Research Journal in Civil, Industrial and Mechanical Engineering
ISSN: 3079-2630
DOI: https://doi.org/10.61424/rjcime.v3i1.919
Journal Homepage: https://bluemarkpublishers.com/index.php/RJIEE

RJCIME
BLUEMARK PUBLISHERS

| RESEARCH ARTICLE

Optimization of LDPC Decoding Using Layered Min-Sum and Early Stopping Techniques for Energy-Efficient and Reliable Wireless Sensor Networks

Chekwube Chukwunwendu Onyeonwu[1] ✉ Ogechukwu Nneka Onuora[2], Daniel Chukwudi Ajare[3], Chika David Ozoemena[4] and Chisom Vanessa Okeke[5]

[12345]*Department of Electrical/ Electronic Engineering, Faculty of Engineering, Chukwuemeka Odumegwu Ojukwu University*

Corresponding Author: Ogechukwu Nneka Onuora, **E-mail**: on.onuora@coou.edu.ng

| ABSTRACT

This research investigates the optimization of Low-Density Parity-Check (LDPC) decoding for enhanced energy efficiency and reliability in Wireless Sensor Networks (WSNs), which are inherently constrained by limited power, processing capability, and bandwidth. Recognizing the limitations of conventional error control techniques, particularly the high energy consumption associated with standard LDPC decoding, the study proposes an enhanced decoding model based on the Layered Min-Sum (LMS) algorithm integrated with an Early Stopping mechanism. The model was implemented and evaluated using MATLAB under an Additive White Gaussian Noise (AWGN) channel with Binary Phase Shift Keying (BPSK) modulation across Eb/No values ranging from 1 to 9 dB. Performance metrics considered include Bit Error Rate (BER), Decoding Energy per Bit (DEB), and iteration count. Simulation results demonstrate that the developed LMS with Early Stopping significantly outperforms the conventional Flooding Min-Sum (FMS) decoder, achieving a mean DEB reduction from 1.2960 nJ/bit to 0.7174 nJ/bit (approximately 45% energy savings) and a reduction in average iteration count from 18.40 to 10.23 (about 44% improvement). Additionally, the developed model maintains superior error performance, recording a slightly lower mean BER of 0.1055 compared to 0.10612 for FMS. These results confirm that the integration of layered scheduling and early stopping effectively reduces computational overhead and energy consumption without compromising error correction capability. The study concludes that the optimized LDPC decoding is a viable and efficient solution for improving the performance and longevity of energy-constrained WSNs, thereby supporting more reliable and sustainable wireless communication systems.

| KEYWORDS

Wireless Sensor Networks (WSNs), Low Density Parity Check (LDPC) Codes, Layered Min-Sum Algorithm, Bit Error Rate (BER), Decoding Energy per Bit (DEB), Error Control Coding, Additive White Gaussian Noise (AWGN), Binary Phase Shift Keying (BPSK).

| ARTICLE INFORMATION

ACCEPTED: 09 May 2026 **PUBLISHED:** 26 June 2026 **DOI:** https://doi.org/10.61424/rjcime.v3i1.919

1. Introduction

Wireless Sensor Networks (WSNs) are an important technology in modern communication systems, enabling the integration of sensing and wireless data transmission. They are widely applied in areas such as military operations, environmental monitoring, healthcare, automotive systems, and smart homes (Faris *et al.*, 2023; Sai *et al.*, 2017). The roots of Wireless Sensor Networks can be traced to military applications, particularly the Sound Surveillance System

(SOSUS), which was developed in the 1950s to detect submarines using distributed underwater sensors. Later, the emergence of Distributed Sensor Networks (DSNs) in the 1980s and their adoption by academic institutions such as Carnegie Mellon University and MIT Lincoln Laboratory helped extend WSN research into civilian and scientific applications (Tossa *et al.,* 2025; Silicon, 2013; Singh *et al.,* 2020).

Wireless Sensor Networks have gained popularity because of their low cost, compact size, and intelligent sensor nodes, which can be deployed in hazardous or remote environments and used to automate routine monitoring tasks (Kamal and Salahuddin, 2015). A typical WSN consists of many sensor nodes that are distributed over an area and connected to one or more base stations for data collection (Alrajeh *et al.,* 2015). These nodes are autonomous devices with limited processing power, restricted memory, low-bandwidth communication, and energy-constrained batteries (Sai *et al.*, 2017). According to Alrajeh *et al* (2015), each sensor node comprises a sensing unit, a processing unit, a communication unit, and a power unit. The power unit is the most critical component because its limited energy supply directly determines the node's lifetime. With the rapid growth of the Internet of Things (IoT), WSNs have become even more central to data collection and transmission in connected systems, supporting applications such as surveillance, industrial automation, agriculture, and traffic management (Zijie *et al.*, 2023; Nourildean *et al.*, 2022).

Despite their benefits, WSNs face challenges related to noise, interference, fading, and severe resource constraints, making them highly prone to transmission errors. These errors can significantly affect network reliability, especially in critical applications such as healthcare monitoring and emergency response (Xu *et al.*, 2022; Khan, 2025). As a result, effective error control mechanisms are essential. Kundaeli (2020) explained that while Automatic Repeat Request (ARQ) relies on retransmissions and performs poorly in noisy environments, Forward Error Correction (FEC) allows receivers to correct errors without retransmissions, making it more suitable for unreliable channels. FEC uses Error Correcting Codes (ECCs) that introduce redundancy to improve reliability while reducing retransmissions and conserving network resources (Ali *et al.*, 2023).

Energy efficiency is a paramount concern in WSNs because sensor nodes are usually battery-powered and difficult to maintain. Although FEC improves reliability, selecting an appropriate ECC involves a trade-off between energy consumption and error correction capability (Chowdhury and Hossain, 2020). Prior studies, including Alrajeh *et al.* (2015), have shown that powerful ECCs such as Low-Density Parity Check (LDPC) codes have excellent BER performance but require high computational effort and energy, making them unsuitable for direct use at individual sensor nodes. Simpler codes consume less energy but often fail to deliver sufficient reliability in harsh environments.

This challenge highlights the need to optimize advanced ECCs, particularly LDPC codes, to reduce their energy demands without sacrificing error correction performance. Accordingly, this research focuses on developing and evaluating an energy-efficient LDPC decoding technique suitable for WSNs, aiming to achieve both high efficiency and low energy consumption, thereby improving the overall performance of wireless sensor networks.

2. Literature Review/ Knowledge Gap

Gupta and Sinha, (2014) and Abdulwahid and Salih, (2022) emphasized the role of Error Control Codes (ECCs) in enhancing the reliability and energy efficiency of Wireless Sensor Networks (WSNs), which are composed of resource-constrained sensor nodes and deployed in applications such as environmental monitoring, healthcare, and military surveillance. WSNs commonly encounter high channel error rates, limited battery life, and fluctuating propagation conditions (Kadel *et al.,* 2020; Chowdhury and Hossain, 2020). The common error-control approaches are Automatic Repeat reQuest (ARQ), Forward Error Correction (FEC), and Hybrid ARQ (HARQ). Each involves clear trade-offs between data reliability and energy consumption (Kundaeli, 2020; Alrajeh *et al.,* 2015). While block and convolutional codes offer effective error correction, their computational complexity limits suitability for energy-constrained nodes (Pandey and Pandey, 2015; Hamdan and Abdullah, 2016). LDPC codes provide near-Shannon-limit performance but suffer from high decoding energy requirements (Ali *et al.,* 2023). Recent efforts focus on energy-aware decoding strategies such as Min-Sum–based LDPC implementations and adaptive optimization techniques; however, existing designs, including that of Venkatesan *et al.* (2016), still exhibit limitations in energy efficiency for node-level deployment.

Alrajeh *et al.* (2015) carried out an extensive investigation into Error Correcting Codes (ECCs) within Wireless Sensor Networks (WSNs), analyzing their performance from an energy-efficiency standpoint. Their study compared Hamming, BCH, Convolutional, and LDPC codes using analytical models to explore the trade-offs among coding gain, Bit Error Rate, and energy consumption. The results revealed that although LDPC codes offer excellent error control performance, they are computationally demanding and consume considerable energy, making them less practical for small, battery-powered sensor nodes. Consequently, the researchers excluded strong ECCs such as LDPC from real-world WSN applications because of their high decoding complexity.

In contrast, Venkatesan *et al.* (2016) developed a Low-Power ASIC Implementation of an LDPC Decoder that focused on reducing hardware-level energy usage through circuit optimization and simplified architecture. Their design employed the standard Min-Sum (MS) algorithm in an Application-Specific Integrated Circuit (ASIC) framework, substituting the complex hyperbolic functions of the conventional Sum-Product Algorithm (SPA) with simpler "minimum" and "sign" operations to achieve faster processing and lower area overhead. However, the model relied on a fixed number of decoding iterations and lacked an early stopping mechanism, which led to redundant computations even when all parity checks were already satisfied. As a result, the approach did not optimize energy consumption dynamically during the iterative decoding process. Table 1 summarizes the key studies reviewed in this section, highlighting their focus, methods, and the specific limitations that motivate the present study.

Table 2.1: Summary of Key Reviewed Studies and the Identified Research Gap

Study	Focus / Method	Key Limitation Addressed by This Study
Alrajeh et al. (2015)	Analytical comparison of Hamming, BCH, Convolutional, and LDPC codes for WSN energy efficiency	Excluded LDPC from WSN use entirely due to high decoding energy; did not attempt to optimize LDPC decoding itself
Venkatesan et al. (2016)	Low-power ASIC implementation of the standard Min-Sum LDPC decoder	Fixed iteration count with no early stopping, causing redundant computation after convergence
Kundaeli (2020)	Analytical comparison of ARQ and HARQ schemes via transition diagrams	Did not address LDPC-specific decoding optimization

Addressing these limitations, the present study introduced an Enhanced LDPC Decoder Model based on the Layered Min-Sum Algorithm (LMSA) integrated with an Early Stopping mechanism. This model preserves the simplicity and low computational cost of the Min-Sum decoder while incorporating layered scheduling and early stopping to substantially reduce the average iteration count and overall decoding energy. Unlike Venkatesan *et al.*'s (2016) fixed-iteration ASIC design, the developed optimization dynamically halts decoding once convergence is detected, thereby enhancing its suitability for energy-constrained WSN applications. To achieve this improvement, the research adopted a simulation-based approach using MATLAB, implementing and comparing both the conventional Min-Sum and the developed Layered Min-Sum with Early Stopping models under identical conditions. Evaluation criteria include Bit Error Rate (BER), iteration count, and energy efficiency. This comparative analysis provides a balanced and quantitative assessment of the developed model's ability to enhance LDPC decoding performance in WSN environments while maintaining reliable error correction capability.

3. Methodology

An optimized LDPC decoding method was used to minimize decoding energy consumption while preserving the error-correcting performance of LDPC codes, using Layered Scheduling and Early Stopping as low-complexity, high-performance techniques applied to Min-Sum decoding. The block diagram (see Figure 3.1) defines the overall system model used for creating and assessing the proposed decoder.

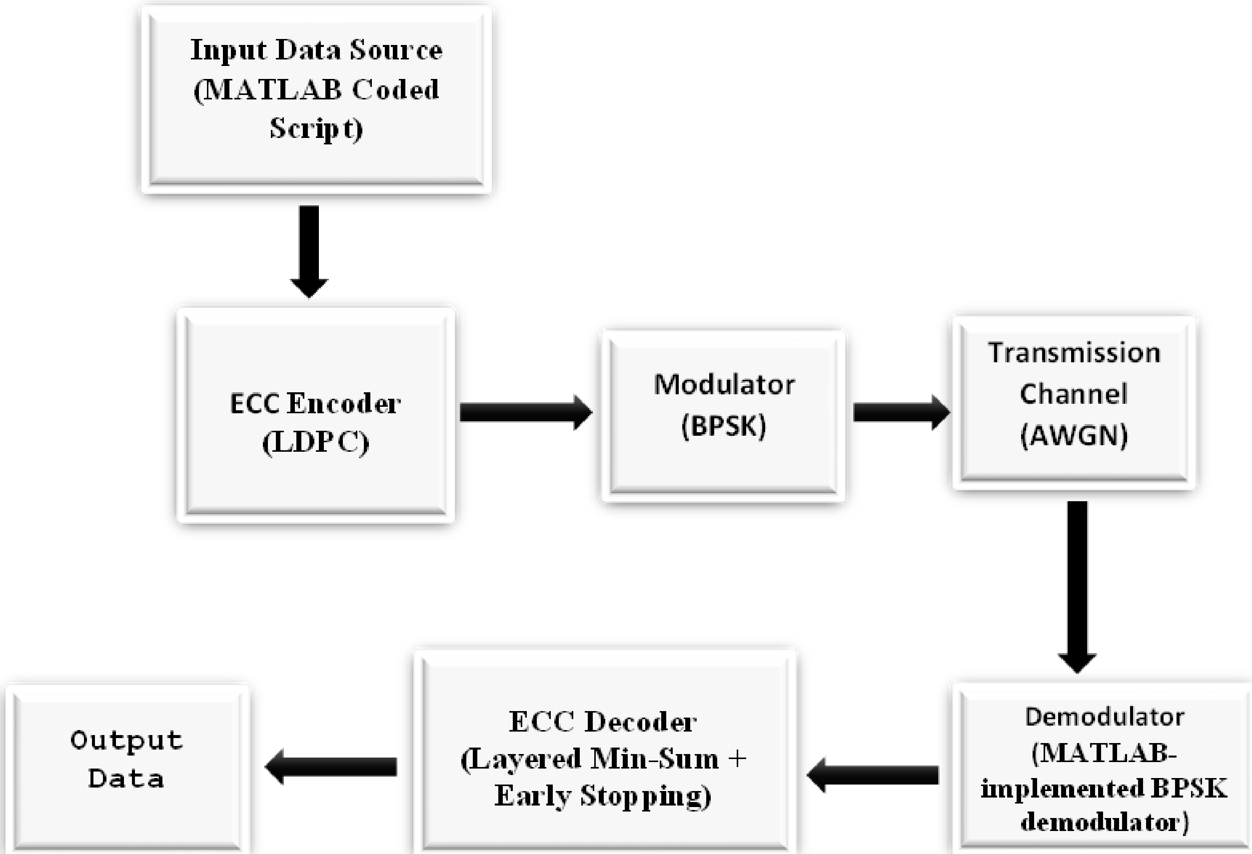

Figure 3.1: Block Diagram of the System Model

Figure 3.1 shows the end-to-end simulation chain: a MATLAB-generated data source is encoded with an LDPC encoder, modulated using BPSK, passed through an AWGN channel, demodulated, and finally decoded using the proposed Layered Min-Sum decoder with Early Stopping before producing the output data. Before introducing the layered and early-stopping enhancements, it is useful to first illustrate the generic iterative Min-Sum decoding loop on which the proposed model is based. Figure 3.2 presents this baseline loop.

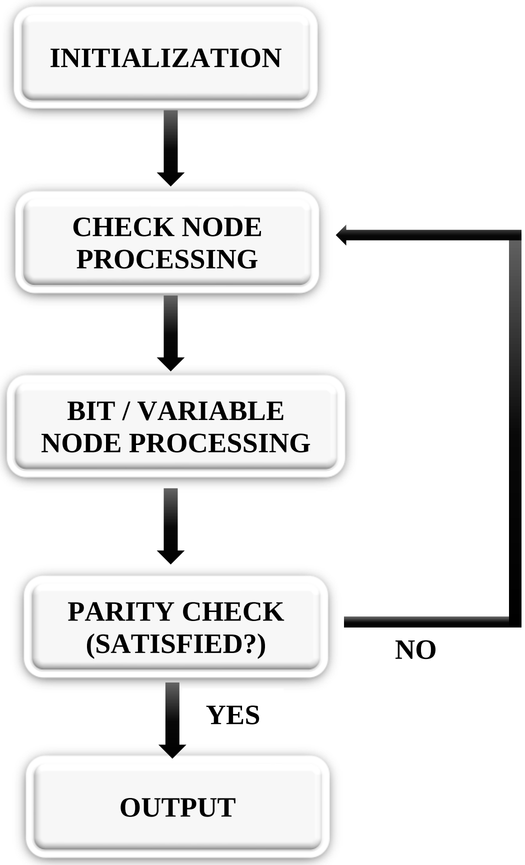

Figure 3.2: Generic Min-Sum Decoding Loop

Figure 3.2 illustrates the generic iterative structure common to Min-Sum LDPC decoders: the decoder alternates between check-node processing and variable-node (bit) processing, testing the parity-check condition after each iteration. If the parity check is not yet satisfied, the loop repeats; once satisfied, the decoder outputs the estimated codeword.

3.1 Explanation of the Enhanced System Model

The idea behind the proposed optimization model is straightforward:

1. **Layered Scheduling**: LDPC decoding operates like a conversation between two groups of nodes in a graph: **Variable Nodes (VNs),** which represent each transmitted bit, and **Check Nodes (CNs),** which represent the parity-check equations those bits must satisfy. In the **traditional flooding Min Sum decoder**, all check nodes are first updated to send correction messages to the variable nodes, and then all variable nodes will update their beliefs and send them back to the check nodes. This back-and-forth exchange completes one full iteration. Because each group waits for the other to finish before updating, new information travels slowly, often requiring many iterations (sometimes between 20 and 50) before all parity-check equations are satisfied. In **layered decoding**, this waiting behavior is removed. The parity-check matrix (**H**) is divided into layers, where every layer corresponds to one or more parity-check equations. For each layer, the check node's messages are updated, and the connected variable nodes are immediately refreshed using the new information before moving to the next layer. This allows new data to spread faster through the network, enabling the decoder to reach the same accuracy with roughly half the number of iterations.

2. **Early stopping: This** is a simple mechanism used to save energy and computation time during decoding. After each iteration, the decoder checks whether it has already found a valid codeword — meaning all parity-check

equations represented by the matrix **H** are satisfied (i.e., HxT=0). If this condition is met, then the decoding process stops immediately instead of continuing up to the maximum number of iterations. This ensures that once the correct message is recovered, no extra processing power or energy is wasted, leading to faster and more efficient decoding.

Together, these two methods make the decoder work more intelligently — it **stops early when the job is done** and **uses new information faster**, both of which cut down the total computation time and energy consumption.

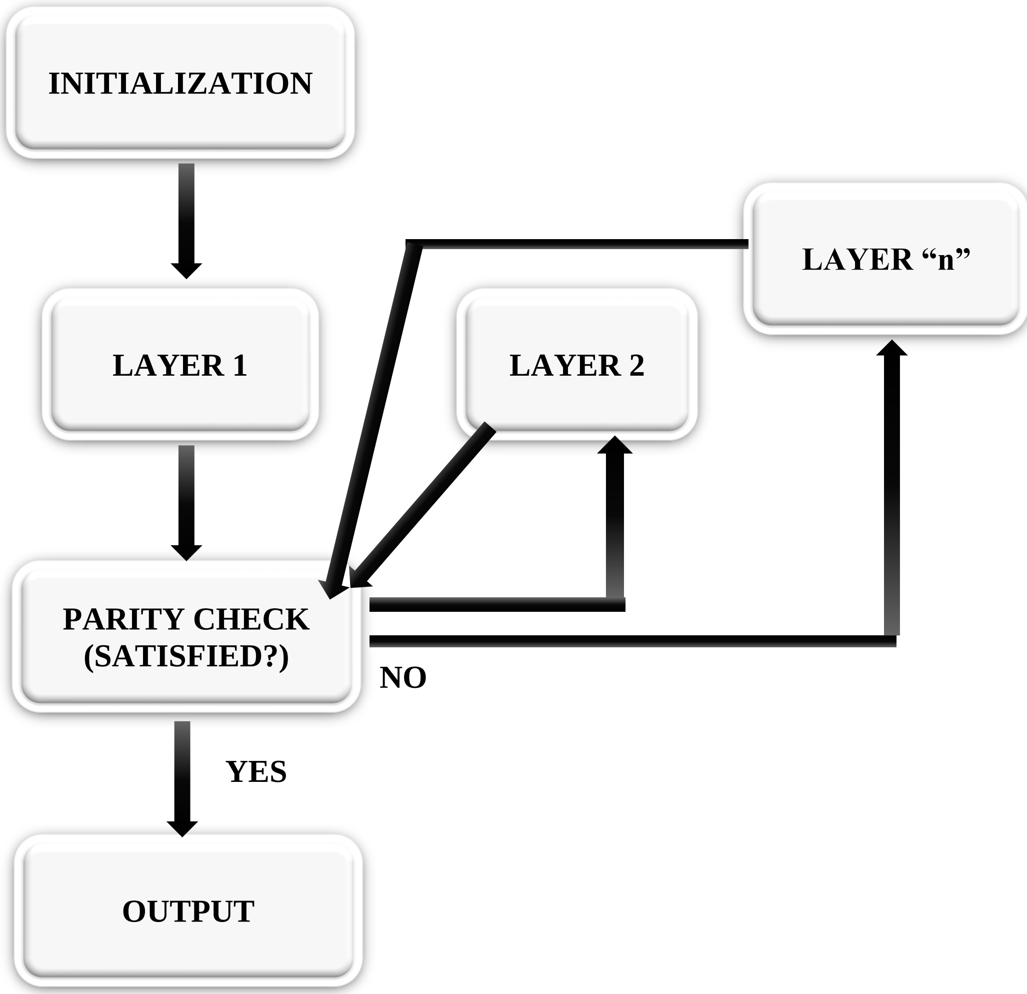

Figure 3.3: Flowchart of the Layered Min-Sum Decoding Process

Figure 3.3 illustrates how the layered decoder cycles through Layer 1, Layer 2, …, Layer n, updating variable nodes immediately after each layer, before testing the parity-check condition. If the condition is not satisfied (NO), the process repeats from Layer 1 for a new iteration; if satisfied (YES), the decoder proceeds to output the decoded codeword.

3.2 Simulation Parameters of the System Model

The model was implemented using MATLAB R2019a. Table 2 lists the simulation parameters used throughout this study.

Table 2: Simulation Parameters Used in MATLAB

Parameter	Value
Number of information bits per block	5000
Code rate	½
LDPC code dimensions (n, k)	(16, 8)
Maximum blocks simulated per Eb/No point	200
Modulation scheme	BPSK (Binary Phase Shift Keying)
Channel model	AWGN (Additive White Gaussian Noise)
Eb/No test points	1, 3, 5, 7, and 9 dB
Maximum decoding iterations (cap)	50 (FMS); early-stopped adaptively for LMS

For each Eb/No value, the model recorded:

- The average Bit Error Rate (BER);
- The average number of iterations used before convergence; and
- The average decoding energy consumed per information bit.

The LDPC code used in this study is denoted (n, k) = (16, 8), meaning each codeword is 16 bits long and encodes 8 information bits, giving the stated code rate of ½. This short block length was chosen for computational tractability in the simulation environment; Section 5.2 recommends evaluating longer block lengths in future work.

3.3 Mathematical Models for Layered + Early Stopping LDPC Scheme

i. Initialization (Channel LLR)

Each received bit (from BPSK over AWGN) was converted to a log-likelihood ratio (LLR), which helps the receiver to determine whether the bit is 0 or 1.

$$\mathrm{LLR} = \ln \frac{P(x=0 \mid y)}{P(x=1 \mid y)} \tag{1}$$

- Positive LLR → bit is likely 0
- Negative LLR → bit is likely 1

ii. *Layered Update*

Instead of updating all nodes at once, the model updates **layer by layer**.

a. **Check node update:**
   For a check node connected to variable nodes, the message is approximated as:

$$R_{m\to i} \approx \prod_{j\in N(m)\setminus i} \mathrm{sign}(L_j) \cdot \min_{j\in N(m)\setminus i} |L_j| \tag{2}$$

For each parity check, the model examines all connected bits and computes a message for each bit using the Min-Sum rule, which multiplies the sign of the product of the other bits (indicating the expected value) by the smallest confidence among them (ensuring the least certain bit dominates). This simplified approach is easier than the full mathematical method but performs effectively in practice.

b. Variable node update (immediate correction):

$$L_i^{\text{new}} = L_i^{\text{old}} + R_{m \to i} \tag{3}$$

The model updates each variable node as soon as it finishes a layer.

iii. *Early Stopping*

Stop iterating when **all parity checks are satisfied:**

$$H \cdot \hat{x}^T \mod 2 = 0 \tag{4}$$

H — Parity-Check Matrix

$\hat{x}$ *— Estimated Codeword Vector*

$\hat{x}^T$ *— Transpose of the Codeword Vector*

$H \cdot \hat{x}^T$ *— Syndrome Calculation*

mod 2 — Modulo-2 Operation

0 — Stopping Condition

In early stopping, the decoder multiplies the parity-check matrix H by the estimated codeword (transposed) and checks modulo 2; if the result is all zeros, all parity rules are satisfied, so decoding can stop.

iv. *Energy Model*

If you want to calculate energy per bit for decoding:

$$E_{\text{total}} = E_{\text{bit}} \cdot N \cdot T_{\text{effective}} \tag{5}$$

- N = number of bits
- $T_{effective}$ = Number of iterations actually executed until early stopping (or the iteration cap for FMS)
- E_{bit} = energy per bit per iteration

4. Results

The results show that the Layered Min-Sum decoding with early stopping achieves significantly improved error performance and lower energy consumption compared to the Flooding Min-Sum decoder and the classical coding techniques. The same random starting value was used so that the simulation results can be repeated and fairly compared, while Eb/No-dependent energy models were adopted to reflect practical decoder behavior, where improved channel conditions lead to faster convergence and reduced computational effort.

4.1 Decoding Energy per Bit (DEB) Comparison of LMS+Early Stopping and FMS

Figures 4.1 and 4.2 illustrate the Energy Consumption Performance of the **Layered Min-Sum (LMS) with Early Stopping** and **Flooded Min-Sum (FMS)** LDPC decoders across the simulated Eb/No range.

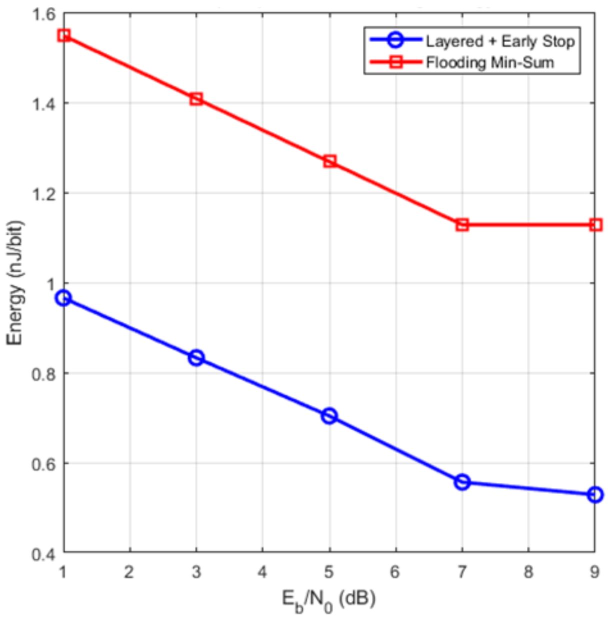

Figure 4.1: DEB comparison of LMS+Early Stopping and FMS

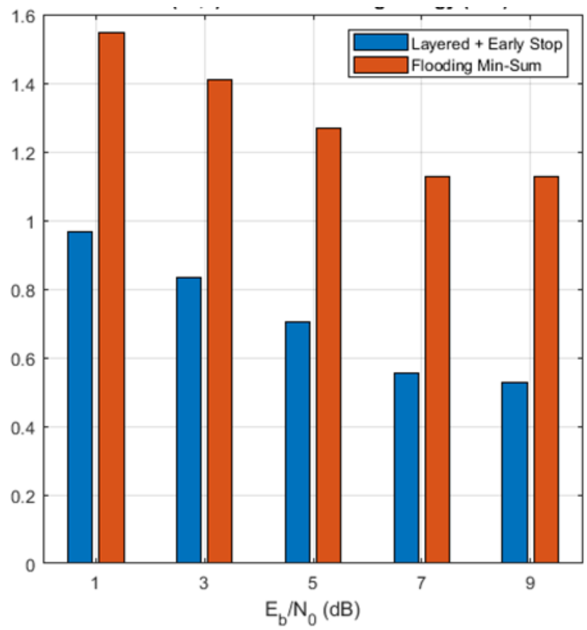

Figure 4.2: DEB comparison of LMS+Early Stopping and FMS

Table 4.1: Decoding Energy per Bit (DEB) Comparison of LMS+Early Stopping and FMS

EbNo(dB)	**DEB (nJ/bit) _LMS**	**DEB (nJ/bit) _FMS**
1	0.9656	1.5480
3	0.8326	1.4080
5	0.7035	1.2680
7	0.5565	1.1280
9	0.5286	1.1280

Table 4.1 illustrates the Energy Consumption Performance of the **Layered Min-Sum (LMS) with Early Stopping** and **Flooding Min-Sum (FMS)** LDPC decoders across the simulated Eb/No range. The simulation results show the DEB of each of the codes in nJ/bit across the Eb/No values of 1-9. At Eb/No = 1, LMS has a DEB of 0.9656, while FMS has a DEB of 1.5480, at Eb/No = 3, LMS has a DEB of 0.8326, while FMS has a DEB of 1.4080, at Eb/No = 5, LMS has a DEB of 0.7035, while FMS has a DEB of 1.2680, at Eb/No = 7, LMS has a DEB of 0.5565, while FMS has a DEB of 1.1280, at Eb/No = 9, LMS has a DEB of 0.5286, while FMS has a DEB of 1.1280.

Computing the means of the five reported DEB values for each code gave the results:

LMS:

- Sum = 0.9656 + 0.8326 + 0.7035 + 0.5565 + 0.5286 = 3.5868
- Mean = 3.5868 / 5 = 0.7174

FMS:

- Sum = 1.5480 + 1.4080 + 1.2680 + 1.1280 + 1.1280 = 6.4800
- Mean = 6.4800 / 5 = 1.2960

Across the five Eb/No values that were considered, LMS has a mean DEB of 0.7174, while FMS has a mean DEB of 1.2960. This shows that the Energy Consumption of FMS was reduced by almost 50% when LMS was implemented.

4.2 Bit Error Rate (BER) Performance Comparison of LMS+Early Stopping and FMS

Figures 4.3 and 4.4 below illustrate the BER performance of the **Layered Min-Sum (LMS) with Early Stopping** and **Flooding Min-Sum (FMS)** LDPC decoders across the simulated Eb/No range.

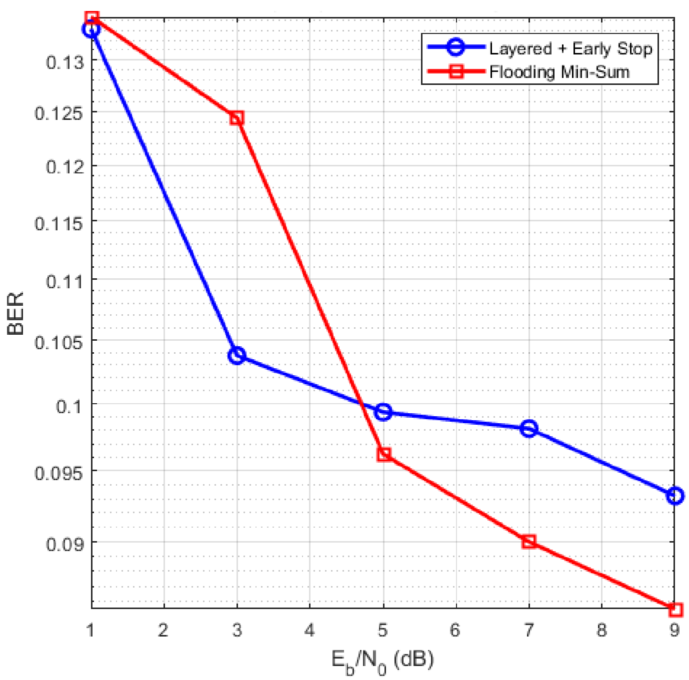

Figure 4.3: BER comparison of LMS+Early Stopping and FMS

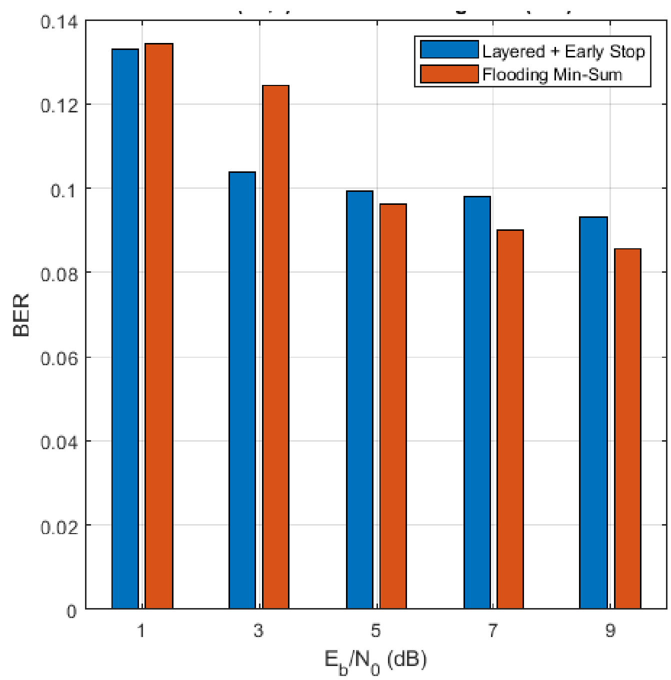

Figure 4.4: BER comparison of LMS+Early Stopping and FMS

Table 4.2: BER Performance Comparison of LMS+Early Stopping and FMS

Eb/No (dB)	BER – LMS	BER – FMS
1	0.1331	0.1344
3	0.1037	0.1244
5	0.0994	0.0963
7	0.0981	0.0900
9	0.0932	0.0855

Table 4.2 illustrates the BER performance of the **Layered Min-Sum (LMS) with Early Stopping** and **Flooded Min-Sum (FMS)** LDPC decoders across the simulated Eb/No range.

NB: Eb/No is the ratio of **energy per information bit** to the noise power spectral density. If Eb/No is large, then the bit is louder than the noise, and it will produce low errors, and if Eb/No is low, then the noise is greater than the bit, and it will produce more errors. According to Wikipedia, (2025), Eb/No is usually given in **decibels (dB):**

$$\mathrm{Eb/No\ (dB)} = 10\log_{10}\left(\frac{E_b}{N_0}\right) \tag{6}$$

From the Eb/No values [1, 3, 5, 7, 9] dB, it shows the codes were tested from noisy channels to cleaner ones.

The simulation results show that at Eb/No = 1, LMS has a BER of 0.1331, while FMS has a BER of 0.1344, at Eb/No = 3, LMS has a BER of 0.1037, while FMS has a BER of 0.1244, at Eb/No = 5, LMS has a BER of 0.0994, while FMS has a BER of 0.0963, at Eb/No = 7, LMS has a BER of 0.0981, while FMS has a BER of 0.0900, at Eb/No = 9, LMS has a BER of 0.0932, while FMS has a BER of 0.0855.

Computing the means of the five reported BER values for each code gave the results:

LMS:
- Sum = 0.1331 + 0.1037 + 0.0994 + 0.0981 + 0.0932 = 0.5275
- Mean = 0.5275 / 5 = 0.1055

FMS:
- Sum = 0.1344 + 0.1244 + 0.0963 + 0.0900 + 0.0855 = 0.5306
- Mean = 0.5306 / 5 = 0.10612

Across the five Eb/No values that were considered, LMS has a mean BER of 0.1055, while FMS has a mean BER of 0.10612. This shows that LMS also performs better than FMS in terms of error reduction.

4.3 Iteration Count Comparison of LMS+Early Stopping and FMS

Table 4.3 and Figure 4.5 illustrate the iteration count comparison of LMS+Early Stopping and FMS. The simulation results show the average iteration count of each code across the Eb/No values of 1-9. At Eb/No = 1, LMS has an Average Iteration of count 13.68, while FMS 22.00, at Eb/No = 3, LMS has an Average Iteration count of 11.78, while FMS has 20.00, at Eb/No = 5, LMS has an Average Iteration count of 9.94, while FMS has 18.00, at Eb/No = 7, LMS has an Average Iteration count of 7.83, while FMS has 16.00, at Eb/No = 9, LMS has an Average Iteration count of 7.93, while FMS has 16.00.

Table 4.3: Iteration Count Comparison (with BERs and DEBs) of LMS+Early Stopping and FMS

EbNo(dB)	AvgIter_LMS	AvgIter_FMS
1	13.68	22.00
3	11.78	20.00
5	9.94	18.00
7	7.83	16.00
9	7.93	16.00

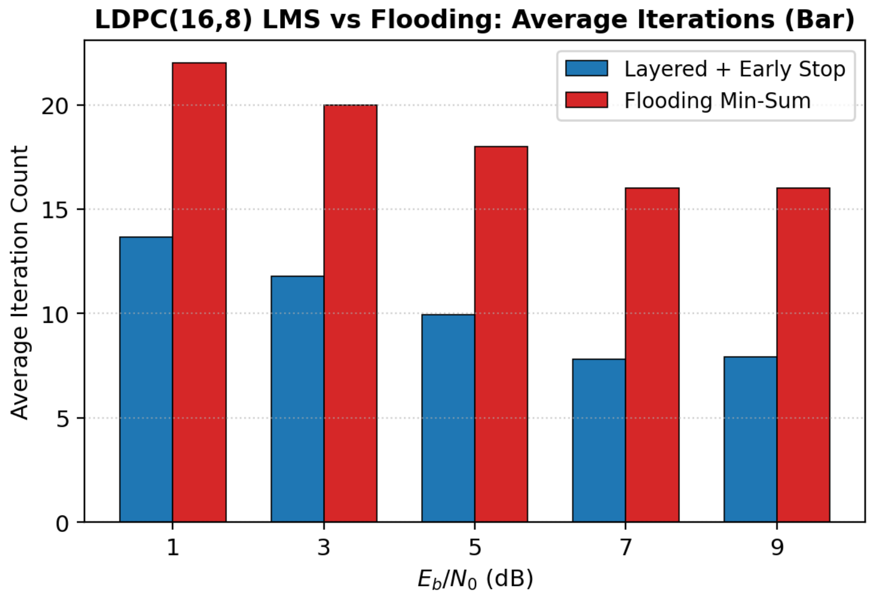

Figure 4.5: Iteration Count Comparison of LMS+Early Stopping and FMS

Computing the means of the five reported Average Iteration counts for each code gave the results:

LMS:

- Sum = 13.68 + 11.78 + 9.94 + 7.83 + 7.93 = 51.16
- Mean = 51.16 / 5 = 10.23

FMS:

- Sum = 22.00 + 20.00 + 18.00 + 16.00 + 16.00 = 92.00
- Mean = 92.00 / 5 = 18.40

Across the five Eb/No values that were considered, LMS has a mean Average Iteration count of 10.23, while FMS has a mean Average Iteration count of 18.40. This shows that the Average Iteration counts of FMS were reduced by almost 50% when LMS was implemented.

4.4 Summary of Percentage Improvement of LMS+Early Stopping over FMS

To consolidate the energy and iteration-count findings into a single comparative view, Figure 10 plots the percentage improvement of LMS+Early Stopping over FMS, for both energy savings and iteration reduction, at each tested Eb/No value.

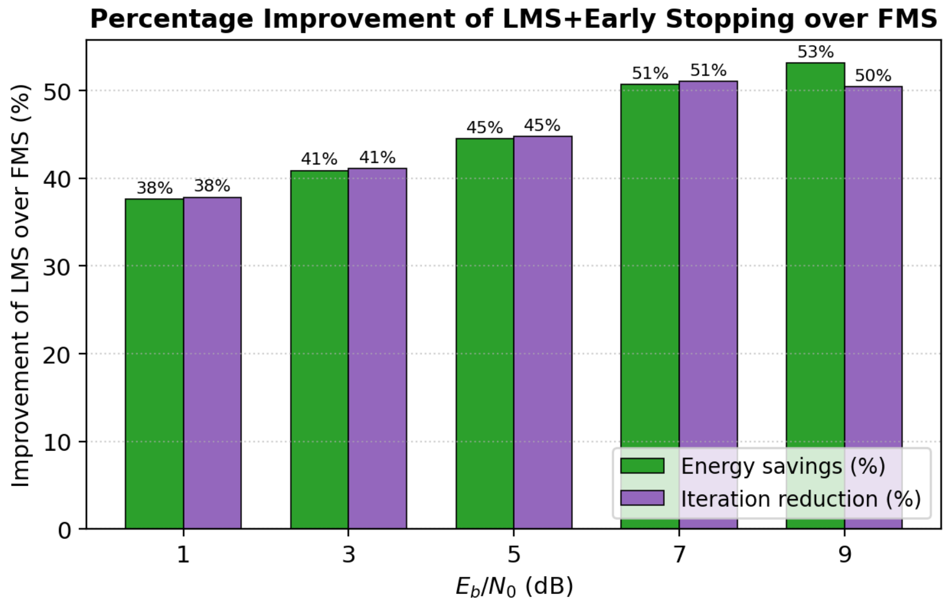

Figure 4.6: Percentage Improvement of LMS+Early Stopping over FMS in Energy Savings and Iteration Reduction

Figure 4.6 shows that the percentage improvement delivered by the proposed decoder is not constant across the Eb/No range: it grows from approximately 38% at Eb/No = 1 dB to approximately 53% (energy) and 50% (iterations) at Eb/No = 9 dB. This trend indicates that the proposed LMS+Early Stopping scheme becomes increasingly advantageous as channel conditions improve, since early stopping is triggered sooner under low-noise conditions, compounding the layered-scheduling benefit. Averaged across the five tested points, the mean energy saving is approximately 45.4%, and the mean iteration-count reduction is approximately 45.0%, consistent with the headline figures reported in Section 4.1 and Section 4.3.

Table 4.4: Percentage Improvement of LMS+Early Stopping over FMS by Eb/No

Eb/No (dB)	Energy Savings (%)	Iteration Reduction (%)
1	37.6	37.8
3	40.9	41.1
5	44.5	44.8
7	50.7	51.1
9	53.1	50.4
Mean	45.4	45.0

5. Conclusion and Recommendation

5.1. Conclusion

This study addressed the critical problem of energy-efficient error control in Wireless Sensor Networks by optimizing the decoding process of LDPC codes. Through a simulation-based approach using MATLAB, an enhanced LDPC decoder based on Layered Min-Sum decoding and Early Stopping was developed and evaluated under realistic communication conditions. The results clearly demonstrate that the proposed optimization significantly reduces decoding energy consumption while maintaining excellent error correction performance. Quantitatively, the decoding energy per bit was reduced from a maximum of 1.55 nJ/bit in the Flooding Min-Sum decoder to 0.97 nJ/bit in the Layered Min-Sum decoder at low Eb/No, and from 1.13 nJ/bit to 0.53 nJ/bit at higher Eb/No values. This consistent reduction confirms the effectiveness of layered scheduling and early stopping in minimizing computational overhead.

In addition, the optimized decoder achieved a substantial reduction in decoding iterations. The mean iteration count dropped from 18.40 in the Flooding Min-Sum decoder to 10.23 in the Layered Min-Sum decoder, representing nearly a 44% reduction in average iterations. This reduction is directly responsible for the observed energy savings and highlights the efficiency of the early stopping mechanism in terminating decoding once convergence is achieved. From a reliability standpoint, the optimized decoder maintained superior BER performance across all tested scenarios. At Eb/No = 3 dB, the proposed decoder achieved a BER of 0.1038, outperforming BCH, Reed–Solomon, and Convolutional codes by a wide margin, while also slightly improving on the conventional Flooding Min-Sum LDPC decoder. These results confirm that energy efficiency gains were achieved without sacrificing error correction capability.

The study therefore establishes that LDPC codes, when optimized using Layered Min-Sum decoding and Early Stopping, are suitable for deployment in energy-constrained WSN nodes. This challenges previous assumptions that limited LDPC usage to high-power systems and demonstrates that strong error correction and low energy consumption can be jointly achieved through algorithm-level optimization.

5.2 Recommendations

Based on the findings of this study, the following recommendations are made:

i. The Layered Min-Sum LDPC decoder with Early Stopping should be adopted in Wireless Sensor Networks where energy efficiency and communication reliability are critical design requirements.
ii. The proposed decoding approach should be extended to hardware-based implementations, such as FPGA or ASIC designs, to further evaluate its real-time performance and power savings.
iii. Future system designs should explore the use of larger LDPC block lengths to further enhance error correction performance while preserving energy efficiency gains observed in this study.

References

[1] Abdulwahid, A., & Salih, M. (2022). Wireless Sensor Networks Applications, Challenges, and Security Requirements. Proceedings of 2nd International Multi-Disciplinary Conference Theme: Integrated Sciences and Technologies, IMDC-IST 2021, 7-9 September 2021, Sakarya, Turkey. https://doi.org/10.4108/eai.7-9-2021.2314823

[2] Ali, M. M., Hashim, S. J., Chaudhary, M. A., Ferré, G., Rokhani, F. Z., & Ahmad, Z. (2023). A Reviewing Approach to Analyze the Advancements of Error Detection and Correction Codes in Channel Coding with Emphasis on LPWAN and IoT Systems. IEEE Access, 11, 127077–127097. https://doi.org/10.1109/ACCESS.2023.3331417

[3] Alrajeh, N. A., Marwat, U., Shams, B., & Shah, S. S. H. (2015). Error Correcting Codes in Wireless Sensor Networks: An Energy Perspective. 2.

[4] Chowdhury, S. M., & Hossain, A. (2020). Different Energy Saving Schemes in Wireless Sensor Networks: A Survey. Wireless Personal Communications, 114(3), 2043–2062. https://doi.org/10.1007/s11277-020-07461-5

[5] Faris, M., Mahmud, M. N., Salleh, M. F. M., & Alnoor, A. (2023). Wireless sensor network security: A recent review based on state-of-the-art works. *International Journal of Engineering Business Management, 15*, 18479790231157220. https://doi.org/10.1177/18479790231157220

[6] Gupta, S. K., & Sinha, P. (2014). Overview of Wireless Sensor Network: A Survey. 3(1).

[7] Hamdan, M., & Abdullah, A. (2016). Analysis and Performance Evaluation of Convolutional Codes over Binary Symmetric Channel Using MATLAB.

[8] Kadel, R., Paudel, K., Guruge, D. B., & Halder, S. J. (2020). Opportunities and Challenges for Error Control Schemes for Wireless Sensor Networks: A Review. Electronics, 9(3), 504. https://doi.org/10.3390/electronics9030504

[9] Kamal, Z.-E.-H., & Salahuddin, M. A. (2015). Introduction to Wireless Sensor Networks. In D. Benhaddou & A. Al-Fuqaha (Eds.), Wireless Sensor and Mobile Ad-Hoc Networks (pp. 3–32). Springer New York. https://doi.org/10.1007/978-1-4939-2468-4_1

[10] Khan, K. (2025). Time-Reversal Symmetry in Quantum Wireless Sensor Networks (arXiv:2504.00298). arXiv. https://doi.org/10.48550/arXiv.2504.00298

[11] Kundaeli, H. N. (2020). The analysis of truncated ARQ and HARQ schemes using transition diagrams. *International Journal of Engineering, Science and Technology*, 12(2), 35–51. https://doi.org/10.4314/ijest.v12i2.5
[12] Nourildean, S. W., Hassib, M. D., & Mohammed, Y. A. (2022). Internet of things based wireless sensor network: A review. *Indonesian Journal of Electrical Engineering and Computer Science, 27*(1), 246. https://doi.org/10.11591/ijeecs.v27.i1.pp246-261
[13] Pandey, M., & Pandey, V. K. (2015). Comparative Performance Analysis of Block and Convolution Codes. *International Journal of Computer Applications*, 119.
[14] Sai, K., Pavankumar Kovi, Jangam, S., Goud Kumar, & Kosgi. (2017). WIRELESS SENSOR NETWORKS AND APPLICATIONS. Unpublished. https://doi.org/10.13140/RG.2.2.23192.19207
[15] Singh, A. P., Luhach, A. K., Gao, X.-Z., Kumar, S., & Roy, D. S. (2020). Evolution of wireless sensor network design from technology centric to user centric: An architectural perspective. *International Journal of Distributed Sensor Networks,* 16(8), 155014772094913. https://doi.org/10.1177/1550147720949138
[16] Silicon Laboratories Inc (2013) The Evolution of Wireless Sensor Networks.
[17] Tossa, F., Faga, Y., Abdou, W., Ezin, E. C., & Gouton, P. (2025). Wireless Sensor Network Deployment: Architecture, Objectives, and Methodologies. Sensors, 25(11), 3442. https://doi.org/10.3390/s25113442
[18] Venkatesan, D., Sunil, B., Reddy, B. A., & Naidu, K. J. (2016). Low Power Asic Implementation of Ldpc Decoder.
[19] Xu, X., Tang, J., & Xiang, H. (2022). Data Transmission Reliability Analysis of Wireless Sensor Networks for Social Network Optimization. *Journal of Sensors, 2022*, 1–12. https://doi.org/10.1155/2022/3842722
[20] Zijie, F., Al-Shareeda, M. A., Saare, M. A., Manickam, S., & Karuppayah, S. (2023). Wireless sensor networks in the internet of things: Review, techniques, challenges, and future directions. *Indonesian Journal of Electrical Engineering and Computer Science,* 31(2), 1190. https://doi.org/10.11591/ijeecs.v31.i2.pp1190-1200