Effect of Constraint Length and Code Rate on the Performance of Enhanced Turbo Codes in AWGN and Rayleigh Fading Channel

T.GNANASEKARAN and V.AARTHI
Department of Electronics and Communication Engineering
Bannari Amman Institute of Technology
Sathyamangalam, Erode District, TamilNadu, India.
t.gnanasekaran@gmail.com, aarthivellingiri@gmail.com

Abstract: - Turbo coding (TC) has been adopted as a channel coding scheme for several 3G mobile systems, in particular 3GPP (Third Generation Partnership Project) and in upcoming 4G standards for high data rates. Turbo decoder uses Maximum A posteriori Probability (MAP), or Soft Output Viterbi Algorithm (SOVA) because it produces error correction near to Shannon’s limit. A simple but effective technique to improve the performance of the decoding algorithms is to scale the extrinsic information exchanged between two decoders. Modified Log MAP (MMAP) and Modified SOVA (MSOVA) algorithms are achieved by fixing an arbitrary value of scaling factor for inner decoder ($S_2$) and an optimized value for the outer decoder ($S_1$). We proposed to enhance the performance of MMAP and MSOVA by optimizing both the scaling factors $S_1$ and $S_2$, thus achieving low bit error rate (BER). This paper investigates the effects of constraint length and code rate on the performance of the enhanced Turbo codes. A comprehensive analysis of the algorithms considering different channel conditions and iterations are also presented.

Key-Words: - Constraint Length, Extrinsic information, MAP, Scaling Factor, SOVA, Turbo codes.

1 Introduction
A major advancement in the channel coding area was introduced by Berrou et al in 1993 by the advent of Turbo codes [1]. TC has shown the best Forward Error Correction (FEC) performance till date. They are revolutionary in the sense that they allow reliable data transmission within a half decibel of the Shannon Limit. A massive amount of research effort has been performed to facilitate the efficiency of TC. Thus TC have been incorporated into many standards used by the NASA Consultative Committee for Space Data Systems (CCSDS) [2], Digital Video Broadcasting (DVB) [3], both Third Generation Partnership Project (3GPP) [4] standards for IMT-2000, Wideband CDMA which requires throughputs from 2 Mb/s to several 100 Mb/s, in 4G and WIMAX.

Two iterative decoding algorithms, Soft Output Viterbi Algorithm [5], [6], [7] and Maximum A posteriori Probability [6], [8] algorithm require complex decoding operations over several iteration cycles. The relative complexity of the decoding algorithms depends on the constraint length. Hence for real time implementation of TC, reducing the decoder complexity while preserving BER performance is an important design consideration.

To overcome the above drawbacks, we present an analysis on the effect of constraint length and code rate on the performance of the proposed decoding algorithms.

2 Turbo Encoder
A basic turbo encoder is a recursive systematic encoder that employs two convolutional encoders in parallel, where the second encoder is preceded by an interleaver and is shown in Fig.1. The interleaver is usually selected to be a pseudo random interleaver that reorders the bits in the information sequence before being fed to the second encoder. The use of

Fig.1 Turbo Encoder
interleaver in conjunction with two encoders result in code words that have relatively few nearest neighbors. This makes the code word relatively sparse. Hence the coding gain achieved by a turbo code is due to the reduction in the number of nearest neighboring code words that result from interleaving. It is observed that the nominal rate at the output of the turbo encoder is 1/3. This increases the redundant bits and hence the error probability decreases.

The RSC component codes shown in the Fig.1 are k=5 (constraint length) code with generator polynomials G_0=31 and G_1=17. These generator polynomials are optimum in terms of maximizing the minimum free distance of the component codes.

### 3 Turbo Decoder

In a typical Turbo decoding system shown in Fig.2, two decoders (DC 1 and DC 2) operate iteratively and pass their decisions to each other after each iteration. These decoders produce soft outputs to improve the decoding performance. Such a decoder is called a SISO decoder [12]. Each decoder operates not only on its own input but also on the other decoder’s incompletely decoded output which resembles the operation principle of turbo engines. This analogy between the operation of the Turbo decoder and the turbo engine gives this coding technique its name, “Turbo codes”. Encoded information sequence X_k is transmitted over the channel, and a noisy received sequence Y_k is obtained. Each decoder calculates the Log Likelihood Ratio (LLR) for the k-th data bit d_k, as

\[
L(d_k) = \log \left[ \frac{P(d_k = 1|Y)}{P(d_k = 0|Y)} \right]
\]

LLR can be decomposed into 3 independent terms,

\[
L(d_k) = L_{apriori}(d_k) + L_e(d_k) + L_c(d_k)
\]  (2)

where \(L_{apriori}(d_k)\) is the a-priori information of \(d_k\), \(L_e(d_k)\) is the channel measurement, \(L_c(d_k)\) is the extrinsic information. Extrinsic information [13] from one decoder becomes the a-priori information for the other decoder at the next decoding stage. LLRs can be calculated by two different SISO algorithms MAP and SOVA.

#### 3.1 MAP Algorithm

The MAP algorithm [8], [14] is an optimal but computationally complex SISO algorithm. The Log MAP and Max Log MAP algorithms are simplified versions of the MAP algorithm. MAP algorithm calculates LLRs for each information bit as

\[
L(d_k) = \log \left[ \frac{\sum_{s_k} \gamma_k(s_k) \alpha_k(s_k) \beta_k(s_k)}{\sum_{s_k} \gamma_k(s_k) \alpha_k(s_k) \beta_k(s_k)} \right]
\]  (3)

where \(\alpha\) is the forward state metric, \(\beta\) is the backward state metric, \(\gamma\) is the branch metric, and \(S_k\) is the trellis state at trellis time k. Forward state metrics are calculated by a forward recursion from trellis time k = 1 to k = N where N is the number of information bits in one data frame. Recursive calculation of forward state metrics is performed as

\[
\alpha_k(s_k) = \sum_{j=0}^{1} \alpha_{k-1}(s_{k-1}) \gamma_j(s_{k-1}, s_k)
\]  (4)

Similarly, the backward state metrics are calculated by a backward recursion from trellis time k = N to k = 1 as

\[
\beta_k(s_k) = \sum_{j=0}^{1} \beta_{k+1}(s_{k+1}) \gamma_j(s_k, s_{k+1})
\]  (5)

Branch metrics are calculated for each possible trellis transition as

\[
\gamma_j(s_{k-1}, s_k) = A_k P(s_k|s_{k-1}) \exp \left[ \frac{2}{N_i} (y_k^s i_x + y_k^p i_x) \right]
\]  (6)

where \(i = (0, 1)\), \(A_k\) is a constant, \(x_k^s\) and \(x_k^p\) are the encoded systematic data bit and parity bit, and, \(y_k^s\) and \(y_k^p\) are the received noisy systematic data bit and parity bit respectively.
3.2 Log MAP Algorithm

To avoid complex mathematical calculations of MAP decoding, computations can be performed in the logarithmic domain [8]. Furthermore, logarithm and exponential computations can be eliminated by the following approximation

$$\max'(x,y) = \ln(e^x + e^y) = \max(x,y) + \ln(1 + e^{y-x}) \quad (7)$$

The last term in max* operation can easily be calculated by using a look-up table (LUT). So (3)-(6) become

$$L(d_k) = \max_{(s_{k-1}, s_k)} \left[ \sum_i \left( s_i(i, s_k) + s_{k-1}(s_k, s_{k-1}) + \tilde{p}_i(s_k) \right) \right]$$

$$- \max_{(s_{k-1}, s_k, d)} \left[ \sum_i \left( s_i(i, s_k) + s_{k-1}(s_k, s_{k-1}) + \tilde{p}_i(s_k) \right) \right] \quad (8)$$

$$\bar{s}_k(s_k) = \max_{s_{k-1}} \left[ \sum_i \left( s_i(i, s_k) + s_{k-1}(s_k, s_{k-1}) + \tilde{p}_i(s_k) \right) \right] \quad (9)$$

$$\tilde{p}_k(s_k) = \max_{s_{k+1}} \left[ \sum_i \left( s_i(i, s_k) + s_{k+1}(s_k, s_{k+1}) + \tilde{p}_i(s_k) \right) \right] \quad (10)$$

$$\sum_i \left( s_i(i, s_k) + s_{k-1}(s_k, s_{k-1}) + \tilde{p}_i(s_k) \right) + \log(P(s_k | s_{k-1}))$$

$$+ \log(P(s_k | s_{k-1})) + K \quad (11)$$

where K is a constant.

3.3 SOVA Algorithm

In this section [10], we explain a variation of the Viterbi algorithm, referred to as the SOVA. SOVA has two modifications over the Viterbi algorithms. The path metrics used are modified to take account of a-priori information when selecting the maximal likelihood path through the trellis. Another modification is made so that it provides a soft output in the form of the a-posteriori LLR \( L(d_k \| y) \) for each decoded bit. \( S'_k \) gives the states along the surviving path at state \( S_k = S \) in the trellis. The probability that this is the correct path through the trellis is given by

$$P(s'_k \| y_{j:k}) = \frac{P(s'_k \wedge y_{j:k})}{P(y_{j:k})} \quad (12)$$

The iterative decoding of Turbo codes uses the a-priori information from a component decoder. It is independent of the channel outputs used by that decoder. The extrinsic LLR \( L_e(u_k) \) for the bit \( u_k \) uses all the available received parity bits and all the received systematic bits except the received values \( y_k^r \) associated with \( u_k \). The systematic bits are also used by the other component decoder, which is the interleaved or deinterleaved version of \( L_e(u_k) \) as its a-priori LLRs. The a-priori LLRs \( L_e(u_k) \) are not truly independent from the channel outputs. The extrinsic LLR \( L_e(u_k) \) is affected by the received systematic bit relatively close to the bit \( u_k \). When LLR \( L_e(u_k) \) is used as the a-priori LLR by the other component decoder, the iterative decoding provides good results. When calculating the LLR of the bit \( u_k \), SOVA must take into account of the probability that the paths merging with the ML path from stage \( k \) to stage \( k+\delta \) in the trellis were incorrectly discarded. This is done by considering the values of the metric difference \( \Delta_i^k \) for all states \( s \) along the ML path from trellis stage \( i=k \) to \( i=k+\delta \) The LLR can be approximated by

$$L(u_k | y) \approx u_k \min_{u_k, u_k+\delta} \Delta_i^k$$

where \( u_k \) is the value of the bit given by the ML path, and \( u_k^i \) is the value of this bit for the path which merged with the ML path and was discarded at trellis stage \( i \). Thus the minimization in [8] is carried out only for those paths merging with the ML path which would have given a different value for the bit \( u_k \) if they had been selected as the survivor path. The path which merges with the ML path, but would have given the same value for \( u_k \) as the ML path, obviously do not affect the reliability of the decision \( u_k \).

4 Enhanced Turbo Decoder

The SOVA and Log MAP algorithms suffer from two distortions: over optimistic soft outputs and correlation between the intrinsic and extrinsic information [15]. The performance is degraded substantially due to first of these distortions and mildly due to the second. The first type of distortion, which depends on \( E_b/N_0 \), is considered. The compensation co-efficient is calculated. The compensation of \( L_e(u_k) \) is possible with a common scaling factor. Algorithms are modified by multiplying extrinsic information \( L_e(d_k) \) with the chosen scaling factor before it is being fed back to
the input [10]. The scaling factor must be chosen in such a way that it gives substantial improvement in the reliability of output from the decoder and decreases the number of iterations involved in attaining the Shannon’s capacity limit of error performance [11]. MMAP and MSOVA algorithms [9], [16] are achieved by fixing an arbitrary value for inner decoder ($S_1$) and an optimized value for the outer decoder ($S_2$). For Enhanced MAP (EMAP) and Enhanced SOVA (ESOVA), both $S_1$ and $S_2$ are optimized. Scaling factor $S_2$ depends on $E_b/N_0$ to give low BER and better performance than modified decoding algorithms. The proposed Turbo decoder with optimized scaling factors is shown in Fig.3. The algorithms are enhanced by multiplying the extrinsic information $(\hat{d}_k)$ with the optimized scaling factors $S_1$ and $S_2$ before it is being fed back to the input and decoder 2 respectively, and are given by

$$z_k = \left[ L_{e2}(\hat{d}_k) \right] \times S_1$$  \hspace{1cm} (14)

$$L_1(\hat{d}_k) = \left[ \frac{2}{\sigma^2} x_k + L_{e1}(\hat{d}_k) \right] \times S_2$$  \hspace{1cm} (15)

![Fig.3 Turbo Decoder with Optimized Scaling Factors](image)

### 5 Simulation Results and Discussion

Transmission of 1140 frames with a frame length of 2048 bits and random interleaver [17] is taken to show the effect of the scaling factors on the performance of TC and to analyze the outcome of various constraint lengths and code rates. The simulation parameters for finding the optimized scaling factors are,

- **Channel**: AWGN
- **Modulation**: Quadrature Phase shift Keying (QPSK)
- **Component encoder**: Recursive convolution codes (RSC)

Table 1 shows the optimized scaling factor ($S_2$) giving the least BER, $E_b/N_0$ and the corresponding BER for EMAP and ESOVA algorithms. It is found that for $E_b/N_0$ greater than 1.0dB the optimized scaling factor for enhanced Log MAP algorithm is constant and is found to be 0.85. For ESOVA, $S_2$ is found to vary with $E_b/N_0$ and is adaptive with respect to $E_b/N_0$. The BER for EMAP algorithm at 3dB is $1.6211 \times 10^{-46}$. Similarly at 3dB, the BER for enhanced SOVA is comparatively reduced and its value is $9.8144 \times 10^{-47}$. Thus ESOVA gives reduced BER and better performance than EMAP algorithm.

### Table 1

<table>
<thead>
<tr>
<th>Eb/No (dB)</th>
<th>EMAP: Optimized Scaling Factor ($S_2$)</th>
<th>BER</th>
<th>ESOVA: Optimized Scaling Factor ($S_2$)</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.89</td>
<td>1.0800x10^4</td>
<td>0.71</td>
<td>1.2528x10^4</td>
</tr>
<tr>
<td>0.5</td>
<td>0.89</td>
<td>5.9358x10^2</td>
<td>0.71</td>
<td>7.5956x10^2</td>
</tr>
<tr>
<td>1</td>
<td>0.88</td>
<td>7.4698x10^3</td>
<td>0.71</td>
<td>2.0261x10^2</td>
</tr>
<tr>
<td>1.5</td>
<td>0.85</td>
<td>7.5571x10^3</td>
<td>0.99</td>
<td>9.0685x10^4</td>
</tr>
<tr>
<td>2</td>
<td>0.85</td>
<td>5.8878x10^4</td>
<td>0.85</td>
<td>3.2388x10^3</td>
</tr>
<tr>
<td>2.5</td>
<td>0.85</td>
<td>1.9629x10^4</td>
<td>0.92</td>
<td>3.9258x10^4</td>
</tr>
<tr>
<td>3</td>
<td>0.85</td>
<td>1.9629x10^3</td>
<td>0.86</td>
<td>9.8144x10^3</td>
</tr>
<tr>
<td>3.5</td>
<td>0.85</td>
<td>1.6211x10^4</td>
<td>0.99</td>
<td>9.8144x10^3</td>
</tr>
<tr>
<td>4</td>
<td>0.85</td>
<td>1.6211x10^4</td>
<td>0.95</td>
<td>9.8144x10^3</td>
</tr>
</tbody>
</table>
Fig. 4 BER plot of various Scaling Factors and $E_b/N_0$ with code generator (7,5), punctured for Log MAP algorithm.

Fig. 5 BER plot of various Scaling Factors and $E_b/N_0$ with code generator (7,5), punctured for SOVA algorithm.
Fig. 6 shows the performance of Enhanced Log MAP algorithm with the scaling factors $S_1=0.9$ and $S_2=0.85$ is giving better results than the Modified Log MAP algorithm with scaling factors $S_1=0.9$ and $S_2=0.755$. The MMAP and EMAP algorithms are also compared with the standard algorithm without any scaling factor, at $E_b/N_0$ of 2.5dB. This graph gives evidence on the improved performance of EMAP algorithm in terms of BER. It is noted that for iteration 4, the BER of MMAP [16] and EMAP algorithms are $0.5 \times 10^{-45}$ and $1 \times 10^{-46}$ respectively.

It is also observed from Fig. 6 that the performance remains constant from iteration 4. It is revealed for Log MAP algorithm, the efficient BER has been achieved by 4 iterations. Thus in the proposed EMAP algorithm, complexity has been reduced by 50% compared to Log MAP algorithm and the BER has been reduced by the order of $10^{-1}$ compared to MMAP algorithm. The main design criterion for any decoding algorithm is to reduce the BER and complexity, which is achieved by the proposed EMAP algorithm.

Fig. 7 shows the performance of Enhanced SOVA algorithm with the scaling factors $S_1=0.56$ and $S_2=0.92$ is giving better results comparing with the Modified SOVA [10] and SOVA algorithm, at $E_b/N_0$ of 2.5dB. At the end of 8th iteration in the decoding part the difference in BER for MSOVA and ESOVA is about $0.8 \times 10^{-1}$. It is noted that as the iteration increases, the performance of ESOVA improves. Since SOVA is less complex compared to Log MAP, the number iterations can be increased to 6 without degradation in performance.

Table 2 gives the summary of the number of iterations required, BER and the percentage of reduction in complexity for each decoding algorithms. Compared to Log MAP algorithm, the complexity of MMAP and EMAP algorithms are reduced. EMAP also shows BER improvement than MMAP but with the similar complexity. Though the complexity reduction of ESOVA is higher than MSOVA, the BER of ESOVA is greatly reduced.

<table>
<thead>
<tr>
<th>Decoding Algorithms</th>
<th>Iteration from which BER is constant</th>
<th>Complexity reduced in %</th>
<th>Corresponding BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log MAP</td>
<td>7</td>
<td>12.5</td>
<td>$9.0897 \times 10^{-5}$</td>
</tr>
<tr>
<td>MMAP</td>
<td>4</td>
<td>50</td>
<td>$1.7777 \times 10^{-5}$</td>
</tr>
<tr>
<td>EMAP</td>
<td>4</td>
<td>50</td>
<td>$1.9629 \times 10^{-6}$</td>
</tr>
<tr>
<td>SOVA</td>
<td>8</td>
<td>0</td>
<td>$1.2759 \times 10^{-5}$</td>
</tr>
<tr>
<td>MSOVA</td>
<td>5</td>
<td>37.5</td>
<td>$1.0796 \times 10^{-5}$</td>
</tr>
<tr>
<td>ESOVA</td>
<td>6</td>
<td>25</td>
<td>$5.8887 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

The summary of scaling factors for various decoding algorithms is shown in Table 3.
Table 3
Scaling Factors for Various Decoding Algorithms

<table>
<thead>
<tr>
<th>Decoding Algorithms</th>
<th>Scaling factor 1 (S₁)</th>
<th>Scaling factor 2 (S₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMAP</td>
<td>0.9 *</td>
<td>0.755</td>
</tr>
<tr>
<td>EMAP</td>
<td>0.9 *</td>
<td>0.85 *</td>
</tr>
<tr>
<td>MSOVA</td>
<td>0.56 *</td>
<td>0.98</td>
</tr>
<tr>
<td>ESOVA</td>
<td>0.56 *</td>
<td>Adaptive *</td>
</tr>
</tbody>
</table>

* - Optimized Scaling Factors

At $E_b/N_0$ of 1.5dB and above EMAP algorithm is better. But for lower $E_b/N_0$ values (<1.5dB), MMAP algorithm is better. So the proposed algorithm yields the lowest BER.

Fig.9 gives the performance of SOVA, MSOVA and ESOVA in AWGN channel. It is found that MSOVA fails to improve in AWGN channel, whereas ESOVA does. At $E_b/N_0$ of 1.5dB and above ESOVA algorithm is better. But for lower $E_b/N_0$ values (<1.5dB), SOVA algorithm is better. At $E_b/N_0$ of 3.5dB, BER of ESOVA is $1.7 \times 10^{-7}$ which shows two fold improvement in performance compared to SOVA and MSOVA.

Similar analysis is done for the Rayleigh Fading channel and is shown in Fig.10 and Fig.11. The performance of EMAP algorithm in fading channel is almost identical to that in AWGN channel for $E_b/N_0$ greater than 2.5dB, which validates the robustness of the EMAP algorithm and is shown in Fig.10.
The performance of the proposed ESOVA algorithm in fading channel is shown in Fig.11. It is almost identical to the performance of AWGN channel for all values of $E_b/N_0$, which validates the robustness of the ESOVA algorithm. On scaling the extrinsic information with optimized scaling factors $S_1$ and $S_2$, the SOVA algorithm is optimized. Thus it is observed that no further enhancement to the algorithm is required. So in both the channel conditions, the proposed Turbo decoding algorithms gave improved performance.

The performance of enhanced decoding algorithms are analyzed considering three code generators $(7,5)$, $(15,13)$ and $(31,17)$ with constraint length $k=3$, 4 and 5 respectively and two code rates.

Similarly $k=5$ code gives further improvement of 0.5dB at BER of $9\times10^{-7}$ than $k=4$ code and is shown in Fig.12. But Rayleigh fading channel shows no performance improvement on increasing the constraint length as shown in Fig.13.

The effect of increasing the constraint length of the component codes used in ESOVA is shown in Fig.14 and Fig.15. For the constraint length four Turbo code we used the optimum minimum free distance generator polynomials [8] for the component codes 15 and 13. The resulting turbo code gives an improvement of about 0.5 dB at a BER of $10^{-7}$ over the curve. For the constraint length five Turbo code we used the generator polynomials 31 and 17, which were the polynomials used by Berrou et al. [1] in the original paper on TC.

It can be seen from Fig.14 and Fig.15 that increasing the constraint length of the turbo code does improve its performance, with the $k=4$ code performing about 0.5 dB better than the $k=3$ code at a BER of $10^{-6}$, and the $k=5$ code giving a further improvement of about 0.2 dB.
However, these improvements are provided at the cost of approximately doubling or quadrupling the decoding complexity. So constraint length 4 Turbo code is considered as the suitable choice giving reduced BER at comparatively reduced complexity. It is also found from Fig.14 and Fig.15 that the proposed ESOVA algorithm in Rayleigh fading channel performs equally well as that of AWGN channel for various constraint lengths.

Fig.16 Code rate and channel comparison for EMAP with k=4

The Fig.16 and Fig.17 shows the effect of code rates 1/2(punctured) and 1/3(unpunctured) on the performance of EMAP and ESOVA algorithm, in AWGN and Rayleigh fading channels for constraint length four TC.

The performance of AWGN channel for EMAP algorithm is superior to the fading channel as shown in Fig.16. Rate 1/3 Turbo code performs about 0.3dB better than rate 1/2 Turbo code at a BER of $8 \times 10^{-6}$ over the curve, for both the channel conditions. This is due to the increased redundancy of code rate 1/3 which gives improved reliability and hence reduces the BER.

Fig.17 Code rate and channel comparison for ESOVA with k=4

Fig.17 shows the code rate and channel comparison for ESOVA with constraint length four TC. Rate 1/3 TC shows improvement of about 0.4dB than rate 1/2 at a BER of $9 \times 10^{-6}$ over the curve. It is also found that the performance of ESOVA algorithm is almost similar in both AWGN and Rayleigh fading channels.

So, on using the ESOVA algorithm optimized performance for fading channel is achieved. The proposed ESOVA is highly robust for practical channel conditions giving lowest possible BER identical to that of the theoretical AWGN channel.

6 Conclusion

Thus optimizing both the scaling factors in Log MAP and SOVA algorithm lead to the improvement in performance of the decoding algorithms in AWGN and Rayleigh fading channels. On increasing the constraint length, the performance of proposed Turbo code improved for EMAP algorithm in AWGN channel. But Rayleigh fading channel shows no performance improvement. It is found that for ESOVA algorithm, increasing the constraint length of the TC does improve its performance, with the k=4 code performing better than the k=3 code at a BER of $10^{-6}$, and the k=5 code giving further improvement in performance, in both AWGN and fading channels. Rate 1/3 TC outperforms rate 1/2 TC for both EMAP and ESOVA algorithms giving improved reliability and reduced BER.

References:


Authors Biography:

**Dr. T. Gnanasekaran**, is graduated from University of Madras in 1989, completed his post graduate and PhD from Anna University Chennai, India. He is presently with Bannari Amman Institute of Technology, Sathyamangalam, TamilNadu, India, as Associate Professor. He has published 12 national and international journals. His areas of interests are Error Control Coding, Advanced Digital Communication, WiMAX and LTE.

**V. Aarthi** completed her under graduation from Anna University Chennai and post graduation from Anna University Coimbatore, India. She is currently with Velalar College of Engineering and Technology, Erode as Assistant Professor. She has presented three papers in national level conferences and three international conferences. She has also published two international journals.