Received: 30 July 2024 | Revised: 29 January 2026 | Accepted: 13 July 2026

DOI: 10.33012/navi.792

ORIGINAL ARTICLE

Novel CRC-Aided Shortened Polar Codes for GPS L1C Subframes 2 and 3

Nuwan J. G. Kankanamge[1] | **Nghi H. Tran***[1] | **Khanh Pham**[2] | **Dan Shen**[3] | **Genshe Chen**[3]

[1] Department of Electrical & Computer Engineering, University of Akron, OH, USA

[2] Air Force Research Laboratory, Kirtland Air Force Base, NM, USA

[3] Intelligent Fusion Technology, Inc., Germantown, MD, USA

Correspondence
*Nghi H. Tran
Email: nghi.tran@uakron.edu

Abstract

This paper proposes promising cyclic redundancy check-aided shortened polar codes that can be used to replace the existing low-density parity-check (LDPC) codes in Global Positioning System (GPS) L1C Subframes 2 and 3. To achieve non-power-of-two frame lengths of 1200 and 548 coded bits in Subframes 2 and 3, respectively, we utilize detailed characterizations of the Kronecker construction of polar codes and propose a novel algorithm to identify 848 and 476 shortening positions from polar generator matrices $\boldsymbol{G}_{2048\times2048}$ and $\boldsymbol{G}_{1024\times1024}$. Different from previously proposed shortening schemes, our method exploits a direct connection between the Hamming weights of columns of the polar mother code and the power-of-two decomposition of the number of shortening positions. For a given reliable sequence, the least reliable bit positions suitable for being shortened are generated by sequentially choosing relevant columns of $\boldsymbol{G}$. Extensive simulations are then conducted to demonstrate the competitiveness of the proposed polar coding schemes over the current LDPC codes used in GPS L1C. Over a static additive white Gaussian noise channel, it is shown that by using a successive cancellation list decoder with list size $L = 32$, the proposed scheme provides significant coding gains, ranging from 0.35 dB to 0.5 dB, at the error rate level of 10^{-4} to 10^{-5}, over GPS LDPC in Subframes 2 and 3 at a comparable decoding complexity. Similar gains are also achieved in a three-state fading channel, which accurately characterizes the significant dynamic fading effects observed in land mobile-satellite services expected in urban global navigation satellite system applications. In addition, with a list size of $L = 1024$, our polar coded schemes can closely approach a finite-length information-theoretical limit for error rate levels of 10^{-4} to 10^{-5} and the block-lengths of GPS Subframes 2 and 3. The advantage of the proposed shortening scheme over the traditional shortening scheme is also clearly demonstrated.

Keywords

CRC, GNSS, GPS, L1C Subframes 2 and 3, LMSS, normal approximation, polar codes, three-state fading

Licensed under CC-BY 4.0 © 2026 Institute of Navigation

1 | INTRODUCTION

Modern global navigation satellite systems (GNSSs) are equipped with several forward-error control coding techniques, including low-density parity-check (LDPC) codes, convolutional codes, and Bose–Chaudhuri–Hocquenghem codes. For example, for L1C, the fourth civilian Global Positioning System (GPS) signal, designed to enable interoperability between GPS and international satellite navigation systems, Subframe 2 and Subframe 3 are protected by rate-1/2 irregular LDPC codes with frame sizes of 1200 coded bits and 548 coded bits, respectively (Lu, 2021; Ortega & Poulliat, 2023; Ortega et al., 2020a; US Space Systems Command, 2022). These LDPC coding schemes have been shown to provide superior error performance over other conventional coding counterparts (Lu, 2021). However, despite their widespread use, it is of great importance to explore new coding schemes that have the potential to bypass these LDPC codes in GNSSs.

Over the last decade, significant efforts and interest have focused on polar codes (Arikan, 2009; Bioglio et al., 2021; Vaezi et al., 2022), a class of capacity-achieving codes. With the aid of an outer cyclic redundancy check (CRC) code, CRC-aided polar codes using successive cancellation list (SCL) decoding have been shown to perform exceptionally well in the short and very-short block-length regimes, e.g., a thousand or fewer bits (Bioglio et al., 2021; Chiu & Su, 2023; Tal & Vardy, 2015). In fact, polar codes have been selected in various 5G New Radio (NR) deployment scenarios (3GPP, 2022; Bae et al., 2019; Bioglio et al., 2021; Le et al., 2021). Owing to the nature of the Kronecker construction of polar codes, a conventional polar code must have a code length corresponding to a power of two. Thus, it is not readily available for use in practical applications, including GNSSs. To overcome this limitation, rate-matching schemes such as traditional puncturing or repetition can be applied to achieve an arbitrary code length (Bae et al., 2019; Bioglio et al., 2021). Another interesting rate-matching technique for polar codes is referred to as shortening (Bae et al., 2019; Bioglio et al., 2021). In shortening, shortened bits are chosen from a frozen set. Their values are known to the decoder and, therefore, beneficial to the decoding process. One of the ways to achieve this is to select shortened bits from the end of each frame (Wang & Liu, 2014), and this approach is considered in 5G NR (Bae et al., 2019; Bioglio et al., 2021). However, because later bits in each frame generally correspond to the most reliable positions, this shortening method, while being simple, might not fully utilize the benefits of channel polarization. Other shortening techniques have also been proposed in the literature (Bioglio et al., 2017; Miloslavskaya, 2015; Oliveira & De Lamare, 2021; Saber & Marsland, 2015; Shin et al., 2013; Zhang et al., 2014). However, these approaches are either heuristic or excessively complex, with a lack of structured shortening patterns.

In this paper, we study the feasibility of using polar codes for GPS and make a direct comparison to the existing LDPC codes in terms of both error correction capabilities as well as computational complexity. Toward this goal, we first exploit the detailed structures of a mother polar code $\boldsymbol{G}_{N\times N}$ to propose a novel shortening algorithm that systematically produces a shortening pattern to achieve a non-power-of-two code length. The method can thus be directly applied to GPS Subframes 2 and 3. Because each column of the generator matrix has a Hamming weight of the power of two, our idea is to first decompose the number of shortening positions into a sequence of integers representing its powers of two. Then, for a given reliable sequence, columns of $\boldsymbol{G}$ are sequentially selected based on these integer values, starting from the greatest value, to produce the least reliable bit positions suitable for being shortened. Explicit shortening patterns are obtained for L1C Subframe 2 and Subframe 3. Specifically, for Subframe 2, starting from

a mother polar code of length 2048 bits $\boldsymbol{G}_{2048\times2048}$, 848 shortening positions are generated from four columns of the mother code, resulting in 1200 coded bits for transmission. In the case of Subframe 3, 476 shortening positions are created using six columns of $\boldsymbol{G}_{1024\times1024}$, leaving 548 remaining positions for coded bits.

To provide a comprehensive comparison, we evaluate the error performance of the proposed scheme over both additive white Gaussian noise (AWGN) and three-state fading channels associated with the land mobile-satellite services (LMSSs) (Filho & Moraes, 2021; Ortega & Poulliat, 2023; Tropea & De Rango, 2022). This three-state fading model is general enough to cover the large dynamic ranges observed in fading effects as well as the time-varying channel conditions caused by the motion of vehicles and pedestrians in urban and suburban regions. Our numerical results show that the proposed polar coding schemes significantly outperform the current LDPC codes used in Subframes 2 and 3. Specifically, in both AWGN and three-state fading, the proposed scheme using an SCL decoder with list size $L=32$ provides significant coding gains, ranging from 0.35 dB to 0.5 dB, with a clock and ephemeris data (CED) error rate (CEDER)/frame error rate (FER) ratio of 10^{-4} to 10^{-5} over GPS LDPC code in Subframes 2 and 3 at a comparable decoding complexity. The gain is even larger if ultra-reliable performances with CEDER/FER levels of 10^{-5} or lower are of interest. This gain is possible because the proposed polar code does not suffer from a high error floor as in GPS LDPC codes. Furthermore, we show that the error performance of our schemes in Subframes 2 and 3 can be significantly enhanced by using a larger list size. In particular, with $L=1024$, the proposed polar coded schemes can closely approach a finite-length information-theoretical limit for CEDER/FER levels of 10^{-4} to 10^{-5} and the block-lengths of GPS Subframes 2 and 3. The advantage of the proposed shortening scheme over the traditional shortening scheme is also clearly demonstrated. The results obtained in this work indicate that the proposed polar coding schemes can overtake the existing LDPC codes to provide ultra-reliable performance in the next generations of GNSSs.

2 | POLAR CODES AND SHORTENING FOR GNSS: PRELIMINARIES

2.1 | CRC-Aided Polar Encoding and Shortening for GPS Subframes 2 and 3

As depicted in Figure 1, with CRC-aided polar encoding, each group of A information bits is first appended by C CRC bits, resulting in a block of $K=A+C$ bits. An (N, K) polar encoder will place these K bits in the input sequence $\underline{u}$ of size $1\times N$ based on a so-called reliability sequence, with $N=2^n$ being a power of two. The reliability sequence defines a sequence of bit-channel positions sorted in increasing order of reliability, which can be obtained by several well-known algorithms, such as the Bhattacharyya parameter (Arikan, 2009), Gaussian approximation (Ochiai et al., 2021), and Huawei approximation (3GPP, 2022) algorithms. Based on this sequence, K information bits are allocated to the bit channels with the highest reliability. Let I be the set of indices of these positions. The remaining set of bit channels corresponding to the set of indices I^c can then be used to carry frozen bits, which are assumed to be all zeros. The polar codeword $\underline{x}$ of length $1\times N$ is calculated as follows:

$$\underline{x}=\underline{u}\boldsymbol{G}_{N\times N} \tag{1}$$

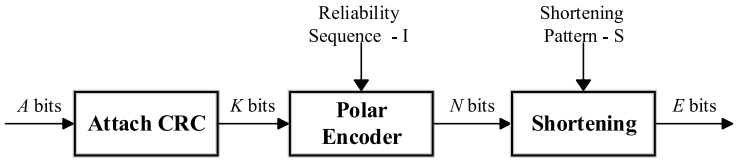

FIGURE 1 A CRC-aided polar encoder with shortening

In Equation (1), the polar transformation matrix $\boldsymbol{G}_{N\times N} = \boldsymbol{G}_{2\times 2}^{\otimes n}$ is n times the Kronecker product of $\boldsymbol{G}_{2\times 2}$, where $\boldsymbol{G}_{2\times 2} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. In applications where the code length E is not a power of two, rate matching via shortening can be used to remove $(N-E)$ bits from $\underline{x}$. Different from puncturing, in shortening, the removed coded bits depend only on frozen bits in $\underline{u}$. A simple way to select these shortened bits is to use weight-1 columns of $G_{N\times N}$ (Bae et al., 2019; Bioglio et al., 2021; Wang & Liu, 2014) starting from the last columns. As we will demonstrate shortly, we can take into account the effect of the reliability sequence and achieve better shortening patterns using other columns of $\boldsymbol{G}_{N\times N}$.

Now, let us consider the use of shortened CRC-aided polar codes in GPS Subframes 2 and 3. For Subframe 2, to ensure exact matching with the current rate-1/2 LDPC and its frame length for a fair comparison, $A = 600$ information bits will be first appended by CRC bits. Although the CRC length can be optimized to further improve the performance (King et al., 2022), we simply adopt an 11-bit CRC 111000100001, resulting in a block of $K = 611$ bits as an input of the polar code $\boldsymbol{G}_{2048\times 2048}$, i.e, we use a (2048, 611) polar code with the respective information set I. Note that while the coded bits in $\underline{c}$ are shortened, the 611 bits (600 source bits plus 11 CRC bits) are allocated to the highest reliability positions in $\underline{u}$, excluding the indices corresponding to the shortening locations. This process defines the information set I, with all other positions—including the shortening indices—set as frozen bits in $\underline{u}$. Next, 848 bits from the 2048 output bits will be shortened to produce a block of $E = 1200$ coded bits for transmission. In a similar manner, for Subframe 3, we have $A = 274$ and $K = 285$, and a (1024, 285) polar code is used. As in Subframe 2, the 285 bits, formed by appending 11 CRC bits to the 274 source bits, are placed in the input vector $\underline{u}$ subject to the constraints imposed by the shortening indices, while all remaining positions are assigned as frozen bits. Then, 476 bits from the 1024 output bits of the polar code are shortened to produce a block of $E = 548$ coded bits.

2.2 | Channel

In the work, we consider the following input/output channel model:

$$Y = \sqrt{\rho}HX + W \tag{2}$$

Here, X is the channel input, which is simply the binary-phase-shift-keying signaling scheme used in the GPS IS-GPS-800J standard (US Space Systems Command, 2022). Furthermore, Y is the channel output, ρ is the average signal-to-noise-ratio (SNR) at the receiver, and W is AWGN with a normalized unit variance. The coefficient H represents the fading gain with a unit variance that follows a fading distribution $f_H(h)$. In the case of static AWGN, $H = 1$ and remains constant. For the fading case, we consider LMSSs in which the dynamics

of the propagation environment and the fluctuations due to atmospheric effects and territory orography can be accurately modeled as a finite-state Markov chain (FSMC) (Sadeghi et al., 2008; Tropea & De Rango, 2022). An FSMC model is characterized by a transition probability matrix P, a state duration T_s, the number of states in the process, and the probability distribution in each state. In the case of a three-state model, the probability transition matrix P is given as follows:

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \tag{3}$$

Here, p_{ij} represents the probability that the state changes from the current state i to the next state j. When the number of state transitions N reaches infinity, the state probability $\pi_i = \lim_{N\to 0} N_i/N$ will be saturated. Note that N_i is the number of times in a given state $i, i = 1, 2, 3$. The steady-state transition probabilities satisfy $\pi P = \pi$, where $\pi = (\pi_1, \pi_2, \pi_3)$. We assume that the state transition matrix of the Markov chain is time-homogeneous, i.e., the probability transition matrix is constant over time. Note that $\sum_i \pi_i = 1$. Now, let the fading distribution at three states be denoted as $f_{H_i}(h)$, where $i = 1, 2, 3$. Because we consider a constant state duration T_s for each transition time between states, the distribution of the fading H can be written as the following mixture (Karasawa et al., 1997):

$$f_H(h) = \sum_{i=1}^{3} \pi_i f_{H_i}(h) \tag{4}$$

Among various three-state LMSS models, the most realistic model consists of state one as direct line of sight (LOS), state two as a low-shadowing condition, and state three as a high-shadowing condition where only multipath components are available at the receiver (Tropea & De Rango, 2022). Furthermore, in urban environments, the commonly used distributions in these three states are Rician, log-normal/Loo, and Rayleigh, respectively. In this work, we consider three states as Rician, log-normal, and Rayleigh distributions, with the following probability density functions:

$$f_{H_1}(h) = f_{rice}(h, \sigma_1) = \frac{h}{\sigma_1^2} \exp\left(-\frac{h^2 + v^2}{2\sigma_1^2}\right) I_0\left(\frac{hv}{\sigma_1^2}\right) \tag{5a}$$

$$f_{H_2}(h) = f_{log}(h, \sigma_2, \mu) = \frac{1}{h\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(\ln(h) - \mu)^2}{2\sigma_2^2}\right) \tag{5b}$$

$$f_{H_3}(h) = f_{ray}(h, \sigma_3) = \frac{h}{\sigma_3^2} \exp\left(-\frac{h^2}{2\sigma_3^2}\right) \tag{5c}$$

Note that in Rician fading, v controls the relative strength of the LOS signal and σ_1 controls the overall variability around the v. In addition, the K factor is defined as the ratio $v^2/2\sigma_1^2$. In log-normal fading, μ and σ_2 are equivalent to the characterization of v and σ_1 in Rician fading. However, these terms are on a logarithmic scale. Finally, σ_3 in Rayleigh fading indicates the variability of the signal over zero mean. Under the assumption of unit variance, it can be verified that $2\sigma_1^2 + v^2 = 1, e^{2\mu + 2\sigma_2^2} = 1$, and $2\sigma_3^2 = 1$.

To maintain consistency with the GPS standards, we adopt an interleaving model that utilizes a symbol interleaver spanning the entire subframe (Ortega et al., 2020a; US Space Systems Command, 2022). This approach yields an ergodic fading channel in which instantaneous fading coefficients vary independently from symbol to symbol, following the three-state fading distribution defined in Equation (4). We would like to note that in the absence of an interleaver, a non-ergodic block fading model may be considered (Ortega et al., 2020a). In this case, the three-state fading channel is characterized by the channel state duration T_s and the coherence duration T over which the channel coefficients remain constant.

2.3 | Polar Decoding with CRC-Aided SCL

SCL decoding (Tal & Vardy, 2015) can be used to decode a CRC-aided polar code, which improves the performance of short and moderate block-lengths under the original successive cancellation (SC) decoding algorithm (Arikan, 2009). The SCL decoding algorithm considers the maximum number of L parallel SC decoders at a given decoding instance. Different from other powerful codes such as LDPC and turbo codes, SC- and SCL-based decoders are implemented via a graph representation of $G_{N\times N}$. Because the frozen bits are all zeros, including the shortening pattern S, their log-likelihood ratios can be set to minus infinity and used in SCL decoding. At the output, a single codeword is selected based on the path matrix, which is updated at every transverse in the decoding tree and CRC check. It should be mentioned that when the list size L is sufficiently large, SCL decoding almost matches the performance of maximum-likelihood (ML) decoders (Baicheva & Kazakov, 2019) at the trade-off of higher decoding complexity. As a new class of codes, efficient decoding methods for polar codes are still evolving. For example, by exploiting properties of the mutual information metric, a so-called selective-path-extension method can be used to further reduce the complexity without affecting the FER (Huawei & HiSilicon, 2016).

3 | PROPOSED SHORTENING SOLUTION

In shortening, the shortening locations in the codeword correspond to frozen bit positions in the encoder input vector $\underline{u}$. That is, bits associated with shortening positions are frozen to zero in $\underline{u}$ and therefore do not carry source information. As we outline in Appendix 7.1, the existing method adopted in 5G NR uses weight-1 columns of the generator matrix, starting from the last columns, to select the last $N-E$ indices for shortening. While this method's simplicity facilitates the variable code lengths required by the standard, it sacrifices the most reliable bit positions in $\underline{u}$, as later input bits are generally more reliable. Because searching over all possible shortening patterns is prohibitively complex, identifying an optimal pattern that sacrifices fewer high-reliability bits still remains a challenging task.

With the objective of preserving as many highly reliable positions in $\underline{u}$ as possible for placing the information bits while simultaneously reducing computational complexity, we develop a novel algorithm to identify a new shortening pattern S for CRC-aided polar codes, with a particular emphasis on the code configurations used for Subframes 2 and 3. Toward this end, in the following, we first study several properties of the mother code $\boldsymbol{G}_{N\times N}$. These properties are helpful in the development of the algorithm.

3.1 | Properties of $G_{N\times N}$

Let us start with the first property of $\boldsymbol{G}_{N\times N}$, which is stated in the following lemma.

Lemma 1. *Consider the m-th column of* $\boldsymbol{G}_{N\times N}$. *For any pair of indices* p *and* q, *if* $G_{pm}=1$ *and* $G_{qm}=0$, *then* $G_{qp}=0$. *This property is illustrated in Figure 2.*

Proof. Let $\left(b_1,b_2,\ldots,b_{n-1},b_n\right),\left(d_1,d_2,\ldots,d_{n-1},b_n\right)$, and $\left(e_1,e_2,\ldots,e_{n-1},e_n\right)$ be the n-bit representations of $(m-1)$, $(p-1)$, and $(q-1)$, respectively. Recall that $2^n=N$. Then, from Arikan (2009), we have the following:

$$G_{pm}=G_{d_1\ldots d_n,b_1\ldots b_n}=\prod_{i=1}^{n}\left(1\oplus b_i\oplus b_i d_i\right)=1 \tag{6}$$

and:

$$G_{qm}=G_{e_1\ldots e_n,b_1\ldots b_n}=\prod_{i=1}^{n}\left(1\oplus b_i\oplus b_i e_i\right)=0 \tag{7}$$

where $\oplus$ is the XOR operation. From Equation (6), it is clear that the following holds:

$$1\oplus b_i\oplus b_i d_i=1 \tag{8}$$

for all i. Then, by combining Equations (7) and (8), we have the following:

$$G_{qm}=\prod_{i=1}^{n}\left(1\oplus b_i\left(1\oplus b_i\oplus b_i d_i\right)\oplus b_i e_i\left(1\oplus b_i\oplus b_i d_i\right)\right)=\prod_{i=1}^{n}\left(1\oplus b_i d_i\oplus b_i e_i d_i\right)=0 \tag{9}$$

From Equation (9), we can see that there exists at least one index k such that $1\oplus b_k d_k\oplus b_k e_k d_k=0$, which results in $b_k=1$, $d_k=1$, and $e_k=0$. Thus, $1\oplus d_k\oplus e_k d_k=0$. Therefore, we have the following:

$$G_{qp}=\prod_{i=1}^{n}\left(1\oplus d_i\oplus e_i d_i\right)=0 \tag{10}$$

□

From Lemma 1, we can conclude that for a given m and p with $G_{pm}=1$, the m-th and p-th columns of $G_{N\times N}$ will have the same zero positions. Furthermore, a

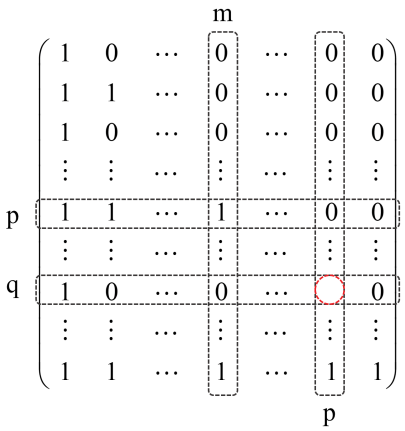

FIGURE 2 A property of $\boldsymbol{G}_{N\times N}$

column created by the bit-wise OR of these two columns will contain zeros in the same positions of column m. The next lemmas will state more specific properties of the bit-wise OR and AND for the rows and columns of $G_{N\times N}$.

Lemma 2. *Consider any set of k rows $(\boldsymbol{g}_{x_1}, \boldsymbol{g}_{x_2}, \ldots, \boldsymbol{g}_{x_k})$ of $G_{N\times N}$, and let $\left(b_1^i, b_2^i, \ldots, b_{n-1}^i, b_n^i\right)$ be the n-tuple binary representation of each index x_i. Furthermore, let $\boldsymbol{v} = \wedge_{i=1}^{k}[\boldsymbol{g}_{x_i}]$ and $\boldsymbol{u} = \vee_{i=1}^{k}[\boldsymbol{g}_{x_i}]$, where $\wedge$ and $\vee$ are the element-wise AND and OR operations, respectively. The Hamming weight of $\boldsymbol{v}$ and $\boldsymbol{u}$, denoted as $H_{\boldsymbol{v}}$ and $H_{\boldsymbol{u}}$, can be calculated as follows:*

- $H_{\boldsymbol{v}} = 2^{\sum_{j=1}^{n}\left(\prod_{i=1}^{k} b_j^i\right)}$
- $H_{\boldsymbol{u}} = \sum_{J\subseteq\{1,\ldots,k\}} (-1)^{|J|+1} H_{\bigwedge_{j\in J}\left[\boldsymbol{g}_{x_j}\right]}$, *where the summation is taken over all possible subsets $\{J\}$ of $\{1,\ldots,k\}$*

Proof. From Arikan (2009), for a given row g_{x_i}, we have the following:

$$H_{g_{x_i}} = 2^{\sum_j b_j^i} \tag{11}$$

By applying the same methodology across multiple rows, we can express $H_{\boldsymbol{v}}$ as follows:

$$\begin{aligned} H_{\boldsymbol{v}} = 1 &+ \bigwedge_{i=1}^{k}\left[b_n^i\right] + \bigwedge_{i=1}^{k}\left[b_{n-1}^i\right] + \bigwedge_{i=1}^{k}\left[b_{n-1}^i b_n^i\right] + \ldots \\ &+ \bigwedge_{i=1}^{k}\left[b_1^i b_2^i \ldots b_{n-1}^i\right] + \bigwedge_{i=1}^{k}\left[b_1^i b_2^i \ldots b_{n-1}^i b_n^i\right] \end{aligned} \tag{12}$$

Because $x \wedge y$ is the same as the regular product xy for binary x and y, $H_{\boldsymbol{v}}$ can be re-written as follows:

$$\begin{aligned} H_{\boldsymbol{v}} &= 1 + \prod_{i=1}^{k} b_n^i + \prod_{i=1}^{k} b_{n-1}^i + \prod_{i=1}^{k} b_{n-1}^i b_n^i + \ldots + \prod_{i=1}^{k} b_1^i b_2^i \ldots b_{n-1}^i + \prod_{i=1}^{k} b_1^i b_2^i \ldots b_{n-1}^i b_n^i \\ &= \prod_{j=1}^{n}\left(1 + \prod_{i=1}^{k} b_j^i\right) = \prod_{j=1}^{n} 2^{\prod_{i=1}^{k} b_j^i} = 2^{\sum_{j=1}^{n}\left(\prod_{i=1}^{k} b_j^i\right)} \end{aligned} \tag{13}$$

To calculate $H_{\boldsymbol{u}}$, consider again row $\boldsymbol{g}_{x_i}$. Let A_{x_i} be the set of column indices $\{m\}$ such that $g_{x_i m} = 1$. It is not difficult to verify that the cardinality, denoted as $|\cdot|$, of the set $\left(\bigcup_{i=1}^{k} A_{x_i}\right)$ is $H_{\boldsymbol{u}}$. In other words, we have the following:

$$\left|\bigcup_{i=1}^{k} A_{x_i}\right| \equiv H_{\bigvee_{i=1}^{k}\left[\boldsymbol{g}_{x_i}\right]} = H_{\boldsymbol{u}} \tag{14}$$

From the inclusion–exclusion principle (Brualdi, 2009), the number of elements in the union $\bigcup_{i=1}^{k} A_{x_i}$ can be expressed as follows:

$$\left|\bigcup_{i=1}^{k} A_{x_i}\right| = \sum_{\varnothing\neq J\subseteq\{1,\ldots,k\}} (-1)^{|J|+1} n \left|\bigcap_{j\in J} A_{x_j}\right| \tag{15}$$

Furthermore, for a given set J, it is apparent that $\left|\bigcap_{j\in J} A_{x_j}\right| = H_{\underset{j\in J}{\wedge}\left[\boldsymbol{g}_{x_j}\right]}$. Therefore, we have the following:

$$H_u = \sum_{\varnothing\neq J\subseteq\{1,\ldots,k\}} (-1)^{|J|+1} H_{\underset{j\in J}{\wedge}\left[\boldsymbol{g}_{x_j}\right]} \tag{16}$$

□

Lemma 3. *Consider any positive integer* $P<2^n$ *with its power-of-two decomposition* $P=2^{p_1}+2^{p_2}+\ldots+2^{p_{k-1}}+2^{p_k}$, $p_1>p_2>\ldots>p_{k-1}>p_k$. *Starting with the first iteration, we choose a row (column) of* $G_1=G_{N\times N}$ *with Hamming weight* 2^{p_1}. *Iteratively, in each iteration i, where* $2\le i\le k$, *we generate an* $N\times N$ *matrix* G_i *by performing an element-wise OR between* $G_{N\times N}$ *and the previously chosen row (column) of* G_{i-1} *of Hamming weight* $\sum_{z=1}^{i-1} 2^{p_z}$. *In* G_i, *there exists a row (column) with a Hamming weight of* $\sum_{z=1}^{i} 2^{p_z}$.

Proof. We first select a row $\boldsymbol{r}_{x_1}$ of Hamming weight 2^{p_1}, with $(b_1^1, b_2^1, \ldots, b_{n-1}^1, b_n^1)$ being the n-tuple binary representation of index x_1. Let B_{x_1} be the set of bit position indices $\{j\}$ such that $b_j^1=1$. Following Equation (11), the number of elements in B_{x_1} is thus p_1. Next, let us consider two new sets, $S_1=B_{x_1}$ and $C_1=\varnothing$. In the i-th iteration starting from $i=2$, we have $S_{i-1}=\left(\bigcap_{z=1}^{i-1} B_{x_z}\right)$ and choose a set C_i that satisfies the following:

$$C_i \in \left(S_{i-2}\cap\overline{S_{i-1}}\right)$$

Note that $S_0=\{1,2,\ldots,n\}$. Now, we choose any set S_i that is a subset of S_{i-1} under the condition that $|S_i|=p_i$. It should be mentioned that we have $\binom{p_{i-2}-p_{i-1}}{1}$ options to select C_i and $\binom{p_{i-1}}{p_i}$ options to select S_i. A new n-tuple set is formed as follows:

$$B_{x_i}=\left(S_i\cup\left(\bigcup_{z=1}^{i}\{C_z\}\right)\right)$$

where x_i is the decimal conversion of the set. Note that there are $\binom{p_{i-1}}{p_i}\times\binom{p_{i-2}-p_{i-1}}{1}$ options to select B_{x_i}. Furthermore, it can be verified that $S_i=\left(\bigcap_{z=1}^{i} B_{x_z}\right)$. In the following, using the induction method, we will show that row $\boldsymbol{r}_{x_i}$ of G_i is the solution. Note that $\boldsymbol{r}_{x_i}=\underset{z=1}{\overset{i}{\vee}}\left[\boldsymbol{g}_{x_z}\right]$.

In iteration $(i-1)$, let us assume that we have the following:

$$H_{\boldsymbol{r}_{x_{i-1}}}=H_{\underset{z=1}{\overset{i-1}{\vee}}\left[\boldsymbol{g}_{x_z}\right]}=\sum_{z=1}^{i-1}2^{p_z} \tag{17}$$

Using Lemma 2, in iteration i, we write $H_{\boldsymbol{r}_{x_i}}$ as follows:

$$H_{\underset{z=1}{\overset{i}{\vee}}\left[\boldsymbol{g}_{x_z}\right]}=\sum_{\varnothing\neq J\subseteq\{1,\ldots,i-1\}}(-1)^{|J|+1}H_{\underset{j\in J}{\wedge}\left[\boldsymbol{g}_{x_j}\right]}+\sum_{\varnothing\neq J\subseteq\{1,\ldots,i\},\, i\in J}(-1)^{|J|+1}H_{\underset{j\in J}{\wedge}\left[\boldsymbol{g}_{x_j}\right]} \tag{18}$$

The first term on the right-hand side of Equation (18) is equal to $H_{\bigvee_{z=1}^{i-1}\left[\mathbf{g}_{x_z}\right]}$. The second term on the right-hand side of Equation (18) can be expressed as follows:

$$\sum_{\varnothing\neq J\subseteq\{1,\ldots,i\},\, i\in J}(-1)^{|J|+1}H_{\bigwedge_{j\in J}\left[\mathbf{g}_{x_j}\right]}=H_{\left[\mathbf{g}_{x_i}\right]}-\sum_{J\subseteq\{1,\ldots,i-1\},|J|=1}H_{\mathbf{g}_{x_i}\wedge\left[\bigwedge_{j\in J}\left[\mathbf{g}_{x_j}\right]\right]}$$
$$+\sum_{J\subseteq\{1,\ldots,i-1\},|J|=2}H_{\mathbf{g}_{x_i}\wedge\left[\bigwedge_{j\in J}\left[\mathbf{g}_{x_j}\right]\right]}+\ldots+\sum_{J\subseteq\{1,\ldots,i-1\},|J|=i-1}(-1)^{|J|+1}H_{\mathbf{g}_{x_i}\wedge\left[\bigwedge_{j\in J}\left[\mathbf{g}_{x_j}\right]\right]}\Bigg] \quad (19)$$

Using the property of H_v in Lemma 2, we further obtain the following:

$$\sum_{\varnothing\neq J\subseteq\{1,\ldots,i\},\, i\in J}(-1)^{|J|+1}H_{\bigwedge_{j\in J}\left[\mathbf{g}_{x_j}\right]}=2^{\left|B_{x_i}\right|}-\sum_{J\subseteq\{1,\ldots,i-1\},|J|=1}2^{\left|B_{x_i}\cap\left(\bigcap_{j\in J}B_{x_j}\right)\right|}$$
$$+\sum_{J\subseteq\{1,\ldots,i-1\},|J|=2}2^{\left|B_{x_i}\cap\left(\bigcap_{j\in J}B_{x_j}\right)\right|}+\ldots+\sum_{J\subseteq\{1,\ldots,i-1\},|J|=i-1}(-1)^{|J|+1}2^{\left|B_{x_i}\cap\left(\bigcap_{j\in J}B_{x_j}\right)\right|} \quad (20)$$

As we show in Appendix 7.2, $\left|B_{x_i}\cap\left(\bigcap_{j\in J}B_{x_j}\right)\right|=p_i+(i-1)-|J|$. The following then holds:

$$H_{\bigvee_{z=1}^{i}\left[\mathbf{g}_{x_z}\right]}=H_{\bigvee_{z=1}^{i-1}\left[\mathbf{g}_{x_z}\right]}+2^{p_i+(i-1)}-\binom{i-1}{1}2^{p_i+(i-2)}+\binom{i-1}{2}2^{p_i+(i-3)}+\ldots+\binom{i-1}{i-1}2^{p_i}$$
$$=\sum_{z=1}^{i-1}2^{p_z}+2^{p_i}[2-1]^{i-1}=\sum_{z=1}^{i}2^{p_z} \quad (21)$$

Because $G_{N\times N}$ is symmetric along the anti-diagonal axis, the result in Equation (21) also holds when we use column operations. □

As we demonstrate next, the results in Lemma 1 facilitate the determination of feasible shortening locations through the analysis of a single column of the generator matrix, thereby eliminating the need to examine all columns associated with the shortening positions. This approach substantially reduces computational overhead in the shortening design process and, at the same time, allows us to generate multiple shortening patterns for further optimizing shortening positions. In addition, Lemmas 2 and 3 establish the theoretical foundation by proving the existence and applicability of the proposed algorithm for any combination of code parameters and shortening lengths. These features are particularly important given that Subframes 2 and 3 necessitate distinct code parameters and shortening configurations.

3.2 | Proposed Shortening Algorithm

We now develop our algorithm. First, it is noted that if we want to shorten an output bit at location i, our set of frozen bits I^c must include all of the row indices of column i at which the bits are ones (Wang & Liu, 2014). Therefore, our approach is to choose $(N-E)$ shortening positions that are determined from exactly $(N-E)$ frozen bits in such a way that the frozen bits are distributed to the least reliable channels for a given reliable sequence. While it is not possible to simultaneously select these $(N-E)$ locations, we first decompose $(N-E)$ into powers of two. This allows us to select, in each step, each of the power-of-two frozen locations, starting from the greatest value, using columns of $G_{N\times N}$. In each of these steps, we can

ALGORITHM 1
Pseudocode of the shortening algorithm

```
input  : N, E, K, W
output : ShortBitLoc in x, InfoBitLoc in u
BinDecom ← GetBinaryDecom(N − E) SelCol ← 0;      // Initialize with zero vector
TargetWeight ← 0;
for i ← 1 to length of BinDecom do
    TargetWeight ← TargetWeight + BinDecom(i);
    TempMat ← OR(G_i, SelCol);                    // Bit-wise OR with columns of G_i
    MaxSum ← 0;
    for j ← 1 to N do
        if Weight(TempMat (j)) == TargetWeight then
            if SumInfoWeight(TempMat (j), W) > MaxSum then
                MaxSum ← SumInfoWeight(TempMat (j), W);
                SelCol ← TempMat (j);
            end
        end
    end
end
ShortBitLoc ← SelCol;
InfoBitLoc ← GetInfBitLoc(ShortBitLoc, W, K)
```

rank and compare the locations using the reliable sequence, e.g., for each choice, we can calculate the summation of the corresponding reliability sequence values and choose the one that corresponds to the least possible sum. For convenience, let us decompose $(N-E)$ to its powers of two as $(N-E)=2^{p_1}+2^{p_2}+\ldots.+2^{p_k}$ with $p_1>p_2>\ldots>p_k$. The proposed algorithm can then be implemented as follows:

(i) Find the column of $G_{N\times N}(=G_1)$ with a Hamming weight of 2^{p_1} that has the largest summation of reliability scores of "0" locations. This column is denoted as $g^{(1)}$.

(ii) Perform bit-wise OR operations between $g^{(1)}$ and each of N columns of $G_{N\times N}$. The resulting N arrays form an $N\times N$ matrix, which is denoted as G_2.

(iii) Select the column $g^{(2)}$ of G_2 with a Hamming weight of $2^{p_1}+2^{p_2}$ that is ranked the best among all columns of Hamming weight $2^{p_1}+2^{p_2}$. From Lemma 3, there exists at least one column with this Hamming weight in G_2.

(iv) Repeat steps (ii) and (iii) k times until we reach a Hamming weight of $(N-E)=2^{p_1}+2^{p_2}+\ldots.+2^{p_k}$. Note that a new matrix G_i is calculated in each iteration i. In the k-th iteration, we obtain the final column $g^{(k)}=g_{shorten}$.

(v) The row indices of all bits 1 in $g_{shorten}$ correspond to the shortened positions in $\underline{x}$.

(vi) It is noted that the result in Lemma 1 guarantees that these k selected columns from $G_{N\times N}$ will have the same "0" row indices as in $g_{shorten}$. Hence, information bits can be positioned in $\underline{u}$ corresponding to the row indices of bits 0 in $g_{shorten}$.

The pseudocode of the algorithm is given in Algorithm 1.

To apply the algorithm to Subframes 2 and 3, we first adopt the 3rd Generation Partnership Project (3GPP) reliability sequence (3GPP, 2022), which is independent of the SNR. For Subframe 2, we must identify 848 shortening positions from $G_{2048\times2048}$. Because $848 = 2^9 + 2^8 + 2^6 + 2^4$, the algorithm proceeds iteratively with $k = 4$. It sequentially selects the 4th, 6th, 26th, and 106th columns of the updated matrices G_i for $i = 1, \ldots, 4$, respectively, initialized with $G_1 = G_{2048\times2048}$. The final shortening pattern is derived from $g_{shorten} = g^{(4)}$, which is the 106th column of G_4, where the indices of the bit 1s correspond to the shortening positions. In a similar manner, for Subframe 3, we shorten 476 bits. Decomposing this length yields $476 = 2^8 + 2^7 + 2^6 + 2^4 + 2^3 + 2^2$. Consequently, the algorithm runs iteratively $k = 6$ times, selecting the 4th, 6th, 10th, 50th, 82nd, and 146th columns of matrices G_i for $i = 1, \ldots, 6$, respectively, initialized with $G_1 = G_{1024\times1024}$. The final shortening positions are obtained from the resulting array $g_{shorten} = g^{(6)}$, which is the 146th column of G_6.

4 | COMPARISON OF PROPOSED POLAR CODES AND GPS LDPC CODES

In this section, numerical results are presented to compare the error correction capability as well as the computational complexity of the proposed polar coding schemes under SCL decoding over GPS LDPC codes in Subframes 2 and 3. Unless otherwise stated, for polar codes, the outer 11-bit CRC with the generator polynomial 111000100001 is used, and a list size of $L = 32$ is adopted for SCL. Meanwhile, the same LDPC codes from the GPS IS-GPS-800J standard (US Space Systems Command, 2022) with 20 belief propagation iterations are used. For these LDPC codes, further increasing the number of iterations results in negligible improvements.

In all results, the error performance is plotted versus C/N_0, where C denotes the carrier power of the navigation signal received at the antenna output and N_0 is the one-sided power spectral density of the AWGN. The metric C/N_0 is relevant to GNSS standards and provides clear insight into the system's practical impact (US Space Systems Command, 2022). According to the GPS L1C interface specifications (US Space Systems Command, 2022), Subframe 2 carries 600 bits of CED, which includes 24-bit CRC for error detection. This CED information is critical for computing position fixes. A CED error is declared whenever the CRC-24 check fails. Thus, for Subframe 2, we use the CEDER, which is defined as the number of incorrect CED messages over the total number of CED messages transmitted (US Space Systems Command, 2022), as the main metric. Subframe 3 carries 274 bits, including 24-bit CRC, and provides auxiliary navigation data (i.e., not CED data) across multiple pages. Thus, we refer to the error rate of Subframe 3 as the FER, which is the ratio of the number of incorrect frames to the total number of transmitted frames.

4.1 | Error Performance Comparison Under AWGN

We first consider a static AWGN channel. Figure 3 presents the CEDER performance of the various coding schemes applied to Subframe 2. As a reference, the performance of the polar coding scheme using the 5G NR shortening pattern in which shortened bits are chosen from the end of each frame (Wang & Liu, 2014) is also provided. It can be seen from Figure 3 that the proposed polar coding scheme

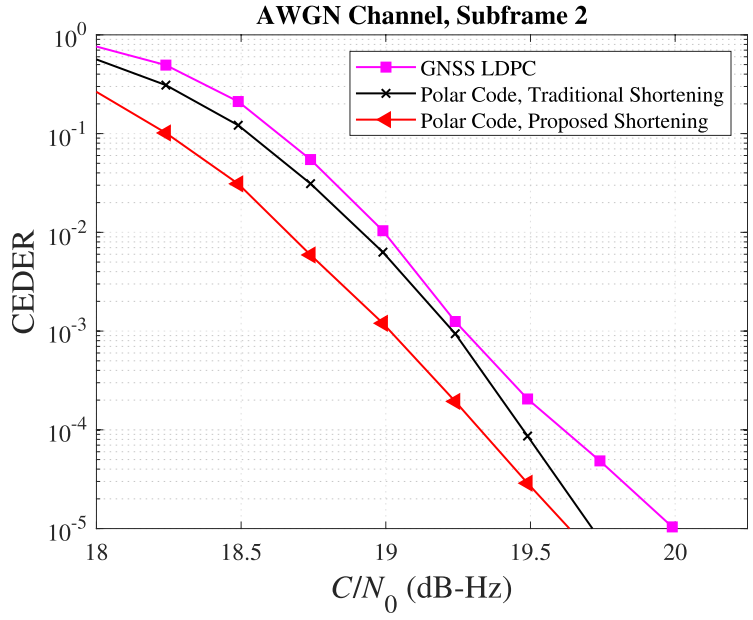

FIGURE 3 CEDER versus C/N_0 of polar and LDPC codes under AWGN in Subframe 2

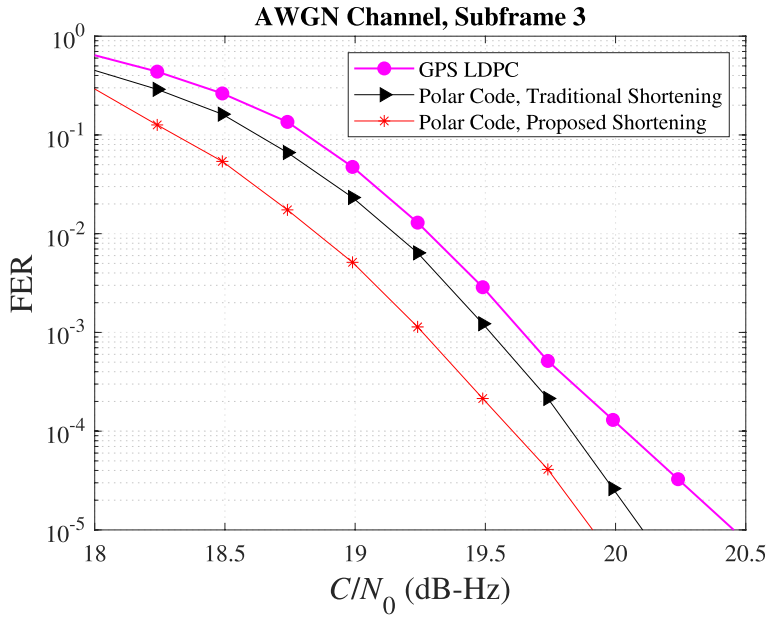

FIGURE 4 FER versus C/N_0 of polar and LDPC codes under AWGN in Subframe 3

outperforms the GPS LDPC code. At a practical CEDER of 10^{-4}, the proposed shortening method achieves a significant coding gain of 0.3 dB over LDPC. If ultra-reliable performances with CEDER levels of 10^{-5} or lower are of interest, additional coding gains, i.e., 0.4 dB or more, can be attained by our polar coding scheme. This improvement can be achieved because, at a sufficiently high C/N_0, the considered polar code does not suffer from a high error floor as in the GPS LDPC code. The superiority of our shortening scheme over the traditional shortening scheme is also clear over a wide range of C/N_0.

The same advantages of the proposed shortening scheme over GPS LDPC codes can also be attained in Subframe 3. As shown in Figure 4, the use of CRC-aided polar code (548,285) provides coding gains of 0.35 dB and 0.55 dB at FERs of

10^{-4} and 10^{-5}, respectively. It should also be noted that, similar to the results for Subframe 2, the proposed shortening scheme outperforms the traditional shortening scheme considered in 5G NR.

4.2 | Error Performance Comparison Under Three-State Fading

In the case of three-state fading, we consider a model in which state-one fading is modeled as Rician with $K = 5,$ while state-two fading follows a log-normal distribution with $\mu = -1/8, \sigma_2 = \sqrt{1/8}$. For state-three Rayleigh fading, we assume $\sigma_3 = \sqrt{1/2}$. In addition, we assume that the state transition matrix P_1 is given as follows:

$$P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0.65 & 0 & 0.35 \\ 0 & 1 & 0 \end{pmatrix} \tag{22}$$

As established in Section 2.2, an interleaved model is used, resulting in ergodic fading, where the channel gains are assumed to change independently from symbol to symbol according to the three-state fading model defined in Equation (4).

Performance comparisons between the proposed polar codes and the LDPC codes over the three-state fading channel are presented in Figures 5 and 6 for Subframe 2 (CEDER) and Subframe 3 (FER), respectively. As a reference, the FER/CEDER attained by the traditional shortening pattern used in 5G NR is also presented. As in the AWGN channel, for an FER/CEDER ratio of 10^{-4}, the proposed scheme yields a notable coding gain of 0.3 dB to 0.4 dB over the LDPC codes. The gains are larger at sufficiently high C/N_0 owing to the error floor behavior experienced in the LDPC-based systems.

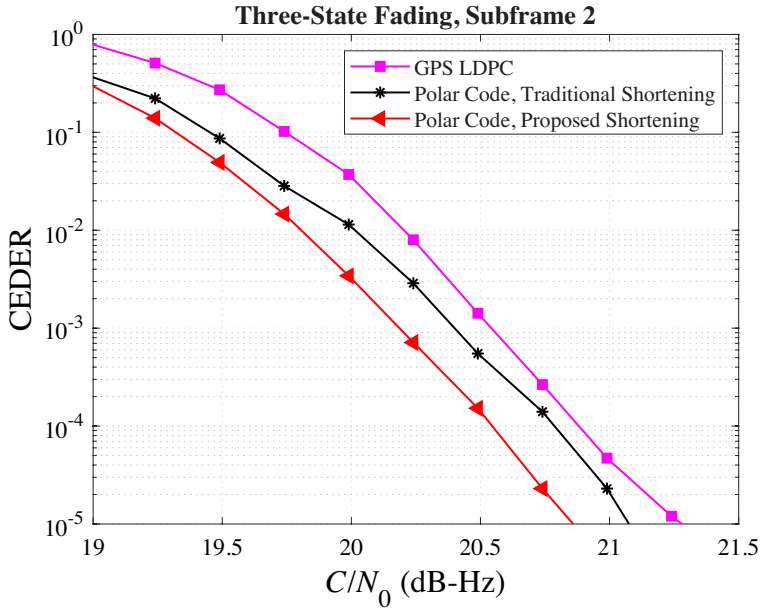

FIGURE 5 CEDER versus C/N_0 of polar and LDPC codes under three-state fading in Subframe 2

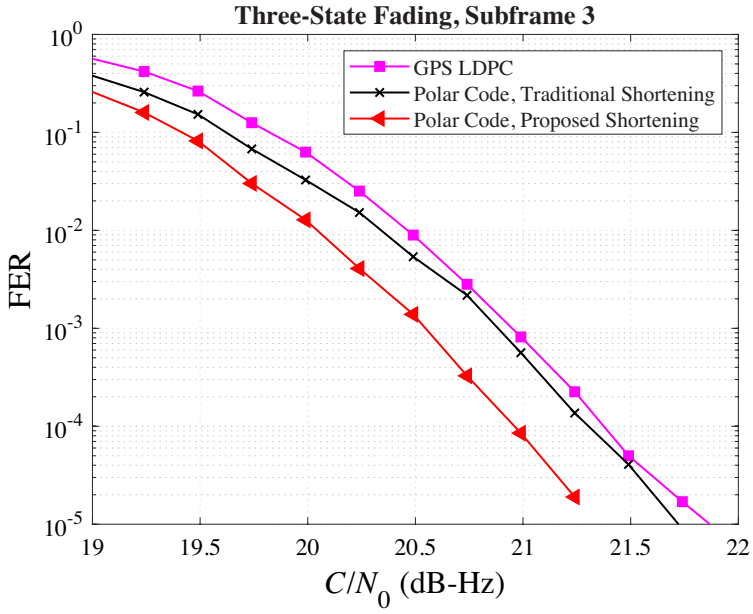

FIGURE 6 FER versus C/N_0 performance of polar and LDPC codes under three-state fading in Subframe 3

4.3 | Computational Complexity Comparison

Given the competitive error performances of the proposed polar codes, it is important to compare their implementation complexity over the well-established GPS LDPC codes. While several metrics are involved, in the following, we focus on the computational decoding complexity, which is an important parameter that directly affects the implementation.

For LDPC code, the number of additional operations or equivalent to decode a frame is given as follows (Huawei & HiSilicon, 2016; Nokia & Alcatel-Lucent, 2016):

$$C_{\text{LDPC}} = I_{\max}\left(2Nd_v + M\left(2d_c - 1\right)\right) + I_{\max}Md_v$$

where $I_{\max}$ is the number of iterations used, N is the code length, and M is the number of parity bits. In addition, d_v is the average variable degree, and d_c is the average check degree. Note that for Subframe 2, we have $I_{\max} = 20, N = 1,200$, and $M = 600$, whereas $d_v = 4.02$ and $d_c = 8.03$, which are calculated from the parity check matrix given by US Space Systems Command (2022). Similarly, for Subframe 3, we have $I_{\max} = 20, N = 548, M = 274, d_v = 3.78$, and $d_c = 7.56$.

The computational complexity of an SCL polar decoder comes from two main components: successive cancellation and path selection. Without using any reduction method, the number of operations is calculated as follows:

$$C^1_{\text{Polar}} = L * N * \log(N) + L(N-1) + K * 2 * L * \log(2L)$$

where L is the list size, N is the polar code size, and K is the number of information bits. In the case of Subframe 2, $L = 32$, $N = 2048$, and $K = 600$; for Subframe 3, $L = 32, N = 1024$, and $K = 274$. If the selective-path-extension method (Huawei & HiSilicon, 2016) is implemented, the number of operations decreases to the following:

$$C^2_{\text{Polar}} = L * N * \log(N) + L(N-1) + 0.5KL\log(2L) \tag{23}$$

TABLE 1
Computational Complexity in Terms of Decoding Operations: Proposed Polar Code Versus LDPC Code

	C_{LDPC}	C^1_{Polar}	C^2_{Polar}
Subframe 2	951,600	1,016,800	844,000
Subframe 3	408,720	465,632	386,720

Note that this method does not affect the error performance (Huawei & HiSilicon, 2016).

Table 1 shows $C_{\text{LDPC}}, C^1_{\text{Polar}}$, and C^2_{Polar} for both Subframes 2 and 3. It is obvious that the two coding schemes have a comparable computational complexity. It is worth mentioning that for polar codes, a further reduction in decoding complexity can be achieved by using more advanced schemes, such as frozen-header-skipping and list-reduced decoders (Huawei & HiSilicon, 2016).

5 | PROPOSED POLAR CODES: INFLUENCE OF LIST SIZE AND APPROACHING FUNDAMENTAL LIMITS

While the coding gains achieved by the proposed polar codes using a list size of 32 over the LDPC codes in Subframes 2 and 3 are significant, it is also of interest to examine how the performance of these schemes evolves with the list size of the SCL decoder and how closely they approach the corresponding information-theoretical limits. For the finite block-lengths we have in Subframes 2 and 3, the limits based on Shannon/constrained capacities no longer hold. Instead, we must find the maximal achievable rate for a given error probability and block-length. Alternatively, we can assess the dependence of the error probability on the block-length for a given transmission rate. Under this line of research, sharp finite-length bounds, which serve as important benchmarks to design finite-length coding schemes, have only been obtained recently (Polyanskiy et al., 2010). Among various finite-length metrics such as the converse bound and the achievability bound (Polyanskiy et al., 2010), a so-called normal approximation (NA) has been considered as an attractive alternative to other tight non-asymptotic bounds (Huang et al., 2021; Polyanskiy & Verdú, 2011; Polyanskiy et al., 2010; Yang et al., 2013, 2018). In fact, the NA derived using the central limit theorem has been considered as a simple tool to accurately approximate these bounds (Durisi & Lancho, 2020; Polyanskiy et al., 2010). For completeness, the calculation of the NA is outlined in Appendix 7.3. Under the context of error correction and information-theoretical limits, the link-quality measure in terms of E_b/N_0, where E_b denotes the energy per information bit, is more commonly used to reflect coding gain benefits (Anghileri et al., 2013; Yang et al., 2018). Consequently, in this section, E_b/N_0 is used unless otherwise stated. The relationship between C/N_0 and E_b/N_0 can be written as $C/N_{0\,\text{dB–Hz}} = E_b/N_{0\,\text{dB}} + 10\log_{10}(R_b)$, where R_b is the information bit rate in bits per second. Note that for GPS L1C, the navigation message is transmitted at 100 coded bits per second with a code rate of 1/2, resulting in an information bit rate of $R_b = 50$ bps.

Figure 7 compares the error performances achieved by the polar codes for Subframe 2 and Subframe 3 with three list sizes, $L = 8$, $L = 32$, and $L = 1024$, in AWGN. The NAs for both cases are also plotted. It can be observed from Figure 7 that for a smaller list size of $L = 8$, the error performance is degraded as compared with the case of $L = 32$ at the trade-off of lower complexity. For both Subframes 2 and 3, there is a considerable gap between the error rates achieved by small list

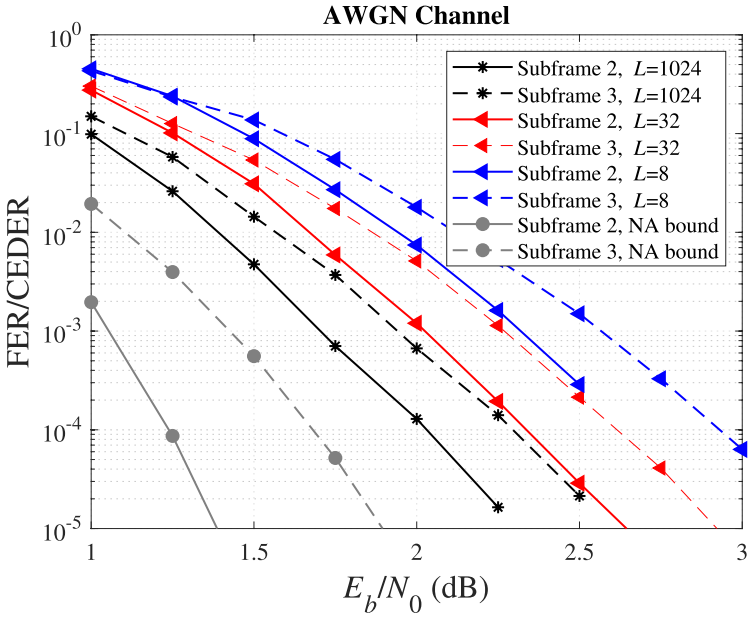

FIGURE 7 NA versus error rate of the proposed polar code using list sizes of 8, 32, and 1024 for Subframes 2 and 3 in AWGN

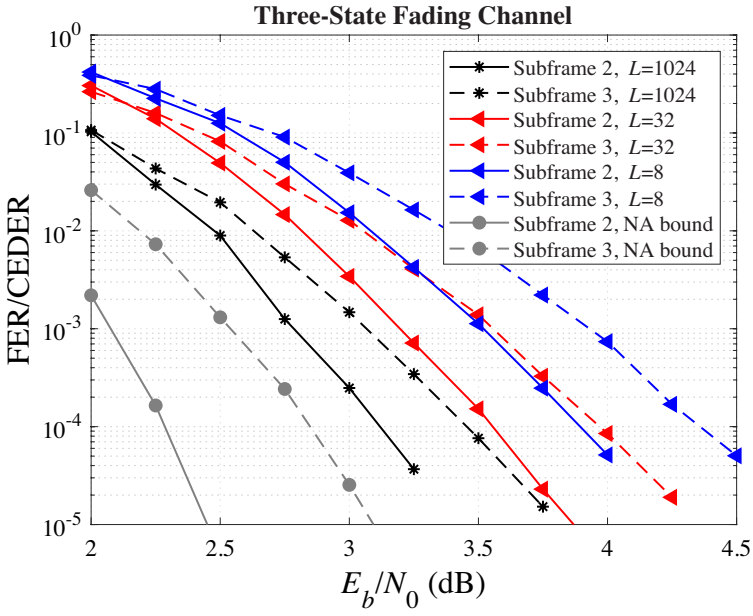

FIGURE 8 NA versus error rate of the proposed polar code using list sizes of 8, 32, and 1024 for Subframes 2 and 3 in three-state fading

sizes of $L=8$ and $L=32$ and the fundamental bound. However, for a list size of $L=1024$, the gap is significantly reduced, and we can approach close to the limit, operating only $0.6-0.7$ dB away. Similar results are also obtained for three-state fading, as shown in Figure 8.

In addition to the error performances, Table 2 details the computational complexity measures C^1_{Polar} and C^2_{Polar} for Subframes 2 and 3 for the three list sizes of $L=8, L=32,$ and $L=1024$. As expected, the substantially higher complexity associated with $L=1024$ reflects the near-optimal NA bound-approaching

TABLE 2
Computational Complexity in Terms of Decoding Operations for the Proposed Polar Coding Schemes with List Sizes of $L = 8$, $L = 32$, and $L = 1024$

	$L = 8$		$L = 32$		$L = 1024$	
	C^1_{Polar}	C^2_{Polar}	C^1_{Polar}	C^2_{Polar}	C^1_{Polar}	C^2_{Polar}
Subframe 2	235,000	206,200	1,016,800	844,000	38,681,600	28,544,000
Subframe 3	107,640	94,488	465,632	386,720	17,705,984	13,076,480

performance of the proposed polar code. In contrast, we achieve significantly reduced complexity in the case of $L = 8$ over $L = 32$ and $L = 1024$, highlighting the trade-off between decoding complexity and error performance.

6 | CONCLUSION

In this work, we have proposed a new CRC-aided shortened polar coding scheme that can be directly used in GPS L1C Subframes 2 and 3 for superior error correction capabilities. The novelty of our method lies in a new shortening algorithm based on power-of-two decomposition of the number of shortening positions and Hamming weights of the columns of the generator matrix. The new algorithm can flexibly generate shortening patterns to achieve non-power-of-two code lengths of 1200 bits and 548 bits for Subframes 2 and 3. Through extensive numerical examples, we showed that with their large FER/CED error rate performance improvement, which can achieve close to the fundamental finite-length limits over both AWGN and three-state fading channels, the proposed polar codes can serve as an attractive alternative for the LDPC codes currently used in GPS L1C protocols. The proposed shortening scheme has also been shown to achieve a better error rate than the traditional shortening scheme adopted in 5G NR.

ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Research Laboratory (AFRL) under Contract No. FA9453-23-P-A032 and FA9453-23-1-0002. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFRL.

REFERENCES

3GPP. (2022). 5G NR: Multiplexing and channel coding (3GPP TS 38.212 version 17.1.0 Release 17). *3GPP Technical Specification*, 1–201.

Anghileri, M., Paonni, M., Fontanella, D., & Eissfeller, B. (2013). Assessing GNSS data message performance: A new approach. *Inside GNSS 2*, 60–70. https://insidegnss.com/wp-content/uploads/2018/01/marapr13-WP.pdf

Arikan, E. (2009). Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, *55*(7), 3051–3073. https://doi.org/10.1109/TIT.2009.2021379

Bae, J. H., Abotabl, A., Lin, H.-P., Song, K.-B., & Lee, J. (2019). An overview of channel coding for 5G NR cellular communications. *APSIPA Transactions on Signal and Information Processing*, *8*, e17. https://doi.org/10.1017/ATSIP.2019.10

Baicheva, T., & Kazakov, P. (2019, September). CRC selection for decoding of CRC-polar concatenated codes. In *Proceedings of the 9th Balkan Conference on Informatics* (pp. 1–5). https://doi.org/10.1145/3351556.3351573

Bioglio, V., Condo, C., & Land, I. (2021). Design of polar codes in 5G new radio. *IEEE Communications Surveys & Tutorials*, *23*(1), 29–40. https://doi.org/10.1109/COMST.2020.2967127

Bioglio, V., Gabry, F., & Land, I. (2017, March). Low-complexity puncturing and shortening of polar codes. In *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (pp. 1–6). https://doi.org/10.1109/WCNCW.2017.7919040

Brualdi, R. A. (2009). *Introductory combinatorics* (5th ed.). Pearson Education.

Chiu, M.-C., & Su, Y.-S. (2023). Design of polar codes and PAC codes for SCL decoding. *IEEE Transactions on Communications*, *71*(5), 2587–2601. https://doi.org/10.1109/TCOMM.2023.3263225

Durisi, G., & Lancho, A. (2020, November). *Transmitting short packets over wireless channels—an information-theoretic perspective.* https://gdurisi.github.io/fbl-notes/

Filho, V. C. L., & Moraes, A. (2021). Modeling multifrequency GPS multipath fading in land vehicle environments. *GPS Solutions 25*. https://doi.org/10.1007/s10291-020-01040-8

Huang, L., Zhao, X., Chen, W., & Poor, H. V. (2021). Low-latency short-packet transmission over a large spatial scale. *Entropy*, *23*(7), 916. https://www.mdpi.com/1099-4300/23/7/916

Huawei & HiSilicon. (2016). On the latency and complexity. *3GPP TSG RAN-R1-164040*, 1–7.

Karasawa, Y., Kimura, K., & Minamisono, K. (1997). Analysis of availability improvement in LMSS by means of satellite diversity based on three-state propagation channel model. *IEEE Transactions on Vehicular Technology*, *46*(4), 1047–1056. https://doi.org/10.1109/25.653078

King, J., Kwon, A., Yang, H., Ryan, W., & Wesel, R. D. (2022, May). CRC-aided list decoding of convolutional and polar codes for short messages in 5G. In *ICC 2022 - IEEE International Conference on Communications* (pp. 92–97). https://doi.org/10.1109/ICC45855.2022.9838726

Le, T.-K., Salim, U., & Kaltenberger, F. (2021). An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17. *IEEE Access*, *9*, 433–444. https://doi.org/10.1109/ACCESS.2020.3046773

Lu, Z. Y. (2021). *Next-generation GNSS signal design: Theories, principles and technologies.* Springer.

Miloslavskaya, V. (2015). Shortened polar codes. *IEEE Transactions on Information Theory*, *61*(9), 4852–4865. https://doi.org/10.1109/TIT.2015.2453312

Nokia & Alcatel-Lucent. (2016). Performance and complexity of turbo, LDPC, and polar codes. *3GPP TSG RAN-R1-162897*, 1–5.

Ochiai, H., Mitran, P., & Vincent Poor, H. (2021). Capacity-approaching polar codes with long codewords and successive cancellation decoding based on improved gaussian approximation. *IEEE Transactions on Communications*, *69*(1), 31–43. https://doi.org/10.1109/TCOMM.2020.3030057

Oliveira, R. M., & De Lamare, R. C. (2021). Design of rate-compatible polar codes based on non-uniform channel polarization. *IEEE Access*, *9*, 41902–41912. https://doi.org/10.1109/ACCESS.2021.3065816

Ortega, L., & Poulliat, C. (2023). Global navigation satellite system channel coding structures for rapid signal acquisition in harsh environmental conditions. *NAVIGATION*, *70*(3). https://navi.ion.org/content/70/3/navi.585

Ortega, L., Poulliat, C., Boucheret, M. L., Aubault-Roudier, M., & Al-Bitar, H. (2020a). Optimizing the co-design of message structure and channel coding to reduce the TTD for a Galileo 2nd generation signal. *NAVIGATION*, *67*(3), 471–492. https://doi.org/10.1002/navi.382

Polyanskiy, Y., Poor, H. V., & Verdu, S. (2010). Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, *56*(5), 2307–2359. https://doi.org/10.1109/TIT.2010.2043769

Polyanskiy, Y., & Verdú, S. (2011, July). Scalar coherent fading channel: Dispersion analysis. In *2011 IEEE International Symposium on Information Theory Proceedings* (pp. 2959–2963). https://doi.org/10.1109/ISIT.2011.6034120

Saber, H., & Marsland, I. (2015). An incremental redundancy hybrid ARQ scheme via puncturing and extending of polar codes. *IEEE Transactions on Communications*, *63*(11), 3964–3973. https://doi.org/10.1109/TCOMM.2015.2477082

Sadeghi, P., Kennedy, R. A., Rapajic, P. B., & Shams, R. (2008). Finite-state Markov modeling of fading channels - a survey of principles and applications. *IEEE Signal Processing Magazine*, *25*(5), 57–80. https://doi.org/10.1109/MSP.2008.926683

Shin, D.-M., Lim, S.-C., & Yang, K. (2013). Design of length-compatible polar codes based on the reduction of polarizing matrices. *IEEE Transactions on Communications*, *61*(7), 2593–2599. https://doi.org/10.1109/TCOMM.2013.052013.120543

Tal, I., & Vardy, A. (2015). List decoding of polar codes. *IEEE Transactions on Information Theory*, *61*(5), 2213–2226. https://doi.org/10.1109/TIT.2015.2410251

Tropea, M., & De Rango, F. (2022). A comprehensive review of channel modeling for land mobile satellite communications. *Electronics*, *11*(5). https://doi.org/10.3390/electronics11050820

US Space Systems Command. (2022). NAVSTAR GPS space segment/user segment L1C interfaces. *IS-GPS-800*, 1–146.

Vaezi, M., Azari, A., Khosravirad, S. R., Shirvanimoghaddam, M., Azari, M. M., Chasaki, D., & Popovski, P. (2022). Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road towards 6G. *IEEE Communications Surveys & Tutorials*, 1–58. https://doi.org/10.1109/COMST.2022.3151028

Wang, R., & Liu, R. (2014). A novel puncturing scheme for polar codes. *IEEE Communications Letters*, *18*(12), 2081–2084. https://doi.org/10.1109/LCOMM.2014.2364845

Yang, W., Collins, A., Durisi, G., Polyanskiy, Y., & Poor, H. V. (2018). Beta–beta bounds: Finite-blocklength analog of the golden formula. *IEEE Transactions on Information Theory*, *64*(9), 6236–6256. https://doi.org/10.1109/TIT.2018.2837104

Yang, W., Durisi, G., Koch, T., & Polyanskiy, Y. (2013, August). Block-fading channels at finite blocklength. In *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems* (pp. 1–4). https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6629767

Zhang, L., Zhang, Z., Wang, X., Yu, Q., & Chen, Y. (2014, June). On the puncturing patterns for punctured polar codes. In *2014 IEEE International Symposium on Information Theory* (pp. 121–125). https://doi.org/10.1109/ISIT.2014.6874807

How to cite this article: Kankanamge, N.J.G., Tran, N.H., Pham, K., Shen, D., & Chen, G. (2026). Novel CRC-aided shortened polar codes for GPS L1C subframes 2 and 3. *NAVIGATION*, *73*. https://doi.org/10.33012/navi.792

7 | APPENDICES

7.1 | Shortening in 5G NR

From Equation (1), a bit x_i in the polar codeword $\underline{x}$ can be expressed as follows:

$$x_i = \underline{u}\boldsymbol{G}_{N\times N}(:, i) \tag{24}$$

where $\boldsymbol{G}_{N\times N}(:, i)$ is the i-th column of $\boldsymbol{G}_{N\times N}$. If x_i is chosen as a shortened bit, the set of frozen bits in the information sequence $\underline{u}$ must include all of the indices of column g(i) that contain "1." Because of the Kronecker structure of $\boldsymbol{G}_{N\times N}$, a simple method adopted in 5G NR is to choose the last bit positions in $\underline{x}$ for shortening. To better illustrate this approach, let us consider a polar code with a mother block-length of $N = 8$, a target code length of $E = 5$, and $K = 3$ information bits, resulting in $|S| = N - E = 3$ bits to be shortened. The generator matrix $G_{8\times 8}$ is given by the following:

$$G_{8\times 8} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & \mathbf{0} & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{25}$$

By selecting the last three indices of the codeword to shorten, i.e., $S = \{6, 7, 8\}$, from the locations of bits "1" in the last three columns of $G_{8\times 8}$, the input $\underline{u}$ at indices $\{6, 7, 8\}$ must be frozen to 0 in order to achieve $x_6 = x_7 = x_8 = 0$. By doing so, we sacrifice these later indices, which might correspond to more reliable polarization channels that could instead be used to carry information bits.

It should be noted that by further examining the other columns of $G_{8\times 8}$, one can select indices {4, 6, 8} for shortening, resulting in an alternative way to place

information bits. This indicates that there exist different shortening patterns. For large N, a comparative performance analysis becomes computationally intractable, making an optimal selection generally unfeasible.

7.2 | Calculating $\left|B_{x_i} \cap \left(\bigcap_{j\in J} B_{x_j}\right)\right|$

Let us first consider the case in which J contains only one element $j < i$. In this case, we have the following:

$$B_{x_i} \cap B_{x_j} = \left(S_i \cup \left(\bigcup_{z=2}^{i}\{C_z\}\right)\right) \cap \left(S_l \cup \left(\bigcup_{z=2}^{j}\{C_z\}\right)\right) \tag{26}$$

Because $C_i \in \left(S_{i-2} \cap \overline{S_{i-1}}\right), C_{j+1} \notin S_j$, and $\left\{C_{j+2}, C_{j+3}, \ldots, C_i\right\} \subseteq S_j$, we have the following:

$$\begin{aligned} B_{x_i} \cap B_{x_j} &= \left(S_i \cup \left(\bigcup_{z=2}^{j+1}\{C_z\} \bigcup_{z=j+2}^{i}\{C_z\}\right)\right) \cap \left(S_j \cup \left(\bigcup_{z=2}^{j}\{C_z\}\right)\right) \\ &= \left[\left(S_i \cup \left(\bigcup_{z=2}^{j+1}\{C_z\} \bigcup_{z=j+2}^{i}\{C_z\}\right)\right) \cap S_j\right] \cup \left[\left(S_i \cup \left(\bigcup_{z=2}^{j+1}\{C_z\} \bigcup_{z=j+2}^{i}\{C_z\}\right)\right) \cap \left(\bigcup_{z=2}^{j}\{C_z\}\right)\right] \\ &= \left[S_i \cup \left(\bigcup_{z=j+2}^{i}\{C_z\}\right)\right] \cup \left[\left(\bigcup_{z=2}^{j}\{C_z\}\right)\right] = \left(S_i \cup \left(\bigcup_{z=2}^{i}\{C_z\}\right)\right) \setminus \{C_{j+1}\} \end{aligned} \tag{27}$$

As a result, we have the following:

$$\left|B_{x_i} \cap B_{x_j}\right| = \left|\left(S_i \cup \left(\bigcup_{z=2}^{i}\{C_z\}\right)\right) \setminus \{C_{j+1}\}\right| = p_i + (i-1) - 1 \tag{28}$$

We can then straightforwardly add any additional set B_{x_l} and iteratively extend the result in Equation (28) to any multi-element $J \subseteq \{1, .., i-1\}$ and obtain the following:

$$\left|B_{x_i} \cap \left(\bigcap_{j\in J} B_{x_j}\right)\right| = \left|\left(S_i \cup \left(\bigcup_{z=2}^{i}\{C_z\}\right)\right) \setminus \bigcup_{j\in J} C_{j+1}\right| = p_i + (i-1) - |J| \tag{29}$$

7.3 | Calculation of NA

For a given channel realization $H = h$, the channel in Equation (2) is a static AWGN channel. Its maximum achievable rate for a given block-length N and error rate $\in$ can be expressed as follows (Durisi and Lancho, 2020; Polyanskiy et al., 2010):

$$R^*(N, \epsilon, h) = C(h) - \sqrt{\frac{V(h)}{N}} Q^{-1}(\epsilon) + O\left(\frac{\log N}{N}\right) \tag{30}$$

In Equation (30), $Q^{-1}(\cdot)$ is the inverse of the Gaussian Q-function. Furthermore, $C(h)$ is the constrained capacity for a given distribution of input X, and $V(h)$ is the channel dispersion. These two parameters depend on a so-called information density $i(X_n, Y_n)$ for a given input/output pair (X_n, Y_n):

$$i\left(X_n, Y_n\right) = \log_2\left(\frac{P_{Y|X}\left(Y_n \mid X_n, H=h\right)}{P_Y\left(Y_n \mid H=h\right)}\right) \tag{31}$$

The channel capacity $C(h)$ and the channel dispersion $V(h)$ can then be calculated as follows:

$$C(h) = \mathrm{E}\left[i\left(X_n, Y_n\right)\right] \tag{32}$$

and:

$$V(h) = Var\left[i\left(X_n, Y_n\right)\right] \tag{33}$$

By neglecting the error term in the expansion in Equation (30), we obtain the instantaneous NA when $H=h$, which becomes the NA under AWGN with $H=1$. In the case of fading, under the interleaved channel model, we can take the average over H and approximate the rate $R^*(N, \in)$ via the NA as follows:

$$R^*(N, \in) \approx I_{\mathrm{Norm}} = C - \sqrt{\frac{V}{N}} Q^{-1}(\epsilon) \tag{34}$$

Note that $C = E_H[C(h)]$ is the ergodic capacity and $V = E_H[V(h)]$ is the ergodic channel dispersion, with the expectations taken over the three-state fading distribution given in Equation (4).

In both AWGN and fading cases, given the calculated C, R, V, and block-length N, the minimum FER, denoted as $\mathrm{FER}_{\min}$ for a given block-length N and transmission rate R, which is commonly referred to as the NA bound, can then be calculated as follows:

$$\mathrm{FER}_{\min} = \epsilon \approx Q\left(\frac{C-R}{\sqrt{V/N}}\right) \tag{35}$$