Throughput-based Design of Polar Codes

Hossein Khoshnevis, Ian Marsland, and Halim Yanikomeroglu
Department of Systems and Computer Engineering
Carleton University, Canada
Email: {khoshnevis, ianm, halim}@sce.carleton.ca

***Abstract*—Typically, forward error correction codes are designed based on the minimization of the error rate for a given code rate. However, for applications that incorporates hybrid automatic repeat request (HARQ) protocol, the throughput is a more important performance metric than the error rate. Polar codes, a new class of error correction codes with simple rate matching and low complexity decoders, can be optimized efficiently for maximization of the throughput. In this paper, we first introduce a method to design throughput-maximizing polar codes for successive cancellation decoding (SCD). Furthermore, since the optimized codes for SCD are not optimal for SC list decoders (SCLD), we propose a rate matching algorithm to find the best rate for the SCLD decoders while using the polar codes optimized for SCD. The resulting codes provide throughput close to capacity with low decoding complexity when used with Type-I HARQ.**

I. Introduction

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, for many practical applications, 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 typically involves an exhaustive search over a set of code rates and employs simulation to estimate the throughput. However, as demonstrated in this paper this process can be greatly simplified for polar codes.

Hybrid automatic repeat request (HARQ), as an efficient error correction scheme, has been employed widely in communication systems including 4G wireless networks [6]. This scheme is expected to play a central role in 5G wireless networks as well, especially in use cases which require ultra-reliable communications. One of the ultimate measures to determine the quality of HARQ schemes is the throughput. Therefore, explicitly considering the throughput when designing the elements of HARQ schemes is of importance. The time-varying nature of the wireless channel requires the use of adaptive modulation and coding (AMC) schemes to achieve high throughput. Typically, when designing the AMC scheme for HARQ, an error correction code corresponding to the highest throughput for each SNR is chosen from a small set of available codes [7]. However, if the code can be designed with the objective of maximizing the throughput, the performance can substantially be improved.

When designing polar codes to minimize the frame error rate (FER), a set of frozen bit-channels must be chosen. The same procedure should be applied when designing polar codes based on the throughput, but the optimal set of frozen bits to maximize the throughput are 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 Type-I HARQ where for the retransmission of a failed codeword, the whole codeword should be retransmitted and incremental redundancy is not employed.

In this paper a simple method for designing polar codes is proposed that is based on maximizing the throughput, instead of the well-studied objective of minimizing the FER. These codes are designed for Type-I HARQ, with successive cancellation decoders (SCD). The throughout of the designed codes is compared with the powerful turbo codes employed in LTE-A [8]. In addition, since polar codes optimized for SCD are suboptimal for SC list decoding (SCLD), a 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. The rest of the paper is organized as follows: The system model is described in Section II, the polar encoder and decoder are reviewed in Section III, the polar code design based on the throughput for SCD is introduced in Section IV. Finally, numerical results are provided in Section V, and conclusions are presented in Section VI.

II. System Model

The communication system includes a single user transmitter and receiver that use a Type-I 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 polar code of length N and code rate $R = K/N$. Each codeword is modulated using BPSK and transmitted through an AWGN channel with a noise variance of $N_0/2$ per dimension. At the receiver, after decoding the received codeword, the correctness of CRC is checked to verify the message. In case of receiving a wrong message, retransmission of the entire codeword is requested by sending a repeat request

978-1-5090-5935-5/17/$31.00 ©2017 IEEE

(NACK) to the transmitter. The system uses AMC employing codes of different rate for each SNR.

III. POLAR CODES

In this section, we briefly review the encoding and decoding of polar codes. The general discrete-input memoryless channel (DMC) can be written as $W{:}\mathcal{C}{\to}\mathcal{Y}$ where $\mathcal{C}$ and $\mathcal{Y}$ denote input and output alphabets, respectively and the probability that given $c \in \mathcal{C}$, $y \in \mathcal{Y}$ is observed can be denoted by $W(y|c)$. The N independent uses of W results in channel W^N that can be expressed as

$$W^N(\mathbf{y} \mid \mathbf{c}) = \prod_{i=1}^{N} W(y_i \mid c_i). \tag{1}$$

Arikan in [4] introduces the polar transformation as a method of linearly transforming N independent uses of the channel to N correlated uses denoted by $W'{:}\mathcal{U}{\to}\mathcal{Y}$. By recursive applying the polar transformation on the binary DMC input, some of the resulting correlated bit-channels are improved and others are degraded. Ideally, for very long codes, the capacity of some bit-channels become one and therefore are used for transmission of information and the capacity of the rest of them become zero and are not used for transmission. For a code of length $N = 2^n$, the first set is called the information set, denoted by $\mathcal{A}$ with cardinality K, and the second set is called the frozen set denoted by $\mathcal{A}^{\mathcal{C}}$ with cardinality $N{-}K$.

A. Encoding

The polar encoder generates the codeword $\mathbf{c}$ from the message word $\mathbf{u} = [u_1, u_2, ..., u_N]$ according to $\mathbf{c} = \mathbf{u}\mathbf{G}_N$ where the K information bits are placed in the elements of $\mathbf{u}$ corresponding to the information set, and the elements of $\mathbf{u}$ corresponding to the frozen set are set to zero. The generator matrix, $\mathbf{G}_N$, is defined based on the binary polarization kernel [4]

$$\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{2}$$

as $\mathbf{G}_N = \mathbf{F}_2^{\otimes n}$, where $\mathbf{F}_2^{\otimes n}$ is the n-th Kronecker power of $\mathbf{F}_2$ defined as $\mathbf{F}_2^{\otimes n} = \mathbf{F}_2 \otimes \mathbf{F}_2^{\otimes (n-1)}$. The use of the Kronecker power facilitates constructing the polar encoder since the structure is repeated regularly.

B. SC Decoding

Proposed in [4], SCD benefits from low complexity which is beneficial in many applications. By applying the polar transform, the transition probabilities of the resulting bit-channels, given the code-bit channel output $\mathbf{y}$ and decoded bits $\mathbf{u}_{1:i-1}$[1], can be computed as [5]

$$W_i(\mathbf{y}, \mathbf{u}_{1:i-1} \mid u_i) = \frac{1}{2^{N-1}} \sum_{\mathbf{u}_{i+1:N} \in \{0,1\}^{N-i}} W^N\big(\mathbf{y} \mid (\mathbf{u}_{1:i-1}, u_i, \mathbf{u}_{i+1:N})\mathbf{G}_N\big). \tag{3}$$

[1]Throughout this paper, the subscript of $\mathbf{u}$ is used to represent the set of indices that define the subset of the elements of $\mathbf{u}$.

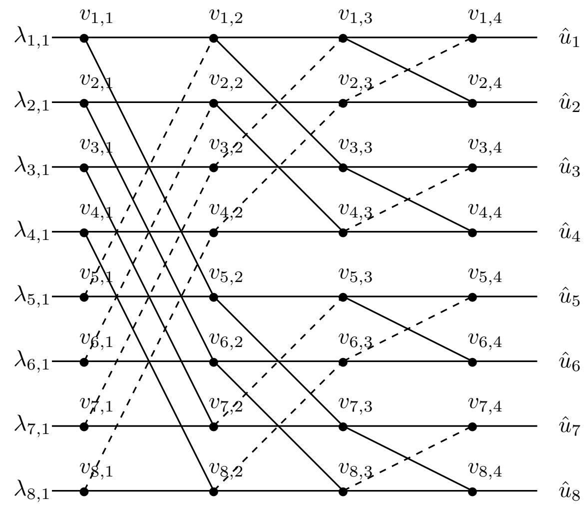

Fig. 1. Polar decoder structure.

Denoting the decoder output vector as $\hat{\mathbf{u}}$, the output log-likelihood ratios (LLRs), given $\mathbf{y}$, $\hat{\mathbf{u}}_{\mathcal{A}^c} = \mathbf{u}_{\mathcal{A}^c}$ and $\mathbf{u}_{1:i-1}$, are

$$\lambda_{i,n+1} = \ln \frac{W_i(\mathbf{y}, \hat{\mathbf{u}}_{1:i-1} \mid 0)}{W_i(\mathbf{y}, \hat{\mathbf{u}}_{1:i-1} \mid 1)}. \tag{4}$$

The final SCD decision rule is the hard decision on $\lambda_{i,n+1}$. Therefore, the SCD works based on successive decoding of elements of $\hat{\mathbf{u}}$. Fig. 1 illustrates the structure of the SCD which is also known as the polar code graph. Each node of this graph is represented by (i,j), where $i \in \{1,..,N\}$ and $j \in \{1,...,n+1\}$ are row and column indices of the graph. The nodes of graph can be partitioned as Type I and Type II based on their indices as

$$\text{type}(i,j) = \begin{cases} \text{I} & \lfloor \frac{i-1}{2^{n+1-j}} \rfloor \ (mod\ 2) \equiv 0 \\ \text{II} & \lfloor \frac{i-1}{2^{n+1-j}} \rfloor \ (mod\ 2) \equiv 1 \end{cases}. \tag{5}$$

The SCD employs soft-hard update decision rules to estimate the message word. As shown in Fig. 1, the decoder first starts by computing the input LLRs as $\lambda_{i,1} = \ln \frac{W_i(\mathbf{y}|0)}{W_i(\mathbf{y}|1)}$. For the intermediate nodes ($2 < j < n+1$), the LLR update rule of the decoder can be given as

$$\lambda_{i,j} = \begin{cases} \lambda_{i,j-1} \boxplus \lambda_{i^+,j-1} & \text{type I} \\ \lambda_{i,j-1} + (1 - 2\hat{v}_{i^-,j})\lambda_{i^-,j-1} & \text{type II} \end{cases}, \tag{6}$$

where $i^{\pm} = i \pm 2^{n+1-j}$ and $\hat{v}_{i,j}$ is a hard estimate of $v_{i,j}$ and the boxplus operator is defined as $\lambda_1 \boxplus \lambda_2 = 2\tanh^{-1}\big(\tanh(\frac{\lambda_1}{2})\ \tanh(\frac{\lambda_2}{2})\big)$. The LLRs are calculated and passed from left to right through the decoder graph till $\hat{v}_{1,n+1}$ is estimated. Then, it starts to pass the hard estimates recursively from right to left to compute the rest of intermediate nodes LLR. To estimate $\hat{v}_{i,j}$ the polar encoding structure is imitated.

IV. Polar Code Design Method

Typically, for the AWGN channel, polar codes are designed to minimize the FER for a given code rate, $R = K/N$, 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 are of primary importance.

For ARQ systems, messages are transmitted indefinitely until they are correctly received. As such, the more relevant metric is the throughput (also referred to as the "goodput"),

$$\eta = \frac{K - L_{\text{CRC}}}{N}(1 - \text{FER}), \tag{7}$$

which is the rate at which information is correctly received (in information bits per channel use). Using this new criteria, codes that are quite different from those that maximize the code rate or minimize the FER can be generated. In rest of this section, first we explain the design method for SCD. Then we adapt the codes designed for the SCD to the SCLD, introduced in [9], by using a rate matching algorithm.

A. Code Design for SCD

To design polar codes, the positions of the information bits (the information set) 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 [10], 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 frequency of the first error event[3].

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.

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

[3]The frequency of the first error event is the number of the first error events for each bit-channel divided by the total number of simulated codewords.

More formally, suppose we simulate N_{SIM} codewords of length N at a given SNR. Let $\epsilon_{i,k} = 1$ if a first error event occurred in the k^{th} bit-channel during the i^{th} simulated codeword transmission, and $\epsilon_{i,k} = 0$ otherwise. [4] For a given hypothetical information set, $\mathcal{A}$, the i^{th} simulated codeword would have been decoded incorrectly if $\Sigma_{k\in\mathcal{A}}\epsilon_{i,k} > 0$. Let $\delta_i = 1$ if $\Sigma_{k\in\mathcal{A}}\epsilon_{i,k} > 0$ out of the N_{SIM} simulated codewords, the number of codewords that would have been incorrectly decodes is $\Sigma_{i=1}^{N_{\text{SIM}}}\delta_i$.

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) at the desired SNR and the position of the first error events has been recorded ($\epsilon_{i,k}$), the information set of the minimum FER polar codes for every code rate from $(1{+}L_{\text{CRC}})/N$ to N/N is determined (i.e. $\forall K \in \{1{+}L_{\text{CRC}}, ..., N\}$) and the corresponding FER is approximated. Then the code rate that maximizes the throughput, (7), is determined, and the associated information set is used to define the optimal polar code at that SNR.

The throughput vs. the code rate for polar codes with length 4096 at SNRs of -2, 0 and 2 is shown in Fig. 2. The throughput initially grows linearly with the code rate until the rate gets sufficiently high, when the effects of the FER start to dominate in (7), after which point the throughput drops dramatically. The existence of an optimal rate to maximize the throughput is clear.

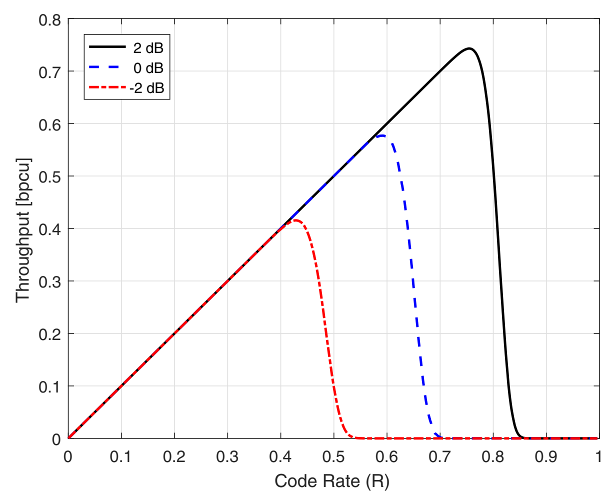

Fig. 2. Throughput vs. the code rate for SCD with $N = 4096$ bits at different SNRs.

B. Rate Matching Algorithm for SCLD

For decoding of each output bit with SCD, the information of other previously decoded bits and the future frozen 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 steps

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

[9]. 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 though, this slightly improves the throughput, it is not highly effective on the term (1−FER) in (7). However, since R is numerically more dominant in (7) 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 [11] to find the code rate corresponding to the maximum throughput.

As explained in [11], the golden 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. 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 $a = K_{\text{SCD}}$ and $b = \min(a + N/10, N)$ where K_{SCD} is the length of the message word of the code optimized for the SCD.

Algorithm 1 Rate Matching for SCLD

Input: Polar code constructed for SCD
Output: The maximum throughput and the corresponding code rate for SCLD

Procedures used in the algorithm:
SCLD_Throughput(R): Computes the throughput of a code with rate R based on the simulation and by employing SCLD. Here, the codes optimized in Section IV-A are used.
Constants:
Golden ratio: $\rho = (\sqrt{5} - 1)/2$
Variables:
a,b,$k1$,$k2$: Variables used for the message length.
$f(k1)$, $f(k2)$: Variables used for the throughput values.
Out_Rate: Code rate corresponding to maximum throughput for SCLD.
$Max\_Throughput$: Maximum throughput achieved by SCLD.
Initialization:
$a = K_{\text{SCD}}$
$b = \min(a + N/10, N)$
The body of algorithm:
$k1 = \lfloor \rho a + (1 - \rho)b \rfloor$
$f(k1)$ =*SCLD_Throughput*($k1/N$)
$k2 = \lfloor (1 - \rho)a + \rho b \rfloor$
$f(k2)$ = *SCLD_Throughput*($k2/N$)
while $|a - b| \leq 1$ **do**
  if $f(k1) > f(k2)$ **then**
    $b = k2$
    $k2 = k1$
    $f(k2) = f(k1)$
    $k1 = \lfloor \rho a + (1 - \rho)b \rfloor$
    $f(k1)$ = *SCLD_Throughput*($k1/N$)
  else
    $a = k1$
    $k1 = k2$
    $f(k1) = f(k2)$
    $k2 = \lfloor (1 - \rho)a + \rho b \rfloor$
    $f(k2)$ = *SCLD_Throughput*($k2/N$)
  end if
  Out_Rate=b/N
  $Max\_Throughput$= *SCLD_Throughput*(b/N)
end while

V. Numerical Results and Discussions

In this section, we provide the performance of the code design algorithms described in Sections IV-A and IV-B, respectively. The system described in Section II is used for all simulations and the CRC sequence is CRC-16-CCITT. Since the AMC is employed, nine different polar codes are designed for measured SNRs with a spacing of 2 dB over the interval $[-6, 10]$ dB.

In Fig. 3, the throughput of SC-decoded polar codes with different lengths changing from 4096 bits to 1048576 bits are shown. At 0 dB, the polar codes of lengths 1048576 and 4096 achieve 94.8% and 80% of the capacity, respectively.

The throughput of SC- and SCL-decoded polar codes of length 4096 is shown in Fig. 4 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 using Algorithm 1 which achieves the 89.3% of the capacity at 0 dB. Therefore, the use of SCLD for decoding of codes designed for SCD doesn't change the throughput substantially in comparison to SCD. However, employing Algorithm 1 for

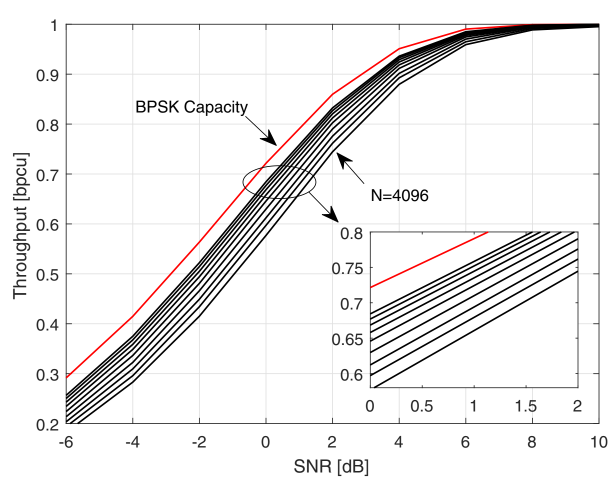

Fig. 3. Throughput comparison of polar codes of different lengths with SCD.

the rate matching can substantially improve the throughput of the code used with SCLD.

Fig. 4 furthermore provides a comparison of the throughput-maximizing polar codes and parallel concatenated (turbo) codes employed in LTE-A [8]. 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 Algorithm 1 to maximize the throughput. The range of message word lengths are 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, Algorithm 1 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 applied in 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 with Algorithm 1 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.

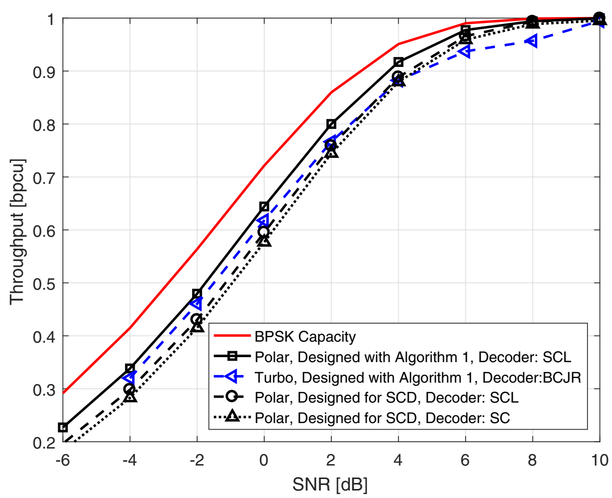

Fig. 4. Throughput comparison of polar codes of length 4096 optimized for SCD decoded with SCD and SCLD and the rate matched polar code using Algorithm 1 decoded with SCLD.

Finally, Fig. 5 shows the code rate R_{SCLD} of the rate matched codes designed using Algorithm 1 plotted against the code rate of the code optimized for SCD. The curves are plotted for different code lengths. Interestingly, the change of R_{SCLD} against R_{SCD} is approximately linear for all code lengths.

VI. Conclusion

In this letter, we proposed a simple method for designing polar code for SC decoding based on throughput maximization for Type-I HARQ. The numerical results shows the codes constructed using this method perform very close to the capacity. Furthermore, we proposed an algorithm for matching the rate

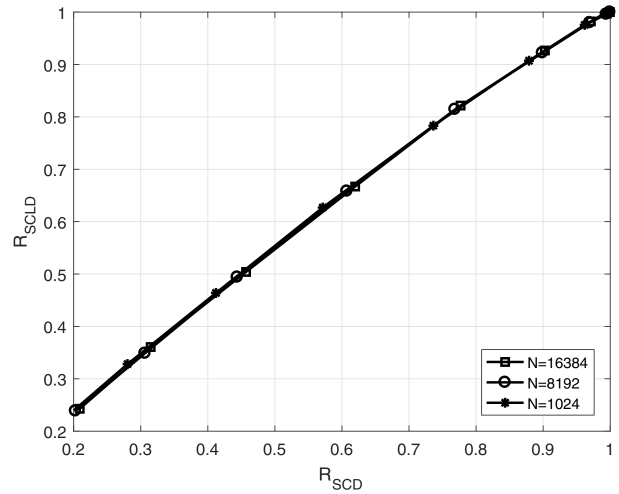

Fig. 5. Comparison of code rate R_{SCLD} found using Algorithm 1 against R_{SCD} for different code lengths.

of the codes designed for SCD to SCLD. The results indicate a substantial improvement when the proposed rate matching algorithm is used to find the optimum rate for SCLD. Due to the substantial throughput improvement that can be realized using carefully designed codes, the idea of throughput-based polar code design should be extended to designing optimal codes for Type-II HARQ systems in the future.

Acknowledgment

This work is supported in part by Huawei Canada Co., Ltd., and in part by the Ontario Ministry of Economic Development and Innovations Ontario Research Fund - Research Excellence (ORF-RE) program.

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. JC. 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. Inform. Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.

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

[6] *LTE; Evolved universal terrestrial radio access (E-UTRA); Medium access control (MAC) protocol specification*, 3GPP TS 36.321, version 10.1.0, release 10, Mar. 2012.

[7] 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. 1995.

[8] *LTE; Evolved universal terrestrial radio access (E-UTRA); Multiplexing and channel coding*, 3GPP TS 36.212, version 12.2.0, release 12, Oct. 2014.

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

[10] P. R. Balogun, I. Marsland, R. Gohary, and H. Yanikomeroglu, "Polar codes for noncoherent MIMO signalling," in *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016.

[11] R. A. Thisted, "Nonliear statistical methods," in *Elements of Statistical Computing: Numerical Computation*, USA: Chapman and Hall/CRC, 1988.