SIMULATED PERFORMANCE OF SERIAL CONCATENATED LDPC CODES

Adam G. Panagos
Electrical Engineering Department
University of Missouri at Rolla

ABSTRACT

With the discovery of Turbo Codes in 1993, interest in developing error control coding schemes that approach channel capacity has intensified. Some of this interest has been focused on low-density parity-check (LDPC) codes due to their high performance characteristics and reasonable decoding complexity. A great deal of literature has focused on performance of regular and irregular LDPC codes of various rates and on a variety of channels. This paper presents the simulated performance results of a serial concatenated LDPC coding system on an AWGN channel. Performance and complexity comparisons between this serial LDPC system and typical LDPC systems are made.

KEYWORDS

Error control coding, low-density parity-check codes, serial concatenated, and encoding complexity.

INTRODUCTION

Putting things simply, the primary goal of an error-control coding system is to achieve channel capacity while minimizing bit-error rate and the complexity of encoding and decoding operations. Shannon's landmark work [1] of 1948 showed that random codes were capacity achieving codes, i.e. high performance codes. However, due to their random construction, methods for encoding and decoding were very impractical.

Since 1948 much effort has been invested in creating error control coding schemes that could achieve capacity but with reasonable encoding and decoding algorithms. Low-density parity-check (LDPC) codes were first introduced in 1963 by Gallager in [2], but except for a few minor developments that followed were largely forgotten until recently. It wasn't until the discovery of Turbo Codes in 1993 by Berrou, Glavieux, and Thitimajshima in [3] that interest in LDPC codes resurged. One of the first interesting results published after this dormant period was a paper by MacKay and Neal that showed that LDPC codes performed as close to the Shannon limit as Turbo Codes [4].

Since this initial work by MacKay and Neal in the mid nineties, a multitude of papers and research has followed. A variety of new methods and alterations have been applied to the original work of Gallager, MacKay, and Neal to further improve system performance and reduce system complexity [5], [6], [7], [8]. The goal of this paper is to investigate a simple method for reducing *encoding* complexity.

LDPC CODE ENCODING COMPLEXITY

LDPC codes are able to achieve near capacity performance on a variety of channels, and because of this are a very much investigated area of research. Although their performance characteristics are very pleasing, they do have the undesirable feature of high encoding complexity. Using a straightforward encoding algorithm the complexity of the encoding is quadratic in the length of the code, n^2. Turbo codes on the other hand have encoding complexity that is linear with n [7].

With the computing power available today, the quadratic encoding complexity is not a significant issue for short length codes. However, short length LDPC codes don't achieve capacity. It is only as the length of the code is allowed to increase that LDPC codes begin to asymptotically approach the Shannon Limit. Thus, good performing codes require large computational encoding requirements.

Various authors have addressed the issue of encoding complexity. Sipser and Spielman for example replaced bipartite graphs used in standard LDPC code construction with expander graphs to achieve linear computationally expensive codes [9]. Another approach at solving this problem was proposed in [7]. In this case, the form of the parity-check matrix is manipulated before encoding with a pre-processing algorithm. Although the pre-processing steps involved in altering the form of the parity-check matrix have non-linear complexity in n, the resulting encoding procedure proposed has approximately linear complexity with n. We note that while both of these approaches were successful in reducing encoding complexity, overall system performance decreased as well.

The approach to reduce encoding complexity taken in this paper is rather straightforward. Since the computational burden of encoding arises with long codes, long codes were simply not used. Instead, multiple stages consisting of relatively short length codes were concatenated in a serial manner. By keeping the length of each code short, we immediately reduce encoding computational requirements. However, the question that now must be answered is, "How well does this system perform"?

As mentioned briefly above, LDPC codes with large n outperform LDPC codes with small n [6]. Although small length LDPC codes were used as building blocks of the serial concatenated system, since each code is constructed in a random fashion, it is hoped that the encoder as a whole performs as well as long length codes. By using a second stage of encoding, we obtain a second "look" at the original message data, similar in concept to the use of interleavers with Turbo Codes. It is hoped that this second encoding stage will result in a dramatic increase in performance even though relatively short length codes are used. While we have certainly decreased the encoding complexity by only choosing small length codes, the performance of this system will have to be evaluated by simulation.

SERIAL CONCATENATED SYSTEM

The serial concatenated error-control coding system devised in this paper can be seen below in Figure 1.

Figure 1 – Serial Concatenated LDPC Code System

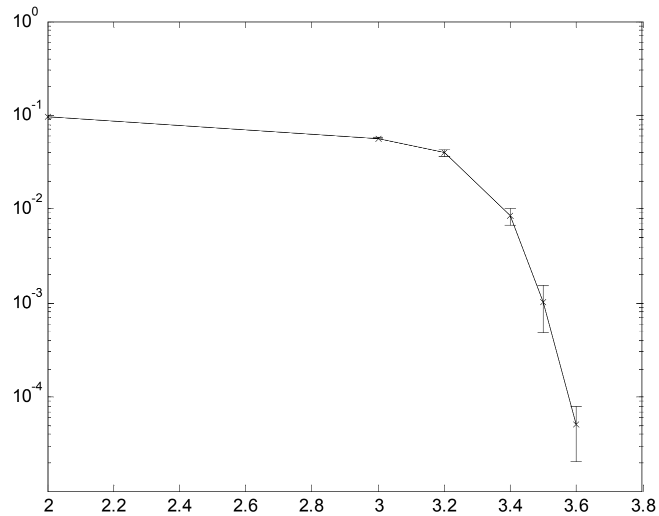
The output of LDPC Encoder #1 consists of the original message bits and the parity bits added by this first stage of encoding. All of these bits are then treated as message bits for the second stage of encoding. Thus, the output of LDPC Encoder #2 consists of the original message bits, parity bits for the original message bits, and parity bits for the output of LDPC Encoder #1. Both encoders are (10,000, 7071) codes and thus the overall rate of this system is 0.5.

After transmission over the additive white Gaussian channel (AWGN), the original message bits are decoded using the belief propagation message passing algorithm. The decoding process proceeds as follows. First, LDPC Decoder #2 takes the output of the channel and, ideally, decodes the received bits to the original message bits and parity bits generated by LDPC Encoder #1. Since LDPC Decoder #1 is expecting some type of *soft* information, i.e. something that would have been received at the output of an AWGN channel, the bits output by LDPC Decoder #2 are converted to the antipodal signals plus and minus one. These values are then treated as "received" values by LDPC Decoder #1 and, ideally, decoded to the original transmitted message bits. A maximum of 100 decoding iterations were used for each decoding stage.

SIMULATION RESULTS

The serial concatenated system of Figure 1 was simulated in the C programming language using software freely available for research purposes at [10]. This software provides all of the necessary building blocks for LDPC code generation, transmission across a variety of channels, and message decoding using the belief propagation decoding algorithm. Having obtained the basic functions necessary for simulation, scripts were written to construct the serial concatenated system and perform numerous simulations at various signal-to-noise ratios. The performance characteristics of this system have been plotted below in Figure 2. The error bars indicate 95% confidence intervals.

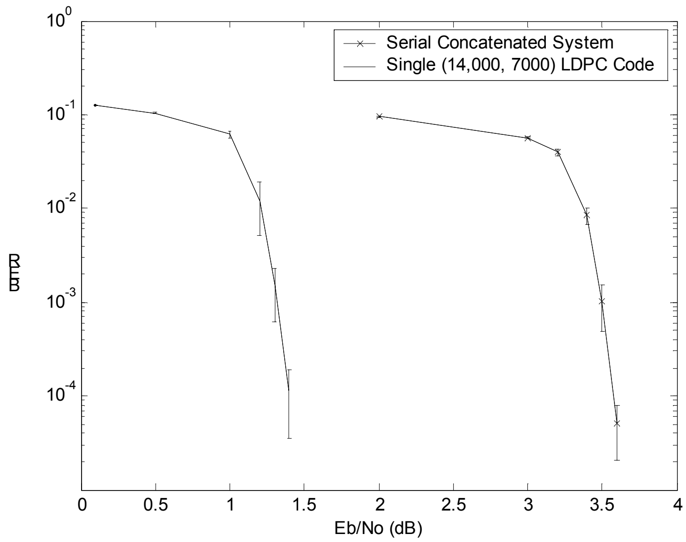
Figure 2 – System Performance



We note that since the construction of the LDPC codes is done in a random manner, the performance characteristics given in Figure 2 represent the performance characteristics of an ensemble of serially concatenated (10000,7071) codes.

It is now important to compare the performance of this serially concatenated system to other coding schemes in order to determine if the reduction in encoding complexity has come at the cost of reduced system performance. Figure 3 compares the performance of the serially concatenated system to the performance of a single LDPC code of equivalent rate and encoding complexity. From this comparison we reach the disappointing realization that there has been a significant reduction in system performance. Thus, based on the simulations performed for this single family of codes, i.e. serially concatenated (10,000, 7071) codes, it appears that serially concatenating LDPC codes in order to reduce encoding complexity and decoding in the “stage-at-a-time” manner described above, leads to the undesirable effect of dramatically degrading system performance.

Figure 3 – System Performance Comparison



FUTURE WORK

Although system performance was expected to decline due to the use of short length LDPC codes, the magnitude of this performance decline was significantly more than anticipated. There are several thoughts as to why the system performance decreased so dramatically.

First, decoding message bits is currently accomplished by completely decoding each stage individually, and then passing appropriate information between the encoders for the next stage of decoding. This decoding method is different in concept than the well known BCJR decoding algorithm used with Turbo Codes [3]. In this algorithm, information is passed between decoders during each iteration of the decoding process. This clever method of passing information is one of the keys in achieving high levels of system performance. Creating a message passing algorithm that allows the exchange of information between decoders during each iteration of the decoding process for this serially concatenated system should increase system performance.

Second, the LDPC codes constructed in this work are known as *regular* LDPC codes since they are constructed by placing a fixed number of checks per column in the parity-check matrix. It has been shown [6] that *irregular* LDPC codes, (codes that are constructed without this restriction), can significantly outperform regular LDPC codes. In this same work it is also shown that there are optimal degree distributions for an LDPC code's corresponding bipartite graph. Using an irregular construction method to obtain known good degree distributions is another approach that should be considered to obtain improved system performance. Although the construction method used here was rather simple, we note that length 4 cycles of the factor graph representation have been removed whenever possible since they have been shown to reduce system performance.

CONCLUSION

A serial concatenated LDPC coding scheme has been designed, simulated, and evaluated. The primary goal of this system was to avoid one of the undesirable characteristics of LDPC codes, namely, encoding complexity. Although the encoding complexity of the system was kept reasonably low by using relatively short length constituent codes, the degradation in system performance as compared to a single LDPC code of equivalent encoding complexity was significant. While the performance results presented here are mediocre at best, several thoughts for improving system performance have been presented and will be investigated in subsequent research.

REFERENCES

[1] Shannon, C.E., "A mathematical theory of communication," Bell System Technical Journal, vol. 27, 1948, pp. 379-423 and 623-656.

[2] Gallager, R.G., "Low-density parity-check codes". Cambridge, MIT Press, MA, 1963.

[3] Berrou, C., Glavieux, A., and Thitimajshima, P., "Near Shannon limit error-correcting coding and decoding: turbo-codes," Proceedings of 1993 IEEE Int. Conf. on Communications, Geneva, Switzerland, pp. 1064 -1070.

[4] MacKay, D.J.C. and Neal, R. M., "Near Shannon limit performance of low density parity check codes," Electronic Letters, vol. 32, Aug., 1996, pp. 1645-1646.

[5] Richardson, T.J. and Urbanke, R. L., "The capacity of low-density parity-check codes under message-passing decoding," IEEE Trans. Info. Theory, vol. 47, Feb., 2001, pp. 599-618.

[6] Richardson, T.J., Shokrollahi, M.A., and Urbanke, R.L., "Design of capacity-approaching irregular low-density parity-check codes," IEEE Trans. Info. Theory, vol. 47, Feb., 2001, pp. 619-637.

[7] Richardson, T.J., and Urbanke, R.L., "Efficient encoding of low-density parity-check codes," IEEE Trans. Info. Theory, vol. 47, Feb., 2001, pp. 638-656.

[8] Luby, M.G., Mitzenmacher, M., Shokrollahi, M.A., and Spielman, D.A., "Improved low-density parity-check codes using irregular graphs," IEEE Trans. Info. Theory, vol. 47, Feb., 2001, pp. 585-598.

[9] Sipser, M. and Spielman, D.A., "Expander Codes," IEEE Trans. Info. Theory, vol. 42, Nov., 1996, pp. 1710-1722.

[10] Neal, R., "Software for low-density parity-check (LDPC) codes" http://www.cs.utoronto.ca/~radford/ldpc.software.html