Design and Enhancement of Bandwidth and Performance of Rate-1/2 Puncturing Based Turbo Convolution Codes Using Simulink

Jitendra Prakash Shreemukh[1] Prof. (Dr.) B.S. Rai[2]
[1,2]Department of Electronics & Communication Engineering
[1,2]Madan Mohan Malaviya University of Technology, Gorakhpur, India

Abstract— It has been observed that particular rate-1/2 turbo code can achieve a lower error floor than that of their rate-1/3 parent codes. Nevertheless, good puncturing patterns can only be identified by means of an exhaustive search, whilst convergence towards low bit error probabilities can be problematic when the systematic output of a rate-1/2 turbo code is heavily punctured. In this paper, we present and study a family of rate-1/2 turbo code. We evaluate their bit error rate performance and we show that they always yield a lower error floor than that of their rate-1/3 parent codes. Furthermore, we compare analytic results to simulations and we demonstrate that their performance converges towards the error floor region, owning to the QPSK modulation technique as compared to the un-modulated one. Consequently, we propose punctured modulated turbo code as a means of improving the bandwidth efficiency and simultaneously lowering its error floor. The investigations are carried out for transmission on additive white Gaussian noise channels. For our simulation size of interleaver selected is 1024*1024 and keeping the number of iteration between 5to8. Simulations are conducted on MATLAB at maximum SNR range of 7 dB. In addition we also simulate the performance of punctured turbo codes with increasing the number of iteration and its effect on the performance is analyzed.

Keywords: Turbo codes, Code component, iterative decoding, Performance, Generator polynomials.

I. INTRODUCTION

In 1949 Claude Shannon published a classic paper [1] that established a mathematical basis for the consideration of the noisy communications channel. In his analysis he quantified the maximum theoretical capacity for a communications channel, the Shannon limit, and indicated that error-correcting channel codes must exist that allowed this maximum capacity to be achieved. The intervening years have seen many well-considered channel codes inch towards the Shannon limit, but all contenders have required large block lengths to perform close to the limit. The consequent complexity, cost, and signal latency of these codes have made them impractical within 3 to 5 dB of the limit, but they provide useful coding gain at higher values of E_b/N_0 and bit error rate. In 1993 Berrou, Glavieux and Thitimajshima [2] proposed a new class of convolution codes called turbo codes whose performance in terms of Bit Error Rate (BER) are close to the Shannon limit [3]. Channel coding is a powerful technique to get reliable communication over noisy channels. The performance of the coding system is bounded by Shannon limit. To provide a reliable communication over a noisy channel a coding technique Known as Turbo code is used. Turbo codes, has increased the interest in the coding area since these codes give most of the gain promised by the channel-coding theorem. Turbo codes are the most powerful error correction code which enables effective and efficient error correction in different field of communication. High secure data transmissions are carried out by encrypting the information in the original form. The major promise of turbo codes is their astonishing performance of bit error rate (BER) at relatively low E_b/N_0 [2]. To give an idea of how powerful turbo codes are, for a frame size of 256×256=65536 bits we can achieve a BER=10^{-5} over AWGN channel at only E_b/N_0=0.7dB, which is very close to Shannon limit [3]. For a Rayleigh fading channel a BER=10^{-5} can be achieved at E_b/N_0=4.3dB which represents a gain of 2.3dB as compared to classical convolution codes with similar complexity [4]. Turbo codes outperform all previous FEC scheme by at least 3dB, thereby doubling the battery lifetime or saving 20% of the required spectrum. Thus it gives a breakthrough in coding technology. In most of current wireless applications its properties make turbo-codes attractive [5]. Different puncturing patterns are suggested to increase the rate of Turbo code without increasing complexity from code rate from 1/2 to 1/3 or more and improve the performance of Turbo code [6-8]. The major challenge with turbo code is the inherent delay associated with the interleaver and the iterative decoding algorithm. Designing a channel code is always a tradeoff between energy efficiency and bandwidth efficiency. Codes with lower rate (i.e. bigger redundancy) can usually correct more errors. 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 more heavy on bandwidth consumption. Also, decoding complexity grows exponentially with code length, and long (low-rate) codes set high computational requirements to conventional decoders. For every combination of bandwidth (W), channel type, signal power (S) and received noise power (N), 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.

For additive white Gaussian noise channels, the formula is:

$$R < W \log_2 (1+S/N) \text{ [bits/second]}$$

In practical settings, there is of course no such thing as an ideal error-free channel. Instead, error-free data transmission is interpreted in a way that the bit error probability can be brought to an arbitrarily small constant. The bit error probability, or bit error rate (BER) used in benchmarking is often chosen to be 10^{-5} or 10^{-6}. Now, if the transmission rate, the bandwidth and the noise power are fixed, we get a lower bound on the amount of energy that must be expended to convey one bit of information. Hence, Shannon capacity sets a limit to the energy efficiency of a code. Although Shannon developed his theory already in the 1940s, several decades later the code designs were unable to come close to the theoretical bound. Even in the beginning of the 1990s, the gap between this theoretical bound and practical implementations was still at best about 3dB. This means that practical codes required about twice as much energy as the theoretical predicted. 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, 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. We have seen that the conventional codes left a 3dB gap between theory and practice. While for the turbo, the first 1/3 rate code proposed in 1993 made a huge improvement: the gap between Shannon's limit and implementation practice was only 0.7dB, giving a less than 1.2-fold overhead. Keeping these in mind, we are now ready to introduce the concept of turbo codes.

II. BINARY TURBO CODE: DEFINITION AND NOTATION

In a simplified turbo code, there are two convolution 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 1. The simplified turbo code block diagram in Figure1: shows only two branches. In general, one can have multiple turbo encoders with more than two branches. Length of parity bits are same as that of the information sequence and rate of turbo code is 1/3 (one input and three output sequence that is one information bits and two parity bit sequences from two RSC encoder). Turbo codes have three enhancements in the coding area. These are the random interleaved and two recursive systematic convolution (RSC) encoders of rate 1/2.

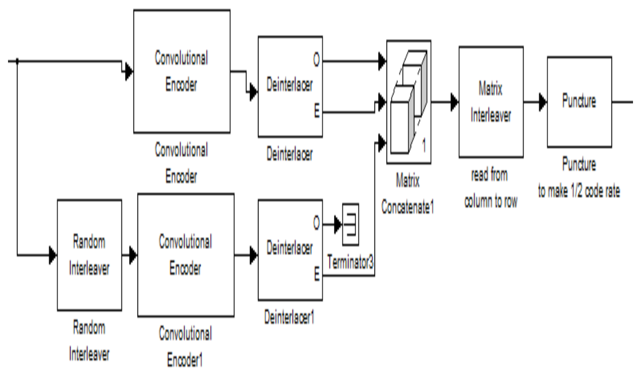

Fig. 1: Block Diagram of Turbo Code Encoder (Sources: Authors Compilation)

The choice of the interleaved is a crucial part in the turbo code design. The task of the interleaver is to "scramble" bits in a (pseudo-)random, albeit predetermined fashion. This serves two purposes. Firstly, if the input to the second encoder is interleaved, its output is usually quite different from the output of the first encoder. This means that even if one of the output code words has low weight, the other usually does not, and there is a smaller chance of producing an output with very low weight. Higher weight, as we saw above, is beneficial for the performance of the decoder [9, 10]. Secondly, since the code is a parallel concatenation of two codes, the divide-and-conquer strategy can be employed for decoding. If the input to the second decoder is scrambled, also its output will be different or "uncorrelated from the output of the first encoder. This means that the corresponding two decoders will gain more from information exchange. Some interleaved used in turbo codes are Row-Columns interleaved, S-type interleaved, Pseudo-random interleaved, Helical interleaved, Uniform interleaved etc. There is no such thing as a universally best interleaved. For short block sizes, the odd-even interleaved has been found to outperform the pseudo-random interleaved, and vice versa. The choice of the interleaved has a key part in the success of the code and the best choice is dependent on the code design. Because it performs a permutation, an interleaved is commonly denoted by the Greek letter π, and its corresponding de-interleaved by π^{-1}.

III. PRELIMINARIES: PUNCTURING

Puncturing is elimination of some bits of a codeword before of sending out it and replacing zero instead of these bits before of decoding. Puncturing is an effective technique to increase the data rate [11]. Some of puncturing patterns are unsuitable and degrade the performance of turbo code. Unsuitable patterns are those patterns which make impossible convergence of iterative decoding and recovering of message because of inordinate elimination of "1" bits of turbo coded sequence but it is verifiable that the probability of error in the received sequence can be small arbitrarily [6]. The value of the punctured bits is set to binary 0 for the simulation purposes. The bits are not transmitted and the code rate decreases accordingly. This is not a classical puncturing of whole lines or rows in a product code.

IV. SIMULATIONS

In this section MATLAB simulations are shown. The performance of turbo codes is evaluated in terms of BER. The main tool for the performance evaluation of punctured turbo codes is computer simulation. Computer simulation generates reliable probability of error estimates as low as 10^{-6} given by Shannon capacity limit. Computer simulation is useful for rather low SNRs since the error probabilities for larger SNRs are difficult to simulate. All simulations were taken with MATLAB. The value of puncturing is set to be zero for simulation purpose. The number of iterations between the decoders is varying between 5 to 8 iteration. In our simulation, the input data frame size is 1024 bits, i.e. the size of punctured turbo code interleaved. Channel model used in simulation is AWGN channel. The decoder implementation is complex and computationally extensive. It includes processing using a number of loops. The decoder decodes iteratively checking the number of errors after every iteration. If the number of errors is zero for iteration, the code will not execute the next iteration to decrease processing load. The maximum range of SNRs for simulation in dB is 7 dB and code rate of parent turbo codes is 1/3. The modulation scheme used for simulation purpose is QPSK technique. In our simulation analysis, the signal power is kept constant.

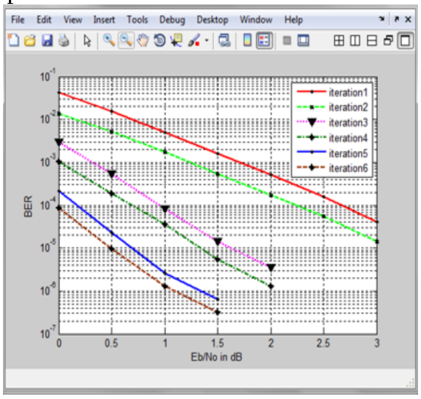

Fig. 2(a): BER for frame size 1024 un-punctured turbo code over AWGN channel

Figure 2(a) shows the BER versus signal-noise ratio plots for interleaver size 1024*1024, R= 1/3 and number of iteration=6 for the turbo code when puncturing phenomenon is not utilized. Figure 2(b) shows a similar graph but was simulated when puncturing is used at the output of RSC encoder with 8 iteration stages. In Figure 2(c) and Figure 2(d) the performance of multiple run punctured turbo code and turbo code with modulation is compared for frame size of 1024 bits and iteration size 6 and 5 respectively. Turbo code performance is compared by looking towards the graphs obtained by simulation results.

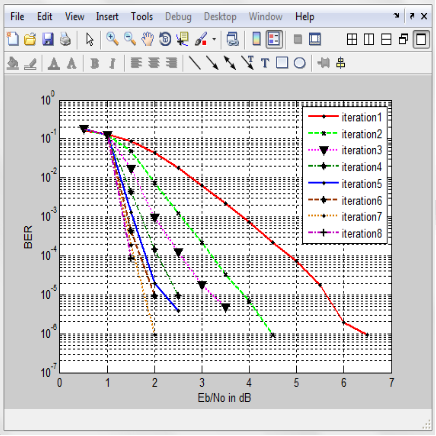

Fig. 2(b): Performance curve for punctured turbo code

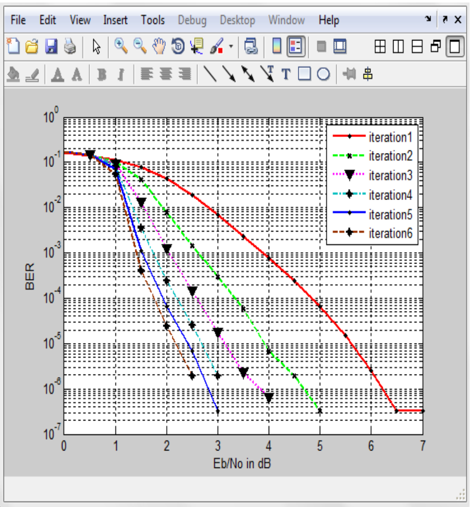

Fig. 2(c): Performance curve for multiple run punctured turbo code

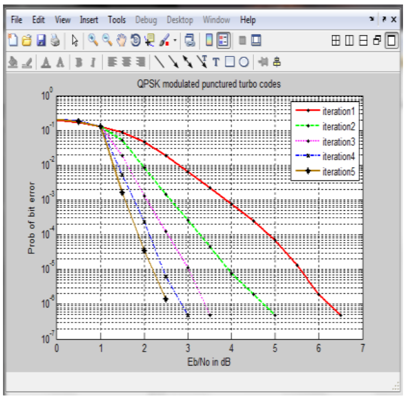

Fig. 2(d): Performance curve for QPSK modulated punctured turbo code

In figure 2(a) excellent performance in terms of BER is achieved since all the parity bits are send out and none of the parity bits are deleted, but bandwidth efficiency is poorer. In our simulation, our concern is to obtain the code which is better in terms of performance as well as bandwidth also. Figure 2(b) turbo code performance of the bit error rate order of 10^{-6} achieved for iteration 1, 2 and 7. For iteration 7 punctured turbo codes gives better BER performance at lower SNR value of 2 dB. Figure 2(c) shows multiple run turbo code (punctured) performance and achieved approximately close to the 10^{-7} order BER performance for some iteration number. Figure 2(c) gives slightly better result as compared to the figure 2(b). Figure 2(d) is simulated result for turbo code along with QPSK modulation scheme and hence due to modulation scheme performance is reduced and BER performance is of order greater to the 10^{-6} is achieved at maximum SNR range of 7 dB, which is closed to the multiple run turbo codes in term of performance criteria but along with modulation scheme bandwidth is also much improved as compared to the un-modulated turbo codes. Hence we can conclude that modulated turbo codes gives comparable performance with reference to un-modulated one and excellent improvement in bandwidth is achieved due to modulation.

V. CONCLUSION

In this paper performance of punctured turbo code has been compared in terms of BER under different condition such as un-punctured, punctured turbo code, multiple run turbo code and modulated turbo code. Bandwidth is reduced as modulation scheme is utilized. The code after modulation gives better performance at moderate SNR range of upto 7 dB. In this section, both punctured and un-punctured turbo code performance is compared, although un-punctured code gives better performance but occupy more bandwidth while punctured turbo code give moderate performance with less bandwidth consumption. Finally on the basis of simulation result, we can conclude that Modulation scheme is going to be an efficient method to improve the bandwidth of turbo code. Modulated Turbo code performance of order approximately 10^{-6} can be easily achieved for some iteration which is approximately much closed to the un-modulated turbo codes but advantage with modulated turbo code is bandwidth is improved. In our simulation we also analyzed that as the number of iteration increases BER performance improves but rate of improvement decreases. Thus, the number of iterations should be kept such as to avoid extra computations. Thus, upper bounds need to be specified for the number of iterations. We believe that our simulation work will be useful for further research in field of turbo coding for further improvement in performance.

REFERENCES

[1] Shannon CE. "Communication in the presence of Noise". Proceedings IRE, Jan 1949, reprinted Proceedings IEEE, Vol 86, No 2 Feb 98.

[2] 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.

[3] Claude Berror and Alain Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," IEEE Transactions on Communications, Vol.44, No.10, 1996, pp.1261-1271.

[4] Hagenauer, J. and Papke, L., "Decoding turbo codes with the soft-output Viterbi algorithm (SOVA)," in Proceedings of International Symposium On Information Theory(Trondheim, Norway, June 1994),p. 164.

[5] Branka, Jinhong, Turbo Codes Principles and Applications, Kluwer, 2000.

[6] M.A. Kousa and A.H. Mugaibel, "Puncturing effects on Turbo Codes," IEE Proceedings on Communications, Vol.149, 2002, pp.132-138

[7] Joachim Hagenauer, "Rate compatible punctured convolutional codes (RCPC Codes) and their Application," IEEE Transactions on Communications, Vol. 36, 1988, pp.389-400

[8] F. Baich, G. Montorsi, and F. Vatta, "Some notes on rate-compatible punctured turbo codes (RCPTC) design," IEEE Transactions on Communications, Vol. 52, 2004, pp.681-684

[9] Y. Kim, J. Cho, W. Oh and K. Cheun, "Improving the performance of turbo codes by repetition and puncturing," Project Report, Division of Electrical and Computer Engineering, Pohang University of Science and Technology.

[10] D. Divsalar, S. Dolinar, and F. Pollara, "Transfer Function Bounds on the Performance of Turbo Codes," TDA Progress Report 42-122, Aug. 1995, Communications Systems and Research Section, R. J. McEliece California Institute of Technology, pp. 44-55

[11] BERROU, C., and GLAVIEUX, A,: 'Near optimum-error correcting coding and decoding: turbo-codes', IEEE Trms. Cornrnun., 1996, 44,pp. 1261-1271

[12] Narushan Pillay and HongJun Xu, " Dual-Repeated-Punctured Turbo Codes on AWGN channels" in IEEE AFRICON 2009.