Rate-adaptive Inner Code for Non-Binary Decoders

Cédric Marchand, Emmanuel Boutillon

Rate-adaptive Inner Code for Non-Binary Decoders

Cédric Marchand, Emmanuel Boutillon
Université de Bretagne Sud, Lab-STICC - UMR 6285
Email: {cedric.marchand, emmanuel.boutillon}@univ-ubs.fr

***Abstract*—In most communication systems, the coding rate of the error control code adapts to the channel condition. In this paper, we propose a new approach. A fixed coding rate GF(q) Non-binary code is used as an outer code, and a cardinal q non-linear inner code is truncated to match the channel condition. The inner code is a Truncated Cyclic Code Shift Keying code. Simulation results for a payload of 120 bits show state-of-the-art performances in a wide range of SNR from -13.5 to 7 dB.**

***Index Terms*—Low SNR, Rate-adaptive, Non-Binary decoder, CCSK, CSK, truncated, sequence.**

I. Introduction

Non-Binary (NB) LDPC codes [1], designed over Galois Field GF(q) have recently been adopted in satellite standards [2]. They benefit from better error-correcting performance than their binary counterpart, especially when targeting short to moderate frame length [3]. Recent standards define a rate-adaptive scheme to adapt to channel conditions [4] or available resources [5]. However, designing a rate-adaptive NB-decoder along with NB codes is still a complex task. In fact, most NB decoder implementations are optimized for a single code rate [6][7][8]. One way to get around the problem is by giving the rate-adaptive task to an inner code.

Code shift keying (CSK) or Cyclic CSK (CCSK) modulation [9] [10] is a 2^m -ary Direct-Sequence Spread Spectrum (DSSS) technique that associates m-bits symbols to 2^m-chips sequences. The CCSK sequences are defined by a circular shift of the initial Pseudo-random Noise (PN) root sequence. An example of application is the Quasi-Zenith Satellite System (QZSS) [11], where the encoded 8-bit symbols of a Reed-Solomon (255,223) code are mapped on CCSK sequences. An example of encoded symbols (first step) mapped on CCSK sequences (second step) is shown in Fig. 1. Without loss of generality, the NB-encoder is in this article an NB-LDPC encoder. This concatenation allows to target efficiently very low SNRs and gives advantages in both modulation and synchronization [12]. The Truncated CCSK (TCCSK) described in [10] consists of not transmitting consecutive bits of the CCSK sequence to obtain code rate flexibility. Fig. 1 also highlights, in a third step, the truncation applied to the CCSK symbols. Thus, the combination of an NB decoder with a TCCSK code allows obtaining flexible rate while keeping the NB decoder optimized for a single rate.

Another simple process to obtain a flexible code is to use a repetition scheme as in 5G [5] for the polar codes. In [13], an NB-LDPC code is concatenated with a repetition scheme where repeated GF symbols are multiplied by a predefined GF coefficient, giving better performance than best low-rate

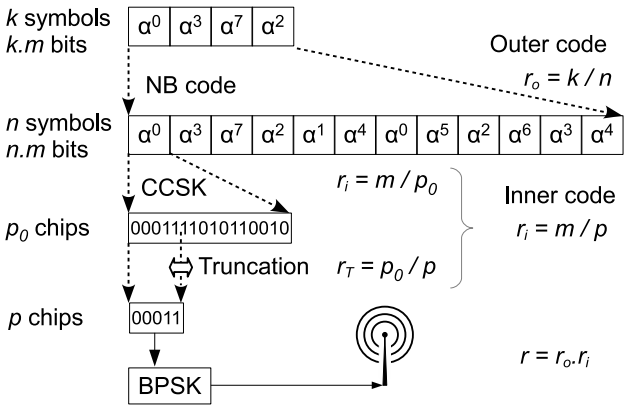

Fig. 1. Inner Truncated CCSK concatenated with outer NB encoder

binary codes. In [14], the non-binary multiplicative coefficients are optimized with the Hamming distance of the binary image of the component code. Similarly, the CCSK code is chosen for its good Hamming distance.

In this paper, the concatenated code is further enhanced to obtain fine-grained code rates and higher code rates than the outer NB code. An example of CCSK coding scheme for $k = 20$ information symbols on GF(64) and FER $= 10^{-3}$ shows a spectral efficiency within 2 dB from achievable capacity at finite blocklength regime [15] at SNR from -13.5 dB to 7 dB.

The paper is organized as follows: Section II introduces NB-codes, the CCSK modulation, and the concatenation of an NB encoder with a CCSK modulation. Section III presents the truncation process applied to the CCSK modulation. Section IV presents simulation results. Finally, Section VI concludes the paper.

II. NB-LDPC codes and CCSK modulation

A. NB-LDPC codes

In this article, a NB-LDPC code is used as NB-code without loss of generality. Throughout this article, NB-LDPC codes are defined over GF(q) where $q = 2^m$. A NB-LDPC encoder encodes k symbols into n symbols as shows in Fig. 1. The outer code rate is given by $r_o = k/n$.

Since the output of a NB encoder are NB symbols that can be directly mapped to modulation of the same order, both the NB-LDPC encoder and decoder do not need modification to be concatenated with the CCSK modulation.

B. Pseudo Noise sequence construction

CCSK uses a Pseudo-Noise (PN) binary sequence of length $p_0 \geq q$ defined as $\mathbf{B}_0 = (\mathbf{B}_0(0), \mathbf{B}_0(1), ..., \mathbf{B}_0(p_0-1))$, where $\mathbf{B}_0(i) \in \{0,1\}, \forall i = 0, ..., p_0 - 1$. The PN sequence can be generated using an m-sequence, an extended m-sequence [16], or a balanced sequence with good auto-correlation properties. An m-sequence has a length of $2^{m'} - 1$ bits and is generated using a Linear Feedback Shift Register (LFSR) of m' bits.

Considering an example of 4-bit LFSR with a feedback polynomial $g(x)$ defined as $g_4(x) = x^4 + x + 1$ and an initialization state with the first flip-flop set at 1 and others set at 0, the obtained m-sequence of 15 chips is

$$\mathbf{B}_0 = [000111101011001]. \quad (1)$$

The m-sequence is a Maximum-Length Sequence (MLS), i.e., during a period of $2^{m'} - 1$ cycles, the m' LFSR registers take all possible $2^{m'} - 1$ states ($2^{m'}$ states minus the all-zero state). In [16], for practical reasons, the MLS is extended to $2^{m'}$ chips by adding the all-zero state. The all-zero state is obtained by inserting a zero to the longest run of 0 in the MLS. Due to the periodicity of MLS, $\mathbf{B}_0(0) = \mathbf{B}_0(q-1) = 0$, the MLS of (1) is extended by one bit as follows

$$\mathbf{B}_0 = [0001111010110010]. \quad (2)$$

By applying one right circular shift to $\mathbf{B}_0$, one can check that a run of m' zeros is obtained. Also, the sequence is balanced with the same number of occurrences of 0 and 1. This result is valid for all Extended MLS (EMLS) with all m' values, provided that the first LFSR flip-flop is initialized with a 1 and others are initialized with zeros.

C. CCSK mapping

The CCSK modulation maps an element $b \in \{0, 1, ..., q-1\}$ to the sequence $\mathbf{B}_b$ defined as the circular left shift of $\mathbf{B}_0$ by b positions, that is

$$\mathbf{B}_b(i) = \mathbf{B}_0(i + b \mod p_0), \forall i = 0, ..., p_0 - 1. \quad (3)$$

where $p_0 \geq q$. Then, a BPSK modulation ($0 \mapsto 1, 1 \mapsto -1$) maps the $\mathbf{B}_b$ sequence to the $\mathbf{P}_b$ sequence.

Let us define the injective function π that gives b the value of the circular shift of $\mathbf{B}_0$ by b positions as a function of the GF symbol a. Table I gives an example where $\mathbf{B}_0$ is given by (2) and $q = 8$. The PN sequence $\mathbf{B}_0$ is shifted circularly by $\pi(a) = b$ positions to transmit a symbol $a \in$ GF(8).

Since binary representation of GF(q) symbols are on $\log_2(q) = m$ bits and the m bits are mapped on p_0 chips, we define the inner code rate as

$$r_i = \frac{m}{p_0}. \quad (4)$$

Considering both the inner CCSK code and the outer NB code, we define the effective code rate as

$$r = r_o \times r_i. \quad (5)$$

TABLE I
SEQUENCE AS A FUNCTION OF a WITH $p_0 > q$

a	$b = \pi(a)$	$\mathbf{B}_b$
0	0	0001111010110010
α^0	1	0011110101100100
α^1	2	0111101011001000
α^2	3	1111010110010000
α^3	4	1110101100100001
α^4	5	1101011001000011
α^5	6	1010110010000111
α^6	7	0101100100001111

Considering an example with GF(64), $r_o = \frac{1}{3}$ and $p_0 = q$, then $r = \frac{1}{3} \times \frac{6}{64} = \frac{1}{32}$. With $p_0 > q$, one can target a much lower effective coding rate. e.g., with $p_0 = 1024$ then $r = \frac{1}{512}$.

III. TRUNCATED CCSK

A. Principle

The TCCSK sends the first p bits $p \leq p_0$ of the CCSK sequence as shown in Fig. 1. The TCCSK mapping is defined as

$$\mathbf{B}_b^p(i) = \mathbf{B}_0(i + b \mod p_0), \forall i = 0, ..., p - 1. \quad (6)$$

The truncation rate is $r_T = \frac{p_0}{p}$ and the inner code rate is updated to

$$r_i = \frac{m}{p}. \quad (7)$$

To avoid r_i greater than 1, p is lower bounded to $p \geq m$. Thus, r_i can take $p_0 - m + 1$ different values between $\frac{m}{p_0}$ and 1, i.e., $\frac{m}{p_0} \leq r_i \leq 1$.

Let us apply a truncation such that $p = m = 3$ to the CCSK mapping shown in Table I, i.e., to transmit a symbol $a \in$ GF(8), only the first 3 chips of the $\mathbf{B}_b$ sequence are transmitted instead of 16 chips. Table II gives the truncated sequence $\mathbf{B}_b^m$ associated to each symbol a.

TABLE II
TRUNCATED SEQUENCE AS A FUNCTION OF a WITH NATURAL MAPPING AND OPTIMIZED MAPPING

a	$b = \pi(a)$	$\mathbf{B}_b^3$	$c = \pi'(a)$	$\mathbf{B}_c^3$
0	0	000	0	000
α^0	1	001	1	001
α^1	2	011	2	011
α^2	3	111	3	111
α^3	4	111	8	100
α^4	5	110	5	110
α^5	6	101	6	101
α^6	7	010	7	010

One can see in Table II that $\mathbf{B}_3^3 = \mathbf{B}_4^3$. When applying truncation to natural CCSK mapping, two different symbols can map the same truncated sequence, which will lead to significant performance degradation. The optimized injective function π' makes sure that all symbols in GF(q) are mapping a different truncated sequence. Hence, π' must validate

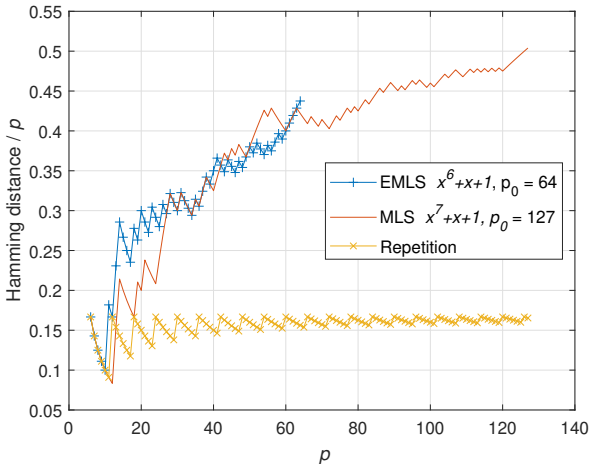

Fig. 2. Average Hamming distance per chip of inner code as a function of p

$$e \neq f \Rightarrow \mathbf{B}^m_{\pi'(e)} \neq \mathbf{B}^m_{\pi'(f)}. \tag{8}$$

When $p_0 > q$, the mapping optimization can be done by an update of π such that (8) is true. In Table II, π is optimized to π' where $\pi'(4)$ is updated to meet the requirement in (8). Thus, when $p_0 > q$, a specific mapping π' is precomputed.

B. Construction of optimized sequence for truncation

By construction, an MLS will produce $q-1$ different truncated sequences of size $p = m$, i.e., all possible different truncated sequences except the all-zero sequence. The EMLS provides the missing truncated sequence, i.e, $\mathbf{B}^T_{16}$ of (2) gives [0000]. Thus, the EMLS validates (8) by definition. For $p_0 > q$, many sequences and π options validate (8) but the challenge is to find out the sequence that provides the best performances.

First, a cost criterion has to be defined to select a root sequence rather than another. The CCSK modulation is considered as a code and its minimum Hamming distance is computed. Each sequence $\mathbf{B}_b$ of the CCSK modulation is studied as a codeword and the CCSK modulation with the highest hamming distance is selected. When $p_0 = q$, an EMLS of degree m can be used as a reference, and the exploration of the best possible solutions can be limited to the selection of the MLS primitive that gives the best Hamming distance. When $p_0 > q$, the exploration space is huge. One solution to obtain a good Hamming distance for $(l+1)\times q > p_0 > l \times q$ is to use an MLS of degree $m+l+1$, then truncate the obtained root sequence to p_0 and update π to meet the requirements in (8).

Fig. 2 shows the average Hamming distance per chip as a function of p for a CCSK constructed from an EMLS, an MLS, or a repetition code. The EMLS is of size $p = 64$ and is generated from polynomial $g_6(x) = x^6 + x + 1$. The MLS is of size $p_0 = 127$ and is generated from polynomial $g_7(x) = x^7 + x + 1$. The min distance of the repetition code is given by $\lfloor p/m \rfloor$. The MLS shows a Hamming distance 3 times better than the repetition code for $p = p_0 = 127$.

The EMLS also has the property to have a uniform distribution of 0s and 1s, assuming a uniform distribution of $a \in \mathrm{GF}(q)$, for all p values. The MLS does not have the same property but can be optimized with π to obtain uniform distribution with less than 1% deviation.

C. Per symbol truncation

The highest truncation also gives the highest code rate gap between two code rates. The two highest truncations for $q = 64$ are $p = 6$ and $p = 7$ which gives according to (5) and (7), $r = 1/3$ and $r = \frac{2}{7}$. The percentage difference between the two r is 15%. When targeting a fine-grained rate-adaptive decoder, a finer code rate adaptation can be obtained by considering per symbol truncation. This technique applies a different truncation value as a function of the j th encoded symbol. The average truncation $\overline{p}$ is given by

$$\overline{p} = \frac{1}{n}\sum_{j=0}^{n-1} p(j), \tag{9}$$

where $p(j)$ is the truncation applied to the j th CCSK sequence associated with the j th symbol. The average inner code rate r_i with per symbol truncation is given by

$$r_i = \frac{m}{\overline{p}}. \tag{10}$$

To obtain fine-grained values between p and $p+1$, we can use CCSK mapping on $p+1$ chips for the first x symbols and CCSK mapping on p chips for the last $n-x$ other symbols. The average $\overline{p}$ value is updated to

$$\overline{p} = p + \frac{x}{n}. \tag{11}$$

Thanks to this process, the code rate gap can be as fine as required. With $n = 10$, the maximum gap is less than 2% and with $n = 100$, the maximum gap is less than 0.2%.

D. CCSK over-truncation

CCSK over truncation consists of sending only $p < m$ bits. This truncation process is called over-truncation since the TCCSK code rate becomes greater than one, i.e., $r_i = m/p > 1$. However, the effective code rate r can not be greater than 1, giving a lower bound to p such that $p > r_o \times m$. With r_o=1/3, it is possible to target r equals 2/3, 1/2, or 2/5 respectively with p equals 3, 4, or 5 bits.

Over-truncation leads to loss of information since (8) constraint can not be found. The NB decoder is in charge of recovering this loss. Over-truncation is applied more on redundant symbols than on information symbols to obtain good performance. Let us consider p^I, the truncation applied to information symbols, and p^R the truncation applied to the redundant symbols. Table III gives the p^I, p^R, and obtained r used for simulations.

IV. Simulation results of TCCSK

In this section, the CCSK demodulation used in simulations and the TCCSK capacity shown in simulations are first described.

TABLE III
r AS A FUNCTION OF p^I AND p^R VALUES

p^I	p^R	$\overline{p}$	r_i	r	r
6	1	8/3	9/4	3/4	0.75
5	2	3	2	2/3	0.667
6	2	10/3	9/5	3/5	0.6
6	3	4	3/2	1/2	0.5
5	5	5	6/5	2/5	0.4
⋮	⋮	⋮	⋮	⋮	⋮
64	64	64	6/64	2/64	0.031

A. CCSK demodulation

The intrinsic associated with each symbol a is obtained from the CCSK demodulation. Let us consider a symbol $a \in \text{GF}(q)$ transmitted using CCSK modulation with $X = \mathbf{P}_a$. Assuming Binary Additive White Gaussian Noise (Bi-AWGN), the received CCSK signal is

$$Y(i) = X(i) + Z(i), \tag{12}$$

where $Z(i)$ are real-valued, mutually independent samples of the $\mathcal{N}(0, \sigma^2)$ normal distribution.

The *a priori* information associated with the received symbol is given by the Log-Likelihood Ratio (LLR) value defined by

$$\Gamma(a) = \log \frac{\Pr(X = \mathbf{P}_{\hat{a}}|Y)}{\Pr(X = \mathbf{P}_a|Y)}, \tag{13}$$

where $\hat{a} = \arg\max \Pr(X = \mathbf{P}_a|Y), a \in \text{GF}(q)$. Hence, we have

$$\begin{aligned}\Gamma(a) &= \sum_{i=0}^{p-1} \log \frac{\Pr(\mathbf{P}_{\hat{a}}(i)|Y(i))}{\Pr(\mathbf{P}_a(i)|Y(i))} \\ &= \sum_{i=0}^{p-1} \frac{\mathbf{P}_{\hat{a}}(i) - \mathbf{P}_a(i)}{2} \log \frac{\Pr(X(i) = +1|Y(i))}{\Pr(X(i) = -1|Y(i))}.\end{aligned} \tag{14}$$

With BI-AWGN channel, we have $\log \frac{\Pr(X(i)=+1|Y(i))}{\Pr(X(i)=-1|Y(i))} = \frac{2}{\sigma^2} Y(i)$ and therefore

$$\Gamma(a) = \frac{1}{\sigma^2} \sum_{i=0}^{p-1} Y(i)\mathbf{P}_{\hat{a}}(i) - Y(i)\mathbf{P}_a(i). \tag{15}$$

Note that by construction, $\Gamma(\hat{a}) = 0$ and $\Gamma(a) \geq 0$.

B. TCCSK capacity

The mutual information between the input U of the TCCSK modulation and the output Y of the channel give the maximal achievable rate r of the TCCSK modulation. I.e.,

$$r = I(U;Y) = H(U) - H(U|Y), \tag{16}$$

where U denotes the channel input, $H(U)$ denotes the entropy of U, and $H(U|Y)$ denotes the conditional entropy of U given Y. For BPSK modulation, $H(U) = 1$ and the conditional entropy $H(U|Y)$ can be estimated numerically by Monte-Carlo simulation by averaging over the channel output Y,

$$H(U|Y) = \mathbb{E}_Y \left[- \sum_{k \in \mathbb{Z}_q} \Pi(k) \log_q \Pi(k) \right], \tag{17}$$

where $\Pi(k) = \Pr(U = k|Y)$ denotes the conditional probability distribution of U given Y.

Let us define the Monte-Carlo simulation that provides $I(U;Y) = f(\text{SNR},p)$. To compute the curve showing capacity r as a function of SNR, one must find the SNR that validates $I(U;Y) = r$ for a given p value.

In [15], Polyanskiy et al. show that the maximal rate achievable with error probability ϵ and block length N is closely approximate by

$$C^* = C - \sqrt{\frac{V}{N}} Q^{-1}(\epsilon), \tag{18}$$

where V is a characteristic of the channel referred to as channel dispersion, and Q is the complementary Gaussian cumulative distribution function.

C. Simulation results

The NB decoder is an NB-LDPC decoder on GF(64) using the Extended Min-Sum algorithm [1], a forward-backward check node implementation [17], a maximum of 30 iterations, $n_m = 20$, and $n_{op} = 25$. The NB-LDPC code is a $k = 20$ and $r_o = 1/3$ code. The Signal-to-Noise Ratio is defined as SNR=$-10\log_{10}(\sigma^2)$ (dB).

Fig. 3 shows simulation results of an NB decoder combined with TCCSK. The CCSK sequence is an MLS of length $p_0 = 127$. The simulation of the NB-decoder alone (without TCCSK) with BPSK modulation provides a reference curve for $r = 1/3$. The TCCSK with $p = 6, 7, 8$, and 9 provide simulation results for $r = 1/3, 2/7, 1/4, 2/9$ respectively. The TCCSK with over truncation at $p = 5$ and $\overline{p} = 4$ provide simulation results for $r = 1/2$ and $2/5$ respectively. One can check that the NB decoder alone with BPSK gives the same simulation result as the NB-decoder concatenated with TCCSK with $p = 6$ ($r_o = 1/3$ and $r_i = 1$). Simulation results of 5G are also shown for comparison with state-of-the-art. 5G results are obtained using flooding BP, a maximum of 50 iterations, LDPC code base 1, 120 information bits, code rates 1/3, 2/5, 1/2 and BPSK modulation.

Fig. 4 shows r as a function of the SNR required to obtain FER$= 0.001$. The simulation result of the TCCSK concatenated with the NB-LDPC is shown on its full code rate range. The circle on the TCCSK range represent the simulation result of the NB decoder with BPSK modulation. The maximal achievable coding rate of a BPSK modulation is shown as a reference. The TCCSK's capacity is obtained from (16) and (17) by simulation. The achievable coding rate of TCCSK modulation for a given FER and frame length [15] is computed from (18). Furthermore, the simulation result of a repetition process concatenated with the NB decoder is also shown. At

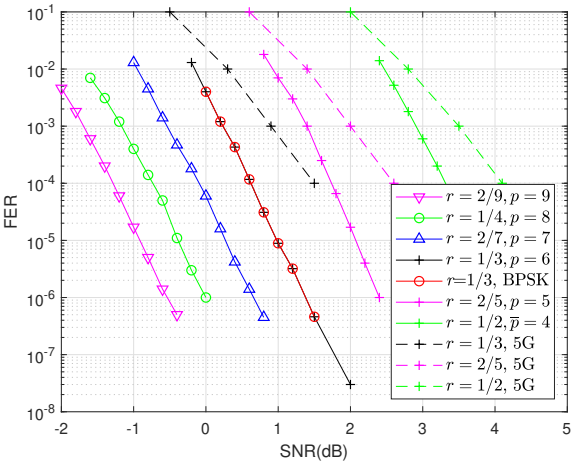

Fig. 3. FER Simulation results with TCCSK.

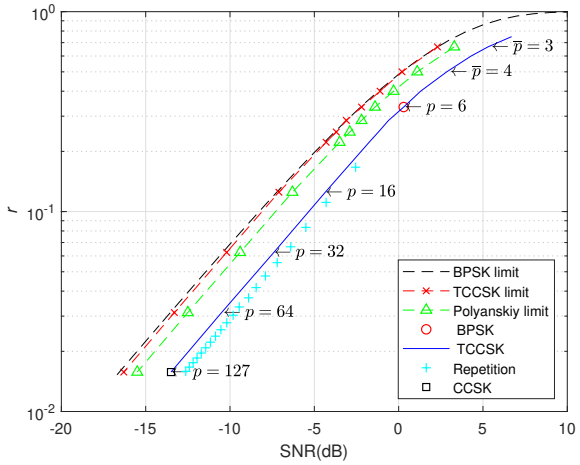

Fig. 4. Full code rate range simulation results with TCCSK.

SNR lower than -5 dB, repetition shows 0.75 dB performance loss compared with the TCCSK solution.

V. Conclusion

In this paper, we show that a low SNR fine-grained rate-adaptive NB decoder can be obtained by combining an NB decoder with a Truncated CCSK modulation. The truncation process shows achievable capacity close to the Shannon capacity.

Acknowledgement

This work has been funded by the french ANR under grant number ANR-19-CE25-0013-01 (QCSP Project: https://qcsp.univ-ubs.fr/).

References

[1] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low complexity decoding for non-binary LDPC codes in high order fields," IEEE Trans. Commun., vol. 58, no. 5, pp. 1365–1375, 2010.

[2] "BeiDou Navigation Satellite System, Signal In Space, Interface Control Document, Open Service Signals B1C (Version 1.0) " China Satellite Navigation Office , December, 2017.

[3] Davey, M. and D.J.C. Mackay. "Low density parity check codes over GF(q)." 1998 Information Theory Workshop (Cat. No.98EX131) (1998): 70-71.

[4] ETSI, "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting,Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)" in ETSI EN 302 307 V1.2.1 (2009-08).

[5] ETSI technical specification, "5G; NR; Multiplexixing and channel coding, " in 3GPP TS 38.212 version 15.2.0 Release 15

[6] C. Marchand, E. Boutillon, H. Harb, L. Conde-Canencia and A. Al Ghouwayel, "Hybrid Check Node Architectures for NB-LDPC Decoders," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 2, pp. 869-880, Feb. 2019.

[7] Y. S. Park, Y. Tao and Z. Zhang, "A Fully Parallel Nonbinary LDPC Decoder With Fine-Grained Dynamic Clock Gating," in IEEE Journal of Solid-State Circuits, vol. 50, no. 2, pp. 464-475, Feb. 2015.

[8] C. Lin, S. Tu, C. Chen, H. Chang and C. Lee, "An Efficient Decoder Architecture for Nonbinary LDPC Codes With Extended Min-Sum Algorithm," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 9, pp. 863-867, Sept. 2016.

[9] A. Y.-C. Wong and V. C. M. Leung, "Code-phase-shift keying: a power and bandwidth efficient spread spectrum signalling technique for wireless local area network applications," in Proc. IEEE Canadian Conf. Elect. Comput. Eng., vol. 2, pp. 478–481, May 1997.

[10] G. M. Dillard, M. Reuter, J. Zeiddler and B. Zeidler, "Cyclic code shift keying: a low probability of intercept communication technique," in IEEE Transactions on Aerospace and Electronic Systems, vol. 39, no. 3, pp. 786-798, July 2003.

[11] "Quasi-Zenith Satellite System Interface Specification - Centimeter Level Augmented Service (IS-QZSS-L6-003)" Cabinet Office, August, 2020.

[12] O. Abassi, L. Conde-Canencia, M. Mansour and E. Boutillon, "Non-binary coded CCSK and Frequency-Domain Equalization with simplified LLR generation," IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), London, 2013, pp. 1478-1483.

[13] K. Kasai, D. Declercq, C. Poulliat and K. Sakaniwa, "Multiplicatively Repeated Nonbinary LDPC Codes," in IEEE Transactions on Information Theory, vol. 57, no. 10, pp. 6788-6795, 2011.

[14] V. Savin, D. Declercq and S. Pfletschinger, "Multi-relay cooperative NB-LDPC coding with non-binary repetition codes over block-fading channels," 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, 2012, pp. 1354-1358.

[15] Y. Polyanskiy, H. V. Poor and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," in IEEE Transactions on Information Theory, vol. 56, no. 5, pp. 2307-2359, May 2010.

[16] U. Fiebig and M. Schnell, "Correlation properties of extended m-sequences," in Electronics Letters, vol. 29, no. 20, pp. 1753-1755, 30 Sept. 1993.

[17] E. Boutillon and L. Conde-Canencia, "Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary LDPC decoders," Electron. Lett., vol. 46, no. 9, pp. 633–634, 2010