

Improving Short-Block LDPC Decoding Using Perturbation-Based Rescue with Maximum Likelihood Based Candidate Validation

Haider Abdulkarim

University of Technology - Iraq

Hussein Khaleel

`hussein.j.khaleel@uotechnology.edu.iq`

University of Technology - Iraq

Marwa K. Farhan

Ministry of Higher Education and Scientific Research

Research Article

Keywords: LDPC, Channel Coding, URLLC, 5G

Posted Date: October 12th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10251569/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Improving Short-Block LDPC Decoding Using Perturbation-Based Rescue with Maximum Likelihood Based Candidate Validation

Haider Abdulkarim¹, Hussein Khaleel², Marwa K. Farhan³

¹College of Communications Engineering, University of Technology, Baghdad, Iraq; Email: haider.a.abdulkarim@uotechnology.edu.iq

²College of Communications Engineering, University of Technology, Baghdad, Iraq; Email: hussein.j.khaleel@uotechnology.edu.iq

³Scholarships and Culture Relations Directorate, Ministry of Higher Education and Scientific Research, Baghdad, Iraq; Email: marwa.farhan@moheer.edu.iq

Corresponding Author: Haider Abdulkarim, haider.a.abdulkarim@uotechnology.edu.iq

Abstract

Meeting the latency and reliability requirements of 5G Ultra-Reliable Low-Latency Communications (URLLC) is enabled through the adoption of short-block Low-Density Parity-Check (LDPC) codes. Traditional decoding of short-block LDPC using Belief Propagation (BP) may fail due to trapping sets, resulting in degraded decoding performance.

This paper proposes a three-stage decoding algorithm that combines low-iteration BP decoding, a perturbation-based rescue process, and Maximum Likelihood (ML) based candidate validation to improve decoding reliability. When the low-iteration BP decoding stage fails to converge, the rescue stage perturbs the least reliable bits to generate multiple decoding candidates, which are subsequently evaluated using an ML-based metric to select the most likely candidate and reduce miscorrections. The proposed algorithm was evaluated through MATLAB simulations with a short-block LDPC configuration for 5G URLLC systems. QAM modulation was used and two channel models were considered, Additive White Gaussian Noise and flat Rayleigh fading. The performance was investigated for different numbers of perturbed bits and different coding rates. The results show improved decoding performance compared to standard LDPC, achieving coding gains exceeding 1 dB at low bit error rates while reducing decoding failures caused by trapping sets.

Keywords: LDPC, Channel Coding, URLLC, 5G.

1. Introduction

One of the challenges in Fifth-Generation (5G) networks is the demand for growing reliability, aiming to keep packet error rates at their lower limits (10^{-6} range) as well as limiting the end-to-end delay below 1ms. 5G networks introduced significant improvements focusing on the use of Ultra Reliable Low Latency Communication (URLLC) data communication. This implies using short block codes, and avoid higher latency codes. Therefore, recent shorter coding schemes for channel coding begin to rise in this scope [1].

The transition toward 5G wireless networks introduces an incredibly tight bottleneck for URLLC. Increasing number of applications such as massive self-controlling swarms, remotely-controlled robot surgery, and fast internet requirements require two extreme conditions: maintaining Bit Error Rate (BER) strictly below the range of 10^{-5} to 10^{-6} , and restricting the end-to-end delays to below 1 millisecond. Physically realizing these microscopic delays mandates the abandonment of long data packets in favor of short block lengths (e.g., 500 bits). However, classical information theory dictates that as block length decreases, the asymptotic advantages of modern channel coding begin to collapse, placing immense pressure on the design of short-block error correction schemes [2].

Robert G. Gallager invented the Low Density Parity Check Codes (LDPC) as part of his Ph.D. dissertation at MIT in 1960 [3]. However, they were ignored in the next 30 years, partially because of its calculations overhead, requiring expensive hardware. It was not until mid-90s, when researchers have rediscovered and implemented the message passing decoding algorithm into a capable hardware [4]. It was shown that LDPC could reach near-Shannon capacity limit [4]. ultimately LDPC was adopted in the 5G New Radio (NR) standard [5].

The basic idea of LDPC is exchanging the soft information (certainty of bits) via a bipartite graph. The graph is random in long blocks, preventing looping-back of information on itself. On the other hand, cycles in shorter blocks are inevitable. When the transmission channel suffers from Rayleigh fading, certain bits suffer from deep fades, resulting in high ambiguous Log-Likelihood Ratio (LLR) that feeds the decoder. Those bits form a trapping set, that deceives the decoder with incorrect beliefs. If the decoder is limited by 100 iterations, for example, it might success to eventually escape the trapping set [6].

Given enough time, a standard Belief Propagation (BP) or Min-Sum decoder might eventually break free from a trapping set; running 50 to 100 iterations can often resolve these localized conflicts. However, the stringent latency budgets of URLLC completely eliminate this luxury, generally capping the allowable iterations to fewer than 10. When the decoder hits this hard iteration ceiling, it aborts the process, resulting in early convergence failure and driving up the error floor [7].

However, URLLC latency limitations require the Belief Propagation iterations not to exceed 10 iterations. To bridge the gap between necessary iteration limits and the resulting error floors, this paper proposes a cognitive three-Stage LDPC decoding algorithm. Rather than treating all frames equally, the proposed algorithm employs an asymmetric processing architecture. Standard frames are processed by a highly efficient, low-iteration BP decoder to minimize average latency. For the fraction of frames that stall in trapping sets, our proposed algorithm dynamically

identifies the weakest channel LLRs and deploys a massively parallelized, ML-bounded perturbation space to forcibly shatter the localized cycles. By strictly bounding the output against the raw channel reality, the proposed algorithm successfully navigates deep fades without sacrificing the strict latency constraints of the 5G standard.

The remainder of this paper is organized as follows. Section 2 provides a review on related literature regarding LDPC decoding techniques. Section 3 introduces the proposed three-stage decoding algorithm in detail and describes its operating principles. Section 4 presents the simulation setup and the results, and discusses the performance of the proposed algorithm. Lastly, Section 5 summarizes this study and suggests directions for future work.

2. Literature Review

Achieving URLLC is one of the main goals of 5G operation. Short-block LDPC coding represents a promising candidate to satisfy the stringent reliability and latency requirements of 5G URLLC. One major research challenge in the field of channel coding remains to be the handling of convergence failures.

The use of message passing algorithms is widely common in the decoding of LDPC codes, among which belief propagation (BP) algorithm delivers solid performance, although suffering from high computational complexity. The Min-Sum Algorithm (MSA) was introduced as a simplified version to reduce computations by introducing some approximations, however at the cost of reduced performance.

Normalized Min-Sum (NMS) and Offset Min-Sum (OMS) decoding algorithms [8] are proposed to overcome the limitations of MSA, however their performances are closely related to the structure of LDPC's parity-check matrix.

For short-block LDPC codes operating under severe channel conditions, such as Rayleigh fading, BP based decoders may require a large number of iterations to converge and can sometimes fail to converge altogether. In URLLC applications, however, the allowable number of decoding iterations is usually limited. Consequently, decoding performance degrades, and the error floor may be reached before the target reliability is achieved [9].

To address the limitations of iterative message-passing decoders, researchers have investigated Ordered Statistic Decoding (OSD) and hybrid BP-OSD schemes [10]. When BP decoding does not succeed, OSD reprocesses the received LLRs. First, the received bits are arranged according to their reliability, and a Most Reliable Basis (MRB) is determined. The actual values of those less reliable are consequently estimated by applying algebraic reprocessing.

OSD delivers excellent decoding performance for short block codes approaching the maximum likelihood error floor [11]. OSD involves Gaussian elimination and matrix inversion and suffers from high computation complexity that scales with the cube of the block length, leading to slow decoding of longer code blocks [12]. The computational overhead, power consumption, and latency make OSD impractical for 5G URLLC applications.

Perturbation based techniques are being explored to enhance the performance of LDPC decoding algorithms. This approach involves tweaking the decoder inputs when convergence stalls in order to escape from trapping sets, and letting the decoder converge to a valid codeword instead [13]. Multi round BP decoding with impulsive perturbation has shown promising results, as it introduces controlled perturbations during the decoding rounds in order to improve the performance of short-length LDPC codes [14].

Despite their effectiveness, perturbation-based methods present a significant challenge. By artificially modifying LLR values, the decoder output may satisfy all parity checks, however this sequence may differ from the actual transmitted message. This phenomenon is known as miscorrection, and may lead to degradation the overall BER performance [15].

This paper focuses on perturbation-based decoding. A detailed description of the proposed algorithm, including its operating principles and implementation, is described in the next section.

3. Methods

The proposed algorithm aims to enhance the performance of standard LDPC by resolving decoding stalls in short-block codes. The algorithm has three-stages. The first stage is a standard LDPC decoder with low iteration count. If the iterations are exhausted and convergence fails, the algorithm moves to the next stage and initiates a rescue process. In the second stage, the algorithm applies heavy perturbation to the codeword bits with the lowest absolute LLRs. These weak bits are forced to absolute confidence limits and help in breaking the trapping sets. Multiple decoded candidates are more likely to converge successfully at the output of this stage compared to standard LDPC. In the third stage, all decoded candidates are validated to eliminate miscorrections using a Maximum Likelihood (ML) based metric. A detailed explanation of the algorithm is presented next.

The architecture of the proposed algorithm is shown in Figure 1, Figure 2, and Figure 3. It operates on the decoder side of an LDPC scheme, and focuses on short-blocks and URLLC requirements. The architecture consists of three stages that lead to the decoding of received frames.

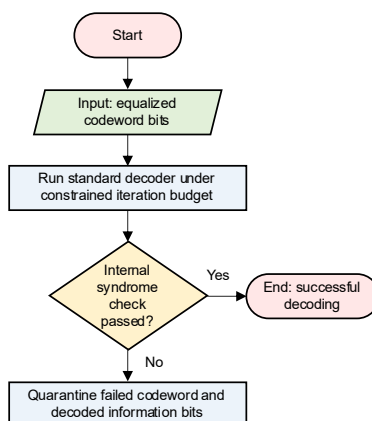


Figure 1: Stage 1 of the proposed algorithm.

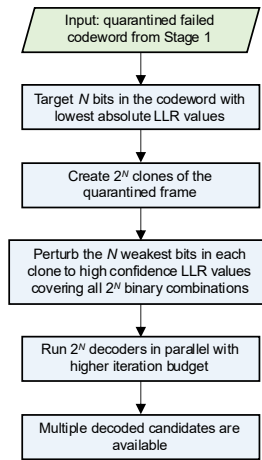


Figure 2: Stage 2 of the proposed algorithm.

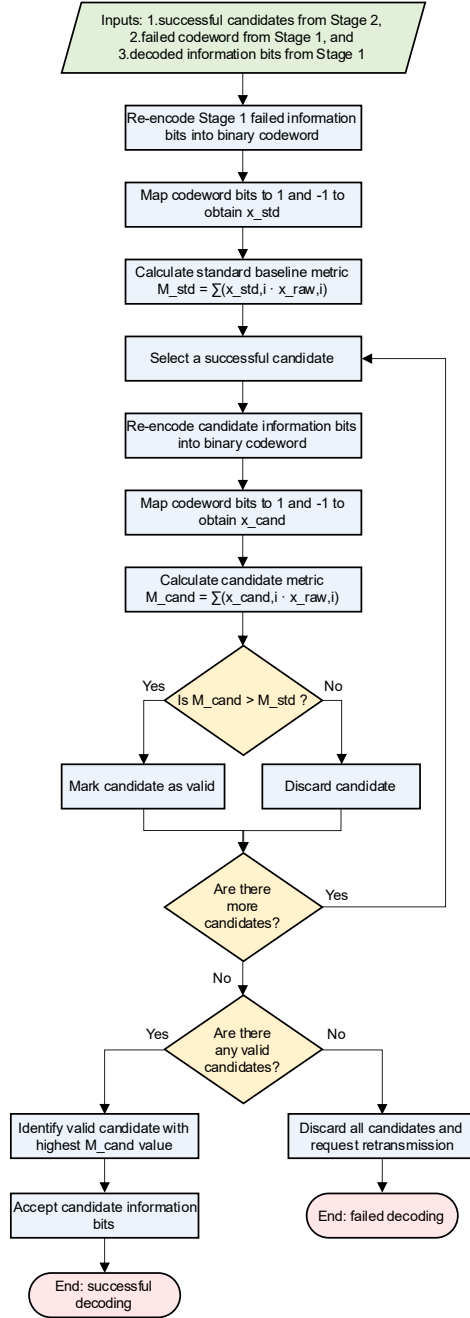


Figure 3: Stage 3 of the proposed algorithm.

Stage 1: Standard LDPC

At the first stage, the input is the raw, equalized codeword bits, that is fed into a standard LDPC decoder. This decoder runs under constrained iteration budget, where the small iteration count aims to maintain the ultra-low average latency required by 5G URLLC specifications [16].

If the internal syndrome check passes before this iteration limit, meaning that all parity equations are satisfied, then the decoding process terminates, and the recovered information bits at the output of this decoder are accepted. Under good channel conditions, this leads to a deterministic low latency processing time for the majority of transmissions [17].

However, if the decoder exhausts the full iteration budget without satisfying the syndrome, the algorithm recognizes that the decoder has failed to converge to valid information bits. In this case, the failed codeword is quarantined and passed to the second stage.

Stage 2: Perturbation Search Space

The input to the second stage is the quarantined frame, i.e. the frame that failed the decoding process. In this frame, the signs of the values represent the decision of the decoder (whether 0 or 1) and the amplitudes are the LLR values representing the decision confidence. Absolute LLR values near zero indicate high uncertainty, while higher absolute LLR values indicate higher confidence. The LLR is expressed as:

$$LLR(x) = \log \left(\frac{P(x = 0 | y)}{P(x = 1 | y)} \right) \quad (1)$$

Where x is the transmitted bit, y is the received channel observation, $P(x = 0 | y)$ represents the probability the current bit being 0 given the received value, and $P(x = 1 | y)$ represents the probability the bit being 1 given the received value.

Within the incoming frame, the algorithm targets a number of N bits that have the lowest absolute LLR values, corresponding to the least certain decoding decisions, and then 2^N clones of the quarantined frame are created. For each cloned frame, the N targeted weak bits are heavily perturbed by setting their LLRs to high confidence values (positive and negative), and these bits are mapped to cover all the 2^N binary combinations. This helps to move the decoder out of a trapping set's local minimum.

The 2^N perturbed frames are subsequently decoded. Multiple instances of the decoder can run in parallel to improve computation speed. The decoder at this stage is set with a higher iteration count compared to the first stage in order for the decoders to settle.

At this point, multiple decoders may pass the syndrome check, and one or more successfully decoded information bits will be available as candidates. In this case, decoders may run into the so-called miscorrection paradox [18]. That is, when LLR values are artificially altered, it is possible that the decoder converges to a codeword that is correct mathematically and it passes the parity check, but this codeword is different from the originally transmitted one. This causes degradation of the overall error rate.

Stage 3: Validation

The third stage of the proposed algorithm evaluates the available candidates and identifies the best among them. For every successful candidate in Stage 2, the algorithm re-encodes the information bits and reconstructs the codeword. These binary values of the codeword are then mapped to have values of 1 and -1, possibly using a standard BPSK representations. The re-encoded and mapped candidate is denoted as x_{cand} . To verify the legitimacy of an x_{cand} , its similarity against the raw, equalized codeword, denoted as x_{raw} , is calculated. x_{raw} is the input to the LDPC decoder in Stage 1. For a codeword of length L , the similarity metric is calculated using the dot product as follows:

$$M_{cand} = \sum_{i=1}^L x_{cand,i} \cdot x_{raw,i} \quad (2)$$

A standard baseline value for this similarity metric, denoted as M_{std} , is first calculated using the failed decoded information bits from Stage 1. These information bits are re-encoded to obtain the binary codeword, then transformed to have values of 1 and -1 (in the same way described previously to obtain x_{cand}). This codeword is denoted as x_{std} , and is used to calculate M_{std} in a similar manner to equation (2) as follows:

$$M_{std} = \sum_{i=1}^L x_{std,i} \cdot x_{raw,i} \quad (3)$$

The algorithm considers a candidate valid only if the value of its similarity metric is higher than that of the standard baseline ($M_{cand} > M_{std}$). The decoded information bits that are finally accepted correspond to the valid candidate that has the highest similarity metric M_{cand} . This candidate acceptance method ensures that the perturbation process can only improve upon the standard performance of the decoder, while successfully avoiding miscorrections and trapping sets.

4. Results and Discussion

A MATLAB based simulation of a transmitter, receiver, and communication channel was used to evaluate the proposed algorithm. QAM was selected for the modulation, and LDPC was selected as the coding scheme. Two channel models were considered, one with an Additive White Gaussian Noise (AWGN), and another incorporating AWGN and flat Rayleigh fading.

Two sets of simulations were carried out, the first set runs a decoder with fixed code rate of 1/2 while varying the number of target bits (N), and the second set considers a single value for the target bits (N) while using multiple code rates.

In all simulations, a codeword length of 504 was defined to capture the environment of a 5G URLLC short-block LDPC system.

Table 1 and Table 2 list the simulation parameters for the first and second simulation sets, respectively.

Table 1: Parameters of simulation set 1.

Information Bits	Codeword Length	Code Rate	N
252	504	1/2	5
252	504	1/2	6
252	504	1/2	7
252	504	1/2	8

Table 2: Parameters of simulation set 2.

Information Bits	Codeword Length	Code Rate	N
168	504	1/3	7
200	500	2/5	7
336	504	2/3	7
378	504	3/4	7

At the transmitter side, the original information bits are generated randomly, then encoded using LDPC. LDPC is provided by MATLAB's Communications Toolbox.

The parity-check matrix H used within LDPC was implemented using the Irregular Repeat Accumulate (IRA) structure [19]. The matrix layout is divided horizontally into two distinct sub-matrices: an information block and a parity block. The information block contains columns with 1s placed at variable indices, creating the "irregular" distribution. While the parity block is generated with a fixed, dual-diagonal staircase pattern of 1s (diagonal and sub-diagonal).

The irregular design of the information block enables the code to perform close to the theoretical Shannon limit, minimizing data corruption in noisy channels. While the dual-diagonal staircase structure of the parity block allows for fast encoding and simpler decoder hardware implementation.

The encoded bits are then modulated into QAM symbols, and transmitted through the channel. AWGN and Rayleigh fading with AWGN channel models were simulated. The signal at the receiver is defined as:

$$y = h \cdot x + n \quad (4)$$

where x is the transmitted QAM symbol, h represents the Rayleigh fading coefficients, and n is the AWGN vector.

At the receiver, the bits undergo equalization assuming exact knowledge of the channel. The equalized bits are demodulated and the soft LLRs are obtained. At this stage, the standard and the proposed LDPC decoders were both simulated. The standard LDPC decoder runs with 20 iterations. While the LDPC decoder of the proposed algorithm runs with 5 iterations at Stage 1, and 20 iterations at Stage 2. The codeword is decoded and the information bits are obtained for both decoders. The BER is calculated for each decoder (standard and proposed) by comparing the information bits out of the decoders against the original information bits from the transmitter.

The BER values are calculated for the range of Signal to Noise Ratio (SNR) from 0 to 8 dB. The BER is calculated for each SNR value, as the average of 100,000 transmitted frames. The performance of the proposed algorithm was evaluated and compared to a standard LDPC by measuring the BER versus SNR using Rayleigh fading and AWGN channels.

The results of simulation set 1, with code rate of $1/2$, are shown in Figure 4 (a), (b), (c), and (d) for target bits N equal 2, 4, 6, and 8 respectively. In all the plots, the proposed algorithm consistently outperforms the standard LDPC in both AWGN and fading channel scenarios.

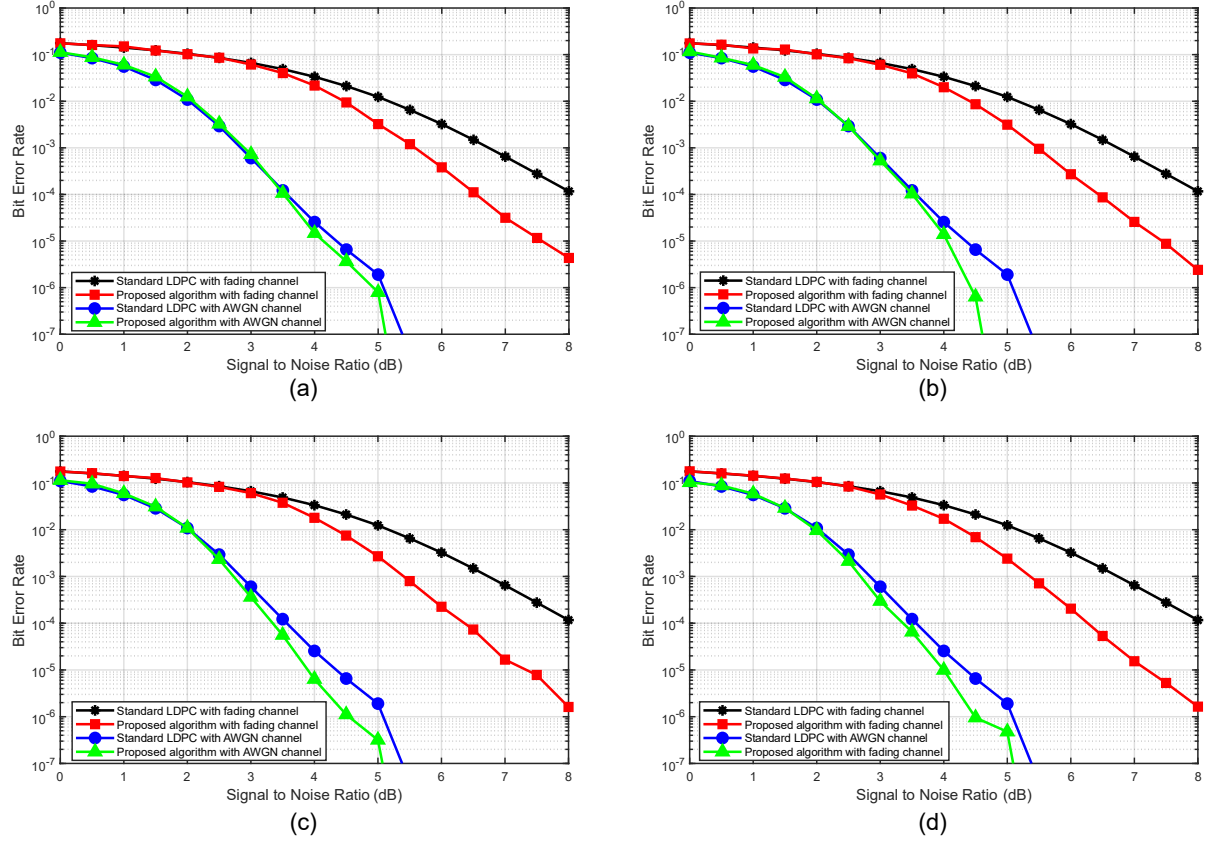


Figure 4: BER against SNR for simulation set 1, with code rate of 1/2. (a) Target bits = 2, (b) target bits = 4, (c) target bits = 6, and (d) target bits = 8.

Under the fading channel condition, the performance gain becomes more evident at moderate and high SNR values from 4 dB to 8 dB, where the slope of the BER curves show a steeper decline compared to the standard LDPC curves. This behavior demonstrates the effectiveness of the proposed algorithm in mitigating the impact of the channel and in reducing decoding errors.

Under AWGN channel, the proposed algorithm achieved lower BER values across all the evaluated SNR levels, however, the improvement is less significant compared to the fading channel scenario. The improvement is again more clear for higher SNR values.

Comparing subplots (a), (b), (c), and (d), increasing the number of target bits N further enhances the decoding performance. Considering the fading channel, for $N=2$, the BER improvement ranges from 1.5 times to 26 for the SNR range 4 dB to 8 dB. This improvement increases with N , where at $N=8$, the BER shows nearly 2 to 70 times improvement for the SNR range from 4 dB to 8 dB.

In terms of the coding gain, i.e. the reduction in the required SNR to achieve a certain BER, the proposed algorithm with $N=2$ achieves an SNR gain of approximately 0.7 dB at a BER of 10^{-2} , and 1.1 dB at a BER of 10^{-3} compared to standard LDPC. For $N=8$ the SNR gain ranges from 0.9 dB at a BER of 10^{-2} to 1.4 dB at a BER of 10^{-3} . Numerical BER values for the different SNRs

under fading channel, comparing standard LDPC and the proposed algorithm with different values of N are listed in Table 1.

Table 3: BER versus SNR for results set 1 under fading channel.

		SNR (dB)					
		3	4	5	6	7	8
BER	Std	0.0664	0.0335	0.0124	0.0032	0.0006	0.0001
	$N=2$	0.0613	0.0217	0.0032	0.0004	0	0
	$N=4$	0.0606	0.0200	0.0031	0.0003	0	0
	$N=6$	0.0606	0.0180	0.0027	0.0002	0	0
	$N=8$	0.0567	0.0171	0.0024	0.0002	0	0

Figure 5 shows the BER versus the target bits N . The plot shows that when N increases, the BER improvement increases as well, however, these gains diminish at higher values of N .

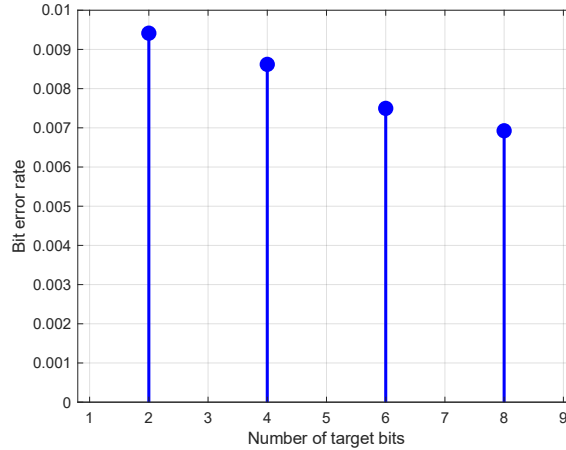


Figure 5: Bit error rate versus number of target bits for SNR = 4.5 dB and code rate 1/2.

The results of simulation set 2, with fixed number of target bits $N=4$ and variable code rates of 1/3, 2/5, 2/3, and 3/4, under fading channel conditions are shown in Figure 6 a, b, c, and d, respectively. The plots show that increasing the code rate causes the overall BER performance for both the standard LDPC and the proposed algorithm to degrade. Lower code rates introduce higher redundancy into the codeword, providing stronger error-correction capability and improved resistance to channel impact, and is more suited for harsh channel environments. In contrast, higher code rates transmit more information bits accompanied by fewer parity bits, which increases spectral efficiency but at the cost of lower error-correction performance, and this is more common in low interference channel conditions.

The plots show that BERs become higher as the code rate increases, where a BER of 10^{-3} is achieved at SNR values of 4.9 dB, 5 dB, 6.8 dB, and 8 dB for the rates of 1/3, 2/5, 2/3, and 3/4 respectively. In terms of the coding gain compared to standard LDPC, the proposed algorithm achieved approximately 1 dB at code rate of 1/3 at $\text{BER}=10^{-2}$, and that gain reduced to approximately 0.5 dB at code rate of 3/4. This demonstrates the narrowing performance gap at higher code rates, reflecting the impact of reduced coding redundancy.

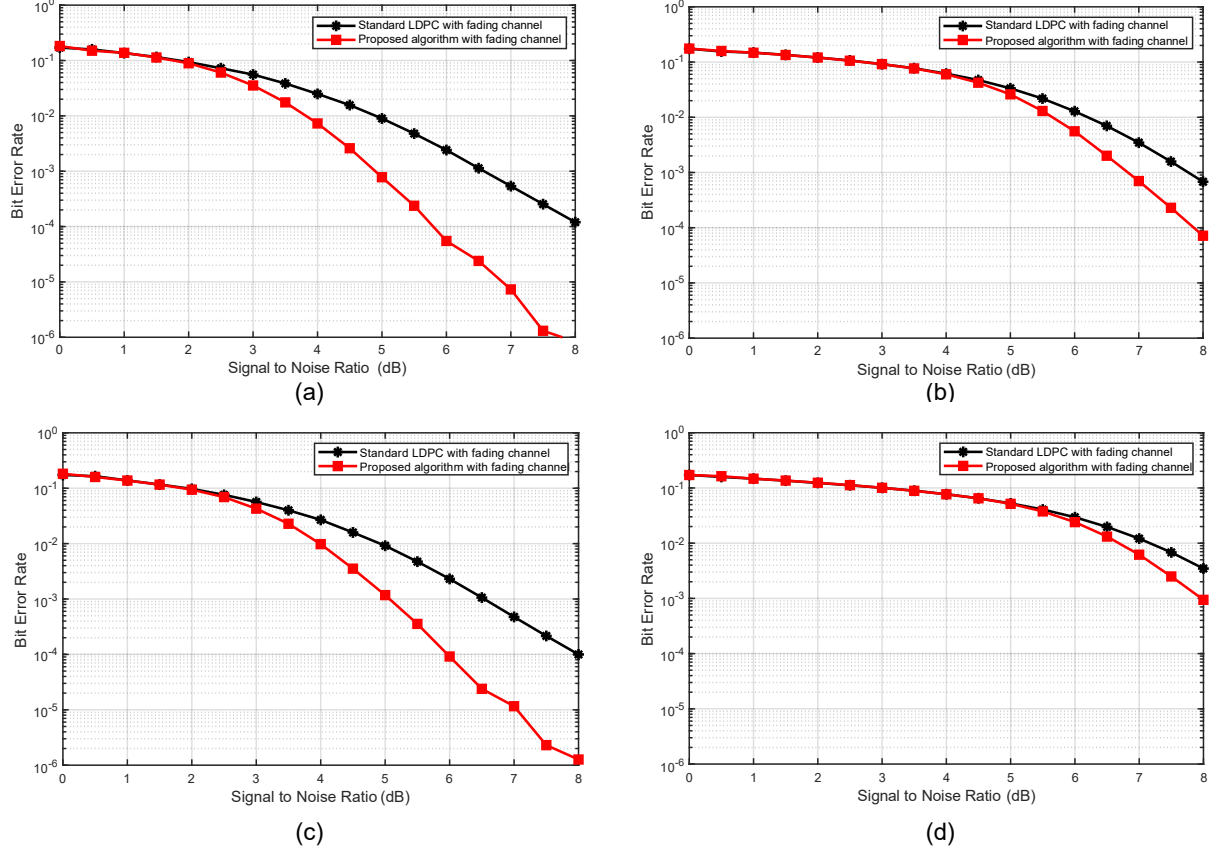


Figure 6: BER against SNR for simulation set 2 with target bits $N = 4$. (a) Code rate = 1/3, (b) code rate = 2/5, (c) code rate = 2/3, and (d) code rate = 3/4.

6. Conclusion

In this paper, a three-stage algorithm based on LDPC was proposed for short-block codes in order to improve the decoding performance of 5G URLLC systems. The algorithm uses low-iteration belief propagation decoding, perturbation-based rescuing, and maximum likelihood based candidate validation stages. The algorithm aims to reduce the risks of decoding trapping sets and miscorrections. The performance of the proposed algorithm was evaluated using MATLAB simulations using AWGN and flat Rayleigh fading channels. The numbers of perturbed bits was varied and different coding rates were tested. For the all considered cases, the proposed decoder demonstrated better decoding accuracy compared to standard LDPC decoder

for SNR values greater than 3 dB. Where lower BERs are achieved at the same SNR value, and demonstrating coding gains of more than 1 dB at low BERs. At the lower SNR values, the performance of the proposed algorithm is comparable to standard LDPC. The increase in the number of perturbed bits generally allows for enhanced performance, however the benefit benefits decrease as the number of perturbed bits become higher. The results suggest that the choice of the number of perturbed bits is a trade-off between the BER performance gains and the computational overhead associated with a greater number of candidate codewords. Future research will study the dynamic selection of the number of perturbed bits, as well as evaluating the algorithm under other channel conditions.

Funding Declaration

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- [1] M. Shirvanimoghaddam et al., “Short Block-Length Codes for Ultra-Reliable Low Latency Communications,” *IEEE Communications Magazine*, vol. 57, no. 2, pp. 130–137, Feb. 2019.
- [2] Y. Polyanskiy, H. V. Poor, and S. Verdú, “Channel Coding Rate in the Finite Blocklength Regime,” 2010.
- [3] Rober G. Gallager, “Low-Density Parity-Check Codes,” *Cambridge, MA: MIT Press*, 1963.
- [4] D. J. C. Mackay and R. M. Neal, “Near Shannon Limit Performance of Low Density Parity Check Codes-To be published in Electronics Letters,” 1996.
- [5] T. J. Richardson and R. L. Urbanke, “The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding,” 2001.
- [6] T. Richardson, “Error Floors of LDPC Codes.”
- [7] H. Lim Meng Kee, N. Ahmad, M. Azri Mohd Izhar, K. Anwar, and S. X. Ng, “A Review on Machine Learning for Channel Coding,” *IEEE Access*, vol. 12, pp. 89002–89025, 2024, doi: 10.1109/ACCESS.2024.3412192.
- [8] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, “Reduced-Complexity Decoding of LDPC Codes,” *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005, doi: 10.1109/TCOMM.2005.852852.
- [9] F. Krieg, M. Rübenacke, A. Zunker, and S. ten Brink, “Long Polar vs. LDPC Codes under Complexity-Constrained Decoding,” May 2025, [Online]. Available: <http://arxiv.org/abs/2508.05485>

- [10] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, May 1995.
- [11] M. Wang, A. Li, and F. Mueller, "Fully Parallelized BP Decoding for Quantum LDPC Codes Can Outperform BP-OSD," May 2026, [Online]. Available: <http://arxiv.org/abs/2507.00254>
- [12] G. Li and X. Yu, "Boosting Ordered Statistics Decoding of Short LDPC Codes with Simple Neural Network Models," May 2025, doi: 10.1109/LCOMM.2024.3475874.
- [13] K. Le, F. Ghaffari, D. Declercq, and B. Vasic, "Efficient Hardware Implementation of Probabilistic Gradient Descent Bit-Flipping," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 4, pp. 906–917, Apr. 2017, doi: 10.1109/TCSI.2016.2633581.
- [14] H. Lee, Y.-S. Kil, M. Jang, S.-H. Kim, O.-S. Park, and G. Park, "Multi-Round Belief Propagation Decoding With Impulsive Perturbation for Short LDPC Codes," *IEEE Wireless Communications Letters*, vol. 9, no. 9, pp. 1491–1494, Sep. 2020, doi: 10.1109/LWC.2020.2994770.
- [15] G. Liva, L. Gaudio, T. Ninacs, and T. Jerkovits, "Code Design for Short Blocks: A Survey," May 2016, [Online]. Available: <http://arxiv.org/abs/1610.00873>
- [16] P. Popovski *et al.*, "Wireless Access for Ultra-Reliable Low-Latency Communication (URLLC): Principles and Building Blocks," Dec. 2017, [Online]. Available: <http://arxiv.org/abs/1708.07862>
- [17] R. Wiesmayr, D. Nonaka, C. Dick, and C. Studer, "Optimizing Puncturing Patterns of 5G NR LDPC Codes for Few-Iteration Decoding," in *2024 58th Asilomar Conference on Signals, Systems, and Computers*, IEEE, Oct. 2024, pp. 788–794. doi: 10.1109/IEEECONF60004.2024.10942813.
- [18] T. Richardson, "Error floors of LDPC codes," in *IEEE Information Theory Workshop (ITW)*, IEEE, 2003.
- [19] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *2nd Int. Symp. Turbo codes and related topics*, 2000, pp. 1–8.