IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 April 2025, accepted 12 May 2025, date of publication 9 June 2025, date of current version 17 June 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3578151

SURVEY

Next-Gen Decoding: Non-Binary LDPC Algorithms for Emerging Power Line and Visible Light Communications

WAHEED ULLAH[1], (Senior Member, IEEE), FENGFAN YANG[2], KWONHUE CHOI[3], (Senior Member, IEEE), AND DUSHANTHA NALIN K. JAYAKODY[4,5,6], (Senior Member, IEEE)

[1]School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg 2050, South Africa
[2]College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[3]School of Computer Science and Engineering, Yeungnam University, Gyeongsan 38541, South Korea
[4]COPELABS, Lusófona University, 1749-024 Lisbon, Portugal
[5]LASI, Center of Technology and Systems (UNINOVA-CTS), 2829-516 Costa de Caparica, Portugal
[6]CIET/EEE, Faculty of Engineering, Sri Lanka Institute of Information Technology, Malabe 10115, Sri Lanka

Corresponding author: Kwonhue Choi (gonew@yu.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by Korean Government through the Ministry of Science and Information and Communication Technology (MSIT) under Grant 2022-0-00024, and in part by the National Research Foundation of Korea (NRF) funded by Korea Government (MSIT) under Grant RS-2024-00452791.

ABSTRACT Non-Binary Low-Density Parity-Check (LDPC) codes have gained significant attention due to their remarkable error correction capabilities in various communication systems. Decoding algorithms play a pivotal role in realizing the potential of non-binary LDPC codes. This paper provides a comprehensive review and analysis of non-binary LDPC decoding algorithms, focusing on their efficiency, complexity, and performance. Furthermore, recent advancements and innovations in non-binary LDPC decoding algorithms are discussed, such as improved message passing strategies, layered decoding techniques, and adaptive algorithms. The review also highlights challenges and open research directions in non-binary LDPC decoding, such as mitigating error floors, reducing decoding complexity, and integrating with emerging communication technologies. Finally, the paper draws conclusions on the current state of non-binary LDPC decoding algorithms, underscoring their promising applications in wireless communication, visible light communication (VLC), and power line communication (PLC). Simulation results demonstrate a marked improvement in bit error rate performance for both VLC and PLC systems, highlighting the practical potential of these advanced decoding techniques.

INDEX TERMS VLC, non-binanry LDPC decoding, decoding algorithms, channel coding, PLC, layered decoding, IoT, message passing algorithm, sum product algorithm.

I. INTRODUCTION

Non-binary Low-Density Parity-Check codes present a promising avenue for optimization [1] in the realm of energy harvesting [2] and energy saving systems. These codes, by virtue of their larger symbol alphabet and sparser parity-check matrices, offer improved error-correcting performance. Although non-binary LDPC codes typically have higher decoding complexity than their binary counterparts, they offer improved error correction performance, making them advantageous in scenarios where reliability is a critical factor.

In energy harvesting applications, where power availability can be intermittent and unpredictable, the robustness of non-binary LDPC codes ensures more reliable data transmission and storage, thereby maximizing the efficiency of the harvested energy. Moreover, their enhanced error correction capabilities translate to fewer retransmissions, which is critical for conserving energy in low-power devices. Among error correction codes, LDPC codes have evolved as powerful tools for ensuring reliable transmission in communication

The associate editor coordinating the review of this manuscript and approving it for publication was Angelo Trotta.

systems. The work by Gallager in 1962 [3] laid the foundation for LDPC codes, which are error-correcting codes with sparse parity-check matrices. Later, Davey and Mackay expanded upon this concept by introducing non-binary LDPC codes [4], where the elements of the parity-check matrix are drawn from a Galois field with order greater than 2 ($GF(q)$, where $q > 2$).

Comparison of binary and non-binary LDPC codes shows that non-binary LDPC codes outperform their binary counterpart, particularly for short length codes. In particular, good LDPC codes over $GF(q)$ are defined by ultra-sparse matrices and thereby easier to decode, and they empirically show better performance around the waterfall region than their binary counterparts, especially for small code-word lengths. Dave and Mackay [4] used Monte Carlo methods to simulate the behavior of the decoding algorithm applied to an infinite LDPC code in an effort to investigate the effects of changes in field order, code construction, and noise level on the decoding performance. These Monte Carlo results have been used to design better codes for practical decoding

Despite the advantageous bit error performance of non-binary LDPC (NB-LDPC) codes, their decoding complexity over $GF(q > 2)$ poses a significant challenge. For the received symbol sequence $\boldsymbol{y} \in GF(q)$, each element necessitates computation, storage in memory, and exchange between variable (bit) nodes and check nodes during decoding. The processing complexity at the check nodes is notably higher compared to binary LDPC codes. A primary drawback of NB-LDPC codes lies in their substantial memory requirements and the complexity of check node processing. At the check nodes, computation involves convolving the messages [5]. This convolution can be represented in the frequency domain by applying fast Fourier transform (FFT) to mitigate computational costs. This adaptation of the belief propagation decoding algorithm, tailored for high-order Galois fields, yields a computational complexity on the order of $q \log_2(q)$.

The performance of LDPC codes, especially under iterative belief propagation algorithms, heavily relies on the distribution of short cycles and the overall structure of the code's Tanner graph [6], [7]. Short cycles play a crucial role, particularly in the error floor phenomenon of low-density parity-check codes, as they contribute to the formation of trapping sets [8]. Consequently, it's valuable to have approximations for the number of cycles of specific lengths in a given parity check matrix graph. Moreover, understanding the distribution of short cycles of various lengths in an ensemble of LDPC codes is also important.

Having knowledge of the cycle distribution in a Tanner graph aids in the analysis and design of LDPC codes. It provides insights into the code's behavior under iterative decoding, helps in identifying potential sources of performance degradation such as trapping sets, and guides efforts to mitigate these issues through code optimization or selection

To ensure the parity check matrix (PCM) of LDPC codes is free of short cycles, a crucial constraint is the Row-Column constraint, which dictates that any two rows or columns of H can share at most one position in common. This constraint guarantees that the Tanner graph of an LDPC code is at least free of four cycles.

Broadly, LDPC codes are constructed using two main approaches:

1. Random-like LDPC Codes: These codes are constructed through computational methods such as progressive edge growth [6] and photograph-based methods [9]. In progressive edge growth, the PCM is incrementally built by adding edges between nodes based on certain criteria, aiming to achieve desired properties such as low error rates and efficient decoding. Photograph-based methods involve using images or visual representations to generate LDPC codes, leveraging properties such as randomness or regularity in the images to construct effective codes.

2. Structured LDPC Codes: These codes are constructed using algebraic techniques, including finite geometries [10], combinatorial structures [11], [12], and finite fields [13], [14]. In this approach, the structure of the LDPC code is determined using mathematical principles, such as properties of finite geometries or combinatorial structures. This allows for the creation of LDPC codes with specific properties or tailored for certain applications. Both random-like and structured LDPC codes have their advantages and are suited for different scenarios. Random-like LDPC codes are often favored for their simplicity and good performance, especially when constructed using advanced computational algorithms. It has been demonstrated that irregular LDPC codes [15], [16] outperform regular LDPC codes.

While randomly constructed regular and irregular LDPC codes demonstrate good performance, their implementation in hardware poses significant challenges due to their high memory requirements and computational complexity. To address this issue, systematically designed LDPC codes called quasi-cyclic (QC) LDPC codes [10], [13], [14] have been introduced. These codes offer performance close to the channel capacity while requiring less memory [17]. QC LDPC codes, represented by $b \times b$ circulant matrices as base matrices, have gained recognition in various communication standards. For instance, they are included in the CCSDS [18] recommendation for deep space and near-earth telemetry applications, where their efficiency and reliability are paramount.

- **Comprehensive Review:** This paper presents the first comprehensive review of non-binary Low-Density Parity-Check codes, analyzing their performance and practical applications in modern communication systems.
- **Focus on Specific Applications:** We specifically examine the application of low complexity non-binary LDPC codes in Visible Light Communication and Power Line Communication systems, highlighting their effectiveness in these environments.

TABLE 1. Non-binary LDPC algorithms.

Message Passing	Majority Logic & Reliability Decoding	Symbol Flipping
QPSA & FFT-QSPA	IHRB & ISRB	Algorithm B and weighted SFD
Min-Sum & Min-Max	Enhanced-IHRB and ISRB	Parallel SFD (PSFD)
Trellis Based Min-Sum	Weighted Bit Reliability Based (wBRB)	Enhanced Serial GBFDA (Improved by MV-SF)
Mixed Domain SPA	Full Bit Reliability	Multiple Voting PSFD,D-SFD, VB MSFD and other variants
Extended-Min-sum	Bit level NB decoding	IJDD and its variants
Other Minor variants	Other Minor variants	Other Minor variants

- **Performance Insights:** The paper demonstrates the superior performance of non-binary LDPC codes, particularly for short-length codes, showing how they can significantly enhance communication reliability and efficiency in VLC and PLC systems.
- **Identification of Advantages:** We identify and articulate the potential advantages of non-binary LDPC codes over binary counterparts, particularly in terms of error correction capabilities and implementation strategies.
- **Guidance for Future Work:** This review provides valuable insights and guidance for researchers and practitioners, suggesting directions for future research and practical implementation strategies in the field of non-binary LDPC codes.

The rest of the paper is organized as follows: Section II provides an overview of the NB-LDPC codes. Section III presents the research background and motivation, while Section IV presents the analysis of the relationships in the NB-LDPC literature. Section V highlights the construction of NB-LDPC codes and Section VI explores various decoding algorithms and their performance trade-offs. Section VII presents an evaluation of the performance of NB-LDPC codes through simulations and Section VIII explains the complexity involved in NB-LDPC decoding. Section IX discusses practical implementations in VLC and PLC systems. Section X covers the future research direction, and finally, Section XI concludes the paper with key contributions.

II. LITERATURE OVERVIEW

Non-binary low-density parity check codes, an extension of binary LDPC codes [4], [19], [20], have demonstrated superior bit error rate (BER) performance. These codes allow codewords to be transmitted as binary sequences, where symbols defined over $GF(q)$ are mapped to binary sequences and transmitted over a binary input channel. Upon reception, the NB-LDPC decoder demaps the received bit vectors back to symbols. However, LDPC codes defined over high-order Galois fields $GF(q > 2)$ face significant computational complexity challenges. To address this, the Fast Fourier Transform (FFT)-based sum product algorithm (FFT-SPA) [5] was introduced for decoding NB-LDPC codes, aiming to reduce the complexity of check node computations in hardware.

The high check node computational complexity of the QSPA decoder is a major obstacle to their finding a place in practical applications. After the invention of the NB-LDPC codes [4], most of the research has focused on how to reduce the computational complexity of the check node processing. To lower the complexity of QSPA, a Fast Fourier transform based SPA (FFT-SPA) algorithm was proposed in [5] to reduce the check node computational complexity from order of $O(q^2)$ to $O(q \log q)$ for each check node update. Other low complexity algorithms called extended min-sum (EMS) [21], [22], trellis based EMS [23], bubble check EMS [24], and min-max algorithms [25] were proposed but they have a performance degradation in comparison to QSPA. Thank you for pointing this out. The Extended Min-Sum (X-EMS) [26], [27] algorithm is an advanced trellis-based decoding method designed to improve the decoding efficiency and error correction performance of LDPC codes. It combines the benefits of the min-sum algorithm with trellis-based approaches to achieve a balance between complexity and performance. Further, the complexity of the computations of all these algorithms is still too high for hardware implementation. On the other hand, reliability based message passing algorithms [28], [29], [30], [31] and the symbol flipping algorithms [32], [33] are simple and computationally very fast for NB-LDPC codes but at a cost of reduced performance.

Reliability based majority logic decoding algorithms offer low complexity as they send only the most reliable field message and in this respect are similar to message passing decoding algorithms. In the iterative majority logic decoding (MLgD) algorithm for non-binary LDPC codes, each symbol is iteratively updated by the extrinsic information-sums (EXIs) with the most reliable field element along each edge of the Tanner graph.

In the bit reliability [34] and symbol flipping NB-LDPC decoding algorithms [35], [36], the codeword is initially transmitted as a binary sequence, and the received binary sequence serves as reliability information for the decoder. The reliability-based majority logic decoding algorithm resembles message passing decoding but transmits only the most reliable field message, making it a computationally efficient algorithm. In the iterative majority logic decoding (MLgD) algorithm for non-binary LDPC codes, each symbol undergoes iterative updates by the extrinsic information sums (EXIs) with the most reliable field element along each edge of the Tanner graph. The iterative reliability-based hard (IHRB) and soft (ISRB) MLgD algorithms [37] exhibit lower complexity but perform optimally only for parity check matrices with high column weights.

Further improvements to the two algorithms in [37] are achieved by introducing soft reliability information at initialization, enhancing the trade-off between complexity and bit error performance through several modifications [28], [31]. A drawback of the IHRB and ISRB algorithms in the literature is their reliance on large column weights. To address this limitation, [38] presents an enhancement to these algorithms by introducing reliability updates in terms of bits rather than symbols. Termed as the weighted bit reliability-based (wBRB) algorithm [34], this bit reliability-based approach requires only integer and Galois field operations, thereby reducing hardware implementation complexity and processing time. To further enhance wBRB decoding algorithms, [34] proposes a full bit-reliability-based decoding scheme, utilizing the binary representation of non-zero entries in the parity check matrix to lower processing complexity.

Symbol flipping decoding algorithms offer low computational complexity but often at the expense of reduced performance. In [39], a majority logic decision-based algorithm was introduced, determining the symbol position to be flipped based on majority decision, while the flipped value is calculated from the channel output by flipping individual bits of the symbol with low reliability. Another approach, termed weighted algorithm B (wt. Algo B) [32], introduces binary Hamming distance and plurality logic for performance enhancement. The parallel symbol flipping decoding (PSFD) algorithm in [33] exhibits good performance only for parity check matrices with large column weights. To improve the PSFD algorithm, a multiple voting-based PSFD (MV-PSFD) algorithm was proposed in [40], and [41]. Furthermore, the symbol-reliability based message-passing (SRBMP) decoding algorithm's performance has been enhanced by the multiple voting symbol flipping (MV-SF) decoding algorithm [42].

Recent advancements include the symbol flipping algorithms: SFD algorithms based on prediction (SFDP) using Hamming Distance (D-SFDP) and SFD algorithms based on prediction (SFDP) using plurality logic (P-SFDP) [43], which offer superior performance compared to most existing symbol flipping algorithms. However, a major drawback is their exhaustive and computationally complex prediction mechanism. The complexity of these algorithms [43] depends on the size of the Galois field order q and the length n of a codeword. Additionally, the decision-symbol reliability-based SFD (DRB-SFD) algorithm [44] aims to demonstrate improved performance in applications such as data storage based on NAND flash memory. However, the study in [44] focuses solely on the algorithm for NAND flash memory without comparing bit error rate (BER) performance with existing algorithms like MV-SF, MV-PSFD, and D-SFDP.

The recent developments in research papers, namely [35] and [36], focus on advancements in symbol flipping decoding algorithms for LDPC codes. These papers introduce novel approaches including low-complexity, high-performance reliability, and voting-based symbol flipping decoding algorithms. The primary objective of these advancements is to streamline the flipping function's complexity while concurrently improving the bit error performance

Non-binary LDPC codes can also be transmitted using direct mapping to the high order modulation. The advantage of NB-LDPC codes using high-order q-ary modulations is the direct encoding and decoding over the q-ary constellation as the binary to non-binary mapping and de-mapping operations are not required. Also the non-binary symbol likelihoods are computed directly without any conversion. The mapping and de-mapping operations are cost effective in terms of complexity and introduce performance degradation that would have to be partially countered by a good choice of mapping and de-mapping at decoder. The NB-LDPC sum-product and its variants can be used to map to the higher order modulation to achieve high data rates. The binary Min-Sum decoding algorithm [45], [46] extended to the non-binary decoding, is known as Extended Min-Sum (EMS) algorithm [22], [23], [47], and it gives better trade-off between hardware complexity and bit error rate performance.

The complexity of the Extended Min-Sum decoding largely stems from the computation at the check node (CN) level. To mitigate this, a Forward-Backward scheme [24], [25] was proposed, involving serial computation in hardware with intermediate results. However, this approach introduces significant latency and reduces throughput, especially with increased finite field sizes in $GF(q)$. Addressing these challenges, a hardware-aware algorithm called syndrome-based EMS algorithm was introduced [48], improving both bit error rate performance and throughput through enhanced parallelism.

Additionally, an M-QAM based hard decision NB-LDPC algorithm was proposed [49], [50], known as the iterative joint detection-decoding (IJDD) algorithm. While this method employs M-QAM modulation for higher throughput, its performance is inferior compared to the q-ary LDPC (QSPA). The algorithm uses hard decision symbol sequences as feedback in each iteration to the M-QAM detector, aiming to steer received messages in the correct direction and mitigate noise effects.

In the recently proposed algorithms [51], [52], if the predicted symbol value meets the checksum criteria, it is considered accurate; otherwise, it undergoes adjustment before being returned to the QAM detector. Algorithm 2, detailed in these papers, represents an advancement to the iterative joint detection-decoding algorithm. It achieves this by employing iterative hard decision-based majority logic to determine the new candidate symbol value. The feedback signal to the QAM detector is fine-tuned based on the Euclidean distance between the current symbol and the newly selected symbol value. As for Algorithm 3, it offers a simplified version of Algorithm 2, achieved through the implementation of a majority voting scheme.

To enhance convergence speed, a layered decoding approach is employed for both binary and NB-LDPC codes. This method utilizes two strategies for layer decoding: row-layered and column-layered. The primary distinction lies in the layer type, although they are theoretically equivalent and offer superior decoding performance. However, due to data dependencies, the initiation of the next layer is contingent upon the completion of the current layer. The column-layered decoding approach, also known as Shuffled decoding, is described in [53], and [54]. In existing literature, the Group-Shuffled message passing decoding algorithms, as outlined in [55], segment either check nodes or variable nodes of the relevant bipartite graph into smaller sub-groups termed layers. Moreover, each primary iteration undergoes subdivision into multiple sub-iterations. These groups may encompass one or more rows and columns. Some methodologies horizontally divide a quasi-cyclic parity check matrix into groups, ensuring each group contains precisely one non-zero entry in every column. This method is prevalent for implementing sum-product LDPC decoder hardware.

In the paper by Ullah et al. [56], [57], two decoding techniques for symbol flipping non-binary LDPC codes are introduced, both based on grouping strategies. Initially, a novel approach is presented, involving the adaptive grouping of variable nodes in each iteration, utilizing both bit reliability and majority voting of each received symbol. Subsequently, a fixed grouping technique is applied to the symbol flipping decoding of non-binary LDPC codes. In this fixed grouping method, each group consists of an equal number of variable nodes, and selected symbols within each group are flipped according to predetermined flipping criteria. These algorithms exhibit superior performance compared to existing methods.

III. RESEARCH BACKGROUND AND MOTIVATION

The background of Non-Binary Low-Density Parity-Check codes encompasses a journey of innovation and adaptation, stemming from the rich history of LDPC codes and the need for enhanced error correction capabilities in modern communication systems.

Message Passing (MP) is effective for large, sparse graphs and is scalable but can be computationally expensive and may not converge for certain structures. Majority Logic Decoding is simple and robust against noise but can degrade with high error rates. Symbol Flipping is efficient for moderate error rates but less effective for high error rates and may require multiple iterations. QPSA reduces computational complexity but may lose some accuracy due to quantization, while FFT-QPSA enhances efficiency using FFT but requires more complex implementation. Min-Sum is simpler and faster than sum-product, making it suitable for hardware, but can be less accurate. Min-Max balances between min and max values but is more complex than Min-Sum. Trellis Based Min-Sum combines trellis structures with Min-Sum for efficient decoding but is more complex. Extended-Min-Sum improves performance over Min-Sum with more accurate approximations but increases computational complexity. IHRB uses reliability for decision making and is effective in iterative settings but may require many iterations, and ISRB offers more accuracy with soft decisions but has higher complexity. Enhanced-IHRB and ISRB improve performance and accuracy but are more complex. wBRB uses weighted reliability for accurate decoding but involves complex weight computation. Full Bit Reliability improves decoding performance with comprehensive assessment but is highly complex. Bit level NB decoding enhances performance for non-binary codes but is more complex. Alg B (wtd Alg B) uses weighted values for flexibility but can be complex. Weighted SFD improves accuracy with weighted symbols but requires careful tuning. PSFD is faster with parallel implementation but needs more resources. Enhanced Serial GBFDA with multiple vote symbol flipping improves performance but increases complexity. Multiple Vote PSFD combines parallel processing with multiple voting for reliability but is computationally complex. SFD-Prediction (D-SFDP and P-SFDP) incorporates prediction for improved performance but adds complexity and may need more iterations.

1. Evolution from Binary LDPC Codes: The foundation of NB-LDPC codes can be traced back to the development of binary LDPC codes by Robert Gallager in the early 1960s [3]. Binary LDPC codes demonstrated remarkable error correction performance approaching the Shannon limit, attracting significant attention from the communication engineering community. However, as communication systems evolved and demanded higher performance, the limitations of binary LDPC codes became apparent, particularly in scenarios with challenging channel conditions.

2. Need for Non-Binary Representation: Binary LDPC codes operate over binary finite fields, limiting their capacity to represent information efficiently. In contrast, non-binary LDPC codes, including NB-LDPC codes, operate over finite fields with sizes greater than two. This non-binary representation offers advantages in terms of error correction capabilities and flexibility. It allows for more efficient encoding and decoding processes, enabling better performance in scenarios with complex noise characteristics [58].

3. Advantages of Non-Binary LDPC Codes: NB-LDPC codes leverage the inherent benefits of non-binary representation to achieve superior error correction performance compared to their binary counterparts [59]. By operating over larger finite fields, NB-LDPC codes can represent information more compactly and effectively, leading to improved reliability in communication systems. Moreover, non-binary LDPC codes exhibit enhanced tolerance to various types of noise and interference, making them well-suited for a wide range of applications.

4. Innovative Construction Techniques: The development of NB-LDPC codes has been accompanied by innovative construction techniques tailored to exploit the advantages of non-binary representation [60]. Techniques such as

progressive edge growth (PEG) algorithms, Quasi-Cyclic constructions, and algebraic geometric (AG) codes have been employed to design efficient NB-LDPC codes with desirable properties. These construction methods enable the creation of LDPC codes optimized for specific communication scenarios and channel characteristics.

5. Research and Standardization Efforts: The research community, supported by organizations like the IEEE, has been actively involved in advancing the theory and practice of NB-LDPC codes. Numerous research papers, conferences, and standardization efforts have contributed to the understanding and dissemination of NB-LDPC technology. As a result, NB-LDPC codes have found application in diverse domains, including wireless communication, optical communication, and storage systems [61].

In the context of 5G and the upcoming 6G technologies, research in communication systems has been increasingly focused on addressing the unique challenges and requirements posed by these advanced wireless networks. These challenges include the need for ultra-reliable low-latency communication (URLLC), massive connectivity for Internet of Things (IoT) devices, and support for high data rates and bandwidth efficiency.

In this context, LDPC codes have garnered significant attention as an essential component for achieving reliable and efficient communication in 5G and 6G networks. LDPC codes offer several advantages, including their ability to achieve near-Shannon capacity performance, their low-complexity decoding algorithms, and their suitability for hardware implementation.

IV. ANALYSIS OF RELATIONSHIPS IN NB-LDPC LITERATURE

The research on NB-LDPC codes encompasses a rich variety of algorithms and construction techniques. However, a comprehensive analysis of these approaches reveals critical interdependencies and trends.

A. ALGORITHMIC ADVANCEMENTS

Early efforts focused on decoding algorithms like the q-ary Sum-Product Algorithm, which provided high accuracy at the cost of computational complexity. This led to the development of more efficient alternatives, such as the Extended Min-Sum and Symbol Flipping algorithms. These advancements represent a trade-off between performance and complexity, catering to specific application requirements.

B. SYNERGY BETWEEN CONSTRUCTION METHODS AND DECODING

The choice of construction method plays a pivotal role in determining the efficiency of decoding algorithms. For instance:

- **Finite Field-Based Constructions:** Suitable for burst-error correction and high-noise scenarios.
- **Protograph-Based Designs:** Offer structured codes with high girth, enhancing iterative decoding performance.
- **Array-Based Constructions:** Optimize hardware implementations due to their regularity and reduced memory requirements.

C. APPLICATION-SPECIFIC INNOVATIONS

The evolution of NB-LDPC codes has been significantly driven by their practical applications. In VLC, low-latency decoding is critical, whereas PLC demands robust performance under interference. These requirements have shaped innovations in both code construction and decoding strategies, highlighting the synergy between theoretical research and real-world demands.

D. RESEARCH GAPS AND FUTURE DIRECTIONS

Despite considerable progress, several challenges remain:

- **Hardware Efficiency:** Further efforts are needed to develop lightweight decoding algorithms for NB-LDPC codes.
- **Cross-Layer Optimization:** Integration of NB-LDPC codes with other advanced technologies, such as RIS and massive MIMO [62], requires holistic optimization across layers.
- **Scalability to 6G Networks:** The unique requirements of 6G, including extreme reliability and ultra-low latency, present an opportunity for NB-LDPC codes to address these demands comprehensively.

This analysis underscores the interconnected nature of research in NB-LDPC codes and highlights pathways for advancing their theoretical and practical utility.

V. CONSTRUCTION OF NONBINARY LDPC CODES

Nonbinary Low-Density Parity-Check (LDPC) codes are an extension of binary LDPC codes, where the code symbols belong to a Galois field $GF(q)$ with $q > 2$. The construction of nonbinary LDPC codes plays a crucial role in achieving improved performance for various communication and storage applications. This section introduces three prominent construction methods: finite field-based, protograph-based, and array-based nonbinary LDPC codes.

A. FINITE FIELD-BASED NONBINARY LDPC CODES

Finite field-based constructions [63] rely on the properties of $GF(q)$ to define the elements of the parity-check matrix H. Each entry in H is either zero or a nonzero element from $GF(q)$, allowing for greater flexibility in code design. These constructions are beneficial for achieving high error correction performance in the presence of burst errors. The utilization of finite fields enables the design of codes with structured properties, which facilitate efficient encoding and decoding.

B. PROTOGRAPH-BASED NONBINARY LDPC CODES

Protograph-based constructions [64], [65] utilize a small base graph, known as a protograph, which is replicated and interconnected to form a larger Tanner graph. The protograph

structure allows for flexible and scalable code designs while retaining desirable properties such as high girth and low encoding complexity. Nonbinary LDPC codes constructed using protographs have been shown to exhibit excellent performance in iterative decoding due to their structured nature.

C. ARRAY-BASED NONBINARY LDPC CODES

Array-based constructions [66], [67] are characterized by their systematic arrangement of symbols in the parity-check matrix, often using cyclic or quasi-cyclic structures. These codes are highly efficient for hardware implementations due to their regularity and reduced storage requirements. Array-based nonbinary LDPC codes are particularly advantageous in applications where low-complexity decoding and high throughput are critical.

D. COMPARISON OF CONSTRUCTION METHODS

Each of the discussed methods offers unique advantages. Finite field-based constructions excel in burst error correction, protograph-based designs provide scalability and structure, and array-based codes are ideal for hardware-friendly implementations. The choice of method depends on the specific application and system requirements.

VI. NB-LDPC DECODING ALGORITHMS

A. PRELIMINARIES

An LDPC code which is fully represented by a Tanner graph, also aide in the description of decoding algorithm. A tanner graph has two types of nodes– variable (bit) node (VN) and check node (CN). Let $GF(q)$ be the Galois field with q elements. Consider a NB-LDPC code $\mathcal{C}$ of length n with a regular parity check matrix H, defined over the $GF(q)$, with m as number of rows and n as number of columns such that each row of H has a constant weight of d_c and each column of H has a constant weight of d_v. Each element $h_{i,j}(1 \leq i \leq m,\ 1 \leq j \leq n)$ of H is an element over the Galois field $GF(q)$ where $q = 2^r$. The non-zero entries of $H(h_{i,j} \neq 0)$ in the Tanner graph show the i^{th} check node connected to the j^{th} variable node. For the column weight d_v and the row weight d_c, the Tanner graph edge connections are shown as $M(j) = \{i : 1 \leq i \leq m, h_{i,j} \neq 0\}$ and $N(i) = \{j : 1 \leq j \leq n, h_{i,j} \neq 0\}$ and for regular matrix $\frac{m}{n} = \frac{d_v}{d_c}$.

Let $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_j, \ldots, \boldsymbol{c}_n)$ be a codeword in $\mathcal{C}$ and the binary representation of j^{th} symbol in $\boldsymbol{c}$ is $\boldsymbol{c}_j = (c_{j,1}, c_{j,2}, \ldots, c_{j,t}, \ldots, c_{j,r}), 1 \leq t \leq r$. This binary sequence is modulated with binary phase shift keying (BPSK), where 1 is modulated as $+1$ and 0 is modulated as -1. The BPSK modulated j^{th} symbol $\boldsymbol{x}_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,t}, \ldots, x_{j,r})$ is transmitted over additive white Gaussian noise (AWGN) channel. The received binary sequence $\boldsymbol{y}_j = (y_{j,1}, y_{j,2}, \ldots, y_{j,t}, \ldots, y_{j,n})$ is obtained from the transmitted sequence $\boldsymbol{x}_j$ after adding the additive white Gaussian channel noise $\boldsymbol{n} \rightarrow \mathcal{N}(0, \sigma^2)$ with zero mean and two sided power spectral density $N_0/2$ as $\boldsymbol{y}_j = \boldsymbol{x}_j + \boldsymbol{n}_j$. The hard decision binary sequence $\boldsymbol{z}_j = (z_{j,1}, z_{j,2}, \ldots, z_{j,t}, \ldots, z_{j,r})$ for the j^{th} received symbol $\boldsymbol{y}_j$ is given by:

$$z_{j,t} = \begin{cases} 1 & \text{if } y_{j,t} \geq 0 \\ 0 & \text{else} \end{cases} \tag{1}$$

The hard decision binary sequence $z_{j,t}$ is mapped to an element in $GF(q)$ to obtain the symbol $\boldsymbol{z}_j$. Let $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_i, \ldots, \boldsymbol{s}_m) = H\boldsymbol{z}^T$ is the syndrome vector, then the i^{th} syndrome of the j^{th} hard decision symbol sequence can be defined for $h_{i,j} \neq 0$:

$$\boldsymbol{s}_i = \sum_{j \in N(i)} h_{i,j} \boldsymbol{z}_j \tag{2}$$

Here it is important to mention that the value of $\boldsymbol{s}_i$ is also an element in $GF(q)$. If $H\boldsymbol{z}^T \neq \boldsymbol{0}_{(1,n-k)}$, which means that some of the variable nodes connected to check node are in error.

Let $GF(q) = (\alpha_1, \alpha_2, \ldots, \alpha_l, \ldots, \alpha_q)$, then for the j^{th} received symbol $\boldsymbol{z}_j$, the priori probabilities $(\gamma_j^{\alpha_1}, \gamma_j^{\alpha_2}, \ldots, \gamma_j^{\alpha_l}, \ldots, \gamma_j^{\alpha_q})$ are equal to $\{\alpha_1, \alpha_2, \ldots, \alpha_l, \ldots, \alpha_q\}$ respectively such as $(\gamma_j^{\alpha_1}, \gamma_j^{\alpha_2}, \ldots, \gamma_j^{\alpha_l}, \ldots, \gamma_j^{\alpha_q}) = 1$ holds. The priori probability for the symbol $\alpha_l \in GF(q)$ is given below.

For $1 \leq j \leq n, 1 \leq l \leq q, 1 \leq t \leq r$, the probability of the received binary sequence $z_{j,t}$ to be 1 for the channel out put $y_{j,t}$ is:

$$\gamma_{j,t}^1 = \frac{1}{1 + \exp(\frac{4y_{j,t}}{N_0})} \tag{3}$$

and the probability for the received binary sequence $z_{j,t}$ to be 0 is $\gamma_{j,t}^0 = 1 - \gamma_{j,t}^1$. Now the probability of $\gamma_{j,t}^{\alpha_l}$ being the element $\alpha_l \in GF(q)$ for the received symbol $\boldsymbol{z}_j$ corresponding to the channel output $\boldsymbol{y}_j$, is given by:

$$\gamma_j^{\alpha_l} = \prod_{t=1}^{r} \gamma_{j,t}^{\alpha_{l,t}}. \tag{4}$$

This probability of the j^{th} symbol is then used as priori information at the initialization of the QSPA. For an LDPC code defined over $GF(q = 2^r)$ where each received symbol α_l must be one of the symbols in $GF(q)$, the check node processing becomes more complicated as there are more possible values of the q to satisfy the parity check equations. For each element defined over $GF(q)$ of the parity check matrix $H(h_{i,j} \neq 0)$, there are q probabilities associated with it as compared to its binary counterpart which has only two probabilities for 0 and 1. Therefore, q messages need to be computed, stored in the memory, and exchanged between the check and variable nodes during this iterative process.

The major bottleneck of the sum-product NB-LDPC decoding algorithm is the check node computational complexity and the large memory required to store the q-tuple

messages. The posteriori probability $\gamma_{i,j}^{\alpha_l}$ (message sent to check node from variable node) is given by:

$$\hat{\gamma}_j^{\alpha_l} = \sum_{\boldsymbol{z}_j \in GF(q)} Pr(\boldsymbol{s}_i = 0 | \boldsymbol{z}_j = \alpha_l) \prod_{j' \in N(i) \backslash j} q_{j',t}^{\alpha_l} \tag{5}$$

The computed values of $\gamma_{i,j}^{\alpha_l}$ for $\alpha_l \in GF(q)$ for $i \in M(j), j \in N(i)$ are then used to update the extrinsic information $q_{i,j}^{\alpha_l}$ to be used in the next iteration and is given as follows:

$$q_{i,j}^{\alpha_l} = \lambda \gamma_j^{\alpha_l} \prod_{i' \in M(j) \backslash i} \hat{\gamma}_{i',j}^{\alpha_l} \tag{6}$$

where the scaling factor λ is chosen such that the sum of all probabilities equal to 1, i.e. $q_{i,j}^{\alpha_1} + q_{i,j}^{\alpha_2} + \ldots + q_{i,j}^{\alpha_q} = 1$. The above mentioned equations serve as the check and variable node processing rule for message passing decoder.

B. DECODING NB-LDPC CODES WITH SPA

The NB-LDPC sum-product decoding algorithm [4], [68] is summarized as follows:

Algorithm 1 Sum Product NB-LDPC Algorithm

1: **Initialization**:
2: For $1 \leq j \leq n, 1 \leq m \leq, 1 \leq l \leq q$:
3: Received symbol sequence: $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_j, \ldots, \boldsymbol{y}_n)$
4: Hard decision symbol sequence: $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_j, \ldots, \boldsymbol{z}_n)$
5: Priori probabilities: $\gamma_j = Pr(\boldsymbol{z}_j = \alpha_l | \boldsymbol{y}_j)$
6: Extrinsic probabilities: $q_{i,j}^{\alpha_l} = \gamma_j^{\alpha_l}$
7: Set $k = 1$ to I_{max}
8: **repeat**
9: Compute $\boldsymbol{s}^k = \boldsymbol{z}^k H^T$
10: **if** $\boldsymbol{s}^k = \boldsymbol{0}$ or $k = I_{max}$ **then**
11: Stop decoding
12: **end if**
13: Compute $\hat{\gamma}_{i,j}^{\alpha_l,k}$ using equation (5)
14: Compute $q_{i,j}^{\alpha_l,k}$ using equation (6)
15: Estimate the symbol sequence as $\boldsymbol{z}^k = \arg\max_{\alpha_l} \hat{\gamma}_{i,j}^{\alpha_l,k}$
16: $k = k + 1$
17: **until** convergence

Each j^{th} symbol of the received sequence $\boldsymbol{y}_j$ has associated likelihood in the order of $GF(q)$, as the order of $GF(q)$ increases the complexity increase in order of $O(q^2)$.

Writing the field elements of $GF(q)$ as power of primitive element α, $GF(q) = (\alpha^{-\infty}, \alpha^0, \alpha^1, \ldots, \alpha^{q-2})$ where α is the primitive element. When two field elements of $GF(q)$ are multiplied, it is equivalent to shift the powers of the primitive elements cyclically and then get the resultant field element. This cyclic shifts are known as permutation.

C. DECODING NB-LDPC CODES WITH FFT-QSPA

The main complexity of the q-ary sum-product algorithm (QSPA) lies in the check node processing. The check node computation is like convolution which can be implemented more easily in the frequency domain as term by term multiplication. In the frequency domain decoding [5], check node message at the k^{th} iteration is computed as:

$$\gamma_{i,j}^k = F^{-1}\left(\prod_{j' \in N(i) \backslash j} F(q_{i,j}^k) \right) \tag{7}$$

here F and F^{-1} are the Fourier and the inverse Fourier transform and $\prod$ is the term by term multiplication. When the probability mass function is defined over the $GF(q)$, then the Fourier transform is the two point, r-dimensional message vector. Both the Fourier and inverse Fourier transform are 2×2 matrices given by:

$$F = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{and} \quad F^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{8}$$

The FFT-QSPA decoding algorithm presented by Davey and Mackay, can be used over the Galois field $GF(q)$ where q is power of 2. Though it can be generalized to the prime power of q as presented in [5] but since q is commonly used as power of 2 for practical applications, therefore, the generalized FFT-QSPA is of not much consideration. Equation (8) is known as binary Fourier transform or the Walsh Hadamard transform. The Fast Fourier Transform based decoding reduces complexity by $O(q \log q)$.

D. LOG-DOMAIN SUM-PRODUCT NB-LDPC ALGORITHM

A simplified version of the QSPA decoding called LogSPA is presented in [69] which uses additions, subtractions, and look-up tables only. The LogSPA decoder requires a fairly accurate knowledge of the signal-to-noise ratio (SNR) at the input of the decoder but less sensitive to quantization noise. The computational complexity for this LogSPA scales as $O(q^2)$. Denoting the $GF_0(q) = (\alpha_2, \alpha_3, \ldots, \alpha_l, \ldots, \alpha_q)$ without the zero element, then the log likelihood ration (LLR) of nonzero elements $\alpha \in GF(q)$ is written as:

$$L(\alpha_j) = \{L(\alpha_j = \alpha_2), L(\alpha_j = \alpha_3), \ldots, L(\alpha_j = \alpha_l), \ldots, L(\alpha_j = \alpha_q)\} \tag{9}$$

$$L(\alpha_j = \alpha_l) = \ln \frac{Pr(\alpha_j = \alpha_l)}{Pr(\alpha_j = 0)} \tag{10}$$

The priority probability of the j^{th} received symbol is given by:

$$\gamma_j = \frac{2}{\sigma^2} \sum_{j \in N(i)} \boldsymbol{y}_{j,t} \tag{11}$$

and the posteriori information is updated as follows:

$$\hat{\gamma}_j^k = \gamma_j + \sum_{j \in N(i) \backslash j'} q_{i,j'}^k \tag{12}$$

The check node message update for the next iteration is computed as follows:

$$q_{i,j}^{k+1} = \gamma_j + \sum_{i \in M(j) \backslash i'} q_{i',j}^k \tag{13}$$

Algorithm 2 NB-LDPC LogSPA Decoding Algorithm

Initialization:
For $1 \leq j \leq n, 1 \leq m \leq, 1 \leq l \leq q$:
  Received symbol sequence: $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_j, \ldots, \boldsymbol{y}_n)$
  Hard decision symbol sequence: $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_j, \ldots, \boldsymbol{z}_n)$
  Priori probabilities: $\gamma_j = \frac{2}{\sigma^2} \sum_{j \in N(i)} \boldsymbol{y}_{j,t}$
  Extrinsic probabilities: $q_{i,j} = \gamma_j$
Set $k = 1$ to I_{max}
repeat
  Compute $\boldsymbol{s}^k = \boldsymbol{z}^k H^T$
  if $\boldsymbol{s}^k = \boldsymbol{0}$ or $k = I_{max}$ **then**
    Stop decoding
  end if
  Compute $\hat{\gamma}_j^k$ using equation (14)
  Update the check node message $q_{i,j}^{k+1}$ using equation (15)
  Estimate the symbol sequence as $\boldsymbol{z}^k = \arg\max_{\alpha_l} \hat{\gamma}_{i,j}^{\alpha_l,k}$
  $k = k + 1$
until convergence

E. MIXED DOMAIN SPA

The equations involved in the Mixed Domain SPA are as follows:

$$\hat{\gamma}_j^k = \gamma_j + \sum_{j \in N(i) \backslash j'} q_{i,j'}^k \tag{14}$$

$$q_{i,j}^{k+1} = \gamma_j + \sum_{i \in M(j) \backslash i'} q_{i',j}^k \tag{15}$$

These equations govern the update of posteriori information and check node messages in the Mixed Domain SPA for NB-LDPC decoding.

1) BENEFITS OF MIXED DOMAIN SPA

1.Efficient Fourier Transform in Probability Domain: Performing the Fourier transform in the probability domain allows for efficient computation of convolution operations, which are essential in many signal processing and communication applications. By leveraging the properties of probability distributions, such as the convolution theorem, FFT-based algorithms can efficiently compute convolutions, leading to faster processing times compared to traditional methods.

2. Log-Domain for Numerical Stability: Logarithmic domain processing is known for its numerical stability, especially when dealing with small probabilities or likelihoods. By operating in the log-domain, the algorithm can handle a wider range of values without encountering numerical precision issues that might arise in linear domain calculations. This is particularly advantageous in iterative algorithms like the Sum-Product Algorithm (SPA), where small probabilities can be encountered during message passing.

Algorithm 3 Mixed Domain SPA for NB-LDPC Decoding

Initialization:
for all $1 \leq j \leq n$ **do**
  Compute the received symbol sequence: $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)$
  Compute the hard decision symbol sequence: $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n)$
  Compute the priori probabilities: $\gamma_j = \frac{2}{\sigma^2} \sum_{j \in N(i)} \boldsymbol{y}_{j,t}$
  Initialize the extrinsic probabilities: $q_{i,j} = \gamma_j$
end for
Iteration:
repeat
  Compute the symbol sequence $\boldsymbol{s}^k = \boldsymbol{z}^k H^T$
  if $\boldsymbol{s}^k = \boldsymbol{0}$ or the maximum number of iterations is reached **then**
    Stop decoding
  end if
  for all variable nodes j **do**
    Compute $\hat{\gamma}_j^k$ using Equation (14)
  end for
  for all check nodes i **do**
    Update the check node message $q_{i,j}^{k+1}$ using Equation (15)
  end for
  Estimate the symbol sequence $\boldsymbol{z}^k$ using MAP estimation based on $\hat{\gamma}_j^k$
  Increment k
until convergence

Algorithm 4 Algorithm 2: Trellis-Based Decoding Scheme

Input: $\boldsymbol{y}'_{i,j}(a)$, the v2c hard messages.
Output: $C'_{i,j}(a)$, the computed c2v messages.
for $j \in N(i)$ **do**
  $b_j \leftarrow \text{argmin}_{a \in \text{GF}(q)} \boldsymbol{y}'_{i,j}(a)$
end for
for $j \in N(i)$ **do**
  Compute the c2v hard messages: $\beta_j \leftarrow \sum_{j' \in N(i)/j} (b_j \oplus j')$
end for
for $j \in N(i)$ **do**
  Convert from normal domain to delta domain: $\boldsymbol{y}'_{i,j}(a \oplus b_j) \leftarrow \boldsymbol{y}'_{i,j}(a)$
end for
Computing c2v messages:
$C'_{i,j}(a) \leftarrow \delta(1\boldsymbol{y}'_{i,j}(a))$
Delta domain to normal domain:
$C'_{i,j}(a) \leftarrow \delta(C'_{i,j}(a \oplus \beta_j))$

3. Reduced Complexity: Combining the FFT-based processing in the probability domain with log-domain check node processing can lead to reduced computational complexity. The FFT algorithm has a complexity of O(n log n), where n is the size of the input data, making it efficient

for large-scale problems. Meanwhile, log-domain operations typically involve simple additions and subtractions, which further contribute to computational efficiency.

4. Improved Performance in Iterative Decoding: In applications such as error correction decoding in communication systems, iterative algorithms like the Sum-Product Algorithm (SPA) are widely used. By employing FFT-LogSPA, which combines FFT-based processing with log-domain computations, the decoding performance can be improved. This improvement arises from both the efficiency gains and the numerical stability provided by the mixed domain approach, leading to better convergence properties and potentially higher error correction capabilities.

5. Flexibility and Adaptability: The mixed domain approach provides flexibility to adapt the algorithm to different applications and scenarios. For instance, the choice of probability distributions in the Fourier transform stage can be tailored to the specific characteristics of the problem at hand. Similarly, the use of the log-domain allows for easy incorporation of additional constraints or modifications to the algorithm, enhancing its adaptability to diverse requirements.

2) DRAWBACKS OF MIXED DOMAIN

The main drawback is that it needs conversion between the two domains and also lookup tables. In the paper [70] a low complexity simplified soft distance (SSD) decoding algorithm for NB-LDPC codes is presented, which is an adaption for NB-LDPC codes of a previously reported SSD algorithm for binary LDPC codes [71], [72], but now operates in the transform domain, called the Fast Fourier Transform Simplified Soft Distance (FFT-SSD) decoding algorithm. This decoding algorithm uses squared Euclidean distance as the metric, which offers as a great advantage that it does not require knowledge of the signal-to-noise ratio of the received signal, and requires only additions, subtractions and look-up tables. Large lookup tables and conversion between the two domains are the main drawback of this algorithm.

F. MIN-SUM NB-LDPC DECODING ALGORITHMS

The Min-Sum Nonbinary LDPC decoding algorithm aims to iteratively improve the reliability of decoded symbols by exchanging information between variable nodes and check nodes in the LDPC code graph. It is known for its simplicity and effectiveness, especially for nonbinary LDPC codes where symbol alphabets have more than two elements.

Algorithm 5 Min-Sum Nonbinary LDPC Decoding Algorithm

Initialization:
Initialize messages $q_{i,j}$ from variable nodes to check nodes using received symbols and reliability weights: $q_{i,j} = \gamma_j$
Initialize reliability weights for received symbols
Iteration:
Compute messages from variable nodes to check nodes:
$q^{(l)}_{j\to i} \leftarrow$ Compute $q^{(l)}_{j\to i}$ for all j and i
Compute messages from check nodes to variable nodes:
$q^{(l)}_{i\to j} \leftarrow$ Compute $q^{(l)}_{i\to j}$ for all i and j
Decision:
Make hard decision on each symbol: $\boldsymbol{z}^k = \text{sign}(\boldsymbol{z}^k)$

G. ITERATIVE RELIABILITY BASED NB-LDPC ALGORITHMS

The concept of hard reliability to improve the performance of the bit reliability based (BRB) decoding algorithm was introduced in [38]. The binary Hamming distance between hard decision symbol sequence $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_j, \ldots, \boldsymbol{z}_n)$ and the extrinsic information-sums (EXIs) indicate the hard reliability of the EXIs. The author in [38] showed that the Hamming distance $d(\boldsymbol{z}_j, \sigma_{i,j})$ indicates the correct probability of extrinsic information-sum $\sigma_{i,j}$. If p_b is raw bit error probability of variable nodes (VNs) over the Galois field GF($q = 2^r$), then the symbol error probability of the individual variable node is $1-(1-p_b)^r$. The relationship between the EXI and Hamming distance [38] is written as:

$$Pr(\sigma_{i,j}|d(\boldsymbol{z}_j, \sigma_{i,j}) = u) = \frac{p_b^u q}{p_b^u q - (1-p_b)^{u-r} + (1-p_b)^{-d_c r + r + u}} \tag{16}$$

for $1 \le u \le r$. To include the reliability derived from the structure of the Galois field, a vector of extrinsic weighting coefficients are defined as $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_k, \ldots, \theta_r]$. Here θ_k shows the extrinsic weighting factor corresponding to $d(\boldsymbol{z}_j^{(k)}, \sigma_{i,j}^{(k)}) = k$ for $0 \le k \le r$. According to equation (16), $d_{\theta(\boldsymbol{z}_j^{(k)}, \sigma_{i,j}^{(k)})} \ge 2$ does not carry much useful information and it has very low probability to be corrected [38]. According to this, the weighting coefficients are distributed as $\theta = [\theta_0, \theta_1, \theta_2]$ for the extrinsic information-sum and are optimized through simulation.

Algorithm 6 Weighted Bit-Reliability-Based (wBRB) Nonbinary LDPC Decoding Algorithm

Initialization:
Initialize messages $q_{i,j}$ from variable nodes to check nodes using received symbols and reliability weights: $q_{i,j} = \gamma_j$
Initialize reliability weights for received symbols
Iteration:
Compute messages from variable nodes to check nodes:
$q^{(l)}_{j\to i} \leftarrow$ Compute $q^{(l)}_{j\to i}$ for all j and i
Compute messages from check nodes to variable nodes:
$q^{(l)}_{i\to j} \leftarrow$ Compute $q^{(l)}_{i\to j}$ for all i and j
Decision:
Estimate the symbol sequence as $\boldsymbol{z}^k = \arg\max_{\alpha_l} \hat{\gamma}_{i,j}^{\alpha_l,k}$

H. SYMBOL FLIPPING DECODING ALGORITHMS

Let $d(\boldsymbol{z}j, \sigma i, j)$ represent the binary Hamming distance between the extrinsic information $\sigma_{i,j}$ and the hard decision

symbol zj, with $\theta d(zj, \sigma i, j)$ denoting the corresponding weighting factor. If η_α indicates the occurrence count of an element $\alpha \in GF(q)$ based on the plurality logic of each element $\sigma_{i,j}$ at the j^{th} variable node, then the decision for an estimated correct symbol $v_j^{(k)}$ at the k^{th} iteration in the weighted algorithm B (wt.Algo B) is determined as follows:

$$v_j^{(k)} = \begin{cases} \alpha, & \text{if } \eta_\alpha \theta_{d(\alpha, \boldsymbol{z}_j^{(k)})} \geq \tau \\ \boldsymbol{z}_j^{(k)}, & \text{otherwise.} \end{cases} \tag{17}$$

In this scenario, α corresponds to the largest product of $\eta_\alpha \theta_{d(\alpha, \boldsymbol{z}_j^{(k)})}$, and the predetermined threshold τ is determined through simulations. Within the voting scheme [33], [40], each unsatisfied check node casts one vote towards the respective variable node. Afterwards, the j^{th} variable node accumulates all such votes, labeled as $V_j^{(k)}$, originating from the failed check nodes during the k^{th} iteration.

$$V_j^{(k)} = \sum_{i \in M(j)} V_{i,j}^{(k)} \tag{18}$$

where $V^{(k)}i, j = 1$ if $\boldsymbol{s}^{(k)}i \neq \mathbf{0}$; otherwise, $V^{(k)}i, j = 0$. Once the accumulated voting $V^{(k)}j$ reaches or exceeds a predefined threshold V_{th}, those variable nodes satisfying the condition $V_j \geq V_{th}$ undergo calculation of the flipping function $E^{(k)}j$ using equations (17) and (18). Assuming $\boldsymbol{\delta}$ retains the positions of all variable nodes for $Vj \geq V_{th}$, the values of $\boldsymbol{\delta}$ at the k^{th} iteration can be determined as follows:

$$\boldsymbol{\delta}_{j'}^{(k)} = V_j^{(k)}, \quad if \quad V_j \geq V_{th} \tag{19}$$

For each $1 \leq j' \leq p$, where p represents the total number of shortlisted variable nodes, this approach simplifies computational complexity from considering all n variable nodes to just a few, namely p. In this scenario, $\delta^{(k)}j' \in \boldsymbol{\delta}^{(k)}$ denotes the position of each shortlisted variable node, and $\boldsymbol{\delta}^{(k)}$ encompasses all such positions. Essentially, $\boldsymbol{\delta}^{(k)}$ comprises positions of variable nodes containing less reliable information, which should be substituted with dependable symbols from $\boldsymbol{z}^{*(k)}j' \in \Gamma_{j'}^{*(k)}$.

The multiple voting method delineated in [40] introduces two voting thresholds, $\zeta_0 > \zeta_1 > 0$, employing the same voting function as defined in (18). When $s^{(k)}j \neq 0$, $V^{(k)}ij = \zeta_0$ is assigned to the variable node with the highest flipping function, while $V^{(k)}ij = \zeta 1$ is assigned to the variable node with the second highest flipping function. Conversely, the proposed algorithms opt for a single-level voting approach in the shortlisting of the least reliable symbols, resulting in reduced memory consumption and computational complexity.

Unlike Algorithm B, which solely considers information prior to flipping, recent algorithms like D-SFDP and P-SFDP [43] leverage both pre-flip and post-flip information. These algorithms also take into account hard reliability to accurately assess the contribution of unsatisfied checksums. For the received hard decision symbol sequence $\boldsymbol{z} = (z1, z2, \ldots, zj, \ldots, zn)$, $(q-1)$ values for each symbol in the sequence are predicted through an exhaustive search for the symbol value that maximizes an objective function. The flipping function, as derived in [43], predicts symbols for each position of the received hard decision sequence $\boldsymbol{z}^{(k)}j$ such that $\boldsymbol{z}^{(k)}j \neq \ddot{\boldsymbol{z}}^{(k)}j$ and $\ddot{\boldsymbol{z}}^{(k)}j$ encompasses all possible symbol values of the $GF(q)$ except for the symbol value of $\boldsymbol{z}^{(k)}j \in GF(q)$. Consequently, there are $q-1$ potential values for each position of $\ddot{\boldsymbol{z}}^{(k)}j$, represented as:

$$\ddot{\Gamma}_j^{(k)} = \{\ddot{\boldsymbol{z}}_j^{(k)} : \ddot{\boldsymbol{z}}_j^{(k)} \epsilon GF(q), \ddot{\boldsymbol{z}}_j^{(k)} \neq \boldsymbol{z}_j^{(k)}\} \tag{20}$$

Explain the binary operation relating to the hard decision symbol $\boldsymbol{z}^{(k)}j$ and its associated channel soft symbol information $\boldsymbol{y}j$ as:

$$\boldsymbol{z}_j^{(k)} \odot \boldsymbol{y}_j = \sum_{t=0}^{r-1} (2z_{j,t}^{(k)} - 1)\boldsymbol{y}_{j,t} \tag{21}$$

The flipping mechanism within the D-SFDP algorithm integrates robust reliability to predict the most trustworthy symbol candidate, which can be formulated as follows:

$$E_j^{(k)}(\ddot{\boldsymbol{z}}_j^{(k)}, \boldsymbol{z}_j^{(k)}) = \ddot{\boldsymbol{z}}_j^{(k)} \odot \boldsymbol{y}_j + \sum_{i \in M_j} \theta_{d(\ddot{\boldsymbol{z}}_j^{(k)}, \sigma_{i,j}^{(k)})} - \boldsymbol{z}_j^{(k)} \odot \boldsymbol{y}_j - \sum_{i \epsilon M_j} \theta_{d(\boldsymbol{z}_j^{(k)}, \sigma_{i,j}^{(k)})} \tag{22}$$

In the P-SFDP algorithm based on plurality logic, the extrinsic weighting coefficients are denoted as $\boldsymbol{\eta} = [\eta_0, \eta_1, \ldots, \eta_l, \ldots \eta_{d_c}]$, where η_l signifies the weighting coefficient corresponding to the occurrence count of $\alpha \in GF(q)$ in the extrinsic information-sum $\sigma^k i, j$ for $0 \leq l \leq dc$. Equation (22) is then modified to incorporate hard reliability, representing the occurrence count of the element $\alpha \in GF(q)$ for the j^{th} symbol, as follows:

$$E_j^{(k)}(\ddot{\boldsymbol{z}}_j^{(k)}, \boldsymbol{z}_j^{(k)}) = \ddot{\boldsymbol{z}}_j^{(k)} \odot \boldsymbol{y}_j + \boldsymbol{\eta}_l(\ddot{\boldsymbol{z}}^{(k)}) - \boldsymbol{z}_j^{(k)} \odot \boldsymbol{y}_j - \boldsymbol{\eta}_l(\boldsymbol{z}^{(k)}) \tag{23}$$

Predicting all $q-1$ values of $\ddot{\boldsymbol{z}}^{(k)}j$ for each symbol in $\boldsymbol{z} = (z1, z2, \ldots, zj, \ldots, zn)$ makes the algorithm computationally intensive and demands significant memory resources to store processed values. Particularly for large values of $GF(q)$ and lengthy code lengths n, the algorithm [43] struggles with slow performance in finding candidate symbol values for positions to be flipped. The flipping function $E^{(k)}j$ of the j^{th} symbol and its corresponding flipped value $v_j^{(k)}$ are computed as follows:

$$E_j^{(k)} = \max_{\ddot{\boldsymbol{z}}_j^{(k)} \in \ddot{\Gamma}_j^{(k)}} E_j^{(k)}(\ddot{\boldsymbol{z}}_j^{(k)}, \boldsymbol{z}_j^{(k)}) \tag{24}$$

$$v_j^{(k)} = \underset{\ddot{\boldsymbol{z}}_j^{(k)} \in \ddot{\Gamma}_j^{(k)}}{\operatorname{argmax}} E_j^{(k)}(\ddot{\boldsymbol{z}}_j^{(k)}, \boldsymbol{z}_j^{(k)}) \tag{25}$$

The computation of the flipping function $E_j^{(k)}$ is highly intricate, and it also demands significant memory resources to store all the $q-1$ predicted values for n symbols.

Shortlisting Candidate Symbols Through Voting: Assume that $\boldsymbol{\delta}$ preserves the positions of all variable nodes where

$Vj \geq V_{th}$. Consequently, the values of $\boldsymbol{\delta}$ at the k^{th} iteration can be established as follows:

$$\boldsymbol{\delta}^{(k)}j' = V^{(k)}j \quad \text{if} \quad V_j \geq V_{th} \tag{26}$$

for each $1 \leq j' \leq p$, with p denoting the total number of short-listed variable nodes, this approach notably diminishes computational complexity from considering all n variable nodes to just a few, namely p.

In this context, $\delta^{(k)}j'$ signifies the position of each short-listed variable node, while $\boldsymbol{\delta}^{(k)}$ encompasses all these positions. Essentially, $\boldsymbol{\delta}^{(k)}$ includes the positions of variable nodes with less reliable information that necessitate replacement with dependable symbols from $\boldsymbol{z}^{*(k)}j' \in \Gamma_{j'}^{*(k)}$.

I. VOTING-BASED SYMBOL VALUE PREDICTION ALGORITHM

In this proposed algorithm, the least reliable variable nodes are short-listed by majority voting, and the candidate symbol value is selected using a prediction method. The prediction-based symbol flipping algorithms (D-SFDP and P-SFDP) offer better performance than most symbol flipping algorithms in the literature, but they come with exhaustive and computationally complex prediction mechanisms. Additionally, they require high memory to store the matrix of predicted values for all symbols in the codeword.

To address these issues, a voting-based approach is used in the proposed algorithm to reduce decoding latency and computational power by short-listing symbols with low reliability. The flipping function $E^{(k)}j(\ddot{\boldsymbol{z}}^{(k)}j, \boldsymbol{z}^{(k)}j)$ in Equation (30) is calculated $q-1$ times for the j^{th} variable node of the codeword n, considering the information before and after flipping to predict the most reliable symbol during each iteration. After calculating $E^{(k)}j(\ddot{\boldsymbol{z}}^{(k)}j, \boldsymbol{z}^{(k)}j)$ for all variable nodes, a matrix of size $n(q-1)$ is formed and must be stored for further processing.

The complexity of the flipping function $E^{(k)}j(\ddot{\boldsymbol{z}}^{(k)}j, \boldsymbol{z}_j^{(k)})$ can be reduced from n to p by first short-listing candidate symbols through voting, as given in Equations (35) and (36). This method also reduces memory requirements from $n(q-1)$ to $p(q-1)$. The newly improved algorithm [35], [36], termed as Voting-Based Symbol Flipping Decoding (V-SFD) algorithm, calculates the flipping function and corresponding flipped value for $j' \in$.

The bit error rate performance and complexity are further improved and simplified by using the voting scheme to flip multiple symbols in each iteration. Once these positions are selected, the bit with less reliability in each symbol is chosen to be flipped based on the channel reliability information This algorithm is termed as voting based multiple symbol flipping decoding (VB-MSFD). The least reliable symbol positions are short-listed by (18) and (26).

J. GROUP-BASED DECODING

This section discuss the novel approach [56] to dynamically reorganize variable nodes in each iteration using bit reliability and majority voting of individual received symbols, tailored for the symbol flipping decoding of NB-LDPC codes. The variable nodes with the lowest reliability are first grouped together, followed by those with the second level of reliability, and so forth, culminating in the last group, which is comparatively more reliable. This grouping process is executed by employing the cumulative density function (CDF) of the received sequence, which may be corrupted by Gaussian random noise. Within this algorithm, the number of variable nodes in each group is determined based on the reliability value assigned to each variable node.

Moreover, the symbol flipping decoding algorithm employs a fixed grouping scheme, where variable nodes are evenly distributed among a predetermined number of groups. During sub-iterations, a group is selected for decoding if it contains a variable node linked to a failed check.

Adaptive Group Shuffling: Given the parity check matrix H defined over $GF(q)$, the received sequence is partitioned into groups, with each group containing variable nodes based on the bit reliability of each symbol.

Variable nodes are grouped according to the reliability of both the channel's received soft and hard decision information. A symbol is deemed less reliable if it possesses a minimum reliability value and vice versa. The following equation is utilized to calculate the reliability of individual

Algorithm 7 Voting Based SFD Algorithm (V-SFD)

1: **Initialization:** For the received symbol $\boldsymbol{y}_j = \{y_{j,1}, y_{j,2}, \ldots, y_{j,r}\}$, compute the hard decision symbol $\boldsymbol{z}_j$ for $1 \leq t \leq r$, $1 \leq j \leq n$.
2: **for** $k = 1$ to I_{max} **do**
3: **Syndrome Check-Sum:** if $\boldsymbol{s}_i^{(k)} = \boldsymbol{0}$ or if $k = I_{max}$, stop decoding and output $\boldsymbol{z}^{(k)}$ as codeword.
4: Determine the unreliable symbol $\delta_{j'}^{(k)}$ using the voting scheme.
5: Calculate $E_{j'}^{(k)}(\ddot{\boldsymbol{z}}_{j'}^{(k)}, \boldsymbol{z}_{j'}^{(k)},)$ for $\boldsymbol{\delta}$ symbols and then calculate $E_{j'}^{(k)}$.
6: Find its flipped value $v_{j'}^{(k)}$ and update $\boldsymbol{z}^{(k+1)}$ with $v_{j'}^{(k)}$.
7: **end for**

Algorithm 8 Voting Based Multiple Symbol Flipping Decoding (VB-MSFD)

1: **Initialization:** For the received symbol $\boldsymbol{y}_j = \{y_{j,1}, y_{j,2}, \ldots, y_{j,r}\}$, compute the hard decision symbol $\boldsymbol{z}_j$ for $1 \leq t \leq r$, $1 \leq j \leq n$.
2: **for** $k = 1$ to I_{max} **do**
3: **Syndrome Check-Sum:** if $\boldsymbol{s}_i^{(k)} = 0$ or if $k = I_{max}$, stop decoding and output $\boldsymbol{z}^{(k)}$ as codeword.
4: Determine the unreliable symbol $\delta_{j'}^{(k)}$ using the voting scheme by equations (18) and (26).
5: Find the flipped values for each candidate symbol.
6: Update $\boldsymbol{z}^{(k+1)}$ with $\acute{\boldsymbol{z}}_{j'}^{(k)}$ and the channel soft information $\boldsymbol{y}^{(k+1)}$ with $\acute{\boldsymbol{y}}_{j'}^{(k)}$ respectively.
7: **end for**

symbols.

$$R_j^{(k)}(\boldsymbol{z}_j^{(k)}, \boldsymbol{y}_j^{(k)}) = \sum_{t=1}^{r} (2z_{j,t}^{(k)} - 1) y_{j,t}^{(k)} \tag{27}$$

here $R_j^{(k)}$ calculates the reliability information of a j^{th} symbol at k^{th} iteration for the hard decision symbol sequence $\boldsymbol{z}_j^{(k)}$ and the channel soft information $\boldsymbol{y}_j^{(k)}$. A symbol is considered to be less reliable if the numerical value of $R_j^{(k)}(\boldsymbol{z}_j^{(k)}, \boldsymbol{y}_j^{(k)})$ is minimum. As the received bit sequence is Gaussian random, therefore, the individual bit reliability is combined as a symbol reliability using equation (4).

Here it is important to mention that each group can have different number of variable nodes. The group containing the variable nodes with maximum votes is considered to be the most unreliable. After decoding of the first group, the next group is decoded under the defined criterion. *Definition*: Use inverse of F_{R_j} for grouping and let α be the total number of groups, then the number of variable nodes are allocated to the group τ if and only if the following condition holds:

$$F_{R_j}\left(1 - \frac{\tau}{\alpha}\right) < R_j \leq F_{R_j}\left(1 - \frac{\tau - 1}{\alpha}\right) \tag{28}$$

In this section, majority voting scheme presented in [35], and [36] is explained where each unsatisfied check node gives one vote to the relevant variable node. The j^{th} variable node then collects all the votes, say $V_j^{(k)}$, from the failed check nodes at k^{th} iteration.

$$V_j^{(k)} = \sum_{i \in M(j)} V_{i,j}^{(k)} \tag{29}$$

where $V_{i,j}^{(k)} = 1$ if $\boldsymbol{s}_i^{(k)} \neq \mathbf{0}$, otherwise $V_{i,j}^{(k)} = 0$. Those variable nodes fulfilling the condition $V_j \geq V_{th}$ will be passed to calculate the flipping function. This reduces the computational complexity from n number of variable nodes to just few variable nodes, say $\boldsymbol{\delta}^{(k)}$. Here $\boldsymbol{\delta}^{(k)}$ stores all the position of those variable nodes having votes greater than the predefined threshold V_{th} and $\delta_{j'}^{(k)}$ is the positions of the j^{th}variable node. In other words, $\boldsymbol{\delta}^{(k)}$ stores the positions of all the variable nodes having less reliable information and must be replaced with reliable symbols from $GF(q)$.

Average voting of each group before flipping are stored until the group processing is terminated for next iteration. However, average voting of each group is re-computed only after symbol flipping in that particular group. Average voting of a group remains the same if no symbol is flipped in a group. At each decoding iteration, three types of voting values are computed as follow:

$\bar{V}_\tau$: Average votes of each group. This voting is fixed after grouping the VNs till next iteration.

$\hat{V}_\tau$: Average votes for group G_τ after decoding.

$\ddot{V}_\tau$: Average votes for group G_τ after decoding of group $\tau - 1$.

Let $\beta^{(k,\tau)}$ be the number of variable nodes in a group τ, then the voting based reliability of each group τ at k^{th} iteration is determined by the following method:

$$\bar{V}_\tau^{(k)} = \frac{V_\tau^{(k)}}{\beta^{(k,\tau)}} \tag{30}$$

where $V_\tau^{(k)}$ is the sum of votes for a group τ.

Let τ be a group to be decoded, then a next group $G_{\tau+1}$ is selected for decoding based on voting using the following criterion:

$$\tau = \begin{cases} G_{\tau+1}, & \text{if } (\hat{V}_\tau > \bar{V}_\tau) \| (\hat{V}_\tau > \ddot{V}_\tau) \\ \text{Terminate sub-iteration}, & \text{otherwise.} \end{cases} \tag{31}$$

Algorithm 9 Algorithm 1: Adaptive Group Shuffling

```
for τ = 1 to α do
    Compute average voting V̄_τ for each group using Equation (30).
    Decode group τ.
    Compute votes V̂_τ after flipping.
    if the condition in Equation (29) is false then
        Terminate sub-iterations.
    end if
    Set τ = τ + 1. If τ = α, terminate sub-iteration.
end for
```

K. QAM BASED SYMBOL FLIPPING DECODING

The codeword $\boldsymbol{c} \in GF(q)$ undergoes mapping to the constellation χ of quadrature amplitude modulation (QAM) prior to transmission. If we define $\Omega(\cdot)$ as the symbol mapping function for the QAM constellation, then the codeword $\boldsymbol{c}$ is mapped as $\boldsymbol{x}j = \Omega(\boldsymbol{c}j) \in \chi$, generating a complex signal vector $\boldsymbol{x} = (\boldsymbol{x}1, \boldsymbol{x}2, \ldots, \boldsymbol{x}j, \ldots, \boldsymbol{x}n)$, which is transmitted over an additive white Gaussian noise (AWGN) channel. The received symbol sequence $\boldsymbol{y} = (\boldsymbol{y}1, \boldsymbol{y}2, \ldots, \boldsymbol{y}j, \ldots, \boldsymbol{y}n)$ is derived from the transmitted sequence $\boldsymbol{x}$ by introducing the complex additive white Gaussian channel noise $\mathbf{n} \rightarrow \mathcal{CN}(0, \sigma^2)$, characterized by zero mean and a two-sided power spectral density of $N_0/2$, such that $\boldsymbol{y}_j = \boldsymbol{x}_j + \boldsymbol{n}_j$.

At the receiver, each symbol $\hat{\boldsymbol{x}}_j$ is estimated by the detector using the maximum likelihood decision rule.

$$\hat{\boldsymbol{x}}_j = \arg\min_{\boldsymbol{x} \in \chi} \|\boldsymbol{y}_j - \boldsymbol{x}\| \tag{32}$$

for $j = 1, 2, \ldots, n$. The aforementioned equation (32) becomes exceedingly complex for high-order QAM. To mitigate this complexity, a sphere's radius D is defined, and the received symbol is estimated for those symbols within the region of D.

$$\hat{\boldsymbol{x}}j = \arg\min \boldsymbol{x} \in \chi |\boldsymbol{y}_j - \boldsymbol{x}| \leq D \tag{33}$$

In this paper [73], a QAM constellation size χ equal to the Galois field size q is adopted. For the given code rate $r_c = k/n$, where k is the information length, the spectral efficiency for this coded scheme is $\zeta = r_c \log_2 |\chi|$ bits per symbol.

At the onset of the non-binary sum-product and min-sum decoding algorithms, the LLR value for each symbol within

the received sequences [74] is calculated, assuming uniform probability distribution across all symbols.

$$LLR(c) = ln\left(\frac{Pr(\boldsymbol{y}|\hat{\boldsymbol{x}})}{Pr(\boldsymbol{y}|\boldsymbol{x})}\right) \tag{34}$$

where $\hat{\boldsymbol{x}}$ is the symbol value which maximizes the probability function $Pr(\boldsymbol{y}|\hat{\boldsymbol{x}})$, i.e.

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in GF(q)}{\operatorname{argmax}}\{Pr(\boldsymbol{y}|\boldsymbol{x})\}. \tag{35}$$

If the value of $LLR(\boldsymbol{c}) = 0$ or close to 0, it means more reliability and vice versa. The Euclidean distance based likelihood method is used to calculate the approximate position of the received symbol using *LLR* which is given as:

$$LLR(\alpha) = \frac{2}{\sigma^2}(d(\boldsymbol{y}, \alpha)^2 - d(\boldsymbol{y}, \hat{\alpha}^2)) \tag{36}$$

here, $\hat{\alpha}^2$ represents the closest QAM point to the received symbol $\boldsymbol{y}$, and α denotes the primitive element of $GF(q)$ within the QAM constellation.

In the novel decoding algorithm [51], [52], an improved iterative hard reliability-based majority logic decoding algorithm is proposed, as described in [53], and [55]. This algorithm not only exhibits minimal complexity during initialization but also facilitates the iterative computation of symbol reliabilities $R_{j,l}$.

Algorithm 10 Enhanced Iterative Joint Detection-Decoding (E-IJDD)

1: **Initialization:** For $1 \leq j \leq n$, $1 \leq l \leq q$ and $1 \leq i \leq m$.
2: **Signal Detection:**
3: Obtain hard decision symbols $\boldsymbol{z_j}$ from received symbols $\boldsymbol{y_j}$
4: Calculate $\hat{\boldsymbol{x}}_{j,1}^{(k)}$, $\hat{\boldsymbol{x}}_{j,2}^{(k)}$, and $\Delta d_j^{(k)}$.
5: **Decoder:**
6: If $\boldsymbol{z}_j = \alpha_l \in GF(q)$, then $R_{j,l} = \gamma$ else $R_{j,l} = 0$.
7: **Repeat**
8: **Iteration** $k = 1$ to I_{max}.
9: **Syndrome Check-Sum:** If $\boldsymbol{s}_i^{(k)} = \sum_{j\in N(i)} h_{ij}\boldsymbol{z}_j^{(k)} = 0$ or if $k = I_{max}$, stop decoding and output $\boldsymbol{z}^{(k)}$ as codeword.
10: Calculate extrinsic information-sums $\sigma_{ij}^{(k)}$.
11: Calculate symbol reliability $R_{j,l}^{(k)}$ and update $R_{j,l}^{(k+1)}$.
12: Select the new candidate symbol value $\boldsymbol{z}_j^{(k+1)}$.
13: Adjust the symbol and feedback to detector.
14: $k = k + 1$ for the next iteration.
15: **Until** convergence.

VII. GLIMPSES OF BER PERFORMANCES CURVES

To provide glimpses of Bit Error Rate (BER) performances for Non-Binary Low-Density Parity-Check codes, the performances curves are for various decoding algorithms are shown, which illustrating how BER varies with different parameters or under various conditions.

The section discusses the BER performance evaluation of various algorithms, as depicted in figures 1, 2 and 3. It indicates that the QSPA algorithm stands out as optimal, with some sub-optimal algorithms demonstrating performance close to it. Message passing algorithms are noted for their superior performance overall, while the reliability-based wBRB algorithm shows competitive performance. However, symbol flipping algorithms still exhibit BER performance that is unsuitable for practical applications, suggesting they have yet to reach a level of effectiveness comparable to other algorithms.

In this paper, the signal-to-noise ratio (SNR) is defined as E_b/N_0 for BPSK, while for higher-order modulation schemes such as QAM, we use E_s/N_0 to account for symbol energy considerations.

The figure 1 displays the Bit Error Rate (BER) versus E_b/N_0 performance curves for several decoding algorithms applied to the NB-LDPC code (204, 102) over $GF(16)$. Each curve illustrates how the BER decreases as E_b/N_0 increases, with lower points on the BER axis at a given E_b/N_0 indicating better performance. The D-SFDP algorithm (blue line with diamonds) performs moderately well, achieving a BER of 10^{-6} at around 6.5 dB E_b/N_0. The MV-PSFD algorithm (red line with down-pointing triangles) shows good performance, achieving a BER of 10^{-6} at around 5.8 dB E_b/N_0. The PSFD algorithm (orange line with circles) has the poorest performance among the seven, with a BER of 10^{-4} at around 6.5 dB E_b/N_0, and it does not reach a BER of 10^{-6} within the range shown. The V-SFD algorithm (blue line with upward-pointing triangles) performs reasonably well, achieving a BER of 10^{-6} at around 6.5 dB E_b/N_0. The VB-MSFD algorithm (brown line) shows excellent performance, achieving a BER of 10^{-6} at around 4.8 dB E_b/N_0, making it one of the best among the seven. The MV-SF algorithm (black line with asterisks) performs well, achieving a BER of 10^{-6} at around 5 dB E_b/N_0. The wBRB algorithm (magenta line with left-pointing triangles) has the best performance, reaching a BER of 10^{-6} at around 4.8 dB E_b/N_0. In summary, the VB-MSFD (green line) and wBRB (magenta line) algorithms demonstrate the best performance, achieving low BERs at lower E_b/N_0 values. MV-SF (black line) and MV-PSFD (red line) also show good performance, while PSFD (orange line) has the worst performance, requiring the highest E_b/N_0 to achieve similar BER values. The remaining algorithms, D-SFDP (blue line) and V-SFD (blue line with upward-pointing triangles), show moderate performance.

In the figure 2, QSPA (black line) has the best performance among the four algorithms, achieving a BER of 10^{-6} at around 4.5 dB E_b/N_0. EMS_{16} (violet line) performs slightly worse than QSPA but better than the other two algorithms, reaching a BER of 10^{-6} at just under 5 dB E_b/N_0. wBRB-s (red line) shows a steeper decline in BER but requires a higher E_b/N_0 (around 5.2 dB) to achieve a BER of 10^{-6}. MV-SF_4 (yellow line) also performs well but not as good as QSPA

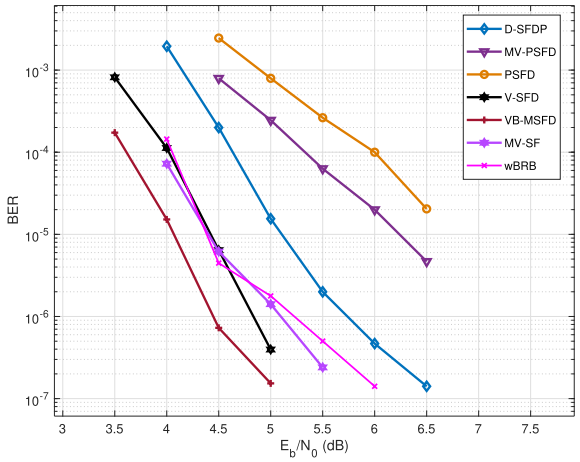

FIGURE 1. BER Curves of the (204, 102) NB-LDPC codes over GF(16).

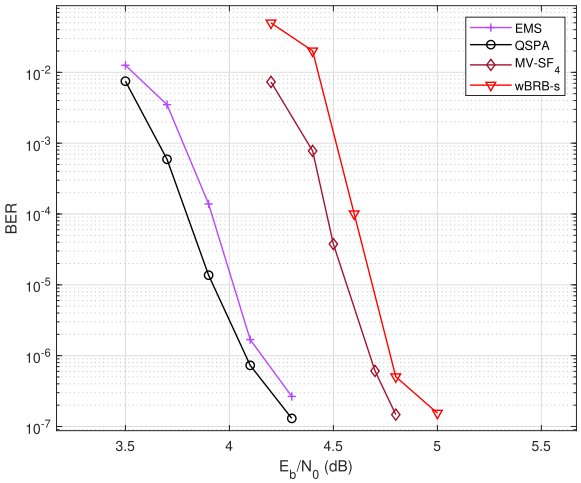

FIGURE 2. BERs of the (837, 726) NB-LDPC code over GF(32).

and EMS_{16}, achieving a BER of 10^{-6} at about 5 dB E_b/N_0, similar to wBRB-s.

The figure 3 shows the BER versus E_s/N_0 performance curves for three different decoding algorithms applied to the NB-LDPC code (45, 2, 3) over $GF(8)$. Each curve demonstrates how BER decreases as E_s/N_0 increases, with lower points on the BER axis indicating better performance. The E-IJDD algorithm (violet line) has the best performance, achieving a BER of 10^{-6} at around 5 dB E_s/N_0, indicating high effectiveness at lower SNR levels. The Improved IJDD algorithm (blue line) performs slightly worse than E-IJDD but significantly better than the IJDD algorithm, reaching a BER of 10^{-6} at around 9 dB E_s/N_0. The IJDD algorithm (dark brown line) shows the steepest decline in BER but requires a much higher E_s/N_0 to achieve low BER values, achieving a BER of 10^{-6} at approximately 15 dB E_s/N_0. Thus, E-IJDD is the most efficient algorithm for error correction, followed by Improved IJDD, while IJDD is the least efficient under the same conditions.

In this work, figures 1 and 2 have been grouped to present a performance comparison for NB-LDPC codes over GF(32) and GF(16), enabling a direct visual comparison

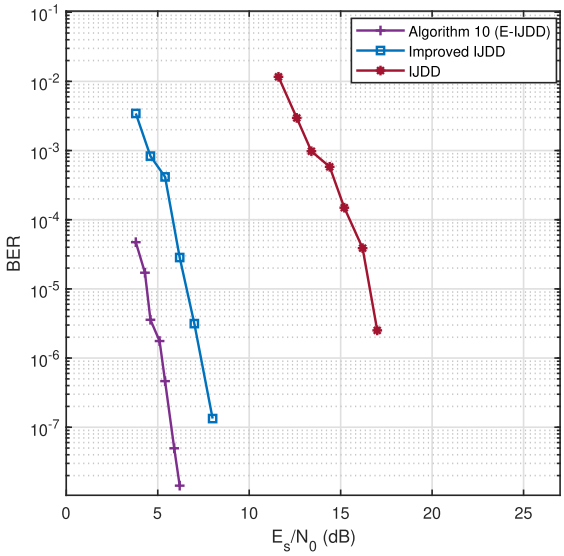

FIGURE 3. BER Performance of NB-LDPC Code (45,2,3) over GF(8).

of their relative performance. Figure 3 provides results for GF(8), illustrating the impact of a smaller field size on decoding performance. The grouping of algorithms within each figure has been carefully arranged to facilitate performance comparisons while maintaining computational efficiency during simulations.

A. THEORETICAL PERFORMANCE BOUND: POLYANSKIY-POOR-VERDU CONVERSE BOUND

To provide a theoretical reference for the performance of the proposed NB-LDPC decoding algorithms, we compare the simulation results with the Polyanskiy-Poor-Verdu (PPV) converse bound for finite block-length coding [75]. This bound establishes the fundamental limit on the achievable rate for a given block length, code rate, and error probability.

For the (837, 726) NB-LDPC code over GF(32), we compute the PPV bound using the following expression:

$$R \leq C - \sqrt{\frac{V}{n}} Q^{-1}(\epsilon) + \frac{\log n}{n} \tag{37}$$

where C is the channel capacity, V is the channel dispersion, n is the block length, and ϵ is the target error probability.

For the given parameters: $n = 837$, $R = 0.867$, and $\epsilon = 10^{-5}$, the computed PPV bound is approximately:

$$R \leq 0.776 \tag{38}$$

This indicates that, despite the high code rate, the algorithm operates near the theoretical limit.

VIII. COMPLEXITY ANALYSIS

In this section, the computation complexity of the proposed algorithms is analysed in detail and compared to that of the Improved IJDD algorithm. Consider an LDPC code (N, d_v, d_c) over GF(q) and let δ represents the nonzero elements in the parity check matrix $\boldsymbol{H}$ where $\gamma = nd_v = md_c$. At the ML QAM detector, q symbols

TABLE 2. Computational complexity of different non-binary LDPC decoding algorithms per iteation.

Algorithms	Number of operations				
	GM	GA	IA/RA	RC/IC	IM/RM
E-IJDD	$3nd_v$	$3nd_v$	$2n(q+1)$	$2n(q-1)$	nq
IIJDD	$3nd_v$	$3nd_v$	$2nd_v+n+2n(q+1)$	$2n(q-1)$	$3nq+4n$
EMS	$2nd_vn_m$	$9n_m(nd_v-2m)$	$n_m(20nd_v-18m-12n)$	$n_m log_2 n_m(9nd_v-12m-4n)$	
IJDD	$2nd_v$	$2nd_v-m$	nd_v+n+4n	$n(2q-3)$	$4n$
wt-Algo.B	$2nd_v$	$3nd_v-m$	nd_v	nd_v	nd_v
ISRB	$2nd_v$	$2nd_v-m$	nd_v+n2^r	$2n2^r-2n$	$n2^r$
V-SFD	$d_v(2d_c-1)$	$pd_v(d_v+r)+2d_v(d_c-1)$	$p(d_v^2+2d_vr+4r+2d_v)$	$nr+p(d_v-1)+2(p-1)$	
M-QAM FFT-QSPA	$(2q+1)nd_v$	$5nd_vq$	$6m+2qnd_vlog_2q$	$7m+qnd_v(d_v+d_c-2)$	$n(q-1)$

IM/IC/IA: Integer Multiplication/Comparision/Adittion; p is the total number of variable nodes stored in $\boldsymbol{\delta}$.
RA/RM/RC: Real Addition/Multiplication/Comparision; GA/GM: Galois Field Addtion/Multiplication;
Notations used: $\gamma = nd_v = md_c$, $L = \boldsymbol{v}^{\xi}$ such that $\xi < d_c, n_m < q$.

of the QAM constellation are tested for each of the n received symbol sequence which results in nq real addition and multiplication. These operation are reduced to $(\frac{n}{4})q$ for the proposed quadrant based detection in this paper. The proposed Algorithms 1 and 3 require z_j only as opposite to IIJDD algorithm which require the pair of $\{(\boldsymbol{z}_{j,1}, \boldsymbol{z}_{j,2}), \Delta d_j\}$ information. This results in further complexity reduction by $2nq + n$ real additions.

At the NB-LDPC decoder initialization, md_c Galois field multiplication and $m(d_v - 1)$ additions are required to compute the m check-sum and qn real comparisons are required to initialize the reliability information in case of Algorithm 2 and 3 only. For Algorithm 2, the update of the extrinsic information-sum $\sigma_{i,j}$ needs pair of $(\boldsymbol{z}_{j,1}, \boldsymbol{z}_{j,2})$, therefore, $2md_v$ Galois field multiplications and $2m(d_c - 1)$ Galois field additions are required. On the other hand Algorithms 1 and 3 require md_c multiplications and $m(d_c - 1)$ additions which means two times less computational complexity than IIJDD algorithm and Algorithm 2.

In the IIJDD algorithm, a pair of message $(v_j, \Delta\eta_j)$ is sent as feedback to the QAM detector while in the proposed algorithms, only the updated soft information $\boldsymbol{y}_j^{(k+1)}$ is used for the feedback. To compute the feedback $\boldsymbol{y}_j^{(k+1)}$, n real addition and multiplications are required for the proposed Algorithm 2 while for Algorithm 1, only ρ real addition and multiplications are required. To calculate the symbol flipping function in the Algorithm 1, pd_v real addition and p real multiplications are required. To compute the candidate symbol value for the position in error, d_v times the check-sum $\boldsymbol{s}_i$ is calculated. Therefore, to predict a single symbol value from the quadrant QAM constellation, $\frac{1}{4}qd_v(d_c - 1)$ Galois field addition and $\frac{1}{4}qd_v(d_c)$ multiplications are required for the symbols.

Table 2 shows the computational complexity of the proposed decoding algorithms in comparisons with other algorithms from the literature.

IX. SOME APPLICATION OF NB-LDPC

A. POWER LINE COMMUNICATION

Power Line Communication technology utilizes existing electrical power lines for data transmission. This model integrates non-binary Low-Density Parity-Check decoding to enhance error correction capabilities, improving reliability and efficiency in 6G mobile network applications. In the transmitter, input data bits are generated or sourced from a higher-layer application and grouped into symbols suitable for non-binary LDPC encoding. The data symbols are then encoded using a non-binary LDPC encoder, generating parity-check bits according to a pre-defined non-binary LDPC parity-check matrix. The encoded symbols are modulated using an appropriate scheme, such as Quadrature Amplitude Modulation or Phase Shift Keying (PSK), and then transformed into the time domain. A cyclic prefix (CP) is appended to each symbol to mitigate inter-symbol interference (ISI), and the symbols are transmitted over power lines, encountering various channel impairments like noise, attenuation, and multipath effects.

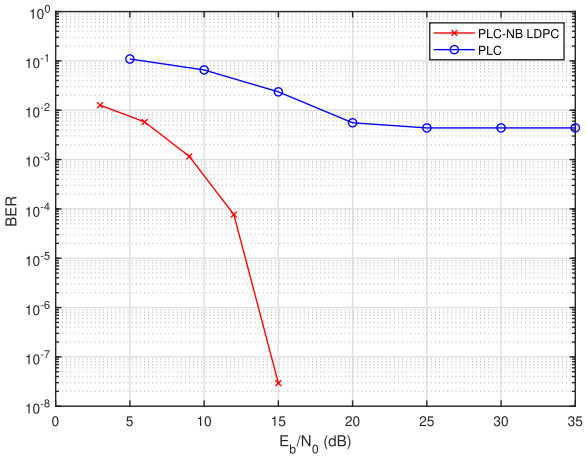

FIGURE 4. BER Curves of PLC using NB-LDPC Codes.

Power Line Communication channels exhibit unique characteristics due to the use of electrical power lines for data transmission. The channel model [76], [77] for PLC needs to account for various impairments and effects that are typically encountered in power line environments. These impairments include additive noise, multipath propagation, frequency selectivity, and time-varying conditions.

The received signal $y(t)$ in a PLC channel can be modeled as:

$$y(t) = h(t) * x(t) + n(t), \tag{39}$$

where $x(t)$ is the transmitted signal, $h(t)$ is the channel impulse response, $*$ denotes convolution, and $n(t)$ represents the noise component.

The channel impulse response $h(t)$ captures the effects of multipath propagation and can be represented as a sum of delayed and attenuated impulses:

$$h(t) = \sum_{i=1}^{L} \alpha_i \delta(t - \tau_i), \tag{40}$$

where α_i is the attenuation factor for the i-th path, τ_i is the delay for the i-th path, L is the number of multipath components, and $\delta(t)$ is the Dirac delta function.

The frequency response $H(f)$ of the channel can be obtained by taking the Fourier transform of the impulse response $h(t)$:

$$H(f) = \mathcal{F}\{h(t)\} = \sum_{i=1}^{L} \alpha_i e^{-j2\pi f \tau_i}. \tag{41}$$

The noise $n(t)$ in PLC channels is typically modeled as a combination of different noise types:

1. Additive White Gaussian Noise (AWGN): This is a basic noise model that accounts for thermal noise and is modeled as a Gaussian process with zero mean and variance σ^2.

2. Impulse Noise: Impulse noise is characterized by short-duration, high-amplitude spikes and can be modeled using a Poisson distribution for the occurrence of impulses and a Gaussian distribution for their amplitude.

3. Colored Background Noise: This type of noise is non-white and has a frequency-dependent power spectral density. It can be modeled as filtered white noise.

Combining these noise components, the total noise $n(t)$ can be represented as:

$$n(t) = n_{\text{AWGN}}(t) + n_{\text{impulse}}(t) + n_{\text{colored}}(t). \tag{42}$$

The complete PLC channel model incorporating the multipath effects and various noise components can be expressed as:

$$y(t) = \left(\sum_{i=1}^{L} \alpha_i \delta(t - \tau_i) \right) * x(t) + n_{\text{AWGN}}(t) + n_{\text{impulse}}(t) + n_{\text{colored}}(t). \tag{43}$$

This model provides a comprehensive representation of the PLC channel, capturing the key impairments that affect data transmission over power lines.

To improve the BER of power line communication, particularly in combating noise associated with PLC channels, we suggest a NB-LDPC decoding algorithm (Algorithm 10) called enhancement iterative joint detection-decoding (E-IJDD) algorithm. This enhanced algorithm utilizes iterative hard decision-based majority logic to select new candidate symbol values.

The algorithm functions by adjusting the feedback value to the QAM detector, calculating the Euclidean distance between the current symbol and the newly selected symbol value. This approach effectively reduces the impact of impulse noise, which is prevalent in PLC environments. The probability density function (PDF) of AWGN is given by:

$$p_{AWGN}(x) = \frac{1}{\sqrt{2\pi\sigma_{AWGN}^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma_{AWGN}^2} \right) \tag{44}$$

The PDF of impulsive noise is given by:

$$p_{Impulsive}(x) = \frac{1}{\sqrt{2\pi\sigma_I^2}} \exp\left(-\frac{(x-\mu_I)^2}{2\sigma_I^2} \right) \tag{45}$$

In a PLC system, the total noise can be modeled as a combination of both AWGN and impulsive noise. The combined noise model can be expressed as a mixture of these two Gaussian processes.

$$p_{Noise}(x) = \alpha p_{AWGN}(x) + (1-\alpha) p_{Impulsive}(x) \tag{46}$$

The figure 4 illustrates a PLC system using QAM modulation where blanking is employed to suppress impulse noise, and the performance is compared to a PLC system using NB-LDPC decoding algorithm without blanking. The graph demonstrates a significant improvement in a BER performance when the E-IJDD decoding algorithm (Algorithm 10) is applied, utilizing a (105, 2, 5) code over Galois Field GF(16). This enhancement underscores the effectiveness of the E-IJDD algorithm in mitigating the effects of impulse noise and improving overall system reliability.

B. VISIBLE LIGHT COMMUNICATION

Visible Light Communication utilizes the visible light spectrum for wireless communication, typically using LEDs for transmission and photodiodes for reception. The channel model for VLC needs to account for various impairments and effects that are typically encountered in indoor and outdoor environments, such as line-of-sight (LOS) propagation, reflections, and ambient light noise.

The received signal $y(t)$ in a VLC channel [78], [79] can be modeled as:

$$y(t) = h(t) * x(t) + n(t), \tag{47}$$

where $x(t)$ is the transmitted signal, $h(t)$ is the channel impulse response, $*$ denotes convolution, and $n(t)$ represents the noise component.

The channel impulse response $h(t)$ for VLC can be represented as a sum of the line-of-sight (LOS) component and the non-line-of-sight (NLOS) components due to reflections:

$$h(t) = h_{\text{LOS}}(t) + h_{\text{NLOS}}(t), \tag{48}$$

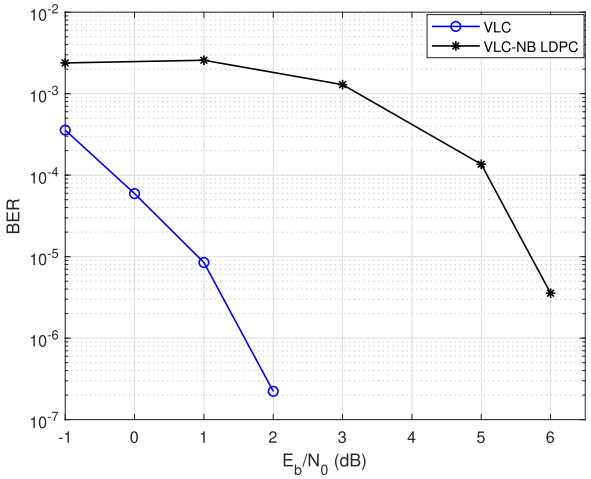

FIGURE 5. BER Performance of NB-LDPC in VLC.

where $h_{\text{LOS}}(t)$ represents the direct path between the transmitter and the receiver, and $h_{\text{NLOS}}(t)$ represents the multipath components due to reflections.

The LOS component $h_{\text{LOS}}(t)$ can be modeled as:

$$h_{\text{LOS}}(t) = H_{\text{LOS}}\delta(t - \tau_{\text{LOS}}), \tag{49}$$

where H_{LOS} is the channel gain for the LOS path, and τ_{LOS} is the delay for the LOS path.

The NLOS component $h_{\text{NLOS}}(t)$ can be modeled as:

$$h_{\text{NLOS}}(t) = \sum_{i=1}^{L} H_{\text{NLOS},i}\delta(t - \tau_{\text{NLOS},i}), \tag{50}$$

where $H_{\text{NLOS},i}$ is the channel gain for the i-th reflected path, $\tau_{\text{NLOS},i}$ is the delay for the i-th reflected path, and L is the number of significant reflections.

The frequency response $H(f)$ of the channel can be obtained by taking the Fourier transform of the impulse response $h(t)$:

$$H(f) = \mathcal{F}\{h(t)\} = H_{\text{LOS}}e^{-j2\pi f\tau_{\text{LOS}}} + \sum_{i=1}^{L} H_{\text{NLOS},i}e^{-j2\pi f\tau_{\text{NLOS},i}}. \tag{51}$$

The noise $n(t)$ in VLC channels typically includes shot noise, thermal noise, and ambient light noise:

1. Shot Noise: This noise is caused by the random nature of photon arrival at the photodiode and can be modeled as a Gaussian process with zero mean and variance σ^2_{shot}.

2. Thermal Noise: This noise is generated by the thermal agitation of charge carriers in the photodiode and can also be modeled as a Gaussian process with zero mean and variance $\sigma^2_{\text{thermal}}$.

3. Ambient Light Noise: This noise is due to ambient light sources such as sunlight and artificial lights, and can be modeled as a Gaussian process with zero mean and variance $\sigma^2_{\text{ambient}}$. Combining these noise components, the total noise $n(t)$ can be represented as:

$$n(t) = n_{\text{shot}}(t) + n_{\text{thermal}}(t) + n_{\text{ambient}}(t). \tag{52}$$

The complete VLC channel model incorporating the LOS, NLOS components, and various noise components can be expressed as:

$$y(t) = \left(H_{\text{LOS}}\delta(t - \tau_{\text{LOS}}) + \sum_{i=1}^{L} H_{\text{NLOS},i}\delta(t - \tau_{\text{NLOS},i})\right) * x(t) + n_{\text{shot}}(t) + n_{\text{thermal}}(t) + n_{\text{ambient}}(t). \tag{53}$$

This model provides a comprehensive representation of the VLC channel, capturing the key impairments that affect data transmission over visible light.

In the context of VLC, shot noise arises due to the quantized nature of light and is typically characterized by the following parameters:

- **Mean current** (μ): This is the average current produced by the photodetector due to the received optical power.
- **Variance of shot noise** (σ^2): This is proportional to the average current.

The shot noise can be expressed in terms of a Gaussian distribution with a mean (μ) and variance (σ^2). The probability density function (PDF) of the shot noise current I in a VLC system can be approximated by a Gaussian (normal) distribution as follows:

$$f_I(i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(i-\mu)^2}{2\sigma^2}\right) \tag{54}$$

where:

- i is the shot noise current.
- μ is the mean current.
- σ^2 is the variance of the shot noise.

1. **Mean Current** (μ):

$$\mu = q \cdot \eta \cdot P_r \tag{55}$$

where:

- q is the elementary charge (1.602×10^{-19} coulombs).
- η is the photodetector responsivity (A/W).
- P_r is the received optical power (W).

2. **Variance** (σ^2):

$$\sigma^2 = 2q \cdot \mu \cdot B \tag{56}$$

where:

- B is the bandwidth of the receiver (Hz).

Putting it all together, the PDF of the shot noise current in a VLC system is:

$$f_I(i) = \frac{1}{\sqrt{4\pi q\eta P_r B}} \exp\left(-\frac{(i-q\eta P_r)^2}{4q\eta P_r B}\right) \tag{57}$$

This PDF describes the statistical behavior of shot noise in a VLC system, assuming that it can be approximated by a Gaussian distribution.

- **Optical Power**: The received optical power P_r can be calculated using the Lambertian model for the LED and the responsivity of the photodiode.
- **Lambertian Emission**: The LED emission pattern can be approximated by a Lambertian radiation pattern, characterized by the Lambertian order m.
- **Channel DC Gain** H_{LoS}: The direct current (DC) channel gain for LoS VLC can be expressed as:

$$H_{LoS} = \frac{(m+1)A_r}{2\pi d^2} \cos^m(\phi) \cos(\psi) \, \text{rect}\left(\frac{\psi}{\Psi_{FOV}}\right) \tag{58}$$

where:

  - m is the Lambertian order.
  - A_r is the physical area of the receiver.
  - d is the distance between the transmitter and receiver.
  - ϕ is the angle of emission with respect to the transmitter's normal.
  - ψ is the angle of incidence with respect to the receiver's normal.
  - Ψ_{FOV} is the field of view of the receiver.

The figure 5 demonstrates that using a Quasi-Cyclic Non-Binary Low-Density Parity-Check code with parameters (120, 4, 8) over GF(16) significantly improves the BER performance in a VLC system. A fixed layer B-MSFD algorithm [56] is used to show the BER performance. The system parameters include a Lambertian order $m = 0.5$, a receiver area of $1 \times 10^{-8}\,\text{m}^2$, a distance of 5 meters between the transmitter and receiver, an emission angle of 70°, an incidence angle of 60°, and a receiver field of view (FOV) of 120°. These conditions highlight the QC NB-LDPC code's effectiveness in enhancing BER performance, especially in challenging VLC environments where factors such as distance, angular misalignment, and limited receiver area play significant roles.

X. FUTURE RESEARCH DIRECTIONS

Algorithm Optimization: Continued research efforts are needed to optimize non-binary LDPC decoding algorithms for improved efficiency and reduced complexity. Exploring novel optimization techniques, such as parallel processing, hardware acceleration, and algorithmic pruning, can lead to significant advancements in decoding performance.

A. ERROR FLOOR MITIGATION

Addressing the phenomenon of error floors remains a key challenge in non-binary LDPC decoding. Future research should focus on developing error floor mitigation techniques, such as modifying code construction methods, enhancing message passing algorithms, and incorporating soft decision feedback.

B. ADAPTIVE DECODING STRATEGIES

Investigating adaptive decoding strategies that dynamically adjust algorithm parameters based on channel conditions and code characteristics can enhance the robustness of non-binary LDPC decoding. Adaptive algorithms can improve error correction performance across varying signal-to-noise ratios and channel impairments.

C. INTEGRATION WITH EMERGING TECHNOLOGIES

With the proliferation of emerging communication technologies such as 5G, Internet of Things (IoT), and quantum communication, there is a need to adapt non-binary LDPC decoding algorithms to suit the requirements of these systems. Research efforts should focus on integrating non-binary LDPC codes with emerging modulation schemes, channel models, and communication protocols.

D. MACHINE LEARNING ASSISTED DECODING

Leveraging machine learning techniques for LDPC decoding holds promise for achieving near-optimal performance. Future research directions may explore the integration of deep learning models, reinforcement learning algorithms, and neural network architectures to enhance the decoding efficiency and adaptability of non-binary LDPC codes.

E. CROSS-LAYER OPTIMIZATION

Collaborative efforts between communication theory, coding theory, and signal processing can lead to cross-layer optimization techniques that exploit the synergies between different system components. Future research should explore holistic approaches that jointly optimize channel coding, modulation, and decoding strategies for enhanced system performance.

F. STANDARDIZATION AND INDUSTRIAL ADOPTION

Encouraging the standardization and adoption of non-binary LDPC codes in industry standards and communication protocols is essential for their widespread deployment. Collaborative initiatives involving academia, industry, and standardization bodies can facilitate the integration of non-binary LDPC coding solutions into commercial communication systems.

By pursuing these future directions, researchers can unlock the full potential of non-binary LDPC decoding algorithms and pave the way for their seamless integration into next-generation communication systems.

XI. CONCLUSION

This review has provided a comprehensive overview of non-binary LDPC decoding algorithms, highlighting their significance in modern communication systems. Non-binary LDPC codes offer superior error correction capabilities compared to binary LDPC codes, making them indispensable for various applications ranging from wireless communication to storage systems. The analysis of different decoding

algorithms, including the sum-product algorithm, min-sum algorithm, and belief propagation algorithm, has underscored the importance of algorithmic efficiency, complexity, and performance trade-offs. While each algorithm has its strengths and weaknesses, ongoing research efforts aim to enhance their decoding capabilities through algorithm optimization, error floor mitigation, and adaptive strategies. Looking ahead, future research directions outlined in this review include algorithm optimization, error floor mitigation, adaptive decoding strategies, integration with emerging technologies, machine learning-assisted decoding, cross-layer optimization, and standardization efforts. These directions offer promising avenues for advancing the state-of-the-art in non-binary LDPC decoding and unlocking their full potential in real-world communication systems.

REFERENCES

[1] C. Han, H. Li, and W. Chen, "Minimum distance optimization with chord edge growth for high girth non-binary LDPC codes," *Electronics*, vol. 9, no. 12, p. 2161, Dec. 2020.

[2] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[3] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[4] M. C. Davey and D. MacKay, "Low-density parity check codes over $GF(q)$," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, Jun. 1998.

[5] L. Barnault and D. Declercq, "Fast decoding algorithm for LDPC over $GF(q)$," in *Proc. Inf. Theory Workshop*, Apr. 2003, pp. 70–73.

[6] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[7] T. R. Halford and K. M. Chugg, "An algorithm for counting short cycles in bipartite graphs," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 287–292, Jan. 2006.

[8] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, "Lowering the error floor of LDPC codes using cyclic liftings," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 724–728.

[9] D. Divsalar, S. Dolinar, and C. Jones, "Construction of protograph LDPC codes with linear minimum distance," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 664–668.

[10] L. Zeng, L. Lan, Y. Y. Tai, B. Zhou, S. Lin, and K. A. S. Abdel-Ghaffar, "Construction of nonbinary cyclic, quasi-cyclic and regular LDPC codes: A finite geometry approach," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 378–387, Mar. 2008.

[11] S. J. Johnson and S. R. Weller, "Regular low-density parity-check codes from combinatorial designs," in *Proc. IEEE Inf. Theory Workshop*, Nov. 2002, pp. 90–92.

[12] B. Vasic, "Combinatorial constructions of low-density parity check codes for iterative decoding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2004, p. 312.

[13] X. Jiang and M. H. Lee, "Large girth non-binary LDPC codes based on finite fields and Euclidean geometries," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 521–524, Jun. 2009.

[14] L. Lan, L. Zeng, Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: A finite field approach," *IEEE Trans. Inf. Theory*, vol. 53, no. 7, pp. 2429–2458, Jul. 2007.

[15] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[16] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[17] Z. Wang and Z. Cui, "Low-complexity high-speed decoder design for quasi-cyclic LDPC codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 1, pp. 104–114, Jan. 2007.

[18] *The Consultative Committee for Space Data Systems (CCSDS)*. Accessed: Aug. 2023. [Online]. Available: https://public.ccsds.org/default.aspx

[19] M. Sarvaghad-Moghaddam, W. Ullah, D. N. K. Jayakody, and S. Affes, "A new construction of high performance LDPC matrices for mobile networks," *Sensors*, vol. 20, no. 8, p. 2300, Apr. 2020.

[20] W. Ullah, Jiangtao, and Y. FengFan, *Improved Min-Sum Decoding Algorithm for Moderate Length Low Density Parity Check Codes*. Cham, Switzerland: Springer, Nov. 2011, pp. 935–943.

[21] D. Declercq and M. Fossorier, "Extended minsum algorithm for decoding LDPC codes over $GF(q)$," in *Proc. Int. Symp. Inf. Theory, (ISIT)*, Sep. 2005, pp. 464–468.

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

[23] E. Li, D. Declercq, and K. Gunnam, "Trellis-based extended min-sum algorithm for non-binary LDPC codes and its hardware structure," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2600–2611, Jul. 2013.

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

[25] V. Savin, "Min-max decoding for non binary LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2008, pp. 960–964.

[26] X. Ma, K. Zhang, H. Chen, and B. Bai, "Low complexity X-EMS algorithms for nonbinary LDPC codes," *IEEE Trans. Commun.*, vol. 60, no. 1, pp. 9–13, Jan. 2012.

[27] S. Zhao, Z. Lu, X. Ma, and B. Bai, "A variant of the EMS decoding algorithm for nonbinary LDPC codes," *IEEE Commun. Lett.*, vol. 17, no. 8, pp. 1640–1643, Aug. 2013.

[28] C. Xiong and Z. Yan, "Improved iterative soft-reliability-based majority-logic decoding algorithm for non-binary low-density parity-check codes," in *Proc. Conf. Rec. 45th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2011, pp. 894–898.

[29] X. Zhang, F. Cai, and S. Lin, "Low-complexity reliability-based message-passing decoder architectures for non-binary LDPC codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 11, pp. 1938–1950, Nov. 2012.

[30] C. Xiong and Z. Yan, "Improved iterative hard- and soft-reliability based majority-logic decoding algorithms for non-binary low-density parity-check codes," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5449–5457, Oct. 2014.

[31] S. Yeo and I.-C. Park, "Improved hard-reliability based majority-logic decoding for non-binary LDPC codes," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 230–233, Feb. 2017.

[32] K. Jagiello and W. E. Ryan, "Iterative plurality-logic and generalized algorithm B decoding of Q-ary LDPC codes," in *Proc. IEEE Inf. Theory Appl. Workshop*, Feb. 2011, pp. 1–7.

[33] C. C. Huang, C. J. Wu, C.-Y. Chen, and C. C. Chao, "Parallel symbol-flipping decoding for non-binary LDPC codes," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1228–1231, Jun. 2013.

[34] Q. Huang and S. Yuan, "Bit reliability-based decoders for non-binary LDPC codes," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 38–48, Jan. 2016.

[35] W. Ullah, L. Cheng, and F. Takawira, "Prediction and voting based symbol flipping non-binary LDPC decoding algorithms," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–6.

[36] W. Ullah, L. Cheng, and F. Takawira, "Low complexity bit reliability and predication based symbol value selection decoding algorithms for non-binary LDPC codes," *IEEE Access*, vol. 8, pp. 142691–142703, 2020.

[37] C.-Y. Chen, Q. Huang, C.-C. Chao, and S. Lin, "Two low-complexity reliability-based message-passing algorithms for decoding non-binary LDPC codes," *IEEE Trans. Commun.*, vol. 58, no. 11, pp. 3140–3147, Nov. 2010.

[38] Q. Huang, M. Zhang, Z. Wang, and L. Wang, "Bit-reliability based low-complexity decoding algorithms for non-binary LDPC codes," *IEEE Trans. Commun.*, vol. 62, no. 12, pp. 4230–4240, Dec. 2014.

[39] B. Liu, J. Gao, G. Dou, and W. Tao, "Weighted symbol-flipping decoding for nonbinary LDPC codes," in *Proc. 2nd Int. Conf. Netw. Secur., Wireless Commun. Trusted Comput.*, vol. 1, Apr. 2010, pp. 223–226.

[40] N.-Q. Nhan, T. M. N. Ngatched, O. A. Dobre, P. Rostaing, K. Amis, and E. Radoi, "Multiple-votes parallel symbol-flipping decoding algorithm for non-binary LDPC codes," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 905–908, Jun. 2015.

[41] D. Zhao, X. Ma, C. Chen, and B. Bai, "A low complexity decoding algorithm for majority-logic decodable nonbinary LDPC codes," *IEEE Commun. Lett.*, vol. 14, no. 11, pp. 1062–1064, Nov. 2010.
[42] F. Garcia-Herrero, D. Declercq, and J. Valls, "Non-binary LDPC decoder based on symbol flipping with multiple votes," *IEEE Commun. Lett.*, vol. 18, no. 5, pp. 749–752, May 2014.
[43] S. Wang, Q. Huang, and Z. Wang, "Symbol flipping decoding algorithms based on prediction for non-binary LDPC codes," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 1913–1924, May 2017.
[44] J. Oh, S. Han, and J. Ha, "An improved symbol-flipping algorithm for nonbinary LDPC codes and its application to NAND flash memory," *IEEE Trans. Magn.*, vol. 55, no. 9, pp. 1–13, Sep. 2019.
[45] W. Ullah, T. Jiang, F. Yang, and S. M. Aziz, "Two-way normalization of min-sum decoding algorithm for medium and short length low density parity check codes," in *Proc. 7th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2011, pp. 1–5.
[46] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406–414, Mar. 2002.
[47] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.
[48] P. Schläfer, N. Wehn, M. Alles, T. Lehnigk-Emden, and E. Boutillon, "Syndrome based check node processing of high order NB-LDPC decoders," in *Proc. 22nd Int. Conf. Telecommun. (ICT)*, Apr. 2015, pp. 156–162.
[49] X. Wang, B. Bai, and X. Ma, "A low-complexity joint detection-decoding algorithm for nonbinary LDPC-coded modulation systems," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 794–798.
[50] S. Zhao, X. Wang, T. Wang, B. Bai, and X. Ma, "Joint detection–decoding of majority-logic decodable non-binary low-density parity-check coded modulation systems: An iterative noise reduction algorithm," *IET Commun.*, vol. 8, no. 10, pp. 1810–1819, Jul. 2014.
[51] W. Ullah, L. Cheng, and F. Takawira, "Predictive syndrome based low complexity joint iterative detection-decoding algorithm for non-binary LDPC codes," *IEEE Access*, vol. 9, pp. 33464–33477, 2021.
[52] W. Ullah, D. N. K. Jayakody, J. Li, and Y. Chursin, "Iterative joint detection-decoding algorithms using Euclidean distance-based feedback," in *Proc. 10th Int. Conf. Inf. Autom. Sustainability (ICIAfS)*, Aug. 2021, pp. 19–24.
[53] J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209–213, Feb. 2005.
[54] Z. Chuan-Gang, Y. Jin-Sheng, L. Xue-Hong, and L. Jia-Ru, "Improvement of shuffled iterative decoding," in *Proc. IEEE Inf. Theory Workshop-ITW Chengdu*, Oct. 2006, pp. 114–116.
[55] A. Manada, K. Yoshida, H. Morita, and R. Tatasukawa, "A grouping based on local girths for the group shuffled belief propagation decoding," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2133–2136, Nov. 2016.
[56] W. Ullah, L. Cheng, and F. Takawira, "Adaptive group shuffled symbol flipping decoding algorithm," *IEEE Access*, vol. 12, pp. 53805–53817, 2024.
[57] W. Ullah, "Symbol flipping decoding of non-binary low density parity check codes," Ph.D. dissertation, University of the Witwatersrand, Johannesburg, South Africa, 2022.
[58] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.
[59] S. T. Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.
[60] M. Lentmaier, A. Sridharan, D. J. Costello, and K. Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.
[61] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems*. Accessed: Aug. 2023. [Online]. Available: https://standards.ieee.org/ieee/802.16/4184/
[62] W. Ullah, F. Yang, and D. N. K. Jayakody, "OTFS modulated massive MIMO with 5G NR LDPC coding: Trends, challenges and future directions," *Comput. Netw.*, vol. 254, Dec. 2024, Art. no. 110751.
[63] S. Zhao, X. Ma, X. Zhang, and B. Bai, "A class of nonbinary LDPC codes with fast encoding and decoding algorithms," *IEEE Trans. Commun.*, vol. 61, no. 1, pp. 1–6, Jan. 2013.
[64] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri, "Non-binary protograph-based LDPC codes: Enumerators, analysis, and designs," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3913–3941, Jul. 2014.
[65] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Prog. Rep.*, vol. 42, no. 154, pp. 45–154, Aug. 2003.
[66] S. Zhao, X. Huang, and X. Ma, "Structural analysis of array-based non-binary LDPC codes," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4910–4922, Dec. 2016.
[67] S. Zhao and X. Ma, "Construction of high-performance array-based non-binary LDPC codes with moderate rates," *IEEE Commun. Lett.*, vol. 20, no. 1, pp. 13–16, Jan. 2016.
[68] M. C. Davey and D. J. C. MacKay, "Monte Carlo simulations of infinite low density parity check codes over $GF(q)$," in *Proc. Int. Workshop Optim. Codes Rel. Topics*, 1998, pp. 9–15.
[69] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-domain decoding of LDPC codes over $GF(q)$," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2004, pp. 772–776.
[70] L. J. Arnone, C. A. Gayoso, C. M. Gonzalez, M. R. Rabini, J. C. Moreira, and P. G. Farrell, "Fast Fourier transform simplified soft-distance decoding algorithm for decoding non-binary LDPC codes," in *Proc. Argentine Conf. Micro-Nanoelectronics, Technol. Appl. (EAMTA)*, Jul. 2014, pp. 13–17.
[71] P. G. Farrell, L. J. Arnone, and J. C. Moreira, "Euclidean distance soft-input soft-output decoding algorithm for low-density parity-check codes," *IET Commun.*, vol. 5, no. 16, pp. 2364–2370, Nov. 2011.
[72] L. J. Arnone, J. Castiñeira Moreira, and P. G. Farrell, "Field programmable gate arrays implementations of low complexity soft-input soft-output low-density parity-check decoders," *IET Commun.*, vol. 6, no. 12, pp. 1670–1675, Aug. 2012.
[73] M. Zhu, Q. Guo, B. Bai, and X. Ma, "Reliability-based joint detection-decoding algorithm for nonbinary LDPC-coded modulation systems," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 2–14, Jan. 2016.
[74] O. Abassi, L. Conde-Canencia, A. Al Ghouwayel, and E. Boutillon, "A novel architecture for elementary-check-node processing in nonbinary LDPC decoders," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 2, pp. 136–140, Feb. 2017.
[75] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.
[76] T. Bai, H. Zhang, J. Wang, C. Xu, M. Elkashlan, A. Nallanathan, and L. Hanzo, "Fifty years of noise modeling and mitigation in power-line communications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 41–69, 1st Quart., 2021.
[77] H. Meng, Y. L. Guan, and S. Chen, "Modeling and analysis of noise effects on broadband power-line communications," *IEEE Trans. Power Del.*, vol. 20, no. 2, pp. 630–637, Apr. 2005.
[78] A. Al-Kinani, C.-X. Wang, L. Zhou, and W. Zhang, "Optical wireless communication channel measurements and models," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1939–1962, 3rd Quart., 2018.
[79] S. Yahia, Y. Meraihi, A. Ramdane-Cherif, A. B. Gabis, D. Acheli, and H. Guan, "A survey of channel modeling techniques for visible light communications," *J. Netw. Comput. Appl.*, vol. 194, Nov. 2021, Art. no. 103206.

WAHEED ULLAH (Senior Member, IEEE) received the B.Sc. degree in electrical and electronics from the University of Engineering and Technology Peshawar, Pakistan, the master's degree in communication and information systems from Nanjing University of Aeronautics and Astronautics, China, in 2012, and the Ph.D. degree from the University of the Witwatersrand, South Africa, in 2022. He has authored several papers that have been published in highly regarded journals and reputable conferences. His research interest includes wireless communication systems and networks. His current research interests include LDPC decoding algorithms, SWIPT, coding based secure communication, massive MIMO, and RIS with RSMA and OTFS modulation for 6G communication. He is the Chair of the Industrial Coordination, IEEE Islamabad Section.

FENGFAN YANG received the B.Sc. degree in electronic engineering from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1990, the M.Sc. degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 1993, and the Ph.D. degree in electronic engineering from Southeast University, Nanjing, in 1997. Since May 1997, he has been with the College of Information Science and Technology, NUAA. From October 1999 to May 2003, he was a Research Associate with the Centre for Communication Systems Research, University of Surrey, Guildford, U.K., and the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada. His major research interests include information theory and channel coding, especially at iteratively decodable codes, such as turbo codes and LDPC codes and their applications for mobile and satellite communications.

KWONHUE CHOI (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic and electrical engineering from Pohang University of Science and Technology, Pohang, South Korea, in 1994, 1996, and 2000, respectively. From 2000 to 2003, he was with the Electronics and Telecommunications Research Institute, Daejeon, South Korea, as a Senior Research Staff Member. In 2003, he joined Yeungnam University, Gyeongsan, South Korea, where he is currently a Professor with the School of Computer Science and Engineering. He has authored a textbook *Problem-Based Learning in Communication Systems Using MATLAB and Simulink* (Wiley, 2016). His research interests include signal design for communication systems, multiple access schemes, diversity schemes for wireless fading channels, and high-mobility communication systems.

DUSHANTHA NALIN K. JAYAKODY (Senior Member, IEEE) received the M.Sc. degree in electronics and communications engineering from Eastern Mediterranean University, Türkiye, and the Ph.D. degree in electronics and communications engineering from University College Dublin, Ireland, in 2014. He is a Professor with Lusófona University, Portugal, and the Faculty of Engineering, Sri Lanka Institute of Information Technology, Sri Lanka. He has held visiting and/or sabbatical positions with the Center for Telecommunications Research, University of Sydney, Australia, in 2015, and Texas A&M University, USA, in 2018. He was a Visiting Professor with the University of Jyväskylä, Finland, from 2019 to 2022, under the framework of the Academy of Finland. He was also a Visiting Professor with the University of Juiz de Fora, Brazil, in 2019. From 2019 to 2022 and again from 2024 to 2026, he was a SPARC Professor with the Department of Electronics and Communication Engineering, National Institute of Technology, Tiruchirappalli, India, under the framework of the Ministry of Human Resource Development, India. From 2014 to 2016, he was a Postdoctoral Research Fellow with the University of Tartu, Estonia, and the University of Bergen, Norway. From 2016 to 2021, he was a Professor with the School of Computer Science and Robotics, National Research Tomsk Polytechnic University (TPU), Russia. From 2019 to 2021, he was the Dean of the School of Postgraduate Studies and Research, Sri Lanka Technological Campus, Sri Lanka, where he was also the Director of Postgraduate, Research, and Impact, from 2021 to 2022. He was also associated with the Department of Engineering and Computer Science/Autónoma TechLab, Universidade Autónoma de Lisboa, Portugal, from 2021 to 2022. He has supervised over ten Ph.D. students, several master's students, more than 50 undergraduate students, and five postdoctoral researchers. Throughout his career, he has secured nearly 6 million in research funding from various international grant agencies and published over 230 peer-reviewed journal articles, conference papers, and books. He is a fellow of IET. In 2021, he received the Sri Lankan Presidential Award for his outstanding research performance, among many other awards and recognitions. He has received several prestigious awards, including the Best Paper Award at IEEE ICCMIT 2017, ETIC 2019, ICARC 2024, ICAC 2024, and IRC 2023; the Education Leadership Award from the World Academic Congress, in July 2019; the Outstanding Faculty Award from National Research Tomsk Polytechnic University, Russia, in 2017 and 2018. He was ranked among the top 2% of scientists in the world in 2021, 2022, and 2023, according to the Stanford-Elsevier list. He was recognized as the Distinguished Researcher in Wireless Communications in Chennai, India, in 2019. He was also received various awards at university level from his previous employers.