Development of turbo decoder for wireless sensor network and its implementation on FPGA

1Supriya P, 2Smt.Bhagya.R

1M.Tech Student: Digital Communication Engineering
2 Asst. Professor; Department of Telecommunication Engineering
R.V. College of Engineering Bangalore, INDIA
Email: supriya.puttaraj@gmail.com

Abstract—Turbo codes have recently been considered for energy-constrained wireless communication applications, since they facilitate a low transmission energy consumption. This paper proposes the performance of Turbo Codes using DPSK modulation technique over the AWGN (Additive White Gaussian Noise) and Rayleigh channel. The performance of turbo codes consisting of a encoder, a communication channel and a decoder, is measured by evaluating the probability of a bit error at the output of the decoder, over a range of signal-to-noise ratio values at the input of the decoder. In order to reduce the overall energy consumption LUT-Log-BCJR decoding algorithm is used. The BER performance can be improved as iterations proceed. At lower BER the high SNR can be achieved in AWGN channel when compared to Rayleigh channel. As the number of iterations increases the SNR can be improved the LUT-Log-BCJR architecture is decomposed into its most fundamental Add Compare Select (ACS) operations and performed using a novel low-complexity ACS unit. The turbo decoder is developed and simulated using modelsim

Keywords – LUT-Log-BCJR, Turbo codes, ACS(Add compare select), WSN.

I. INTRODUCTION

Turbo codes were first introduced by Claude Berrou., A Glaviex and P. Thitimajshima in the year 1993 showing its introduction results in a high gain and low complex coding with power saving feature as important aspect at higher values of Eb/Nb.

In 1949 Claude Shannon showed that if the communication channel is subjected to noise or interference then the channel is able to transmit information with some errors. These errors are generally occurred due to introduction of noise with the transmitted signal provided the capacity of channel is not exceed and whose performance is in terms of bit error rate (BER) are closed to Shannon’s limit Turbo codes consist of concatenation of two convolution codes. Turbo codes give better performance at low SNRs (signal to noise ratio). Interestingly, the name Turbo was given to this codes because of the cyclic feedback mechanism (as in Turbo machines) to the decoders in an iterative manner. The turbo encoder transmits the encoded bits which form inputs to the turbo decoder. The turbo decoder decodes the information iteratively. Turbo codes can be concatenated in series, parallel or in a hybrid manner. Concatenated codes can be classified as parallel concatenated convolution codes (PCCC) or serial concatenated convolution codes (SCCC).

The probability of error in any signalling scheme is the function of signal to noise ratio. The use of PCCC reduces this probability of error, as in PCCC both the encoders operate on the same information bits. In SCC, one encoder encodes the output of another encoder. The hybrid concatenation scheme consists of the combination of both parallel and serial concatenated convolution codes. The turbo decoder has two decoders that performs iterative decoding.

Turbo codes have been used also for its high coding gain. Wireless sensor networks are considered to be energy efficient because they rely on batteries that are light and inexpensive and can operate on extended periods of time. The WSN’s energy consumption is mainly dominated by the transmission energy. Turbo codes have recently found application in these scenarios, since their high coding gain facilitates reliable communication when using a reduced transmission energy. However the reduced transmission energy is counteracted by the turbo decoder’s energy consumption. For this reason Turbo codes designed for energy efficient WSNs must reduce the overall energy consumption i.e Sensor’s transmission energy and Turbo decoder’s energy consumption. The BCJR algorithm is an algorithm for maximum a posterior decoding of error correcting codes defined on trellises (principally convolutional codes). The algorithm is named after its
inventors: Bahl, Cocke, Jelinek and Raviv. This algorithm is critical to modern iteratively-decoded error-correcting codes including turbo codes and low-density parity- check codes. The BCJR algorithm contains multiplications and divisions in computation which is complex and due to this reason it was not used for almost 30 years. Since the invention of turbo codes BCJR algorithm have been started to be used. Now many variants have been evolved from BCJR algorithm. They are

1. MAX-LOG BCJR ALGORITHM
2. LOG-LUT BCJR ALGORITHM

All the computations when being transferred into the logarithmic domain becomes additions and subtractions which reduce the complexity. The Max-Log-BCJR algorithm appears to lend itself to both high-throughput scenarios, as well as to the above mentioned energy-constrained scenarios. This is because low turbo decoder energy consumption is implied by Max-Log-BCJR algorithm’s low complexity. However, this is achieved at the cost of degrading the coding gain by 0.5 dB compared to the optimal Log-BCJR algorithm. This motivates the employment of the Look-Up-Table-Log-BCJR (LUT-Log-BCJR) algorithm in energy-constrained scenarios, since it approximates the optimal Log-BCJR more closely than the Max-Log-BCJR and therefore does not suffer from the associated coding gain degradation.

II. TURBO ENCODER

A classical Turbo encoder is composed of two recursive systematic convolution (RSC) encoders, as shown in Fig.1. The input information sequence is encoded twice by the two RSC encoders. The first encoder processes the information in its original order, while the second encoder processes the same sequence in a different order obtained by an interleaver. In this scheme the systematic bit sequence is also transmitted to the decoder. As shown in the figure, sequence c and d are the output of each encoder. Sequence a is the systematic bit sequence and b is the interleaved systematic bit sequence.

![Fig.1. Turbo encoder](image)

III. TURBO DECODER

In the decoding process, as shown in Fig.2, two A Posteriori Probability (APP) decoders are used correspondingly for the two convolution encoders in the encoding scheme to get the minimal bit error probability. In the figure, \( \hat{a}, \hat{c}, \hat{d} \) are the soft decisions sequence corresponding to the output sequence a, c and d in Figure obtained by the demodulator. The purpose of an APP decoder is to compute a posteriori probabilities on either the information bits or the encoded symbols. The algorithm was originally invented by Bahl, Cocke, Jelinek and Raviv in 1972, so called BCJR algorithm. The capability of generating soft decisions of it is well suited for iterative decoding schemes. In Figure, the two decoders are working alternatively in an iterative way. To get the correct order of the input sequences, a identical interleaver with the one used in the encoding scheme and a corresponding de-interleaver is used between the decoders. An extra interleaver is used for providing the systematic sequence for both of the decoders.

![Fig.2. Turbo decoder](image)

IV. PROPOSED WORK

A binary turbo encoder is a parallel concatenation of two recursive systematic convolutional encoder of rate 1/2, the source bits are input to the first encoder while the second encoder is fed by the interleaved version of the original data. Block interleaver is used. The output of the turbo encoder consists of the systematic bits of first encoder, parity bits of the first encoder, parity bits of the second encoder. In order to increase the code rate ‘1/2’ to ‘1/3’, puncturing is applied and the resultant codewords are transmitted over the channel as shown in Fig. 3.

A turbo decoder consists of two decoders separated by an interleaver and de-interleaver. The decoding is an iterative process in which the extrinsic information is exchanged between two decoders. Each turbo iteration is divided into two half iterations. During the first half iteration LUT-Log-BCJR decoder 1 is enabled. It receives the soft channel information and the priori information LA1 from the other constituent LUT-Log-.
BCJR decoder through de-interleaving to generate the extrinsic information LE1 at its output. Likewise during the second half iteration LUT-Log-BCJR decoder 2 is enabled and it receives the soft channel information and the priori information LA2 from LUT-Log-BCJR decoder 1 through interleaving to generate extrinsic information LE2 at its output. This iterative process repeats until the decoding has converged or the maximum number of iterations has been reached.

V. PROPOSED LUT-LOG-BCJR ARCHITECTURE

In this section, we propose a novel LUT-Log-BCJR architecture for energy-constrained scenarios, which avoids the wastage of energy that is inherent in the conventional architecture. LUT-Log-BCJR algorithm comprise only additions, subtractions and the max* calculation of (12, Equation-2). While each addition and subtraction constitutes a single ACS operation, each max* calculation can be considered equivalent to four ACS operations. Now the max* operation is implemented in four steps. The MAX-LOG-BCJR algorithm does not use the approximation it simply finds out the maximum of two LLRs, so it only uses one ACS operation after the calculation of alpha or beta state metric. Similarly, fewer ACS operations are required, when employing the Constant-Log-BCJR algorithm. These alternative algorithms reduce the hardware complexity and increase the throughput, therefore reducing the decoder’s energy consumption. However, this is achieved at the cost of requiring higher transmission energy to achieve the same BER performance as the LOG LUT BCJR algorithm.

All the ACS operations are performed in 2’s complement fixed point representation. As shown in the above Fig 4 inputs and outputs are represented in 7-bit fixed point representation.

VII. RESULTS

The performance of the turbo codes has been measured through MATLAB simulation. The performance of a Turbo codes is evaluated using Bit Error Rate (BER) chart. Y axis is the BER of the decoding result after a certain times of iterative decoding and X axis is SNR. The BER v/s SNR plot for turbo codes for different iterations in an Rayleigh channel and AWGN channel is shown and turbo decoder is developed and simulated using modelsim. from the encoder and decoder wave window it is observed that encoder and decoder outputs are same.
VIII. CONCLUSION

BCJR algorithm has been ignored for the past 30 years due to its high complexity in computation. Log BCJR algorithm due to its reduced complexity have found interest in turbo codes. The proposed architecture employs fewer magnitude ACS units which implements the entire LOG-BCJR algorithm in parallel. BCJR algorithm is simulated using Modelsim. the BER performance is simulated using Matlab. From the simulated results it is observed that at lower BER the high SNR can be achieved in AWGN channel when compared to Rayleigh channel. As the number of iterations increases the SNR can be improved.

REFERENCES


