Research Article

Performance Comparison of Turbo coder and low-density parity check codes

Mrs.Channaveeramma.E[1], K.M.Palaniswamy[2]

[1] Assistant Professor of ECE Department, Navodaya Institute of Technology, Raichur, Karnataka, India.
[2]Professor of ECE Department, Dr. T. Thimmaiah Institute of Technology, Oorgaum, Kolar Gold fields, Karnataka, India.

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract
This paper investigates the two powerful forward error correction techniques, Turbo codes and LDPC codes. The different code parameters such as code rate, decoding iterations, and block length are considered under AWGN channel. The strengths and performance hindrance facts of both the coding techniques been summarized.

Keywords: Turbo codes, LDPC, AWGN, FEC.

1. INTRODUCTION

Wireless communication systems will suffer from the noise introduced in the channels. Channel codes are the essential part of wireless communication systems which help in detection and correction of errors due to the noise introduced in the channel. Turbo codes and LDPC codes are the Forward Error Correction (FEC) channel coding techniques which have the error correcting capability near to Shannon codes along with improvement in transmission rate and energy efficiency.Turbo codes were introduced in 1993[1]. LDPC codes were discovered in 1960 by R.Galleger in his Ph.D dissertation at MIT.They became implementable, after the discovery of Turbo codes[2]. The satellite communications such as DVB-RCS, telecommunications such as 3G, 4G, Wireless metropolitan standards IEEE 802.16(WiMax) uses turbo codes[3]. G.hn/G.9960 (ITU-T standard for networking over power lines, phone lines and coaxial cable), 802.3 an(10GBps ethernet over twisted pair), CMMB(China multimedia mobile broadcasting), DVB-S2/DVB-T2/DVB-C2(Digital video broadcasting , second generation), DMB-T/H(Digital video broadcasting), Wimax(IEEE 802.16e standard for microwave communications), 802.11n-2009(wi-Fi standard) are the few standards where the LDPC codes are employed.[4]

2. TURBO CODES

2.1 Turbo encoding techniques:

To be more specific, concatenated codes can be of two types: parallel or serial. The type of concatenated codes used by Berrou et al. was of the parallel type which is shown in Figure.1

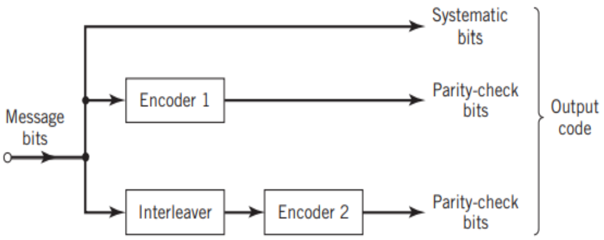

Figure.1: Basic structure of the Turbo encoder.

Figure 1 depicts the most basic form of a turbo code generator that consists of two constituent Recursive systematic encoders, which are parallelly concatenated by an interleaver. The interleaver is an input–output mapping device that, takes the symbols at the input and produces identical symbols at the output but in a different temporal arrangement.Turbo codes use a pseudo-random interleaver, which operates only on the systematic message bits. The dimensions of the interleaver used in turbo codes is often very large, on the order of several thousand bits. The two reasons of using an interleaver in a turbo code are:

- The interleaver ties together errors that are easily made in one half the turbo code to errors that are exceptionally unlikely to occur within the other half; this can be indeed one reason why the turbo code performs better than a conventional code.

- The interleaver provides robust performance with respect to mismatched decoding, a issue that arises when the channel statistics aren't known or are incorrectly specified. Ordinarily, but not necessarily, the same code is used for both constituent encoders in Figure 1.

In Figure 1, the input data stream is applied directly to encoder 1 and the pseudorandomly reordered version of the same data stream is applied to encoder 2. The systematic bits (i.e., original message bits) and the two sets of parity-check bits generated by the two encoders constitute the output of the turbo encoder. This makes it a rate 1/3 turbo code. Although the constituent codes are convolutional, in reality, turbo codes are block codes with the block size being determined by the periodic size of the interleaver. Moreover, both RSC encoders in Figure 1 are linear. We may therefore describe turbo codes generally as linear block codes. To know the beginning and the end of a codeword, initialize the encoder to the all-zero state and then encode the data. After encoding a certain number of data bits, a number of tail bits are added so as to make the encoder return to the all-zero state at the end of each block; thereafter, the cycle is repeated. The termination approaches of turbo codes are, to terminate the first RSC code in the encoder and leave the second one undetermined, leading to increased noise in second RSC code and steep drop in error performance at low SNR. The dominant factor affecting the error floor is interleaver. A more refined approach is to terminate both constituent codes in the encoder in a symmetric manner which reduces the error floor by an order of magnitude as compared to the previous approach. The puncturing of the parity-check-bits generated by two encoders were done by deleting certain parity-check-bits, hence increasing data rate; the message bits were kept unaffected during puncturing. Basically, puncturing is the inverse of extending a code. The net result of parallel concatenation is a turbo code that appears essentially random to the channel by virtue of the pseudo-random interleaver, yet it possesses sufficient structure for the decoding to be physically realizable. Coding theory asserts that a code chosen at random is capable of approaching Shannon's channel capacity, provided that the block size is sufficiently large. The constituent codes recommended for turbo codes are short constraint length RSC codes. The recursive codes make the internal state of the shift register depend on past outputs. This affects the behaviour of the error patterns, with the result that a better performance of the overall coding strategy is attained. The LTE rate 1/3 encoder is shown in Figure.2.

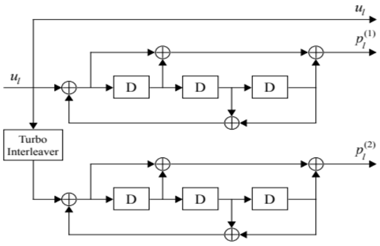

Figure.2: LTE rate 1/3 encoder

2.2 Turbo decoding techniques:

Iterative decoder Structure:

Let us commence our discourse by considering the overall structure of the iterative turbo decoder shown in Fig. 1. Two component decoders are linked by interleavers in an exceedingly structure just like that of the encoder. As seen within the figure, each decoder takes three inputs: 1) the systematically encoded channel output bits; 2) the parity bits.

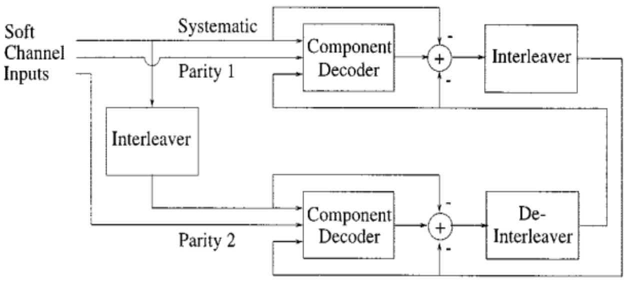

Fig.3. Turbo Decoder Structure

This cycle is repeated, and with every iteration the BER of the decoded bits tends to fall. However, the development in performance obtained with increasing numbers of iterations decreases because the number of iterations increases. Hence, for complexity reasons, usually only about eight iterations are used. Such an iterative decoder employs two component soft-in soft-out decoders, and that we have described the MAP, Log-MAP, Max-Log-MAP and SOVA algorithms, which may all be used as component decoders. The MAP algorithm is perfect for this task, but it's extremely complex. The Log-MAP algorithm could be a simplification of the MAP algorithm, and offers the identical optimal performance with an inexpensive complexity. the opposite two algorithms, the Max-Log-MAP and also the SOVA, are both less complex again, but provides a slightly degraded performance. so as to measure the expected coding performance we also provided a spread of performance results employing a sort of codec parameters.

3. LDPC CODES

The two most vital important advantages of LDPC codes over turbo codes are:
- absence of low-weight codewords and
- iterative decoding of lower complexity.

With reference to the problem of low-weight codewords, we usually find that atiny low number of codewords in an exceedingly turbo codeword are undesirably near the given codeword. because of this closeness in weights, once during a while the channel noise causes the transmitted codeword to be mistaken for a close-by code. In contrast, LDPC codes may be easily constructed so they are doing not have such low-weight codewords and that they can, therefore, achieve vanishingly small BERs. (The error-floor problem in turbo codes will be alleviated by careful design of the interleaver.) Turning next to the difficulty of decoding complexity, we note that the computational complexity of a turbo decoder is dominated by the MAP algorithm, which operates on the trellis for representing the convolutional code employed in the encoder. the quantity of computations in each recursion of the MAP algorithm scales linearly with the amount of states (with 16 states or more) within the trellis. In contrast, LDPC codes use a straightforward parity-check trellis that has just two states. Consequently, the decoders for LDPC codes are significantly simpler to style than those for turbo decoders. However, a practical objection to the utilization of LDPC codes is that, for big block lengths, their encoding complexity is high compared with turbo codes. It is argued that LDPC codes and turbo codes complement one another, giving the designer more flexibility in selecting the proper code for extraordinary decoding performance.

3.1 LDPC encoding techniques:

Some Constructions of LDPC Codes:
The most obvious thanks to design an LDPC code is to construct a sparse check matrix H with suitable properties. Important criterion includes efficient encoding, near-capacity performance, and low error floor. For long codes, randomly choosing a check matrix nearly always produces a decent code. However, in practical applications the block lengths might not be large enough, possibly thanks to latency constraints. Most LDPC codes are constructed pseudorandomly, where the fundamental construction is random but also features some structure, and certain bad configurations—such as 4-cycles—are avoided by the development or removed afterwards. Many (pseudo)random and algebraic constructions of LDPC codes are proposed. The LDPC codes may be constructed using different techniques.

- A Simple Random Construction (MacKay–Neal Construction)
- Gallager’s Random Construction
- Permutation Matrices based on Random Construction
- Cyclic Shift Matrices based on Algebraic Construction
- Finite Geometries based on Algebraic Constructions
- Repeat-Accumulate Codes
- Graph-based constructions
- the bit filling construction
- Combinatorial constructions

Combinatorial designs lead to check matrices that are both sparse and highly redundant in the sense that they contain many linearly dependent columns, i.e., M is much larger than $N - K$. The linearly dependent check equations help achieve good performance of iterative decoding even for moderate blocklengths.

3.2 LDPC Decoding techniques:

3.2.1Iterative Decoding:

Each bit node passes its bit value as a binary message to every adjacent check node, and every check node passes its check value as a binary message to every adjacent bit node. Significant overlap between two rows of H or, equivalently, short cycles within the TG reduce the effectiveness of the bit-flipping algorithm and will even prevent the algorithm from converging. for instance, assume that two rows of H have two “1-positions” in common. this implies that two senseword bits are simultaneously involved in two checks—thus, when both of the checks are 1, it's impossible to make a decision which bit is in error—and, also, that there's a cycle of length 4 within the TG. it's then possible that in each iteration of the bit-flipping algorithm, the 2 bits are flipped simultaneously and, over the iterations, each of them is alternatingly correct and incorrect. Hence, in this case, the algorithm fails to converge.

Belief Propagation Algorithm
Cycles within the TG adversely affect the performance and convergence of iterative BP decoders. In fact, cycles cause correlations within the probabilities gone by the BP decoder; the smaller the cycles the less the number of iterations that are correlation-free. Removing girth 4 and girth 6 will cause Large improvements in decoding performance. However, because the the girth of the TG is increased further, the extra improvements tend

to diminish. For LDPC codes with very large blocklength, the TG is assumed to be acyclic. the number of iterations needed to realize convergence of BP decoding increases when the channel SNR is reduced. Below a threshold SNR, the BP decoder doesn't converge in any respect.

3.3 Advantages of LDPC codes over turbo codes:

- LDPC codes have better block error performance on bursty channels.
- LDPC codes are well suited for high rates, and any block length. Where as rate of turbo codes need to be adjusted using a puncturing scheme.
- Less error floor tends to occur at a lower BER.
- The encoder and decoder do not require interleavers.
- A single LDPC code can be universally good for a collection of channels.
- There exist iterative LDPC decoding algorithms that are easy to implement, have moderate complexity (which scales linearly with the blocklength), and are parallelizable in hardware. In particular, LDPC decoding using the belief propagation (sum-product) algorithm tends to be less complex than turbo decoding using the BCJR algorithm.
- LDPC decoders inherently check if a codeword satisfying the check equations has been found, and otherwise declare a decoding failure. In contrast, turbo decoders usually need to perform additional operations to compute a stopping criterion, and even then, it is not clear if the decoding result corresponds to a codeword satisfying the check equations.

4. CONCLUSION

In this paper, we provide comparison between the coding schemes: turbo and LDPC codes in terms of their structure, encoding and decoding. It is observed that LDPC codes have better performance in terms of blocklength flexibility, doesn’t require the interleavers and puncturing operations, moderate decoding complexity and lower error floor. On the other hand, The encoding complexity of Turbo codes for large blocklength is less as compared to LDPC codes. In terms of implementation, turbo codes are more practical in contrast to LDPC codes.

REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1,” IEEE International Conference on Communications, 1993.

[2] R.G. Gallarger, “Low Density Parity Check codes,” IRE Transactions on Information Theory, 1962.

[3] C.-H. Lin, C.-Y. Chen, E.-J. Chang, and A.-Y. Wu, “A 0.16 nJ/bit/iteration 3.38 mm 2 turbo decoder chip for WiMAX/LTE standards,” in Integrated Circuits (ISIC), 2011R

[4] Roberts MK (2019) Simulation and implementation design of multi-mode decoder for WiMAX and WLAN applications. Measurement 131:28–34

[5] Y. Kou, S. Lin, and M. P. C. Fossorier, ‘‘Low-density parity-check codes based on finite geometries: A rediscovery and new results,’’ IEEE Trans. Inf. Theory, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[6] MacKay, D. J., & Neal, R. M. (1997). Near Shannon limit performance of low density parity check codes. Electronics letters, 33(6), 457–458.

[7] P. H. and Y. Wu, “On the complexity of turbo decoding algorithms,” IEEE Vehicular Technology Conference, vol. 2, pp.1439-1443, May 2001.

[8] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” IEEE Globecom, pp. 1680–1686, 1989.

[9] S. Benedetto and G. Montorsi, “Design of parallel concatenated convolutional codes,” IEEE Trans. Commun., vol. 44, pp. 591–600, May 1996.

[10] 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

[10] C. H. Lee and W. Wolf, ‘‘Implementation-efficient reliability ratio based weighted bit-flipping decoding for LDPC codes,’’ Electron. Lett., vol. 41, no. 13, pp. 755–757, Jun. 2005.

[11] H. R. Zeidan and M. M. Elsabrouty, ‘‘Low complexity iterative decoding algorithm for low-density parity-check (LDPC) codes,’’ in Proc. 1st IFIP IEEE Wireless Days WD, Nov. 2008, pp. 1–5.

[12] C.-L. Wang, X. Chen, Z. Li, and S. Yang, ‘‘A simplified min-sum decoding algorithm for non-binary LDPC codes,’’ IEEE Trans. Commun., vol. 61, no. 1, pp. 24–32, Jan. 2013

[13] A. Salbiyono and T. Adiono, ‘‘LDPC decoder performance under different number of iterations in mobile WiMax,’’ in Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS), Dec. 2010, pp. 1–4.

[14] D. Declercq and M. Fossorier, ‘‘Decoding algorithms for nonbinary LDPC codes over GF(q),’’ IEEE Trans. Commun., vol. 55, no. 4, pp. 633–643, Apr. 2007.