Efficiency Enhancement Techniques for Wireless Communication Systems

Ranjeet Singh Tomar and G.S. Tomar

Department of Electronics & Communication Engineering, ITM, Gwalior, India
Machine Intelligence Research Labs, India Part, Gwalior 474011 India

E-mail: yuvraajrst@yahoo.co.in & gstomar@ieee.org

***Abstract* - In this paper Forward Error correction Techniques have been analysed and investigated. It is used to enhance the efficiency and accuracy of information transmitted over a channel. The principle this method is to add some redundancy to the data to be transmitted over a channel. Amongst so many technique of FEC coding, Turbo codes are more commonly used now a days due to its compatibility. Turbo Codes have been licensed on a worldwide basis to ST Microelectronics. The investigation of applications of turbo codes has produced new concept for spatial transmission with greater efficiency. this paper aims at getting efficient and reliable communication.**

Keywords: Turbo Codes, Spatial transmission, FEC.

I. INTRODUCTION

In a communication transmission system, data is transferred from a transmitter to a receiver across a physical medium of transmission or wireless channel [1]. The channel wired or wireless is generally affected by noise or fading, which introduces errors in the data being transferred from sender to receiver [2]. The transmission of data over a specific channel has to be pre defined with specific bandwidth and data rate for evaluation of channel [3]. During transmission of data over a channel, errors are obvious having depth as per prevailing conditions instantaneously [4]. For correcting the errors, we have to use error control strategies. There are two main error control strategies adopted in communication systems:

1. Error correcting techniques
2. ARQ technique

II. ERROR CORRECTING TECHNIQUES

In communication systems, the introduction of errors while data is over a channel is big issue to be considered for efficient and reliable system. To overcome such errors in data, forward error correction coding is used to enhance the efficiency and accuracy of information transmitted. If 'n' number of bits is to be sent, a FEC system will add 'k' redundancy bits, therefore total n+k bits are sent through the transmission channel [5]. Such a redundancy will help the receiver to detect any error inside received data and, depending on the FEC algorithm, receiver can correct data without querying the sender for more information, this is why it is called Forward Error Correction technique. There are many Forward Error Correction techniques amongst them turbo codes are more commonly used nowadays due to their compatibility and simplicity [6].

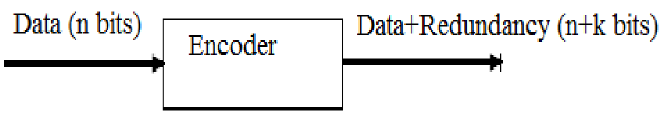

Fig.1 Forward Error Correction technique

A Turbo code is basically a composition of several FEC components, often placed in parallel but sometimes in other schemes, for example in serial or mixed [7]. The idea is to have several FEC encoders that take the same data pool as input but with bits ordered differently inside this data pool. To be more precise, it is a bit like if each FEC encoder had a different view angle of the data, therefore producing a different redundancy. At the opposite, the decoding part uses the same number and types of FEC components, placed identically. As each decoder has its own view angle and redundancy of the data, one decoder may correct some errors that other decoders won't be able to correct and vice versa. For example this case is often encountered with burst errors. The key point of turbo code architecture is that each decoder sends its corrections to next decoder. The last decoder sends its corrections to the first one, involving an incremental and recursive correction process [8]. After a certain number of iterations, the data has been greatly corrected and the error rate is quite low, allowing the consumer to use it. The number of iterations on the data can be decided according to the size of the data and type of channel with probability of errors during transmission. Turbo Code is the first coding scheme that approaches the theoretical limit expressed by the Shannon-Hartley law. The code approaches up to 0.5 dB of the theoretical limit, while conventional code needs about 3 to 4 dB more than the theoretical limit [9]. This has enhanced the acceptability of the codes and thus accuracy of the communication system using such codes.

III. THE LIMIT TO CAPACITY

In the field of Information theory, Shannon calculated a theoretical maximum rate at which data could be transmitted over a channel perturbed by additive white Gaussian noise (AWGN) with an arbitrarily low bit error rate [10]. This maximum data rate, the capacity of the channel, was shown to be a function of the average received signal power, the average noise power N, and the bandwidth of the system W. This standard function is known as the Shannon-Hartley Capacity [8], which can be stated as.

$$C = W\log_2(1+\frac{S}{N})$$

If W is in Hz, then the capacity, C, is in bits/s. Shannon stated that it is theoretically possible to transmit data over such a channel at any rate $R \le C$ with an arbitrarily small error probability by use of a sufficiently complicated coding scheme. Rather than using S/N, We often use information bit energy per bit to noise power spectral density ratio Eb/No, to allow systems with different coding or modulation schemes to be compared on an equal basis [11]. The two quantities are related by

$$\frac{S}{N} = \frac{E_b}{N_o} \times \frac{R}{W}$$

Shannon theory also states that there is a limiting value of Eb/No below which there can be no error free transmission at *any* data rate. Using the identity

$$\lim_{x \to 0} (1+x)^{1/x} = e$$

We can calculate this limiting value of Eb/No_ by letting

$$x = \frac{E_b}{N_o} \cdot \frac{C}{W}$$

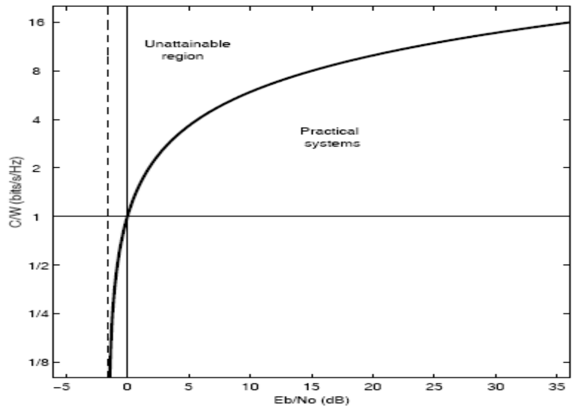

Fig.2 Normalized Channel Capacity versus Channel SNR

Then we can rewrite Equation

$$\frac{C}{W} = x\log(1+x)^{1/x}$$

$$1 = \frac{E_b}{N_o}\log_2(1+x)^{1/x}$$

In the limit, as $\frac{C}{W} \to 0$

$$\frac{E_b}{N_o} = \frac{1}{\log_2 e} = 0.693$$

This value expressed in dB is Eb/No = -1.59 dB, representing the Shannon bound on Eb/No. It is not possible in practice to reach this limit, because this would require codes so long, and of such complexity, that it would be impracticable to decode them. Shannon's work initiated a search for codes that could approach the bounds shown in Fig.1 and yet have realizable decoding complexity. By the late 1980's, performance within a few dBs of the Shannon bound had been demonstrated using powerful serially-concatenated codes. These codes tended to comprise a Reed-Solomon outer code in series with a long constraint length convolution inner code. However, Viterbi decoding of such convolutional codes is slow and memory intensive [2]. Then, in 1993, Claude Berrou and his colleagues at ENST, France, announced a new type of powerful error control code. Berrou and his colleagues showed that a rate 1/2 turbo code, Frame size 65536 and 18 decoding iterations could achieve BER = 10^{-5} at Eb/No = 0.7 dB with QPSK transmission over the AWGN channel. At this spectral efficiency (C/W = 1 bit/s/Hz), the Shannon theoretical limit for arbitrarily low BER transmission is Eb/No = 0 dB. Therefore, Berrou and his colleagues had approached within 0.7 dB of the Shannon limit. Further, this performance could be achieved with a practical, iterative decoding method [13]. This brought hopes of correcting errors in large size of data and limited bandwidth of the channel, which is the normal constraint around the upcoming technological world.

IV. TURBO ENCODER

In a simplified turbo code, there are two convolutional encoders in parallel. The information bits are scrambled before entering the second encoder. The codeword in a turbo code consists of the input bits - i.e. the code is systematic – followed by the parity check bits from the first encoder and then the parity bits from the second encoder, as depicted in Figure 3. The simplified turbo code block diagram in Figure 3 shows only two branches. In general, one can have multiple turbo encoders with more than two branches. The convolutional code at every branch is called the constituent code (CC). The CCs can have similar or different generator functions. We will concentrate on the usual configuration with two branches having the same CC. A PAD is shown in the figure to append the proper sequence of bits to terminate all the encoders to the all-zero state. This is because a convolutional code may be used to generate a block code if we use beginning and tail bits [12]. If we have one encoder then the required tail is a sequence of zeros with length equal to the memory order *m*. The problem of terminating both encoders simultaneously seems to be difficult because of the interleaver. However, it is still possible to do with *m* tail bits only. In general we can have another interleaver before the first encoder but usually it is replaced with a delay line to account for the interleaver delay and keep the branches working simultaneously [14]. Puncturing can be introduced to increase the rate of the convolutional code beyond that resulting from the basic structure of the encoder. Some codes are called *recursive* since the state of the internal shift register depends on the past outputs. Figure 4 illustrates a non-recursive nonsystematic convolutional code with its corresponding recursive systematic code. C1 and C2 are the check bits. Note that for the systematic convolutional encoder, one of the outputs, C1, is exactly the input sequence. In turbo codes Recursive Systematic Convolutional (RSC) codes are proved to perform better than the non-recursive ones. An RSC encoder can be obtained from an non-systematic non-recursive encoder by setting one of the outputs equal to the input (if we have one input) and using a feed back. The trellis and the free distance (d_{free}) also will be the same for both codes. Of course, the output sequence does not correspond to the same input in the two codes because the two generator functions are not the same.

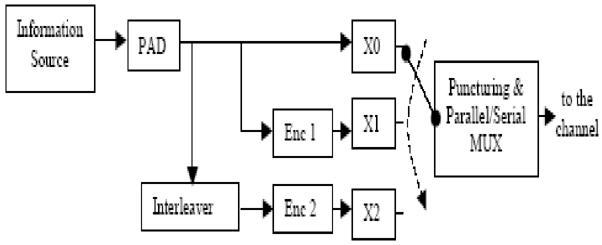

Fig.3 Simplified Turbo Encoder

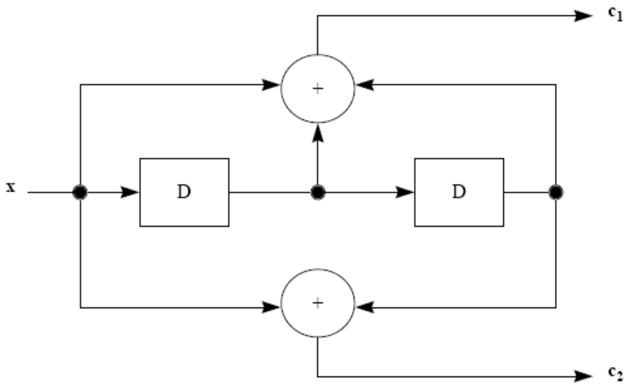

Fig.4 (a) Conventional convolutional encoder (Non- Recursive Non-Systematic Code) with r=1/2 and K=3.

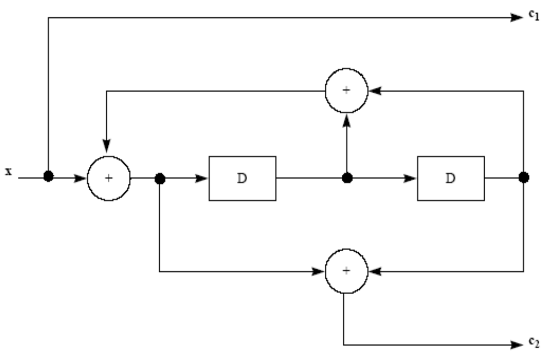

Fig.4 (b) The RSC encoder obtained from Fig.4 (a) with r=1/2 and K=3.

V. TURBO DECODER

Receiver has two decoders, one for each encoder. As shown in Figure 5, decoding is done in an iterative process where the decoders use feedback to share information with each other after completion of each iteration process. Each component decoder accepts "soft" a priori info and output "soft" a posteriori info (SISO). "Turbo" means exchanging soft output iteratively. In practice the number of iterations does not exceed 18, and in many cases 6 iterations can provide satisfactory performance [15]. Actually, the term turbo code is given for this iterative decoder scheme with reference to the turbo engine principle. The first decoder will decode the sequence and then pass the hard decision together with a reliability estimate of this decision to the next decoder. Now, the second decoder will have extra information for the decoding; a priori value together with the sequence [13]. Additional coding gain can be achieved by sharing information between the two decoders in an iterative fashion. This is achieved by allowing the output of one decoder to be used as a priori information by the other decoder. Each decoder estimates the a posteriori probability (APP) of each data bit, which is used as a priori information by the other decoder. Decoding continues for a set number of iterations. Performance generally improves till the threshold level.

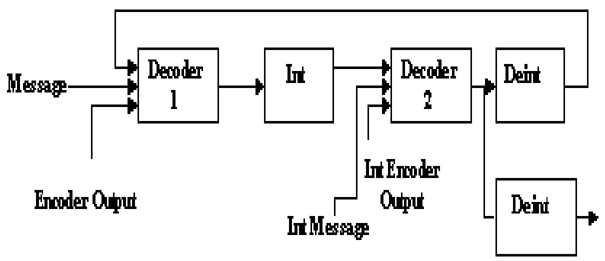

Fig.5 Block Diagram of a Turbo Decoder

The interleave in-between is responsible for making the two decisions uncorrelated and the channel between the two decoders will seem to be memory less due to interleaving. There are two well-known soft-input / soft-output decoding methods, namely, MAP (Maximum A Posteriori) decoding algorithm and SOVA (Soft Output Viterbi Algorithm)

VI. SIMULATION RESULTS

In this experiment the simulated results available are shown in Figure-7. The plot is between BER with respect to SNR for different frame sizes. It shows that increase in the frame size for decoding iteration 8, BER is improved for higher SNR. Higher frame size and SNR gives better result i.e. reduced BER. Increase in the frame size results to increase in latency. Hence this type of reconfiguration may be suitable for delay insensitive applications e.g. data transfer. Figure-8 shows the effect of puncturing on BER. It is obvious that code rate is increased by puncturing the encoder output. This way bandwidth requirement is reduced for transferring the same amount of information. But Puncturing results to the increased BER. Hence this type of reconfiguration may be suitable where bandwidth requirement is low and QoS is not that important e.g. video telephony. Figure-6 shows the result of BER with respect to SNR for different number of decoding iterations and fixed frame size. It shows that increase in decoding iterations results to higher decoding time and latency. By increasing the number of decoding iterations, BER is improved until threshold is reached. Figure-6 also shows that number of decoding iterations required is lesser for higher SNR till the threshold level. Table-1 shows the result of the simulation for AWGN (additive white Gaussian noise) channel. In this simulation different frame sizes and different code rate are used. The table shows the performance of the turbo coded system with unpunctured code rate 1/3 and punctured code rate of ½ for constraint length 3 and decoding iteration 8. The result shows that an acceptable BER's (10^{-4}) can be obtained in AWGN channel by having the Eb/No (ratio of energy per bit to the noise power spectral) to approximately 1.5 dB with a code rate of 1/3 and. Similarly an acceptable BER's (10^{-4}) was obtained in AWGN channel with the code rate 1/2 and Eb/No=2.0 dB at frame size 1024 and can be further reduced by increasing frame size. Latency keeps increasing with increased frame size. Assuming a practical data speed of 128 kbps in future communication systems, and seeing the BER results in Table-1 a 256 bit frame size has BER of order 10^{-3} and is appropriate for real time voice applications. The 1024 bit frame size has a BER of order 10^{-4} and is appropriate for real time videoconference and JPEG images. The 4096 bit frame has a BER of order 10^{-5} and longer latency. Hence this is appropriate for playback of compressed video. The 16384 bit frame has a BER of order 10^{-6} and long latency. Hence this is appropriate for data and file transfer. Figure-6, 7, 8, and Table-1 shows the BER with respect to several parameters of the turbo code for constraint length 3. This shows that turbo code can achieve different level of BER with the same network infrastructure.

A. *Effect Of Decoding Iterations*

Frame size 1024, Rate

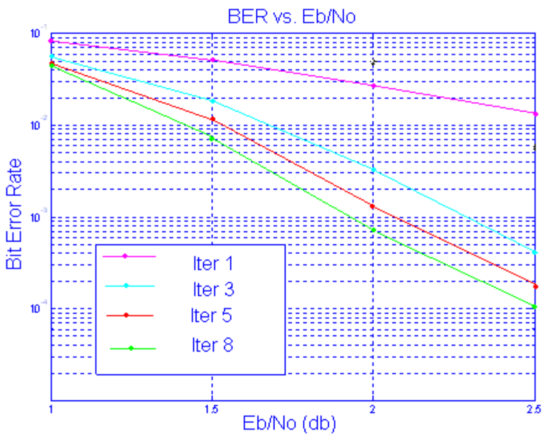

Fig. 6(a) BER with number of iterations

B. *Effect Of Decoding Iterations*

Frame size 1024, Rate 1/3

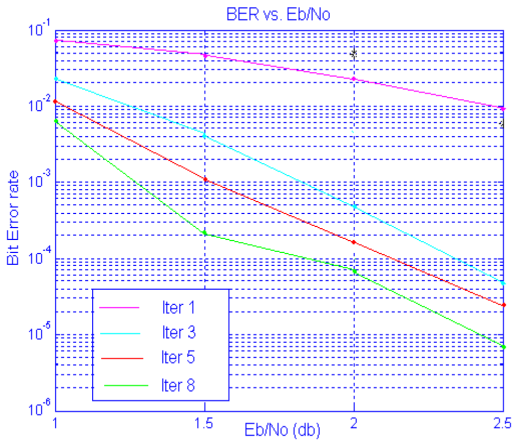

Fig.6 (b) BER with number of iterations

Figures 6, 7, 8 show the reconfiguration possibility of the turbo code in terms of frame size, puncturing and decoding iterations. We find out that, the desired BER is achievable with reconfiguration of the turbo code. Hence depending on the BER, power, speed and other requirements, parameters of the turbo code can be reconfigured dynamically.

C. *Ber For Different Frame Size*

Iteration 8 Rate ½

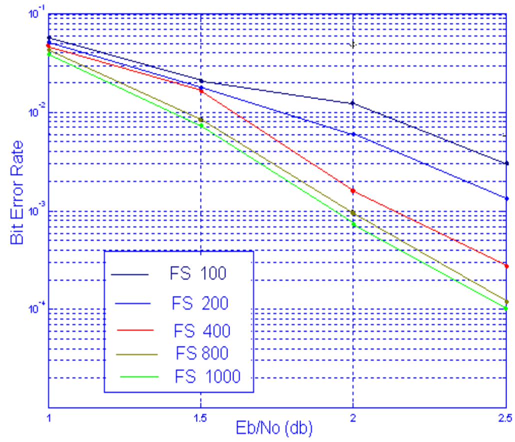

Fig. 7: BER with different frame size

D. *Effect Of Puncturing*

Iteration 8 Frame size 1024

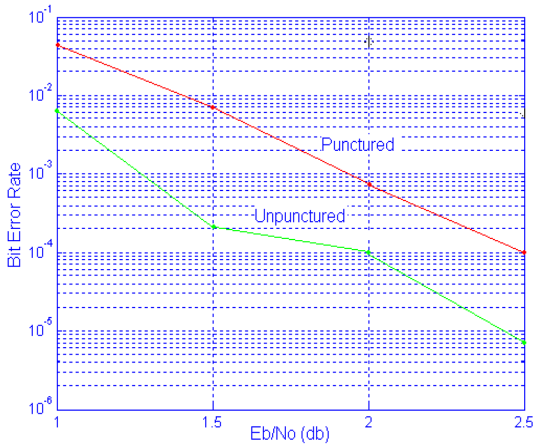

Fig. 8: Effect of puncturing on BER

Table-1: BER for different frame size and code rate of turbo code

Frame Size (bits)	Rate	Eb/No (dB)	BER
256	1/2	2.0	$8.4*10^{-03}$
1024	1/2	2.0	$7.4*10^{-04}$
4096	1/2	2.0	$1.8*10^{-05}$
16384	1/2	2.0	$4.6*10^{-06}$
256	1/3	1.5	$6.3*10^{-04}$
1024	1/3	1.5	$2.2*10^{-04}$
4096	1/3	1.5	$9.7*10^{-05}$
16384	1/3	1.5	$1.4*10^{-06}$

VII. ADVANTAGES OF FEC CODES OR TURBO CODES

As turbo codes are a very recent technique, the application fields are growing. Today, they are mainly used for spatial transmissions, like at the European Spatial Agency and at the NASA. There are also some considerations on it for the new mobile phone generation, called UMTS. Unlike QPSK, Turbo Code allows for two HDTV stream channels to be broadcast on a 27-MHz transponder. Furthermore, it increases the maximum number of standard TV channels from five to eight. If the system is used with the same data rate as QPSK with Viterbi-RS coding, the area size of its antenna may be reduced by 30%.

VIII. LIMITATION OF FORWARD ERROR CORRECTING CODES

Error correcting codes, sometimes referred to as forward error correcting (FEC) codes or channel codes, have become an invaluable tool in ‘closing’ the link budget of wireless-based digital communications systems. That is, by allowing a system to operate at a lower signal to noise ratio than would otherwise be the case, a desired quality of service over a link can be achieved within the transmit power or antenna gain constraints of the system. This property of error correcting codes is often referred to as ‘power efficiency’. The channel encoder adds code bits to the transmission bit stream, based on the data bits at its input. These extra bits are used by the channel decoder at the receiver to correct errors introduced into the transmission stream by a noisy or fading channel.

XI. THE DISADVANTAGES OF ERROR CORRECTING CODES ARE TWO-FOLD

Firstly, the injection of extra bits into the transmission stream has the effect that, if the original rate of transmission of ‘useful’ data bits is to remain the same, the symbol rate over the channel must be increased, thus increasing the bandwidth needed to transmit the signal. Frequently, bandwidth expansion is not an option in modern wireless systems, where frequency spectrum is often highly regulated and bandwidth is costly. If the bandwidth is strictly limited, then the effective data rate is decreased by the inclusion of an error correcting code. This problem is particularly troublesome at a time when new technologies are evolving, such as 3G, Digital Audio and Video Broadcasting, Wireless Local Loop and Bluetooth that propose high-speed data transmission over wireless channels.

A second disadvantage of error correcting codes is that they add complexity to the design of a communications system. The decoder complexity can affect the development time, physical size, power consumption, memory requirements and total delay of a receiver.

X. CONCLUSION

With performance results of the turbo code, we find that different BER can be achieved by reconfiguring the turbo code parameters. Hence dynamic turbo code reconfiguration based on the resource constraint in the communication network is an effective method. For example if there is a power constraint then the turbo code can be reconfigured with help of more iteration, frame size, unpuncturing and other parameters to achieve the same BER. If the information is delay sensitive then turbo code may be configured with reduced frame size, higher code rate and lower iterations. This way there are varieties of reconfigurations possible based on the available network infrastructure and Quos requirements. The result shows that turbo code can be reconfigured in terms of frame size and code rate based on the channel model and SNR to suit the BER and latency requirements for real time voice, real time video, playback of compressed video, data and file transfer applications. In similar way turbo code parameters can be configured to suit the different QoS classes i.e. conversation class, streaming class, interactive class and background class. This way we find that turbo codes are very helpful in future generation mobile communication from the reconfigurability as well as QoS requirement aspect.

REFERENCES

[1] Berrou, C., Glavieux, A. and Thitimajhima, P., “Near Shannon limit error correcting coding and decoding: turbo-codes,” *Proc. Of ICC ‘93*, Geneva, May 1993, pp. 1064-1070.

[2] Berrou, C. and Glavieux, A., “Near Optimum Error Correcting Coding and Decoding: Turbo-Codes,” *IEEE Trans. on Communications*, vol. 44, no.October 1996, pp. 1261-1271.

[3] Benedetto, Sergio and Montorsi., Guido “Design of parallel concatenated convolutional codes,. *IEEE Transactions on Communications*, vol. 44, No. 5, May 1996, pp. 591-600.

[4] Blackert, W. J., Hall, E. K., and Wilson, S. G., "Turbo code termination and interleaver conditions," *Electronic Letters*, vol.31, No.24, November 1995, pp. 2082-2084.

[5] Battail, Gerard, "A conceptual framework for understanding turbo codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, No. 2, February 1998, pp. 245-254.

[6] Dharmendra. K .Singh, "Turbo code study and simulation for mobile communication systems", MSc Thesis, University College Cork, September 2001.
[7] Dharmendra Kumar Singh, Liam Marnane, Patrick Fitzpatrick "Reconfigurability of Turbo Codes for Differentiated QoS" ISSC 2003, Limerick. July 1-2.
[8] C.E. Shannon, "A Mathematical Theory of Communications", Bell Systems Technical Journal, January 1948.
[9] TSG-RAN Working Group 1(Radio) meeting 4 Yokohama, Japan 19-20, April 1999 "Puncturing Algorithm for Turbo Code".
[10] Ali H. Mugaibel and Maan A.Kousa "Turbo Codes: Promises and Challenges"
[11] Benedetto, S., et al., "A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes," IEEE Communications Letters, vol. 1, no. 1, January 1997, pp. 22-24.
[12] S.Benedetto and G. Montorsi, "Tutorial 11: Turbo Codes: Comprehension, Performance Analysis, Design, Iterative Decoding", IEEE International Conference on Communications (ICC'97), Montreal, Quebec, Canada, June 8-12, 1997.
[13] M. Oberg, P.H. Siegel, "The Effect of Puncturing in Turbo Encoders", Proceedings of the International Symposium on Turbo Codes and Related Topics, pp. 204-207, Brest, France, 1997.
[14] S. Dolinru, D. Divsatar, "Weight Dkribution for Turbo Codes Using Random and Nonrandom Interleaver", TDA Progress Report 42-122, pp. 56 -65, Aug 1995.
[15] Cruz-Perez, F.A.; Hernandez-Valdez, G.; Ortigoza-Guerrero, L, "Performance evaluation of mobile Wireless communication systems with link adaptation", IEEE Communications Letters, Volume 7, Issue 12, Dec. 2003 Page(s): 587 – 589.

BIOGRAPHIES

Geetam Singh Tomar received his Ph.D. degrees in Electronics Engineering from RGPV, Bhopal. He has served Indian Air Force, Madhav Institute of Technology & Science Gwalior, Indian Institute of Information Technology & Management, Gwalior. His research interest areas are Wireless Networks, Sensor Networks and Embedded Systems. Presently he is with MIR Labs (USA), India part. He is also chief editor of three International Journals.

Ranjeet Singh Tomar is working as Associate Professor in Department of Electronics and Communication Engg. At Institute of Technology and Management, Gwalior, India. He received his M. Tech. Degree from MNIT, Jaipur, India. His research interest areas are Wireless Networks and Communication, Wireless Sensor Networks, Digital & Mobile Communication, Signal Processing and Vehicular Technology.