

# Performance Comparison for Iterative Decoder using Max-Log-Map and Fully Parallel Turbo Decoding Algorithm

Ashritha.R.Shetty<sup>1</sup>, C.Y. Gopinath<sup>2</sup>, Dr Hemanth Kumar .A.R<sup>3</sup>

*PG Student [DEC], Dept. of ECE, Bangalore Institute of Technology, Bengaluru, Karnataka, India<sup>1</sup>*

*Associate Professor, Dept. of ECE, Bangalore Institute of Technology, Bengaluru, Karnataka, India<sup>2</sup>*

*Professor, Dept. of ECE, Bangalore Institute of Technology, Bengaluru, Karnataka, India<sup>3</sup>*

**Abstract**–Iterative decoder implementation for turbo codes is an demanding assignment. Several algorithms have been projected to facilitate the implementation of iterative decoder for turbo codes. This paper examines the implementation of an iterative decoder for turbo codes using the MAX-LOG-MAP algorithm and Fully parallel turbo decoding algorithm (FPTD). Despite the fact that the MAX-LOG-MAP practices turbo encoded bits in a serial forward-backward style, the proposed algorithm functions in a fully-parallel behaviour, processing all bits in both components of the turbo code at the same time. The FPTD algorithm is attuned with all turbo codes, including those of the LTE and WiMAX standards. BER performance among these two algorithms is envisaged.

**Index terms**– Turbo decoder, Iterative decoding, Bit error rate (BER), Long-term-evolution (LTE), WiMAX.

## I. INTRODUCTION

The last two decades has witnessed the revolution of wireless communication by channel codes that are assisted from iterative decoding algorithms. Most of the highly developed wireless communication standards agreed upon to turbo codes as the channel coding technique because of its close to Shannon error correcting performance. Nevertheless, optimal turbo decoding by means of BCJR (MAP) algorithm entails composite mathematical procedures and calculations due to which the system implementation is cumbersome in contrast to the decoding of other codes like convolution codes [1-3]. Consequently, cut down sub-optimal variations of the BCJR algorithms are by and large used for implementation. Such transformation of the BCJR algorithm is LOG-MAP algorithm and MAX-LOG-MAP algorithm.

Two vivid half iterations are carried out for the decoding procedure, where in the dependability of received bits is calculated as extrinsic standards using interleavers and soft-input-soft-output (SISO) decoders in an iterative approach. Even half iteration involves decoding process to be performed on the non-interleaved data and parity, where as odd half iteration involves interleaved data to be decoded. The extrinsic values which would denote the consistency of the

information bits are sent to one more half iteration by fleeting through the interleaver/deinterleaver unit until the satisfactory error stage is achieved.

Fully parallel turbo decoding (FPTD) has drastically amplified parallelism. It works in a completely parallel behavior. By distinction, the processing throughput of turbo decoders is narrowed down by the innately serial nature of the Log-BCJR algorithm which is obligated by the data dependence of its forward and backward recursions. This prompts the FPTD algorithm, which distributes with the recursions of the MAX-LOG-MAP algorithm and the associated data dependencies, making possible fully-parallel turbo decoding. Fully-parallel algorithm is well-suited with every turbo code, including those of the LTE and WiMAX standards [8].

In this paper, we present a comprehensive implementation of FPTD and MAX-LOG-MAP based iterative turbo decoder along with their performance being evaluated. Here section II relates to turbo encoder giving brief description about its operation and gives operation of both max-log-map ad FPTD algorithms. Section III and IV gives results and conclusion.

## II. IMPLEMENTATION OF ITERATIVE TURBO DECODER

### A. Turbo Encoder

In Fig.1, it is evident that the feed-through passes through one block of  $K$  information bits, which are called systematic bits denoted by  $x^s_k$ , where  $k = 0, 1, \dots, K - 1$ . The parity engendered by the convolutional encoder is denoted by  $x^{p1}_k$ . By permuting the systematic bits through the interleaver, the second sequence of parity is generated which passes through the second convolutional encoder. This is denoted by  $x^{p2}_k$ . The RSC encoder used in turbo code are identical with generator matrix  $g_0(D)=1+D^2+D^3$  and  $g_1(D)=1+D+D^3$ , where  $g_0$  and  $g_1$  are feedback and feed forward polynomials. The transfer function of each encoder is given by  $G(D)=\begin{bmatrix} 1, & g_1(D) \\ & g_0(D) \end{bmatrix}$

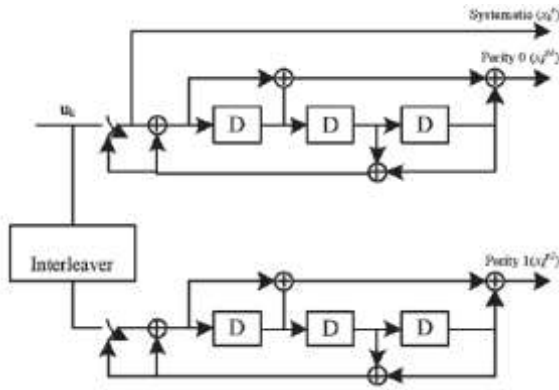


Fig1: Rate=1/3 Turbo encoder

**B. Turbo decoder**

**1. MAX-LOG-MAP**

Design for the iterative decoder for the frame size of 1024 bits and 6 iterations is involved in the process. The iterative decoder involves MAX-LOG-MAP based two soft-input soft-output (SISO) decoders, matrix interleaver and matrix deinterleaver as shown in Fig. 2.

On the receiver side, the steadfastness of the bits is calculated iteratively by switching over the extrinsic LLRs among the two SISO decoders based on eq.(1)[5].

$$L(u_k|y) = \log \left[ \frac{p(u_k = +1|y)}{p(u_k = -1|y)} \right] \quad (1)$$

where  $p(u_k = +1|y)$  and  $p(u_k = -1|y)$  symbolize the probabilities of bit  $u_k$  being +1 and -1 correspondingly. The MAP algorithm, which provides the a posteriori probability for every bit, is used in iterative decoding of turbo codes[5]. The MAP algorithm offers the probability of the decoded bit  $u_k$  being either +1 or -1 for the received symbol sequence  $y$  by estimation of the LLR values as (1).

The LLR computation begins soon after first value of  $\beta$  is accessed. LLR calculation using  $\alpha$ ,  $\beta$  and  $\gamma$  values is given as, where  $\alpha$ ,  $\beta$  and  $\gamma$  are forward, backward and branch metrics[7].

$$LLR = \max_{X_k=1} (\alpha_{k-1}(S_{k-1}) + \gamma_{k-1}(S_{k-1}, S_k) + \beta_{k-1}(S_k)) - \max_{X_k=0} (\alpha_{k-1}(S_{k-1}) + \gamma_{k-1}(S_{k-1}, S_k) + \beta_{k-1}(S_k)) \quad (2)$$

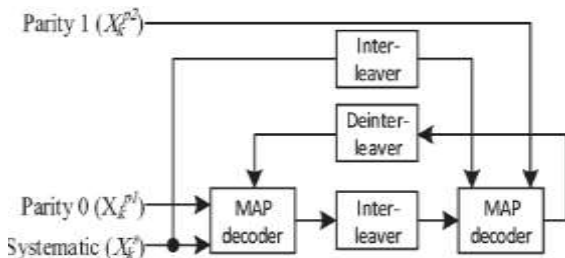


Fig 2: Turbo decoder for max-log-map decoder.

where  $S_k$  is the trellis current state at time  $k$  and  $S_{k-1}$  is the trellis previous state at that instant. Unambiguous storage of these LLR values is not needed since they are restructured and stored in the matrix interleaver or deinterleaver.

In (2),  $\gamma_k$  is the branch metric for the received information bit  $x_k$  at time  $k$  and the current trellis state  $S_k$ , eloquent about the state from which the between branch came was  $S_{k-1}$ .  $\alpha_{k-1}$  is the forward metric computed recursively after the calculation of  $\gamma$  at each stage.  $\beta_{k-1}$  is the backward metric whose calculation starts after  $\alpha$  calculation is ended. Forward metric ( $\alpha$ ) and backward metric ( $\beta$ ) calculations are distinct to higher priority tasks[7].

$$\alpha_{k+1}(S_{k+1}) = \max(\alpha_k(S_k) + \gamma_{k+1}(S_k, S_{k+1})) \quad (3)$$

$$\beta_{k-1}(S_{k-1}) = \max(\beta_k(S_k) + \gamma_k(S_{k-1}, S_k)) \quad (4)$$

The initial conditions are  $\alpha_0(S) = 0$  if  $S = 0$ , and

$\alpha_0(S) = -\infty$  or else. Similarly,  $\beta_N(S) = 0$  if  $S = 0$  and  $\beta_N(S) = -\infty$  or else. Based on the LLR value, the message is decoded.

**2. Fully parallel turbo decoder**

The dispensation throughput of turbo decoders is restricted by the intrinsically serial character of the Max-Log-Map algorithm, which is obligated by the data dependencies of its forward and backward recursions. This stimulates the novel turbo decoder algorithm, which dispenses with the recursions of the Max-Log-Map algorithm and the allied data dependencies, facilitating fully-parallel turbo decoding. More particularly, the planned fully-parallel turbo decoder algorithm is competent of dealing out with all bits corresponding to the mutual convolution codes at the same time[8].

Fig.3 demonstrates fully parallel turbo decoder. Subsequent to their communication over a wireless channel, the three encoded frames  $b_1^u$ ,  $b_2^u$  and  $b_3^u$  might be demodulated and supplied to the turbo decoder of Fig.3.

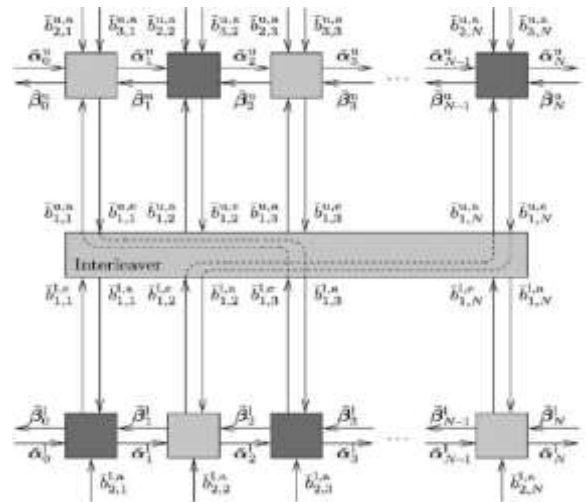


Fig 3: Fully Parallel Turbo Decoder.

Nonetheless, due to the outcome of noise in the wireless channel, the demodulator would be doubtful of the bit values in these encoded frames. As a result, as an alternative of providing frames encompassing  $N$  hard-valued bits, the demodulator offers three frames each including  $N$  soft-valued *a priori* Logarithmic Likelihood Ratios (LLRs)  $\bar{b}_2^{u,a} = [\bar{b}_{2,k}^{u,a}]_{k=1}^N$ ,  $\bar{b}_3^{u,a} = [\bar{b}_{3,k}^{u,a}]_{k=1}^N$ ,  $\bar{b}_2^{l,a} = [\bar{b}_{2,k}^{l,a}]_{k=1}^N$ . Here, LLR pertaining to the bit  $b_{j,k}$  is defined by

$$\bar{b}_{j,k} = \ln \frac{\Pr(b_{j,k} = 1)}{\Pr(b_{j,k} = 0)} \quad (5)$$

Where the superscripts “a,” “e,” or “p” may be appended to point towards an *a priori*, extrinsic or *a posteriori* LLR in that order [8]. The demodulator supplies these *a priori* LLRs to the fully parallel turbo decoder’s  $2N$  algorithmic blocks, which are shown in Fig.3 arranged in two rows. More exclusively, the *a priori* parity LLR  $\bar{b}_{2,k}^{u,a}$  and the *a priori* systematic LLR  $\bar{b}_{3,k}^{u,a}$  are provided to the  $k$ th algorithmic block in the upper row shown in Fig.3. Besides, the interleaver of Fig.3 provides the  $k$ th algorithmic block in the upper row with the *a priori* message LLR  $\bar{b}_{1,k}^{u,a}$ . In the intervening time, the  $k$ th algorithmic block in the lower row is correspondingly provided with the *a priori* LLRs  $\bar{b}_{1,k}^{l,a}$  and  $\bar{b}_{2,k}^{l,a}$ . Note that the algorithmic blocks in the lower row of Fig.3 are not provided with any *a priori* systematic LLRs, hence eradicating the requirement to interleave  $\bar{b}_3^{u,a}$ . In addition to the above mentioned LLRs, the  $k$ th algorithmic block in every row is as well made available with a vector of *a priori* forward state metrics  $\bar{\alpha}_{k-1} = [\bar{\alpha}_{k-1}^{(S_k-1)}]_{S_k=1}^{M-1}$  and a vector *a priori* backward state metrics  $\bar{\beta}_k = [\bar{\beta}_k^{(S_k)}]_{S_k=0}^{M-1}$ , as will be detailed below. Every algorithmic block drives in an indistinguishable mode by means of the equations provided in (1)–(4). The proposed fully-parallel turbo decoding algorithm. These equations are assured in an entirely comprehensive manner, allocating them to be functional to any turbo code, having any state transition diagram and any number  $L$  of *a priori* LLRs per algorithmic block. The projected fully-parallel turbo decoder is activated iteratively, where each of the  $I$  iterations contains the manoeuvre of all algorithmic blocks shown in Fig.3. The turbo decoder might be measured to be fully-parallel, since each iteration is completed within just  $T = 1$  time period, by operating all  $2N$  of the algorithmic blocks simultaneously. In common, the extrinsic data formed by each algorithmic block in Fig.3 is swapped over with those provided by the connected algorithmic blocks, to be used as *a priori* information in the next decoding iteration [11].

More specifically, the  $k$ th algorithmic block in each row provides the vectors of extrinsic state metrics  $\bar{\alpha}_k$  and  $\bar{\beta}_{k-1}$  for the neighbouring algorithmic blocks to make use of in the

next decoding iteration. Furthermore, the  $k$ th algorithmic block in each row passes the extrinsic message LLR  $\bar{b}_{1,k}^e$  through the interleaver, to be used as an *a priori* LLR by the associated block in the other row during the next decoding iteration. In the meantime, this block in the other row provides an extrinsic message LLR which is used as the *a priori* message LLR  $\bar{b}_{1,k}^a$  for the duration of the next decoding iteration. Note that the interleaver of Fig.3 may be hard-wired if only a single interleaver pattern is essential or it may adopt a reconfigurable design so as to hold up any random interleaver pattern. At the start of the first decoding iteration conversely, no extrinsic information is accessible. In this case, the  $k$ th algorithmic block in each row employs zero values for  $\bar{\alpha}_k$ ,  $\bar{\beta}_{k-1}$ ,  $\bar{b}_{1,k}^a$ . As an immunity to this, however, the first algorithmic block in the each row employs  $\bar{\alpha}_0 = [0, -\infty, -\infty, \dots, -\infty]$  throughout all decoding iterations, as the convolutional encoders at all times start on from an initial state of  $S_0 = 0$ . In the same way, the final algorithmic block from the all rows makes use of  $\bar{\beta}_N = [0, 0, 0, \dots, 0]$  all the way through all decoding iterations, given that the final state of the convolutional encoders  $S_N$  is not known in progress to the receiver, when cessation is not employed. Subsequent to the conclusion of the final decoding iteration, an *a posteriori* LLR belonging to the  $k$ th message bit  $\bar{b}_{1,k}^u$  may be acquired as  $\bar{b}_{1,k}^{u,p} = \bar{b}_{1,k}^{u,e} + \bar{b}_{2,k}^{u,a}$ . An inference of the message bit  $\bar{b}_{1,k}^u$  may then be obtained as the upshot of the binary test  $\bar{b}_{1,k}^{u,p} > 0$ .

### III. RESULTS AND SIMULATIONS

In this paper we first examine the performance and complexities of the Max-Log-Map and FPTD algorithms. The output of the implementation was monitored using MATLAB tool. For BER measurement the output of turbo encoder is BPSK modulated and then transmitted over AWGN channel. The simulation comes about demonstrate that the turbo code is an effective error correcting coding system under SNR situations.

- When BER and the iterations is considered, it is shown that more cycle will get bring down BER, however interpreting delay will be more.
- At the point when frame size is considered, turbo code with frame size will show signs of improvement execution.
- At the point when code rate is viewed as, the higher the coding rates needs more transmission capacity.
- Expanding the quantity of iterations is very little help in low areas of SNR.
- In center to high regions of SNR expanding the quantity of iterations builds the execution of the turbo codes.
- Intricacy in FPTD calculation is lessened to half as that of Max-Log-Map calculation as it is parallel operation.

- Less augmentations and subtractions is considered in FPTD than contrasted with Max-Log-Map calculation.

BER performance and efficiency of iterative decoder using Max-Log-Map and FPTD algorithm for different number of iterations is shown in Fig 4, Fig 5, Fig 6 and Fig 7 respectively.

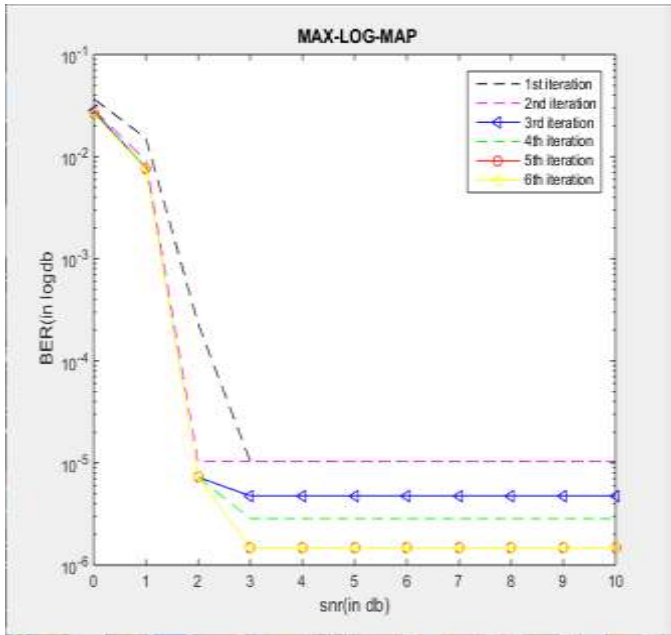


Fig 4: BER performance of Max-Log-Map algorithm for frame size=1024 bits and for 6 iterations.

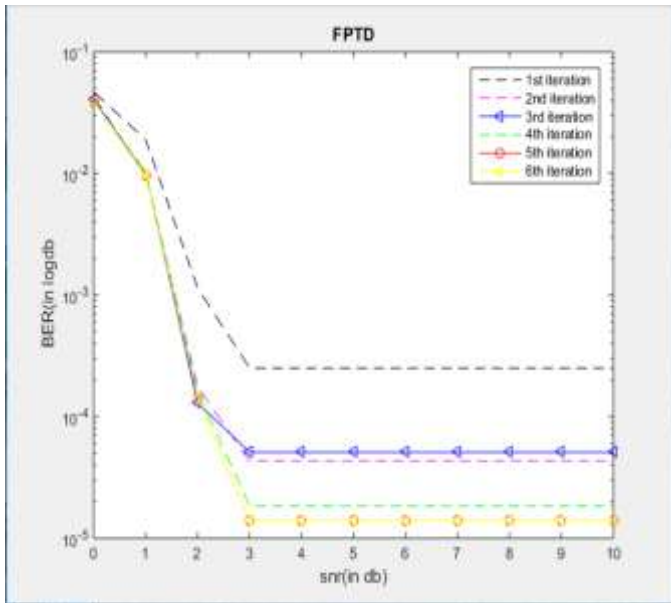


Fig 5: BER performance of FPTD algorithm for frame size=1024 bits and 6 iterations.

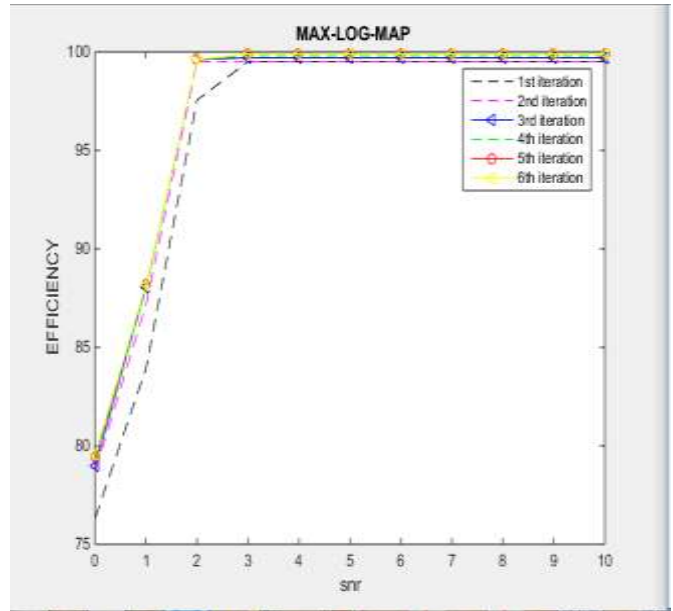


Fig 6: Efficiency of Max-Log-Map algorithm.

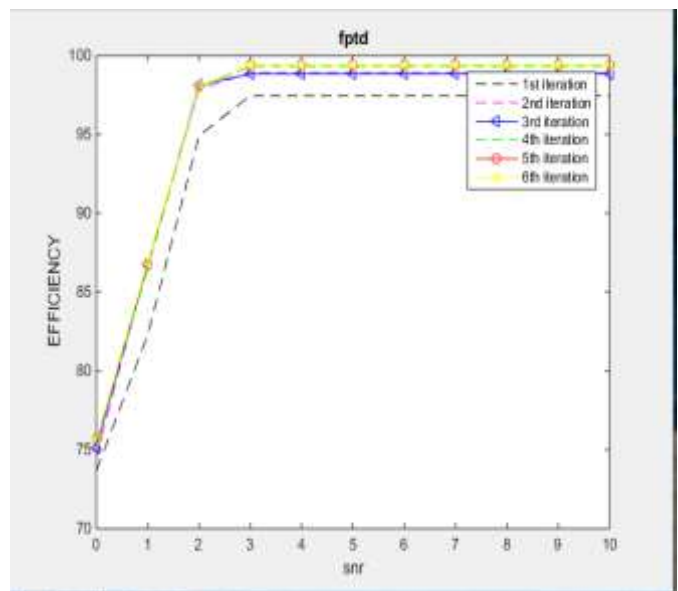


Fig 7: Efficiency of FPTD algorithm.

#### IV. CONCLUSION

In this paper, we have discussed the design and implementation of Max-Log-Map and FPTD algorithm based iterative decoder. The FPTD algorithm which eliminates the data dependencies of the Max-Log-Map algorithm and facilitates fully parallel operation. FPTD algorithm is suitable with both LTE and WiMAX standards. The interleaver used is QPP interleaver. The throughput of FPTD is superior compared Max-Log-Map algorithm. But FPTD takes more number of iterations to converge to same BER as that of Max-Log-Map whereas time taken by Max-Log-Map is more compared to FPTD.

## REFERENCES

- [1]. L.R Bahl, I. Cocke, F. Jelinek, J. Raviv, "Optimal decoding of linear block codes for minimizing symbol error rate", IEEE transaction on Information Theory, vol. 20, pp. 284- 287, 1974.
- [2]. C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon limit errorcorrecting coding and decoding: Turbo codes", IEEE International Conference on Communications, ICC'93, vol. 2, pp. 1064–1070, May 1993.
- [3]. Shu Lin, Daniel J. Costello Jr., "Error Control Coding", 2nd Edition Prentice Hall, 2004.
- [4]. M. Jordan. "Turbo code codec Implementation and Performance in a Software Radio", in proceeding of IEEE Military Communications Conference (MILCOM), vol.1, pp. 525-529, 1999.
- [5]. Arash Ardakani and Mahdi Shabany. "A Novel Area-Efficient VLSI Architecture for Recursion Computation in LTE Turbo Decoders", IEEE international, june 2015.
- [6]. Y Lin, S Mahlke, T Mudge, C. Chakrabarti, A. Reid, K. Flautner, "Design and Implementation of Turbo Decoders for Software Defined Radio", IEEE Workshop on Signal Processing Systems Design and Implementation proceeding, SIPS '06, pp. 22 – 27, 2006.
- [7]. Shivshankar Mishra, Anjali Mishra. "FPGA Implementation of Iterative Decoder for TurboCodes for Software Defined Radio" , IEEE conference, 2016
- [8]. Robert G. Maunder, Senior Member, IEEE, "Fully parallel turbo decoding algorithm", 2015.
- [9]. LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding, ETSI TS 136 212 V12.0.0 ed., 2013.
- [10]. Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Broadband Wireless Access Systems, IEEE Std. 802.16-2012, Aug. 2012.
- [11]. C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," IEEE Trans. Commun., vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

## BIOGRAPHY



**Ashritha.R.Shetty** received her B.E degree in E&C from Visvesvaraya Technological University, Belagavi, India. Currently she is pursuing M.Tech in Digital Electronics and Communication Engineering at BIT, Bangalore. Her area of interest includes Wireless Communication and Embedded

Systems.



C.Y. Gopinath received his B.E in E&C from U.V.C.E, Bangalore, in 1985. He specialized in Industrial Electronics from S.J.C.E, Mysore in 1989. He is presently working as Associate Professor in the Department of Electronics and Communication Engineering at BIT, Bangalore. He is currently research Scholar in Visvesvaraya Technological University. His area of interest includes Computer Architecture, Wireless Communication and Coding Techniques. He has published 5 international papers in various indexed journals, and national conferences.



**Dr Hemanth Kumar A. R** is presently working as Professor & Principal Investigator R&D,ECE, at BIT, Bangalore. He completed his B.E in E&C from Manipal Institute of Technology in the year 1999. He is specialized in

Digital Communication from B.M.S College of Engineering in 2003 and received his PhD from Manipal University in 2011. His area of interest includes MANETS, Clustering the network, Path Planning and Routing algorithms. He has published 29 international papers in various indexed journals, and national conference in India, Singapore, Thailand.