A Low Latency SCAN-Flip Polar Decoder for 5G Vehicular Communication

Yu Wang(✉), Lirui Chen, Shikai Qiu, Li Huang, and Zuocheng Xing

National University of Defense Technology, Changsha, China
{wangyu16,chenlirui14,qiushikai17,huangli16,zcxing}@nudt.edu.cn

Abstract. Polar codes are widely considered as one of the most promising channel codes for future wireless communication. However, at short or moderate block lengths, their error-correction performance under traditional successive cancellation (SC) decoding is inferior to other modern channel codes, while under list decoding outperforms at the cost of high complexity and long latency. Successive cancellation flip (SCF) decoding is shown having competitive performance compared to that of list decoding but suffers from a long decoding latency. In this work, we propose the SCAN-Flip decoding algorithm by introducing the flipping idea into soft cancellation (SCAN) decoding. The proposed algorithm improves the error-correction performance of soft cancellation decoding and accelerates the convergence of iterative calculation, leading to lower execution-time. Besides, we also propose a new path metric to improve the performance of our SCAN-Flip decoder further. Simulation results show that the proposed decoder has a much smaller average number of iterations than that of SCF at equivalent frame error rate. At equivalent max number of iterations, the error-correction performance of SCAN-Flip outperforms SC-Flip by up to 0.25 dB at bit error rate of 10^{-4}.

Keywords: Successive cancellation flip · Soft cancellation · Belief propagation · Low latency · Polar codes

1 Introduction

Recently vehicle-to-vehicle (V2V) communication is widely considered as one of the most promising technology for intelligent transportation systems (ITSs) to ensure traffic safety [2]. However, its requirements of ultra reliability and low latency pose significant challenges for physical layer design. As a key enabling technology, channel coding has significant influence on reliable transmission. In this domain, polar codes [4] are the first channel codes that provably achieve the capacity of various communication channels and have been recently selected for the control channel in the 5G enhanced Mobile BroadBand (eMBB) scenario [12] to provide low latency and reliable communication. However, for polar

Supported by the NSF of China (Grant No. 61170083) and Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20114307110001).

J. C. Ferreira et al. (Eds.): INTSYS 2018, LNICST 267, pp. 157–170, 2019.
https://doi.org/10.1007/978-3-030-14757-0_12

codes at short to moderate block lengths, the error-correction performance under successive cancellation (SC) decoding is worse than the turbo or low-density parity-check (LDPC) codes. In order to improve the performance of the finite block length, many decoding methods, such as SC list (SCL) decoding [13] and SC stack (SCS) decoding [8], are introduced. Nonetheless, these methods suffer from higher computational complexity and longer decoding latency than that of the original SC decoding algorithm. On the other hand, in order to reduce the decoding latency, belief propagation (BP) decoding is proposed in [3], with parallel message propagating. However, its error-correction performance is worse than that of SC decoding.

As an alternative decoding method of SC, the soft-cancellation (SCAN) decoder proposed in [9] is a combination of SC decoder and BP decoder, based on a sequential message propagating schedule, which is similar to the SC decoding process. The SCAN decoder has better performance than SC and BP decoder. As another iterative decoder, the successive-cancellation flip (SCF) decoder proposed in [1] is shown to be capable of providing error-correction performance comparable to that of SCL decoder with a small list size, while keeps the complexity close to that of SCAN. The idea of SCF decoder is to allow a given number of new decoding attempts, where one or several positions are flipped in the sequential decoding. However, this decoding method suffers from a higher worst-case latency when choosing the wrong flipping position.

Contribution: In this work, we introduce the flipping idea of SCF decoder into SCAN decoder by initializing the β log-likelihood ratio (LLR) according to previous decoding pass. The simulation result shows that the new decoding algorithm has a better error-correction performance and lower decoding latency than that of the original SCF decoder. Besides, we propose a new path metric for our decoder to further improve the error-correction performance.

The remainder of this work is organized as follows: in Sect. 2, an overview of polar codes, SCAN decoding, and SCF decoding algorithms are presented. In Sect. 3, the SCAN-Flip decoding method is detailed, while Sect. 4 describes a modification on SCAN-Flip by introducing a new path metric to correct more erroneous bits. Section 5 reports the simulation results, and conclusions are drawn in Sect. 6.

2 Preliminary

2.1 Polar Codes

Polar codes characterized by $(N, K, \mathcal{I})$ belong to linear block codes, where $N = 2^n$ is the length of the polar code, K is the number of information bits, and a set $\mathcal{I}$ indicates the positions of the K information sub-channels. Polar codes can achieve channel capacity via the phenomenon of channel polarization [4]. The channel polarization theorem states that, as the code length N goes to infinity, a polarized subchannel becomes either a noiseless channel or a pure noise channel. By transmitting information bits over the reliable subchannels and transmitting frozen bits which are known by both transmitter and receiver over the unreliable subchannels, polar codes can achieve the capacity under an SC decoder. The encoding procedure of a polar code can be represented with a matrix multiplication like $\mathbf{x} = \mathbf{u}G_N$, where vector $\mathbf{u}$ means the source codeword containing information bits, while vector $\mathbf{x}$ means the encoded codeword and G_N is the generator matrix.

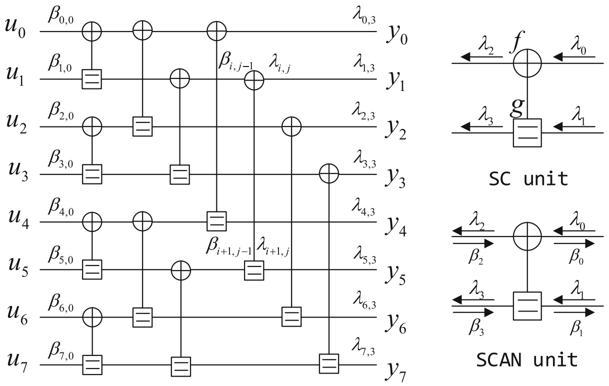

Fig. 1. Factor graph and message propagation mechanism for $N = 8$ polar code.

As for the decoding process, we denote by vector y_0^{N-1} the data received from the channel and use them as the decoder input. The output of decoder is denoted by vector $\hat{u}_0^{N-1}$, as shown in Fig. 1, where $\lambda_{i,3}$ denotes the LLR value of y_i. The decoding procedure of the SC decoding can be interpreted as an iterative procedure with a complexity of $\mathcal{O}(N\log N)$ for one decoding attempt. Let $\hat{u}_i$ denotes the estimate of the bit u_i at the final hard decision. Each estimate $\hat{u}_i$ is calculated according to the LLR value $\lambda_i = \log(\frac{\mathrm{Pr(\mathbf{y},\hat{u}_0^{i-1}|u_i=0)}}{\mathrm{Pr(\mathbf{y},\hat{u}_0^{i-1}|u_i=1)}})$ by using the hard decision function h:

$$\hat{u}_i = h(\lambda_i) = \begin{cases} u_i & \text{if } i \notin \mathcal{I} \\ \frac{1-\mathrm{sign}(\lambda_i)}{2} & \text{if } i \in \mathcal{I} \end{cases} \tag{1}$$

where by convention $\mathrm{sign}(\lambda_i) = \pm 1$. At the same time, the LLRs at different calculation stage l are computed iteratively as follows:

$$\lambda_{i,l} = \begin{cases} f(\lambda_{i,l+1}; \lambda_{i+2^l,l+1}) & \text{if } \frac{i}{2^l} \text{is even} \\ g(\hat{s}_{i-2^l,l}; \lambda_{i-2^l,l+1}; \lambda_{i,l+1}) & \text{otherwise} \end{cases} \tag{2}$$

where $\hat{s}$ denotes the partial sum of $\hat{u}_0^{i-1}$, which are the previously decoded bits from 0 to $i-1$. And in the LLR domain, the function f and g perform the following calculation for given inputs LLRs λ_a and λ_b.

$$f(\lambda_a, \lambda_b) = \log(\frac{e^{\lambda_a+\lambda_b}+1}{e^{\lambda_a}+e^{\lambda_b}}) \tag{3}$$

$$g(\lambda_a, \lambda_b, u_s) = (-1)^{u_s}\lambda_a + \lambda_b \tag{4}$$

2.2 Soft Cancellation Decoding

The SCAN decoding algorithm could be seen as a mixture of the SC and BP decoding algorithms [11]. Its operating schedule is similar to the SC decoder, while its message propagation is close to the BP decoder. The message propagation mechanism of SCAN decoding is illustrated in the factor graph Fig. 1. The left-propagating and right-propagating LLRs at row i and stage j is denoted by $\lambda_{i,j}$ and $\beta_{i,j}$, respectively. Compared with the original SC decoding, the introduction of $\beta_{i,j}$ propagating increases the efficiency of information dissemination in the decoding process.

In the factor graph, the LLR values $\lambda_{i,n}$ and $\beta_{i,0}$ do not update during the decoding process, since the $\lambda_{i,n}$ are the LLRs of received bits, while the $\beta_{i,0}$ are the LLR values initialized by the bit type such that:

$$\beta_{i,0} = \begin{cases} +\infty \text{ if } i \notin \mathcal{I} \\ \;\;0 \;\;\; \text{if } i \in \mathcal{I} \end{cases} \tag{5}$$

The message propagating of a unit factor graph is shown in the bottom right corner of Fig. 1. In the kth iteration, for a unit factor graph, the $\lambda_{i,j_0}^{(k)}$, $\lambda_{i,j_1}^{(k)}$, $\beta_{i+1,j_2}^{(k)}$ and $\beta_{i+1,j_3}^{(k)}$ are LLR values sent to the unit, while $\beta_{i,j_0}^{(k)}$, $\beta_{i,j_1}^{(k)}$, $\lambda_{i+1,j_2}^{(k)}$ and $\lambda_{i+1,j_3}^{(k)}$ are LLR values sent from the unit and can be computed as follows:

$$\lambda_{i+1,j_2}^{(k)} = f(\beta_{i+1,j_3}^{(k-1)} + \lambda_{i,j_1}^{(k)}, \lambda_{i,j_0}^{(k)}) \tag{6}$$

$$\lambda_{i+1,j_3}^{(k)} = f(\beta_{i+1,j_2}^{(k-1)}, \lambda_{i,j_0}^{(k)}) + \lambda_{i,j_1}^{(k)} \tag{7}$$

$$\beta_{i,j_0}^{(k)} = f(\beta_{i+1,j_2}^{(k)}, \beta_{i+1,j_3}^{(k)} + \lambda_{i,j_1}^{(k)}) \tag{8}$$

$$\beta_{i,j_1}^{(k)} = f(\beta_{i+1,j_2}^{(k)}, \lambda_{i,j_0}^{(k)}) + \beta_{i+1,j_3}^{(k)} \tag{9}$$

The function f in the above equations is the same as that in the SC decoder. After predetermined maximum T_{max} iterations, the estimation of $\hat{u}_i$ can be computed by Eq. 10, which is a little different from the hard decision function of SC decoder.

$$\hat{u}_i = h(L_i) = \begin{cases} u_i & \text{if } i \notin \mathcal{I} \\ \frac{1-\text{sign}(\lambda_{i,0}^T + \beta_{i,0}^T)}{2} & \text{if } i \in \mathcal{I} \end{cases} \tag{10}$$

2.3 Successive-Cancellation Flip Decoding

The SCF decoding algorithm is a slightly-modified SC decoding algorithm. The procedure of SCF decoding starts by going through a regular SC decoding pass. After the first decoding pass of SCF, the nested cyclic redundancy check (CRC) is verified and a flipping list of least reliable estimated bits is built. In case the CRC matches, the decoding procedure stops, and the estimated $\hat{u}_0^{N-1}$ is output. Otherwise, another SC-decoding pass is launched according to the flipping list. In this pass, once the location of the information bit that corresponds to the least reliable bit is reached, its estimated bit is flipped, and the subsequent positions are decoded by using the standard SC decoding. And when a decoding attempt finishes, the CRC is verified again. The nested CRC codes are concatenated with the information bits, and can be calculated on-the-fly. In this regard, the real coding rate for polar codes is $R = (K + L_{CRC})/N$, where L_{CRC} denotes the length of CRC codes.

The above procedure is repeated until the CRC check pass or a predetermined maximum number of decoding attempts is reached. Consequently, the SCF decoding keeps the computational complexity close to that of SC, while having error-correction performance close to that of SCL. It provides a tunable trade-off between the decoding performance and the decoding complexity. However, since the sequential nature of flipping list, the decoding throughput of SCF decoding is variable, and the average decoding latency depends on the channel condition.

3 SCAN-Flip Decoding

Different from the SC decoder, the SCAN decoder can use the information after $\hat{u}_i$ in the decoding procedure by $\beta_{i,j}$ propagating. Due to its efficient dissemination of information, the SCAN decoding algorithm has a lower bit error rate (BER) than SC decoding. However, considering the erroneous bit decisions caused by error propagation in SC decoder, the error propagation also affects SCAN decoder, which is caused by the calculation of $\beta_{i,j}$ LLRs in the message propagation. For these reasons, we introduce the flipping idea of SCF to SCAN decoding and propose the SCAN-Flip decoding algorithm.

3.1 SCAN-Flip Decoding Algorithm

The error propagation mechanism in SCAN decoder is different from that in SC decoder, whose previous erroneous estimated bits would affect subsequent estimation directly by partial sum $\hat{s}$ calculation. In SCAN decoder, the error information is propagated by the calculation of $\beta_{i,j}$ LLRs. Furthermore, the calculation of $\beta_{i,j}$ LLRs are affected by both initial value $\beta_{i,0}$ and $\lambda_{i,j}$ LLRs. Besides, it was proved in [9] that clipping the $\beta_{i,0}$ LLRs of already correct estimated bits to $+\infty, -\infty$ can accelerate the convergence of subsequent iterative calculation at the end of one iterative calculation.

Based on these points, we first propose SCAN-Flip decoding algorithm by introducing the flipping idea into SCAN decoder. It starts iterative decoding by performing a standard SCAN decoding. At the end of the first pass, we use the nested CRC code to check the result. If the CRC checking pass, the estimated codeword is assumed to be correct. If not, a second iteration is launched. In parallel with the first SCAN decoding pass, a set of low reliable estimated bits are stored and sorted. The second iteration starts from the least reliable one $\hat{u}_{i-least}$ in the set. In this iteration, the $\beta_{i,0}$ LLRs before the corresponding index *i-least* with the least reliable decision are initialized by $+\infty, -\infty$ according to the estimated bits got in the first pass, while the $\beta_{i-least,0}$ is set the opposite value. The remaining $\beta_{i,0}$ LLRs are set according to its bit type just like that in standard SCAN decoding.

$$\beta_{i,0} = \begin{cases} \frac{1-\text{sign}(\hat{u}_i)}{2} * \infty & \text{if } i < i_{least} \\ -\frac{1-\text{sign}(\hat{u}_i)}{2} * \infty & \text{if } i = i_{least} \\ +\infty & \text{if } i > i_{least} \;\&\; i \notin \mathcal{I} \\ 0 & \text{if } i > i_{least} \;\&\; i \in \mathcal{I} \end{cases} \tag{11}$$

The second SCAN iteration starts basing on this initialization and is followed by a CRC check. This procedure is repeated until either the CRC pass or a predetermined maximum number of iterations T_{max} is reached. The details of SCAN-Flip decoding process are described in Algorithm 1.

Algorithm 1. SCAN-Flip Decoding Algorithm

```
Input: the received vector $y_1^N$, maximum iteration $T_{max}$
Output: a decoded vector $\hat{u}_1^N$
Init$\beta_1(I)$
$(\hat{u}_1^N, \{\lambda_i^{(1)}\}_{i\in I}) \leftarrow$ SCAN$(y_1^N, \beta_1)$
if CRC($\hat{u}_1^N$)=success then return $\hat{u}_1^N$
else Init$\beta_2(\{\lambda_i^{(1)}\}_{i\in I})$
end if
for t = 2,...,$T_{max}$ do
    $(\hat{u}_1^N, \{\lambda_i^{(t)}\}_{i\in I}) \leftarrow$ SCAN$(y_1^N, \beta_t)$
    if CRC($\hat{u}_1^N$)=success then return $\hat{u}_1^N$
    else Init$\beta_{t+1}(\{\lambda_i^{(1)}\}_{i\in I})$
    end if
end for
return $\hat{u}_1^N$
```

3.2 Oracle-Assisted SCAN Decoder

In order to examine the potential benefits of modified initialization of $\beta_{i,0}$ LLRs, we employ an order-ω oracle-assisted SCAN (OA-SCAN) decoder, which can get the accurate index of ω erroneous bits in the estimated codeword. The $\beta_{i,0}$ LLRs before these indexes are initialized correctly.

As shown in Fig. 2 we compare the performance of the standard SCAN decoder with that of the oracle-assisted order-ω SCAN decoder for a polar code with $N = 1024$ and $R = 0.5$ over an AWGN channel. In Fig. 2, the ideal OA-SCAN decoder means the initialization of $\beta_{i,0}$ LLRs is wholly set according to u_0^{N-1}. The performance of OA-SCAN decoder with order-1 could be seen as the lower bound of SCAN-Flip decoder. From Fig. 2, we observe that the ideal oracle decoder can significantly improve the performance of the SCAN decoder, while

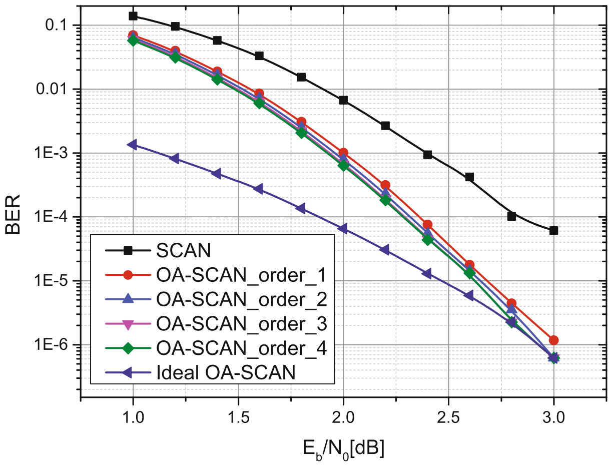

Fig. 2. BER performance of oracle-assisted SCAN decoders with different order compared to the SCAN decoder for $N = 1024$ and $R = 0.5$.

the OA-SCAN decoder with order-ω decoders perform inferior at low signal-to-noise ratio (SNR). Besides, the OA-SCAN decoders with different ω order have similar performance. Based on this, in the latter work, we focus on approaching the performance of order-1 OA-SCAN decoder.

3.3 Metric for SCAN-Flip Decoder

Based on above discussion, it is clear that, by identifying the flipping position of the erroneous bits and using this information to initialize the $\beta_{i,0}$ can improve the performance of the SCAN decoder significantly. However, we can not exactly locate the positions of erroneous bits by CRC checking. In order to identify the positions and correct the erroneous bits, in [10], a Viterbi-aided SC decoder is proposed to provide additional protection for the noisiest bits. To verify its effectiveness, we first analyze the distribution of erroneous bits decision in the decoded codeword via Monte Carlo simulations. In this analysis, we employ an ideal OA-SCAN decoder to identify the positions of erroneous bits, which are caused by channel noise, in decoding a polar code with $N = 1024$ and $R = 0.5$ over AWGN channel with various SNR.

From Fig. 3, we observe that the distribution of erroneous bits is irregular. In other words, we still need to identify the flipping positions in the decoding process. In order to identify the positions more accurately and rapidly, we need to use more efficient metric, since the metric affects the rank in the flipping list. In current researches, the absolute LLR values are used as the metric in the original SC-Flip decoder, while a new metric for determining the position of erroneous bits is proposed in [6,7]. This metric is designed to find the bits, which have more probability to correct the trajectory of the SC-Flip decoder by considering the sequential aspect of the SC decoder. In [14], a critical set containing the

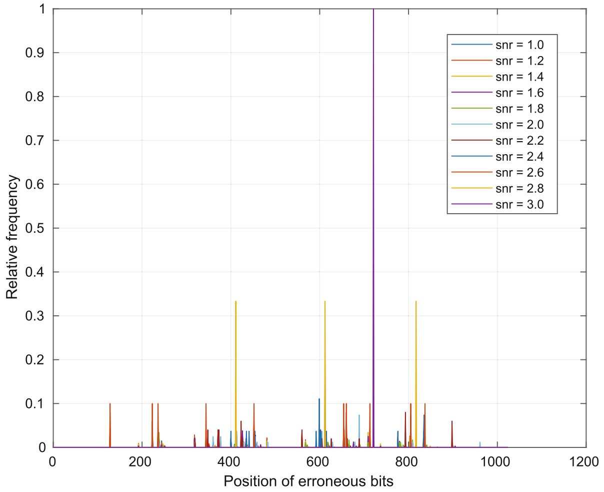

Fig. 3. Normalized distribution of error for polar code (1024,512) over AWGN channel with various SNR[dB], obtained simulating 2×10^7 frames.

first information bits of each rate-1 constituent codes is proposed to reduce the comparing scope. In order to compare the effectiveness of these metrics, we make a hit ratio comparison of these metrics via Monte-Carlo simulation, by running an order-1 OA-SCAN decoder to find the position of the first error, then declaring a hit if the position is in the list got by different metrics.

From Fig. 4, we can conclude that the metric M_α proposed in [6] has a much higher hit ratio than other metrics. Higher hit ratio means earlier to find the first erroneous bit position, which leads to lower decoding latency. Therefore, we choose the M_α as our metric in SCAN-Flip decoder. Furthermore, in order to improve the hit ratio of M_α, we make Monte-Carlo simulation to find the optimal parameter α of M_α as that do in [6]. In Fig. 5, the simulation result shows that different values of α have little effect on the BER performance, while the average rank of the first error bit in the set first degrades as α increasing and then upgrades when α adopting more higher values. Based on this result, we adopt $\alpha = 0.9$ to calculate the M_α value in our SCAN-Flip decoder.

4 Enhanced SCAN-Flip Decoding

In the above simulations of proposed SCAN-Flip decoder, we find that for many wrong estimated codewords the first error is already identified by the flipping list, while the CRC check fails. The reason for this phenomenon is that there are more than one error in the estimated codeword or that the decoding algorithm can not find out all of the errors in the limited decoding attempts. If we use the result got by wrong bit-flip decoding, we will miss the result with less errors. Therefore, we need to check the correctness of each bit-flip decoding attempt, in order to

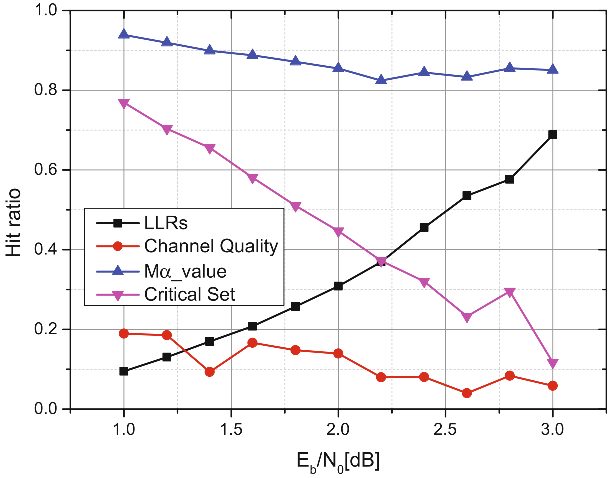

Fig. 4. Hit ratio curves of different flipping sets got by oracle-assisted SCAN decoders for decoding polar code (1024,512) with different metric over AWGN channels.

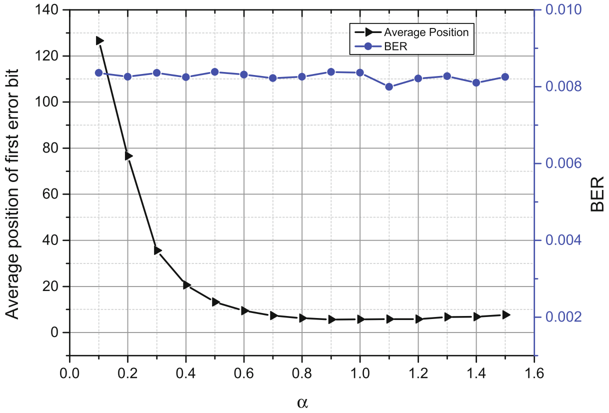

Fig. 5. BER performance and average position of first error bit in the flipping set got by oracle-assisted SCAN decoders for decoding polar code (1024,512) with different α over an AWGN channel with $\mathrm{E_b/N_0} = 2.5\,\mathrm{dB}$.

find more than one erroneous bit in the subsequent iterative calculation. From the view of decoding attempts, the SC-Flip decoder could be seen as a variation of SCL by exploring different decoding paths in different decoding attempts. Inspired by this, we modify the decoding procedure of our proposed SCAN-Flip decoder, basing on the path metrics. By comparing the path metrics of different decoding attempts, we could affirm the correctness of current flipping.

Different from the path metric proposed in [5], we use the accumulation of M_α as the path metric $Path^{(t)}_{M_\alpha}$. After each decoding attempt, we compare its path metric with the previous one. If the new path metric is lower than the previous one, we use the new result to update the flipping set and initialize the $\beta_{i,0}$. If not, we still use the result of the previous one. The details of this Enhanced-SCAN-Flip (E-SCAN-Flip) decoding process are described in Algorithm 2.

Algorithm 2. Enhanced SCAN-Flip Decoding Algorithm

```
Input: the received vector $y_1^N$, maximum iteration $T_{max}$
Output: a decoded vector $\hat{u}_1^N$
Init$\beta_1(I)$
$(\hat{u}_1^N, \{\lambda_i^{(1)}\}_{i \in I}, Path_{M_\alpha}^{(1)}) \leftarrow$ SCAN$(y_1^N, \beta_1)$
if CRC$(\hat{u}_1^N)$=success then return $\hat{u}_1^N$
else Flipping_set$(\{\lambda_1^{(1)}\}_{i \in I})$ Init$\beta_2(Flipping\_set)$
end if
for t = 2,...,$T_{max}$ do
    $(\hat{u}_1^N, \{\lambda_i^{(t)}\}_{i \in I}, Path_{M_\alpha}^{(t)}) \leftarrow$ SCAN$(y_1^N, \beta_t)$
    if CRC$(\hat{u}_1^N)$=success then return $\hat{u}_1^N$
    else
        if $Path_{M_\alpha}^{(t)} < Path_{M_\alpha}^{(t-1)}$ then
            Flipping_set$(\{\lambda_i^{(t)}\}_{i \in I})$
            Init$\beta_{t+1}(Flipping\_set)$
            Update$(\hat{u}_1^N)$
        else
            Init$\beta_{t+1}(Flipping\_set)$
        end if
    end if
end for
return $\hat{u}_1^N$
```

5 Simulation Results

In this section, the BER performance and the average number of iterations of the proposed SCAN-Flip decoding algorithm are investigated via Monte-Carlo simulation. Specifically, we focus on the transmissions with BPSK modulation over AWGN channel. In order to compare with other researches, the interest SNR regime is 1.0 dB to 3.0 dB. Polar codes are constructed targeting a SNR of 2.5 dB with parameters $N = 1024$ and $K = 512$ using Gaussian approximation construction method and concatenated with a CRC-8 with generator polynomial $g(D) = D^8 + D^7 + D^6 + D^4 + D^2 + 1$.

The BER performance of SCAN-Flip decoders with different predetermined maximum decoding attempts T_{max} are shown in Fig. 6. From the comparison, we observe that the maximum number of iteration has little effect on the performance since these decoders have the same correcting order. They can only

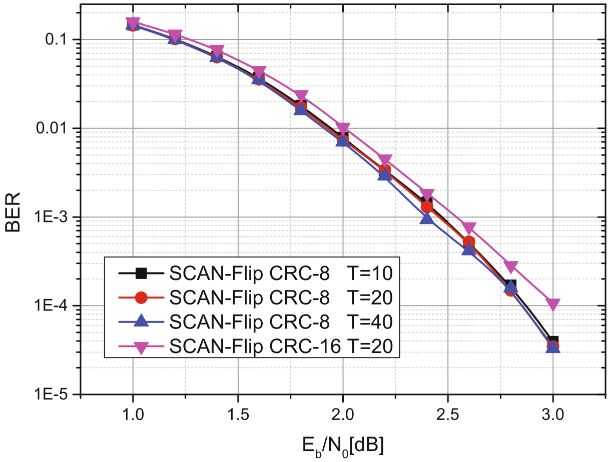

Fig. 6. BER performance of SCAN-Flip decoder with different predetermined maximum decoding attempts T_{max} and different CRC code lengths for polar code (1024,512). CRC-16 is the 16 bits CCITT CRC code.

correct one single error in a codeword. Then, we make a comparison of SCAN-Flip decoders with different length CRC code. The decoder with CRC-16 has a little inferior performance for its real code rate is little large than that of the decoder with CRC-8, which also reflects that the CRC-8 can provide enough checking capability for SCAN-Flip decoder as shown in Fig. 6.

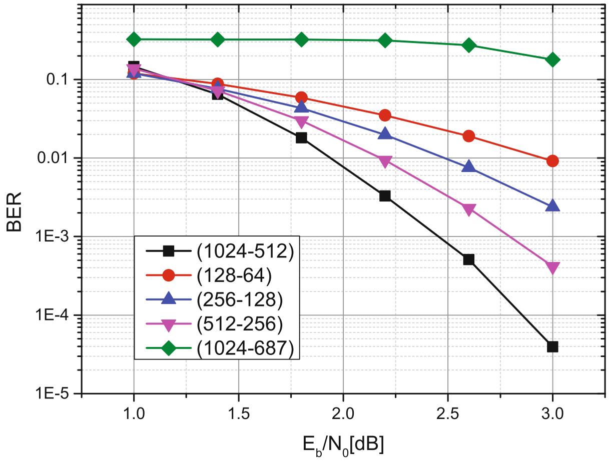

Fig. 7. BER performance of polar codes with different code rates and code lengths, concatenated with CRC-8, $T_{max} = 10$.

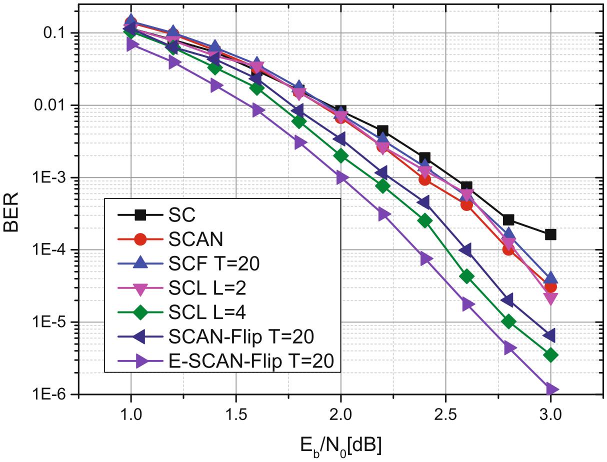

Fig. 8. BER performance for polar code (1024,512), CRC-8 with different decoding approaches.

In order to evaluate the BER performance of SCAN-Flip decoders with different code rates and different code lengths, we make comparison with the same condition above. In Fig. 8, these polar codes are all constructed targeting a SNR of 2.5 dB, concatenated with CRC-8, while the predetermined maximum decoding attempt is $T_{max} = 10$. We could conclude that as the code length increases, the BER performance of SCAN-Flip decoder improves, since the longer polar codes have better rate of polarization. And the error-correction performance of SCAN-Flip decoder degrades when the code rate increases. The reason for this is that the polarization is not complete at the code length $N = 1024$. Higher code rate means that more information bits will be transmitted through the subchannels with noise, which increases the probability to be estimated erroneously.

In Fig. 8, we compare the performance of SCAN-Flip with that of CRC-aided SCL with $L = \{2, 4\}$ and CRC-8, the SCF decoder with the maximum number of iterations $T_{max} = 20$ and CRC-8, SCAN decoder and SC decoder as the baseline decoding method. We observe that the performance of the SCAN-Flip decoder with $T_{max} = 20$ is a little weaker than that of SCL decoder with list size $L = 4$. At the bit error rate of 10^{-4}, SCAN-Flip outperforms SC-Flip by up to 0.25 dB. Moreover, we further plot the performance of our enhanced SCAN-Flip decoder. Its error-correction performance is even better than that of SCL ($L = 4$) decoder, at the cost of higher computational complexity.

In order to evaluate the computational complexity of SCAN-Flip decoder, we make simulations to calculate its average number of iterations, since it is directly proportional to the computational complexity. In Fig. 9, we compare the average number of iterations of SCAN decoder, SCF decoder, SC decoder, normalized SCL decoder, and our SCAN-Flip decoder. The comparisons are made at frame error rate (FER) of 10^{-4} with same parameters as above. At high SNR, the SCF

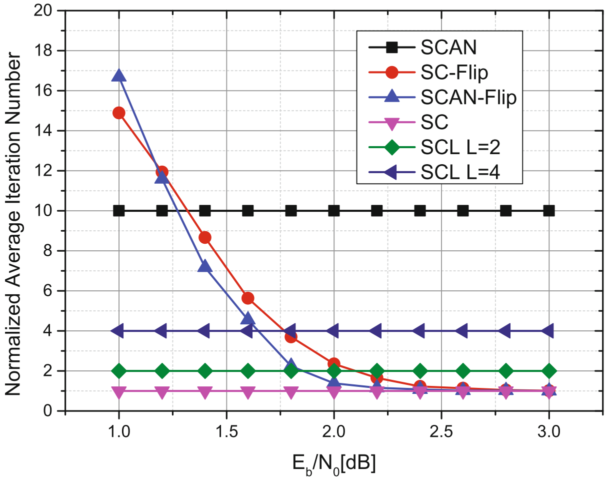

Fig. 9. Normalized average iteration number for polar code (1024,512), CRC-8, $T_{max} = 20$ with different decoding approaches.

decoder and SCAN-Flip decoder both have low iteration number. However, since the SCAN-Flip decoder has a more efficient message propagating mechanism, it has a lower iteration number than that of SCF decoder at low SNR.

6 Conclusion

In this paper, we propose the SCAN-Flip polar decoder by introducing the flipping idea into the SCAN decoder. It works by initializing the $\beta_{i,0}$ LLRs corresponding to the former iteration result and affecting the estimated result using message propagation. Furthermore, we propose a path metric designed for correcting more errors of our proposed SCAN-Flip decoder. The simulation results show that the performance is competitive with SCL decoding, while the computational complexity is almost as low as that of the SC decoder. Besides, at the equivalent FER, the SCAN-Flip decoder has less number of iterations than that of SC-Flip decoder, which leads to lower decoding latency.

References

1. Afisiadis, O., Balatsoukas-Stimming, A., Burg, A.: A low-complexity improved successive cancellation decoder for polar codes. In: 2014 Asilomar Conference on Signals, Systems and Computers, pp. 2116–2120 (2014)
2. Araniti, G., Campolo, C., Condoluci, M., Iera, A.: LTE for vehicular networking: a survey. Commun. Mag. IEEE **51**(5), 148–157 (2013)
3. Arikan, E.: A performance comparison of polar codes and Reed-Muller codes. Commun. Lett. IEEE **12**(6), 447–449 (2008)

4. Arikan, E.: Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans. Inf. Theory **55**(7), 3051–3073 (2008)
5. Balatsoukas-Stimming, A., Parizi, M.B., Burg, A.: LLR-based successive cancellation list decoding of polar codes. IEEE Trans. Signal Process. **63**(19), 5165–5179 (2014)
6. Chandesris, L., Savin, V., Declercq, D.: An improved SCFlip decoder for polar codes, pp. 1–6 (2017)
7. Chandesris, L., Savin, V., Declercq, D.: Dynamic-SCFlip decoding of polar codes. IEEE Trans. Commun. **66**(6), 2333–2345 (2018)
8. Chen, K., Niu, K., Lin, J.: Improved successive cancellation decoding of polar codes. IEEE Trans. Commun. **61**(8), 3100–3107 (2013)
9. Fayyaz, U.U., Barry, J.R.: Low-complexity soft-output decoding of polar codes. IEEE J. Sel. Areas Commun. **32**(5), 958–966 (2014)
10. Fazeli, A., Tian, K., Vardy, A.: Viterbi-aided successive-cancellation decoding of polar codes. In: GLOBECOM 2017–2017 IEEE Global Communications Conference, pp. 1–6 (2017)
11. Hussami, N., Korada, S.B., Urbanke, R.: Performance of polar codes for channel and source coding. In: IEEE International Symposium on Information Theory, pp. 1488–1492 (2009)
12. Support, M.: Final report of 3GPP TSG RAN WG1 #87 v1.0.0 Feburary 2017. http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_88/Docs/R1-1701552.zip
13. Tal, I., Vardy, A.: List decoding of polar codes. IEEE Trans. Inf. Theory **61**(5), 2213–2226 (2012)
14. Zhang, Z., Qin, K., Zhang, L., Zhang, H., Chen, G.T.: Progressive bit-flipping decoding of polar codes over layered critical sets (2017)