated given the code-bits for the upper-levels [14]. To this end, after decoding of each level, the received message word is re-encoded and is fed to the demapper to reduce the LLR estimation ambiguity of the next level since the knowledge about the upper-level bits limits the cardinality of lower-level symbol-sets for the LLR calculation. The LLR calculation at each level can be written as

$$\lambda_{n,i,l} = \ln \frac{\sum_{x \in \mathcal{X}_n^0} p(y_{i,l} \mid x_{i,l} = x)}{\sum_{x \in \mathcal{X}_n^1} p(y_{i,l} \mid x_{i,l} = x)}, \tag{2}$$

where $n = 1, 2, ..., B$, $\mathcal{X}_n^0$ and $\mathcal{X}_n^1$ are the set of all constellation points with zero or one in their n^{th} position given the upper-level code-bits, respectively, and $p(y_{i,l} \mid x_{i,l} = x)$ is the likelihood function of $y_{i,l}$ being received given x is transmitted.

A. HARQ Protocols

In this paper, we design MLPCM for four HARQ protocols. The protocols are divided into two groups, level-dependent and level-independent, based on the dependency of levels for decoding.

Level-Dependent HARQ Protocols

In level-dependent protocols, all levels of a multilevel codeword are decoded and an ACK is fed back to the transmitter only if all levels are correct. Indeed, all levels of the multilevel code are dependent and all levels are seen as one codeword. Thus, only one CRC for checking the correctness of the codeword is employed.

The LLRs used for each decoding attempt depend on whether NC or CC is used. For NC level-dependent (NC-D) HARQ, the LLRs depend only on the received samples for the latest retransmission of the codeword, so the LLRs are given by (2). For CC level-dependent (CC-D) HARQ, the LLRs depend on the received samples from all retransmissions, according to

$$\lambda_{n,i,L} = \ln \frac{\sum_{x \in \mathcal{X}_n^0} \prod_{l=1}^{L} p(y_{i,l} \mid x_{i,l} = x)}{\sum_{x \in \mathcal{X}_n^1} \prod_{l=1}^{L} p(y_{i,l} \mid x_{i,l} = x)}, \tag{3}$$

where L is the number of retransmissions.

Level-Independent HARQ Protocols

In level-independent protocols, as proposed in [15], independent codewords with their own CRC are transmitted on each level of MLC. For each new codeword, the receiver decodes the codeword of each level independently and checks whether it is a valid codeword. When a codeword of a specific level is invalid, the same codeword is retransmitted during the next transmission on the same level while on all other upper and lower levels, new codewords containing new messages are transmitted. In this protocol, the receiver waits until a codeword of a level is decoded correctly before attempting to decode the next levels.

An example of the protocol is shown in Fig. 1. In this example, codewords A_1, and B_1, are transmitted using the first and second levels of a two-level MLC, and codeword A_1 is decoded incorrectly. No attempt is made to decode B_1, (since its LLRs cannot be calculated without reliable knowledge of A_1), so a NACK is sent to the transmitter informing that A_1 failed. The transmitter responds by retransmitting A_1 on level 1, and transmitting a new level-2 codeword, B_2, on level 2. In this example the receiver successfully decodes A_1 after this transmission, so now it can attempt to decode B_1 (using the received samples from the previous transmission) and B_2 (using the current received samples). If B_1 fails, the transmitter is instructed to send A_2 and B_1. If B_2 fails, the transmitter sends A_2 and B_2, and if B_1 and B_2 succeed, A_2 and B_3 are transmitted (as shown in Fig. 1).

The throughput of level-independent protocols is expected to be more than the level-dependent protocols, because in level-dependent protocols all levels with the same message words are retransmitted while in level-independent protocols, only erroneous upper-levels are retransmitted and lower-levels are used to transmit new codewords.

For NC level-independent (NC-I) HARQ the LLRs are calculated according to (2), the same as for NC-D. For CC level-independent (CC-I) HARQ, the combining scheme is a little different than that of CC-D, because different codewords may be transmitted on each level during each transmission, so the LLRs are given by

$$\lambda_{n,i,l} = \ln \frac{\sum_{x \in \mathcal{X}_n^0} p(y_{i,l} \mid x_{i,l} = x)}{\sum_{x \in \mathcal{X}_n^1} p(y_{i,l} \mid x_{i,l} = x)} + \lambda_{n,i,l-1}. \tag{4}$$

For all protocols, the system uses AMC employing modulations and codes of different spectral efficiencies and rates for each SNR. The AMC is applied through the channel state information based mode-selection at the transmitter to enhance the throughput.

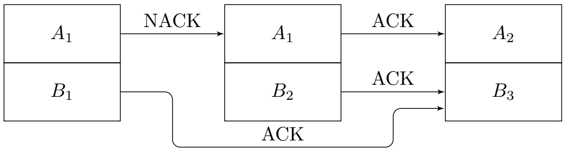


Figure 1: Illustrative example of the use of level-independent HARQ.

III. Low Complexity LLR Estimation for MLC/MSD

Typically, at the decoder of a coded-modulation scheme, calculation of code bit LLRs using (2) is much more complex than actually decoding the binary FEC codes. Thus, reducing the LLR calculation complexity can substantially simplify the system. In this section, we simplify the LLR calculation for QAM and PAM constellations with SPM. The results of this section are used throughout the paper for both decoding and the code construction. For notational simplicity we drop the subscript i and l in the following, as we are only discussing the detection of a single symbol corresponding to one channel-use.

A. *LLR Calculation for Set-partitioned QAM*

The LLR calculation for QAM with a two dimensional (2D) SPM needs a relationship between the LLR value and the real and imaginary parts of the received sample. This results in a function with two input variables and high arithmetic complexity. While this function can be divided into several regions and piecewise approximations with planes can be employed to simplify the LLR calculation, the number of regions grows fast for large size QAM constellations. As an alternative, a 2D SPM can be decomposed into two independent 1D SPMs. While decomposition of 2D to 1D mappings is well-known and straightforward for Gray mapping, in the following we proposed a new technique that is suitable for SPM.

We want to map B bits, $[c_1 c_2 ... c_B]$ onto a 2^B-point square QAM constellation with SPM, such that bit c_1 has the lowest reliability, followed by c_2, and so on up to c_B which is the most reliable. This can be achieved using the simple device shown in Fig. 2, that involves a simple code and two PAM symbol mappers with natural mapping. By using this device at the transmitter, the LLR calculator at the receiver is greatly simplified. Using this device, we first precode the bits, giving $b = [b_1, b_2, ..., b_B]$ where

$$b_k = \begin{cases} c_k \oplus c_{k+1} & \forall \,\text{odd}\, k, \\ c_k & \forall \,\text{even}\, k. \end{cases} \tag{5}$$

Then, we send the bits in odd-indexed positions to a $2^{B/2}$-PAM symbol mapper for transmission over the I-channel, and the other bits to another $2^{B/2}$-PAM symbol mapper for the Q-channel. From Fig. 2, $b_k^I = b_{2k-1} = c_{2k-1} \oplus c_{2k}$ and $b_k^Q = b_{2k} = c_{2k}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2018.2880204, IEEE Transactions on Communications

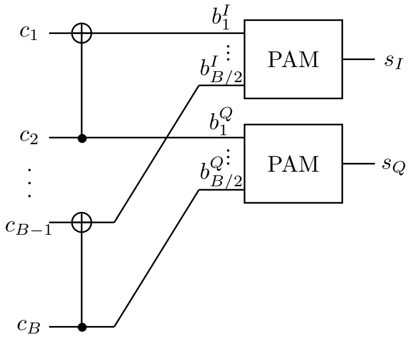


Figure 2: A transform to map natural numbers to 16-QAM with 2D SPM using two independent 4-PAMs with SPM.

The PAM symbol mappers are identical and employ natural mapping between the input bits $\mathbf{d} = [d_{B/2}...d_2 d_1]$ (where $d_k = b_k^I$ or b_k^Q for the I-channel and Q-channel mappers, respectively), and the points

$$\mathrm{SM}_{\mathrm{PAM}}[\mathbf{d}] = 2d - (2^{B/2} - 1), \tag{6}$$

where $d = d_{B/2}2^{B/2-1} + ... + d_2 2^1 + d_1 2^0$ is the integer representation of $\mathbf{d}$. Equivalently, we can write

$$\mathrm{SM}_{\mathrm{PAM}}[\mathbf{d}] = \sum_{k=1}^{B/2}(2d_k - 1)2^{k-1}. \tag{7}$$

Note that with this mapping, the least significant bit, d_1, is also the least reliable, and the most significant bit, $d_{B/2}$, is the most reliable. The transmitted QAM symbol is then

$$\begin{aligned}\mathrm{SM}_{\mathrm{QAM}}[\mathbf{c}] &= \mathrm{SM}_{\mathrm{PAM}}[\mathbf{b}^I] + j\mathrm{SM}_{\mathrm{PAM}}[\mathbf{b}^Q] \\ &= \sum_{k=1}^{B/2}(2b_k^I - 1)2^{k-1} + j\sum_{k=1}^{B/2}(2b_k^Q - 1)2^{k-1} \\ &= \sum_{k=1}^{B/2}(2[c_{2k-1} \oplus c_{2k}] - 1)2^{k-1} + j\sum_{k=1}^{B/2}(2c_k - 1)2^{k-1},\end{aligned} \tag{8}$$

where $j = \sqrt{-1}$.

Theorem 1. *The resulting constellation in (8) uses SPM.*

Proof: Observe that

$$\begin{aligned}&\mathrm{SM}_{2^B\text{-QAM}}[\mathbf{c}] = \\ &\sum_{k=1}^{B/2-1}(2[c_{2k-1} \oplus c_{2k}] - 1)2^{k-1} + j\sum_{k=1}^{B/2-1}(2c_k - 1)2^{k-1} \\ &+ (2[c_{B-1} \oplus c_B] - 1)2^{B/2-1} + j(2c_B - 1)2^{B/2-1} \\ &= \mathrm{SM}_{2^{B-2}\text{-QAM}}[\mathbf{c}_{1:B-2}] + \mathrm{SM}_{4\text{-QAM}}[\mathbf{c}_{B-1:B}]2^{B/2-1},\end{aligned} \tag{9}$$

so that the 2^B-QAM constellation can be constructed by enlarging a 2^{B-2}-QAM constellation by a factor of 4. Under the assumption that the 2^{B-2}-QAM constellation has SPM, then the enlarged constellation does as well. Each point $x = \mathrm{SM}_{2^{B-2}-QAM}[\mathbf{c}_{1:B-2}]$ in each subset of the smaller constellation is replaced by four points in the larger constellation, $x + (-1 - j)2^{B/2-1}$, $x + (1 + j)2^{B/2-1}$, $x + (1 - j)2^{B/2-1}$, $x + (-1 + j)2^{B/2-1}$, with the same bit labellings as x (i.e., $\mathbf{c}_{1:B-2}$), but with 00, 01, 10 or 11 appended, respectively. Since all the points in each set-partitioned subset of the smaller constellation fall on some regular lattice, and the new points in the enlarged subset fall on the same lattice, the minimum distance between points in the subset remains the same. By adding two more layers of the set-partitioning, creating subsets with minimum distance of $2^{B/2}$ and $\sqrt{2} \times 2^{B/2}$, the resulting 2^B-QAM constellation has SPM, provided the 2^{B-2}-QAM constellation has SPM. But since by inspection, the 4-QAM constellation given by (8) with $B = 2$ has SPM, it follows by induction that the 2^B-QAM constellation has SPM for all even values of B. ■

Note that the constellation with SPM in (8) is regular since all subsets at a specific level n have the same average Euclidean distance spectrum and consecutively the same capacity [16]. Therefore, one binary code can be constructed for all subsets within each level.

At the receiver, the LLRs of the code bits, $\mathbf{c}$, are readily computed from the received sample. The real and imaginary parts of each received sample are sent to two different $2^{B/2}$-PAM LLR calculators, which can use (2) to calculate the LLRs corresponding to $\mathbf{b}^I$ and $\mathbf{b}^Q$, respectively. The LLRs of the first PAM bit-channels are calculated, giving λ_1^I and λ_1^Q, which are combined to give the LLR of c_1,

$$\lambda_1^c = \lambda_1^I \boxplus \lambda_1^Q, \tag{10}$$

where $\boxplus$ is the boxplus operator defined as $\lambda_1 \boxplus \lambda_2 = 2\tanh^{-1}\big(\tanh(\frac{\lambda_1}{2})\tanh(\frac{\lambda_2}{2})\big)$. Once the upper code (of which c_1 is a code-bit) has been decoded, and an estimate $\hat{c}_1$ of c_1 based on the code has been generated, the LLR of c_2 can be calculated as

$$\lambda_2^c = (1 - 2\hat{c}_1)\lambda_1^I + \lambda_1^Q, \tag{11}$$

and the second code can be decoded, giving $\hat{c}_2$. Armed with $\hat{c}_1$ and $\hat{c}_2$, the LLR calculators can detect the second PAM bit-channels, giving λ_2^I and λ_2^Q. These in turn are used to calculate

$$\begin{aligned}\lambda_3^c &= \lambda_2^I \boxplus \lambda_2^Q, \\ \lambda_4^c &= (1 - 2\hat{c}_3)\lambda_2^I + \lambda_2^Q.\end{aligned} \tag{12}$$

This process is repeated until all levels have been decoded.

B. LLR Calculation for Set-partitioned PAM

In last section, we showed how to separate a QAM constellation with 2D SPM into two independent PAM constellations, which by itself significantly reduces the complexity of LLR calculations, but the complexity is still needlessly high. Although the LLR calculation for BPSK is easy, computed as $\lambda = -\frac{4y}{N_0}$ for the AWGN channel, for PAM constellations direct application of (2) is of high complexity even for moderately sized constellations. In this section, we try to simplify the LLR estimation of PAM constellations.

The LLR calculation for a MSD based on the Jacobi theta functions is proposed in [17], in which knowledge of the values of Jacobi theta functions is required for the LLR calculation. However, saving and referring to values of these functions needs large memory capacity. Instead, the LLR of a MSD can be approximated by the dominant term since $\ln(\sum_t e^{-|h_t|}) \approx -\min_t(|h_t|)$. Therefore, the LLR of a MSD in (2) can be approximated by

$$\lambda_n \approx \frac{1}{N_0}[-\min_{x\in\mathcal{X}_n^0}|y-x|^2 + \min_{x\in\mathcal{X}_n^1}|y-x|^2]. \qquad (13)$$

This approximation, known as the max-log approximation (MLA) has been used for LLR estimation of Gray-mapped QAM [18]. In [19], three methods, namely a MLA in (13), a log-separation algorithm (LSA) and a mixed algorithm, are proposed to reduce the complexity of the LLR calculation of a MSD for phase shift keying (PSK) and amplitude-phase-shift keying (APSK) constellations. In [19], it is shown that LSA is slightly better for low SNRs and for moderate-to-high SNRs the MLA is essentially the same as the exact calculation using (2). Further analysis of the MLA can result in substantial simplification of the LLR calculation. Here, by analyzing the MLA and employing the piecewise linear approximation of the LLRs, we reduce the LLR calculation complexity for PAM constellation with SPM.

Proposition 1. *Let $\mathcal{X}_n$ be a $M_n = 2^{B-n+1}$-point subset of a 2^B-PAM constellation given that the $n-1$ upper-level (least significant) bits are known, and let x_d be the $(d+1)^{th}$ largest element of $\mathcal{X}_n$. For $d \in 0,1,...,M_n-2$ and $\mathbb{L}$ a large number, let $\Omega_d = \{y \in \mathbb{R} \mid -\mathbb{L}\delta_d + x_d < y \le x_{d+1} + \mathbb{L}\delta_{d-M_n+2}\}$, be the interval $(x_d, x_{d+1}]$ except the first interval extends down to $-\infty$ and the last up to $+\infty$. Then, for $y \in \Omega_d$*

$$\lambda_n(y) \approx \frac{1}{N_0}(-1)^d(x_{d+1}-x_d)(x_{d+1}+x_d-2y). \qquad (14)$$

Proof: Since $\mathcal{X}_n$ is just a M_n-PAM constellation with natural mapping that has been scaled by 2^{n-1} and shifted by an amount that depends on the known upper level bits, the label of the least significant bit changes with each consecutive point. Therefore, for $y \in \Omega_d$ the pair $(\min_{x\in\mathcal{X}_n^0}|y-x|^2, \min_{x\in\mathcal{X}_n^1}|y-x|^2)$ is either (x_d, x_{d+1}) or (x_{d+1}, x_d) with the first applying when d is even and the second when d is odd. Thus (13) can be written as $\frac{1}{N_0}(-1)^d[-|y-x_d|^2 + |y-x_{d+1}|^2]$, which reduces to (14). ■

The LLR as a function of the received symbol y, approximated by (14) is compared with the exact LLR computed by (2) as shown in Fig. 3, for the first level of 4-PAM, 8-PAM and BPSK with $\gamma = 6$ dB. This approximation is quite accurate at SNRs of interest but require much less effort to evaluate than the exact expression. Even thought this approximation is less accurate at low SNRs, we were unable to find a case where the use of the simplified LLRs had any discernible impact on the throughput of the codes designed in this paper.

The proposed LLR approximation for QAM with 2D SPM consists of the max-log-based LLR approximation for two PAMs with SPM and a linear device. This can be computed using $9B-8$ real additions, $5B-6$ real multiplications, B real divisions, $2B$ exponential operations, and B logarithm operations. In contrast, if we estimate the LLR directly using (2), we need $8 \times 2^B - 7 - 2B$ real additions, $6 \times 2^B - 7$ real multiplications, $B-1$ real divisions, $2 \times 2^B - 4$ exponential operations, and $B-1$ logarithm operations. Note that the complexity of the simplified LLR approximation grows with B while the exact estimation grows with 2^B. The proposed LLR simplification method is less complex when $B \ge 2$.

IV. Throughput as a Design Metric

Typically, for the AWGN channel, polar codes are designed to minimize the FER for a given code rate, R, at a given SNR. That is, the K elements of the information-set are chosen in an attempt to provide as low a FER as possible. Alternatively, for a given SNR and a target FER, one can choose the information-set to be as large as possible (thereby maximizing the code rate), while ensuring the target FER is not exceeded. However, for systems employing HARQ, neither the FER nor the code rate is of primary importance. For ARQ systems, messages are transmitted indefinitely until they are correctly received. As such, the throughput, which is the rate that message bits are correctly received (in message bits per channel use), is a more relevant metric. By using this new criterion, codes that are quite different from those that maximize the code rate or minimize the FER can be designed.

For the NC-D protocol, where a single code is used across all binary channels of the constellation and received samples from previous transmission are not exploited, the throughput is given by

$$\eta_{\text{NC-D}} = \frac{K}{NB}(1-P_K), \qquad (15)$$

where P_K is the FER when the message word length is K. The optimization problem, for a given codeword length, $N_{\text{tot}} = NB$, is to find K, and the associated information-set of the polar code, that maximize $\eta_{\text{NC-D}}$. When the NC-I protocol is used, the throughput, since the levels are independent, is

$$\eta_{\text{NC-I}} = \sum_{n=1}^{B}\frac{K_n}{N}(1-P_n^{K_n}), \qquad (16)$$

where K_n is the length of the information-set of the codeword transmitted on the n^{th} binary channel of the constellation (with $K = \sum_n K_n$ still being the total number of transmitted message bits.) and $P_n^{K_n}$ is the corresponding FER of that code.

When Chase-combining is used, the FER decreases depending on the number of retransmissions, so the throughput of the CC-D protocol is

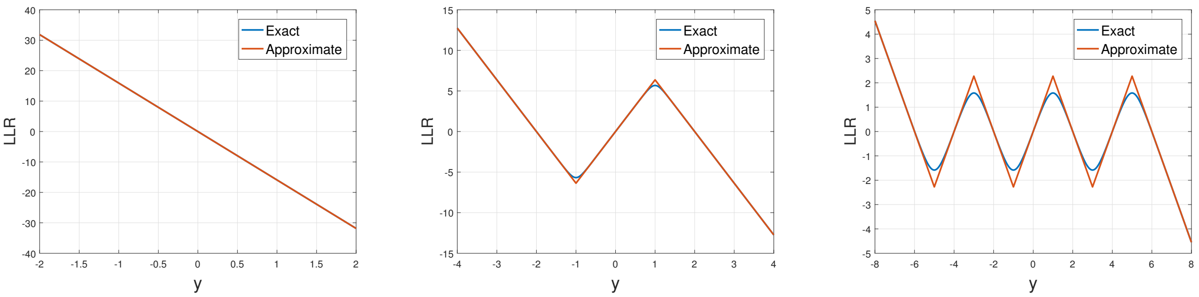
$$\eta_{\text{CC-D}} = \frac{K}{NB\sum_{l=1}^{L}\prod_{l'=1}^{l-1}P_{l'}^K}(1-P_L^K), \qquad (17)$$

where P_l^K is the FER of the l^{th} transmission, given that the previous $l-1$ transmissions of the codeword failed. For CC-I, the throughput is

$$\eta_{\text{CC-I}} = \sum_{n=1}^{B}\frac{K_n}{N\sum_{l=1}^{L}\prod_{l'=1}^{l-1}P_{n,l'}^{K_n}}(1-P_{n,L}^{K_n}). \qquad (18)$$

V. Polar Code Design Methods for BPSK

In this section, we start by explaining the simulation-based design method for SCD. Then, we describe the design method for NC and CC HARQ using GA which can design the code with low complexity.



(a) (b) (c)

Figure 3: Comparison of the exact and the approximate estimation of LLRs as a function of y for a) a BPSK, b) the first level of 4-PAM, and c) the first level of 8-PAM, respectively.

A. Simulation-based Code Design for SCD

To design polar codes, the positions of the information bits must be determined. Determining the information-set by using Monte Carlo simulation, proposed by Arikan in [4], is one of the methods of polar code design which benefits from high flexibility for adapting to a variety of practical channels. In the simulation-based design method, as described in [20], the transmission of a large number of message words is simulated and SCD decodes bits subsequently from the first to the last. Then, the number of the first error events[2] for each bit-channel is measured. The number of transmitted codewords for achieving the sufficient statistic can be decreased if, after recording each first error event, the corresponding bit is corrected to prevent propagating that error and the next bit-channels are examined subsequently. When the polar code is designed for a predetermined rate R at a specific SNR, the best information-set is chosen to minimize the FER by finding the K message bit positions with the lowest number of the first error events.

By recording the position of each first error event for each simulated codeword, it is easy to evaluate the FER for any information-set. Any simulated codeword with at least one first error event in positions specified by that information-set would also have been decoded incorrectly by a real decoder for the polar code defined by that information-set. Thus, the FER for a given information-set can be approximated by dividing the number of incorrectly decoded codewords by the number of simulated codewords.

More formally, suppose we simulate N_{SIM} codewords of length N at a given SNR. Let $\epsilon_{m,\kappa} = 1$ if a first error event occurred in the κ^{th} bit-channel during the m^{th} simulated codeword transmission, and $\epsilon_{m,\kappa} = 0$ otherwise[3]. For a given hypothetical information-set, $\mathcal{A}$, the m^{th} simulated codeword would have been decoded incorrectly if $\Sigma_{\kappa\in\mathcal{A}}\,\epsilon_{m,\kappa} > 0$. Let $\xi_m = 1$ if $\Sigma_{\kappa\in\mathcal{A}}\,\epsilon_{m,\kappa} > 0$. Out of the N_{SIM} simulated codewords, the number of codewords that would have been incorrectly decoded is $\Sigma_{m=1}^{N_{\text{SIM}}}\xi_m$.

Using this method, it is straightforward to design polar codes to maximize the throughput. Once the simulation of a sufficiently large number of codewords has completed (typically $N_{\text{SIM}} = 10000$ codewords is sufficient for low-to-moderate SNRs) at the desired SNR and the position of the first error events has been recorded ($\epsilon_{m,\kappa}$), the information-set of the minimum FER polar codes for every code rate from $\frac{1}{N}$ to 1 is determined (i.e., $\forall K \in \{1,...,N\}$) and the corresponding FER is approximated. Then, the code rate that maximizes the throughput, (15), is determined, and the associated information-set is used to define the optimal polar code at that SNR.

B. GA-based Code Design for SCD

As shown in [7], polar codes can be designed with low complexity using the GA. In GA, the LLR distributions for each node in polar graph can be approximated with a Gaussian distribution, in which the variance of the distribution σ^2 is two times the average LLR of a specific bit-channel. Therefore, to implement the GA-based design for polar codes we only need to update the average LLR through the polar graph. The updating average LLR rule for upper bit-channels is $\phi^{-1}(1-[1-\phi(\bar{\lambda}_1)][1-\phi(\bar{\lambda}_2)])$ and for lower bit-channels is $\bar{\lambda}_1+\bar{\lambda}_2$ where $\bar{\lambda} = E[\lambda_1]$ [21]. From [22], $\phi(h)$ is approximated as

$$\phi(h) \triangleq \begin{cases} 1 & h = 0, \\ \exp(-0.4527h^{0.86} + 0.0218) & 0 < h \leq 10, \\ \sqrt{\frac{\pi}{2}}(1 - \frac{10}{7h})\exp(\frac{-h}{4}) & 10 < h. \end{cases} \tag{19}$$

Here, we design the throughput-optimal polar codes using the GA and compare it with the simulation-based approach. For designing the code we call Function NC-Binary-Design(GA-BER(γ,N),N) in Appendix A. Function NC-Binary-Design approximates the FER of the code based on a well-known bound given as [11]

$$P_K = 1 - \prod_{i=1}^{K}(1 - v_i), \tag{20}$$

where v_i is the bit error rate (BER) of the i^{th} bit-channel given previous bit-channels are frozen. These BERs are estimated using Function GA-BER in Appendix A. In the next step, the corresponding throughput for each message-length

[2] For each codeword, the first error event defined as the first erroneous output bit. This error does not include the propagated error and just represents the error of the bit-channel.

[3] Let $Z_\kappa = \Sigma_{m=1}^{N_{\text{SIM}}}\epsilon_{m,\kappa}$ be the total number of first error events in the κ^{th} bit-channel. The information-set, $\mathcal{A}$, of the minimum-FER code of rate K/N contains the values of κ with the K smallest values of Z_κ.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2018.2880204, IEEE Transactions on Communications

is computed and the code rate with the highest throughput is chosen. From (15) and (20), the throughput can be written as

$$\eta_{\mathrm{NC},K} = \frac{K}{NB}\prod_{i=1}^{K}(1-v_i). \tag{21}$$

Proposition 2. *The FER estimation in P_K is monotonically increasing with respect to K.*

Proof: The FER estimation in (20) can be written in a recursive form as

$$P_K = 1-(1-v_K)\prod_{i=1}^{K-1}(1-v_i) = P_{K-1} + v_K\prod_{i=1}^{K-1}(1-v_i). \tag{22}$$

Thus, (22) monotonically increases with K since $v_K\prod_{i=1}^{K-1}(1-v_i) > 0$ due to $0 < v_i < 1$. ■

Proposition 3. *The throughput in (15) is a unimodal function of the code rate.*

Proof: The first forward difference can be given as

$$\Delta\eta_K = \eta_{K+1} - \eta_K = \frac{1}{NB}\big[1-(K+1)v_{K+1}\big]\prod_{i=1}^{K}(1-v_i). \tag{23}$$

Since $\Delta\eta_K > 0$ for $K < \frac{1}{v_{K+1}} - 1$ and $\Delta\eta_K < 0$ for $K > \frac{1}{v_{K+1}} - 1$, the difference equation has only one root, so η_K has only one maxima. Thus, (15) is unimodal. ■

The throughput vs. the code rate for polar codes with length 4096 at SNRs of -2, 0 and 2 is shown in Fig. 4. From Proposition 2, we know $P_K > P_{K-1}$. Thus, the throughput initially grows (nearly linearly for the FER-minimal codes) with the code rate until the rate gets sufficiently high that the effects of the FER start to dominate in (15), after which point the throughput drops dramatically. The existence of an optimal rate to maximize the throughput is clear which is shown in Proposition 3.

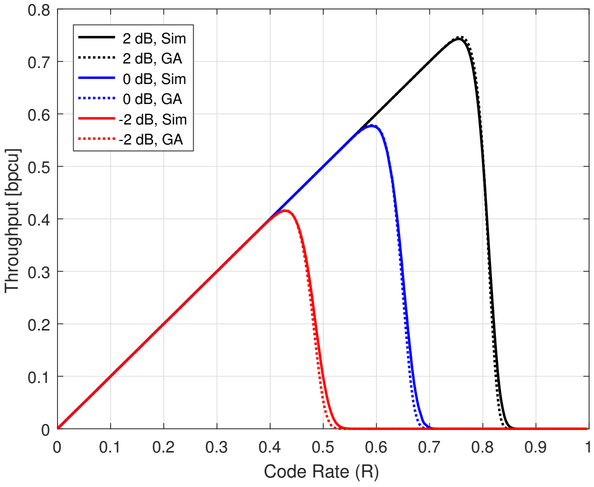


Figure 4: Throughput of NC protocols vs. the code rate with $N = 4096$ bits at different SNRs for BPSK with SCD.

C. Code Design for Chase-combining

To design the code for the CC HARQ schemes, the code design method for NC should be modified since the average LLR with each retransmission increases to $\frac{4l}{N_0}$, where $l \in \mathbb{N}$ is the retransmission attempt number. In this case, the procedure of the design includes running the Function GA-BER in Appendix A to estimate the order of bit-channels for the first transmission since the majority of codewords are decoded correctly in the first step. To find the rate corresponding to the maximum throughput, the FERs using the Function GA-BER for each retransmission are estimated and (17) is computed. For the stopping criterion, one can use a threshold on the probability of error of the maximum message length, e.g., $P_L^N < 10^{-6}$. However, this criterion converges slowly. As a fast criterion, the lack of change of the maximum throughput is employed in this paper. The code design for CC HARQ is described formally in Function CC-Binary-Design in Appendix A. To design the code, we should call Function CC-Binary-Design(γ,N). The codes designed for CC have slightly higher rates than the codes designed for NC.

D. Code Design for Incremental Redundancy

CC has the advantage over NC that the received sample from all the retransmissions are used when making decisions. IR schemes have the additional advantages that each retransmission does not have to be of the same length, and can belong to an extended code instead of merely repeating code bits. A very good incremental redundancy scheme based on punctured polar codes was recently proposed in [23]. In this scheme, a family of L polar codes of lengths $N_1 < N_2 < \ldots < N_L$ are designed for a fixed message word of K bits. By using quasi-uniform puncturing [24] along with careful repetition of message bits at the encoder and some minor modifications to the polar decoder (see [23] for details), these codes are rate-comparable. The throughout-maximizing design technique presented here can easily be extended to this IR scheme. Although better performance can be achieved by using a small incremental transmission size (e.g., one bit at a time), to simplify the scheduling and to provide a meaningful comparison to the NC and CC schemes, in the following we consider an increment of N bits (i.e., $N_l = lN$). The design procedure is the same as described as in Function CC-Binary-Design, except $\bar{\lambda} = 4 \times 10^{\gamma/10}$ is used in line 3 and v =GA-BER($\bar{\lambda}, N_l$) is used in line 4 and the BER sorting is repeated for all retransmissions.

A comparison of the throughput of NC, CC and IR protocols is provided in Fig 5. The IR scheme is designed for $L = 4$. Observe that the IR code achieves the throughput of NC in the first transmission since the information set of codes are the same, and the throughput of retransmissions shows a higher peak.

VI. Polar Code Design Methods for QAM

To design MLPCM for PAM and QAM constellations, the GA method can be employed. In this section, we explain the design steps of MLPCM and provide algorithms to maximize the throughput of HARQ schemes.

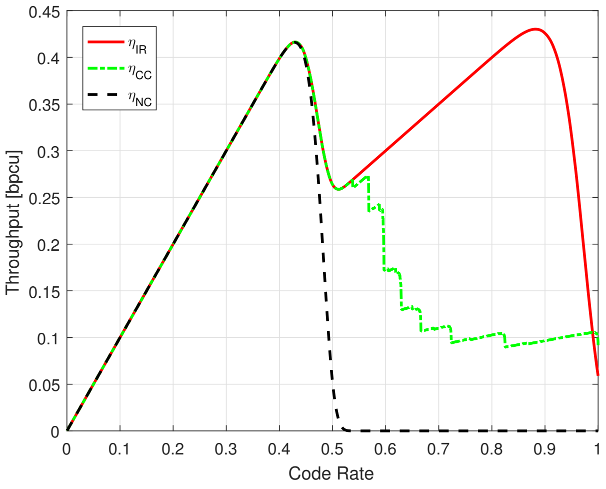


Figure 5: Throughput comparison of NC, CC, and IR protocols vs. the code rate with $N = 4096$ bits and SCD for BPSK at $\gamma = -2$ dB.

A. GA-based Code Design for SCD

To construct polar codes using GA, the average LLR for each binary channel of a QAM constellation is estimated and an independent binary polar code is constructed for each level. Due to the Gaussian noise, the LLR distribution for a BPSK constellation in an AWGN channel is $\mathcal{N}(\frac{4}{N_0}, \frac{8}{N_0})$. However, the LLR distributions of each level of a multilevel coding scheme for PAM and QAM constellations are not known. The average LLR of QAM with 2^B points can be estimated using the average LLRs of the constituent PAM with $2^{B/2}$ points based on the transform described in Theorem 1. In this theorem, we indeed use a linear block code to estimate the LLRs of QAM with 2D SPM from LLRs of PAM. This linear block code is similar to the second stage of a polar encoder and hence, a modified GA can be employed to estimate the average LLRs of QAM. The corresponding average LLR estimation for $b = 1, \ldots, B/2$ can be expressed as

$$\begin{aligned} \bar{\lambda}_{\mathrm{QAM},2b-1} &= \phi^{-1}(1 - [1 - \phi(\bar{\lambda}_{\mathrm{PAM},b})]^2), \\ \bar{\lambda}_{\mathrm{QAM},2b} &= 2\bar{\lambda}_{\mathrm{PAM},b}. \end{aligned} \tag{24}$$

To estimate the average LLRs of a PAM constellation, we use the piecewise linear estimation in Proposition 1. Consequently, since the LLR is a linear function of y, the average LLR of each level of PAM, given a zero is transmitted, can be computed through integration. For the first level of PAM, the integral is given by

$$\bar{\lambda}_1 = \sum_{d|x_d \in \mathcal{X}_1^0} \int_{\Omega_d} \Pr(x_d)\lambda_1(y) \frac{1}{\sqrt{\pi N_0}} e^{\frac{-(y-x_d)^2}{N_0}} dy, \tag{25}$$

where $\Pr(x_d)$ is the probability of the transmission of x_d within $\mathcal{X}_1^0$. For example for 4-PAM, $\Pr(x_d) = 0.5$ and for 8-PAM, $\Pr(x_d) = 0.25$. For other levels, the integration is taken over the corresponding set $\mathcal{X}_n^0$. Therefore, the general form of (25) can be written as

$$\bar{\lambda}_n = \sum_{d|x_d \in \mathcal{X}_n^0} \int_{\Omega_d} \Pr(x_d)\lambda_n(y) \frac{1}{\sqrt{\pi N_0}} e^{\frac{-(y-x_d)^2}{N_0}} dy, \tag{26}$$

where $\lambda_n(y)$ is given by the LLR approximation in (14). As a numerical example, for 4-PAM and 8-PAM, the average LLRs of the binary channels are $[6.3, 31.9]$ and $[0.7, 6.0, 30.5]$ at $\gamma = 10$ dB, respectively.

In [7], it is noted that the FER of a MLPCM can be approximated with $1-\prod_{n=1}^{B}(1-P_n)$. By extending this bound, the total FER of the MLPCM can be approximated as

$$P_K = 1 - \prod_{n=1}^{B}\prod_{i=1}^{K_n}(1 - v_n^i), \tag{27}$$

where v_n^i is the BER of the i^{th} bit-channel in n^{th} level given previous bit-channels are frozen.

To find the level code rates, [7] suggests using the equal error probability rule in which all levels of a MLPCM have approximately the same FER. However, this requires solving a program to find code rates. From (27), one can observe that the MLPCM works like a longer single binary polar code while it observes a variety of equivalent SNRs corresponding to different levels. Thus, to determine level code rates, the bit-channel reliabilities $v_n^i, \forall i = 1, \ldots, N$ can be measured for all levels $n = 1, \ldots, B$ and among them, those with the lowest genie-aided BERs are chosen as the total information-set [11]. This automatically determines the rate of each level since some bit-channels of each level are in the total information-set. When we use this rule to design the code, the FERs of different levels are very close. However, this rule does not require solving any program to determine the code rates which highly simplifies the MLPCM design. The entire code design procedure for the NC-D protocol is mentioned in Algorithm 1 provided in Appendix A.

An algorithm for designing throughput optimal codes for NC-I protocol is presented in Algorithm 2 in Appendix A. Due to the independence of levels, the algorithm designs a binary code for each level independently. The same algorithm can be developed for the CC-I protocol by substituting the Function CC-Binary-Design instead of the Function NC-Binary-Design in Algorithm 2. Similar procedure can be used to design the IR-I protocol based on the code design method explained in Section V-D for BPSK. Due to the space limitation, we do not mention the design algorithm for the CC-D and IR-D protocols but it is only the extension of Function CC-Binary-Design and Algorithm 1.

VII. Rate Matching Algorithm for SCLD

For decoding of each output bit with SCD, the information of other previously decoded bits and the future information bits are not used. To overcome these shortcomings, the SCLD records a list containing different possible decoded message words and keeps only L most likely ones after each step [12]. A CRC sequence is usually added to message bits when SCLD is used, to increase the probability of finding the most likely message word. Throughout this paper for SCLD, only one CRC sequence is used for both list decoding and ARQ. Typically, the codes designed for SCD are used for SCLD as well since the SCL core decoder is SCD. However, these codes are suboptimal for SCLD.

When throughput-maximizing codes optimized for SCD are used with SCLD, the FER is lower than the FER of SCD. Even

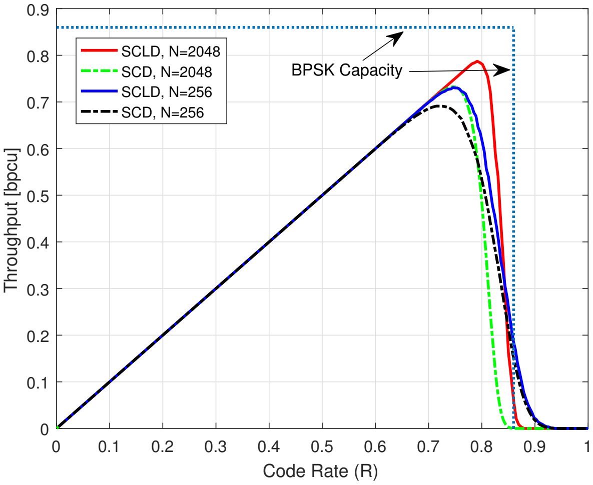


Figure 6: Throughput of NC protocol vs. the code rate with different code lengths at $\gamma = 2$ dB for BPSK.

though this slightly improves the throughput, it is not highly effective on the term $(1-P_K)$ in (15). However, since R is numerically more dominant in (15) when the FER is small, it can be increased more significantly to improve the throughput. Therefore, we introduce a rate matching algorithm for SCLD. The algorithm employs the golden-section search method to find the code rate corresponding to the maximum throughput.

The golden-section search method iteratively measures the objective function at different points and updates the answer range interval $[a, b]$ until this interval is narrowed down around the final value of the decision variable. For a detailed explanation of the golden-section search method refer to [25] and references therein. Here, the objective function is the actual throughput of SCLD measured using simulation and the decision variable is the message word length. The proposed algorithm is fast, e.g., for $N = 16384$ it finds the optimum rate in around 15 iterations, corresponding to 16 evaluations of the objective function. For initialization of the algorithm, we use $a = K_{\text{SCD}}$, the message word length for SCD, and $b = \min(a + \frac{BN}{10}, BN)$. The algorithm is formally presented in Function SCLD-Rate-Match in Appendix A. Note than when we use the algorithm for the NC-D protocol, we should call Function SCLD-Rate-Match(γ,K,N,$\mathbf{idx}_{\text{tot}}$,$B$) to repeat the simulation for all B binary channels of a constellation. However, for the NC-I protocol we call SCLD-Rate-Match($10\log_{10}\frac{\bar{\lambda}_n}{4}$,$K_n$,$N$,$\mathbf{idx}_n$,1) for each binary channel $n = 1, \ldots, B$, independently.

A comparison of throughput vs. the code rate for SCD and SCLD at different lengths is presented in Fig. 6. Observe that the rate-matched codes for SCLD substantially improve the throughput for all code lengths. Furthermore, as we increase N, the throughput tends to the capacity at which the throughput and the code rate of the throughput-maximizing code eventually are equal.

VIII. Numerical Results and Discussions

In this section, we provide the performance analysis of the code design algorithms described in Sections V and VI, respectively. The system described in Section II is used for all simulations and the CRC sequence is CRC-16-CCITT.

The throughput of SC- and SCL-decoded polar codes of length 4096 vs. SNR (E_s/N_0) is shown in Fig. 7 in comparison to BPSK capacity. The SCLD list size is 32 for all curves. The lowermost black curve shows the throughput of the polar code designed using SCD and decoded with SCD that achieves the throughput of 80% of the capacity at 0 dB. The second black curve is the throughput of the code designed for SCD and decoded using SCLD which achieves 82.5% of the capacity at 0 dB. The topmost curve under the capacity shows the performance of the code designed for SCD and rate matched for SCLD which achieves the 89.3% of the capacity at 0 dB. Therefore, the use of SCLD for decoding of codes designed for SCD does not change the throughput substantially in comparison to SCD. However, employing Function SCLD-Rate-Match for the rate matching can substantially improve the throughput of the code used with SCLD.

Fig. 7 furthermore provides a comparison of the throughput-maximizing polar codes and parallel concatenated (turbo) codes employed in LTE-A [26]. The BCJR decoder with 5 iterations is employed for decoding of the turbo codes with codeword lengths of around 4096. The LTE-A turbo code rate is optimized using the golden-section search (similar to Function SCLD-Rate-Match but the BCJR is employed) to maximize the throughput. The range of message word lengths is limited to 40:8:512, 528:16:1024, 1056:32:2048 and 2112:64:4200 bits which provides us with 157 different choices for the code rate. In this case, a golden-section search is used to search all the possible 157 choices for the code rate and selects the code rate corresponding to the highest throughput. Due to code rate limitations, the optimization procedure was only

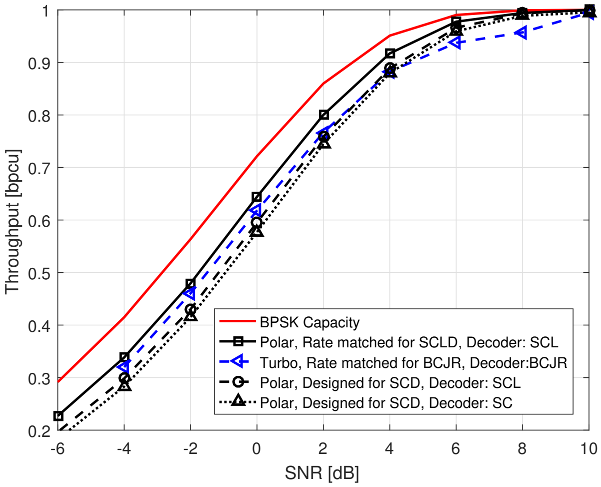


Figure 7: Throughput comparison of polar codes of length 4096 optimized for SCD decoded with SCD and SCLD, and the SCLD rate matched codes decoded with SCLD.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2018.2880204, IEEE Transactions on Communications

applied in the SNR range between -4 and 10 dB. Furthermore, the turbo code lengths are slightly higher than 4096. It can be observed that turbo code performance is close to the optimized polar code using SCLD-Rate-Match at low SNRs. However, as the SNR increases, the performance of turbo degrades and at high SNRs, it is even worse than the polar code optimized for SCD. Note that the complexity of BCJR with 5 iterations is more than that of SCLD.

The comparison of the CC HARQ schemes proposed in [10] for SCD and [9] for SCLD and the CC HARQ design method introduced in Function CC-Binary-Design is provided in Fig. 8 for BPSK. To construct the polar codes using algorithms provided in [9] and [10], the K at each SNR is considered the same as K found using Function CC-Binary-Design. The results indicate that the methods proposed in [9] and [10] with at least $N - K$ times more complexity, cannot generate codes better than the Function CC-Binary-Design.

In addition, in Fig. 8, the comparison of IR codes with CC and NC is also provided. To construct IR codes, the method described in Section V-D is used. The NC codes optimized using Function NC-Binary-Design have approximately the same performance as the codes generated for the CC using Function CC-Binary-Design for situations where we can adapt the code when the SNR changes. In this case, CC does not have any sensible advantage over NC since when we design the codes for NC or CC, the algorithms try to minimize the number of retransmissions and keep it in order of at most one to maximize the throughput. Observe that IR code outperforms NC with SCD and SCLD by 4.53% and 3.82% of the BPSK capacity at $\bar{\gamma} = -4$ dB, respectively. At moderate-to-high SNRs, the IR code cannot improve the throughput anymore.

The comparison of NC-I and NC-D protocols with SCD and SCLD is presented in Fig. 9 for 16-QAM. At all SNRs, NC-I and NC-D with SCLD perform slightly better than the corresponding protocol with SCD. At 4 dB, NC-D with SCD and SCLD achieves 69% and 73% of the capacity and NC-I with SCD and SCLD achieves 74% and 84% of the capacity, respectively. For all SNRs, the throughput of the code is essentially identical regardless of whether the exact LLR calculation of (2) or the simplified LLR calculation of Section III is used. This is despite the fact that the simplified LLR approximation is not very exact at low SNR.

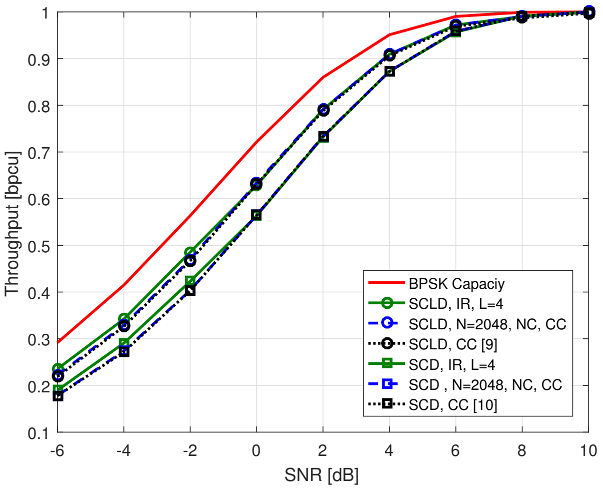


Figure 8: Comparison of proposed NC, CC, and IR codes and code design methods in [10] and [9] for SCD and SCLD, respectively.

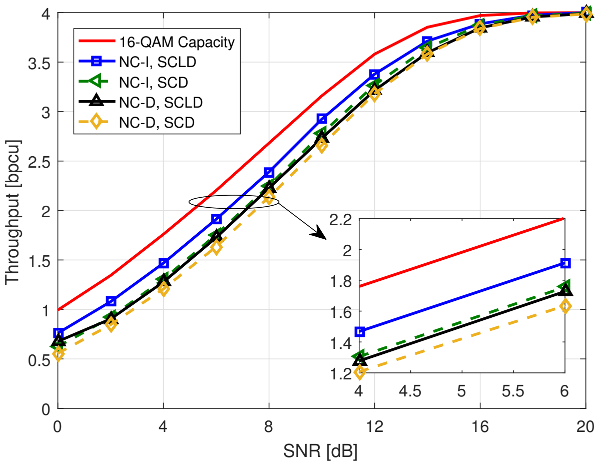


Figure 9: Throughput comparison of NC-D and NC-I protocols with SCD and SCLD for $N_{\text{tot}} = 2048$ and 16-QAM.

In all previous results, we assumed we can adaptively change the code and the modulation. In those cases, CC HARQ protocols do not provide any advantage over NC protocols. However, once the number of codes is limited, the advantage of CC HARQ is highlighted. In Fig. 10, the performance of CC-I protocol with two codes designed at 4 dB and 14 dB is compared with the IR HARQ and CC HARQ schemes constructed using polar codes and BICM in [13] for 16-QAM. We observe that when we use the code designed at 14 dB for the whole range of SNR, IR HARQ is better than the proposed throughput-optimal codes for CC-I protocol at a few SNRs. However, on average the proposed MLPCM with CC-I protocol performs up to 3 dB better than IR HARQ and CC HARQ in [13]. In case we use one more code designed at 4 dB, CC-I at all SNRs achieves higher throughput than schemes proposed in [13]. The reason for this superiority is the good design of MLPCM scheme in conjugation with CC-I scheme. We also compared the MLPCM scheme with IR-I HARQ constructed based on the method explained in Section V-D. Observe that it is only slightly better than CC at low SNRs.

IX. Conclusion

In this paper, we simplified the accurate LLR estimation of QAM constellations with 2D SPM using a simple linear relationship and two PAM LLR estimators and we proposed a method for generating a SPM for QAM constellations with an even number of address-bits. We also simplified the LLR estimation of PAM constellations with SPM. Furthermore, we proposed a set of algorithms for designing MLPCM by

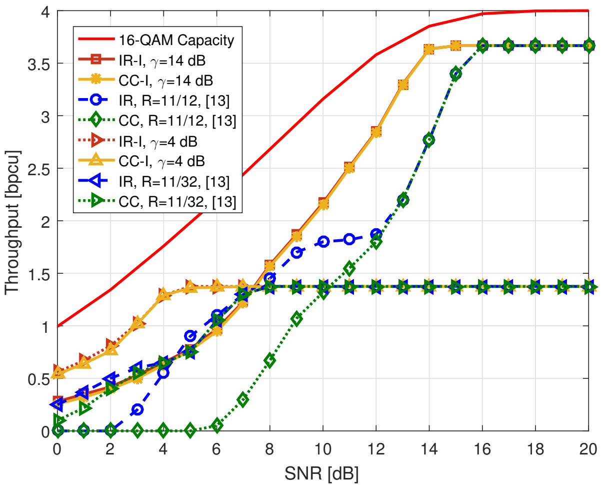


Figure 10: Comparison of CC-I with two codes and the IR and CC HARQ proposed in [13] for 16-QAM.

maximizing the throughput for NC and CC HARQ schemes. The numerical results show the codes constructed using these methods perform very close to the capacity, e.g., within 1.2 dB of the 16-QAM capacity with $N = 2048$ bits. The results indicate an improvement up to 15% of the capacity at 4 dB when we use level-independent protocols and rate-matched SCLD. Due to the substantial throughput improvement that can be achieved using carefully designed codes, the idea of throughput-based polar code design should be extended to designing optimal codes for IR HARQ systems in the future.

Appendix A
Algorithms to Generate Codes

The algorithms used in this paper to generate codes are mentioned in this section.

Function NC-Binary-Design($\mathbf{v}$,N)

```
Input: v and N
Output: v: The sorted indices of the information-set
        idx and the message length $K_{\text{opt}}$
Procedures: [v,idx] =Sort(v): Sorts the input vector in
  an increasing order and outputs the sorted indices idx
  and sorted values v.
[v,idx]=Sort(v)                              ▷ Sort the bit-channels
for $\kappa = 1 : N$ do
    $P_\kappa = 1 - \prod_{i=1}^{\kappa}(1 - v_i)$        ▷ Estimate FER for all $K$s
    $\eta_\kappa = (1 - P_\kappa)\frac{\kappa}{N}$                ▷ Estimate $\eta_\kappa$
$K_{\text{opt}} = \arg\max_{\kappa} \eta_\kappa$             ▷ Find $K$ with highest $\eta_\kappa$
$R = \frac{K_{\text{opt}}}{N}$
return idx, $K_{\text{opt}}$
```

Function GA-BER($\bar{\lambda}$,N) (From [21], modified)

```
Input: $\bar{\lambda}$ and $N$
Output: v: The vector of the BERs of bit-channels
$\bar{\lambda}_{1,1} = \bar{\lambda}$
for $m = 1 : \log_2 N$ do
    for $i = 1 : 2^{m-1}$ do
        $\bar{\lambda}_{2i-1,m+1} = \phi^{-1}(1 - [1 - \phi(\bar{\lambda}_{i,m})]^2)$
        $\bar{\lambda}_{2i,m+1} = 2\bar{\lambda}_{i,m}$
for $\kappa = 1 : N$ do
    $v_\kappa = Q\Big(\sqrt{\frac{\bar{\lambda}_{\kappa,\log_2(N)+1}}{2}}\Big)$
return v
```

Function CC-Binary-Design(γ,N)

```
Input  : $\gamma$ [dB] and $N$
Output: The maximum throughput code and the
        message length $K_{\text{SCD}}$
$\eta' = 1$, $\eta = 0$, $\bar{N}_0^\kappa = 0$, $P_0^\kappa = 0$,$l = 1$
while $\eta' \neq \eta$ do
    $\bar{\lambda} = 4l \times 10^{\gamma/10}$
    v=GA-BER($\bar{\lambda}$,$N$)              ▷ Determine BERs
    if $l = 1$ then
        [v,idx]=Sort(v)           ▷ Sort the bit-channels
    else
        v = v(idx)
    for $\kappa = 1 : N$ do
        $P_l^\kappa = 1 - \prod_{i=1}^{\kappa}(1 - v_i)$          ▷ FER estimation
        $\bar{N}_l^\kappa = \bar{N}_{l-1}^\kappa + N\prod_{l'=1}^{l-1} P_{l'}^\kappa$  ▷ Effective code length
        $\bar{\eta}_\kappa = \frac{\kappa}{\bar{N}_l^\kappa}(1 - P_l^\kappa)$             ▷ Throughputs
    $K_{\text{opt}} = \arg\max_{\kappa} \bar{\eta}_\kappa$       ▷ Find $K$ with highest $\eta_\kappa$
    $\eta' = \eta$
    $\eta = \bar{\eta}_{K_{\text{opt}}}$                 ▷ Find highest throughput
    $l = l + 1$         ▷ Increment retransmission number
$R_{\text{SCD}} = \bar{\eta}_{K_{\text{opt}}}$
return idx, $K_{\text{SCD}}$
```

Algorithm 1: GA-based design for NC-D with QAM

```
Input  : $\bar{\boldsymbol{\lambda}}_{\text{PAM}}$ for PAM, B, and N
Output: The maximum throughput codes and $K_{\text{SCD}}$
Procedures: AverageLLR($\bar{\boldsymbol{\lambda}}_{\text{PAM}}$): Calculates $\bar{\lambda}$ of a
  QAM based on (24) for a PAM constellation.
$\bar{\boldsymbol{\lambda}}_{\text{QAM}}$=AverageLLR($\bar{\boldsymbol{\lambda}}_{\text{PAM}}$)
for $n = 1 : B$ do
    $\mathbf{v}_n$=GA-BER($\bar{\lambda}_{\text{QAM},n}$,$N$)
[$\mathbf{idx}_{\text{tot}}$,$K$]=NC-Binary-Design([$\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_B$],$BN$)
return $\mathbf{idx}_{\text{tot}}$, $K_{\text{SCD}}$
```

```
Algorithm 2: GA-based design for NC-I with QAM
  Input  : $\bar{\boldsymbol{\lambda}}_{\text{PAM}}$: The average LLR vector for a PAM
           and $N$
  Output: The maximum throughput codes and the
           message length for the SCD
  $\bar{\boldsymbol{\lambda}}_{\text{QAM}}$=AverageLLR($\bar{\boldsymbol{\lambda}}_{\text{PAM}}$)
  for $n = 1 : B$ do
      $\mathbf{v}_n$=GA-BER($\bar{\lambda}_{\text{QAM},n}$,$N$)
      [$\mathbf{idx}_n$,$K_{\text{SCD},n}$]=NC-Binary-Design($\mathbf{v}_n$,$N$)
  return $\mathbf{idx}_1$,..., $\mathbf{idx}_B$, $\mathbf{K}_{\text{SCD}}$
```

```
Function SCLD-Rate-Match($\gamma$,$K_{\text{SCD}}$,$N$,$\mathbf{idx}$,$B$)
  Input  : $\gamma$, $K_{\text{SCD}}$, $N$, $\mathbf{idx}$, and $B$
  Output: The maximum throughput and the
           corresponding code rate for the SCLD
  Procedures: f($K$)=SCLD_Throughput($\gamma$,$K$,$N$,$\mathbf{idx}$,$B$):
    Employs SCLD to simulate and estimate the
    throughput of a MLPCM with a total message length
    of $K$, a level-code length of $N$, the indices idx
    designed for SCD, a $2^B$-QAM constellation and $\gamma$.
  Golden ratio: $\rho = \frac{\sqrt{5}-1}{2}$
  $a = K_{\text{SCD}}$
  $b = \min(a + \frac{BN}{10}, BN)$
  $k_1 = \lfloor \rho a + (1-\rho) b \rfloor$            ▷ First point to test
  $f(k_1)$ =SCLD_Throughput($\gamma$,$k_1$,$N$,$\mathbf{idx}$,$B$)
  $k_2 = \lfloor (1-\rho) a + \rho b \rfloor$            ▷ Second point to test
  $f(k_2)$ = SCLD_Throughput($\gamma$,$k_2$,$N$,$\mathbf{idx}$,$B$)
  while $|a - b| > 1$ do
      if $f(k_1) > f(k_2)$ then
          $b = k_2,\ k_2 = k_1,\ f(k_2) = f(k_1)$
          $k_1 = \lfloor \rho a + (1-\rho) b \rfloor$
          $f(k_1)$ = SCLD_Throughput($\gamma$,$k_1$,$N$,$\mathbf{idx}$,$B$)
      else
          $a = k_1,\ k_1 = k_2,\ f(k_1) = f(k_2)$
          $k_2 = \lfloor (1-\rho) a + \rho b \rfloor$
          $f(k_2)$ = SCLD_Throughput($\gamma$,$k_2$,$N$,$\mathbf{idx}$,$B$)
  return $k_1$
```

References

[1] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1983.

[2] D. Divsalar and F. Pollara, "On the design of turbo codes," *JPL TDA, Progr. Rep. 42-123*, Nov. 1995.

[3] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.

[4] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.

[5] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no.10, pp. 6562-6582, Sept. 2013.

[6] R. H. Deng, "A type I hybrid ARQ system with adaptive code rates," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 733-737, Feb./Mar./Apr. 1995.

[7] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221-3227, Nov. 2012.

[8] H. Saber and I. Marsland, "An incremental redundancy hybrid ARQ scheme via puncturing and extending of polar codes," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 3964-3973, Nov. 2015.

[9] H. Liang, A. Liu, Y. Zhang, and Q. Zhang, "Analysis and adaptive design of polar coded HARQ transmission under SC-List decoding," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 798-801, Dec. 2017.

[10] K. Chen, K. Niu, Z. He, and J. Lin, "Polar coded HARQ scheme with chase combining," in Proc. *IEEE WCNC*, Istanbul, Turkey, Apr. 2014.

[11] M. Seidl, A. Schenk, C. Stierstorfer, and J. B. Huber, "Polar-coded modulation," *IEEE Trans. Commun.*, vol. 61, no.10, pp. 4108-4119, Oct. 2013.

[12] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015.

[13] M. El-Khamy, H. P. Lin, J. Lee, H. Mahdavifar, and I. Kang, "HARQ rate-compatible polar codes for wireless channels," *IEEE Globecom*, San Diego, CA, USA, Dec. 2015.

[14] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 371-377, May 1977.

[15] Q. Luo and P. Sweeney, "Hybrid-ARQ protocols based on multilevel coded modulation," *Electronics Lett.*, vol. 39, no. 14, pp. 1063-1065, July 2003.

[16] J. Huber and U. Wachsmann, "Capacities of equivalent channels in multilevel coding schemes," *Electronics Lett.*, vol. 30, no. 7, pp. 557-558, Mar. 1994.

[17] A. B. H. Fredj and J-C. Belfiore, "LLR computation for multistage decoding," in Proc. *IEEE European Conf. Networks and Commun. (EuCNC)*, Athens, Greece, June 2016.

[18] R. Pyndiah, A. Picart, and A. Glavieux, "Performance of block turbo coded 16-QAM and 64-QAM modulations," in Proc. *IEEE Globecom*, Singapore, Nov. 1995.

[19] G. Gül, A. Vargas, W. H. Gerstacker, and M. Breiling, "Low complexity demapping algorithms for multilevel codes," *IEEE Trans. Commun.*, vol. 59, no. 4, pp. 998-1008, Apr. 2011.

[20] P. R. Balogun, I. Marsland, R. Gohary, and H. Yanikomeroglu, "Polar code design for irregular multidimensional constellations," *IEEE Access*, vol. 5, pp. 21941-21953, Nov. 2017.

[21] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of polar code constructions for the AWGN channel," arXiv preprint, No1501.02473, Jan. 2015.

[22] S-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657-670, Feb. 2001.

[23] P. Yuan, F. Steiner, T. Prinz, G. Böcherer, "Flexible IR-HARQ scheme for polar-coded modulation" in Proc. *IEEE Wireless Commun. and Networking Conf. Workshops (WCNCW)*, Barcelona, Spain, Apr. 2018.

[24] K. Niu, K. Chen, and J. Lin, "Beyond turbo codes: Rate-compatible punctured polar codes," in Proc. *IEEE ICC*, Budapest, Hungary, June 2013.

[25] R. A. Thisted, *Elements of Statistical Computing: Numerical Computation*. USA: Chapman and Hall/CRC, 1988.

[26] *LTE; Evolved universal terrestrial radio access (E-UTRA); Multiplexing and channel coding*, 3GPP TS 36.212, version 12.2.0, Oct. 2014.