Indonesian Journal of Electrical Engineering and Computer Science
Vol. 26, No. 1, April 2022, pp. 521~530
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v26.i1.pp521-530

Turbo polar code based on soft-cancelation algorithm

Wallaa Yaseen Alebady[1,2], Ahmed Abdulkadhim Hamad[1]
[1]Department of Electrical Engineering, College of Engineering, University of Babylon, Babylon, Iraq
[2]Department of Medical Instrumentation Technologies Engineering, Hilla University College, Babylon, Iraq

Article Info

Article history:

Received Nov 29, 2021
Revised Jan 20, 2022
Accepted Feb 5, 2022

Keywords:

PCSTPC
Polar codes
Soft cancellation
Systematic polar codes
Turbo polar codes

ABSTRACT

Since the first polar code of Arikan, the research field of polar codes has been continuously active. Improving the performance of finite-code-length polar codes is the central point of this field. In this paper, the parallel concatenated systematic turbo polar code (PCSTPC) model has been proposed to improve the polar codes performance in a finite-length regime. On the encoder side, two systematic polar encoders are used as constituent encoders. While on the decoder side, two single iteration soft-cancelation (SCAN) decoders are used as soft-in-soft-out (SISO) algorithms inside the iterative decoding algorithm of the parallel concatenated systematic turbo polar code (PCSTPC). As compared to the optimized turbo polar code with SCAN and BP decoders, the proposed model has about 0.2 dB and 0.48 dB gains at BER=$10^{(-4)}$, respectively, in addition to 0.1 dB, 0.31 dB, and 0.72 dB gains over the TPC-SSCL32, TPC-SSCL16, and TPC-SSCL8 models, respectively. Moreover, the proposed model offers less complexity in comparison with other models, therefore requiring less memory and time resources.

Corresponding Author:

Wallaa Yaseen Alebady
Department of Electrical Engineering, College of Engineering, University of Babylon
Babylon, Iraq
Email: walaa.lafta@student.uobabylon.edu.iq

1. INTRODUCTION

Polar codes have changed the performance equation in the coding theory since the first polar-code of Arikan in 2009 [1]. The attention of most researchers in the coding theory field has been focused on the polar codes. The successive cancellation decoder (SC) is the first polar decoding algorithm that is proposed by Arikan in [1]. Since the polar code with the SC decoder is the first capacity-achieving code at the infinity code length. The performance of polar codes deteriorates at the finite code length and small SNR, this paper presents a turbo polar code to improve the performance of polar codes, especially at the finite-code-length and small SNR. There are several decoding algorithms suggested to improve the performance of polar codes at small and moderate code length.

Tal and Vardy [2] propose the successive cancellation list (SCL) decoder. The SCL decoder offers excellent performance, especially when coupling the CRC code as a genie-code along with it [3]-[5]. At a long and moderate code length, the complexity of SCL is very high. Niu and Chen [6] propose the successive cancellation stack (SCS) to reduce the high complexity of the SCL decoder. The belief propagation (BP) decoder can also be used as a decoding algorithm for polar codes [7]. Fayyaz [8] proposes the soft successive cancellation (SCAN) decoder as a soft-in-soft-out (SISO) decoding algorithm. The other soft-in-soft-out algorithm is the soft successive cancellation list (SSCL) decoder, which offers a higher complexity as compared to the SCAN decoder [9]. Among the above decoding algorithms, only the BP, SCAN, and SSCL algorithms can be used as SISO algorithms with the turbo iterative decoding process.

Turbo-polar codes (TPCs) use the concatenation structure and the iterative decoding algorithm to improve the performance of the classic systematic polar codes (SPCs). The works of [9], [10] present turbo polar codes with SSCL and BP decoders as iterative decoding algorithms. Because of the large number of iterations and the list size of the BP and SSCL algorithms, respectively, the complexity of the previous turbo polar codes is very high. Therefore, the SCAN decoder is used as a SISO algorithm for the turbo iterative decoding algorithm [11], [12]. The results of these codes will be compared with the results of the present work later on in section 6.

The remainder of this paper is organized as follows: section 2 describes the systematic polar encoding and the successive cancellation decoder. In section 3, the soft successive cancellation (SCAN) decoder is explained. The structure of the concatenated turbo polar code is presented in section 4. Section 5 presents the complexity analysis of the presented turbo polar code and other codes. Finally, in sections 6 and 7, the simulation results and the conclusion are presented, respectively.

2. PRELIMINARIES

2.1. Systematic polar code encoder (SPC)

To improve the bit error rate (BER) performance of the polar codes, Arikan proposes the SPC, which offers a BER performance over the non-systematic polar codes [13]-[15]. The SPC offers the ability to separate the parity bits and the information bits after the encoding process. The original polar coder can be used to create the systematic polar codes (SPCs) by modifying the original expression [13], [16]-[18], as (1):

$$\mathrm{x} = \mathrm{u} * \mathrm{F}^{\otimes \mathrm{n}} = \mathrm{u} * \mathrm{G}$$
$$F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = first\ order\ kronecker\ matrix \quad (1)$$

where $\mathrm{G} = \mathrm{F}^{\otimes \mathrm{n}}$; $u, x \in \mathbb{F}^N$, $G \in \mathbb{F}^{N \times N}$, and $\mathrm{F}^{\otimes \mathrm{n}}$ is the nth Kronecker power of F. For designing polar codes, $\mathbb{F}$ must be fixed at the binary field.

The binary vector u consists of two parts: $u = (u_{f^c}, u_f)$ for $f^c \subset (1, 2, \ldots, N)$. The first part (u_{f^c}) carries the user information, while the second part (u_f) carries the frozen bits that are fixed in the encoding process and are known at the decoder. Thus, the encoded bits in (1) can be separated as:

$$x = u_{f^c} G_{f^c} + u_f\, G_f \quad (2)$$

where $f = (1,2,3, \ldots, N) \backslash f^c$. G_{f^c} is a submatrix of G comprising of rows that correspond to indices f^c and G_f are the same G_{f^c} but with indices f. By selecting the length of the set f^c, the code rate can be adjusted as $R = K/N$, where K is the length of set f^c. For systematic polar codes (SPCs), the code in (2) can be rewritten as:

$$x_A = u_{f^c} * G_{f^c\,A} + u_f\, G_{f\ A} \quad (3)$$

$$x_B = u_{f^c} * G_{f^c\,B} + u_f *\ G_{f\ B} \quad (4)$$

where A and B correspond to indices of user and parity bits in SPC, respectively. $G_{f^c\,A}$ is the sub-matrix of G comprising of elements $(G_{a,b})$ with $a \in\ f^c$ and $b \in A$, and likewise for the other sub-matrices.

For Systematic Polar Codes (SPCs), x_A corresponds to u_{f^c} and u_f is fixed as frozen bits known as zero vectors. So, the u_{f^c} of systematic code can be calculated through (3) as (5):

$$u_{f^c} = \left(x_A - u_f\, G_{f\ A}\right) * G_{f^c A}^{-1}\ = (x_A) * G_{f^c A}^{-1} \quad (5)$$

by substituting (5) in (4), the parity bits (x_B) of SPC can be calculated as (6).

$$x_B = \left[(x_A) * G_{f^c A}^{-1}\right] * G_{f^c\,B} \quad (6)$$

From (5) and (6), the conclusion can be drawn that the systematic polar code can be achieved if (only if) f^c and A have the same number of indices, and $G_{f^c\,A}$ is an invertible matrix. Thus, (5) and (6) can be rewritten as (7) and (8).

$$u_{f^c} = (x_{f^c}) * G^{-1}_{f^c f^c} \quad (7)$$

$$x_f = [(x_{f^c}) * G^{-1}_{f^c f^c}] * G_{f^c f} \quad (8)$$

2.2. Successive cancelation (SC) decoder

As the native decoding algorithm, Arikan proposed the SC decoder for the polar codes [1]. For polar code (N, K), a decoding tree of an n-depth can be used to illustrate the decoding process, where $n = log_2 N$. Figures 1 and 2 show the decoder structure and 3-depth tree for polar code (8,4), respectively. In decoding the tree and at t-stage, there are 2^{n-t} nodes. Each node receives and propagates $\alpha_i^t and\, \beta_i^t$ vectors, respectively, from and to the parent node, where $0 \le t \le n$. The received soft information α_i^t and the propagated hard information β_i^t consist of 2^t values, as shown in Figure 2 (e.g. the first node of the second stage receives four LLRs $\alpha_{0:3}^2$ $[\alpha_0^2, \alpha_1^2, \alpha_2^2, \alpha_3^2]$ from the root node and propagates four binary bits $\beta_{0:3}^2$ $[\beta_0^2, \beta_1^2, \beta_2^2, \beta_3^2]$). To compute the LLRs α_i^{t-1} of the left child of the t-node, the following expression must be applied:

$$\alpha_l^{t-1} = f(\alpha_l^t, \alpha_{l+2^{t-1}}^t) \quad (9)$$

while the LLRs α_i^{t-1} of the right child are computed after estimating the hard information β_i^{t-1} of the left child as (10):

$$\alpha_l^{t-1} = g(\alpha_{l-2^{t-1}}^t, \alpha_l^t, \beta_{l-2^{t-1}}^{t-1}) \quad (10)$$

where l is a vector index ranging from $to\ 2^t$.

The message bits $\widehat{m}_i$ are estimated based on the values of LLRs α_i^0, where these LLRs can be computed as (11):

$$\alpha_i^0 = \log\left(\frac{Pr(R, \widehat{m}_{0:i-1} | m_i = 0)}{Pr(R, \widehat{m}_{0:i-1} | m_i = 1)}\right) \quad (11)$$

where R is the received log-likelihood ratio (LLRs) vector of the transmitted codewords through a channel. Based on the following rule, the message bits $\widehat{m}_i$ can be computed as (12).

$$\widehat{m}_i = \begin{cases} 0 & if \alpha_i^0 \ge 0 \\ 1 & otherwise \end{cases} \quad (12)$$

The estimated value of the frozen bit is $\widehat{m}_i$=0, regardless of the value of the α_i^0. The feedback hard information β_i^t can be computed by a linear combination of some decoded hard bits that are previously estimated as:

$$\beta_l^{t-1} = \beta_l^{t-2} \oplus \beta_{l+2^{(t-1)-1}}^{t-2} \quad (13)$$

$$\beta_{l+2^{(t-1)-1}}^{t-1} = \beta_{l+2^{(t-1)-1}}^{t-2} \quad (14)$$

where $\beta_l^0 = \widehat{m}_i$. The f and g functions are defined as (15) and (16) [19].

$$f(a, b) = sign(a * b) * \min(|a|, |b|) \quad (15)$$

$$g(a, b, \beta) = [(-1)^{1-2\beta} * a] + b \quad (16)$$

For the systematic polar codes, the decoding process needs an extra step to obtain the systematic information bits. To estimate the information bits, the outcomes of the decoder ($\widehat{m_l}$) must be re-encoded (i.e., multiplied by generator matrix [G]) to obtain the systematic information ($\hat{\beta}_i^n$). This extra step can be avoided by propagating the hard decisions ($\widehat{m_l}$) in the decoding tree [20].

After the last iteration of the SC decoder, the resulting matrices α and β contain information about noisy received LLRs, recovered encoded bits, and estimated message bits. $\beta(:,1)$ contains hard reversed-indexed information for the recovered encoded bits, whereas $\alpha(:, n+1)$ contains LLRs for message bits that correspond to the hard decisions at $\beta(:, n+1)$. Noisy received LLRs at $\alpha(:,1)$ are not updated throughout the decoding process.

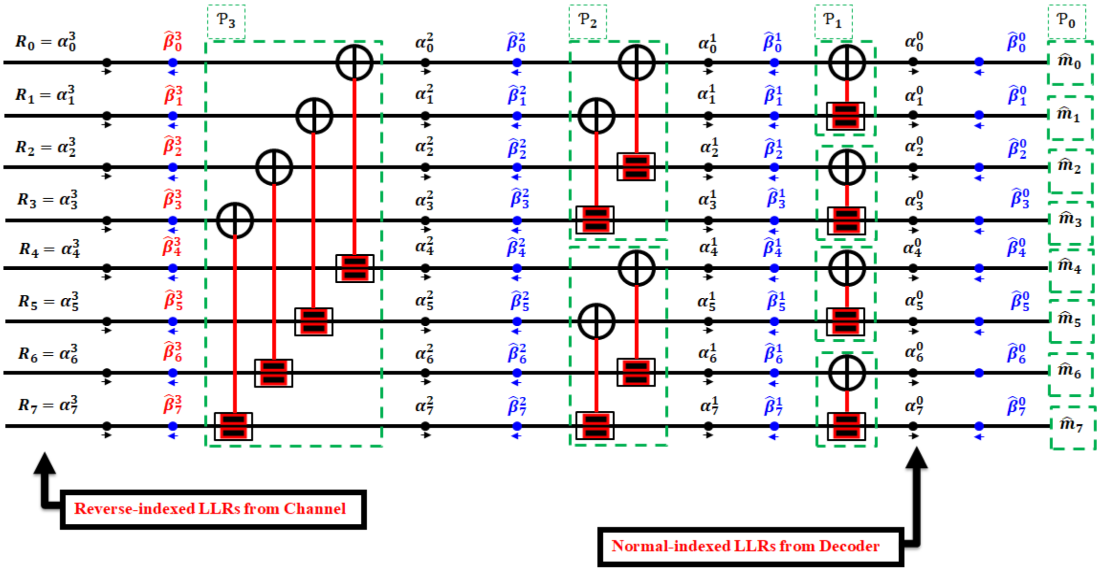


Figure 1. Factor graph of polar code with N=8

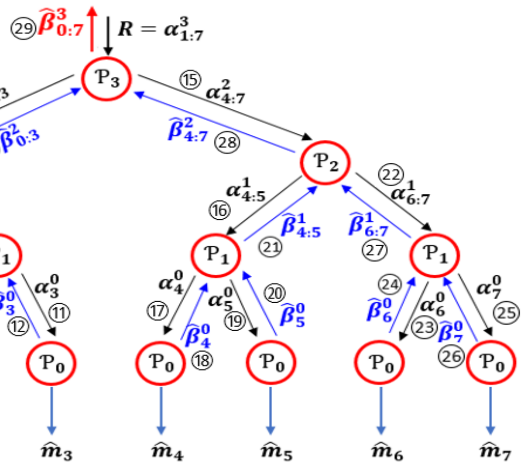


Figure 2. Decoding tree of polar code with N=8

3. SOFT-CANCELATION DECODER (SCAN)

Based on the SC's schedule, the SCAN decoder massively reduces the decoding complexity, as compared to the flooding schedule of the belief-propagation (BP) decoder [8]. In the SCAN algorithm, the soft information is propagated in both directions (i.e., from right to left and verse versa). Although the propagation of soft information in both directions somewhat increases the complexity and latency in comparison with the SC decoder, it improves the performance of the decoding process. The polar factor graph in Figure 1 of the SC decoder can be used with the SCAN decoder. In the beginning, the received noisy LLRs from the channel are put in vector α_i^0 at left; where $0 \le i \le 2^n - 1$. The knowledge of frozen bits' locations can be exploited by the SCAN decoder to initialize the vector β_i^n by a priori information, where the frozen locations are filled by infinity because these locations correspond to zero bits. The locations of information may be zero or one equiprobably, so they are filled with zero LLRs as (17) [21]:

$$\beta_i^n = \begin{cases} \infty & if\ i \in f \\ 0 & otherwise \end{cases} \tag{17}$$

while computing α_i^n with $\alpha_{0:i-1}^n$ which has already been computed; α_j^t (where $0 \leq t \leq n, 0 \leq j \leq 2^n - 1$) is partially populated from top left to bottom right. This partially computed α matrix can be used as a priori information to update β from top right to bottom left. At the final step with fully populated α_j^t (when $t = 0$ and $j = 2^n - 1$), all nodes of the factor graph contain soft extrinsic information (LLRs) which correspond to the values of the fully populated β matrix. At the first iteration, all values of β are zeros except β_i^n, where $0 \leq i \leq 2^n - 1$, so there is no extrinsic information that can be used by the SCAN decoder. At the second iteration and other successive iterations, the SCAN decoder exploits the extrinsic information computed with previous iterations [8], [22].

The updating process can be described as follows based on the factor graph of the basic polar kernel in Figure 3 [23]:

$$\alpha_0^{t-1} = f(\alpha_0^t, \alpha_1^t + \beta_1^{t-1}),\ \alpha_1^{t-1} = \alpha_1^t + f(\alpha_0^t, \beta_0^{t-1})$$
$$\beta_0^t = f(\beta_0^{t-1}, \beta_1^{t-1} + \alpha_1^t),\ \beta_1^t = \beta_1^{t-1} + f(\beta_0^{t-1}, \alpha_0^t)$$

where f is the box-plus operator ($\boxplus$) as (18) [24], [25].

$$x \boxplus y \triangleq \frac{1+e^{x+y}}{e^x+e^y} = 2tanh^{-1}\left[\tanh\left(\frac{x}{2}\right) \times \tanh\left(\frac{y}{2}\right)\right] \quad (18)$$

The previous form of box-plus requires more memory resources and execution time, so a hardware-friendly approximation can be used instead of the previous equation as (19).

$$f(x,y) = x \boxplus y \cong sign(x * y) * \min(|x|, |y|) \quad (19)$$

From Figure 4, it can be stated that the systematic polar codes with the SCAN decoder have the same frame-error rate (FER) as the non-systematic polar codes. In terms of bit-error-rate (BER), the systematic polar codes offer a significant advantage over non-systematic codes, as shown in Figure 4.

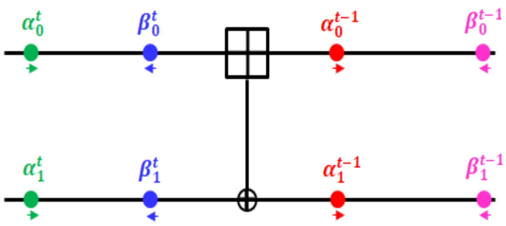


Figure 3. The factor graph of basic polar kernel

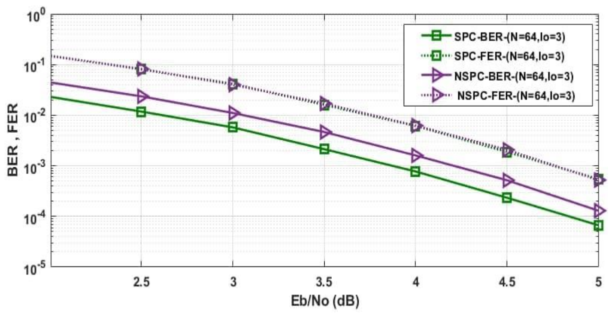


Figure 4. BER and FER comparisons between systematic and non-systematic PC (64,32)

4. TURBO CONCATENATED POLAR CODES

The construction of the turbo codes depends on several methods. However, the presented turbo polar code uses the traditional way that depends on two recursive and systematic codes at the encoder side. While at the decoder side, two iterative soft-in soft-out (SISO) decoders (such as SCAN, BP, and SSCL) are used.

4.1. Turbo encoder structure

Figure 5 shows the encoder's construction of turbo polar code. This construction includes two SPC encoders: the first encoder receives the original information bits (d) and generates the corresponding encoded bits, whereas the second encoder receives the interleaved data bits (d^π) and generates the corresponding encoded bits. The output of the construction shown in Figure 5 consists of three branches multiplexed with each other. The first branch carries the original user bits (d), the second branch conveys the parity bits of the first encoder (p^1), while the third line carries the parity bits of the second encoder (p^2). The multiplexed output of the concatenated encoder will be $c = [d\ p^1\ p^2]$, which is modulated by the BPSK modulator and transmitted through the AWGN channel.

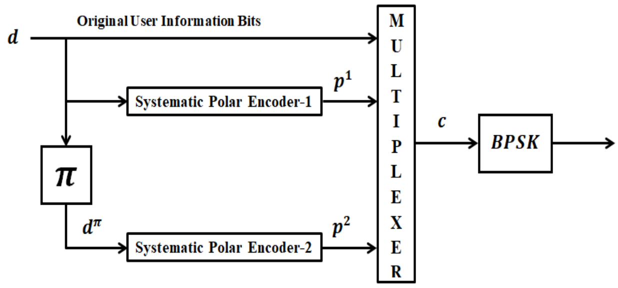


Figure 5. The structure of turbo polar encoder

4.2. Iterative decoding algorithm

The encoded information after being transmitted over the AWGN channel is decoded by several algorithms, successive cancellation (SC), successive cancellation list (SCL), successive cancellation stack (SCS), belief propagation (BP), soft successive cancellation list (SSCL), and soft cancellation (SCAN). However, most of these algorithms cannot be applicable with turbo polar codes (TPCs) because of their hard outputs (1 or 0). Only three of these algorithms contain the soft-in-soft-out (SISO) property and can therefore be used with turbo-polar codes, namely: belief propagation (BP), soft successive cancellation list (SSCL), soft cancellation (SCAN).

As shown in Figure 6, the SISO soft cancellation (SCAN) algorithm has been used with the proposed turbo polar code (TPC). The mechanism of the Iterative decoding algorithm can be summarized as follows:

a) The received LLRs from the channel are separated into three vectors, namely, $R_d, R_{p1}, and\ R_{p2}$, which are LLRs of the systematic bits, the parity bits of the first encoder and the parity bits of the second encoder, respectively.

b) R_d, R_{p1}, and the scaled prior information (φ_1^s) are the inputs of the SCAN-1 decoder that has one output (b_1). The internal procedure of SCAN-1 is shown in Figure 6 (The small red box on the top right). Based on the (f^c) and (f) sets, $\overline{R}_d$ is merged with R_{p1} and used to initialize the first column of α matrix of SCAN-1 decoder, where the vector ($\overline{R}_d$) corresponds to f^c indecies while the vector (R_{p1}) corresponds to f indecies.

c) The extrinsic LLR (ε_1) can be calculated as follows [9], [11]: $\varepsilon_1 = b_1 - \frac{2*R_d}{\sigma^2} - \varphi_1^s$, where σ^2 is the noise variance.

d) The extrinsic LLR (ε_1) is interleaved and passed to the SCAN-2 decoder as prior information (φ_2).

e) R_d^π, R_{p2}, and the scaled prior information (φ_2^s) are the inputs of the SCAN-2 decoder that has one output (b_2). The internal procedure of SCAN-2 is the same as SCAN-1 (see Figure 6: The red box).

f) The extrinsic LLR (ε_2) can be calculated as follows [9], [11]: $\varepsilon_2 = b_2 - \frac{2*R_s^\pi}{\sigma^2} - \varphi_2^s$.

g) The extrinsic LLR (ε_2) is de-interleaved and passed to of the SCAN-1 decoder as prior information (φ_1).

h) After the last outer-iteration, the estimated bits ($\hat{d}$) are obtained by applying the hard decision to the output of the SCAN-2 decoder (b_2) and then de-interleaving it.

The stopping mechanism of the iterative decoding process can be developed based on two criteria. The first one depends on the number of outer iterations: after the last iteration, the iterative decoding process is finished. The second criterion depends on a particular rule. If this rule is satisfied, the iterative decoding process is terminated. Immediately after each iteration, the estimated vector ($\hat{x}$) is calculated to be used for estimating the vector ($\hat{u}$), as [12]:

$$\hat{u} = \hat{x} * [G_N]^{-1} = \hat{x} * G_N$$

The frozen bits of the vector (u) are fixed with polar codes and known at the receiver, so these bits with the frozen bits of the vector ($\hat{u}$) can be used to satisfy the second criterion of the stopping mechanism through the:

$$S = u(f) + \hat{u}(f)$$

if the S factor equals zero, the iterative decoding process stops. Otherwise, the decoding process continues [12].

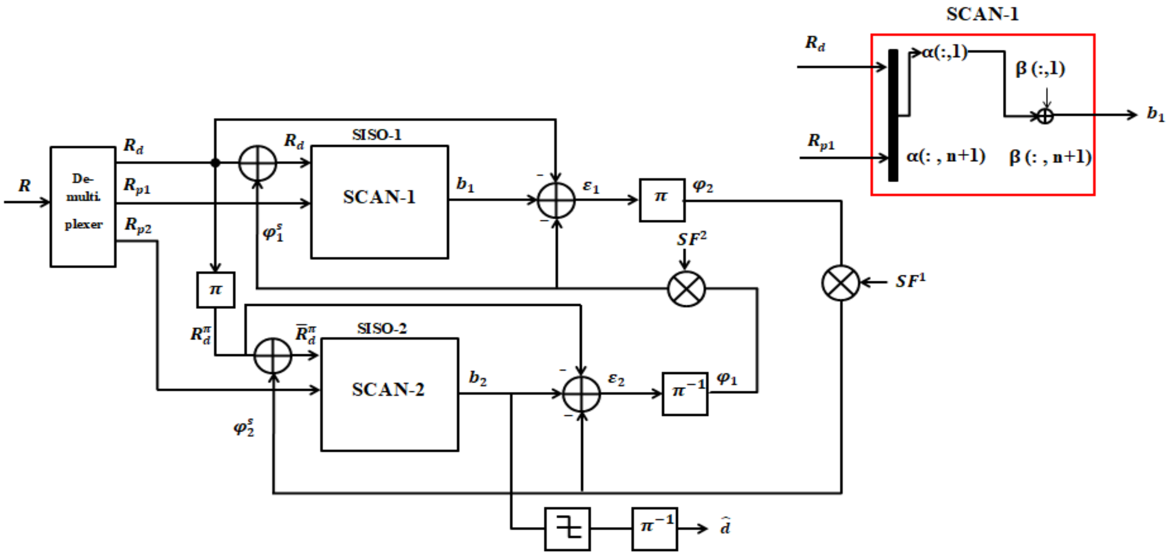


Figure 6. Iterative decoding of the proposed model

5. SCAN ITERATIVE DECODING COMPLEXITY ANALYSIS

As stated previously, only the SCAN, BP and SSCL decoders can be used with the turbo iterative decoding process; so, in this section, the complexity of these decoders is described and compared. A large amount of complexity is consumed by the SISO-1 and SISO-2 decoders, while the other components consume less complexity compared to the complexity of SISO-1 and SISO-2 decoders. As a result, it is possible to compute the complexity analysis of iterative decoding without taking into account the complexity of these components.

The complexity of a single SCAN decoder is $O(I_{s_i} * NlogN)$; where $I_{s_i}$is the number of inner SCAN iterations. Thus, the complexity of a turbo iterative decoding based on a SCAN decoder is twice the complexity of a single SCAN decoder if the outer iteration equals one ($I_{outer} = 1$). The general formula of the SCAN iterative decoding process is: $I_{outer} * (2 * I_{s_i} * NlogN)$.

Table 1 shows the total complexity formulas of the iterative decoding with SCAN, BP and SSCL decoders. $I_{s_i}$and I_{B_i} represent the inner iterations numbers of the SCAN and BP decoders, respectively, whereas L_{list} is the list size of the soft successive cancellation list (SSCL) decoder. The BP decoder requires sixty inner iterations ($i.e.\ I_{B_i} = 60$) to approach the performance of the SCAN decoder when working with only two inner iterations (i.e. $I_{s_i} = 2$). Obviously, the inner iterations of the SCAN decoder are smaller than the inner iterations of the BP decoder. For the SSCL decoder, the list size (L_{list}) is always 32 or 64 [2]. It can therefore be stated that the complexity of turbo iterative decoding with SCAN decoder is smaller than the others.

Table 1. Complexity formulas of different SISO decoders

Decoder Scheme	Complexity Formula
SCAN	$I_{outer} * (2 * I_{s_i} * NlogN)$.
BP	$I_{outer} * (2 * I_{B_i} * NlogN)$
SSCL	$I_{outer} * (2 * L_{list} * NlogN)$

6. RESULTS AND DISCUSSION

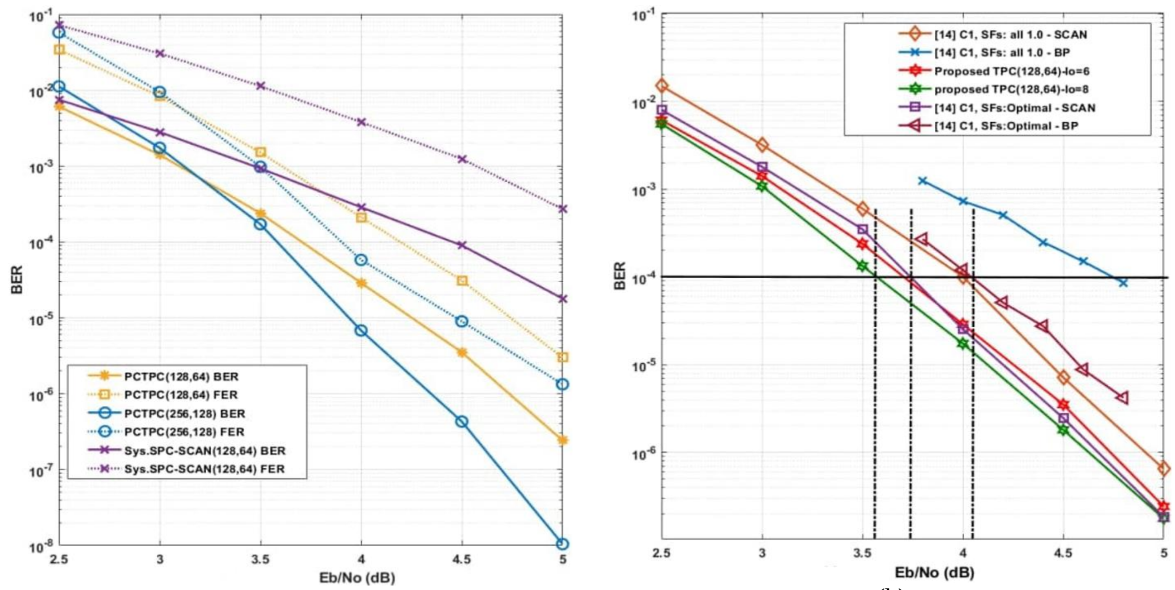
In this section, the simulation results (Figure 7) of the proposed model are presented in detail and compared with the results of other models shown in [9], [11], [14]. All results in this section are carried out over the AWGN channel and with a single-iteration SCAN algorithm. The maximum number of simulation frames varies according to the SNRs, with the upper limit of the frame errors being 1000 errors.

Figure 7(a) shows the performance of the proposed scheme in terms of bit error rate (BER) and frame error rate (FER). These results are determined with the following parameters: i) $I_{outer} = 6$, ii) $I_{SCAN} = 1$, iii) SF^1 = [0.63], and iv) SF^2 = [0.75].

From Figure 7(a), it can be seen that the PCSTPC with a (128.64) SPC offers a more reliable performance compared to standalone (128.64) SPC with a SCAN algorithm. At BER=10^{-4}, the proposed

PCSTPC (128.64) has a 0.77 dB gain over the standalone (128.64) SPC. These results prove the theory proposed in [12], which states that “at finite code lengths, the performance of polar codes can be improved by using the turbo polar codes (TPCs)” [12].

Figure 7(b) compares the BER performances of the proposed model and the optimized model in [14]. The optimized model uses two decoder algorithms: SCAN and BP decoders. At BER=10^{-4}, the proposed scheme (with eight outer-iterations) has approximately 0.2 dB and 0.48 dB gains over the optimized models of SCAN and BP in [14]; respectively. At six outer-iterations, the BER-performance of the proposed and the optimized SCAN models approaches with each other. However, the proposed model offers an approximate gain of 0.35 dB over the optimal BP model (which has three outer-iterations and 60 inner-iterations).



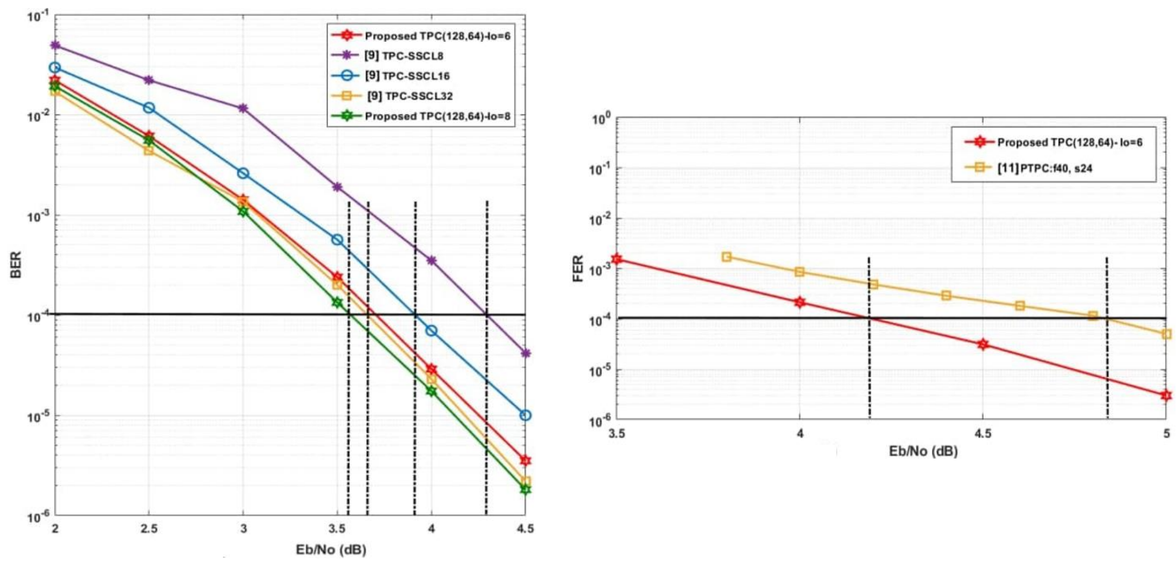
(a) (b)

Figure 7. Simulation results of (a) Comparison between the proposed PCSTPC and SPC-SCAN and (b) Comparison between the proposed PCSTPC and the optimized model [14]

Figure 8(a) illustrates a comparison between the proposed model and the TPC-SSCL model (presented in [9]) in terms of BER performance. At BER=10^{-4}, the proposed TPC offers 0.1 dB, 0.31 dB, and 0.72 dB gains over the TPC-SSCL32, TPC-SSCL16, and TPC-SSCL8 models, respectively. The proposed model with six outer-iterations also offers a good performance as compared to the TPC-SSCL models.

In terms of FER, the performance of the proposed model is compared with the performance of the punctured turbo polar code (PTPC) model in [11] with the same (128.64) SPC component code. The proposed scheme has approximately 0.72 dB gain over the PTPC model, as shown in Figure 8(b). The parameters of all previous models are presented in Table 2.

Finally, the proposed model has a superior advantage over all previous models in terms of complexity. The low complexity of the SCAN decoder and the low inner iterations (just single iteration) make the proposed model require less hardware and memory resources to be implemented practically. Although the TPC-SSCL32 model has a high degree of complexity, the proposed algorithm requires only eight outer-iterations instead of six to exceed the BER performance of the TPC-SSCL32 model, as shown in Figure 8(a). The mathematical expressions of complexity are shown in Table 2 for each model. These expressions are very crucial to determine the memory and time requirements for implementing any model.



(a) (b)

Figure 8. Simulation results of (a) Comparison between the proposed model and the TPC-SSCL model [9] and (b) FER comparison between the proposed PCSTPC and the PTPC model in [11]

Table 2. Summary of the studied models' parameters

Features	[9] Model	[14] Model	[11] Model	Proposed Model
Enco. Comp.	SPC and SPC	SPC and SPC	SPC and SPC	SPC and SPC
SISO-1	SSCL	BP or SCAN	SCAN	SCAN
SISO-2	SSCL	BP or SCAN	SCAN	SCAN
Interleaver	Random	Random	Random	LTE interleaver
Puncturing	×	×	✓	×
$I_{BP,SCAN,or\ list}$	$I_{list} = 8, 16, 32, 64$	$I_{BP} = 60,\ I_{SCAN} = ?$	$I_{SCAN} = 4$	$I_{SCAN} = 1$
I_{outer}	6	3, 6	18	6, 8
Complexity	$O[2I_{outer} * (I_{list} * N\log N)]$ From **Medium** (with $I_{list} = 8$) to **Very High** (with $I_{list} = 64$)	$O[2I_{outer} * (I_{BP\ or\ SCAN} * N\log N)]$ (BP→**Very High**) (SCAN→**Medium**)	$O[2I_{outer} * (I_{SCAN} * N\log N)]$ **High**	$O[2I_{outer} * (I_{SCAN} * N\log N)]$ **Low**

Notes:
R_p: Rate of SPC $= K/N$
I_{BP}: Inner iterations number of BP decoder
I_{SCAN}: Inner iterations number of SCAN decoder
I_{list}: List size of SSCL decoder
I_{outer}: Number of iterations between the SISO algorithms

7. CONCLUSION

The turbo polar code based on SCAN algorithm is introduced in this paper to improve the performance of the original polar code at a finite code length regime. The simulation results show that the presented model offers an excellent performance at finite-code lengths with low complexity as a result of using the single iteration SCAN algorithm. As a SISO decoder, the single iteration SCAN algorithm works perfectly with the iterative decoding algorithm, as each SCAN decoder helps the other to improve its performance. Therefore, the overall performance of the proposed TPC exceeds that of a standalone SPC with the SCAN decoder. From the simulation results, the conclusion can be drawn that the turbo polar codes (TPCs) have overcome the error floor problem, which is one of the major disadvantages in the turbo codes.

REFERENCES

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," in *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009, doi: 10.1109/TIT.2009.2021379.

[2] I. Tal and A. Vardy, "List decoding of polar codes," in *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015, doi: 10.1109/TIT.2015.2410251.

[3] W. K. Abdulwahab and A. A. Kadhim, "Internal pilot insertion for polar codes," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 1495–1504, 2021, doi: 10.11591/ijeecs.v22.i3.pp1495-1504.

[4] T. Murata and H. Ochiai, "Performance analysis of CRC codes for systematic and nonsystematic polar codes with list decoding," *Wirel. Commun. Mob. Comput.*, vol. 2018, 2018, doi: 10.1155/2018/7286909.

[5] K. Niu and K. Chen, “CRC-aided decoding of polar codes,” *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, 2012, doi: 10.1109/LCOMM.2012.090312.121501.
[6] K. Niu and K. Chen, “Stack decoding of polar codes,” *Electron. Lett.*, vol. 48, no. 12, pp. 695–697, 2012, doi: 10.1049/el.2012.1459.
[7] E. Arıkan, “Polar codes: A pipelined implementation,” *Proc. 4th Int. Symp. Broad. Commun. ISBC 2010*, 2010, pp. 11–14.
[8] U. U. Fayyaz and J. R. Barry, "Low-complexity soft-output decoding of polar codes," in *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 958-966, May 2014, doi: 10.1109/JSAC.2014.140515.
[9] Z. Liu, K. Niu and J. Lin, "Parallel concatenated systematic polar code based on soft successive cancellation list decoding," *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2017, pp. 181-184, doi: 10.1109/WPMC.2017.8301804.
[10] D. Wu, A. Liu, Y. Zhang, and Q. Zhang, “Parallel concatenated systematic polar codes,” *IEEE Electronics Letters*, vol. 52, no. 1, pp. 43–45, 2016.
[11] Y. Qi, D. Chen, and C. Zhang, "Punctured turbo-polar codes," *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*, 2019, pp. 736-741, doi: 10.1109/ICCSN.2019.8905352.
[12] Q. Zhang, A. Liu, Y. Zhang, and X. Liang, "Practical design and decoding of parallel concatenated structure for systematic polar codes," in *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 456-466, Feb. 2016, doi: 10.1109/TCOMM.2015.2502246.
[13] E. Arikan, "Systematic polar coding," in *IEEE Communications Letters*, vol. 15, no. 8, pp. 860-862, August 2011, doi: 10.1109/LCOMM.2011.061611.110862.
[14] Z. Liu, K. Niu, J. Lin, J. Sun, and H. Guan, "Scaling factor optimization of turbo-polar iterative decoding," in *China Communications*, vol. 15, no. 6, pp. 169-177, Jun. 2018, doi: 10.1109/CC.2018.8398513.
[15] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, “Fast polar decoders: Algorithm and implementation,” *IEEE J. Sel. Areas Commun*., vol. 32, no. 5, pp. 946–957, 2014, doi: 10.1109/JSAC.2014.140514.
[16] R. Benkhouya, I. Chana, and Y. Hadi, “Study of the operational SNR while constructing polar codes,” *Int. J. Electr. Comput. Eng.*, vol. 10, no. 3, pp. 3200–3207, 2020, doi: 10.11591/ijece.v10i3.pp3200-3207.
[17] K. Niu, K. Chen, and J. R. Lin, “Beyond turbo codes: Rate-compatible punctured polar codes,” *IEEE Int. Conf. Commun.*, pp. 3423–3427, 2013, doi: 10.1109/ICC.2013.6655078.
[18] Q. Jan, S. Hussain, M. Furqan, Z. Pan, N. Liu, and X. You, “A novel flip-list-enabled belief propagation decoder for polar codes,” *Electron*., vol. 10, no. 18, pp. 1–15, 2021, doi: 10.3390/electronics10182302.
[19] A. Mishra *et al.*, “A successive cancellation decoder ASIC for a 1024-bit polar code in 180nm CMOS,” *Proc.-2012 IEEE Asian Solid-State Circuits Conf. A-SSCC*, no. 2, 2012, pp. 205–208, doi: 10.1109/IPEC.2012.6522661.
[20] B. Le Gal, C. Leroux, and C. Jego, “Multi-gb/s software decoding of polar codes,” *IEEE Trans. Signal Process*., vol. 63, no. 2, pp. 349–359, 2015, doi: 10.1109/TSP.2014.2371781.
[21] C. Pillet, C. Condo, and V. Bioglio, “SCAN list decoding of polar codes,” *IEEE Int. Conf. Commun*., vol. 2020-June, 2020, doi: 10.1109/ICC40277.2020.9149202.
[22] L. Zhang, Y. Sun, Y. Shen, W. Song, X. You, and C. Zhang, “Efficient fast-scan flip decoder for polar codes,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2021-May, 2021, doi: 10.1109/ISCAS51556.2021.9401398.
[23] J. Tong, H. Zhang, X. Wang, S. Dai, R. Li, and J. Wang, “A soft cancellation decoder for parity-check polar codes,” *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC*, vol. 2020-Augus, 2020, doi: 10.1109/PIMRC48278.2020.9217144.
[24] A. Alamdar-Yazdi and F. R. Kschischang, “A simplified successive-cancellation decoder for polar codes,” *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, 2011, doi: 10.1109/LCOMM.2011.101811.111480.
[25] U. U. Fayyaz and J. R. Barry, “Polar codes for partial response channels,” *IEEE Int. Conf. Commun*., pp. 4337–4341, 2013, doi: 10.1109/ICC.2013.6655247.

BIOGRAPHIES OF AUTHORS

Wallaa Yaseen Alebady was born in Karbala, Iraq, in 1993. He received his B.Sc. degree in Electrical Engineering from the University of Babylon, Babylon, Iraq in 2015, and M.Sc. degrees also from the University of Babylon, Babylon, Iraq in 2018, in Electronics and Communication Engineering. He is now Ph.D candidate in the Babylon University/ Electronics and Communication Engineering. His research interests include Internet of things (IoT), WSN, Computer applications, Information theory and coding, and polar codes. He can be contacted at email: wallaa.alebady@yahoo.com.

Ahmed Abdulkadhim Hamed is Assistant Professor received BSc degree in Electrical and Electronic Engineering/Avionics Engineering from the Military Engineering College, Baghdad, Iraq in 1991, MSc degree in Communication and Radar in 1995. Hamad received his PhD in communication engineering from University of Technology, Baghdad, Iraq, 2007. He is a teaching staff in Babylon University, Babylon, Iraq since 2006 till now. The area of research interest includes digital communication, information theory and coding, cryptography, and FPGA applications. He can be contacted at email: eng.ahmed.ak@uobabylon.edu.iq.