Asymmetrical Extended Min-Sum for Successive Cancellation Decoding of Non-Binary Polar Codes

Joseph Jabour, Ali Chamas Al Ghouwayel, E. Boutillon

Asymmetrical Extended Min-Sum for Successive Cancellation Decoding of Non-Binary Polar Codes

JOSEPH JABOUR[1,2], ALI CHAMAS AL-GHOUWAYEL[3], EMMANUEL BOUTILLON[1]
[1]Laboratoire Lab-STICC, UMR CNRS 6285, Université Bretagne Sud, Lorient, France
[2]Lebanese International University, Beirut, Lebanon.
[3] Université Paris-Panthéon-Assas, Efrei Research Lab, Paris, France.
joseph.jabour@univ-ubs.fr, ali.ghouwayel@efrei.fr, emmanuel.boutillon@univ-ubs.fr

***Abstract*—In this paper, an optimization of the non-binary polar decoders is realized in two aspects. Firstly, the L-bubble sorter, a simplified version of the Extended Min-Sum (EMS) algorithm, is applied to the non-binary polar decoder. Then, a criterion is set to evaluate the relative reliability of the two inputs of the non-binary kernel. Accordingly, the kernel processes the two inputs differently. The proposed asymmetrical process reduces the decoder's complexity by reducing the number of candidates to be computed. Over the Galois field of size 64, the size of computed candidates is reduced by a factor of 2 compared to the EMS algorithm. When simulated over a real additive white Gaussian noise channel with a binary phase shift keying modulation, the proposed decoder showed a performance degradation of 0.25 dB compared to the min-sum algorithm. However, when the cyclic-code shift keying modulation is used, the degradation is reduced to around 0.1 dB.**

***Index Terms*—Non-binary Polar Codes, CCSK, Extended Min-Sum, Successive Cancellation**

I. INTRODUCTION

Short packet communication is the dominant type of communication in the Internet of Things (IoT) ecosystem, where sensors, machines, and systems interact. The emerging applications of short-packet communication require ultra-reliable, low-latency communication [1]. Therefore, non-binary error-correcting codes are becoming more attractive than ever due to their error-correction capabilities over short frames. In addition, data transmission over ultra-low Signal-to-Noise Ratio (SNR) can be achieved by association of the non-binary codes with Cyclic Code-Shift Keying (CCSK) modulation [2].

Polar codes are the latest capacity-approaching error-correcting codes invented by Erdal Arikan in 2008 [3]. A polar code is defined for a message of size K and a code of size N, constructed based on a defined generator matrix G_N. The polar codes are based on channel polarization which transforms a physical channel into multiple (virtual) sub-channels that are polarized into either noiseless or extremely noisy (useless) channels. The Non-Binary Polar Codes (NB-PCs) can be extended from the binary codes by assigning non-binary coefficients to the elements of the generator matrix [4].

The complexity of NB-PC decoders is much higher when compared with their binary counterparts. Therefore, different contributions were proposed to reduce the complexity of the NB-Polar codes. In [5], the authors propose a low-complexity Successive Cancellation (SC) decoder based on the Min-Sum (MS) algorithm, which is derived from the original SC algorithm [3], but defined in the Log-Likelihood Ratio (LLR) domain. Processing the LLR messages in the logarithmic domain allows replacing all the multiplication operations with only additions; hence, no multipliers are required. In addition, the authors proposed a simplified MS algorithm based on truncating one of the two inputs of the nodes from q down to n_0 with $0 < n_0 \ll q$ which allows for reducing the complexity from order $\mathcal{O}(q^2)$ down to the order of $\mathcal{O}(q \times n_0)$ for a kernel of order 2. In addition, a recent contribution [6] allowed further reduction in the check node processing by using the Extended Min-Sum (EMS) algorithm, which reduces the complexity of the check node down to an order of $\mathcal{O}(n_m^2)$ with $n_m \ll q$. Moreover, in [7], the authors proved that using CCSK modulation allows for setting all kernel coefficients to 1 without performance degradation and hence, reduces both the encoding and the decoding processes.

This paper proposes a simplified SC decoder by analyzing the relative reliability of the input messages. In addition, the check node processing employs an efficient algorithm called L-bubble [8] for further reduction in the arithmetic operations required per node. The paper is outlined as follows, a brief introduction to CCSK modulation is mentioned in section II. Then, the polar transformation, encoding, and successive cancellation (polar) decoding are discussed in section III. Subsequently, the proposed asymmetrical SC decoder is covered in section IV. The simulation results and the complexity analysis of the proposed decoder are tackled in section V and lastly, an overall conclusion is drawn in section VI.

II. CYCLIC-CODE SHIFT KEYING MODULATION

The Cyclic-Code Shift Keying (CCSK) modulation [9] is a modulation technique based on the spread-spectrum modulation, where a message of size $p-$bits represented in decimal as α, $\alpha \in GF(q = 2^p)$, is modulated over a Pseudo-random Noise (PN) sequence of size q. Therefore, the spreading factor can be deduced as $S_F = \frac{p}{q}$.

Assume a fundamental PN sequence (corresponds to $\alpha = 0$) denoted as ψ_0 such that $\psi_0 = (\psi_0(0), \psi_0(1), \cdots, \psi_0(q-1))$ with $\psi_0(k) \in \{0, 1\}\ \forall\ k = 0, \cdots, q-1$. A PN sequence ψ_α for any symbol $\alpha \in GF(q)$ can be deduced from the fundamental PN sequence ψ_0 by circularly shifting the latter as follows:

$$\psi_\alpha(k) = \psi_0((k + \alpha) \mod q)\ \forall\ k = 0, \cdots, q-1. \quad (1)$$

Let $\psi = (\psi_{x_0}, \psi_{x_1}, \cdots, \psi_{x_{N-1}})$ denote the CCSK modulated frame of the codeword $x = (x_0, \cdots, x_{N-1})$ where x_i is a symbol of $GF(q)$ (thus of size p-bits). The length of the frame is thus Nq bits (or chips). The chips of ψ are modulated over a Binary Phase Shift Keying (BPSK) and transmitted over a Binary Additive White Gaussian Noise (BI-AWGN) channel with noise variance $\sigma^2 = 10^{-\frac{E_s/N_0}{10}}$ where E_s/N_0 represents the SNR value. The i^{th} transmitted CCSK symbol ψ_{u_i} is thus received as a vector $y_i = (y_i(k))_{k=0,1,\ldots,q-1}$ with $y_i(k) = (2\psi_{x_i}(k) - 1) + w_i(k)$ with $w_i(k)$ a sample of the additive noise. The intrinsic LLR vector $L_i^{(0)}$ computed from the channel observation is thus given as $L_i^{(0)} = (L_i^{(0)}(0), \cdots, L_i^{(0)}(q-1))$, with, for a given GF symbol α (see [4] for further details):

$$L_i^{(0)}(\alpha) = \sum_{k=0}^{q-1} \frac{2y_i(k)}{\sigma^2}(\psi_{\hat{x}_i}(k) - \psi_\alpha(k)), \tag{2}$$

where $\hat{x}_i$ represents the maximum likelihood decision on y_i defined as

$$\hat{x}_i = \underset{\alpha \in\ GF(q)}{\operatorname{argmax}} \{\sum_{k=0}^{q-1} \frac{2y_i(k)}{\sigma^2}\psi_\alpha(k)\}. \tag{3}$$

Note that by construction, the LLR values are all positive and $L_i^{(0)}(\hat{x}_i) = 0$.

III. Polar Codes and Decoders

The structure of a polar code depends on the kernel being adopted. A kernel G_2 is a generator matrix for $N = 2$ and is considered the basic transformation unit of a polar code:

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & \gamma \end{bmatrix}\ ,\ \gamma \in GF(q). \tag{4}$$

The kernel G_2 transforms the input vector $u = (u_0, u_1)$ into an output $x = (x_0, x_1)$ by the following input-output relation

$$\begin{cases} x_0 = u_0 \oplus u_1 \\ x_1 = \gamma \circledast u_1 \end{cases} \tag{5}$$

where $\circledast$ represents the field multiplication and $\oplus$ represents the field addition over $GF(q)$.

The polar encoder generates a codeword based on the polar transformation that leads to channel polarization. The encoder encodes the message m of size K into a codeword x of size N by allocating the message m into the K most reliable channel positions (having the least error probability $\mathcal{P}_\epsilon$) such that $u_i = m_j : i \in \mathcal{A}_D, 0 \leq j < K$ where $\mathcal{A}_D$ represents the set of the K most reliable input positions (Data channels). The remaining $N - K$ noisiest channels are frozen such that $u_i = 0\ \forall\ i \notin \mathcal{A}_D$. Subsequently, the message u is transformed into the codeword x using a generator matrix $G_N = G_2^{N\otimes}$, with $\otimes$ representing the Kroneker product. Nevertheless, since the frozen channels have a fixed input value (zero), they are used at the decoding stage to help in obtaining the originally transmitted message.

The SC decoder generates an estimated $\hat{u}$ of the information message u from the channel information on x. It consists of $n = \log_2(N)$ layers where each layer has $N/2$ kernels. Each kernel is composed of a single parity Check Node (CN) and a repetition node called Variable Node (VN).

As N increases, the processing of the decoder gets more intricated since the estimation of a symbol $\hat{u}_i, 0 \leq i < N$, depends on all previous estimates $\hat{u}_j, 0 \leq j < i$ and the received codeword. Consider the decoder for $N = 4$ as shown in Fig. 1. The decoder consists of $n = \log_2(4) = 2$ layers, and the LLR vectors generated at each layer l, $1 \leq l \leq n$, are denoted by $L_i^{(l)}$, with $L_i^{(l)} = (L_i^{(l)}(0), L_i^{(l)}(1), \cdots, L_i^{(l)}(q-1))$, $0 \leq i < N$. For instance, at layer 1, the input LLR vectors $L_i^{(0)}, 0 \leq i < N$, corresponds to the channel observation computed in (2).

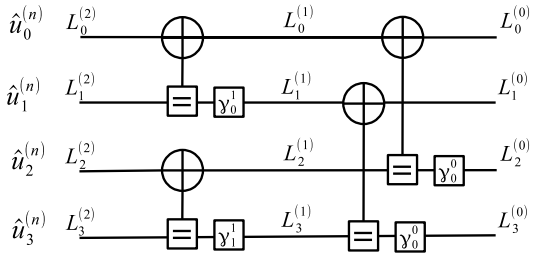

Fig. 1. Polar Decoder for $N = 4$

The processing schedule of a kernel at layer l and position t is illustrated in Fig. 2. The kernel has two LLR input messages at indices θ_t^{l-1} and ϕ_t^{l-1} that can be deduced as follows

$$\begin{aligned} \theta_t^{l-1} &= 2t - (t \mod 2^{n-l}) \\ \phi_t^{l-1} &= 2^{n-l} + 2t - (t \mod 2^{n-l}) \end{aligned} \tag{6}$$

for $t = 0, \cdots, N/2 - 1$. The position t and layer l are omitted from the indices θ_t^{l-1} and ϕ_t^{l-1} in the sequel for the simplicity and readability of the notations. The two LLR input vectors to the kernel are denoted $L_\theta^{(l-1)}$ and $L_\phi^{(l-1)}$ which are processed by the kernel to obtain the output vectors $L_\theta^{(l)}$ and $L_\phi^{(l)}$.

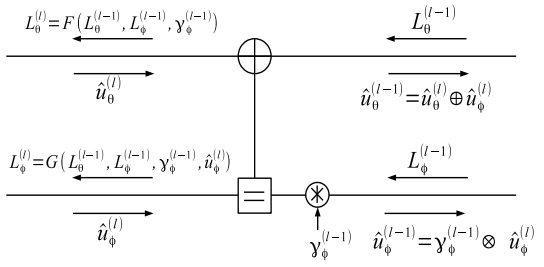

Fig. 2. Processing Stages of the t^{th} kernel at layer l

The check node update (represented as function $F()$ in Fig. 2) can be recursively computed as

$$L_\theta^{(l)}(\beta) = \min_{\eta \in GF(q)} (L_\theta^{(l-1)}(\beta \oplus \eta) + L_\phi^{(l-1)}(\gamma \circledast \eta)) \quad \forall\ \beta \in GF(q). \tag{7}$$

Similarly, the variable node update (represented as function $G()$ in Fig. 2) can be computed recursively as,

$$L_\phi^{(l)}(\eta) = L_\theta^{(l-1)}(\hat{u}_\theta^{(l)} \oplus \eta) + L_\phi^{(l-1)}(\gamma \circledast \eta)\ \forall\ \eta \in GF(q), \tag{8}$$

where $\hat{u}_\theta^{(l)}$ is the estimated symbol at index θ of layer l.

Normalizing the LLR vectors generated in (8) should be always maintained to prevent arithmetic overflow due to data accumulation. The normalization is performed on an LLR vector $L_i^{(l)}$ by subtracting the minimum LLR value of the vector from all elements as follows

$$L_i^{(l)}(\alpha) = L_i^{(l)}(\alpha) - \min(L_i^{(l)}) \ \forall \alpha \ \in \ GF(q) \tag{9}$$

The estimated symbols are back-propagated to obtain the corresponding symbols at the previous layer by computing the following

$$(\hat{u}_\theta^{(l-1)}, \hat{u}_\phi^{(l-1)}) = (\hat{u}_\theta^{(l)} \oplus \hat{u}_\phi^{(l)}, \gamma \circledast \hat{u}_\phi^{(l)}) \tag{10}$$

The hard decision $\hat{u}_i^{(n)}$ at layer $l = n$ and for a node $i = 0, 1, \cdots, N-1$ is estimated as

$$\hat{u}_i^{(n)} = \begin{cases} 0, & \text{if } i \notin \mathcal{A}_D \\ \underset{\alpha \in \ GF(q)}{\operatorname{argmin}} \ L_i^{(n)}(\alpha), & \text{if } i \in \mathcal{A}_D. \end{cases} \tag{11}$$

IV. Asymmetrical EMS-based SC Decoder

The complexity of the CN update in the SC decoder can be reduced by reducing the computation performed to estimate the reliability of the symbols $u_\theta^{(l)} \in GF(q)$. Similar to Non-Binary Low-Density Parity Check (NB-LDPC) codes, the EMS [10] can be used to reduce the number of computations performed as in [6]. This algorithm, named SC-EMS, consists of selecting the most n_m competitive elements in the input vectors $L_\theta^{(l-1)}$ and $L_\phi^{(l-1)}$ that are more likely to contribute to the output $L_\theta^{(l)}$.

Let $L_\phi^{(l-1)}$ be the ordered subset of $L_\phi^{(l-1)}$ composed of the n_m smallest elements (i.e., the LLR corresponding to the n_m most reliable symbols). $\bar{L}_\phi^{(l-1)} = (\bar{L}_\phi^{(l-1)}(j))_{j=0,1,\ldots,n_m-1}$, with $\bar{L}_\phi^{(l-1)}(j) = L_\phi^{(l-1)}(\eta_j^{(l-1)})$, where $\eta_j^{(l-1)}$ corresponds to the j^{th} most reliable GF symbol in $L_\phi^{(l-1)}$. Similarly, $\bar{L}_\theta^{(l-1)}$ is the ordered subset of $L_\theta^{(l-1)}$, with $\bar{L}_\theta^{(l-1)}(i) = L_\theta^{(l-1)}(\beta_i^{(l-1)})$ where $\beta_i^{(l-1)}$ corresponds to the i^{th} reliable GF symbol in $L_\theta^{(l-1)}$. The truncation of both CN inputs down to n_m reduces the entire computation from q^2 operations performed in Min-Sum SC (SC-MS) decoder expressed in (7) down to n_m^2 where $n_m \ll q$.

Let the matrix T^Σ consist of n_m^2 elements where each element is of two tuples $T_{i,j}^{\Sigma\oplus}$ and $T_{i,j}^{\Sigma+}$ that correspond to the GF symbol and LLR value obtained by the addition of the GF elements $\beta_i^{(l-1)}$ and $\eta_j^{(l-1)}$, and the LLR elements $\bar{L}_\theta^{(l-1)}(i)$ and $\bar{L}_\phi^{(l-1)}(j)$ respectively such that

$$\begin{aligned} T_{i,j}^{\Sigma+} &= \bar{L}_\theta^{(l-1)}(i) + \bar{L}_\phi^{(l-1)}(j) \\ T_{i,j}^{\Sigma\oplus} &= \beta_i^{(l-1)} \oplus (\gamma \circledast \eta_j^{(l-1)}). \end{aligned} \tag{12}$$

The n_m^2 elements of the matrix T^Σ are sorted in the ascending order of $T^{\Sigma+}$, and the most reliable n_m symbols (denoted as $S_\Sigma^\oplus$) with their corresponding LLRs (denoted as S_Σ^+) are used to update the message L_θ^l such that

$$L_\theta^{(l)}(S_\Sigma^\oplus(i)) = S_\Sigma^+(i), 0 \leq i < n_m. \tag{13}$$

An efficient approach for selecting the n_m most reliable elements from the matrix $T^\Sigma = (T^{\Sigma\oplus}, T^{\Sigma+})$ is presented in [8] for NB-LDPC codes. Using the L-bubble sorter in the SC decoder (indicated as SC-LEMS) takes advantage of the sorted nature of the input messages $\bar{L}_\theta^{(l-1)}$ and $\bar{L}_\phi^{(l-1)}$ to significantly reduce the computations required for obtaining the vector $S^\Sigma = (S^{\Sigma\oplus}, S^{\Sigma+})$. The complexity of the CN is reduced by considering only the first two rows and two columns of T^Σ for obtaining the n_m elements of S^Σ.

The SC-LEMS decoder can even be simplified by considering that, among the two inputs of a CN, one is relatively more reliable than the other. This allows the processing of the two inputs asymmetrically and hence, is named Asymmetrical Extended Min-Sum for SC decoder (SC-AEMS). The truncation level is replaced from n_m elements for both input vectors with two truncation levels: the first level is of n_L elements assigned for the most reliable input vector, and the second is of n_H assigned for the least reliable input vector. This can be estimated by performing a statistical analysis at the check nodes (not presented due to space limitations).

The criterion to assess which input is more reliable is derived from [11] by comparing the two elements of index z of input vectors $\bar{L}_\theta^{(l-1)}$ and $\bar{L}_\phi^{(l-1)}$. The input with the highest LLR value is considered the most reliable, and its truncation length is thus set to n_L. This assessment determines the size of potential candidates of $\bar{L}_\theta^{(l-1)}$ (denoted as n_θ) and $\bar{L}_\phi^{(l-1)}$ (denoted as n_ϕ) as shown in Algo. 1. The potential elements from the inputs are processed using the L-bubble to generate the n_H most reliable candidates of L_θ^l. The comparison index z can be the first or the second non-zero element, i.e., $z = 1$ or $z = 2$.

Algorithm 1: Asymmetrical EMS-based Successive Cancellation Check Node

Input: $\bar{L}_\theta^{(l-1)}$, $\bar{L}_\phi^{(l-1)}$, $(\beta^{(l-1)})$, $(\eta^{(l-1)})$, n_L, n_H, i

Pre-processing Steps: Comparison of the z^{th} most reliable LLR value:

if $\bar{L}_\theta^{(l-1)}(z) > \bar{L}_\phi^{(l-1)}(z)$ **then**

  $(n_\theta, n_\phi) \leftarrow (n_L, n_H)$

else

  $(n_\theta, n_\phi) \leftarrow (n_H, n_L)$

end

Processing Steps:

Step 1: Generate T^Σ as in (12) with $i = 0, \cdots, n_\theta - 1$ and $j = 0, \cdots, n_\phi - 1$.

Step 2: Extract n_H most reliable LLR values in T_Σ and their corresponding GF symbols to obtain S_Σ^+ and $S_\Sigma^\oplus$ using L-bubble sorter.

Step 3: Compute L_θ^l as in (13)

Output: $L_\theta^l(\alpha) \ \forall \ \alpha \in GF(q)$

V. Simulation Results and Discussion

The SC-AEMS decoder reduces the decoding complexity and/or latency by reducing the number of computed elements required to generate an output. In addition, it helps avoid routing congestion in the parallel implementation of the decoder.

In the SC-MS decoder, the size of the input messages is q elements. Therefore, the CN unit processes the q^2 elements of the two inputs to obtain the most reliable LLRs for the q symbols (as depicted in (7)). For the SC-EMS-based decoder, the size of the input messages is reduced down to n_m and the CN operations (13) are reduced from q^2 down to $n_m\sqrt{n_m}$ when the bubble check is used as in [12].

The SC-LEMS algorithm allows for reducing the internal operations of the CN from $n_m\sqrt{n_m}$ down to $4 \times n_m - 4$ (computing only the first two rows and columns T^{Σ}). In addition, by analyzing the relative reliability, the complexity of the CN is further reduced. The computed elements of T^{Σ} are reduced from $4 \times n_m - 4$ down to $2 \times (n_H + n_L) - 4$ with $n_L < n_H \ll q$. The arithmetic operations a CN requires over the different decoders are summarized in Table I. In all algorithms that use sorting and normalization, the real addition operations are less than the GF additions. This is because the first element in both inputs is equal to zero. Therefore, the LLR values of the first row and the first column are identical to their corresponding non-zero input elements (see (12)).

TABLE I
Arithmetical Operations Performed per Check Node

Algorithm	GF Additions	Real Additions
SC-MS	q^2	q^2
SC-EMS	$n_m\sqrt{n_m}$	$n_m\sqrt{n_m}$
SC-LEMS	$4 \times n_m - 4$	$2 \times n_m - 3$
SC-AEMS	$2 \times (n_H + n_L) - 4$	$(n_H + n_L) - 3$

Using Monte-Carlo simulation over $GF(64)$, an SC-AEMS decoder with $n_H = 8$, $n_L = 20$, and $z = 2$ is found to perform as well as the SC-LEMS and is negligibly degraded compared to the performance of the SC-MS decoder. Therefore, the CN processing can be realized by performing the arithmetic and logic (over $GF(64)$) operations as indicated in Table I. The required GF and LLR additions for the MS decoder are 4096 operations for each. Furthermore, the required GF and LLR addition operations for the SC-EMS decoder with $n_m = 20$ are 90 and 90 respectively. For the SC-LEMS, this is reduced to 76 and 37 operations, respectively. Lastly, in the proposed SC-AEMS decoder with $n_H = 20, n_L = 8$, the total GF additions are only 52 operations and 25 LLR additions. The computed candidates in the SC-AEMS decoder are reduced by 42% (for GF operation), and by 72% (for LLR additions) compared to the SC-EMS decoder.

On the other hand, the SC-AEMS decoder requires a comparator at each check node. This sums up to $N/2$ check nodes in each layer over the $n - 1$ layers (at the last layer, only one element from each input is required to get a hard decision). Therefore, $N/2 \times n - 1$ comparators are required.

The proposed decoder is simulated over a BI-AWGN channel with CCSK modulation and a sequence of size $q = 64$ chips. The coefficient γ is set to 1 for all codes based on the results provided in [7], and the frozen positions are obtained by analyzing the error rate at each position $i = 0 \cdots N - 1$ using a genie-aided decoder [3]. The decoder has the following parameters for $GF(64)$, $n_H = 8$, $n_L = 20$, and $z = 2$, fixed overall codes over $GF(64)$.

The effective code rate over a CCSK modulation can be deduced as $r_e = r \times S_F$ where r is the coding rate and S_F is the spreading factor. Since the simulations are performed over $GF(q = 64)$, the effective code rate is $r_e = r \times \frac{6}{64}$.

In Fig. 3, the Frame Error Rate (FER) performance of different decoding approaches is presented. The black plot represents the minimum achievable FER over finite-block coding [13] for $K = 21$ symbols and $N = 64$ over CCSK modulation on $GF(64)$ and the dashed-black plot represents the FER of the SC-MS decoder. Similarly, the dashed-blue and the dotted-blue plots represent the FER of the SC-LEMS decoder with $n_m = 20$ and $n_m = 14$ respectively. The dashed-cyan plot represents the FER of the proposed SC-AEMS decoder.

In addition, the FER performance of the binary polar SC and Successive Cancellation List (SCL) decoders is plotted using two approaches. The first approach is assuming a binary polar code with a similar payload (in bits) and the same code rate as the effective code rate of the NB-PC, i.e., $K_b = Kp$ and $r_b = r_e$. Therefore, the simulated binary polar code has $K_b = 21 \times 6$ and $N_b = K_b/r_b = 4096$. The FER performance of the aforementioned binary decoder is illustrated via the violet plots where the dotted-violet and the dashed-violet plots represent the SC decoder and the SCL decoder with a list size of 32 respectively. The second approach is assuming a binary polar code with the same payload as the NB-PC and the same coding rate (instead of the effective code rate), i.e., $K_b = Kp$ and $r_b = r$, along with a repetition of $1/S_F$ (shifting the plot by $10\log_{10}(S_F)$) such that the effective coding rate is similar to that of the NB-PC. The FER performance of this approach is depicted via the red plots where the dotted-red and the dashed-red plots represent the SC decoder and the SCL decoder respectively. As shown, the SC decoder of the NB-PC outperforms the binary SCL of the two approaches.

The simulation results are depicted in Fig. 4 for the code length $N = 256$ symbols over rates $r \approx 1/3$ (left plot) and $r \approx 2/3$ (right plot). The finite block regime bound is plotted in black and corresponds to the minimum achievable FER. The simulated decoders are SC-MS (in black), SC-LEMS with $n_m = 20$ (in dashed blue), SC-LEMS with $n_m = 14$ (in dotted blue), and SC-AEMS (in cyan). As shown in both figures, the decoders have a comparable performance with degradation of around 0.1 dB at a FER of 10^{-5}.

The SC-LEMS decoder with $n_m = 14$ has an arithmetic complexity similar to that of the SC-AEMS decoder with $n_H = 20$ and $n_L = 8$. As shown in Fig. 3 and Fig. 4, the performance of the SC-LEMS decoder with similar complexity of that the SC-AEMS has a degraded performance of around

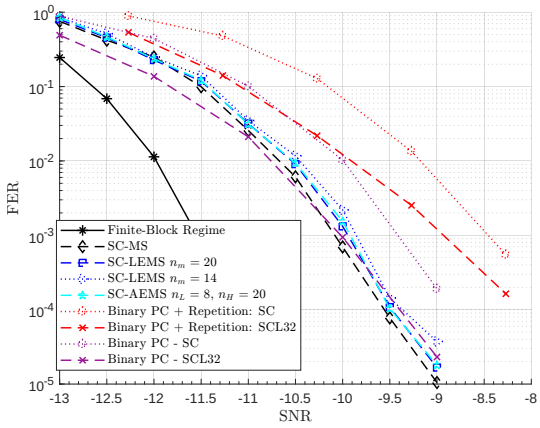

Fig. 3. FER Comparison of Different Binary and Non-Binary Polar Decoders

0.1 dB. Thus, the asymmetrical EMS algorithm helps reduce the SC decoder's complexity without affecting its performance significantly.

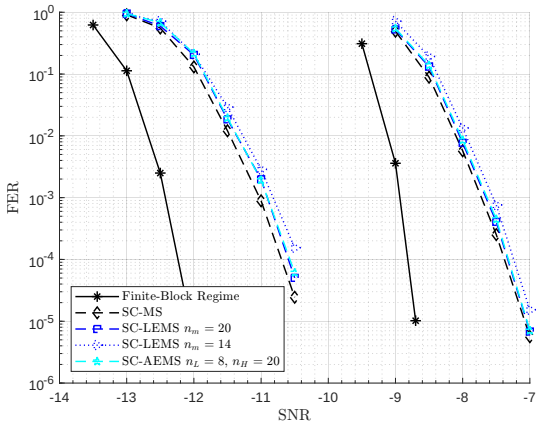

Fig. 4. Simulation Results over $GF(64)$ for $N = 256$, $r \approx 1/3$ ($r_e \approx 1/32$) and $r \approx 2/3$ ($r_e \approx 1/16$) respectively.

The performance of the SC-AEMS decoder has a similar behavior for different code lengths and rates tested up to $N = 1024$ symbols, with a maximum degradation of around 0.15 dB at a FER of 10^{-5}.

Furthermore, the performance of the proposed decoder is simulated over a BI-AWGN channel with BPSK. The performance of the SC-AEMS and SC-LEMS decoders is degenerated by around 0.25 dB and 0.13 dB, respectively, compared to the performance of the SC-MS decoder.

Lastly, the proposed asymmetrical decoder is presented within the frame of the successive cancellation. However, it can be easily extended within the list decoder for higher decoding performance.

VI. Conclusion

This paper presents the asymmetrical extended min-sum successive cancellation decoder. The optimization process targets the polar kernel by employing the L-bubble version of the EMS algorithm to reduce the size of the computed candidates. Furthermore, the overall complexity of the decoder is reduced by assessing the relative reliability of the check node inputs before processing. This helps in reducing the size of the considered candidates from n_m down to n_H and n_L leading to asymmetrical sizes of the input vectors. The size of the message with the lowest (highest) relative reliability is set to n_H (n_L), with $n_L < n_H$. In addition, the L-bubble sorter is used to reduce the arithmetic operations down to $2 \times (n_L + n_H) - 4$. For $GF(64)$, the computed candidates in the SC-AEMS decoder are reduced by 42% (GF additions) and 72% (real additions) compared to the EMS-based SC decoder, with a maximum degradation of 0.15 dB over CCSK modulation to that of the SC-MS. Lastly, further work is required to examine the quantization effect on the decoder, synthesize the proposed decoder, and assess its complexity compared to the state-of-the-art decoders.

Acknowledgment

This work has been partially funded by the French ANR under grant number ANR-19-CE25-0013-01 and the Lebanese International University. Special Thanks to Cédric Marchand and Hussein Hijazi for their fruitful discussions.

References

[1] G. Durisi, T. Koch, and P. Popovski, "Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets," *Proceedings of the IEEE*, 2016.

[2] O. Abassi, L. Conde-Canencia, M. Mansour, and E. Boutillon, "Non-Binary Coded CCSK and Frequency-Domain Equalization with Simplified LLR Generation," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013.

[3] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Transactions on Information Theory*, 2009.

[4] V. Savin, "Non-Binary Polar Codes for Spread-Spectrum Modulations," in *2021 11th International Symposium on Topics in Coding (ISTC)*, 2021.

[5] F. Cochachin, L. Luzzi, and F. Ghaffari, "Reduced Complexity of a Successive Cancellation Based Decoder for NB-Polar Codes," in *2021 11th International Symposium on Topics in Coding (ISTC)*, 2021.

[6] P. Chen, B. Bai, and X. Ma, "Non-Binary Polar Coding with Low Decoding Latency and Complexity," *Journal of Information and Intelligence*, 2022.

[7] F. Cochachin and G. Fakhreddine, "A Lightweight Encoder and Decoder for Non-Binary Polar Codes," in *2023 22nd International Conference on Wireless Networks (ICWN)*, 2023.

[8] E. Boutillon and L. Conde-Canencia, "Simplified Check Node Processing in Non-Binary LDPC Decoders," in *2010 6th International Symposium on Turbo Codes Iterative Information Processing*, 2010.

[9] G. Dillard, M. Reuter, J. Zeiddler, and B. Zeidler, "Cyclic Code-Shift Keying: A Low Probability of Intercept Communication Technique," *IEEE Transactions on Aerospace and Electronic Systems*, 2003.

[10] D. Declercq and M. Fossorier, "Extended Min-Sum Algorithm for Decoding LDPC Codes Over GF(q)," in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, 2005.

[11] C. Marchand and E. Boutillon, "NB-LDPC Check Node with Pre-sorted Input," in *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, 2016.

[12] E. Boutillon and L. Conde-Canencia, "Bubble check: a simplified algorithm for elementary check node processing in Extended Min-Sum non-binary LDPC decoders," *IEE Electronics Letters*, vol. 46, 2010.

[13] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," *IEEE Transactions on Information Theory*, 2010.