AN OVERVIEW OF TURBOCODES

Jeena Thomas[1], Ranjitha C R[1], Indrasena N V[2]
[1]PG Scholar, [2]Assistant professor,
Dept.of Electronics & Communication, Thejus College of Engineering
[1]Email: ttjeena@gmail.com

Abstract— This paper focus on the various encoding and the decoding schemes of turbo codes and representing various design parameters of the turbo coding scheme. Turbo code is one of the high performance forward error correction codes used in communication systems, it offers the performance nearer to the Shannon limit. The commonly using Turbo encoder is parallel concatenated recursive systematic convolutional encoders separated by an input output mapping device is known as an interleaver. Similarly the Turbo decoding can be done using different algorithms. BCJR algorithm is the commonly used decoding algorithm for Turbo codes. This paper proposes a brief study about the different decoding algorithms such as Viterbi algorithm, BCJR algorithm, SOVA algorithm, Log-map decoding algorithm.

Keywords— Turbo encoder, Turbo decoder, Interleaver, RSC code, BCJR algorithm, Viterbi algorithm, SOVA algorithm.

INTRODUCTION

Error-control codes, also called error-correcting codes or channel codes, are the key component of digital transmission system. Channel coding is obtained by providing controlled redundancy into the transmitted digital sequence. Turbo codes is the new class of high-performance forward error correction (FEC) codes, and which is the first practical codes closely approach to the channel capacity. The Turbo code has the capacity nearer to the Shannon limit. Shannon limit is the theoretical maximum information transfer rate of the channel. A basic 1/3 rate turbo code is obtained by the parallel concatenation of two ½ rate recursive systematic convolutional encoder separated by an interleaver. There are two kinds of convolutional codes ; non-systematic convolutional (NSC) and recursive systematic convolutional (RSC) codes. By combining the turbo code with a multi-level modulation the spectral efficiency of turbo coded systems can be increased [12]. The turbo coded systems, which are spectral efficient can be classified into two. They are the non-binary turbo code combined with a multi-level modulation and the other one is the coded modulation (CM) with the binary turbo code. The recursive systematic convolutional codes are the main component of Turbo Codes. Those are based on Linear Feedback Shift-Registers (LFRS) and act as pseudorandom scramblers. There are several parameters affect the performance of turbo codes such as component decoding algorithms, number of decoding iterations, generator polynomials, constraint lengths of the component encoders and interleaver type. For a concatenated scheme, the Turbo decoding algorithm should not limit itself to passing hard decisions among the decoders then the turbo code work properly. The initially proposed turbo codes were parallel concatenated convolutional codes (PCCC).Then, the serial concatenated convolutional codes (SCCC) and the hybrid concatenated convolutional codes (HCCC) were proposed.

TURBO CODE ENCODER

Turbo code is one of the high performance forward error correction codes used in communication systems, it offers the performance nearer to the Shannon limit. It is the first practical codes, which closely approach the channel capacity. The general form of Turbo encoder involves two recursive systematic convolutional (RSC) encoders separated by an interleaver. These two encoders used are normally identical and it is said to be symmetrical. If the modulo sum of two valid code words resulting a valid codeword which is known as linear codes. The Turbo code is a linear code. A linear code is said to be 'good' if that has high-weight codewords.

A. Parallel Concatenated ConvolutionalCodes (PCCC)

Turbo-codes is also known as parallel-concatenated recursive systematic convolutional code [1]. The Turbo code has the parallel structure, in which recursive systematic convolutional (RSC) codes working in parallel is used to create the "random" versions of the message.

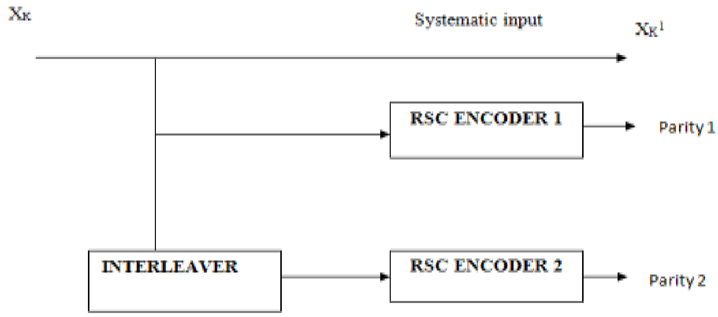

Fig 1: Turbo encoder [1]

Usually two parameters are used to describe convolutional codes that are code rate 'r' and constraint length 'k'. So that code rate can be expressed as 'k/n'. The state information of the convolutional encoder is stored by using the shift registers. In order to avoid the excessive decoding complexity RSC encoder with short constraint length is considered. The RSC encoder can be created from the conventional non-recursive non-systematic convolutional encoder by feeding back one of its encoded output to its input.

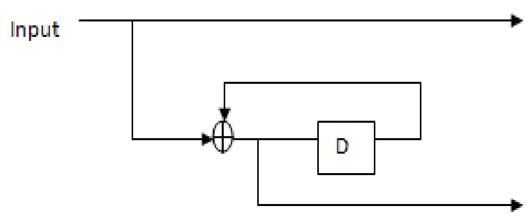

Fig 2: Recursive Systematic Convolutional Encoder[2]

The recursive systematic convolutional encoder tends to produce higher weight code words. So that it is well suitable for turbo codes as compared to non-recursive systematic convolutional encoders.

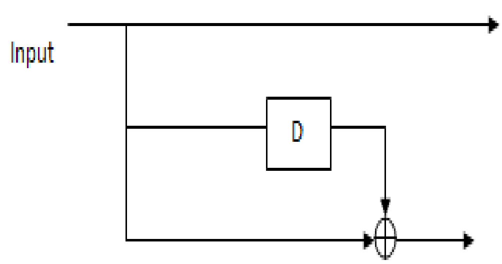

Fig 3: Non-recursive Systematic Convolutional Encoder[2]

Two or more RSC codes, each with a different interleaver is involved in the parallel structure. Interleaver is a device, which permutes the data sequence in some predetermined order. The codeword is formed by considering only one of the systematic outputs from the two component encoders. The systematic output from the other component encoder is the permuted version of the systematic output, which is already selected. A bit-error probability of 10^{-5} is obtained using a rate 1/2 code over an additive white Gaussian noise (AWGN) channel. Recursive systematic convolutional code [2], which has a feedback path that adds the content of the shift register to the input bit. At low signal to noise ratio it offers better performance. Interleaver is an input output mapping device, which change the positions of the bits in each block of data before it enters the second encoder. So the encoder input is not correlated. The interesting property of the RSC code is only a small fraction of finite weight information sequence gets low redundancy coded sequences at the encoder's output. [6].

For high data rates longer interleavers are used. Long interleavers [2] introduce long delays for lower data rates. The BER performance improved with the interleaver size increases is known as interleaver gain. In order to maximise interleaver gain parallel concatenated convolutional codes are use recursive convolutional encoders. For every input information bit, the PCCC outputs a three-bit code word that consists of the systematic bit, and the parity bits, which are generated using the two recursive systematic convolutional encoders. The selection of choice of component codes and interleaver type are the key parameters considering in the performance of a turbocode. RSC code has an infinite impulse response but in the NRC code the impulse response is finite. Because of this RSC code and NRC code has different minimum weight input. The main factor in the designing of convolutional codes is the constraint length. The constraint length is measured as the number of memory elements plus one.

B. Serial Concatenated Convolutional Codes (SCCC)

In SCCC have the outer code and the inner code which is separated by using the interleaver. As a result decoupling takes place at the output of encoder from the input of the inner encoder [2]. A rate 1/3 SCCC that is formed by the rate 1/2 outer code, which is the non-recursive convolutional (NRC) code, and the rate 2/3 inner code, which is recursive convolutional (RC) code.

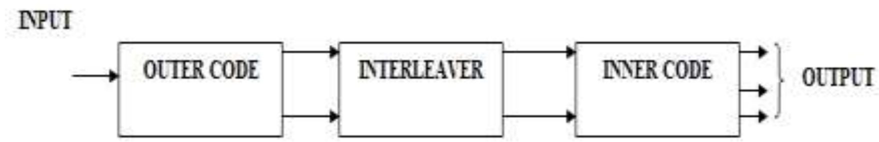

Fig 4: Serial Concatenated Convolutional Codes (sccc) [2]

Parallel concatenated convolutional codes perform better for high BER values. While or low BER values SCCCs perform very well. This performance varies depending on the interleaver size. The performance of an SCCC improves with increasing the interleaver size. For an SCCC to attaining an interleaver gain, a recursive inner encoder must be used. To maximize this interleaver gain the NRC outer code is necessary.

C. Hybrid Concatenated Convolutional Codes (HCCC)

It is the combination of PCCCS and SCCCS. HCCCS become SCCCS without upper branch [3] and it becomes PCCCS without outer code. The disadvantage of HCCC is that they introduce significant amount of delay. For low coding rates HCCCs are known for better performance. Because of using two encoders and one interleaver, the decoding delay is significant. So that HCCCs provide good performance for extremely high data rates, in which the resulting delay is tolerable. The Turbo code depends on several factors such as memory size, generating polynomials, number of decoding iterations, interleaver type and interleaver size [13].

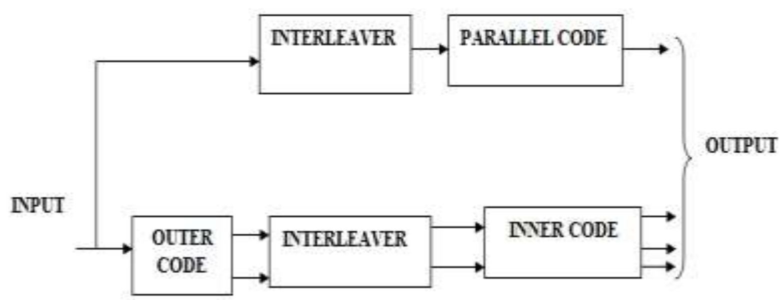

Fig 5: Hybrid Concatenated Convolutional Codes (hccc) [2]

TURBO DECODERS

Decoding of convolutional code in the turbo code can be done by passing soft information from one decoder to the next. The Turbo decoder [7] comprises of two serially interconnected soft-in soft-out (SISO) decoders. The decoding takes place on the noisy versions of systematic bits and two sets of parity bits to produce an estimate of original message bits. There are two kinds of decoding algorithms, soft output viterbi algorithm (SOVA), which is proposed by Hagenauer and Hoher based on the Viterbi algorithm and the BCJR algorithm proposed by Bahl, Cocke, Jelinek and Raviv.

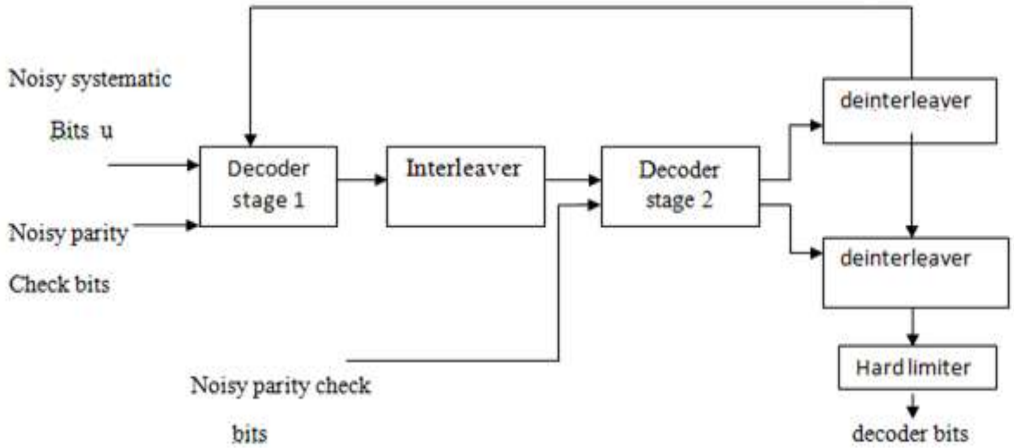

Fig 6: Turbo decoder [7]

For decoding of turbo codes different types of algorithms are available [3]. The trellis-based estimation is the base of each of the algorithm and is classified into two types. They are sequence estimation algorithms and symbol-by-symbol estimation algorithms. Sequence estimation algorithms can be classified as Viterbi algorithm, SOVA (soft output Viterbi algorithm) and improved SOVA.

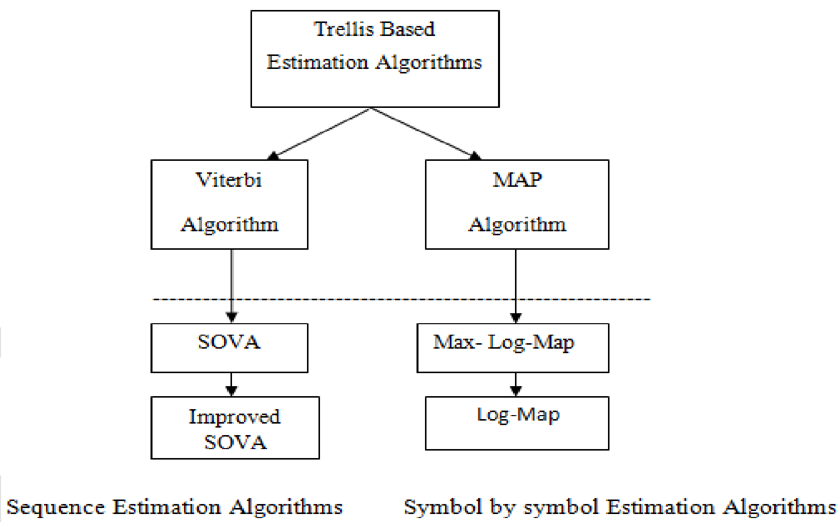

Fig 7: Decoding algorithms for Turbo codes [4]

Symbol-by-symbol estimation algorithms are classified as the MAP algorithm, Max-Log-Map and the Log-Map algorithm. Sequence estimation algorithms are less complex than symbol-by-symbol estimation algorithms. The BER performance of the symbol by symbol algorithm is much better than the sequence estimation algorithms. The algorithms namely The MAP, SOVA, LOG-MAP, MAX-LOG-MAP, improved SOVA produces soft outputs [4]. Viterbi algorithm is a hard-decision output decoding algorithm and the SOVA is soft-output producing Viterbi algorithm. Maximum a-posteriori algorithm, is named as BCJR algorithm, which is an optimal decoding technique for linear codes that minimizes the probability of symbol error. This is in contrast to the commonly used Viterbi algorithm. The maximum length sequence estimation (MLSE) is the principle of the viterbi algorithm. The Viterbi algorithm reduces the sequence (or word) error probability.

A. BCJR ALGORITHM

The BCJR algorithm is implemented to solve the maximum a posteriori probability detection problem, which is a soft input soft output decoding algorithm with two recursions that is forward and backward both involve soft decisions invented by Bahl, Cocke, Jelnek and Raviv. The viterbi algorithm is an algorithm, which operates on the principle of the maximum likelihood decoding. The maximum likelihood decoder, which examine received sequence and detect a valid path which has the smallest hamming distance from the received sequence. The viterbi algorithm is a soft input hard output algorithm, in which only the forward recursion involving soft decisions is possible. The BCJR algorithm is more complex than the viterbi algorithm because of backward recursions.

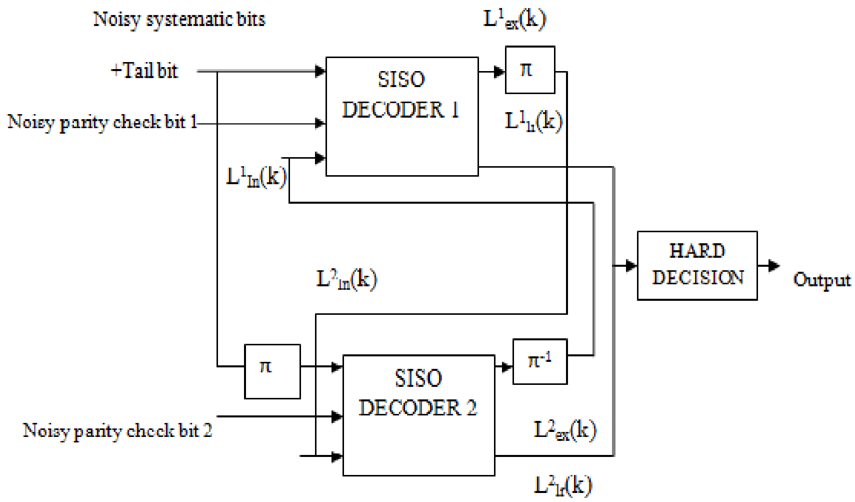

Fig 8: Structure of a Turbo Decoder based on either SOVA or the BCJR Algorithm [8]

In decoding section the received sequence is partitioned in to three, that are systematic bits, and parity check bits 1 and 2 [8]. Here the systematic bits, parity check bits1 and a priori information, which is taken from SISO Decoder 2 is taken as the input to SISO Decoder 1 and the decoder 1 outputs extrinsic information and the log likelihood ratio as a result of estimation of a bit sequence by use of SOVA. SISO Decoder, which produces a-posteriori information by decoding a-priori information. Systematic information, parity information and a priori information are the inputs to the SISO Decoder.

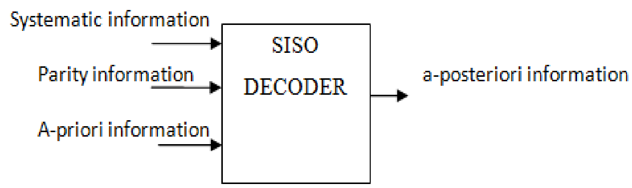

Fig 9: Structure of the SISO Decoder[8]

Consider u = (u_1, u_2 … u_N) be the information bits represented by the binary random variables. In the case of systematic encoders, one of the outputs xs = (xs_1, xs_2… xs_N) is similar to the information sequence u and the next is the parity information sequence output xp = (xp_1, xp_2… xp_N). In the MAP decoding scheme[4], the decoder decides whether uk =+1 or uk = -1, which depends on the sign of the log-likelihood ratio (LLR). In the case of radix-2 trellises the log domain computations of the BCJR algorithm can be separated in to three main categories that are branch metric computation, forward / backward metric computation and combination of forward and backward state metrics.

The interleaved version of the extrinsic information is provided as an input to decoder 2, where it is used as a priori information and the decoding is performed together with an interleaved version of the systematic bits and the parity check bits. SISO decoder 2 – also based on SOVA like SISO decoder 1,which outputs extrinsic information and a log likelihood ratio. For a second iteration the SISO decoder takes the deinterleaved version of extrinsic information and the log likelihood ratio and is used as a-priori information in SISO decoder 1. Two LLR outputs after the number of iterations are used to make a hard decision. In the case of BCJR decoding of a convolutional turbo encoder 8 to 10 iterations are conducted.

B. VITERBI DECODING ALGORITHM

The viterbi algorithm was introduced in 1967.The maximum likelihood decoding of convolutional codes can be executed by using this algorithm [8]. This algorithm works by rejecting the less likely paths and keeping the most likely path through the trellis in each node. A hard decision on the transmitted sequence means that the path selection leaves with a single path in the Trellis. By using this algorithm the maximum likelihood sequence can be found. At the early point of the decoding process, loss of valuable information takes place due to the hard decision making. a- priori information the viterbi algorithm accepts the soft-inputs in the form of but it does not produce soft-outputs. By using the encoders Trellis diagram viterbi algorithm works as maximum likelihood sequence estimator. So that it selects a path with the highest likelihood by looking all possible sequences Trellis diagram.

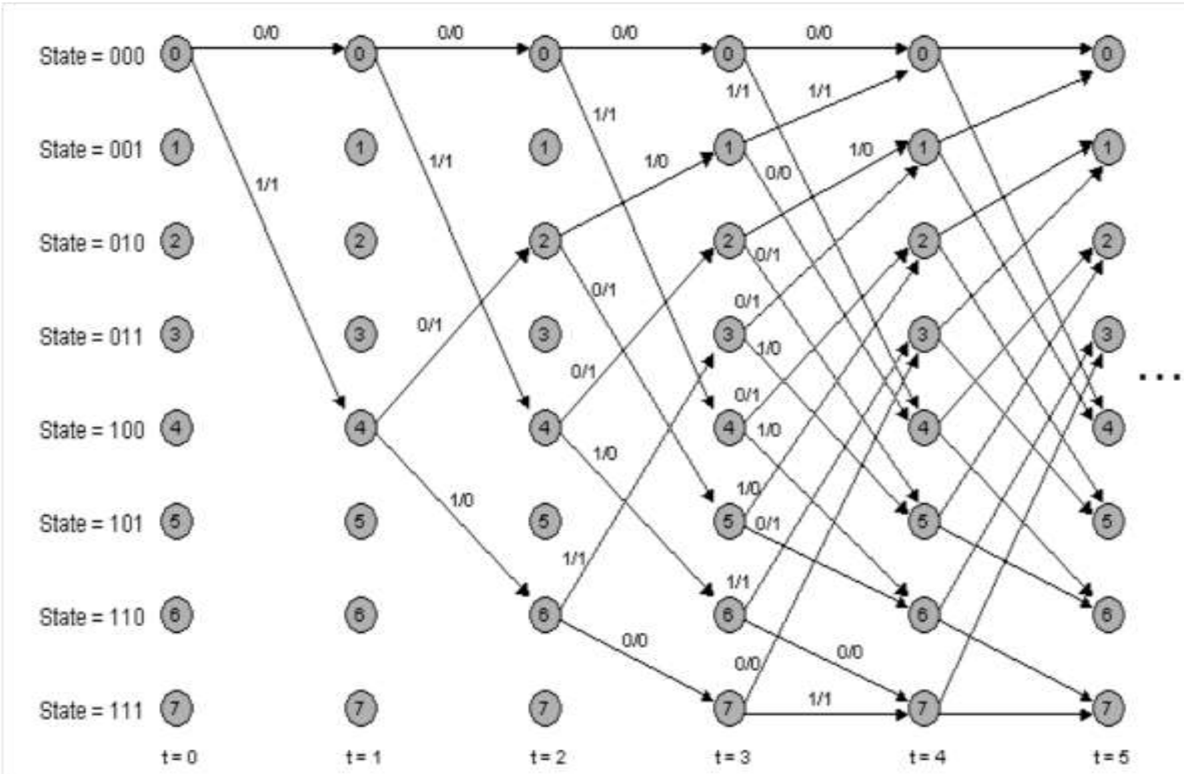

Fig 10: Trellis Diagram for one 8 State Constituent Encoder [8]

It finds which path has the highest likelihood by considering the Hamming distance between incoming bits and possible transitions in the encoder (or Trellis) as a metric. The BCJR algorithm, which produces a soft estimate for each bit by considering the incoming bits as a maximum a- posteriori probability (MAP) detection problem. But the viterbi algorithm finds the most likely sequence and instead of maximizing the likelihood function for each bit it estimates several bits at once. So BCJR algorithm has the best performance than the Viterbi algorithm. Consider a constituent encoder with its trellis diagram, at which several possible paths are available. The amount of memory required to calculate the all possible paths is very large. So to reduce the amount of memory viterbi introduces paths through the Trellis diagram with smallest Hamming distance are known as survivor paths. Consider K is the constraint length of the encoder that is the encoders memory plus one (K = M+1), the Viterbi only takes 2^{K-1} survivor paths. The Viterbi algorithm works well on the small frames on the Trellis diagram. So for each iteration, decision of the best path is calculated. The decoding window moving forward through the branch and depends on the code in the Frame new decisions are made.

C. SOVA ALGORITHM

SOVA algorithm is proposed by Hangenauer and Hoeher 1989. It is a modified form of Viterbi algorithm [3]. The reliability of bit sequences or the a- posteriori probabilities of the state transitions are produced by this algorithm**.** There are two major modifications used from the Viterbi algorithm to SOVA, those are the maximum likelihood path selected by path metrics is modified and the algorithm is modified to provide soft output to every decoded bit. At low Signal to noise ratio SOVA's estimation of probability is good. This algorithm has higher similarities to the viterbi algorithm except some modifications like computing the transition and bit reliabilities. If there is a difference between two path metrics then the reliability of bits is updated. In addition to the most likely path sequence, a reliability value of each estimated bit is calculated using this sub-optimum algorithm. In the SOVA algorithm the soft output is updated by considering two path sequences, which is named as survivor and concurrent path sequences. Several modifications are done to improve the SOVA algorithm. That is the normalisation of the extrinsic information is takes place by multiplying using a correcting factor, which depends on the variance of the decoder output and by inserting two or more correcting coefficients the correlation in the decoder input is achieved.

D. ITERATIVE TURBO DECODING

In the first iteration consider the first component decoder. The decoder receives the transmitted systematic bits, and the parity bits, from the first encoder [4]. Half of the parity bits must be punctured at the transmitter to obtain the half rate code. For the punctured bits the turbo decoder must insert zeros in the soft channel output. Then the soft channel inputs are processed by the first component decoder to produce its estimate of the conditional LLRs of the data bits. Consider u_k be the input information bit, The a-posteriori log likelihood ratio in the first iteration from the first component decoder is represented by using $L_{11}(U_K / Y)$. The first component decoder will have no a-priori information about the bits in the case of first iteration [12]. The channel sequence containing the interleaved version of the received systematic bits, and the parity bits from the second encoder received by the second component decoder. In the case of Turbo decoder, if the parity bits generated by the encoder are punctured before transmission it will need to insert zeroes in to this sequence. Decoder can use the conditional LLR provided by the first component decoder to generate a-priori LLRs. extrinsic information from the component decoder is used as the a-priori LLRs in iterative turbo decoder. Then arrange the decoded data bits after being interleaved by the same order as they were encoded by the second encoder. At the end of the first iteration the second component decoder uses the received channel sequence and the a-priori LLRs to produce its a-posteriori LLRs. Then the first component encoder again processes its received channel sequence at the second iteration. When iterative process is continues average the BER of the decoded bits will fall.

D. LOG MAP DECODING ALGORITHM

The MAP decoding algorithm and the LOG MAP Decoding algorithm [9] is based on the same idea. But the benefit of this algorithm is it simplifies the computation by discarding the multiplicative operations. The multiplicative operations, which is computationally more expensive by comparing it in to the addition operations in terms of the processing speed of the microprocessor. The implementation of this algorithm is very difficult that is to store the probabilities in the computation of the log-likelihood ratio it needs large amount of memory. The decoder structure used in this algorithm is shown in below, where Λ_2, Λ_1 represents the a-priori information and $\Lambda_{1e}, \Lambda_{2e}$ represents the a-posteriori information.

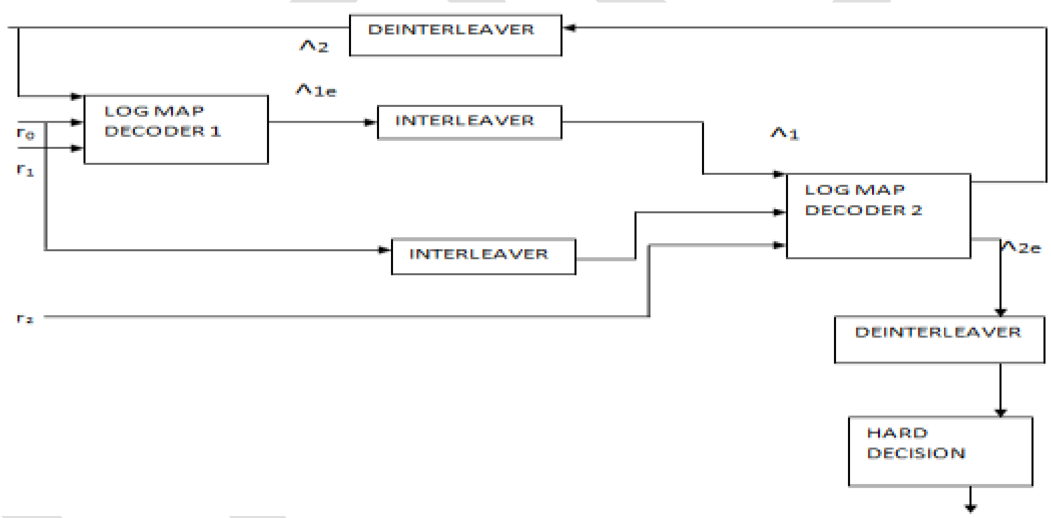

Fig 9: Log-MAP algorithm [9]

CONCLUSION

The Turbo code is a high performance forward error correcting code. The forward error correction is capable of locating the positions, where the errors occurred and which is corrected. The Turbo code has the parallel structure, in which Recursive systematic convolutional (RSC) codes working in parallel is used to create the "random" versions of the message. Two or more RSC codes, each with a different interleaver is involved in the parallel structure. Then the details about the serial concatenated convolutional code and the hybrid concatenated convolutional code are also described. Turbo decoding is takes place on the noisy versions of systematic bits and two sets of parity bits to produce an estimate of original message bits. The Log-MAP algorithm has high performance as compared with soft output viterbi algorithm decoding scheme ie, MAP decoding scheme takes an approximation from this SOVA decoder. So that MAP algorithm gets superior performance than the soft output viterbi algorithm.

REFERENCES:

[1] Raffi achiba mehrnaz mortazavi booz,allen & hamilton, inc mclean, virginia william fizell "Turbo code Performance and Design Trade-offs" milsatcom jteo falls church, virginia. 0-7803-6521-6/$10.(0c0) 2 000 ieee.
[2] Sarpreet Singh, Rupinder kaur "Techniques for Turbo Decoding Using Parallel Processing, Comparative Analysis" International Journal of Advanced Research in Computer Science and Software Engineering.
[3] Poonam, Mr. Bhavneesh Malik, Mr. Gaurav Kochar. " A Review on Turbo Codes using Different Algorithms" International Research Journal of Engineering and Technology (IRJET).
[4] Jason P. Woodard and Lajos Hanzo. "Comparative Study of Turbo Decoding Techniques: An Overview" ieee transactions on vehicular technology, vol. 49, no. 6, november 2000.
[5] Claude Berrou, Ramesh Pyndiah, Patrick Adde, Catherine Douillard and Raphaël Le Bidan. "An Overview of Turbo Codes and Their Applications" GET/ENST Bretagne, Laboratoire TAMCIC (UMR CNRS 2872), PRACom Technopôle Brest Iroise.
[6] Simon haykin. "Communication Systems" fourth edition, john wiley&sons inc.
[7] Jesper Kjeldsen. "Turbo Coding" Aalborg University Institute for Electronic Systems.
[8] Prathyusha Allala. "Genetic Optimization of Turbo Decoder" the School of Electrical Engineering and Computer Science and the Russ College of Engineering and Technology.
[9] Rupinder kaur, Sarpreet Singh. "Parallel Processing Based Turbo Decoder Design using Viertibi Algorithm" International Journal of Application or Innovation in Engineering & Management (IJAIEM).
[10] A. Burr. "Turbo-codes: The Ultimate Error Control Codes" electronics & communication engineering journal august 2001.
[11] Guido montorsi. "Unveiling Turbo codes: Some Results On Parallel Concatenated Codes" conference paper · october 1995.
[12] Sorin adrian barbulescu. "Iterative Decoding of Turbo codes and other Concatenated Codes" The school of electronic engineering faculty of engineering university of south Australia.