


## A High-Throughput Reconfigurable LDPC Codec for Wide Band Digital Communications



Divyashree Yamadur Venkatesh<sup>1</sup>, Komala Mallikarjunaiah<sup>1</sup>, Mallikarjunaswamy Srikantaswamy<sup>2\*</sup>

<sup>1</sup> Department of Electronics and Communication Engineering, SJB Institute of Technology, Bengaluru, Karnataka 560060, India

<sup>2</sup> Department of Electronics and Communication Engineering, JSS Academy of Technical Education, Bengaluru, Karnataka 560060, India

Corresponding Author Email: [pruthvi.malli@gmail.com](mailto:pruthvi.malli@gmail.com)

<https://doi.org/10.18280/jesa.560402>

### ABSTRACT

**Received:** 19 June 2023

**Revised:** 14 July 2023

**Accepted:** 25 July 2023

**Available online:** 31 August 2023

#### Keywords:

*low-density parity-check (LDPC), forward error rate (FER), bit error rate (BER), signal to noise ratio (SNR), codec design, error correction codes, reconfigurable parallel processing*

In recent days, an extensive digital communication process has been performed. Due to this phenomenon, proper maintenance of communication encoding and decoding parallel operation without any overhead such as signal attenuation code rate fluctuations during the digital parallel communication process can be minimized and optimized by adopting parallel encoder and decoder operations. To overcome the above-mentioned drawbacks by using proposed reconfigurable code rate cooperative (RCRC) using low-density parity check (LDPC) code for Gigabits Wide Code Encoder/Décoder Operations. The proposed RCRC is capable to vary the switch parallel operation as per the load. The low-density parity check (LDPC) is used for linear error correcting code in the communication process. it is very essential to reduce the power dissipation and noise in a transmission channel. Due to these phenomena, the decrease in power dissipation and enhancement of the accuracy in communication process are achieved and also operate over gigabits/sec data and it effectively performs linear encoding, dual diagonal form, widens the range of code rate and optimal degree distribution of LDPC mother code and all daughter codes for effectively performing the parallel switching operations in highly complex and wide range of code rate communication process. It is the highest upper bounded code rate as compared to the existing methods. The proposed method optimizes the transmission rate and is capable to operate on a 0.98 code rate. It is the highest upper bounded code rate as compared to the existing methods. the proposed method's implementation has been carried out using MATLAB and as per the simulation result, the proposed method is capable of reaching a throughput efficiency greater than 8.2 Gigabits per second with a clock frequency of 160MHz.

## 1. INTRODUCTION

A challenge of designing RC LDPC codes, which need to improve size of rate of code, because to widen the overall channel size and to give a good and proper service depending on channel conditions. LDPC codes are a class of error-correcting codes that are widely used in communication systems to detect and correct errors in transmitted data. LDPC codes are known for their excellent error-correcting performance and low complexity. RC LDPC codes are a specific type of LDPC codes that have a regular structure, where the parity-check matrix is composed of regular column patterns. The regularity of RC LDPC codes simplifies the encoding and decoding processes, making them more efficient and suitable for practical applications. The design of LDPC codes involves constructing a parity-check matrix with desirable properties. RC LDPC codes are characterized by having a regular column structure, where each column of the parity-check matrix has the same weight, meaning it contains the same number of 1's. This regularity makes the decoding process more straightforward by enabling parallel processing and efficient hardware implementations.

RC LDPC codes have been adopted in various communication standards, such as wireless communication

systems (e.g., Wi-Fi, 4G, 5G) and satellite communication systems, due to their good error-correcting capabilities, low decoding complexity, and regular structure that facilitates efficient hardware implementations. Overall, RC LDPC codes are a specific type of LDPC codes with a regular column structure, which provides advantages in terms of encoding, decoding, and implementation efficiency, making them a popular choice for error correction in modern communication systems. Puncturing LDPC refers to a technique used to adjust the code rate of LDPC codes by intentionally removing some of the parity bits during encoding. This allows for the transmission of data at a different rate than the original LDPC code. LDPC codes are typically designed with a specific code rate in mind, which represents the ratio of information bits to the total number of transmitted bits. However, there may be scenarios where it is desirable to transmit data at a different rate, either to accommodate varying channel conditions, increase data throughput, or meet specific system requirements.

Puncturing LDPC allows for rate adaptation by selectively discarding certain parity bits generated during the encoding process. Instead of transmitting all the parity bits, a subset of them is chosen to be transmitted, while the remaining parity bits are discarded. The information bits remain the same, but

the code rate is effectively adjusted by puncturing. The puncturing process is carefully designed to ensure that the punctured LDPC code maintains its error-correcting capabilities while operating at the desired rate. The choice of which parity bits to puncture is based on specific rules and algorithms, considering factors such as the impact on error correction performance and the desired code rate. At the receiver side, the punctured LDPC code is decoded using the standard LDPC decoding algorithm.

To reduce Frame Error Rate (FER) or increase Signal-to-Noise Ratio (SNR) in LDPC decoding, you can employ several techniques and strategies. Here are some commonly used methods:

**Iterative Decoding:** LDPC decoding is an iterative process where messages are exchanged between variable nodes and check nodes to improve the reliability of the decoding.

**Code Design:** The choice of LDPC code design can significantly impact the decoding performance. There are various methods for constructing LDPC codes, such as regular or irregular code designs, structured codes, or protograph-based codes. Optimizing the code design for your specific requirements, such as target SNR or FER, can enhance the decoding performance.

**Code Rate Selection:** LDPC codes offer flexibility in choosing the code rate. By selecting an appropriate code rate, you can trade-off between error correction capability and information rate. Lower code rates provide better error correction at the expense of lower information rate, which can help reduce FER and improve decoding performance at low SNR.

**Decoding Algorithm Optimization:** The LDPC decoding algorithm itself can be optimized for improved performance. Techniques such as algorithm variations (e.g., Min-Sum, Offset Min-Sum, or Sum-Product), schedule optimization, or convergence criteria adjustments can enhance the decoding performance and reduce FER.

The decoding process takes into account the knowledge of the punctured parity bits, which were not transmitted, to recover the original information bits. However, due to the puncturing process, the decoding performance of the punctured LDPC code may be affected compared to the original full-rate LDPC code. Puncturing LDPC provides flexibility in adjusting the code rate of LDPC codes to match specific transmission requirements. It enables efficient utilization of available resources and allows for adaptive transmission based on channel conditions or other system constraints. However, it's important to carefully design the puncturing scheme to ensure that the desired rate adaptation is achieved without compromising the error correction capabilities of the code. Puncturing LDPC provides flexibility in adjusting the code rate of LDPC codes to match specific transmission requirements.

It enables efficient utilization of available resources and allows for adaptive transmission based on channel conditions or other system constraints. However, it's important to carefully design the puncturing scheme to ensure that the desired rate adaptation is achieved without compromising the error correction capabilities of the code [1-5]. Code designs major challenge with specified way to maintain the performance of both codes are nearer to potential. Piercing leads to modification of degree distribution, is a main element for finding the decoding act, it's complex to get the best daughter codes at highest code levels.

A group of RC LDPC codes is recommended by document

with different code levels. By asserting a perfect transmission pattern with a zero-filling encryption algorithm, the method to get RC codes doesn't change daughter codes degree of distribution. Thus, the method forces rates to a 0.98. Designed from shifted identity matrices, the codes match the implementation of high-speed parallel encoders and decoders.

Rate-Compatible LDPC (RC-LDPC) codes are a family of LDPC codes that offer different code rates while maintaining the same underlying structure. These codes are designed to provide rate adaptability, enabling flexible and efficient use of LDPC codes in communication systems.

The idea behind rate-compatible LDPC codes is to construct a set of LDPC codes with varying rates that share a common structure. This common structure allows for efficient encoding and decoding algorithms to be used across the different codes in the set. The different rates are achieved by puncturing, which involves selectively removing parity bits from the original LDPC code.

LDPC codes with unchanged degree distributions, it means that the degree distribution of the variable and check nodes remains fixed throughout the LDPC code construction. Typically, this refers to using fixed-degree LDPC codes, where the number of connections or edges per node remains constant.

In LDPC codes with unchanged degree distributions, the code's performance and properties are determined by the fixed degree distribution itself. These codes are often characterized by regular structures and can provide a good trade-off between performance and complexity.

The switching speed of an LDPC (Low-Density Parity-Check) decoder refers to the speed at which the decoder can process incoming data and provide a decoded output. The switching speed is an important factor in determining the maximum data rate or throughput that the LDPC decoder can handle.

The deployment results to a design of field programmable gate array (FPGA) devices, which show that a 32-parallel encoder for the said LDPC codes having rates from 0.5 to 0.98 is having a capacity to reach an outturn of 8.2(1.9) Gigabits per second (Gbps) with the help of a clock frequency of 180MHz and absorbing only 0.3% (12%) of Xilinx Virtex-5's overall resources. Rate-compatible LDPC codes system is as shown in Figure 1. The information 'm' bits are applied to the RC-LDPC and the output of RC-LDPC generates a codeword 'C', which is then applied to modulation process. the modulated output is applied to the channel.

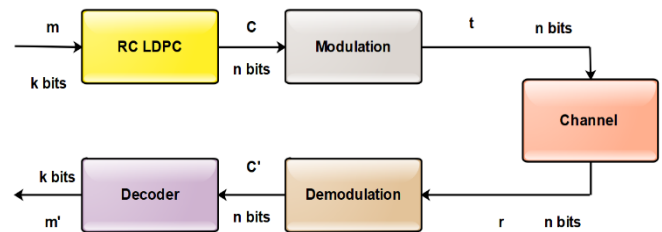


Figure 1. The fundamental block diagram of RC-LDPC codes

## 2. METHODOLOGY

### 2.1 Build rate compatible LDPC codes

Rate-compatible LDPC (Low-Density Parity-Check) codes

are a family of error correction codes that provide different code rates while maintaining similar performance characteristics. These codes are widely used in various communication systems, such as wireless communication, optical communication, and storage systems. Rate-compatible LDPC codes offer the advantage of flexible code rates, allowing for efficient and adaptive transmission over varying channel conditions. By adjusting the code rate, the system can achieve a good balance between data rate and error correction capability. The design and encoding algorithm for rate-compatible LDPC (RC-LDPC) codes involve creating a family of LDPC codes with different rates while maintaining a common structure. In LDPC (Low-Density Parity-Check) codes, a submatrix refers to a subset of rows and columns within the larger parity check matrix that defines the code. These submatrices play a crucial role in the encoding and decoding processes of LDPC codes. The performance of rate-compatible LDPC codes is often evaluated in terms of their error correction capability, decoding complexity, and implementation efficiency. When properly designed and implemented, rate-compatible LDPC codes can achieve near-Shannon limit performance, which is the theoretical maximum data rate for a given channel. The parity check matrix of an LDPC code typically has a sparse structure, with a low density of ones, hence the name "low-density." This sparsity is a desirable property as it enables efficient encoding and decoding algorithms. However, it also means that the matrix can be quite large and dense in practice, making it challenging to implement LDPC codes in hardware or software systems. RC LDPC codes have been considered using a subsequent parent matrix, which is having a set of  $i \times X$  ( $I+J$ ) sub-matrices as shown in Eq. (1).

$$M = \begin{bmatrix} m_{1,1} & . & . & m_{1,j} & K & . & . \\ m_{2,1} & . & . & m_{2,j} & K & K & . \\ . & . & . & . & . & . & . \\ m_{i-1,1} & . & . & m_{i-1,j} & K & K & K \\ m_i & . & . & m_{i,j} & . & K & K \end{bmatrix} \quad (1)$$

where, 'K' is represented the it's an  $a \times a$  identity matrix and the null matrices are indicated by unmarked spaces. For  $1 \leq i \leq I$  and  $1 \leq j \leq J$  the submatrix  $m_{i,j}$  within 'M' at position  $(i,j)$  is either a shifted identity matrix [6-9]. The shifted identity matrix is gained by performing a circularly shifting the rows of 'K' to the right by  $S_{i,j}$  positions i.e.,  $m_{i,j} = K(S_{i,j})$ .  $S_{i,j}$  shows shifting coefficient, and has been selected from the restricted set-in range 0 and 'a'. The main purpose is to prevent short and medium length cycles depending on the methods in [6-9]. The submatrix size is represented by 'a'. To differentiate among conventional rows and columns, we access column of submatrices, which is present like a block column inside M. It has I block-rows and (J+1) block columns inside the matrix (1) [10, 11].

High-speed parallel processing of LDPC (Low-Density Parity-Check) codes is an important aspect of their implementation in practical communication systems. LDPC codes are known for their inherent parallelism, which allows for efficient and fast decoding using parallel processing techniques. In the iterative decoding process of LDPC codes, the flipping of variable nodes and check nodes can be performed in parallel. This approach enables multiple variable and check nodes to be processed simultaneously, increasing the decoding speed. SIMD is a parallel processing technique where multiple data elements are processed simultaneously

using a single instruction. SIMD architectures, such as Intel's SSE (Streaming SIMD Extensions) and ARM's NEON, can be utilized to perform LDPC decoding operations on multiple code bits simultaneously.

High-speed LDPC decoding can be achieved by employing multiple processing units or cores. Each processing unit can independently decode a portion of the code word, and the results can be combined at the end to obtain the final decoded output. This approach leverages the parallel processing capabilities of modern multi-core CPUs or dedicated hardware accelerators.

## 2.2 Transmission and encoding algorithm

LDPC (Low-Density Parity-Check) codes are widely used for error correction in communication systems. The transmission and encoding algorithm for LDPC codes involves preparing the input data, encoding it using the LDPC parity check matrix, and transmitting the encoded data. Shifted identity matrix LDPC codes, also known as circulant LDPC codes, are a specific class of LDPC codes that utilize a shifted identity matrix as the parity check matrix. The parity check matrix of a shifted identity matrix LDPC code has a structured circulant pattern, where each row is a cyclic shift of the previous row. The diagonal elements of the shifted identity matrix are typically set to 1, while the off-diagonal elements are set to 0.

Assume splitting of a codeword 'X' into (J+K) parts, as shown in Eq. (2).

$$X = [p_1, p_2, p_3, \dots, p_j, q_1, q_2, q_3, \dots, q_i] \quad (2)$$

where,  $p_j$  is presented partitioning codewords with  $1 \leq j \leq J$  shows systematic bits of 'a' vector and 'q' with  $1 \leq i \leq J$ , that is a vector of q parity bits. With coding basics,  $MX^T = 0^T$ , results to a parity bit as Eqs. (3) and (4).

$$Q_1^T = \sum_{j=1}^J m_{1,k} P_k^T \quad (3)$$

$$Q_i^T = \sum_{k=1}^K m_{i,j} P_k^T + Q_{i-1}^T \quad (4)$$

with  $2 \leq i \leq I$ , it is agreed that the introduction of LDPC codes is linear time encoded. The code presented in equation 1 possess leas rate  $R_L$  is given in Eq. (5).

$$R_L = J/(J + I) \quad (5)$$

The daughter code having highest rate,  $R_H$  as shown in Eq. (6) is resulted by passing only one parity bit vector  $q_i$ .

$$R_M = J/(J + 1) \quad (6)$$

For ARQ which is called as Automatic Repeat Request transmission, step-by-step transmission of systematic and parity bits are as follows:

- transmit parity bit vector  $q_i$  and systematic bit vector  $p$ .
- transfer parity bit vectors  $[q_1, q_2, \dots, q_{i-1}]$  according to the following order, if asked.

$$\begin{aligned} & q_{\frac{j}{2}}; \\ & q_{j/4}, q_{3j/4}; \\ & q_2, q_4, q_6 \dots \dots \dots (the \text{ rest even numbers}); \end{aligned}$$

$q_1, q_3, q_5 \dots \dots \dots$  (the rest even numbers).

### 2.3 Case study

Rate-compatible and rate-adaptive LDPC codes are two variations of LDPC codes that offer flexibility in adjusting the code rate to accommodate varying transmission conditions and requirements.

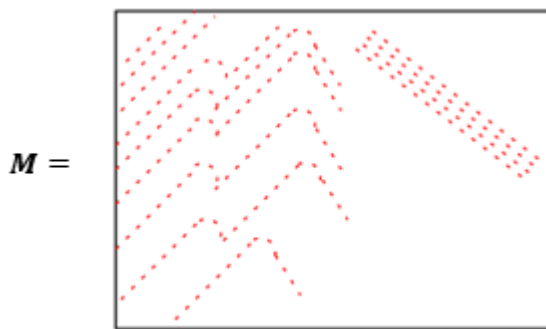
Rate-compatible and rate-adaptive LDPC codes are two different approaches used in coding theory to provide flexibility in data transmission over unreliable communication channels. While they share some similarities, they have distinct characteristics and applications. Rate-compatible LDPC codes are designed to support multiple code rates derived from a single base code. They offer a range of code rates without sacrificing the overall code performance.

These codes allow for incremental transmission rates, meaning the sender can adapt the code rate based on the channel conditions or desired trade-offs between error resilience and data rate.

Rate-compatible LDPC codes typically exhibit excellent error correction capabilities at moderate to high code rates. However, their performance may degrade at lower code rates due to the limitation of the base code design.

To construct an LDPC code with a  $32 \times 48$  array of sub-matrices, we can follow a structured LDPC code construction approach such as Gallager codes or Tanner codes. Create a single sub-matrix: design a small regular or structured sub-matrix that satisfies the desired code rate. This sub-matrix should have a low density of ones and meet the code's error-correcting requirements.

Replicate the single sub-matrix in a  $32 \times 48$  array, filling the entire array. Ensure that each sub-matrix in the array maintains the same structure as the original sub-matrix. For instance, assume a  $32 \times 48$  array of sub-matrices, which is considered as a mother matrix, across the entire text [12-14]. It is simple to extend the suggested approach to LDPC codes having any numbers of block-rows and block-columns. Solid bars in a Figure 2 shows an offset ID matrix with unmarked spaces shows null matrices. A least code rate of  $R_L = 0.5$  is associated with mother code.



**Figure 2.**  $32 \times 48$  sub matrix array constitutes a LDPC mother matrix. Shifted identity matrix forms solid bars and null matrices found unmarked spaces

We assume splitting a codeword  $x$  to smaller parts, say 64, is given in Eq. (7)

$$X = [p_1, p_2, p_3, \dots \dots p_{32}, q_1, q_2, q_3, \dots \dots q_{32}] \quad (7)$$

where,  $p_j$  with  $1 \leq j \leq 32$  indicates a vector of 'a'

systematic bits and  $q_i$  with  $1 \leq i \leq 32$  is a vector of 'a' parity bits. A systematic bit vector 'p' is initially transmitted by the transmitter along with the parity bit vector  $q_{32}$  [15-17]. It is discovered that; daughter code has the maximum rate  $R_M = 0.98$ . Send the other parity bit vectors in the following order if requested:

$$q_{12}, q_6, q_{18}, q_2, q_4, q_8, q_{10}, q_{14}, q_{16}, q_{20}, q_{22}, q_1, q_3, q_5, q_7, q_9, q_{11}, q_{13}, q_{15}, q_{17}, q_{19}, q_{21}, q_{23}.$$

We shall discuss the benefits of transmitting the parity bit vectors  $p_{32}$  first, in Section IV.

### 2.4 Comparative analysis of rate-compatible code and rate-adaptive code

RC- LDPC codes are a family of LDPC codes that share a common structure while offering different code rates within the family. These codes are designed to be compatible, meaning they can be concatenated or switched between different rates without significant degradation in performance. Rate-adaptive LDPC codes, also known as adaptive rate LDPC codes, are a type of LDPC codes that dynamically adjust the code rate based on the transmission conditions. These codes allow for flexible adaptation to optimize performance and efficiency in varying channel conditions or system requirements. Rate-adaptive LDPC codes are created [18-20], which are indicated with the help of mother matrix, and it also includes an array of circularly shifted identity sub-matrices. By deleting the highest rows from the mother matrix, the parity-check matrices of daughter codes with higher levels are obtained i.e., by row-deleting approach, the rates are adapted. By piercing few of the parity bits of the mother code, the daughter codes of rate-compatible LDPC codes are obtained. By deleting few columns from the parity portion of the mother matrix, the greater rates of parity-check matrices of daughter codes are generated. Table 1 show the performance analysis between reconfigurable code rate cooperative (RCRC) and Rate-Adaptive LDPC Codes.

**Table 1.** Performance analysis between reconfigurable code rate cooperative (RCRC) and Rate-Adaptive LDPC Codes

Particulars	Rate Adaptive LDPC Code	Reconfigurable Code Rate Cooperative (RCRC)
System parity check bits matrix for code length applications	This method is constant in various row removal adaptive modulation and coding algorithm	This method is constant in various column removal adaptive modulation and coding algorithm

### 3. IMPLEMENTATION PROCESS OF UNIVERSAL ENCODER

Designing a universal LDPC encoder and decoding algorithm involves creating algorithms that can handle arbitrary LDPC codes with different parameters and structures. Additionally, LDPC decoding can involve advanced mathematical techniques, data structures, and algorithms. Implementation details, such as message update rules, scheduling strategies, or termination criteria, may vary

depending on the chosen decoding algorithm and LDPC code construction. Overall, designing a universal LDPC encoder and decoding algorithm requires a thorough understanding of LDPC codes, coding theory, iterative decoding techniques, and performance analysis. The goal is to create a flexible and efficient solution that can handle different LDPC codes effectively.

An XOR processor configuration in an FPGA (Field-Programmable Gate Array) can be used to accelerate the XOR operations required in LDPC encoding and decoding algorithms. XOR operations are fundamental in LDPC codes, as they are used for bitwise calculations between the information bits, parity bits, and the LDPC code's parity check matrix.

The fact is that usage of single encoder/decoder pair, decoding and encoding of RC LDPC codes can be done and is one of its key benefits [21-24]. In this section, we walk through how the universal encoder is implemented in practice. Think about the following code having 'I' as 24 and 'j' as 24 in II.B). Due to the fact that  $m_{i,j}$  is either a null matrix or a shifted identity matrix, the matrix vector multiplications in Eqs.(3) and (4) is computed as follows:

$$X_{i,j}^T = m_{i,j} p_j^T \quad (8)$$

where,  $X_{i,j}^T$  with  $1 \leq i \leq 32$  and  $1 \leq j \leq 32$  indicates a vector of 'a' bits.  $m_{i,j}$  indicates shifted identity matrix i.e.,  $m_{i,j} = k(S_{i,j})$ . By shifting  $p_j$  to the left side by  $S_{i,j}$  places,  $X_{i,j}$  is obtained.  $m_{i,j}$  is a null matrix and  $X_{i,j}$  is a zero vector. Using a simple XOR processor, the computation in (3) and (4) can be implemented [25, 26].

$$q_i(k) = \sum_{j=1}^{24} x_{i,j}(k) + s_{i-1}(k) \quad (9)$$

where,  $q_0(k) = 0, 1 \leq i \leq 24$  and  $1 \leq j \leq a$ . The  $k$  - bits in the bit vector  $x_{i,j}$  is represented by every signal to the input  $x_{i,j}(k)$ . To shorten the time, it takes for signals to propagate from input to output, a tree design is used. A made up of 32 XOR processors are used to design a 32-parallel encoder and integrated into MATLAB and FPGA devices. The target XC5VFX200T device's created implementation was placed and routed using the Xilinx Development System tool suite (ISE 10.1) with the speed option set to -2. Estimation of highest clock frequency was carried out using Xilinx static time analysis. The results give the following observation: 80 LUTs, slices of 416 with 467 flip-flops were used by 32 parallel encoders. Thus, the total resources consumed by encoder were only 3/10 percentage in an XC5VFX200T device [27, 28]. LDPC codes are widely used in high-speed communication systems, including those operating at gigabit rates. LDPC codes have been proven to be effective in achieving high data rates and reliable error correction. LDPC codes are known for their excellent error correction performance and their ability to approach the theoretical limits of channel capacity. They achieve this by exploiting the sparsity of the parity-check matrix, which allows for efficient encoding and decoding algorithms. In the context of gigabit communication, LDPC codes can be utilized to provide reliable and efficient error correction for data transmission at speeds of billions of bits per second (gigabits per second). These codes are often used in various applications, such as optical communication systems, high-speed wired communication, and wireless communication protocols.

The clock frequency and throughput values of an LDPC decoder or encoder depend on various factors, including the implementation architecture, LDPC code parameters, and the target performance requirements. The clock frequency represents the rate at which operations are executed in the LDPC system. The achievable clock frequency depends on the hardware implementation and the critical path of the LDPC decoder or encoder design. FPGA designs typically have clock frequencies ranging from a few tens of megahertz to several hundred megahertz, depending on the specific FPGA device and the complexity of the design.

Analysis of the Xilinx static timing shows that, in an XC5VFX200T device the highest encoding clock frequency of 32 parallel encoder attained is 460MHz. An encoding speed of 7200 Mbps was attained by the encoder with the use of 180MHz frequency for the encoding clock [29-32].

#### 4. DECODING ALGORITHM AND PERFORMANCE ANALYSIS

LDPC codes are widely used in high-speed communication systems, including those operating at gigabit rates. LDPC codes have been proven to be effective in achieving high data rates and reliable error correction. LDPC codes are known for their excellent error correction performance and their ability to approach the theoretical limits of channel capacity. They achieve this by exploiting the sparsity of the parity-check matrix, which allows for efficient encoding and decoding algorithms.

In the context of gigabit communication, LDPC codes can be utilized to provide reliable and efficient error correction for data transmission at speeds of billions of bits per second (gigabits per second). These codes are often used in various applications, such as optical communication systems, high-speed wired communication, and wireless communication protocols.

RC- LDPC codes are designed to provide efficient and flexible error correction capabilities across a range of code rates. Here we are using Sum product algorithm to decode the implemented RC LDPC codes effectively. Sum and product of external communication in every column and each row is performed by this algorithm [19, 20]. Rather than using different series column processing in classical sum product method here we are using junction sum product sort out method. This decoding algorithm includes the computation of external communication in column processing into series processing. RC LDPC codes offer multiple code rates within a code family. This allows for the selection of different code rates based on specific transmission requirements, channel conditions, or system constraints. The flexibility of code rates enables efficient utilization of available resources while adapting to changing channel conditions.

RC LDPC codes are designed to facilitate seamless transitions between different code rates within the same code family. This means that switching between code rates can be achieved without significant performance degradation. The smooth transitions enable adaptive transmission schemes, where the code rate can be dynamically adjusted to maximize throughput or improve error correction capability. RC LDPC codes maintain a common structure across the code rate family. This allows for reusability of encoding and decoding algorithms, reducing implementation complexity and enabling efficient hardware or software designs. The compatibility of



encoding and decoding algorithms across code rates simplifies system integration and deployment.

That is, we are using a processor called junction series column processor to operate each series from bottom series to top series in every loop.

To effectively utilize RC LDPC codes, it is important to carefully select the appropriate code rate based on the specific transmission conditions, desired data rates, and error correction requirements. Additionally, performance analysis and optimization of encoding and decoding algorithms can help achieve the desired trade-offs between code rate, error correction performance, and throughput.

#### 4.1 Zero filling decoding algorithm

Before introducing the null-filling method for decoding the implemented RC codes. We shall talk over about the mother code. Assume the mother matrix as 'M' consisting of  $i \times a$  series and  $(j + i) \times a$  columns, in which  $(j + i)$  and  $I$  are block columns and series. Hence this code is sort out with the junction processor from bottom series to top series inside the matrix  $M$  in every loop [21, 22]. When sort out is processing the series column processor captures the pair extrinsic messages  $y_{mn}$  from matrix and log likelihood ratio (LLR) ( $Z_n$ ) from matrix after then it calculates the new ratio  $Z'_n$  and extrinsic communication  $y'_{mn}$  with  $1 \leq m \leq i \times a$  and  $1 \leq n \leq (j + i) \times a$ . The extracted n-bits LLR at received signal is given by Eq. (10).

$$Z_n = \log(s_n^0/s_n^1) \quad (10)$$

where,  $s_n^0$  and  $s_n^1$  is described as nth bit probabilities of '0' or '1' respectively. The Mother code with the least speed: Systematic segment and Parity segment are sent, where Log Likelihood Ratio of all the segments can be captured out of the signal collected [33-35]. For Daughter Code: few parity segments are not sent, where Nth segment is large positive constant which is not sent and assigned to  $Z_n$  with value 0.

### 5. PERFORMANCE ANALYSIS

Implemented RCRC-LDPC code has to evolved for its performance. Hence the simulations are performed along with the Binary Phase-Shift Keying Modulation in Additive White Gaussian Noise Channel for representing RC codes by Mother code matrices which is as seen in the above Figure 2 where  $a = 72$ . This method will finish if the correct code is originated else when it reaches 50 loops in the cycle. The proposed RCRC-LDPC has an advantage when it pushes upper rates (R) to 0.98 by transmitting a single parity bit vector (P24). The  $q_{32}$  code performs well so it is sent to loop. Frame Error Rate (FER) is a metric used to evaluate the performance of LDPC codes, representing the probability that an entire frame or block of data is received with errors. The frame error rate at 60th loop vs signal to noise ratio is given in Figure 2 which has length=1900 and Rate=0.98 [36-39]. The query is that why we are sending the segment vector  $q_{32}$ . The contrast occurred for Frame error rate (FER) at 60th loop with the signal-to-noise ratio (SNR) per bit  $\frac{E_b}{N_0}$  as shown in Figure 3 for those source code with the distance 1900 and a speed of 0.98. This clearly says the source code with the segment vector  $q_{32}$  gives top and best result. We are rewriting the Eq. (3) and Eq.

(4) by sending a vector  $S_i$  to describe why transferral plan attains the good performance which is given Eq. (11).

$$s_i^T = \sum_{k=1}^i (\sum_{j=1}^J m_{i,j} p_j^T) \quad (11)$$

with  $1 \leq i \leq J$ . It is shown in the Eq. (11) here the vector  $s_i$  is encoding with the matrix  $m_i$ , here  $m_i$ , indicates the top a block series in structured portion of  $M$ . Dispatching  $s_i$  with  $i < I$ , since the level sharing in  $M$  is dissimilar from ideal sharing in  $M$ , degradation of decoding result takes place. If we send vector  $s_i$ , the ideal level sharing keeps no change and the best performance is attained by transmission scheme.

Source code, the LDPC\_encode function takes a binary message vector and a parity-check matrix H as inputs, and returns the encoded codeword. The LDPC\_decode function performs LDPC decoding using the Belief Propagation algorithm. It takes a received vector, the parity-check matrix H, and an optional maximum number of iterations as inputs, and returns the decoded message.

Well-ordered LDPC codes, it typically implies a specific ordering of the bits in the codeword and parity-check matrix. Well-ordered LDPC codes are designed to have certain properties that simplify the encoding and decoding processes.

In Rate-Compatible (RC) LDPC codes, different code rates are achieved by puncturing the base LDPC code. The code iteration technique is often employed in RC LDPC codes to facilitate the smooth transition between different code rates within the code family.

The well\_ordered\_LDPC\_encode function takes a binary message vector and a well-ordered parity-check matrix H as inputs. It returns the encoded codeword, where the message bits are placed in the beginning of the codeword and the remaining bits are filled with zeros. By utilizing code iteration in RC LDPC codes, a smooth transition between different code rates can be achieved during the decoding process. This enables flexible adaptation to varying channel conditions, ensuring efficient error correction performance and data rates for different transmission scenarios. It's important to note that the specific implementation details of code iteration in RC LDPC codes can vary based on the chosen LDPC code construction technique and the specific requirements of the application. Different decoding algorithms, such as belief propagation or message passing, can be used in the code iteration process to achieve reliable error correction.

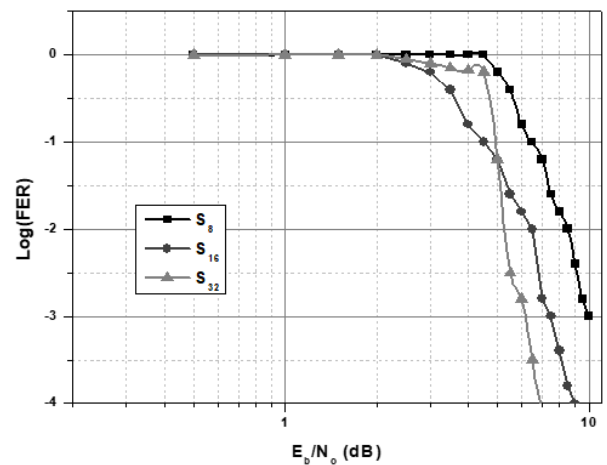
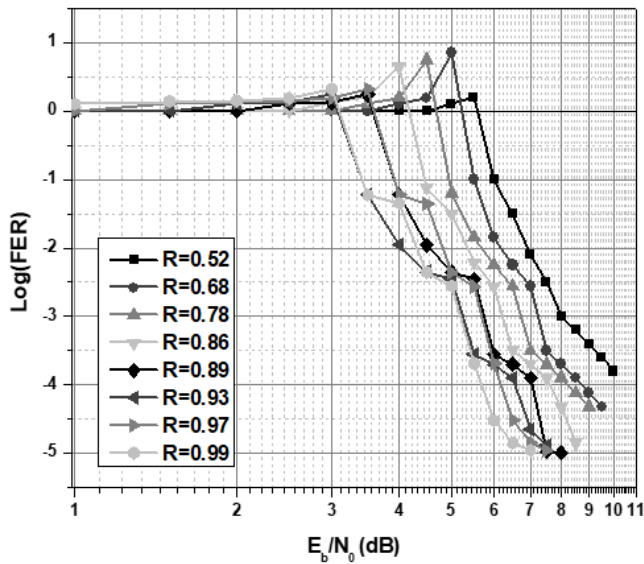
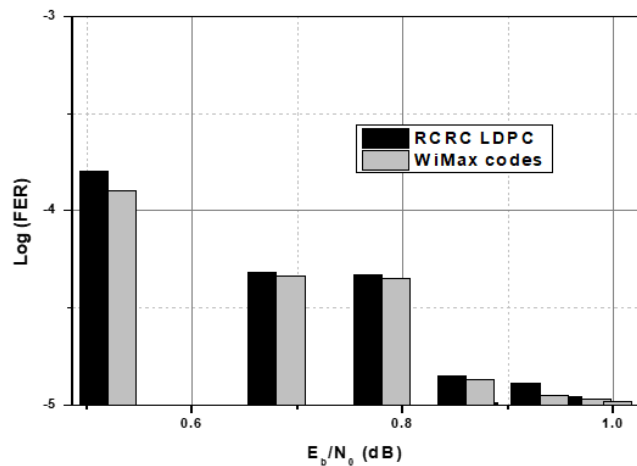


Figure 3. 60<sup>th</sup> code iteration with a range of 1900 and 0.98 code rate with respect to the transmission parity bit vector

Code iteration involves iterating between different punctured codes within the RC LDPC code family during the decoding process. Initially, the decoding process starts with the lowest code rate within the family. The decoding algorithm iterates until convergence or a predetermined number of iterations, exchanging messages between variable nodes and check nodes. After each iteration, the decoding algorithm switches to the next higher code rate within the RC LDPC code family and continues the iterative process. This iterative process continues until the highest code rate is reached or until convergence is achieved at a particular code rate.



**Figure 4.** The performance analysis of FER at the 60<sup>th</sup> code iterations of the LDPC codes with respect to the eight different code rates using the proposed RCRC-LDPC

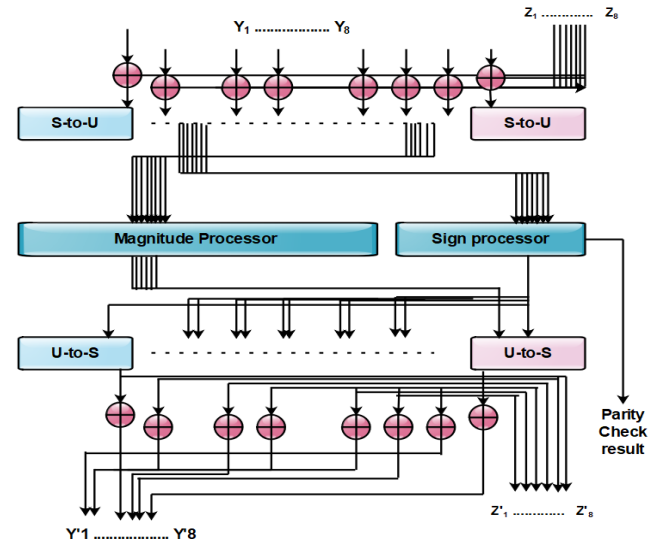


**Figure 5.** The performance analysis between RCRC LDPC and WiMax with respect to eight different code rates

In Figure 4 it is shown that the FER at the 60<sup>th</sup> loop versus  $E_b/N_0$  for 8 speeds, i.e.,  $R=0.99, 0.97, 0.93, 0.89, 0.86, 0.78, 0.68,$  and  $0.52$ . In the codes, there is a stable number of well-ordered bits, 1896. In order to standard the implemented RC LDPC codes, the contrast existing with the LDPC codes for the code of range 2562 and speed 0.52, 0.68, 0.78 and 0.86 are mentioned in the IEEE 802.16e standard, also known as WiMax codes. Figure 5 shows the performance analysis between RCRC LDPC and WiMax with respect to eight different code rates [40-42].

## 6. IMPLEMENTATION OF UNIVERSAL DECODER

Here we are implementing a 36 analogous decoder into action. For RC codes with  $a=72$ . It contains 36 junction series-column processors to calculate the hardware asserts utilization and decoding speed of a decoder. It is shown in the Figure 2 that mother matrices have a top 8 series level, that is maximum eight '1's in the mother matrix at each row. In the Figure 3 it is shown that the junction series-column processors have a level of 8. We have "s-to-u"(u-to-s) box labelled indicates is a converter. Here The converter is converting input wave from(to) a wave number to(from) a positive number.



**Figure 6.** Reconfigurable code rate cooperative (RCRC) and low-density parity check (LDPC) row-column processor with respect to the eight degrees

In Figure 6, there is a processor which consists of both inputs and outputs, eight in number. The processor is designed using fifteen XOR gates and its carryout sign functioning as given in Eqs. (12) and (13).

$$P = \prod_{i=1}^8 \text{sign}(x_i) \quad (12)$$

$$P_i = P \times \text{sign}(x_i) \quad (13)$$

with  $1 \leq i \leq 8$ . The magnitude operation is performed by the processor known as magnitude processor which has 8 inputs and 8 outputs in Figure 6.

$$y_j = \emptyset \left[ \sum_{i=1, i \neq k}^8 |x_i| \right] \quad (14)$$

where,  $y_j$  is represented the magnitude operation,  $\emptyset$  is describe the 7-bit look-up-table (LUT) function using 4 bits for fractional part. The measurement for the considered case code's submatrices is 72, the 36 analogous decoder is implemented with a MATLAB- XC5VFX200T FPGA tool with the rate of -2, it intakes below expedient: chop 8%, flipflop 12%, and Look up Table 12%. Timing analysis publishes above results, for 36 parallel decoder the maximum decoding clock frequency attained is 250 MHz with 180 MHz as decoding clock frequency and iteration number of 6 decoder is capable of attaining a decoding rate of 1.9 Gbps [43-45].

## 7. CONCLUSION

The proposed method RCRC LDPC code enhances the code rates and it is capable to optimize the transmission rate. The proposed method efficiently distributes the unchanged degree codes used for the more communication operation to increase the switching operation. Due to these activities, the mother code and all daughter codes very desired result. The proposed method optimizes the transmission rate and is capable to operate on a 0.98 code rate. It is the highest upper bounded code rate as compared to the existing methods. The proposed method enhances parallel operations of 32 encoders and 44 decoders with a throughput rate of 8.2 with 1.9 Gbps by using a clock frequency of 180MHz and power consumption is reduced by 0.54% as compared to the existing methods and FER is reduced to 0.12% and signal to noise ratio increases to 1.2% as compared to the existing method (WiMAX). Due to this, the proposed method is capable to operate high bit rate data without any overheads, and millions of electric components are integrated efficiently with more accelerated performance.

**Future scope:** Nowadays, 5G LTE communication is an emerging field. Due to more devices being connected to the 5G LTE network, this research work will be enhanced to 128-bit operational and increases the switching speed.

## ACKNOWLEDGMENTS

The authors would like to thank SJB Institute of Technology, JSS Academy of Technical Education, Bengaluru, Visvesvaraya Technological University (VTU), Belagavi and Vision Group on Science and Technology (VGST) Karnataka Fund for Infrastructure strengthening in Science & Technology Level – 2 sponsored “Establishment of Renewable Smart Grid Laboratory” for all the support and encouragement provided by them to take up this research work and publish this paper.

## REFERENCES

- [1] Papatheofanous, E.A., Reisis, D., Nikitopoulos, K. (2021). LDPC hardware acceleration in 5G open radio access network platforms. *IEEE Access*, 9: 152960-152971. <https://doi.org/10.1109/ACCESS.2021.3127039>
- [2] Su, B.S., Lee, C.H., Chiueh, T.D. (2022). A 58.6/91.3 pJ/b dual-mode belief-propagation decoder for LDPC and polar codes in the 5G communications standard. *IEEE Solid-State Circuits Letters*, 5: 98-101. <https://doi.org/10.1109/LSSC.2022.3167423>
- [3] Belhadj, S., Abdelmounaim, M.L. (2021). On error correction performance of LDPC and Polar codes for the 5G Machine Type Communications. In 2021 IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, pp. 1-4. <https://doi.org/10.1109/IEMTRONICS52119.2021.9422665>
- [4] Kim, N., Kim, J. (2021). Early termination scheme for 5G NR LDPC code. In 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, pp. 933-935. <https://doi.org/10.1109/ICTC52510.2021.9621117>
- [5] Petrović, V.L., El Mezeni, D.M. (2020). Reduced-complexity offset min-sum based layered decoding for 5G LDPC codes. In 2020 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, pp. 1-4. <https://doi.org/10.1109/TELFOR51502.2020.9306590>
- [6] Manjunath, T.N., Mallikarjunaswamy, S., Komala, M., Sharmila, N., Manu, K.S. (2021). An efficient hybrid reconfigurable wind gas turbine power management system using MPPT algorithm. *International Journal of Power Electronics and Drive Systems*, 12(4): 2501. <https://doi.org/10.11591/ijpeds.v12.i4.pp2501-2510>
- [7] Li, L., Zhang, K., Jiao, J., Sun, Y., Wu, S., Wang, Y., Zhang, Q. (2019). Performance analysis of finite length non-binary raptor codes under ordered statistics decoder. In 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, pp. 1-5. <https://doi.org/10.1109/VTCFall.2019.8891217>
- [8] Thazeen, S., Mallikarjunaswamy, S., Saqhib, M.N., Sharmila, N. (2022). DOA method with reduced bias and side lobe suppression. In 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, pp. 1-6. <https://doi.org/10.1109/IC3IOT53935.2022.9767996>
- [9] Dayananda, P., Srikantaswamy, M., Nagaraju, S., Velluri, R., Doddananjedevuru, M.K. (2022). Efficient detection of faults and false data injection attacks in smart grid using a reconfigurable Kalman filter. *International Journal of Power Electronics and Drive Systems*, 13(4): 2086-2097. <https://doi.org/10.11591/ijpeds.v13.i4.pp2086-2097>
- [10] Mahendra, H.N., Mallikarjunaswamy, S., Subramoniam, S.R. (2023). An assessment of vegetation cover of Mysuru City, Karnataka State, India, using deep convolutional neural networks. *Environmental Monitoring and Assessment*, 195(4): 526. <https://doi.org/10.1007/s10661-023-11140-w>
- [11] Lu, T., He, X., Kang, P., Xing, J., Tang, X. (2023). Parity-check matrix partitioning for efficient layered decoding of QC-LDPC codes. *IEEE Transactions on Communications*, 71(6): 3207-3220. <https://doi.org/10.1109/TCOMM.2023.3261380>
- [12] Liu, J.C., Wang, H.C., Shen, C.A., Lee, J.W. (2018). Low-complexity LDPC decoder for 5G URLLC. In 2018 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Chengdu, China, pp. 43-46. <https://doi.org/10.1109/PRIMEASIA.2018.8597812>
- [13] Liang, C.Y., Li, M.R., Lee, H.C., Lee, H.Y., Ueng, Y.L. (2019). Hardware-friendly LDPC decoding scheduling for 5G HARQ applications. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp. 1418-1422. <https://doi.org/10.1109/ICASSP.2019.8682481>
- [14] Katyushnyj, A., Krylov, A., Rashich, A., Zhang, C., Peng, K. (2020). FPGA implementation of LDPC decoder for 5G NR with parallel layered architecture and adaptive normalization. In 2020 IEEE International Conference on Electrical Engineering and Photonics (EExPolytech), St. Petersburg, Russia, pp. 34-37. <https://doi.org/10.1109/EExPolytech50912.2020.9243997>
- [15] Lin, C.Y., Liu, L.W., Liao, Y.C., Chang, H.C. (2021). A 33.2 Gbps/ITER. Reconfigurable LDPC decoder fully



- compliant with 5G NR applications. In 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, pp. 1-5. <https://doi.org/10.1109/ISCAS51556.2021.9401329>
- [16] Ortega-Ortega, A.L., Bravo-Torres, J.F. (2017). Combining LDPC codes, M-QAM modulations, and IFDMA multiple-access to achieve 5G requirements. In 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, pp. 1-5. <https://doi.org/10.1109/CONIELECOMP.2017.7891828>
- [17] Nga, N.T., Khanh, C.H., Tho, N.Q. (2022). An investigation of the 5G LDPC and polar decoding performance in spatial correlated MIMO-OFDMA system. In 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, pp. 1-7. <https://doi.org/10.1109/IMCOM53663.2022.9721755>
- [18] Papatheofanous, E.A., Reisis, D., Nikitopoulos, K. (2021). The LDPC challenge in software-based 5G new radio physical layer processing. In 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece, pp. 312-317. <https://doi.org/10.1109/MeditCom49071.2021.9647697>
- [19] Vigneswari, P., Sivakumari, S. (2021). Performance analysis of iterative minsum message passing decoding algorithm for 5G NR LDPC codes. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, pp. 142-146. <https://doi.org/10.1109/ICICV50876.2021.9388463>
- [20] Mallikarjunaswamy, S., Nataraj, K.R., Rekha, K.R. (2014). Design of high-speed reconfigurable coprocessor for next-generation communication platform. In: Sridhar, V., Sheshadri, H., Padma, M. (eds) Emerging Research in Electronics, Computer Science and Technology. Lecture Notes in Electrical Engineering, vol. 248. Springer, New Delhi. [https://doi.org/10.1007/978-81-322-1157-0\\_7](https://doi.org/10.1007/978-81-322-1157-0_7)
- [21] Li, F., Zhang, C., Peng, K., Krylov, A.E., Katyushnyj, A.A., Rashich, A.V., Tkachenko, D.A., Makarov, S.B., Song, J. (2021). Review on 5G NR LDPC Code: Recommendations for DTTB System. IEEE Access, 9: 155413-155424. <https://doi.org/10.1109/ACCESS.2021.3121587>
- [22] Sampath, K.J., Kumar, N.K., Yeswanth, K., Snehith, K., Anooja, B. (2021). An efficient channel coding architecture for 5G wireless using high-level synthesis. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 674-680. <https://doi.org/10.1109/ICOEI51242.2021.9453001>
- [23] Shrinidhi, J., Krishna, P.S., Yamuna, B., Pargunarajan, K. (2020). Modified min sum decoding algorithm for low density parity check codes. Procedia Computer Science, 171: 2128-2136. <https://doi.org/10.1016/j.procs.2020.04.230>
- [24] Muskan, A., Raj, T., Nisar, T., Abbas, M., Naz, S., Tiwari, U. (2023). Various channel coding schemes for 5G. In 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, pp. 1-13. <https://doi.org/10.1109/SCEECS57921.2023.10062998>
- [25] Pramanik, A., Maity, S.P., Sarkar, S. (2018). Compressed sensing image reconstruction by low density parity check codes and soft decoding of space time block codes. Computers & Electrical Engineering, 72: 553-565. <https://doi.org/10.1016/j.compeleceng.2018.01.014>
- [26] Pavithra, G.S., Pooja, S., Rekha, V., Mahendra, H.N., Sharmila, N., Mallikarjunaswamy, S. (2023). Comprehensive analysis on vehicle-to-vehicle communication using intelligent transportation system. In: Ranganathan, G., EL Alloui, Y., Piramuthu, S. (eds) Soft Computing for Security Applications. ICSCS 2023. Advances in Intelligent Systems and Computing, Springer, Singapore. [https://doi.org/10.1007/978-981-99-3608-3\\_62](https://doi.org/10.1007/978-981-99-3608-3_62)
- [27] Mahendra, H.N., Mallikarjunaswamy, S., Kumar, D.M., Kumari, S., Kashyap, S., Fulwani, S., Chatterjee, A. (2023). Assessment and prediction of air quality level using ARIMA model: A case study of Surat City, Gujarat State, India. Nature Environment & Pollution Technology, 22(1): 199-210. <https://doi.org/10.46488/NEPT.2023.v22i01.018>
- [28] Umashankar, M.L., Mallikarjunaswamy, S., Sharmila, N., Kumar, D.M., Nataraj, K.R. (2023). A survey on IoT protocol in real-time applications and its architectures. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds) ICDSMLA 2021. Lecture Notes in Electrical Engineering, vol 947. Springer, Singapore. [https://doi.org/10.1007/978-981-19-5936-3\\_12](https://doi.org/10.1007/978-981-19-5936-3_12)
- [29] Mahendra, H.N., Mallikarjunaswamy, S., Subramoniam, S.R. (2023). An assessment of built-up cover using geospatial techniques—A case study on Mysuru District, Karnataka State, India. International Journal of Environmental Technology and Management, 26(3-5): 173-188. <https://doi.org/10.1504/IJETM.2023.130787>
- [30] Pooja, S., Mallikarjunaswamy, M., Sharmila, S. (2023). Image region driven prior selection for image deblurring. Multimedia Tools and Applications, 82: 24181-24202. <https://doi.org/10.1007/s11042-023-14335-y>
- [31] Rathod, S., Ramaswamy, N.K., Srikantaswamy, M., Ramaswamy, R.K. (2022). An efficient reconfigurable peak cancellation model for peak to average power ratio reduction in orthogonal frequency division multiplexing communication system. International Journal of Electrical and Computer Engineering, 12(6): 6239-6247. <http://doi.org/10.11591/ijece.v12i6.pp6239-6247>
- [32] Mallikarjunaswamy, S., Basavaraju, N.M., Sharmila, N., Mahendra, H.N., Pooja, S., Deepak, B.L. (2022). An efficient big data gathering in wireless sensor network using reconfigurable node distribution algorithm. In 2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP), Bengaluru, India, pp. 1-6. <http://doi.org/10.1109/CCIP57447.2022.10058620>
- [33] Mahendra, H.N., Mallikarjunaswamy, S. (2022). An efficient classification of hyperspectral remotely sensed data using support vector machine. International Journal of Electronics and Telecommunications, 68(3): 609-617. <http://doi.org/10.24425/ijet.2022.141280>
- [34] Shivaji, R., Nataraj, K.R., Mallikarjunaswamy, S., Rekha, K.R. (2022). Implementation of an effective hybrid partial transmit sequence model for peak to average power ratio in MIMO OFDM system. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds) ICDSMLA 2020. Lecture Notes in Electrical Engineering, vol. 783.

- Springer, Singapore. [https://doi.org/10.1007/978-981-16-3690-5\\_129](https://doi.org/10.1007/978-981-16-3690-5_129)
- [35] Savitha, A.C., Jayaram, M.N. (2022). Development of energy efficient and secure routing protocol for M2M communication. *International Journal of Performability Engineering*, 18(6): 426-433. <https://doi.org/10.23940/ijpe.22.06.p5.426-433>
- [36] Venkatesh, D.Y., Mallikarjunaiah, K., Srikantaswamy, M. (2022). A comprehensive review of low density parity check encoder techniques. *Ingénierie des Systèmes d'Information*, 27(1): 11-20. <https://doi.org/10.18280/isi.270102>
- [37] Thazeen, S., Mallikarjunaswamy, S., Saqhib, M.N. (2022). Septennial adaptive beamforming algorithm. In 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, pp. 1-4. <https://doi.org/10.1109/SIST54437.2022.9945753>
- [38] Mahendra, H.N., Mallikarjunaswamy, S., Basavaraju, N.M., Poojary, P.M., Gowda, P.S., Mukunda, M., Navya, B., Pushpalatha, V. (2022). Deep learning models for inventory of agriculture crops and yield production using satellite images. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, pp. 1-7. <https://doi.org/10.1109/MysuruCon55714.2022.9972523>
- [39] Mahendra, H.N., Mallikarjunaswamy, S., Nooli, C.B., Hrishikesh, M., Kruthik, N., Vakkalanka, H.M. (2022). Cloud based centralized smart cart and contactless billing system. In 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 820-826. <https://doi.org/10.1109/ICCES54183.2022.9835856>
- [40] Mallikarjunaswamy, S., Sharmila, N., Siddesh, G.K., Nataraj, K.R., Komala, M. (2022). A novel architecture for cluster based false data injection attack detection and location identification in smart grid. In: Mahanta, P., Kalita, P., Paul, A., Banerjee, A. (eds) *Advances in Thermofluids and Renewable Energy . Lecture Notes in Mechanical Engineering*. Springer, Singapore. [https://doi.org/10.1007/978-981-16-3497-0\\_48](https://doi.org/10.1007/978-981-16-3497-0_48)
- [41] Thazeen, S., Mallikarjunaswamy, S., Siddesh, G.K., Sharmila, N. (2021). Conventional and subspace algorithms for mobile source detection and radiation formation. *Traitement du Signal*, 38(1): 135-145. <https://doi.org/10.18280/ts.380114>
- [42] Satish, P., Srikantaswamy, M., Ramaswamy, N.K. (2020). A comprehensive review of blind deconvolution techniques for image deblurring. *Traitement du Signal*, 37(3): 527-539. <https://doi.org/10.18280/ts.370321>
- [43] Umashankar, M.L., Ramakrishna, M.V., Mallikarjunaswamy, S. (2019). Design of high speed reconfigurable deployment intelligent genetic algorithm in maximum coverage wireless sensor network. In 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, pp. 1-6. <https://doi.org/10.1109/IconDSC.2019.8816930>
- [44] Mahendra, H.N., Mallikarjunaswamy, S., Rekha, V., Puspallatha, V., Sharmila, N. (2019). Performance analysis of different classifier for remote sensing application. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(1): 7153-7158. <https://doi.org/10.35940/ijeat.A1879.109119>
- [45] Thazeen, S., Mallikarjunaswamy, S. (2023). The effectiveness of 6t beamformer algorithm in smart antenna systems for convergence analysis. *IJUM Engineering Journal*, 24(2): 100-116. <https://doi.org/10.31436/iiumej.v24i2.2730>