A Recursive Turbo Decoding Algorithm for Error Detection and Correction in IOT Networks

Ravi Solanki[1], Prof. Vipin Kasera[2]
Department of CSE, VITM, Indore[1,2]

Abstract:* The internet of things framework has seen a rapid change in terms of the applications and users worldwide. However, the need for trustworthiness to satisfactory quality of service is of utmost importance keeping in mind the nature of data transfer in wireless media. The advent of high compute power processors with miniature sizes and low power consumption, implementing relatively complex algorithms has become possible which is necessity for internet of things applications. This research paper focusses on the design and implementation of the code blocks of turbo codes based on the BCJR algorithm so as to couple the bits in the code blocks in the composite transport block. The information and parity bits are to be coupled so as to have more information sharing within the transport block and hence reduce the error rate steeply in section of the error waterfall. The proposed technique attains lower bit error rate performance compared to the conventional un-coded and hard coded counterparts. A comparative analysis with respect to the error rate has been done so as to evaluate the quality of service of the proposed work. The lower error rate of the proposed work ensures the high quality of service and trustworthiness of the IoT system.

Keywords: Internet of things, *Turbo Codes, trustworthiness, error rate, bit sharing.*

I. INTRODUCTION

One of the major challenges of the internet of things framework is the chances of bit flipsin the data to be sent. A typical IoT framework is depicted in figure 1.

Fig.1 The IoT framework

The IoT framework owing to the wireless or unguided media has to be designed such that it exhibits satisfactory quality of service [1]. The metrics may be considered to be:

1) Error Rate
2) Throughput
3) Latency

Most of the parameters though could be managed under at least one governing constraint which is [2]:

$$Data\ Rate_C \leq Capacity_C \qquad (1)$$

Here,
$Data\ Rate_C$ is the actual data rate through the channel.
$Capacity_C$ denotes the permissible channel capacity of the IoT network.

The IoT framework typically exhibits a steep fall in the waterfall region of the error curve and then a diminishing error rate [3].

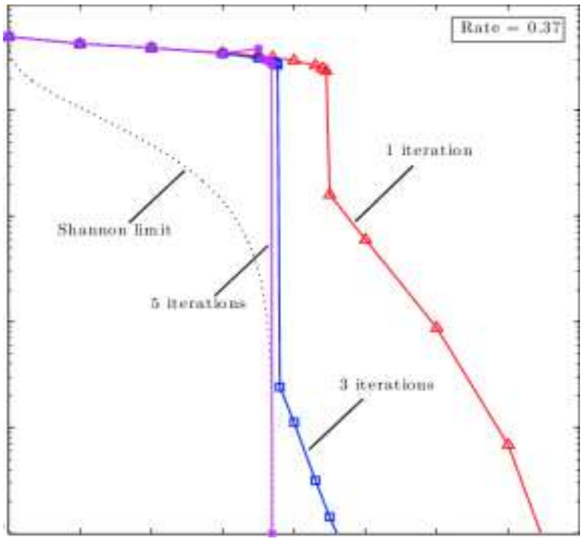


Fig.2 Typical error performance of Shannon's limit

The typical Shannon's limit is exhibited by a sharp fall in the bit error rate upto or beyond 10^{-5} for an SNR range of 0=10 dB [4]. Typically, the error drops as a function of the iterative decoding in several error detection and correction coding techniques [5]. One of the most effective error detection and correction mechanisms in this regard is the recursive turbo codes [6]-[7]. This category of codes show high adherence to the Shannon's limit [8]-[9]

The turbo encoding mechanism is typically described by the following attributes [10]:

1) Encoder
2) Decoder
3) Channel
4) Interleaver
5) De-interleaver
6) Recursive block

The encoding mechanism is depicted in figure 3.

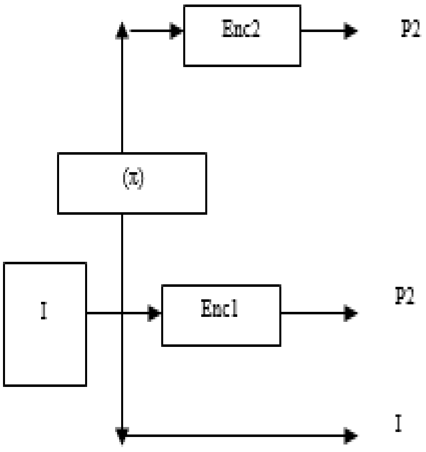


Fig.3 Turbo encoder

The turbo encoder is characterized by:
Input bits.
Parity bits
Here,
I represents the information bits
Interleaver π
Encoders

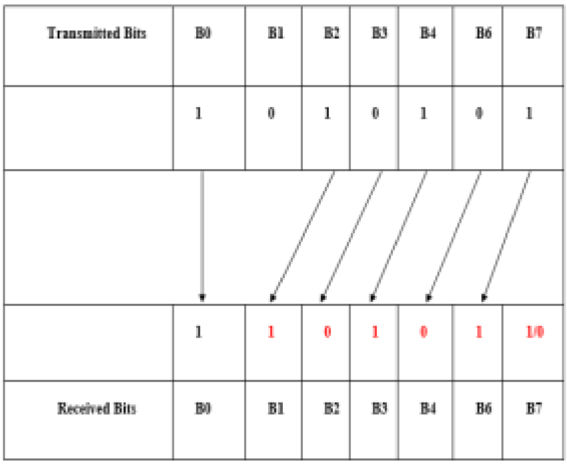
The encoding mechanism is typically performed in a way so as to enhance the reliability of the system [11]. This happens due to the fact that the encoder has three bits as the output for one bit as the input. The encoders are typically symmetric in nature or even asymmetric based on the type of encoder design [12]-[13]. The information bit shared is then passed on to render 3 bits which are [14]:

1) Same unaltered bit.
2) Encoded bit (P1)
3) Encoded bit (P2).

The difference among the bits P1 and P2 lie in the fact that both the bits are distinguished by the act of the interleaver. While the information bit 'I' directly goes to the encoder 1, the other encoder receives a modified version of the information bit [15]. The two encoders may or may not be similar. In case both exhibit a similar structure, the encoding is termed as symmetric encoding [16]. The role of the interleaver is exemplified in the next section.

II. INTERLEAVING AND PUNCTURING

The interleaving mechanism is fundamentally derived so as to reduce the burst errors in a network [17]. This can be understood through the following diagram.

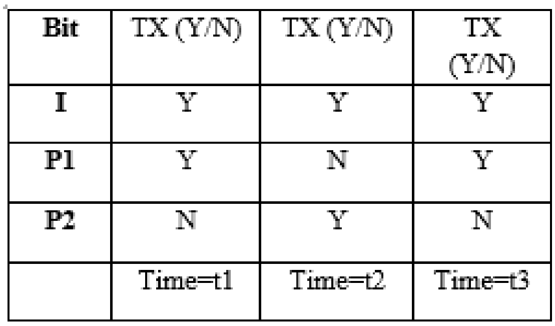


Transmitted Bits	B0	B1	B2	B3	B4	B6	B7
	1	0	1	0	1	0	1
	1	1	0	1	0	1	1/0
Received Bits	B0	B1	B2	B3	B4	B6	B7

Fig.4 Burst Errors

The interleaving mechanism is fundamentally used to circumvent the domino effect of errors [18]. This can be seen form figure 4. As there is a missing bit in bit location 2, there is an error in bit 2 which is received by the receiver. The error progresses as the receiver doesn't have cognizance of the transmitters bits. This leads to a cascading progression of the bits and hence the error in one bit results in the errors in other multiple bits. This however can be mitigated in case, the error propagation mechanism is stopped [19]. The exact is done by the interleaver as the interleaver combines the bits into chunks and separates the correlation among the bits. This is however, true only for burst errors with memory and not for random errors [20].

While burst errors are bits which have a cascading effect, the random errors are the errors which can occur at any bit location at any given instance of time [21].



Bit	TX (Y/N)	TX (Y/N)	TX (Y/N)
I	Y	Y	Y
P1	Y	N	Y
P2	N	Y	N
	Time=t1	Time=t2	Time=t3

Fig.5 Puncturing

The puncturing mechanism is based on the planned non-transmission of the bits at some intervals of time. The information bit is not omitted but one of the most common techniques is to omit the parallel transmission of both the parity bits. This reduces the bit transmission rate of the system [22].

While the original coding rate is 1/3, the new coding rate remains only ½. This happens due to the suppression of one bit at a time [23].

III. TURBO DECODING

The major challenges with error detection and correction for IoT networks are [24]:

1. IoT networks are prone to noise and disturbance effects causing increase in bit error rate of the system. This decreases the reliability and trustworthiness of the system.

2. Often IoT networks are resource constrained in terms of memory and processing power. Hence coding techniques with relatively low computational complexity in terms of number of iterations are needed.
3. Lesser iterations are also needed to minimize the latency (delay) of the system as IoT networks can be used for time critical applications.

4. There exists a fundamental trade off between the number of iterations and Bit Error Rate (BER) of the system where higher iterations would result in lower BER but would significantly increase the system's latency and complexity.

Typically two deciders are employed for decoding in the cascading manner [25]. The BCJR based algorithm is used for the decoding of the codes.

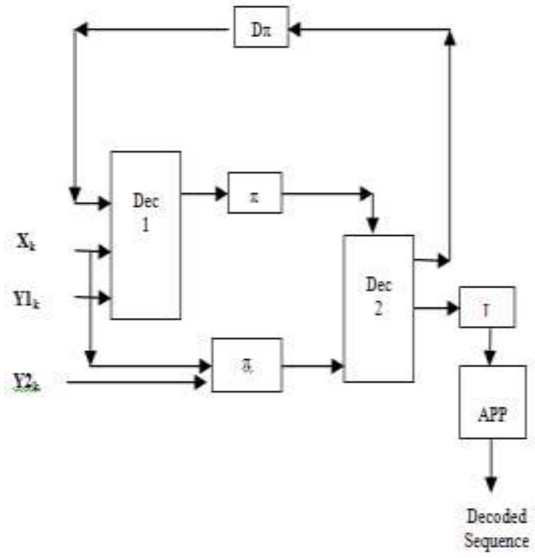


Fig.6 The turbo decoding mechanism

The figure above depicts the block diagram of the turbo decoder which comprises of two decoders. The interleaver is represented by (π) and the de-interleaver is represented by (Dπ).
The decoding process is done in a manner which incorporates both the decoders which are designated as D1 and D2. The information bit I and one of the parity bits is fed to decoders 1 and 2 respectively. Each of the decoders surmise the output based on the input information received, and the verdict of the other decider [26]. Thus the feedback loop connects decider 1 and decoder 2's outputs in a recursive manner in which the iterative process takes place in the decoding mechanism. At the beginning of the decoding process, the output of any one of the decoders is considered to be equi-probable probabilities of 1 or zero occurring. However, the final bit pattern is considered at the output terminal of D2 [27].

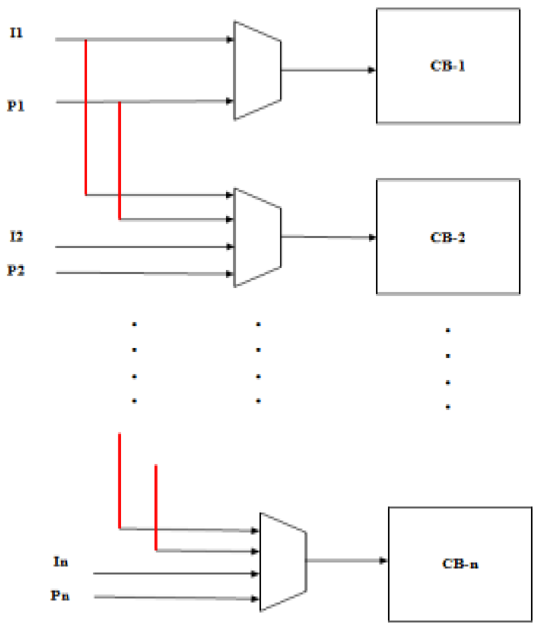


Fig.7 The proposed approach

In the proposed approach, both information and parity bits are coupled. Previous approaches do not have a method to couple both I & P.

In the proposed scheme, the information bits are designated by I and the interleaved bits are denoted by P. In this case, the n code blocks (CB) constitute a transport block (TB). The transport block vector (T) is segmented into 'n' code blocks

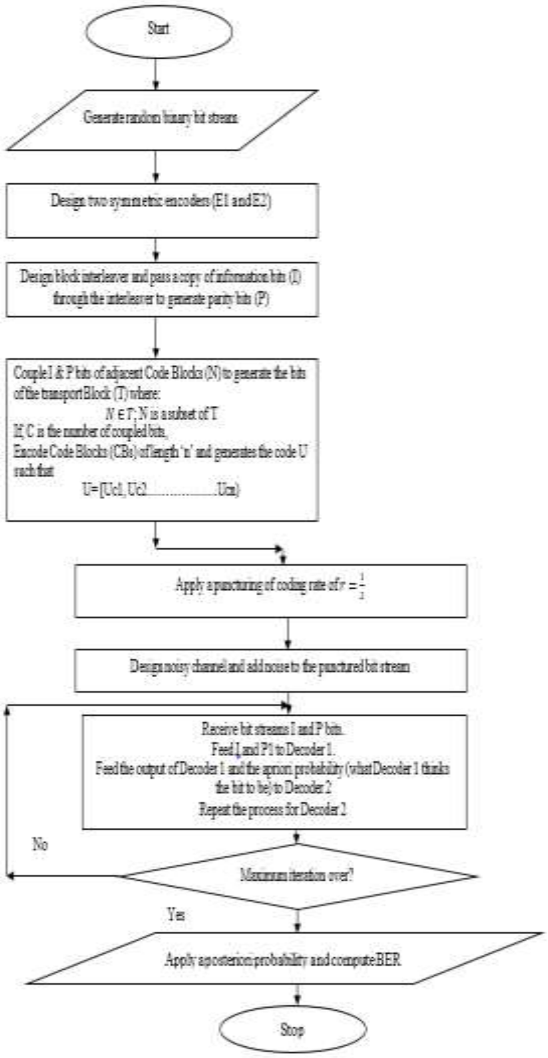

Start
Generate random binary bit stream
Design two symmetric encoders (E1 and E2)
Design block interleaver and pass a copy of information bits (I) through the interleaver to generate parity bits (P)
Couple I & P bits of adjacent Code Blocks (N) to generate the bits of the transport Block (T) where:
N ∈ T; N is a subset of T
If, C is the number of coupled bits,
Encode Code Blocks (CBs) of length 'n' and generates the code U such that
U=[Uc1, Uc2.....................Ucn)
Apply a puncturing of coding rate of r = 1/2
Design noisy channel and add noise to the punctured bit stream
Receive bit streams I and P bits.
Feed I and P1 to Decoder 1.
Feed the output of Decoder 1 and the apriori probability (what Decoder 1 thinks the bit to be) to Decoder 2
Repeat the process for Decoder 2
No
Maximum iteration over?
Yes
Apply aposteriori probability and compute BER
Stop

Fig.8 Proposed Flowchart

Figure 8 depicts the flowchart of the proposed system. The flowchart represents the sequential steps to implement the proposed system.

IV. EXPERIMENTAL RESULTS

The system has been designed on MATLAB 2020a. To emulate the actual data streams generated by a multitude of devices in an IoT network, random binary data has been generated.

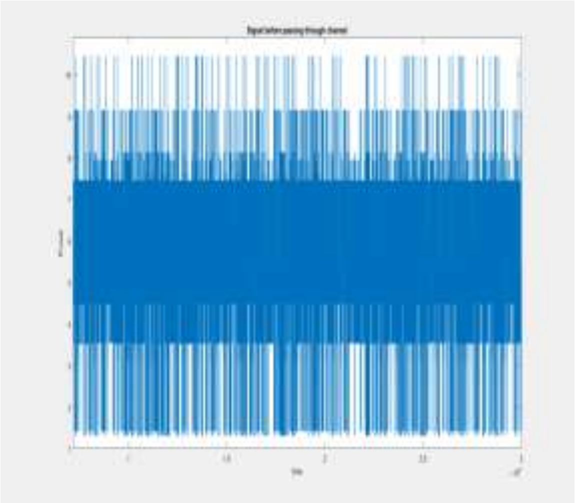


Fig.9 binary bits

Figure 9 depicts the binary data stream generated to emulate random binary data transmission.

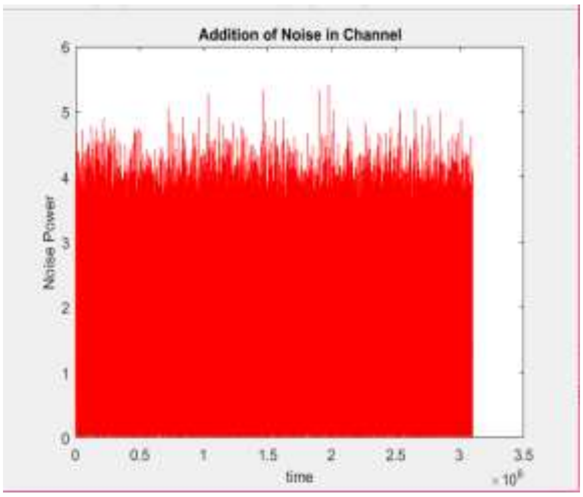


Fig.10 Addition of disturbance

Figure 10 depicts the addition of noise in the wireless channel. Random noise has been added so as to replicate the channel conditions in an actual IoT network. The random fluctuations in the noise as a function of time has been shown in the figure.

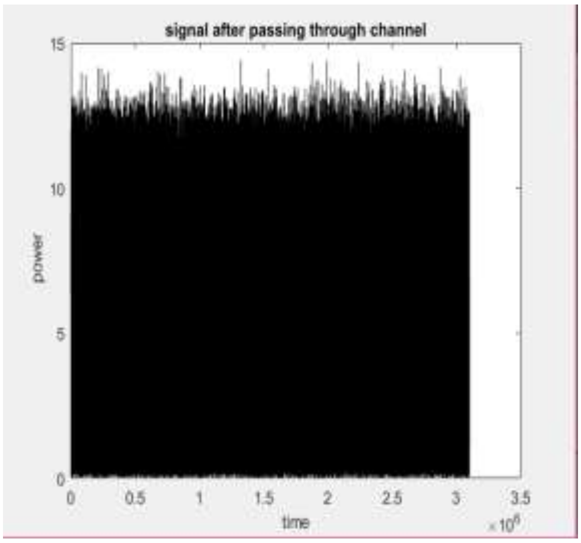


Fig.11 Effect of noise addition.

The effect of noise addition on the binary data stream in the time domain has been depicted in figure 11. It can be seen that the binary data stream has been manipulated by the addition of noise.

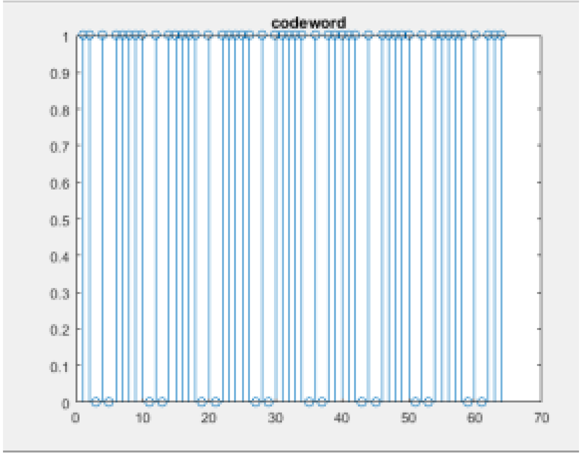


Fig.12 Formation of the turbo Code-word

Figure 12 depicts the binary code-word generated by the proposed system. The binary representation of the code word has been shown.

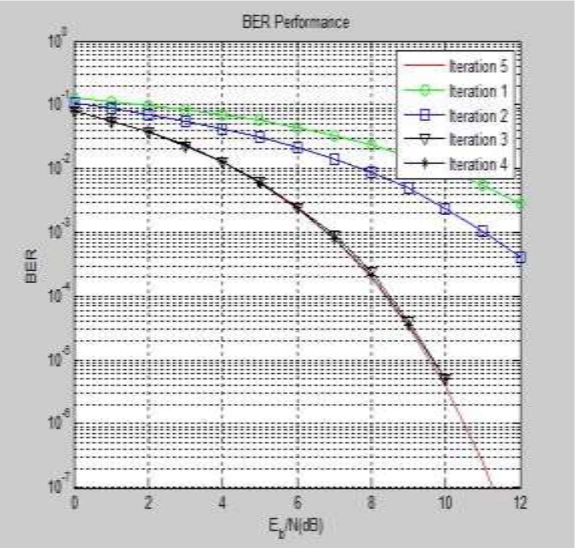


Fig.13 error as function of iterations.

Figure 13 depicts the bit error rate of the proposed system as function of iterations. It can be observed that as the iterations increase, the BER of the system continuously plummets. To represent the signal strength for binary data, the term energy per bit E_b has been used.

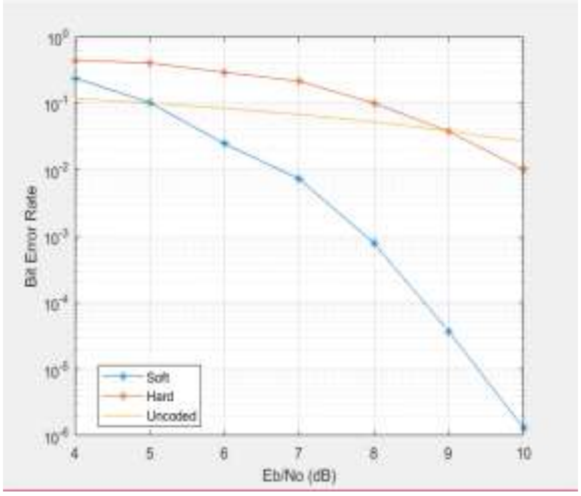


Fig.14 Coded and un-coded error rates

Figure 14 depicts the comparative analysis of the un-coded, hard coded and soft coded (Turbo) versions of binary transmission for the system. It can be clearly observed that the soft turbo coded probabilistic approach attains the steepest fall in the error rate compared to the hard coded and un-coded counterparts.

Conclusion:

Internet of Things (IoT) networks are finding applications in different domains such as automation, climate monitoring, defense etc. However, due to the wireless medium of the nature, noise and disturbance effects cause increase in the bit error rate of the system thereby causing low reliability and trustworthiness. The essence of the turbo coding mechanism lies in the fact of steep fall in errors near the relatively low SNR values. The proposed system implements coupling of both I and P bits in adjacent code blocks. To recreate a practical IoT scenario, noise is added to the bit stream. The experimental results are gauged in terms of the iterations to attain low values of the errors for the low Shannon range SNR. The BER obtained by the proposed system is 10-7 which is significantly less than the BER of the previous system 10-5 [1]. Thus it can be concluded that that the proposed system outperforms existing work in terms of the bit error rate (BER) which is the predominant metric for system reliability and trustworthiness.

References

[1] S. Zhao, J. Wen, S. Mumtaz, S. Garg and B. J. Choi, "Spatially Coupled Codes via Partial and Recursive Superposition for Industrial IoT With High Trustworthiness," in IEEE Transactions on Industrial Informatics 2022, vol. 16, no. 9, pp. 6143-6153.

[2] Chaofan Chen , Li Li, Li Wang, Shuai Wang 1, Xiangming Li 1, George K. Karagiannidis, “Noncoherent Detection With Polar Codes”, IEEE 2021.

[3] Saeedeh Moloudi, Michael Lentmaier, Alexandre Graell i Amat, “Spatially Coupled Turbo-Like Codes”, IEEE 2020.

[4] Suihua Cai, Nina Lin, and Xiao Ma, “Block Markov Superposition Transmission of BCH Codes with Iterative Erasures-and-Errors Decoders”, IEEE 2019.

[5] Gianluigi Liva, Lorenzo Gaudio, Tudor Ninacs and Thomas Jerkovits, “Code Design for Short Blocks: A Survey”, IEEE 2018.

[6] Erdal Arıkan, Daniel J. Costello, Jr., Joerg Kliewer, Michael Lentmaier, Paul Siegel, Ruediger Urbanke, Michael Pursley, “Guest Editorial Recent Advances in Capacity Approaching Codes”, IEEE 2017.

[7] Boulat A. Bash, Dennis Goeckel, Saikat Guha, Don Towsley, “Hiding Information in Noise: Fundamental Limits of Covert Wireless Communication”, IEEE 2015.

[8] Zunaira Babar, Soon Xin Ng and Lajos Hanzo, “EXIT-Chart Aided Near-Capacity Quantum Turbo Code Design”, IEEE 2015.

[9] Michael Lentmaiery, Saeedeh Moloudiy, and Alexandre Graell i Amat, “Braided Convolutional

Codes – A Class of Spatially Coupled Turbo-Like Codes”, IEEE 2014.

[10] Tsung-Yi Chen, Kasra Vakilinia, Dariush Divsalar, and Richard D. Wesel, “Protograph-Based Raptor-Like LDPC Codes”, IEEE 2014.

[11] A. Amat S. Moloudi and M. Lentmaier, “Spatially coupled Turbo codes: Principles and finite length performance,” IEEE 11th ISWCS, pp. 883– 887, 2014.

[12] T. Breddermann and V. Peter, “Rate-compatible insertion convolutional Turbo codes: Analysis and application to LTE,” IEEE Trans. Wireless Comm., vol. 13, no. 3, pp. 1356–1366, Mar. 2014.

[13] K Niu, K Chen, JR Lin, “Beyond turbo codes: Rate-compatible punctured polar codes”, IEEE 2013

[14] MM Wilde, MH Hsieh, Z Babar, “Entanglement-assisted quantum turbo codes”, IEEE 2013

[15] G Liva, E Paolini, B Matuz, S Scalise, “Short turbo codes over high order fields”, IEEE 2013

[16] K Anwar, T Matsumoto, “Accumulator-assisted distributed turbo codes for relay systems exploiting source-relay correlation”, IEEE 2012

[17] K Niu, K Chen, “CRC-aided decoding of polar codes”, IEEE 2012

[18] SH Lee, JK Kwon, “Turbo code-based error correction scheme for dimmable visible light communication systems”, IEEE 2012

[19] T. Breddermann and P. Vary. “EXIT functions for parallel concatenated insertion convolutional codes,” IEEE, 2011.

[20] MM Wilde, MH Hsieh, “Entanglement boosts quantum turbo codes”, IEEE 2011

[21] G Liva, S Scalise, E Paolini, “Turbo codes based on time-variant memory-1 convolutional codes over Fq”, IEEE 2011

[22] QT Dong, M Arzel, C Jego, “Stochastic decoding of turbo codes:, IEEE 2010

[23] M Cluzeau, M Finiasz, JP Tillich, “Methods for the reconstruction of parallel turbo codes”, IEEE 2010

[24] A. Jangid and P. Chauhan, "A Survey and Challenges in IoT Networks," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 516-521.

[25] R. G. Maunder, "A Fully-Parallel Turbo Decoding Algorithm," in IEEE Transactions on Communications, 2015, vol. 63, no. 8, pp. 2762-2775.

[26] Y. Jiang, S. Kannan, H. Kim, S. Oh, H. Asnani and P. Viswanath, "DEEPTURBO: Deep Turbo Decoder," 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2019, pp. 1-5.

[27] P Durga, B Yamuna, P Salija, “Implementation of low complex SOVA in GNU radio”, Procedia in Computer Science, Elsevier 2018, vol. 143, pp. 876-885.