Implementation and Performance Evaluation of Turbo Decoder

Kandula Saitharun¹, Sambashivudu Korutla², K.V.S Ramakrishna Reddy³

¹,²Assistant Professor in the Department of ECE, University College of Engineering and Technology for women, Warangal, Telangana, India  
³ Department of ECE, University College of Engineering and Technology for women, Warangal, Telangana, India  
¹kandulasaitharun@gmail.com, ²sambashivudu@gmail.com, ³ramakrishnareddy.kvs@gmail.com

Abstract: Turbo codes propose best performance when evaluate to convolutional codes at modest Bit Error Rate (BER) and huge block length. This is workout on the theory of efficient iterative deciphering technique. The Turbo codes were established to reduce the difficulty of iterative turbo decoding though the recital of the method is approach to Shannon capacity. Repeat Accumulate, Irregular Repeat Accumulate & Accumulate Repeat Accumulate codes are few of the recently discovered turbo_like code. These Turbo codes are utilized in various significant wireless protocols from deep space communications to cellular communications. The goal of this manuscript is to study and analyze the two prime candidates for interpreting turbo codes, i.e. SOVA and Log_MAP decoding algorithms. The simulations are executed using MATLAB software. Bit Error Rate (BER) performance comparison of both SOVA and Log_MAP turbo decoding algorithms are prepared. While doing this manuscript, research was taken on how different system elements such as frame length, number of iterations and code rates influence the performance of turbo codes by Log_MAP and SOVA decoding algorithms.

Keywords- Decoding Algorithms, turbo-codes, SOVA, Log-MAP, Max-Log-MAP.

I. INTRODUCTION

Turbo codes consists of two Recursive Systematic Convolutional (RSC) codes separated by an interleave that are coupled in parallel termed parallel concatenation. The major difference between both the convolutional and turbo codes is that convolution codes exhibits higher production for bigger constraint length, the turbo codes has a short constraint length, which preserve as a constant value in most of application [1]. However, it attains a significant coding upgrading at low coding rates. An essential feature for achieving this growth is due to make use of the Soft Input Soft Output (SISO) to generate the soft results using decoder algorithm. The iterative approach of the various turbo decoding algorithms has very large complexity when evaluated to conventional forward error correction decoding methods. The most important of the iterating decoder strategies are Soft Output Viterbi Algorithm (SOVA) & Maximum a Posteriori (MAP) probability algorithm. These techniques require complex operations at the decoder end with several decoding iterative sequence. By minimizing the problem of decoding of turbo codes in real time whereas developing Bit Error Rate (BER) is a considerable design factor. Log_MAP algorithm needs huge number iterations to gain a moderately better BER at small Signal to Noise Ratio. This result in unnecessary time delays and also increases computational complexity. So an additional turbo decoding scheme such as SOVA was recommended. This SOVA depends on conventional Viterbi Algorithm, which generates soft output rather than hard output. It affords finest suitable path arrangement along with consistency rate of each and every one of the received bits, which treated as the soft predictable output. When contrasted to Log_MAP, SOVA decreases the system complexity and also time delay. Conversely the character performance of Log_MAP is quite doing extremely well than SOVA.
The proposed approximation technique to demonstrate that there exists rate-1/2 punctured turbo codes that can get lower error floors than that of their rate-1/5 parent codes. In particular, we explain that a particular puncturing method can be used to decrease the rate of a turbo code from 1/5 to 1/2 and at the same time attain a coding gain at low bit error probabilities, when suboptimal iterative decoding is used.

II. RELATED WORK

Usually, the performance of a digital communication system consisting of a transmitter, a communication channel and a receiver, is calculated by evaluating the probability of a bit error at the output of the receiver, greater than a range of signal-to-noise ratio values at the input of the receiver [2]. The main tools for the performance evaluation of a communication system are either computer simulation or the derivation of analytic expressions for the bit error probability (BER). Computer simulation generates reliable error probability estimates as low as 106 for low signal-to-noise ratios. Performance analysis at lower error probabilities and, subsequently, higher signal-to-noise ratios (SNR), using computer simulation is time-consuming and impractical; hence an analytic approach is the only alternative. When a simple uncoded communication system is considered, such as a two level modulation scheme over a memoryless channel, a hard decision is made upon reception of each individual modulated bit. In those cases, it is relatively uncomplicated to derive a correct expression for the probability of error, provided that the probability distribution of noise on the channel is known. When high order modulation methods are used, where bits are clustered into symbols, or when channel codes are used, where a decoding result can only be made upon reception of a series of bits, statistical development of an exact expression for the probability of error is either too complex or impractical. In such cases, the typical approach is to gain an upper bound on the error probability. A typical approach to upper bound the error probability is by taking the addition of the error probabilities for the entire possible erroneous events. In additive white Gaussian noise channels this upper bound, identified as the union bound, gradually approaches and, at large values of signal-to-noise ratios, ultimately merges with the curvature of the genuine performance of a turbo code [3]. However, improved bounds that expect the performance at low signal-to-noise ratios and on quasi-static fading channels [4] have also been derived.

III. DECODING OF TURBO CODES

The decoding algorithms of turbo code functions on the basis of trellis structure estimating technique. There are two types of estimation techniques, they are:

a. sequence wise estimation technique.

b. symbol wise estimation techniques.

Both viterbi algorithm and SOVA are sequence evaluation algorithms while map algorithm, log map & the Max log map algorithm are symbol estimation algorithms. On an evaluation over these two algorithms gives symbol estimation is hard such as execution takes lots of time than sequence based. All algorithms i.e. SOVA, MAP, log_MAP, MAX-LOG-MAP[5] offers soft decisions. But Viterbi algorithm however generates hard decisions as the output[6]. SOVA is the algorithm that generates soft output.
As shown in figure, two component decoders are linked by Deinterleaver and interleaver pair.

**Maximum Aposterior Probability (MAP)**

Let us consider a block or convolutional encoder described by a trellis, and a sequence $x = x_1; x_2; x_3;\ldots x_N$ of $N$ $n$-bit code words or symbols at its output, where $x_k$ is the symbol created by the encoder at time $k$. The corresponding data or message input bit, $u_k$ can receive on the values -1 or +1 with an a priori probability $P(u_k)$, from which we may describe the so called Log Likelihood Ratio (LLR).

$$L(u_k) = \ln \frac{pr(u_k == +1)}{pr(u_k == -1)}$$

In the MAP algorithm there is calculation of three parameters such as:

1) $\alpha$ forward path metric.
2) $\beta$ backward path metric.
3) $\gamma$ total branch metric.

**Simplified Version of MAP Algorithm**

The MAP algorithm experience from a main disadvantage: it needs to perform various multiplications.

$$\max^*(a, b) = \begin{cases} \max(a, b) + \ln(1 + e^{-|a-b|}) & \text{LOG MAP} \\ \max(a, b) & \text{MAX LOG MAP} \end{cases}$$

The values of the function $\ln(1 + \exp^{-|a-b|})$ are usually stored in a look-up table with just eight values. The element $\ln(C_k)$ in the expression of $\gamma_k(s', s)$ will not be used in the $L(u_k/y)$ computation. The LLR is given by the final expression

$$L(u_k/y) = \max^*_{R_1}[A_{k-1} + \gamma(s', s) + B_k(s)] - \max^*_{R_0}[A_{k-1} + \gamma(s', s) + B_k(s)]$$

The Log_MAP algorithm uses exact formulas so its performance equals that of the BCJR algorithm although it is simpler, which means it is preferred in implementations[7]. Then, the MAX_log_MAP algorithm utilizes approximations; therefore its performance is a little worse.
IV. IMPLEMENTATION DETAILS

4.1 Approximation to the Correction Function:

This section gives a brief review of existing algorithms which approximates the correction function in order to achieve a simple implementation yet improved performance as compared to Max-Log-MAP.

![Figure 4.1: Flow Diagram for Linear MAP](image)

4.1.1 Constant Log-MAP Algorithm

In this algorithm [8] proposed as, the correction function \( f_c(x) \) is approximated with the following rule:

\[
f_c(x) = \begin{cases} 
\frac{3}{8} & -2 < x < 2 \\
0 & \text{Otherwise}
\end{cases}
\]  

---

Eq.4.1

The constant Log-MAP algorithm offers a simple implementation in hardware but with trade off in performance.

4.1.2 Linear Log-MAP Algorithm:

A linear approximation [9] to the correction functions by employing the MacLaurin Series expansion. It is observed that the correction function is effective when \( f_c(x) \) is around zero. Then, the Maclaurin series can be oppressed to estimate the correction function about zero. By ignoring Maclaurins series order two and above the approximation for the correction term is specified as:

\[
f_c(x) = \max(0, \ln 2 - 0.5x) \quad x < 1.5 \quad ---\text{Eq.4.2}
\]

![Figure 4.2: Flow Diagram for Linear MAP](image)
This approximation gives improved performance than the Constant Log_MAP algorithm and needs only a simple linear implementation.

4.1.3 Multistep Log-MAP Algorithm:

A more accurate and elegant solution to approximate the correction function is given in [10]. The approximation to the correction term as suggested is given as:

\[ f_c(x) = \frac{\ln 2}{2[x]} \] \hspace{1cm} \text{Eq.4.3}

Where \([x]\) denotes the largest integer that is smaller or equal to \([x]\). The correction term specified at this point is a more precise yet simple approximation to the correction function. Note that division by 2 can be easily done in digital systems by implementing \([x]\) number of binary shifts. The algorithm uses shift registers accumulate the constant \(\ln(2)\) to perform the division. Though, in order to make possible fast computation, a high speed shift register is required for this algorithm.

4.1.4 Hybrid LOG-MAP Approximation:

The hybrid approximation is proposed [11] as:

\[ f_c(x) = \begin{cases} 
0.6512 - 0.3251 x & x < 1.5 \\
\frac{\ln 2}{2[x]} & \text{otherwise}
\end{cases} \] \hspace{1cm} \text{Eq.4.4}

In the region of \(|x| < 1.5\) the hybrid algorithm employs a linear polynomial fit. The accuracy of the approximation is verified through the goodness of the fit test against the LOG_MAP.

4.2 Proposed Algorithms for Jacobian Function

The following algorithms are proposed for the improvement of the performance of MAP algorithms either in the decoding state or equalization state

4.2.1 Improved Linear Log-Map Approximation

In the novel development, approximation has the advantage over the linear and constant Log map in terms of accuracy as well as hardware simplicity. The proposed algorithm is expressed as follows:

\[ f_c(x) = \begin{cases} 
\max(0, \ln 2 - 0.5x) & x < 1.5 \\
\text{LUT}(\text{pre-defined values}) & \text{Otherwise}
\end{cases} \] \hspace{1cm} \text{Eq.4.5}

The best in the improved linear Map is its doesn't add neutral values to the max value, it adds some weights to the max values which in rather improves the performance of the decoding algorithm.

4.2.2 Improved Step Log-Map Approximation

Inspired by observing the curve of the exact correction term, propose new approximation correction terms. The step algorithm for the correction function is just shifting the \(\ln 2\) value, the improved step is proposed as follows:

\[ f_c(x) = \frac{\ln 2}{2[x]} \] \hspace{1cm} \text{Eq.4.6}
Where \([x]\) is the biggest integer that is lesser or equal to \(x\). These two correction terms are more accurate than Eq 4.1 to Eq 4.4. Moreover they are competitive when considering the implementation complexity. Note that dividing by 2 is a much uncomplicated job for digital circuits. With Eq 4.6, the correction term can be obtained by shift the register storing the constant log 2. The required shift time is finding out by \(x\) which is in fact characterized by binary integers in the fixed point digital circuits.

V. SIMULATION RESULTS

The results show here for the various code rate and interleaver. As the depth of the interleaver increases then the BER performances of that system is also increases and as the code rate to 1/5 to 1/2 the performance degradation is happened. There must be a tradeoff in choosing the interleaver length and code rate, as the code rate of 1/5 is taken then the Data rate of the system comes down as the interleaver depth increases then the delay in system is increases.

The figures show the performance of the Turbo decoder for different proposed algorithms and the table shows the complexity of the above mentioned algorithms \(N\) is the number of states of the encoder, and the above table is for decoding a single information bit.

![Figure 5.1: BER Performance of Improved Linear approximation](image1)

![Figure 5.2: BER Performance of Improved Step approximation](image2)
This thesis focuses on the implementation of Turbo decoder. For that, we first reviewed the currently using methods for the performance improvement in the turbo decoder through MAP decoding algorithm. From the existing methods low complexity, low latency Step algorithm reduces the power consumption of the decoder.

The accurate Log-MAP algorithm involves computationally intensive operations in order to achieve the ideal performance. Avoiding the correction function, the Log-MAP algorithm decreases to the simple Max-Log-MAP algorithm. However, this rough approximation provides a capacity loss, and therefore the correction term will have to be integrated or approximated which provides us the suboptimal Log-MAP algorithm. The novel improved linear and step map algorithm is a suboptimal Log-MAP solution that achieves nearly identical performance to the Log-MAP algorithm. The improved linear and step approximation offers a simple implementation on hardware involving shift registers, multiplications, comparators, and addition operations. In addition, we also show that the improved algorithm outperforms existing Log-MAP based algorithm.

REFERENCES

[11] Li Li Lim and David Wee Gin Lim, 'Hybrid log-map algorithm for turbo decoding over AWGN channel," ICWMC The sevent international conference on wireless and mobile communications,2011.

Author’s Profile

Kandula Saitharun is working as an Assistant Professor in the Department of ECE, University College of Engineering and Technology for women, Warangal, Telangana, India. Received Master of Technology from NIT Warangal in 2015. His area of interest in Signal Processing and Communication.

Sambashivudu Korutla is working as an Assistant Professor in the Department of ECE, University College of Engineering and Technology for women, Warangal, Telangana, India. Received Master of Technology from Ganapathy Engineering College, Warangal in 2013. His area of interest in Signal Processing and Communication, VLSI Design.

K.V.S Ramakrishna Reddy is working in the Department of ECE, University College of Engineering and Technology for women, Warangal, Telangana, India. Received B.Sc(Electronics) from Vaagdevi Degree & PG College, Warangal. His area of interest in Communication.