

Performance Evaluation of Software Multi-Threshold Decoders for Self-Orthogonal Codes in Modern Broadband Wireless Communication Systems

Nurlan Tashatov¹, Gennady Ovechkin², Zhuldyz Sailaukyzy^{3*}, Eldor Egamberdiyev⁴,
Dina Satybaldina⁵, Gulmira Danenova⁶, Zarina Khassenova⁷

Research Institute of Information Security and Cryptology, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan^{1, 4, 5}

Department of Information Technologies, Abylkas Saginov Karaganda Technical University, Karaganda, Kazakhstan^{2, 3, 6}

School of Digital Technology and Artificial Intelligence, D. Serikbayev East Kazakhstan Technical University,
Ust-Kamenogorsk, Kazakhstan⁷

Abstract—5G and Internet of Things (IoT) wireless systems face challenges to reliable data transmission due to multipath fading, intersymbol interference, and the need for low-complexity Forward Error Correction (FEC). Conventional FEC techniques, such as Low-Density Parity-Check (LDPC) and turbo codes, provide high reliability but are unsuitable for resource-constrained IoT devices due to high decoding complexity. The aim of this study is to evaluate Multi-Threshold Decoders (MTDs) applied to Self-Orthogonal Codes (SOCs) as a low-complexity FEC solution in Orthogonal Frequency-Division Multiplexing (OFDM) and Multiple Input Multiple Output (MIMO) systems with Space-Time Coding (STC). The systems are modeled under ITU-R (Outdoor A, TU6, RA6) and 3GPP Spatial Channel Model (Urban Macro/Micro) fading environments and compared with LDPC (WiMAX, DVB-S2) and turbo codes in terms of Bit Error Rate (BER), Signal-to-Noise Ratio (SNR), decoder complexity, antenna diversity, modulation order, and throughput. Results indicate that SOC+MTD outperform short LDPC and turbo codes under deep fading while achieving reliability comparable to long LDPC codes at significantly lower decoding complexity. Min-sum refinement and approximate Maximum-Likelihood (ML) detection provide up to 2 dB additional SNR gain, and 1×3 antenna diversity reduces required E_b/N_0 by ~7 dB at BER = 10^{-5} . Higher-order modulations such as 8PSK and 16APSK achieve 1.5–2× higher bit rates with moderate SNR penalties, while Open Computing Language (OpenCL) based Graphics Processing Unit (GPU) acceleration enables a 32-fold increase in simulation speed. These findings demonstrate that SOCs decoded with MTD represent a promising low-complexity, high-reliability FEC approach for 5G and IoT physical layers.

Keywords—Forward Error Correction; Multi-Threshold Decoder; Self-Orthogonal Codes; Orthogonal Frequency-Division Multiplexing; Multiple Input Multiple Output; Bit Error Rate; signal-to-noise ratio; LDPC; turbo codes; space-time coding; fading channels; GPU acceleration; OpenCL

I. INTRODUCTION

Wireless communication systems are constantly evolving to meet the needs of society and industry for fast, high-quality transmission of large amounts of data with minimal latency and seamless connectivity for billions of devices [1]. Designers must take into account intersymbol interference (ISI) and fading, which arise due to the nature of wireless channels [2, 3].

ISI occurs due to reflections from buildings or terrain along which radio waves propagate. Delays in different signal pathways result in significant signal overlap, which leads to symbol distortion and a rising Bit Error Rate (BER). To keep up the required BER, it is necessary to accept a decreased data rate, which decreases channel capacity.

Fading can be defined as the variation of channel power over time and frequency. It leads to both constructive and destructive interference from the various possible propagation paths between the transmitter and receiver antennas. It occurs on a spatial scale comparable in magnitude to the carrier wavelength and is frequency dependent (small-scale fading). Another type of fading, which is not frequency related, is actually caused by signal attenuation with distance and shadowing by large objects such as buildings and hills. Such large-scale fading is usually taken into account in base station placement plans. Variations in signal amplitude, connection breaks, and degradation of multimedia data transmission quality are caused by small-scale multipath fading. It is aggravated in 5G networks because of the high data rates and density of mobile devices supported; the use of millimeter-wave frequencies; spatial diversity; and multiplexing [4]. When a moving Unmanned Aerial Vehicle (UAV) connects to a ground station, radio waves bounce off of the ground and other objects, causing the channel to change quickly and the Doppler shift to happen [5]. Due to the Internet of Things (IoT) devices have very small antennas with limited power, they are more vulnerable to experience fading. This is especially true in crowded urban or industrial areas where inter-symbol interference is more likely to happen [6]. In wireless sensor networks, fading and inter-symbol interference make it hard to collect data because the sensors are low-power and in hard-to-reach places (like forests, underground, or inside buildings). Therefore, several types of wireless communication standards have important design challenges related to channel fading, ISI, and interference suppression.

A variety of techniques are employed in modern wireless communications to enhance signal quality and reduce fading. Among these methods are orthogonal frequency division multiplexing [7, 8] combined with diversity techniques and beamforming, spatial diversity techniques, and Multiple Input Multiple Output (MIMO) systems [9, 10]. The Forward Error

*Corresponding author.

Correction (FEC) algorithms are crucial in this context as they can rectify data compromised by noise, interference, and fading without required retransmission [11, 12]. IEEE 802.16m (WiMAX) [13] and EN 302 307 (DVB-S2) [14] standards employ the Low-Density Parity-Check (LDPC) codes with varying code lengths and coding rates. These codes, as well as turbo codes, convolutional codes, and polar codes, are required for battery-powered devices in wireless sensor networks, as they provide a balance between computational complexity and error correction efficiency [15]. In 5G mobile networks, polar codes [16] are used for control channels, for which reliability and low latency are important requirements [17, 18]. These codes are not easy to decode, but researchers are developing more effective algorithms [19].

For 6G and future communication systems, in addition to further optimization of Polar and LDPC codes, it is proposed to explore the possibilities of machine learning for adaptive coding, optimization of code parameters, and improvement of decoding algorithms, especially in dynamic and complex channels [20]. Another promising direction is the development of new code constructions, algorithms for their decoding, which will even better combine low latency, high performance, low computational complexity, optimization for hardware implementation and flexibility, compatibility with other methods of reducing fading and interference (Orthogonal Frequency-Division Multiplexing (OFDM), MIMO, etc.) [21, 22].

In this study, the performance of iterative decoders for Self-Orthogonal Codes (SOCs), designed based on the Massey threshold decoder [23] and known as Multi-Threshold Decoders (MTDs) [24, 25] or Multi-Stage Threshold Decoders [26, 27], is investigated for wireless networks. Previous studies of MTD performance in fading MIMO radio channels [28] demonstrated that MTDs can nearly optimally decode even very long codes with low computational complexity, providing high efficiency under such conditions.

The application of MTDs for SOC in MIMO-OFDM wireless channels with multipath effects is examined, along with approaches to improve their performance in such environments. Additionally, aspects of their software implementation using parallel computing technologies are analyzed to evaluate their potential for 5G/Beyond and IoT applications.

However, despite the significant progress in modern channel coding, including the adoption of LDPC and turbo codes in WiMAX, DVB-S2, and 5G systems, their practical use in resource-constrained IoT and latency-sensitive applications remains limited by high decoder complexity and energy consumption. Existing research on reducing decoding complexity often sacrifices error correction performance or relies on specialized hardware platforms. Meanwhile, studies on threshold-based decoders, such as MTDs applied to SOC, have not yet been thoroughly evaluated in real-world broadband fading channels (e.g., ITU-R, 3GPP SCM) combined with MIMO-OFDM and STC. Moreover, the feasibility of implementing MTDs on portable software platforms (e.g., OpenCL) remains underexplored. Thus, there exists a research

gap in achieving both high reliability and low complexity decoding for modern wireless physical layers across diverse fading scenarios. This study addresses this gap by providing a comprehensive performance evaluation of SOC+MTD under realistic broadband channel models and by presenting a practical, software-based implementation using GPU acceleration.

The main contributions of this work are as follows:

- 1) A comprehensive performance evaluation of SOC+MTD decoding under standardized fading conditions (ITU-R Outdoor A, TU6, RA6 and 3GPP SCM Urban Macro/Micro) is conducted for the first time. The results demonstrate that SOC+MTD achieve BER performance comparable to DVB-S2 LDPC (16,200 bits) while maintaining up to $10\times$ lower decoding complexity and outperform turbo codes under deep-fading by 2–3 dB.
- 2) The integration of SOC+MTD with MIMO-OFDM and space-time coding (STC) is investigated, showing that 1×3 antenna diversity reduces the required E_b/N_0 by approximately 7 dB at $BER = 10^{-5}$, and effective STC matrices for 2×2 and 4×4 configurations are identified with optimized SNR-throughput trade-offs.
- 3) The combination of multi-stage threshold decoding with approximate Maximum-Likelihood (ML) detection and min-sum refinement delivers additional SNR gains up to 2 dB over MMSE-based schemes, without increasing decoding complexity.
- 4) A portable OpenCL-based decoder implementation for SOC+MTD is developed, achieving $32\times$ faster execution (480 Mbps) compared to CPU, and $12\times$ faster performance relative to baseline CUDA deployments, enabling feasible real-time processing on heterogeneous platforms.

Despite the significant advances in channel coding techniques, several critical limitations remain unaddressed in prior work. First, while threshold-based decoders such as MTDs have been studied in Gaussian and AWGN channels, their performance under complex broadband fading environments, especially those combining MIMO, OFDM, and STC, remains insufficiently evaluated. Second, existing studies on SOC+MTD architectures largely lack experimental validation against established standards such as ITU-R and 3GPP SCM channel models. Third, the applicability of MTD-based solutions in resource-constrained and real-time communication systems has not been validated through portable and hardware-agnostic software implementations. Most prior implementations either rely on proprietary GPU frameworks or are constrained to FPGA platforms without demonstrating performance portability. Therefore, a clear research gap exists in the combined evaluation of SOC+MTD performance under realistic wireless propagation conditions alongside portable high-throughput decoder implementations. The present study addresses these gaps by benchmarking SOC+MTD against state-of-the-art LDPC and turbo codes in standardized fading environments and by developing an OpenCL-based parallel decoder suitable for heterogeneous systems, including embedded platforms.

The study is organized as follows: Section II reviews related work in the area of error correction schemes for MIMO-OFDM systems and their software implementations. Section III details the MTD encoding and decoding algorithms, providing a theoretical background. Section IV describes the materials and methods, including the simulation environment and experimental parameters. Section V presents the results covering the MTD performance in multipath channels (A), STC integration (B), and Graphics Processing Unit (GPU)-accelerated simulation (C). Section VI summarizes the conclusions.

II. RELATED WORKS

This section details the outcomes of a review of relevant research on improving error correction and modulation techniques to address issues such as intersymbol interference (ISI) and erasures in MIMO-OFDM wireless communication systems, with a focus on 5G and IoT applications.

Information transmitted over communication channels can be modified by noise and interference, resulting in the data received at the receiver side not matching the data at the transmitter side. To assess the reliability of data transmission, the performance of communication channels is determined by the parameters Signal-to-Noise Ratio (SNR) and BER, or Block Error Rate (BLER) [29]. The higher the SNR, the higher the communication quality and the fewer errors. The lower the BER, the better the quality of data transmission. Thus, the correlation between BER and SNR is inversely proportional and nonlinear. Channel coding does not change the physical SNR level of the channel. However, it significantly improves the efficiency of using this SNR. Due to its error correction capability, a channel-coded system can achieve the same BER as a system without coding, but under conditions of much lower SNR. Intersymbol interference and fading do not reduce SNR. They are additional sources of distortion that lead to an increase in BER even at high SNR. FEC schemes encode data by adding redundant bits before transmission. At the receiving side, error correction is performed by reconstructing the corrupted data based on channel code decoding algorithms. Coding methods do not change SNR, but significantly improve the relationship between BER and SNR. They allow achieving much lower BER at the same SNR level, which makes communication more reliable and efficient.

Channel codes have been developed to an advanced level in ensuring the fidelity of modern wireless communication systems. The evolution of forward error correction techniques aligns with the evolution of mobile communication systems: from simple convolutional codes in 2G to more robust turbo codes used in 3G/4G and finally special classes of LDPC and polar codes used in 5G. The primary task at hand is about discovery and fine-tuning applicable solutions related to massive MIMO, OFDM systems focusing on fast as well as efficient software plus hardware implementations of encoders together with decoders capable of meeting required performance.

Turbo codes, LDPC, and polar codes were simulated under MATLAB by the authors in [30] to show performance estimates for these codes under the effects of intersymbol interference. It was a simulation of a channel model with binary phase shift keying Binary Phase Shift Keying (BPSK) and additive white

Gaussian noise (AWGN). Two types of equalizers at the receiver front end were used by the researchers to mitigate the adverse effects due to ISI: ZF equalizer and Minimum Mean-Square Error (MMSE) equalizer. Simulation results indicated that iterative LS estimation of channel pulse response improved the performance of all three codes - LDPC codes, polar codes, and turbo codes within the operational SNR range. The LDPC codes proved themselves as the best code in correcting errors over channels suffering from intersymbol interference.

Marques da Silva et al. [31] investigated MIMO-OFDM systems with LDPC for underwater acoustic communication (UWA) systems. UWA face multipath propagation conditions [32] from reflections off the water surface and bottom as well as obstacles, which reduces their quality. To beat the intersymbol interference, the authors incorporate OFDM, single carrier frequency domain equalization (SC-FDE), and MIMO system with LDPC codes that have been used in recently developed generations of cellular communications known as 5G [33]. Simulation results for UWA communication with 4×32 MIMO showed the performance improvement obtained with LDPC codes to be in the order of 5 dB. The experiments demonstrated significant data transmission rate up to 125.7 kbps, as well as spectral efficiency up to 3.5 bps/Hz.

Tikka and Sivashanmugam [34] proposed an error correction hybrid coding algorithm (EC-HCA) combining quasi-cyclic LDPC (QC-LDPC) and space-time block coding (STBC) for MIMO-OFDM systems. This approach mitigates ISI by leveraging STBC to avoid interference among transmit antennas, achieving a peak signal-to-noise ratio (PSNR) of 10.24 dB in a 2×2 MIMO setup, outperforming methods like Alamouti encoding [35], polar-convolutional coding [36], and optimal power allocation scheme with turbo codes [37].

To satisfy the bandwidth starvation codes and tall decoders of performance based on parallel computation technologies are being assembled by wireless communication systems engineers and researchers. Along with hardware platforms (GPU, Field-Programmable Gate Array (FPGA), SoC), the choice of application programming interface (API), CUDA or OpenCL is critical for developing high-performance encoders and decoders [38].

In [39], a LDPC decoder that can run as fast as 10 Gbps on a GPU device (GeForce RTX 2080Ti) has been described. Performance of (8448, 26,112) LDPC decoder has been tested under an AWGN channel with BPSK modulation. The software implementation of this decoder has been written in C using CUDA technology. SIMD instructions multiply by four the number of code words processed and further paralleling between code words. The early termination (ET) mechanism reduces the number of decoding iterations at high SNR, thereby almost doubling that channel throughput which is achievable in modern designs while keeping the same error correction performance [40-42].

The initial validation of FEC decoders prior to hardware implementation and integration with communication systems has consistently been hard work. A hardware-software methodology utilising the Xilinx Alveo U200 device and its parallel execution in OpenCL has been detailed in [43]. The

results demonstrate significant acceleration in both the HDL modelling process and FPGA prototype methodologies.

This study investigates the performance of MTDs for SOC in MIMO-OFDM wireless channels with multipath fading and ISI. Like traditional threshold decoders it is based primarily upon short-integer addition, i.e., check summing. Thus, the performance limitations of such a decoder reside only in the maximum data flow rate shift registers within the decoder can attain and how many parallel registers are being employed within the decoding algorithm itself. Among the quickest building blocks in current digital technology are single-bit modulo-2 adders, adders designed for small integers, and standard shift registers. Using this approach for hardware implementation of an MTD results in approximately three orders of magnitude higher performance relative to other algorithms under conditions of high noise. An MTD built upon Altera FPGA that has 40 decoding iterations yields about 1.6 Gbits/s decoding throughput [44]. In [45], the CUDA decoder gets to run at about 350 Mbps on NVIDIA GTX970 and 815 Mbps on NVIDIA GTX1080 for the code with block length equal to 1600 symbols.

The review is about how actively code methods against intersymbol interference and fading in modern wireless communications are developed. From simple convolutional codes to more complex turbo and LDPC codes, researchers constantly algorithmically enhance efficiency. Particularly, LDPC codes perform well over ISI channels. But even though modern decoders like LDPC do perform very well, their hardware implementation and complexity are an issue so that powerful computing platforms (GPU, FPGA) and special APIs (CUDA, OpenCL) have to be used. It is on this note that MTDs have been considered an adequate solution. Because of low computational complexity based on simple integer operations, MTDs have enormous prospects for parallel implementations and attaining startling performances (on the order of gigabits per second) on FPGAs. They become the code of preference in those channels with large intersymbol interference and fading since decoding must be done efficiently with low hardware implementation costs.

III. BACKGROUND

This section describes the encoding and decoding algorithms for SOC based on threshold decoding. These algorithms enable low-complexity error correction, critical for high-speed wireless systems like MIMO-OFDM.

A. Self-Orthogonal Codes (SOCs) and Encoder for SOCs

Self-orthogonal codes (SOCs) are a special class of convolutional codes that allow to simplify the decoding process. Fig. 1 shows a convolutional SOC with a code rate $R = 1/2$, minimum code distance $d = 5$, code length $n = 14$, and generator polynomial $g(x) = 1 + x + x^4 + x^6$, and tap weight $J = 4$. The encoder processes information bit by bit. For each bit, it feeds the bit into a shift register. Simultaneously, it calculates a parity bit using a four-input semi-adder. Both the information bit and the calculated parity bit are then sent out.

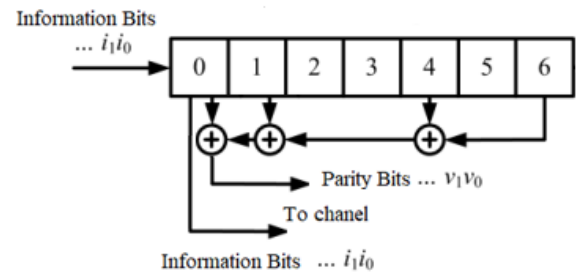


Fig. 1. Encoder scheme for convolutional SOCs.

This code corresponds to the parity-check matrix H :

$$H = \begin{bmatrix} 1 & & & & & & 0 \\ 1 & 1 & & & & & \\ 0 & 1 & 1 & & & & \\ 0 & 0 & 1 & 1 & & & \\ 1 & 0 & 0 & 1 & 1 & & \\ 0 & 1 & 0 & 0 & 1 & 1 & \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} : I \quad (1)$$

Within the matrix H , consider the rows numbered 0, 1, 4, and 6. In the first column of these rows, there is a single '1'. Furthermore, in all other columns of these same rows, there is at most one '1' present.

The encoding process operates as follows:

- 1) The shift register receives an information vector $i = (i_0, i_1, \dots, i_6, \dots)$ symbol by symbol, with i_0 in cell 7, i_1 in cell 6, and so forth.
- 2) The j -th parity symbol is calculated by performing a cyclic shift for j between 0 and 6:

$$v_j = \sum_{k=1}^4 i_{(j-g_k) \bmod 7} \quad (2)$$

- 3) A parity vector $v = (v_0, v_1, \dots, v_6, \dots)$ is then sent over the channel in combination with the information vector.

The encoding process is formalized in Algorithm 1, using the following parameters:

nk - number of the information branches;

nr - number of the parity branches;

K - registers length (the length of the one information branch and the length of the one parity of branches);

$poly$ - generator polynomial;

$poly(u, j, p)$ - position of p -tap from u -th information branch to j -check branch

J - array of number of the taps;

$J(i, j)$ - number of the taps from i -th information branch to j -check branch;

$\%$ - binary operator yields the remainder from the division of the first expression by the second;

$\oplus$ - addition modulo 2.

$\ominus$ - addition modulo 2.

Algorithm 1: Encoding algorithm using self-orthogonal codes

Input: Information array $i[nk, K]$
Output: Check array $v[nr, K]$

```

1  For  $k \leftarrow 0$  to  $K-1$  do
2      For  $j \leftarrow 0$  to  $nr-1$  do
3           $v(j,k) \leftarrow 0$ 
4          For  $u \leftarrow 0$  to  $nk-1$  do
5              For  $p \leftarrow 0$  to  $J(u,j)-1$  do
6                   $v(j,k) \leftarrow v(j,k) \oplus i(u, \text{poly}(u,j,p)+k) \% K$ 
7              end
8          end
9      end
10 end

```

To decipher the code, the received channel vector, represented as $\bar{Q}$ (which is the sum of the transmitted code word $\bar{A}$ and the noise vector $\bar{E}$), is multiplied by the code's checking matrix. This operation yields a vector known as the syndrome:

$$\bar{S} = H\bar{Q} = H(\bar{A} + \bar{E}) = H\bar{A} + H\bar{E} = H\bar{E}. \quad (3)$$

Detailed explanations of the theoretical underpinnings and mathematical formulas concerning parity-check relationships and the orthogonality characteristic of self-orthogonal codes (SOC) can be found in [46, 47]. These codes will be explored in the subsequent sections.

B. Threshold Decoding for Convolutional SOC's

Threshold decoding of convolutional self-orthogonal codes (SOC) was proposed by J. L. Massey. This method uses the principle of local parity checks and voting. It is a simplified version of decoding that does not require complex calculations, such as the Viterbi algorithm [48]. The main idea of the method is that for each information bit to be decoded, several local syndromes are calculated. Due to the self-orthogonality property, each of these checks contains, in addition to the decoded bit, no more than one other error. If most of the checks for one bit indicate the presence of an error (i.e., their syndromes are nonzero), then a decision is made to correct this bit. The threshold decoding method is simple to implement, as can be seen in Fig. 2, which shows a prototype of a threshold decoder for convolutional SOC's with $R = 1/2$, minimum code distance $d = 5$, code length $n = 14$, tap weight $J = 4$ and generator polynomial $g(x) = 1 + x + x^4 + x^6$.

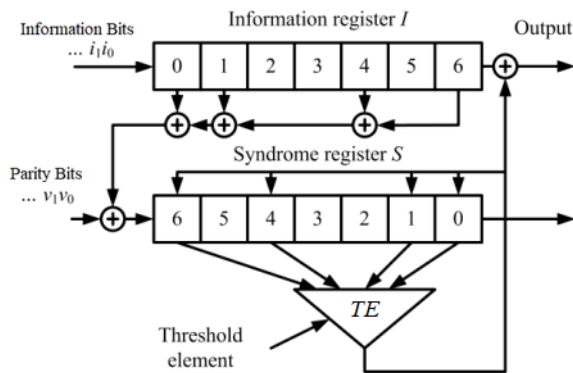


Fig. 2. Threshold decoder scheme for convolutional SOC's.

The threshold element (TE) aggregates the checks associated with the current decoded information bit and compares the sum with the threshold value. The threshold value of a given code is calculated by:

$$T = \frac{J+1}{2} \quad (4)$$

The threshold decoding operates as follows:

1) Defining a set of check syndromes, for i -th decoded information bit S_i is a set of check syndromes. Each syndrome in this set, $S_{i,k}$, contains as an added the error associated with the i -th bit and, according to the self-orthogonality property, at most one other error.

2) For each set of check syndromes S_i , a checksum L_i is calculated as the sum modulo 2 of all syndromes:

$$L_i = \sum_k S_{i,k} \quad (5)$$

3) The resulting checksum L_i is compared with a given threshold T . If $L_i > T$, it means that most of the checks indicate an error in the i -th information bit. In this case, a decision is made to correct this bit.

4) After the bit is corrected, its value is used to recalculate the syndromes.

5) The decoder moves to the next information bit and repeats the process.

However, error propagation substantially hindered the success of this error correction technique. If the decoder incorrectly corrects one bit, this error can affect local checks for subsequent bits. As a result, instead of one erroneous bit, several can be incorrectly corrected, which leads to a significant deterioration in decoding quality.

C. Multi-threshold Decoding for Convolutional SOC's

Multi-threshold decoding is an iterative variant of threshold decoding that uses an additional difference register (D) to diminish the downstream effects of errors. Before decoding, the D vector is filled with zeros. At the first iteration, the MTD operates in the same way as a conventional threshold decoder. For i -th information symbol, it calculates the checksum L_i and makes a decision to change this symbol if L_i exceeds a given threshold. But the decisions made about changing bits are stored in the D register. Starting with the second iteration, the decoder begins to use the information accumulated in the D register. Now decisions about correcting bits are made not only on the basis of the current checksum, but also taking into account the contents of the D register.

Fig. 3 shows the schematic of two iterations of the MTD operation for convolutional SOC's with $R = 1/2$, minimum code distance $d = 5$, code length $n = 14$, tap weight $J = 4$ and generator polynomial $g(x) = 1 + x + x^4 + x^6$.

The decoding process computes, for a selected symbol i_j , the function:

$$L_j = \sum_{p \in \mathcal{O}_j} s_p + d_j \quad (6)$$

where, d_j is the difference vector element for i_j , s_p is the p -th syndrome element in the checks for i_j , and $\mathcal{O}_j$ is the set of checks for the j -th information symbol. The number of terms equals the

code distance d . If $L_j > T$, where $T = \lfloor (d-1)/2 \rfloor$, then i_j , its checks, and d_j are inverted, and decoding proceeds to another symbol i_m , $m \neq j$. If $L_j \leq T$, the next symbol is decoded immediately.

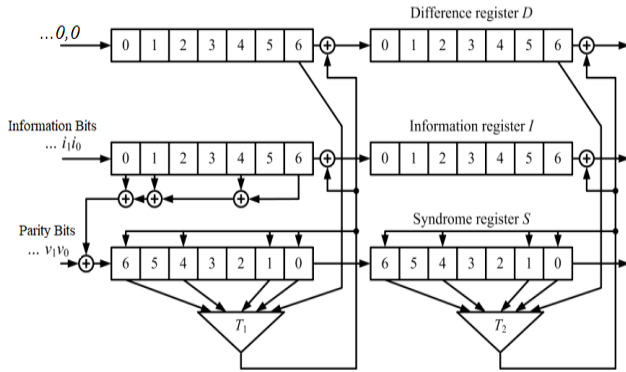


Fig. 3. Multi-threshold decoder scheme for convolutional SOCs.

The MTD algorithm is detailed in Algorithm 2.

One big problem with MTDs is that they tend to spread errors through feedback branches, which makes it more likely that syndrome errors and decoding failures will happen [25]. Mitigation means picking codes with the least amount of error set overlap and adjusting the parameters. Even so, MTDs are not very hard to decode because each step only needs simple additions and threshold comparisons.

Algorithm 2: Multi-threshold decoding algorithm for self-orthogonal codes

Input: Information array $i[nk, K]$, received from the noisy channel
Parity array $v[nr, K]$
 $maxIter$ -number of decoding iterations

Output: Information array $i[nk, K]$, after error correction procedure

```
1 Difference vector  $D[nk, K]$  is initially filled with zeros
2 For  $k \leftarrow 0$  to  $K-1$  do
3   For  $j \leftarrow 0$  to  $nr-1$  do
4      $s(j,k) = v(j,k)$ 
5     For  $u \leftarrow 0$  to  $nk-1$  do
6       For  $p \leftarrow 0$  to  $J(u,j)-1$  do
7          $s(j,k) \leftarrow s(j,k) \oplus i(u, (poly(u,j,p)+k)\%K)$ .
8       end
9     end
10    end
11  end
12 For  $iter \leftarrow 0$  to  $maxIter$  do
13   For  $k \leftarrow K-1$  to  $0$  do
14     For  $u \leftarrow 0$  to  $nk-1$  do
15        $sum \leftarrow D(u,k)$ 
16       For  $j \leftarrow 0$  to  $nr-1$  do
17         For  $p \leftarrow 0$  to  $J(u,j)-1$  do
18            $sum \leftarrow sum + s(j, (k - poly(u,j,p))\%K)$ .
19         end
20       end
21     If  $sum > T$  then
```

```
22    $i(u,k) \leftarrow 1 - i(u,k)$ 
23    $D(u,k) \leftarrow 1 - D(u,k)$ 
24   For  $j \leftarrow 0$  to  $nr-1$  do
25     For  $p \leftarrow 0$  to  $J(u,j)-1$  do
26        $s(j, (k - poly(u,j,p))\%K) \leftarrow 1 - s(j, (k - poly(u,j,p))\%K)$ 
27     end
28   end
29 endIf
30 end
31 end
32 end
```

For code distances $d < 25$ and the number of iterations of decoding Iter up to 50 iterations, the complexity is about:

$$N1 \approx (d + 2)(Iter + 4)$$

If you can deal with a 0.1 dB energy loss, the complexity can be cut down to

$$N2 \approx c1 d + c2 Iter$$

where, $c1$ and $c2$ are very small integers. Because of this simplicity, extra iterations don't have much of an effect on performance, which makes it possible to make software for 5G and IoT systems that works well.

IV. MATERIALS AND METHODS

A. Space-Time Coding in MIMO Systems

The research studies the efficiency of using MTD in communication channels with fading when using OFDM technology to combat multipath. When using one transmitting and one receiving antenna, this approach is not efficient enough, since the presence of deep and long fading within one code block usually leads to the fact that it will contain many errors even after decoding. To improve the reliability of data transmission, it is possible to additionally use the technology of spatial diversity during transmission and reception, i.e., to transmit and receive data using several antennas (MIMO systems). In this case, the antennas should be spaced so far apart that the correlation of signals on them is minimal. Then the probability that all radio signals between all antennas will simultaneously be subject to fading is quite small. As a result, the quality of communication is significantly improved with significantly lower energy costs. It should be noted that when using MIMO technology, there is a supplementary avenue to realize space-time coding. In this case, there are options for increasing the transmission rate (for example, when one antenna transmits one symbol - Spatial Multiplexing) or improving the energy efficiency of transmission (the signal transmitted by the antenna is a function of several transmitted symbols). This is determined by the space-time code matrix.

Currently, there are many proven space-time coding matrices, including those specified by standards. For example, the IEEE 802.16e (WiMAX) standard provides for the use of 1, 2, 3 and 4 transmitting antennas [49].

When using one transmitting antenna, space-time coding is not used.

For two transmitting antennas with STC the following transmitting matrices can be used:

$$A = \begin{bmatrix} S_i & -S'_{i+1} \\ S_{i+1} & S'_i \end{bmatrix} \quad \text{Orthogonal Alamouti Matrix, code rate 1}$$

$$B = \begin{bmatrix} S_i \\ S_{i+1} \end{bmatrix} \quad \text{Non-orthogonal V-BLAST Matrix, code rate 2}$$

$$C = \frac{1}{\sqrt{1+r^2}} \begin{bmatrix} S_i + jrS_{i+3} & rS_{i+1} + S_{i+2} \\ S_{i+1} - rS_{i+2} & irS_i + S_{i+3} \end{bmatrix} \quad \text{Non-orthogonal Matrix, code rate 2}$$

$$r = \frac{-1+\sqrt{5}}{2},$$

where, S_i is the transmitted symbol of the signal constellation; j is the imaginary unit; $()'$ is the complex conjugation operation.

For four transmit antennas with STC the following transmit matrices can be used:

Orthogonal, code rate 1:

$$A = \begin{bmatrix} S_i & -S'_{i+1} & 0 & 0 \\ S_{i+1} & S'_i & 0 & 0 \\ 0 & 0 & S_{i+2} & -S'_{i+3} \\ 0 & 0 & S_{i+3} & S'_{i+2} \end{bmatrix}$$

Non-orthogonal, code rate 2:

$$B = \begin{bmatrix} S_i & -S'_{i+1} & S_{i+4} & -S'_{i+5} \\ S_{i+1} & S'_i & S_{i+5} & S'_{i+4} \\ S_{i+2} & -S'_{i+3} & S_{i+6} & -S'_{i+7} \\ S_{i+3} & S'_{i+2} & S_{i+7} & S'_{i+6} \end{bmatrix}$$

$$C = \begin{bmatrix} S_i \\ S_{i+1} \\ S_{i+2} \\ S_{i+3} \end{bmatrix} \quad \text{Non-orthogonal, code rate 4}$$

To decode the STC code, a system of equations is first constructed

$$B = G S + N,$$

where, S is a column vector of transmitted signals; G is a matrix depending on the channel coefficients; N is complex Gaussian noise; B is a column vector depending on the received signals. From this expression, using a maximum likelihood detector, an array of logarithms of the likelihood ratio for the bits of each of the transmitted symbols is obtained.

As an example, let us consider how the matrix G and the vectors S and B are formed for two transmitting antennas using the STC matrix A :

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix},$$

where, S_1, S_2 are signals transmitted over two-time intervals.

$$B = \begin{bmatrix} R_1^{(1)} \\ R_1^{(2)'} \\ \dots \\ R_n^{(1)} \\ R_n^{(2)'} \end{bmatrix},$$

where, $R_k^{(m)}$ is the signal received by the k -th antenna at time m .

$$= \begin{bmatrix} h_{11}^{(1)} & h_{21}^{(1)} \\ h_{21}^{(2)'} & -h_{11}^{(2)'} \\ \dots & \dots \\ h_{1n}^{(1)} & h_{2n}^{(1)} \\ h_{2n}^{(2)'} & -h_{1n}^{(2)'} \end{bmatrix},$$

where, $h_{kp}^{(m)}$ is channel coefficient from the k -th transmitting to the p -th receiving antenna in the m -th time interval.

Three demodulation algorithms were evaluated:

$\hat{\Theta} = (G'G + 2\sigma_n^2 I)^{-1} G'S$ - MMSE Demodulation, where I is the identity matrix and σ_n^2 is the noise variance

$\hat{\Theta} = \frac{\sum_{\Theta} \Theta \cdot \exp\{-\frac{1}{2}(S-G\Theta)'(U)^{-1}(S-G\Theta)\}}{\sum_{\Theta} \exp\{-\frac{1}{2}(S-G\Theta)'(U)^{-1}(S-G\Theta)\}}$ - Optimal, Maximum-Likelihood (ML) Demodulation, where Θ is the constellation set.

Approximate ML Demodulation considers only the constellation point closest to the received signal for soft-decision computation.

Soft decisions from these algorithms are fed to the MTD for error correction, enhancing decoding performance in fading channels.

B. Simulation Details

Standardized wideband fading channel models were adopted to emulate urban, suburban, and rural deployments. The ITU-R channels [50] included Outdoor Channel A, Typical Urban 6 (TU6), and Rural Area 6 (RA6). TU6 was modeled with the delay-power profile [0, 0.2, 0.5, 1.6, 2.3, 5] μ s and [-3, 0, -2, -6, -8, -10] dB, respectively. RA6 used equally spaced taps with delays [0, 0.1, 0.2, 0.3, 0.4, 0.5] μ s and powers [0, -4, -8, -12, -16, -20] dB. To obtain the spatial structure essential for MIMO, the 3GPP SCM in Urban Macro (UMa) and Urban Micro (UMi) configurations was utilized, incorporating angle-of-arrival/angle-of-departure spreads and realistic base-station/user equipment (UE) heights [51, 52]. Doppler frequencies corresponding to speeds up to 50 km/h were applied to model mobility. These models were selected for their compatibility with WiMAX and 5G urban deployments, representing scenarios pertinent to IoT and low-altitude UAV communications characterized by multipath and spatial diversity.

OFDM with two Fast Fourier transform (FFT) sizes (512 and 1024 subcarriers) was used at the physical layer. To balance spectral efficiency and intersymbol interference (ISI)

robustness, cyclic prefix (CP) ratios of 1/8 and 1/16 were used in each scenario.

Quadrature Phase-Shift Keying (QPSK), 8 Phase Shift Keying (8PSK), 16 Adaptive Phase Shift Keying (16APSK), and 16QAM constellations were considered. For each scenario, the constellation order was chosen to find a balance between spectral efficiency and implementation complexity. In low-SNR or high-diversity setups, QPSK was used as the base; in the same band-width, 8PSK was used when moderate spectral efficiency was needed. 16APSK and 16QAM were enabled in higher-throughput settings and combined with stronger channel coding and/or spatial multiplexing as appropriate. In SCM-based MIMO studies, 16QAM was configured for cases emphasizing spatial multiplexing, whereas 16APSK was used in the OFDM settings following DVB-S2-compatible mapping.

Antenna configurations (1×1, 2×2, and 4×4) were configured. Space-time coding (STC) employed matrices A, B, and C to cover orthogonal and non-orthogonal designs at different code rates. For 2 transmit antennas, matrix A provided a rate-1 orthogonal design, while matrices B and C provided higher-rate, non-orthogonal options. For 4 transmit antennas, both rate-1 orthogonal and higher-rate non-orthogonal designs were instantiated. Selection of STC and spatial mode (diversity or multiplexing) depended on the target spectral efficiency and channel model. Receiver processing assumed per-subcarrier channel state information adequate for STC decoding.

Demodulators included MMSE and approximate ML detectors. Detector choice was aligned with the modulation order and MIMO/STC setting to balance complexity and soft-information quality.

Coding options covered SOC decoded via MTD, LDPC codes [53] from IEEE 802.16e (WiMAX) and DVB-S2, and turbo codes [54]. Code lengths in the study included 2,016 bits (WiMAX LDPC), ~10,000 bits (turbo), 16,200 and 64,800 bits (DVB-S2 LDPC), and SOC lengths around 36,864 and 64,800 bits. LDPC decoding used a normalized or offset min-sum family implementation, while turbo decoding used max-log-MAP. Table I shows a summary of the most important simulation settings and options.

The simulation environment was built in MATLAB, with C employed to accelerate the performance of the encoder, decoder, and other complex processing blocks requiring significant computational resources. The simulations were done on a workstation that met the following requirements. The simulation took place on a workstation that had the features shown in Table II.

C. Software Implementation of the MTD for SOC

In addition to a previous implementation of a multi-threshold decoder (MTD) for self-orthogonal codes (SOCs) on CUDA, a portable OpenCL implementation is being developed for heterogeneous devices, including systems-on-chips and FPGA accelerators. The primary goal is not to outperform existing CUDA kernels on NVIDIA GPUs, but rather to: i) create an implementation functionally equivalent to the base CUDA version, and ii) evaluate the performance of the same multi-threshold decoding algorithm for self-orthogonal codes on non-NVIDIA architectures.

TABLE I. THE CHANNEL'S EXPERIMENTAL PARAMETERS

Component	Specification/parameter
Channel	ITU-R: Outdoor A, TU6, RA6. SCM: UMa, UMi.
Mobility	0–50 km/h (Doppler per carrier frequency).
OFDM	FFT=512 or 1,024; CP=1/8 or 1/16
MIMO/STC	1×1, 2×2, 4×4; STC matrices A (orthogonal, rate 1), B (non-orthogonal, rate 2), C (non-orthogonal, rate 2 or 4).
Modulation	QPSK, 8PSK, 16APSK, QAM16
Demodulation	MMSE; approximate ML
Coding	SOC+MTD (36,864 and 64,800 bits; ≤25 iterations); LDPC (WiMAX 2016 bits; DVB-S2 16,200 bits, 64,800 bits); Turbo (≈10,000 bits).
Decoding	SOC (MTD, ≤25 iterations); LDPC (min-sum) Turbo (max-log-MAP)
Code rates	SOC R≈1/2; DVB-S2 LDPC R=1/2; Turbo R=1/2

TABLE II. WORKSTATION SPECIFICATIONS

Component	Specification
Processor	Intel Core i7 4790K
Motherboard	ASUS Z97
Graphics Card	NVIDIA Quadro K4000
RAM	Kingston 16 GB
HDD	Western Digital RE 4 TB

The decoder is implemented in OpenCL 1.2. The OpenCL kernels replicate the SOC+MTD algorithmic pipeline used in the CUDA version, including fixed-point LLR computation, threshold and multi-pass updates, syndrome checks, and early termination logic (see Fig. 4).

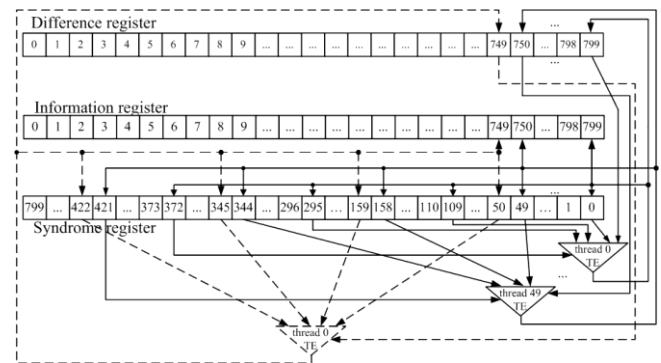


Fig. 4. Scheme of multi-threshold decoder using 50 concurrent threads.

V. RESULTS AND DISCUSSION

A. Performance of MTD for SOC and Other Error Correction Methods in Multipath Channels

This section presents simulation results evaluating the performance of multi-threshold decoders (MTDs) for SOC and

other error correction methods in standard ITU-R and SCM channel models. For MTD simulations, extensive parameter optimization was performed, including the selection of the code, number of decoding iterations, threshold values, weight coefficients for each iteration, and the method for determining syndrome weights.

Fig. 5 illustrates the BER performance of MTDs for a self-orthogonal code (SOC) with a code rate of 1/2 and a length of about 32,000 bits on the signal-to-noise ratio per bit in a channel of the Outdoor Channel A type using different types of modulation (curves labeled "SOC, MTD, ..."). In this case, a demodulator was used that formed only hard decisions regarding the decoded bits (the use of soft decisions of the demodulator will improve the characteristics by another 1..1.5 dB). OFDM with 1,024 carriers with the main parameters from the IEEE 802.16e (WiMAX) standard was also applied. The guard interval represented one-sixteenth of the OFDM symbol's total duration. Note that when using 8PSK modulation, the loss in energy compared to QPSK is about 3 dB, and when using 16APSK modulation, the loss is about 5 dB. At the same time, these types of modulation allow increasing the bit rate of transmission by 1.5 and 2 times, respectively, without expanding the frequency band.

The same figure shows the "turbo..." curve characteristics of a turbo code with a code rate of 1/2 and a code block length of about 10,000 bits using QPSK modulation with the same other channel and OFDM parameters. The constructive length of the component codes was equal to 4. The max-log-MAP algorithm for decoding the component codes was used to decode the turbo code. Note that the turbo code characteristics are significantly worse than the MTD characteristics. This is explained by the fact that such a code length is insufficient to cope with fading in the communication channel. The characteristics of a fairly short low-density code of the IEEE 802.16e (WiMAX) standard with a length of 2,016 bits with the same other parameters of the data transmission system are shown in Figure 5 by the "LDPC WiMAX, ..." curve. The min-sum decoding algorithm was used to decode these and other low-density codes. It should be noted that this code copes with fading as poorly as the turbo code. When using longer low-density codes of the DVB-S2 standard of 16,200 and 64,800 bits with a code rate of 1/2, the characteristics shown by the curves "LDPC DVB-S2 16,200, ..." and "LDPC DVB-S2 64,800, ..." are obtained, respectively. These codes are already capable of providing characteristics somewhat better than MTD. However, as already noted, the complexity of implementing their decoders is tens of times greater than that of MTD. Moreover, for self-orthogonal codes used in conjunction with MTD, with an insignificant increase in complexity (approximately 4 times), it is possible to ensure the characteristics shown in Figure 5 by the curves "SOC, min-sum, ..." for the QPSK, 8PSK, and 16APSK modulation types. These results are better than the characteristics of the basic MTD by approximately 1..1.5 dB and comparable to the results provided by low-density codes of length 16200 of the DVB-S2 standard.

Similar performance characteristics were observed for other ITU-R channel models. The main conclusions from these results are consistent with those already made.

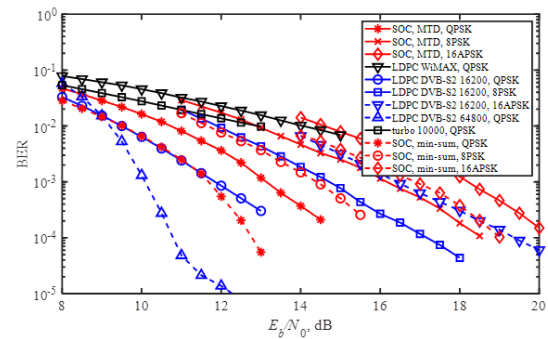


Fig. 5. Simulation results in ITU-R outdoor channel A.

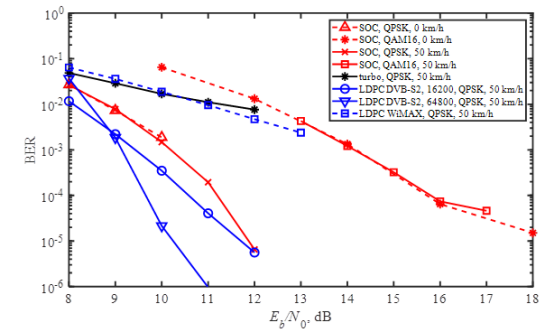


Fig. 6. Simulation results in SCM Urban Macro channel.

Fig. 6 illustrates the behavior of these codes within the SCM of the Urban macro channel model (results were also obtained for other SCM types). The OFDM modulation parameters are the same as those in Figure 5. Shown here are the characteristics of self-orthogonal codes for QPSK and 16 Quadrature Amplitude Modulation (QAM16) at receiver speeds of 0 and 50 km/h relative to the transmitter (group of curves "SOC,...", as well as the characteristics of a turbo code of about 10,000 bits in length (curve "turbo..."), low-density codes of the DVB-S2 standard of 16,200 (curve "LDPC DVB-S2, 16,200...") and 64,800 (curve "LDPC DVB-S2, 64,800...") bits in length, and low-density codes of the 802.16e (WiMAX) standard (curve "LDPC WiMAX..."). Note that the movement of the transmitter and receiver at a speed of 50 km/h has virtually no effect on the obtained characteristics. And the ratio between the efficiency of turbo, short and long low-density codes and the multi-threshold decoder of self-orthogonal codes remains the same as for the ITU-R channel model.

B. Application of STC with MTD in Fading Channels

The simulation used a multi-threshold decoder with 25 decoding iterations for the constructed self-orthogonal code with a code rate of $R = 8/16$, a code distance of 17, and a length of 36,864 bits. OFDM multiplexing with 512 carriers was used in conjunction with MTD. The guard interval was 1/8 of the OFDM symbol length. Conventional QPSK was applied as a simulation. The channel simulation used the default six-path TU6 profile of COST recommendation 259 with a delay profile of [0 0.2 0.5 1.6 2.3 5] μ s and a power profile of [-3 0 -2 -6 -8 -10] dB. The maximum Doppler frequency F_d was equal to 0.

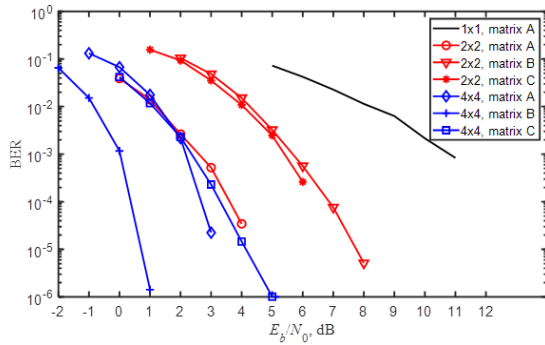


Fig. 7. MTD performance for different STC matrices and antenna configurations.

Fig. 7 presents the MTD attributes using different numbers of transmit and receive antennas (1×1, 2×2, 4×4) with different STC matrices. Note that the energy efficiency increases significantly with an increase in the number of antennas. Furthermore, for two transmit and receive antennas with the same STC rate, matrix C proves to be the best. When operating with four transmit antennas, matrix B, for which the STC code rate is 2, has the best energy efficiency. When switching to matrix C, which allows for the transmission of twice as much data in the same time, the energy efficiency degrades by approximately 2 dB.

Fig. 8 shows the MTD characteristics under the above-described conditions using various demodulation algorithms. Using the approximate optimal algorithm (curves marked "app opt"), the results are almost 2 dB better than those using the MMSE algorithm. This means that the quality of soft decisions under these conditions greatly influences the effectiveness of the error correction scheme.

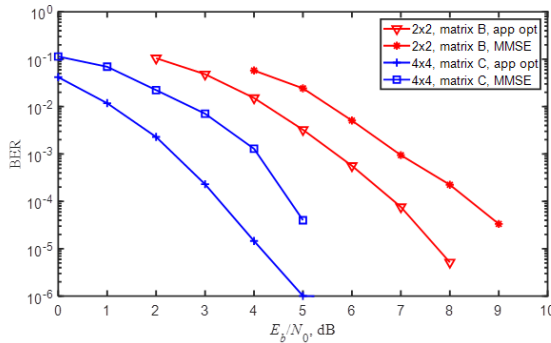


Fig. 8. MTD performance for different demodulation algorithms.

Fig. 9 evaluates MTD performance across different channel profiles. In addition to TU6, the six-beam RA6 profile from the same COST 259 recommendation was used here. It is characterized by a delay profile of [0 0.1 0.2 0.3 0.4 0.5] μ s (equidistant beams) and a power profile of [0 -4 -8 -12 -16 -20] dB. The MMSE algorithm was used for demodulation. Note that the reflected beams have lower power than in TU6. As a result, the efficiency of the data transmission system for such a channel is approximately 2 dB worse than for TU6.

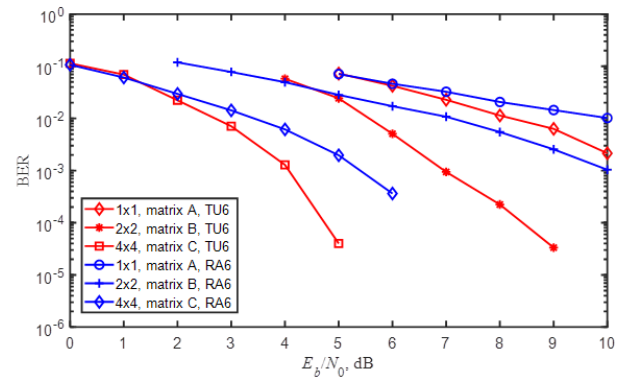


Fig. 9. MTD performance for different channel profiles.

The results of the comparison of the efficiency of the SOC with MTD and LDPC codes for the UMi SCM scenario using MIMO are presented below. To combat multipath, OFDM multiplexing with the parameters discussed earlier was again used. The noise-correcting codes were a self-orthogonal code with a code rate of 1/2 and a length of 32,768 bits, decoded using MTD, and an LDPC code of the DVB-S2 standard with a code rate of 0.44 and a length of 16,200 bits. In Fig. 10, the curves "SOC, 1x1, QPSK" and "LDPC, 1x1, QPSK" present the dependences of the decoding error probability of the self-orthogonal code and the LDPC code on the signal-to-noise ratio per bit using QPSK modulation. In this case, a demodulator was used that generates only hard decisions regarding the decoded bits. It should be noted that these codes provide approximately the same data transmission reliability. When switching to two transmit and two receive antennas, the system's energy efficiency decreases slightly (curves "SOC, 2x2, QPSK" and "LDPC, 2x2, QPSK"), but the bit rate doubles without expanding the used bandwidth.

Note that for MTD, the same increase in data rate is achieved by switching to QAM16 modulation instead of QPSK, but the efficiency of such a system (curve "SOC, 1x1, QAM16") is almost 2 dB worse than the MIMO variant. It should also be noted that the effect of using MIMO technology for self-orthogonal codes was greater than for LDPC codes. If the bit rate does not increase (using a single transmit antenna and QPSK modulation), using multiple receive antennas can significantly improve data transmission reliability. An example of the characteristics of such systems for two and three receiving antennas is shown in Fig. 10 by the curves "SOC, 1x2, QPSK" and "SOC, 1x3, QPSK". The gain compared to one receiving antenna with a target decoding error probability per bit of 10^{-5} was 4 and 7 dB, respectively. The same improvement is obtained when using LDPC codes (curve "LDPC, 1x2, QPSK").

From Fig. 10, it also follows that, when attempting to further increase the data transmission rate due to MIMO, the MTD characteristics deteriorate significantly (curves "SOC, 4x4, QPSK" and "SOC, 2x2, QAM16"). Therefore, such options for increasing the transmission rate are impractical to use in practice, or it is necessary to use additional methods to improve the characteristics.

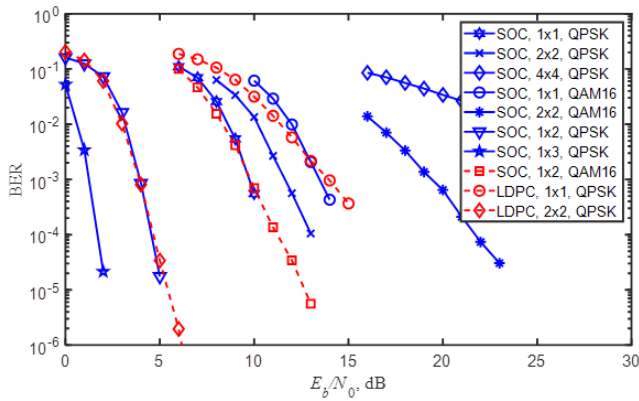


Fig. 10. Simulation results in the SCM urban micro channel with MIMO.

C. GPU-Accelerated Simulation Performance

A data-parallel scheme [55] is adopted in which each independent codeword is processed end-to-end by a single OpenCL work-item. All MTD stages—LLRs (Log-Likelihood Ratios) calculation, threshold tests, bit-updates, syndrome checks, and early stopping—are executed sequentially within that item, preserving the required ordering without cross-item synchronization. Parallelism is exposed by launching many such items concurrently, one per dataset. This maximizes device occupancy and eliminates global synchronization overhead, while local memory buffers are used to stage per-codeword data and tables.

```
kernel decode_mtd(codeword_i):  
    // local buffers for the i-th code word  
    preload_indices_to_local()  
    llr = load_llr_i()  
    for iter in 1..I_max:  
        apply_thresholds(llr) // sequential steps of the MPD  
        algorithm  
        update_bits_and_metrics()  
        if check_syndrome(): break  
    store_decision_i()
```

This approach achieved a simulation throughput of 300 kbps (see Table III), which, however, is insufficient.

Based on these unsatisfactory results, a study was conducted to identify the bottleneck limiting simulation throughput. This bottleneck turned out to be memory access. This is due to the specific memory architecture in OpenCL, where the data being processed can be stored in local or global memory [56]. Due to architectural features, accessing global memory takes significantly longer than accessing local memory. It was hypothesized that reducing global memory accesses and increasing local memory utilization would decrease the execution time of the encoder and decoder operations.

To mitigate the memory-access bottleneck identified in the OpenCL implementation, the data-parallel SOC+MTD scheme is reorganized to maximize local-memory utilization. Each

work-group processes a batch of K codewords: LLRs and index tables are fetched from global memory using coalesced vector loads and staged in double-buffered local memory (A/B). All MTD iterations then run in-place on local buffers; only final decisions and optional statistics are written back to global memory. Where supported, subgroup collectives (ballot/shuffle/pop count) accelerate thresholding and reductions. This scheme minimizes global-memory traffic, reduces synchronization overhead, and improves device occupancy.

TABLE III. DECODING SPEED OF MULTI-THRESHOLD DECODERS FOR SOCS

Ref.	Device (GPU/CPU)	Compute model	Throughput [Mbit/s]
[25]	Intel Core i7-4770 (4 cores)	Baseline (no GPU)	15
[45]	NVIDIA GeForce GTX 970	Standard GPU pipeline: CUDA kernels	350
[45]	Intel Core i7 4770K NVIDIA GeForce GTX 1080	Standard GPU pipeline: CUDA kernels	815
proposed	Intel Core i7 4790K NVIDIA Quadro K4000	OpenCL kernels (data-parallel)	120
proposed	Intel Core i7 4790K NVIDIA Quadro K4000	OpenCL kernels (local-memory optimized)	480

The proposed approach maximized GPU resource utilization, increasing the data transmission system simulation speed on a PC with an NVidia Quadro K4000 GPU graphics accelerator to 120 Mbps. As Table III shows, the OpenCL implementation with minimized global memory access resulted in an 8-fold increase in throughput compared to the CPU implementation, but significantly less than the parallel CUDA decoder.

The OpenCL implementation was designed primarily for portability, including deployment on OpenCL-enabled FPGAs, rather than for achieving performance levels comparable to the CUDA model. OpenCL remains the preferred method when cross-vendor support and FPGA portability are required, providing a unified approach for hardware implementation of channel coding systems.

The simulation results demonstrate that the MTD applied to SOCs exhibits competitive performance compared to traditional LDPC and turbo codes, especially under challenging multipath fading conditions. For instance, as depicted in Fig. 5, the SOC+MTD configuration achieves a BER of 10^{-5} at approximately 2 dB lower SNR compared to the turbo code of similar rate and block length. This is primarily due to the iterative threshold correction mechanism, which effectively adapts to rapidly varying error patterns caused by fading.

In contrast, short-length LDPC codes, such as the WiMAX-standard 2016-bit implementation, suffer from significant error floors in deep fade conditions. Even with powerful min-sum decoding and QPSK modulation, they are unable to correct clustered errors induced by delay spread and Doppler shift.

Longer LDPC codes, such as DVB-S2 (16200 bits, 64800 bits), achieve better BER performance, approaching that of SOC+MTD. However, they require significantly higher decoding complexity and memory bandwidth, making them less feasible for real-time or IoT scenarios.

Furthermore, integrating SOC+MTD with approximate ML demodulation results in performance gains of up to 2 dB over MMSE-based demodulation (see Fig. 8). This improvement highlights the benefit of combining soft-decision observations with MTD iterations, especially in high-diversity MIMO setups.

The introduction of antenna diversity (1×3) shows a significant reduction in required E_b/N_0 (≈ 7 dB at $BER = 10^{-5}$), demonstrating SOC+MTD's effective exploitation of spatial redundancy. In contrast, increasing modulation order from QPSK to 16APSK yields a spectral efficiency boost of $2\times$, with only moderate SNR penalties ($\sim 4\text{--}5$ dB). This confirms the scalability of SOC+MTD under different throughput constraints.

Finally, the OpenCL-based GPU implementation accelerates MTD decoding by a factor of $32\times$ (480 Mbps) compared to CPU-only processing (15 Mbps). Unlike specialized CUDA solutions, the portability of the OpenCL approach enables deployment across heterogeneous devices, including embedded SoCs, making SOC+MTD a practical high-throughput solution for 5G and IoT platforms.

In summary, the presented results confirm that SOC decoders using MTD offer a robust and computationally efficient alternative to conventional FEC methods under realistic wireless conditions. Their low complexity makes them especially suitable for IoT and real-time communication systems where hardware resources and latency budgets are constrained.

A comparative analysis clearly demonstrates that SOC+MTD outperforms short-length LDPC and turbo codes in fading conditions, while achieving DVB-S2 equivalent reliability with significantly lower decoding complexity.

VI. CONCLUSION

This study presents a comprehensive performance evaluation of Multi-Threshold Decoders (MTDs) for Self-Orthogonal Codes (SOCs) applied to modern broadband wireless communication systems, including OFDM and MIMO configurations with Space-Time Coding. Extensive simulations across standardized fading channel models (ITU-R and 3GPP SCM) demonstrate that SOC decoders using MTD achieve a level of reliability comparable to state-of-the-art LDPC and turbo codes, while requiring significantly lower decoding complexity. This advantage makes MTD particularly suitable for resource-constrained 5G and IoT devices.

The results further support the applicability of MTDs in high-mobility scenarios (up to 50 km/h) without significant performance degradation and demonstrate sensitivity to modulation and antenna diversity choices, enabling system designers to make informed trade-offs between energy efficiency and transmission rate. The integration with approximate Maximum-Likelihood detection and min-sum refinement leads to an additional SNR gain of up to 2 dB, while using a 1×3 antenna configuration can reduce the required E_b/N_0

by approximately 7 dB at a BER of 10^{-5} . GPU-based parallel implementation leveraging OpenCL achieved a $32\times$ improvement in simulation throughput, demonstrating the potential of software-based MTD acceleration for real-time applications.

However, the study is limited to static channel assumptions within standard models, and real-world scenarios with non-stationary or highly nonlinear interference environments were not considered. Furthermore, only conventional GPU architectures were evaluated for software acceleration; FPGA and heterogeneous SoC platforms remain unexplored.

Future research will focus on investigating hybrid SOC+MTD designs with machine learning-based decoding adaptation for dynamic IoT networks.

Overall, the findings underscore that MTD-enabled SOC decoders are a promising candidate for low-complexity, high-reliability forward error correction in next-generation wireless systems, especially where computational and energy constraints are critical.

ACKNOWLEDGMENT

The research was funded by the Science Committee of the Ministry of Higher Education and Science of the Republic of Kazakhstan within the grant "AP19679505 – Research for multi-threshold decoding of convolutional codes and their software and hardware implementation for high-speed radio channels with fading".

REFERENCES

- [1] M. Alsabah, M. A. Naser, B. M. Mahmood, S. H. Abdhussain, M. R. Eissa, A. Al-Baidhani, N. K. Noordin, S. M. Sait, K. A. Al-Utaibi, and F. Hashim, "6G wireless communications networks: A comprehensive survey," *IEEE Access*, vol. 9, pp. 148191–148243, 2021, <https://doi.org/10.1109/ACCESS.2021.3124812>.
- [2] S. R. Saunders and A. A. Aragón-Zavala, *Antennas and Propagation for Wireless Communication Systems*, 3rd ed., Chichester, UK: John Wiley & Sons, 2024, pp. 1–34.
- [3] A. F. Molisch, *Wireless Communications: From Fundamentals to Beyond 5G*, 3rd ed., Chichester, UK: John Wiley & Sons, 2022, pp. 19–22.
- [4] A. Darghouthi, A. Khelifi, and B. Chibani, "Performance analysis of 5G waveforms over fading environments," in *Proc. Int. Wireless Communications and Mobile Computing (IWCMC)*, Harbin City, China, June 28 – July 2, 2021, pp. 2182–2187, <https://doi.org/10.1109/IWCMC51323.2021.9498589>.
- [5] J. Viana, J. Madeira, P. Sebastião, F. Cercas, A. Mihovska, and R. Dinis, "Increasing reliability on UAV fading scenarios," *IEEE Access*, vol. 10, pp. 30959–30973, 2022, <https://doi.org/10.1109/ACCESS.2022.3149588>.
- [6] L. Xiao, S. Li, Y. Qian, D. Chen, and T. Jiang, "An overview of OTFS for Internet of Things: Concepts, benefits, and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7596–7618, 2022, <https://doi.org/10.1109/IIOT.2021.3132606>.
- [7] A. Sayed, M. Khatun, T. Ahmed, A. A. Piya, P. Chakraborty, and T. Choudhury, "Performance analysis of OFDM system on multipath fading and inter symbol interference (ISI) using AWGN," in *Computational Intelligence in Pattern Recognition*, 1st ed., Singapore: Springer, 2021, ch. 3, pp. 25–36, https://doi.org/10.1007/978-981-16-2543-5_3.
- [8] A. Narasimhamurthy, M. Banavar, and C. Tepedelenlioglu, *OFDM Systems for Wireless Communications, Synthesis Lectures on Algorithms and Software in Engineering*, Switzerland: Springer Nature, 2022, pp. 1–66, <https://doi.org/10.1007/978-3-031-01513-7>.
- [9] S. A. Janawade, P. Krishnan, K. Kandasamy, S. S. Holla, K. Rao, and A. Chandrasekar, "A low-complexity solution for optimizing binary

- intelligent reflecting surfaces towards wireless communication," *Future Internet*, vol. 16, no. 8, Jul 2024, Art. no. 272, <https://doi.org/10.3390/fi16080272>.
- [10] P. Sharma, R. N. Tiwari, P. Singh, P. Kumar, and B. K. Kanaujia, "MIMO antennas: Design approaches, techniques and applications," *Sensors*, vol. 22, 2022, Art. no. 7813, <https://doi.org/10.3390/s22207813>.
- [11] R. Kadel, K. Paudel, D. B. Guruge, and S. J. Halder, "Opportunities and challenges for error control schemes for wireless sensor networks: A review," *Electronics*, vol. 9, 2020, Art. no. 504, <https://doi.org/10.3390/electronics9030504>.
- [12] M. Bettayeb, S. Ghunaim, N. Mohamed, and Q. Nasir, "Error correction codes in wireless sensor networks: A systematic literature review," in *Signal Processing, and Their Applications (ICCSPA)*, Sharjah, United Arab Emirates, Mar 2019, pp. 1–6, <https://doi.org/10.1109/ICCSPA.2019.8713725>.
- [13] *IEEE Standard for WiMAX 802.16-2004, IEEE Std 802.16-2004, Institute of Electrical and Electronics Engineers (IEEE)*, Oct. 1, 2004. Accessed: July 15, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/1350465>.
- [14] *European Standard EN 302 307 for Digital Video Broadcasting (DVB-S2)*. Accessed: July 15, 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302300_302399/30230701/01.04.01_60/en_30230701v010401p.pdf.
- [15] M. M. Ali, S. J. Hashim, M. A. Chaudhary, G. Ferré, F. Z. Rokhani, and Z. Ahmad, "A reviewing approach to analyze the advancements of error detection and correction codes in channel coding with emphasis on LPWAN and IoT systems," *IEEE Access*, vol. 11, pp. 127077–127097, 2023, <https://doi.org/10.1109/ACCESS.2023.3331417>.
- [16] I. El Kaime, R. Benkhrouya, A. Air Madi, and H. Erguig, "An optimized method for polar code construction," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, pp. 148–157, 2023, <https://doi.org/10.14569/IJACSA.2023.0140118>.
- [17] K. Arora, J. Singh, and Y. S. Randhawa, "A survey on channel coding techniques for 5G wireless networks," *Telecommunication Systems*, vol. 73, pp. 637–663, Apr 2020, <https://doi.org/10.1007/s11235-019-00630-3>.
- [18] N. Kumar, D. Kedia, and G. Purohit, "A review of channel coding schemes in the 5G standard," *Telecommunication Systems*, vol. 83, pp. 423–448, Jun 2023, <https://doi.org/10.1007/s11235-023-01028-y>.
- [19] C. Zhu, Y. He, and Z. Dou, "Polar code BP decoding optimization for green 6G satellite communication: A geometry perspective," *Axioms*, vol. 14, 2025, Art. no. 174, <https://doi.org/10.3390/axioms14030174>.
- [20] C. Shivappa, N. K. Ramaswamy, R. C. Mullegowda, R. K. Ramaswamy, and M. Srikanthaswamy, "Efficient Low-Complexity Encoding and Decoding Algorithms for Global Navigation Satellite Systems," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 25499–25506, Aug. 2025, <https://doi.org/10.48084/etasr.11457>.
- [21] A. Elsanousi and S. Oztürk, "Performance Analysis of OFDM and OFDM-MIMO Systems under Fading Channels," *Engineering, Technology & Applied Science Research*, vol. 8, no. 4, pp. 3249–3254, Aug. 2018, <https://doi.org/10.48084/etasr.2209>.
- [22] M. Rowshan, M. Qiu, Y. Xie, X. Gu, and J. Yuan, "Channel coding toward 6G: Technical overview and outlook," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2585–2685, 2024, <https://doi.org/10.1109/OJCOMS.2024.3390000>.
- [23] J. L. Massey, "Threshold decoding," Ph.D. dissertation, Dept. of Electrical Engineering, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, 1962.
- [24] V. Zolotarev and G. Ovechkin, "Algorithm of multithreshold decoding for Gaussian channels," *Automation and Remote Control*, vol. 69, pp. 1086–1100, 2008, <https://doi.org/10.1134/S0005117908060180>.
- [25] V. V. Zolotarev, I. U. B. Zubarev, and G. V. Ovechkin, *Optimization Coding Theory and Multithreshold Algorithms*. Geneva, Switzerland: International Telecommunication Union, 2015. Accessed: June 25, 2025. [Online]. Available: https://decoders-zolotarev.ru/wp-content/uploads/2020/11/zolotarev_itu.pdf.
- [26] M. Ullah, K. Okada, and H. Ogivara, "Multi-stage threshold decoding for self-orthogonal convolutional codes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 93, no. 11, pp. 1932–1941, 2010, <https://doi.org/10.1587/transfun.E93.A.1932>.
- [27] M. A. Ullah, R. Omura, T. Sato, and H. Ogivara, "Multi-stage threshold decoding for high-rate convolutional codes for optical communications," in *Seventh Advanced International Conference on Telecommunications (AICT 2011)*, St. Maarten, Netherlands Antilles, Mar 20–25, 2011, pp. 87–93.
- [28] G. Ovechkin, D. Satybaldina, Z. Sailaukyzy, G. Danenova, and N. Tashatov, "Methods of multi-threshold decoders use in MIMO systems and methods of assessing their performance," in *8th International Symposium on Innovative Approaches in Smart Technologies Innovative (ISAS)*, Istanbul, Türkiye, Dec. 2024, pp. 1–6, <https://doi.org/10.1109/ISAS64331.2024.10845577>.
- [29] M. Yang, C. Bian, and H. S. Kim, "OFDM-guided deep joint source channel coding for wireless multipath fading channels," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 584–599, 2022, <https://doi.org/10.1109/TCCN.2022.3151935>.
- [30] A.-M. Cuc, F. L. Mergoş, and C. Grava, "Performance analysis of Turbo codes, LDPC codes, and polar codes over an AWGN channel in the presence of inter symbol interference," *Sensors*, vol. 23, 2023, Art. no. 1942, <https://doi.org/10.3390/s23041942>.
- [31] M. Marques da Silva, R. Dinis, J. Aleixo, and L. M. Oliveira, "On the performance of LDPC-coded MIMO schemes for underwater communications using 5G-like processing," *Applied Sciences*, vol. 12, no. 11, 2022, Art. no. 5549, <https://doi.org/10.3390/app12115549>.
- [32] A. Ahmed, A. Taimur, R. Shahid, Z. Farid, K. Amanullah, and Z. Najam, "Parameters affecting underwater channel communication performance," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 350–356, 2018, <https://doi.org/10.14569/IJACSA.2018.091050>.
- [33] M. M. da Silva, R. Dinis, and G. Martins, "On the performance of LDPC-coded massive MIMO schemes with power-ordered NOMA techniques," *Applied Sciences*, vol. 11, 2021, Art. no. 868, <https://doi.org/10.3390/app11188684>.
- [34] V. G. Tikka and R. Sivashanmugam, "Error correction using hybrid coding algorithm for BER reduction in MIMO-OFDM," *International Journal of Intelligent Engineering & Systems*, vol. 15, no. 6, pp. 618–626, 2022, <https://doi.org/10.22266/ijies2022.1231.55>.
- [35] S. Ghazi-Maghrebi and B. Akbarian, "Improved performance of Alamouti scheme using cyclic delay diversity and Doppler diversity for MIMO systems-based," *Wireless Personal Communications*, vol. 116, pp. 1971–1992, 2021, <https://doi.org/10.1007/s11277-020-07775-4>.
- [36] A. Agarwal and S. N. Mehta, "PC-CC: An advancement in forward error correction using polar and convolutional codes for MIMO-OFDM system," *Journal of King Saud University – Computer and Information Sciences*, vol. 32, no. 8, pp. 917–927, 2020, <https://doi.org/10.1016/j.jksuci.2017.12.003>.
- [37] S. Pyla, R. K. Padma, and S. N. Bala, "Capacity and BER performance improvement in integrated MIMO-OFDM system using optimal power allocation, channel estimation, and turbo coding," *International Journal of Communication Systems*, vol. 34, no. 14, 2021, Art. no. e4915, <https://doi.org/10.1002/dac.4915>.
- [38] A. Asaduzzaman, A. Trent, S. Osborne, C. Aldershof, and F. N. Sibai, "Impact of CUDA and OpenCL on parallel and distributed computing," in *8th International Conference on Electrical and Electronics Engineering (ICEEE)*, Antalya, Turkey, 2021, pp. 238–242, <https://doi.org/10.1109/ICEEE52452.2021.9415927>.
- [39] J. Dai, H. Yin, Y. Lv, W. Xu, and Z. Yang, "Multi-Gbps LDPC decoder on GPU devices," *Electronics*, vol. 11, 2022, Art. no. 3447, <https://doi.org/10.3390/electronics11213447>.
- [40] R. C. Li, X. Zhou, H. Y. Pan, H. Y. Su, and Y. Dou, "A high-throughput LDPC decoder based on GPUs for 5G New Radio," in *IEEE Symposium on Computers and Communications (ISCC)*, Rennes, France, 2020, pp. 1–7, <https://doi.org/10.1109/ISCC50000.2020.9219558>.
- [41] C. Tarver, M. Tonnemacher, H. Chen, J. Z. Zhang, and J. R. Cavallaro, "GPU-based LDPC decoding for 5G and beyond," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 278–290, 2021, doi: <https://doi.org/10.1109/OJCAS.2020.3042448>.

- [42] J. Ling and P. Cautereels, "Fast LDPC GPU decoder for cloud RAN," *IEEE Embedded Systems Letters*, vol. 13, pp. 170–173, 2021, <https://doi.org/10.1109/LES.2021.3052714>.
- [43] D. Suárez, V. Fernández, H. Posadas, and P. Sánchez, "Accelerating the verification of forward error correction decoders by PCIe FPGA cards," *IEEE Embedded Systems Letters*, vol. 15, no. 3, pp. 157–160, 2023, <https://doi.org/10.1109/LES.2022.3218289>.
- [44] V. Zolotarev, G. Ovechkin, A. Issainova, D. Satybaldina, and N. Tashatov, "Effective multithreshold decoding algorithms for wireless communication channels," in *10th IEEE International Conference on Application of Information and Communication Technologies (AICT)*, Baku, Azerbaijan, 2016, pp. 1–5, <https://doi.org/10.1109/ICAICT.2016.7991760>.
- [45] V. Zolotarev, G. Ovechkin, P. Ovechkin, D. Satybaldina, N. Tashatov, and D. Sankibayev, "High throughput software multi-threshold decoder on GPU," in *Third IEEE International Conference on Mathematics and Computers in Sciences and Industry (MCSI)*, Chania, Greece, 2016, pp. 168–171, <https://doi.org/10.1109/MCSI.2016.039>.
- [46] J. Lieb, R. Pinto, and J. Rosenthal, "Convolutional codes," in *Concise Encyclopedia of Coding Theory*, W. C. Huffman, J.-L. Kim, and P. Solé, Eds., 1st ed., Boca Raton, FL, USA: Chapman and Hall/CRC, 2021, pp. 197–226, <https://doi.org/10.1201/9781315147901>.
- [47] X. Kai, X. Zhu, Y. Wang, and F. Fu, "Construction of binary self-orthogonal codes," *Applied Mathematics & Information Sciences*, vol. 18, no. 1, pp. 1–10, 2024, <https://doi.org/10.1007/s12095-023-00677-9>.
- [48] A. R. Buzdar, L. Sun, M. W. Azhar, M. I. Khan, and R. Kashif, "Area and Energy Efficient Viterbi Accelerator for Embedded Processor Datapaths," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 3, pp. 267–274, 2017, <https://doi.org/10.14569/IJACSA.2017.080355>.
- [49] *IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, IEEE Std 802.16e-2005*, New York, NY, USA: IEEE, 2006. Accessed: July 15, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/1603394>.
- [50] International Telecommunication Union, "Guidelines for evaluation of radio transmission technologies for IMT-2000," *Recommendation ITU-R M.1225*, Geneva, Switzerland: ITU, 1997. Accessed: July 15, 2025. [Online]. Available: <https://www.itu.int/rec/R-REC-M.1225>.
- [51] 3rd Generation Partnership Project (3GPP), "Spatial channel model for multiple input multiple output (MIMO) simulations (Release 6)," *3GPP TR 25.996 V6.1.0*, Sophia Antipolis, France: 3GPP, 2003. Accessed: July 15, 2025. [Online]. Available: <https://www.3gpp.org/dynareport/25996.htm>.
- [52] 3rd Generation Partnership Project (3GPP), "Study on 3D channel model for LTE (Release 12)," *3GPP TR 36.873*, Sophia Antipolis, France: 3GPP, 2014. Accessed: July 15, 2025. [Online]. Available: <https://www.3gpp.org/dynareport/36873.htm>.
- [53] D. J. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity check codes," *Electronics Letters*, vol. 32, pp. 1645–1646, 1996, <https://doi.org/10.1049/el:19961141>.
- [54] X.-G. Xia, "Understanding turbo codes: A signal processing study," *Journal of Information and Intelligence*, vol. 2, no. 1, pp. 1–13, 2024, <https://doi.org/10.1016/j.jiixd.2023.10.003>.
- [55] H. Xiao, B. Guo, H. Zhang, and C. Li, "A parallel algorithm of image mean filtering based on OpenCL," *IEEE Access*, vol. 9, pp. 65001–65016, 2021, <https://doi.org/10.1109/ACCESS.2021.3068772>.
- [56] Khronos Group, *The OpenCL Specification*, Version 3.0, Beaverton, OR, USA: Khronos Group, 2021. Accessed: June 25, 2025. [Online]. Available: https://www.khronos.org/registry/OpenCL/specs/3.0-unified/pdf/OpenCL_API.pdf.