International Journal of Research in Advent Technology, Vol.2, No.7, July 2014

E-ISSN: 2321-9637

Comparison of Single Stage And Double Stage Interleaver Performance

Sumit Kumar[1], Hemant Dalal[2]
Department of Electronics and Communication[1, 2], CBS Group of Institution[1, 2]
Email: kumarsumit4201990@gmail.com [1], hemantdalal87@gmail.com [2]

Abstract- This paper represents the need of error correction codes in communication system. The us eof turbo code in iterleaver and use of modified turbo code at the interleaver stage enhance the system performance. It enhance the idea of use and design of double stage interleaver. Such implementation improves the BER of the system in comparision to single stage interleaver.

Index Terms- CTC, MTC, SNR, BER, ECC.

1. INTRODUCTION

The efficient design of a communication system that enables reliable high-speed services is challenging. 'Efficient design' refers to the efficient use of primary communication resources such as power and bandwidth. The reliability of such systems is usually measured by the required signal-to-noise ratio (SNR) to achieve a specific bit error rate [John G. Proakis, 2001]. A bandwidth efficient communication system with perfect reliability, or as reliable as possible using as low as possible SNR is desired.

Error correction coding (ECC) [C. Heegard, 1999] is a technique that improves the reliability of communication over a noisy channel. The use of the appropriate ECC allows a communication system to operate at very low error rates, using low to moderate SNR values, enabling reliable high-speed communication over a noisy channel. Although there are different types of ECC that can be used for channel coding, they all have one key objective in common, namely, achieving a high minimum Hamming distance to improve the code performance that occurs only for few codeword.

1.1. *Shannon Limit*

Both the communication channel and the signal that travels through it have their own bandwidth. The bandwidth *B* of a communication channel defines the frequency limits of the signals that it can carry. In order to transfer data very quickly, a large bandwidth is required. Unfortunately, every communication channel has a limited bandwidth

In 1948, Shannon's theory set the fundamental limits on the efficiency of communications systems. Shannon theory states that probability of error in the transmitted data can be reduced by an arbitrary amount provided that the rate at which data is transmitted through the channel does not exceed the channel capacity and formulated by Shannon as:

$$C = W log_2\left(1 + {}^{S}/_{N}\right) bits/sec \quad \ldots(1.1)$$

With C being the channel capacity, the maximum amount of bits that can be transmitted through the channel per unit of time, W the bandwidth of the channel and S/N being the signal to noise ratio (SNR) at the receiver.

This theory went against the conventional methods of that time, which consisted of lowering probability of error by raising SNR, means by increasing the power of the transmitted signal. Unfortunately, although Shannon's theorem sets down the fundamental limitations upon the on communication efficiency, it provides no methods through which these limits can be reached.

1.2. *Channel coding*

The task of channel coding is to encode the information sent over a communication channel in such a way that in the presence of channel noise, errors can be detected and/or corrected. There are two types of channel coding technique.

1.2.1. Backward Error Correction (BEC) Coding Technique

Backward error correction [John G. Proakis, 2001] is a technique used for error detection only. At the receiver error can be detected by the redundancy bits added by the encoder at the transmitter but can not be corrected at the receiver end. If an error is detected, the sender is requested to retransmit the message. While this method is simple and sets lower requirements on the code's error-correcting properties, it on the other hand requires duplex communication and causes undesirable delays in transmission. This technique is used where delay in the transmission can be tolerated.

1.2.2. Forward Error Correction (FEC) Coding Technique

Forward error correction [John G. Proakis, 2001] is a technique used for error detection as well as correction at the receiver end. In FEC technique redundancy bits are added to the information bits using channel encoder. If an error is detected at the receiver end, the message in not retransmitted, while it is corrected using redundancy bits added by channel encoder. In FEC technique channel decoder is capable of detecting as well as correcting a certain numbers of errors, i.e. it is capable of locating the position of error. Number of error that can be corrected at the receiver end depends on how much bit error correcting code is used at the transmitter end by the channel encoder. An N bit error correcting code can detect N+1 bit of error and correct N bit error. Since FEC codes require only simplex communication, they are especially attractive in wireless communication systems. FEC technique improves the energy efficiency of such systems. Designing a channel code is always a trade-off between energy efficiency and bandwidth efficiency. Codes with lower rate (i.e. bigger redundancy) can usually correct more errors [Molisch, 2011]. If more errors can be corrected, the communication system can operate with a lower transmit power, transmit over longer distances, tolerate more interference, use smaller antennas and transmit at a higher data rate. These properties make the code energy efficient. On the other hand, low-rate codes have a large overhead and are hence consume more bandwidth. Also, decoding complexity grows exponentially with code length, and long (low-rate) codes set high computational requirements to conventional decoders.

There is a theoretical upper limit on the data transmission rate R, for which error-free data transmission is possible. This limit is called channel capacity or also Shannon capacity. Although Shannon developed his theory already in the 1940s, several decades later the code designs were unable to come close to the theoretical limit due to decoder complexity.

Hence, new codes were sought that would allow for easier decoding. One way of making the task of the decoder easier is using a code with mostly high-weight code words. High-weight code words, i.e. code words containing more ones and less zeros, can be distinguished more easily. Another strategy involves combining simple codes in a parallel fashion [Li. Ping, 2001], so that each part of the code can be decoded separately with less complex decoders and each decoder can gain from information exchange with others. This is called the divide-and-conquer strategy. Turbo codes use second method to achieve near Shannon limit performance.

2. SYSTEM DESIGN

CTC encoder consists of parallel concatenation of two rate $R = 1/2$ RSC encoders using random Interleaver. Fig. 1 below shows component used to design this turbo encoder block. Parameter trellis structure defines number of state, constrained length, code generator and feedback connection for convolutional encoder. Trellis structure is given by generator polynomial. Random interleaver interleaves the information bit sequence using random permutations.

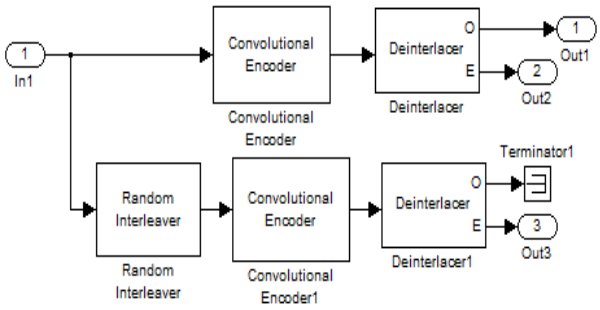

Fig. 1 TCC Encoder

Deinterlacer separates the elements of the input signal to generate the output signals. The odd-numbered elements of the input signal become the first output signal, while the even-numbered elements of the input signal become the second output signal. To adjust code rate from $R = 1/4$ to $R = 1/3$ the odd bit sequence of second convolutional encoder is terminated. Parameters used for different component of CTC encoder are shown below in the table 1.1.

Table 1.1 parameters for CTC encoder

RSC Encoder	Parameters	Values
	Trellis	Poly2trellis(5, [37 21],37)
	Output	uncated (reset every frame)
Random terleaver	o. of Element	1024*128
	Initial seed	54123

- **Parallel to serial converter and Serial to Parallel Converter**

Parallel to serial converter is used in the transmitter to concatenate output of Deinterlacer to transmit signal through the channel. At the receiver end the received signal is converted back to parallel form using select row block.

- **Puncturing and Padding Zeros**

Puncturing is used to adjust code rate at the transmitting end. Puncturing vector define puncturing vector. The puncturing vector used is $[1\ 1\ 0\ 1\ 0\ 1]$. Puncturing vector shows that 3^{rd} and 5^{th} bit of every six bit is not transmitted. Padding zeros block is used at the receiving end. Zeros are padded using same vector as puncturing vector used at the transmitting end. Two zeros are added for every four bits of received signal. Zeros are added at the position of punctured bit.

- **AWGN channel**

AWGN channel add white Gaussian noise to the input signal. The input and output signals can be real or complex. When the input signal is real this block adds real Gaussian noise and produces a real output signal. When the input signal is complex, this block adds complex Gaussian noise and produces a complex output signal. Probability distribution for noise is Gaussian distribution depends on the variance. Variance of the channel is calculated using the equation 3.1.

$$Noise\ variance = signal\ power \times symbol\ period \Big/ sampel\ time \times 10^{\frac{E_s/N_0}{10}} \quad \ldots(2)$$

- **Iterative SISO decoder**

SISO iterative decoder is used for decoding turbo code. Soft information is exchanged between two decoders. Soft output ($L(u)$) of first decoder is used by second decoder after interleaving, to make a decision about APP of information bits. Soft output of second decoder is fed back to first decoder after deinterleaving and suitable delay. Random deinterleaver is used and delay value should be multiple of interleaver length. APP of parity bits is terminated using terminator.

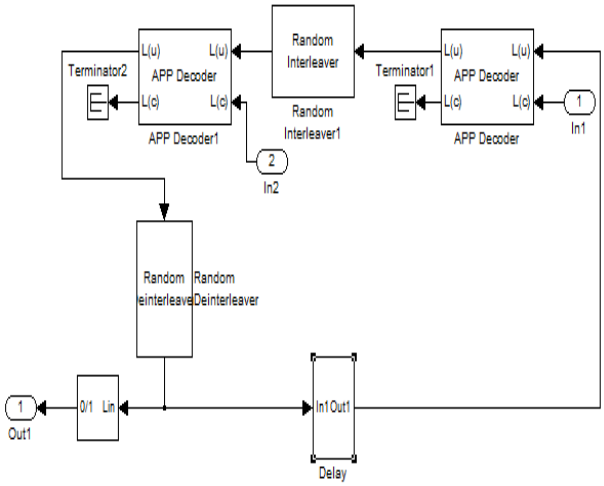

Fig.2 Iterative SISO CTC Decoder

Table 1.2 Parameters for SISO Decoder

Name of Block	Parameter	Value
APP Convolutional decoder	trellis	Poly2trellis (5, [37 21],37)
	Termination Method	Truncated
	Number of Scaling Bits	3
	Decoding Algorithm	Max Log MAP
Random Interleaver and Deinterleaver	Number of Elements	1024*128
	Seed	54123
Delay	Delay Samples	1024*128

Hard decision about the information bits is made by likelihood to binary transformation block. Information bit is decoded as one if its APP is greater than positive otherwise decoded as zero.

- **Error Rate Calculation**

The Error Rate Calculation block compares input data from the transmitter with input data from the receiver. It calculates the error rate by dividing the total number of unequal pairs of data elements by the total number of input data elements from one source. This block can be used to compute either symbol or bit error rate, because it does not consider the magnitude of the difference between input data elements. If the inputs are bits, then the block computes the bit error rate. If the inputs are symbols, then it computes the symbol error rate. The block output is a three-element vector consisting of the error rate, followed by the number of errors detected and the total number of symbols compared. This vector can be sent to either the workspace or an output port. Table 3.4 shows parameter used for error rate calculation block.

Table 1.3 Parameter for Error Rate Calculation

Parameter	Value
Receive Delay	0
Computation Delay	0
Computation mode	Entire Frame
Output Data	Port

- **Display**

Display block display the value of BER calculated by error rate calculation block. The amount of data that appears and the time steps at which the data appears depend on the Decimation block parameter and the `Sample Time`. The Decimation parameter enables to display data at every *n*th sample, where *n* is the decimation factor. The default decimation, `1`, displays data at every time step. The `Sample Time`, which can be set with set_param, specifies a sampling interval at which to display points.

Table 1.4 Parameter for Display

Parameter	Value
Output Format	Short
Decimation	1

Turbo Decoder

In a typical communications receiver, a demodulator is often designed to produce soft decisions, which are then transferred to a decoder. Such a decoder could be called a *soft input/hard output* decoder. With turbo codes, where two or more component codes are used, and decoding involves feeding outputs from one decoder to the inputs of other decoders in an iterative fashion, a hard-output decoder would not be suitable. That is because hard decisions as input to a decoder degrade system performance (compared to soft decisions). Hence, what is needed for the decoding of turbo codes is a *soft input/soft output* decoder [J. Hagenauer, 1995]. A decoding algorithm that accept a priori information as its input and produces a posteriori information as its output is called a *soft input soft output* algorithm.

Fig.3 shows the schematic diagram for SISO turbo iterative decoding [C. Berrou, 1996]. It consists of two SISO decoder connected in a closed fashion through interleaver and deinterleaver. The two decoders exchange their soft information to improve their estimates for information sequence.

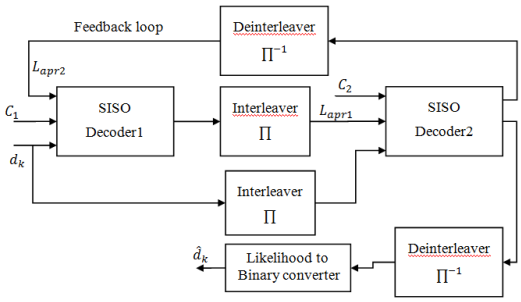

Fig. 3 SISO Turbo Decoder structure

First decoder takes as input the received information sequence, output codeword of encoder1 and soft information feedback from second decoder. Second decoder produce a priori probability estimate of the information sequence using the soft output produced by first decoder and output codeword of second encoder. After a certain number of iteration the output of the two decoders become nearly same and improvement in the performance due to information exchange becomes negligible. SISO decoder stops further iteration and output of the last stage output make a hard decision of the information sequence.

There are two main algorithms to decode the data. One is Viterbi Algorithm (VA) [G. D. Forney, 1973] and the other one is Maximum A Posteriori Algorithm (MAP) [P. Robertson, 1995]. The first algorithm finds the most probable output data se
with the least Hamming distance is found. The MAP algorithm finds the marginal probability that the received bit was 1 or 0. Since the bit could occur in many different codeword, the sum of all the probabilities is considered. The MAP algorithm is preferred as it minimises the bit error probability.

3. SIMULATION RESULTS

Simulation result shows that BER performance improves as signal to noise ratio increases and BER converges at a good rate as signal to noise ratio increases. From the simulation result it is observed that, as the number of iterations increases BER performance improves.

BER performance of the system improves for iterative decoding as number of iteration increases. Ideally iterative decoding is stopped when APP of both the decoder is exactly equal. This will take a long time to decode information bit sequence so iterative decoding is stopped when improvement in BER performance of successive iteration is nearly equal. Hard decision about the information bit is taken when BER performance for successive iteration shows no improvement. We obtain different results for MTC

with double stage interleaver and compare single stage and double stage interleaver.

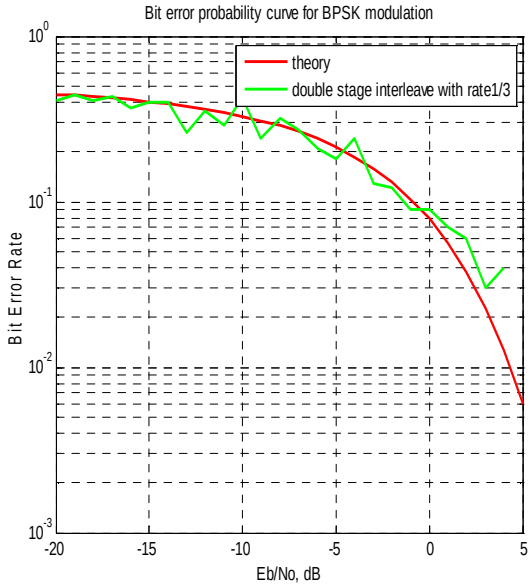

Fig. 4 BER for code rate 1/3 for double stage interleaver

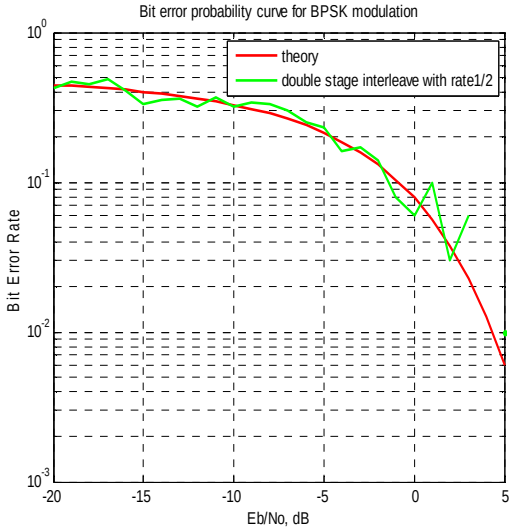

Fig. 5 BER for code rate 1/2 for double stage interleaver

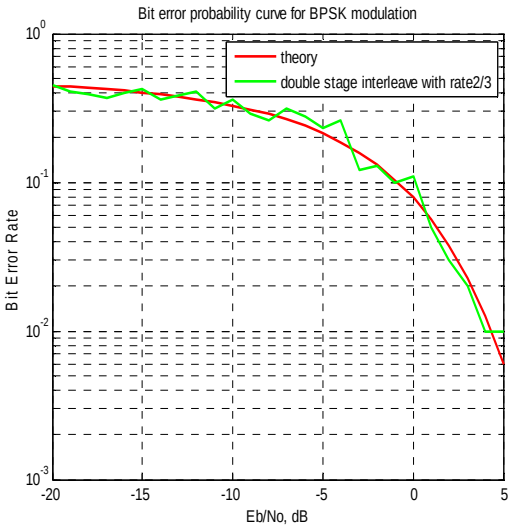

Fig. 6 BER for code rate 2/3 for double stage interleaver

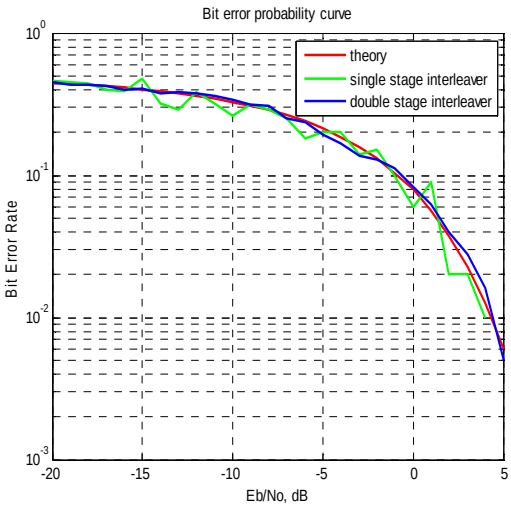

Fig. 7 Comparison of BER of single stage and double stage interleaver.

4. CONCLUSION

Conclusion from this research work is that decoder complexity is reduced by a factor of nearly two for MTC as compared to CTC with a negligible loss of BER performance. This loss of BER is compensated by reduction of decoder complexity. Memory requirement is much less for decoding MTC as compared to CTC. Double stage interleaver performs much better than the single stage interleaver.

REFERENCES

[1] John G. Proakis, Digital Communications, 4th Edition, McGraw-Hill International Edition, 2001

[2] C. Heegard and S. B. Wicker, Turbo Coding, Kluwer Academic Publishers, 1999.

[3] Molisch, wireless communications, Wiley publication Ltd. 2nd edition , 2011.

[4] Li. Ping, "Turbo-SPC Codes," *IEEE Trans. Comm.*, vol. 49, No. 5, pp. 754-759, May 2001.

[5] Li. Ping, X. Huang and N. Phamdo, "Zigzag Codes and Concatenated Zigzag Codes," *IEEE Trans. Inform. Theory*, vol. 47, No. 2, pp. 800-807, Feb. 2001.

[6] J. Hagenauer and P. Hoeher, "Viterbi algorithm with soft-decision output and its applications," *Proc., IEEE*, pp. 1680-1686, 1989.

[7] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Comm.*, vol. 44, pp. 1261–1271, Oct. 1996.

[8] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in proc. *IEEE ICC'93*, vol. 2, pp.1064-1070, May 1993.

[9] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 218–278, Mar. 1973.

[10] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and suboptimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Comm. (ICC'95)*, pp. 1009–1013, June 1995.

[11] P. Robertson and P. Hoeher, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *IEEE Trans. Comm.*, vol. 8, pp. 119–125, Mar.-Apr. 1997.

[12] Bharat, K.; Broder, A. (1998): A technique for measuring the relative size and overlap of public Web search engines. Computer Networks, **30**(1–7), pp. 107–117.

[13] Broder, A.; Kumar, R.; Maghoul, F.; Raghavan, P.; Rajagopalan, S.; Stata, R.; Tomkins, A.; Wiener, J. (2000): Graph structure in the Web. Computer Networks, **33**(1–6), pp. 309–320.

[14] Chakrabarti, S. (2000): Data mining for hypertext: A tutorial survey. SIGKDD explorations, **1**(2), pp. 1–11.