Journal of Engineering Science and Technology
Vol. 15, No. 2 (2020) 820 - 830

DESIGN AND ANALYSIS OF TURBO CODE FOR WIRELESS COMMUNICATION NETWORKS

NADA I. NAJIM*, M. F. GHANIM

Computer Engineering Department,
College of Engineering, University of Mosul, Mosul, Iraq
*Corresponding Author: nadaalmaaroof2017@gmail.com

Abstract

Nowadays, one of the most important forward error correction schemes is turbo code. This code is widely used because of its good performance and its simple components. This work includes the whole design and simulation of Turbo code encoder and decoder using LABVIEW. The implementation had been done through implementing four types of the Maximum Aposteriori Probability (MAP) algorithm. The results compare different scenarios of the system with data sent over Rayleigh flat fading channel in order to show the performance of the encoder and decoder clearly.

Keywords: Forward error correction, MAP, Turbo code, Wireless communications.

1. Introduction

Nowadays, wireless networks are spreading more and more. The majority of installed networks have become wireless due to the simplicity of installation, as they do not need an infrastructure. This does not mean that the role of the wired networks is being eliminated. Instead, the wireless networks are considered as a complementary of the wired networks. With all types of networks: personal and local area networks (PAN and LAN) and wide area networks (WAN), the length of data packet, designing cost and intensity constrain the system ability to work appropriately at multiple standards [1, 2].

The plan of turbo code is to bring the work as closer as possible to the Shannon limit. Turbo codes are generally utilized as a part of most recent wireless standards for example Worldwide Interoperability for Microwave Access (WiMAX) and Long-Term Evolution (LTE) and International Mobile Telecommunications-Advanced IMT-Advanced [3]. The fundamental turbo code consists of encoder and decoder. The turbo encoder architecture comprises of at least two segment encoders connected in parallel and an interleaver placed in between to isolate them. The turbo decoders process the bits of information utilizing soft decision. The decoding is hence done iteratively to promote the reliability [4, 5]. The significance of turbo codes has been discussed in many papers. Most researches on Turbo Code have focused on showing the performance of this code with different parameters; such as number of iteration, type of the technique used in decoding, length of code, rate, generator polynomial and type of channel in calculating the Bit Error Rate (BER) for each case [6, 7]. Shrivastava et al. [8], Imran [9] and Mohammed and Obaid [10] try to identify the parameters that bring about the best possible performance. Other researchers such as La Rosa et al. [11] and Zhang et al. [12] have studied the implementation of turbo-decoder on platform to reduce the computational complexity and increase the parallelization.

Turbo codes are error-correcting codes that are familiar and used in current generation of wireless communications [2]. Two identical encoders usually used in the design of a turbo encoder. These encoders are connected to each other in parallel concatenation to allow simpler decoding, an interleaver is installed between the two encoders to make input bits to the second encoder look different input for the first, and this breaks the relation of the output of one encoder to the other. The input bits are also contained in the output and this is because of the nature of the encoder's system [13, 14].

A similar way is used to build the decoder. However, here the two decoders are connected in serial. Serially concatenated codes give better performance when compared to the parallel-concatenated codes. The error bursts that are coming from the output of the first decoder are then scattered at the interleaver, which is fitted just between the decoders [15, 16].

2. Design of Turbo Encoder

Turbo encoder closely follows the design structure mentioned before, i.e., the turbo coder scheme is a Parallel Concatenated Convolutional Code (PCCC) with two constraint length 4 RSC (Recursive Systematic Convolutional) encoders. The coding rate is approximately $r = 1/3$. K represents the data bits length of the turbo encoder input, where $40 \leq K \geq 5114$. The structure of turbo encoder is illustrated in Fig. 1 [17, 18].

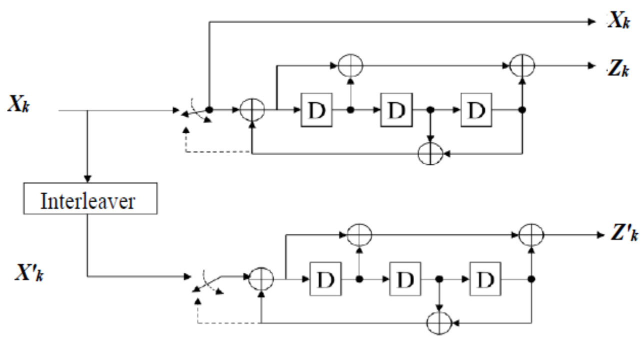

Fig. 1. Structure of turbo encoder [17].

The transfer function of length K = 4 (8-state) constituent code for PCCC is shown in Eq. (1), and the two convolutional encoders are identical with feedback and feed forward generator polynomials shown in Eqs. (2) and (3) [13].

$$G(D) = \left[1, \frac{g_{1(D)}}{g_0(D)}\right] \tag{1}$$

$$g_0(D) = 1 + D^2 + D^3 \tag{2}$$

$$g_1(D) = 1 + D + D^3 \tag{3}$$

Prior to encoding, the initial value of each shift register of the 8-state constituent encoders shall be all zeros. The turbo encoder output is (X_1, Z_1, Z'_1, X_2, Z_2, Z'_2,…, X_K, Z_K, Z'_K), where X_1 to X_K are the bits input to the both first (upper) 8- state constituent encoder and turbo encode of the internal interleaver. K is the number of bits, while Z_1 to Z_K and Z'_1 to Z'_K are the bits output from the two constituent convolutional encoders. In other words, first and second 8-state constituent encoders, respectively. The bits output from Turbo code internal interleaver are denoted by X'_1 to X'_K, which are the input of the second (lower) constituent encoder.

3. Design of Turbo Decoder

The architecture of turbo decoder is clarified in Fig. 2. Based on studies by Iliev [15] and Valenti and Sun [17], it operates in an iterative manner. The Log Likelihood Ratio (LLR) is computed through the decoder. The decoder uses what it had received of code words, together with the knowledge about the structure of the code, in the calculation of LLRs. The complexity of the process necessitate the use of more than one probabilistic processor.

This is because the interleaver is present at the encoder side and this complicates the process and makes it impractical to be done by single processer. Hence, the global LLR is achieved after the breakdown of the job between two RSC decoders. The decoders use soft-input-soft-output (SISO) and the data is exchanged between them, however, the data has to pass through the interleaver before being processed by the next decoder [2, 15].

4. Simulation of Turbo Code using LABVIEW Friction Drag Coefficient

In this work, LABVIEW is used for turbo code simulations. The proposed systems contain both encoder and decoder. The used channel for this simulation is Rayleigh flat fading channel while the packets of data have length of 40 to 5114 bits. For decoder, the used iterations are up to 14 iterations with encoding rates equal to 1/2 and 1/3.

On the other hand, two Recursive Systematic Encoders installed in parallel and separated by a pseudo-random inter-leaver. In fact, binary phase shift key (BPSK) is used to modulate the encoded data packet. Figures 3and 4 show the simulation blocks of turbo encoder and decoder respectively.

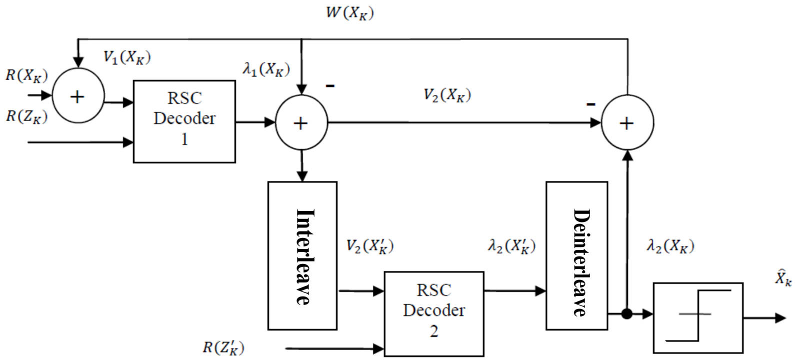

Fig. 2. Turbo decoder architecture [17].

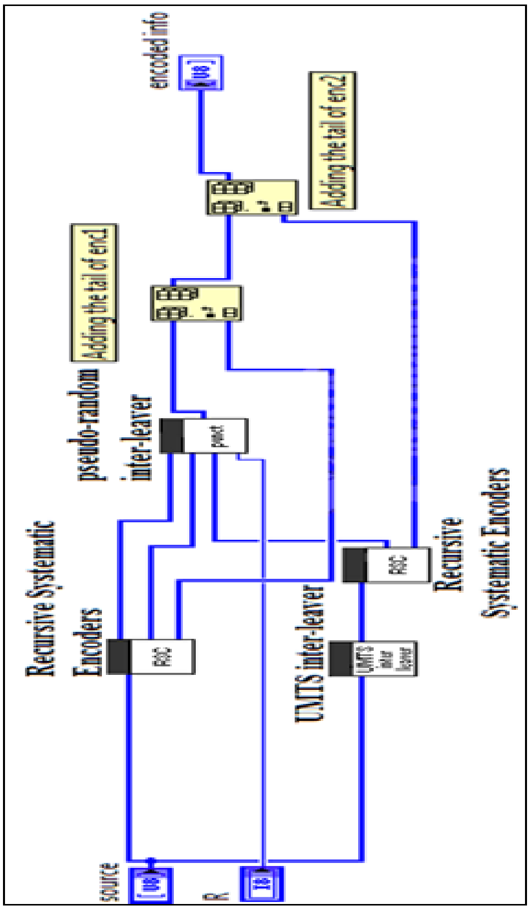

Fig. 3. LABVIEW implementation for turbo encoder.

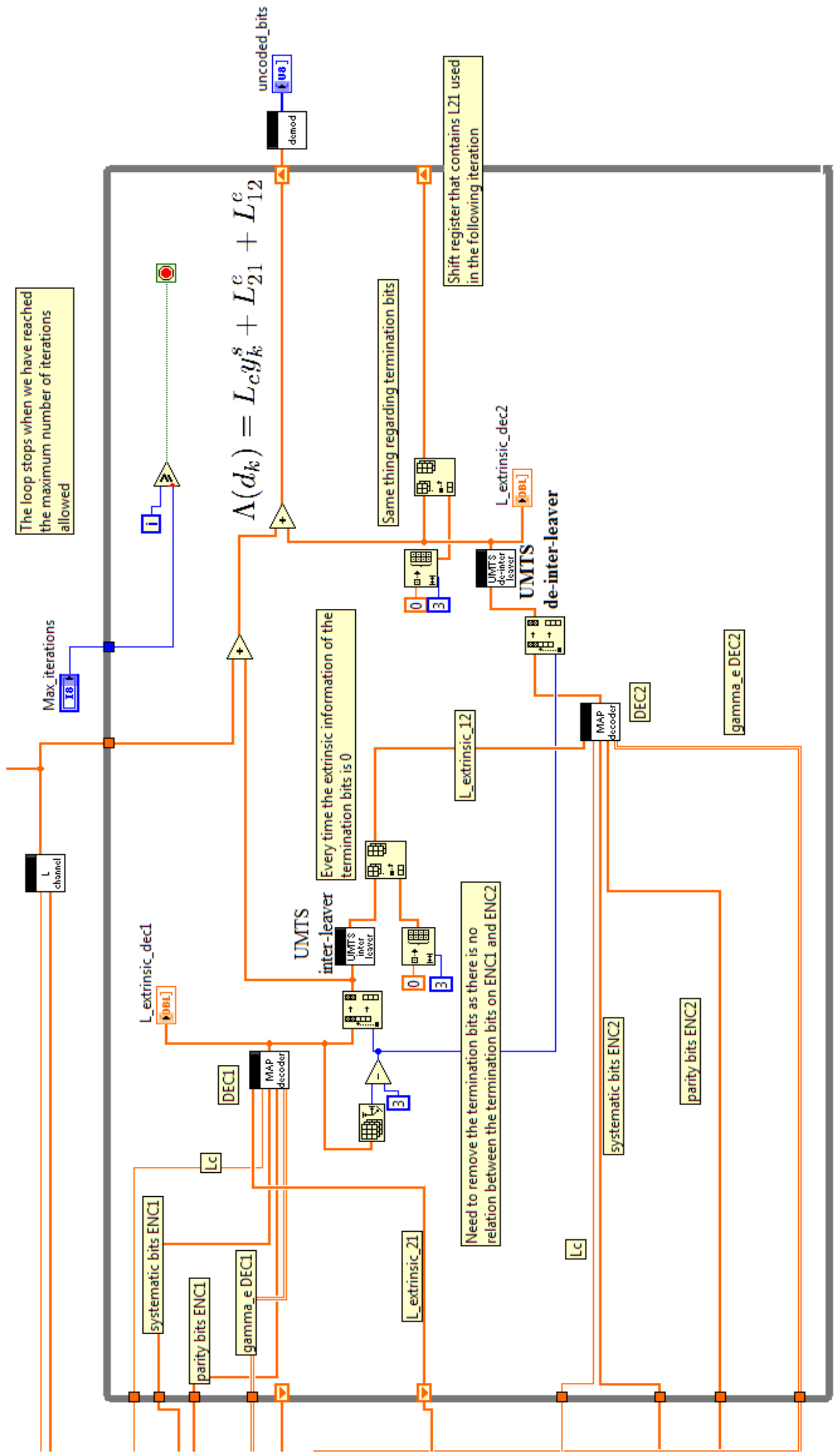

Fig. 4. LABVIEW implementation for turbo decoder.

Implementation of the iterative decoder as defined by Ryan [19], where the logarithm of likelihood ratio is calculated as:

$$\Lambda(d_k) = y_k^s + L_{21}^e + L_{12}^e \quad (4)$$

where the different elements are:

- L_{12}^e and L_{21}^e are the extrinsic information values calculated by the first and second decoders.
- y_k^s represents the systematic bits calculated by the first encoder.

5. Results and Discussion

Turbo codes can reduce a big portion of the error rate during the reduction of the Signal to Noise Ratios (SNRs), in case the volume of the data packet length K is big, and the exchange process is random.

The turbo codes are iterative codes, and it is been observed that the higher number of iterations is shown to produce the better performance. Figure 5 shows the BER performance for the iterations range from (0 to 14) at data packet length $K = 40$.

In this figure, BER was found equal to 0.005 after first decoder iteration and the value of BER improved to 0.000142 when number of iteration is increased to five decoder iterations.

However, when the iteration number is very high, there is no big improvement in the performance. Therefore, and in order to achieve the ideal code performance, the highest limits of iterations and SNR need to be determined. In Fig. 6, the data packet length $K = 530$ followed as similar behaviour as of Fig. 5. However, BER value of around 10^{-4} achieved at SNR 3.5 dB at data packet length $K = 530$ after first iterations.

On the other hand, the same BER can be gained at higher SNR (SNR 5dB) but with data packet length $K = 40$. In other words, the same BER can be obtained, when the data packet length K is increased but at lower SNR. It is also true that the BER performance enhanced at higher data packet length K as shown in Fig. 7.

Finally, Figs. 5 to 7 show that the higher number of iterations produced the better performance. However, this is only true at certain interval, because the BER value reached a steady state and decreased no further after a specified number of iterations.

Thus, to achieve a lower BER there should be a change in either data packet length K or the SNR. Increasing the SNR means the power of the signal increases. Therefore, a lower BER can be obtained. Thus, an improved decoding of data can be achieved as the a priori information of the data available improves. This is also can be achieved if the data packet length K increases.

BER against different encoding rates and different number of decoding iterations are also simulated in this paper. Figure 8 at data packet $k = 530$ illustrated the number of iterations function on the evolution of BER. It is also obvious that the BER values are much higher at encoder rates $R = ½$ than encoder rates $R = ⅓$.

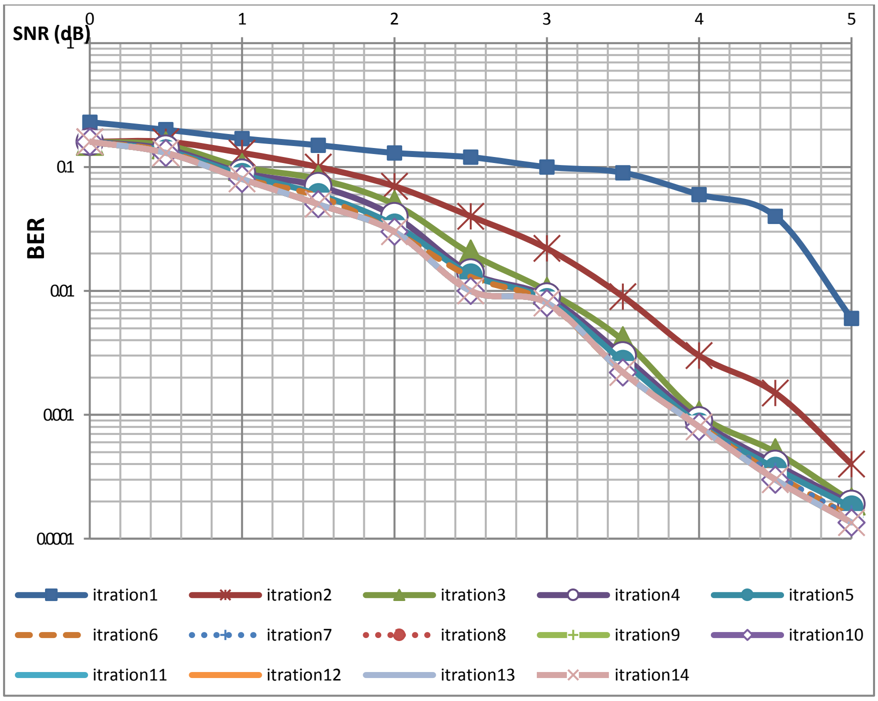

Fig. 5. BER of turbo decoder over Rayleigh channel for data packet $K = 40$.

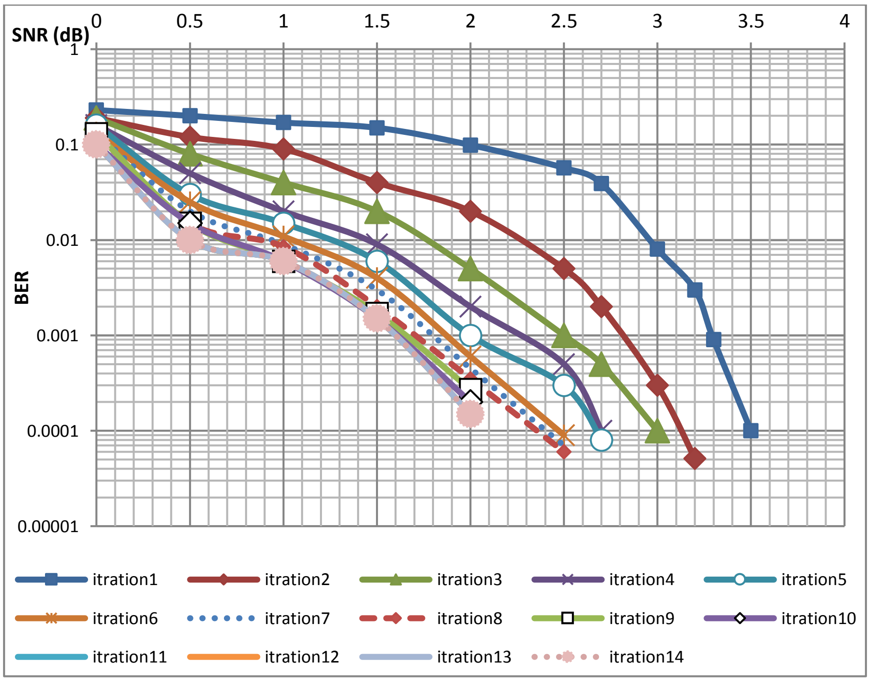

Fig. 6. BER of turbo decoder over Rayleigh channel for data packet $K = 530$.

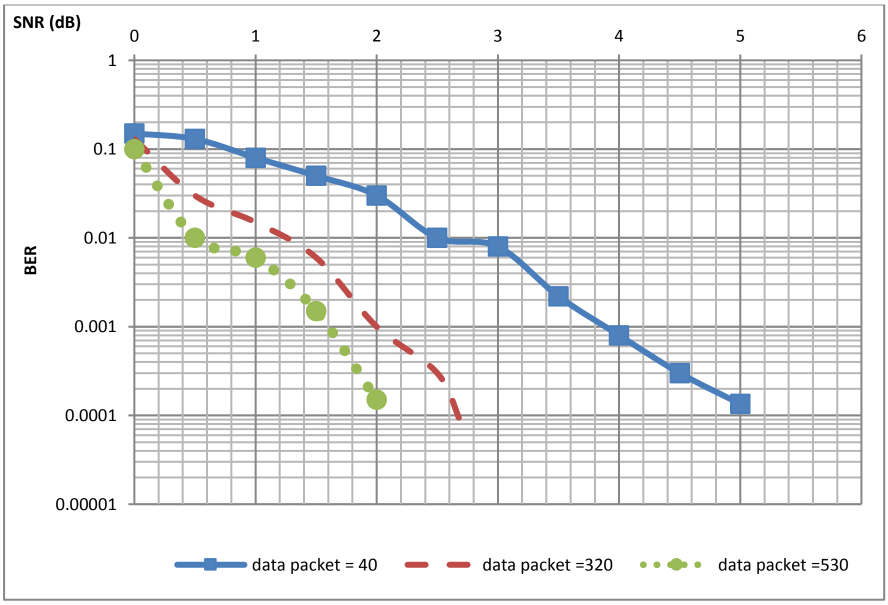

Fig. 7. BER of turbo code after 14 decoder iterations.

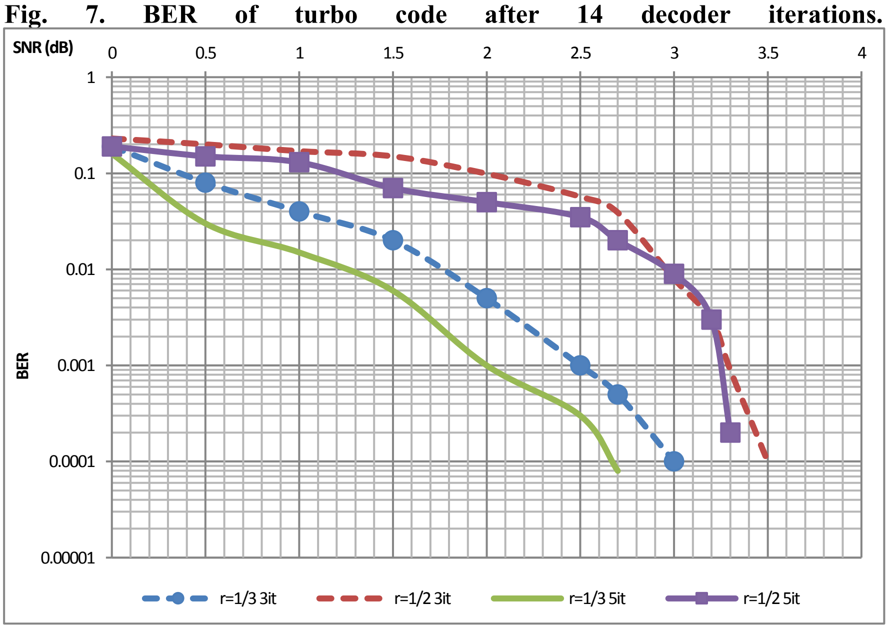

Fig. 8. BER performance for data packet *K* =530 at different encoding rates and different number of decoding iterations.

BER performance over Rayleigh channel for Maximum Aposteriori Probability (MAP) algorithm is done using four types of MAP algorithm at data packet length *K* =530 then a plot is produced for the BER against 14 iterations as a function of SNR. As shown in Fig. 9, a log MAP algorithm BER of approximately 10^{-4} was recorded at SNR 2dB after 14 decoder iterations. In addition, the max-log MAP BER was the highest in contrast to the log MAP one, which was at the lowest. In conclusion, the significantly worst performance was seen with the max-log MAP algorithm, while the log MAP algorithm performed the best. However, when complexity is involved, the max-log MAP has the least complexity and it only consumes little time in simulation. Therefore, there was a play on the complexity at the expense of performance. The log MAP algorithm, on the other hand, is more complex, yet it is still able to produce similar BER performance, compared to max-log MAP algorithm, using less iteration. In conclusion, complexity and error performance are at a trade-off during the designing of turbo codes.

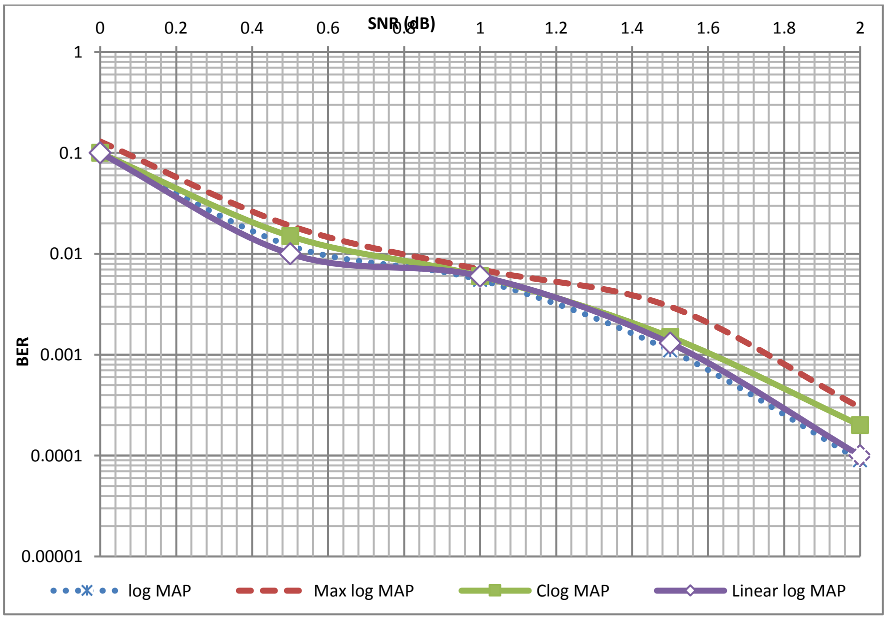

Fig. 9. BER of turbo code for data packet *K* = 530 and 4 types of MAP algorithm after 14 decoder iterations over a Rayleigh fading channel.

6. Conclusions

The turbo codes are iterative, and it has been observed that the higher number of iterations is shown to produce the better performance; the optimum improvement can be achieved at 5 to 14 iterations. During the simulation, there was a significant drop in the BER as the iteration increased in the first five iterations. In addition, there was no significant improvement from the fifth to the fourteenth iteration. The only conclusion of this is that the rate of performance improvement decreases with the increase in the number of iterations. Therefore, to keep the BER low, there must be an alteration in data packet length *K* or the SNR. The randomization the interleaver introduced is tremendously affecting the performance of the turbo code. The higher weight code words is highly probable if the interleaver size *K* increased; in other words, the decoder will perform better. During the study, the four types of the Maximum Aposteriori Probability (MAP) algorithm were included. In conclusion, the log MAP algorithm, while it is the most complex, performs the best, in contrast to the max-log MAP, the least complex, which performs the worst.

References

1. Anwar, M.I.; Virtanen, S.; and Isoaho, J. (2008). A software defined approach for common baseband processing. *Journal of Systems Architecture*, 54(8), 769-786.
2. Abdulwahab, W.K.; and Kadhim, A.A. (2018). Comparative study of channel coding schemes for 5G. *Proceedings of the International Conference on Advanced Science and Engineering (ICOASE)*. Duhok, Iraq, 239-243.
3. Gamage, H.; Rajatheva, N.; and Latva-Aho, M. (2017). Channel coding for enhanced mobile broadband communication in 5G systems. *Proceedings of the European Conference on Communications (EuCNC)*. Oulu, Finland, 1-6.
4. Berrou, C.; Glavieux, A.; and Thitimajshima, P. (2003). Near Shannon limit error correcting coding and decoding: Turbo codes. *Proceedings of IEEE International Conference on Communications (ICC)*. Geneva, Switzerland, 1064-1070.
5. Anghel, C.; Stanciu, C.; and Paleologu, C. (2017). Efficient FPGA implementation of a CTC turbo decoder for WiMAX/LTE mobile systems. *IntechOpen, Field - Programmable Gate Array*, 2, 25-55.
6. Yang, L.; Xie, Y.; Wu, X.; Yuan, J.; Cheng, X.; and Wan, L. (2018). Information-coupled turbo codes for LTE systems. *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. Barcelona, Spain, 13 pages.
7. Chaware, T.; and Pathak, N. (2013). Performance of turbo product code in WiMAX. *International Journal of Advanced Computer Science and Applications (IJACSA)*. Special issue on International Conference and Workshop on Advance Computing, 43-47.
8. Shrivastava, B.; Raut, Y.; and Mishra, R.S. (2011). Performance of Turbo Code for UMTS in AWGN channel. *International Journal of Advanced Computer Research*, 1(1), 46-51.
9. Imran, A. (2013). *Software implementation and performance of UMTS Turbo decoder.* Master Thesis. Faculty of Computing and Electrical Engineering, Tampere University of Technology, Tampere, Finland.
10. Mohammed, S.J.; and Obaid, A.A. (2017). Design and implementation different types of turbo decoder with various parameters. *International Journal of Computer Applications*, 166(11), 44-48.
11. La Rosa, A.; Lavagno, L.; and Passerone, C. (2005). Implementation of a UMTS turbo decoder on a dynamically reconfigurable platform. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(1), 100-106.
12. Zhang, C.; Hu, Z.; and Chen, J. (2014). A parallelized implementation of turbo decoding based on network on chip multi-core processor. *Journal of Engineering Science and Technology Review*, 7(1), 52-59.
13. Liberg, O.; Sundberg, M; Wang, Y.-P.E.; Bergman, J.; and Sachs, J. (2017). *Cellular internet of things: Technologies, standards and performance*. London, United Kingdom: Academic Press.
14. Zhu, M.; Mitchell, D.G.M.; Lentmaier, M.; Costello, D.J.; and Bai, B. (2017). Braided convolutional codes with sliding window decoding. *IEEE Transactions on Communications*, 65(9), 3645-3658.

15. Iliev, T. (2007). A study of turbo codes for UMTS third generation cellular standard. *Proceedings of the International Conference on Computer Systems and Technologies*. Rousse, Bulgaria, 1-6.
16. Iscan, O.; Lentner, D; and Xu, W. (2016). A comparison of channel coding schemes for 5G short message transmission. *Proceedings of the IEEE Globecom Workshops.* Washington, D.C., United States of America, 1-6.
17. Valenti, M.C.; and Sun J. (2001). The UMTS turbo code and an efficient decoder implementation suitable for software-defined radios. *International Journal of Wireless Information Networks*, 8(4), 203-215.
18. Gamage, H.G. (2017). *Waveforms and channel coding for 5G.* Master Thesis. Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland.
19. Ryan, W.E. (1998). A turbo code tutorial. *Proceedings of IEEE Globecom*. Sydney, New South Wales, Australia, 1-9.