Turbo Codes

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 Eb/No 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". In seven pages the authors described an approach to coding that, in their supporting analysis, indicated that it was possible to operate within 0.7dB of the Shannon limit. The potential performance offered by turbo codes has excited both academic and industrial researchers. The last 7 years has seen a consequent explosion of research into all aspects of turbo codes.

This paper examines the principles of turbo coding and decoding, focussing on the coding and decoding algorithms. It then examines the performance of turbo codes, both in multilevel and simple constellations. It concludes by examining the state of turbo code research and development.

Principles of Turbo Codes

It is theoretically possible to approach the Shannon limit by using a block code with large block length or a convolutional code with a large constraint length. The processing power required to decode such long codes makes this approach impractical.

Turbo codes overcome this limitation by using recursive coders and iterative soft decoders. The recursive coder makes convolutional codes with short constraint length appear to be block codes with a large block length, and the iterative soft decoder progressively improves the estimate of the received message.

Coding

A specific type of convolutional coder is used to generate turbo codes. The convolutional coder shown in Figure 1a has a single input, x, outputs p_0 and p_1, and a constraint length K=3. Multiplexing the outputs generates a code of rate R=1/2.

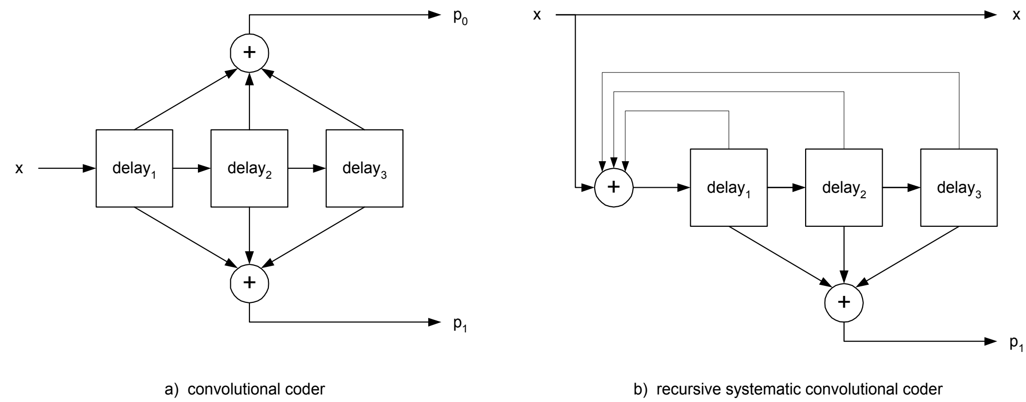

Figure 1 Convolutional Coders

The convolutional coder shown in Figure 1b differs in that one of the outputs, p_0, has been "folded back" and is presenting one of its output sequences at the coder input, making it recursive. This has the effect of increasing the apparent block length without affecting the constraint length of the coder. The input is also presented as one outputs of the coder, making it systematic. Such coders are thus called recursive systematic convolutional (RSC) coders.

In non-recursive convolutional codes it is common practice to flush the coder with zeros to bring the decoder to an end state. Flushing with zeros does not readily work with recursive coders, however relatively simple binary arithmetic can establish the input sequence that will generate a zero state. RSC codes can thus be made to appear like linear block codes.

A turbo code is the parallel concatenation of a number of RSC codes. Usually the number of codes is kept low, typically two, as the added performance of more codes is not justified by the added complexity and increased overhead. The input to the second decoder is an interleaved version of the systematic *x*, thus the outputs of coder 1 and coder 2 are time displaced codes generated from the same input sequence. The input sequence is only presented once at the output. The outputs of the two coders may be multiplexed into the stream giving a rate R=1/3 code, or they may be punctured to give a rate R=1/2 code. This is illustrated in Figure 2.

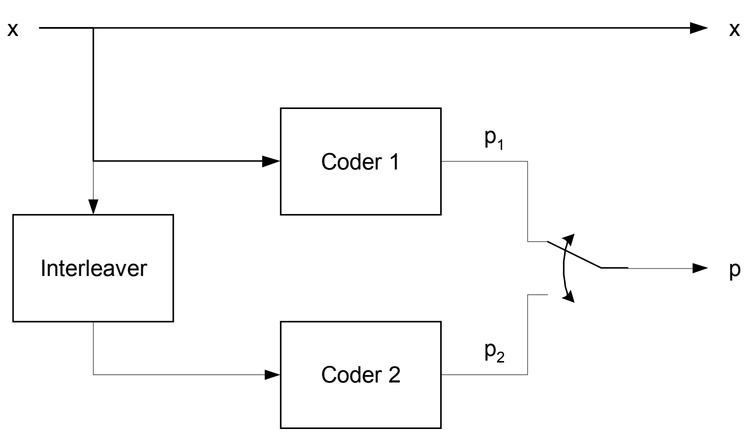

Figure 2 Punctured Rate R=1/2 Turbo Coder

The interleaver design has a significant effect on code performance. A low weight code can produce poor error performance, so it is important that one or both of the coders produce codes with good weight. If an input sequence *x* produces a low weight output from coder 1, then the interleaved version of *x* needs to produce a code of good weight from coder 2. Block interleavers give adequate performance, but pseudo random interleavers have been shown to give superior performance[3].

Decoding Principles

At the receiver, the signal is demodulated with its associated noise and a soft output provided to the decoder. The soft output might take the form of a quantized value of the decoded bit with its associated noise, or it may be a bit with associated probability (i.e. 1 with P(1)=0.65). Most often it is the log likelihood ratio (LLR), which is defined as:

$$\Lambda_i = \ln\left(\frac{P\langle m_i = 1 | y'\rangle}{P\langle m_i = 0 | y'\rangle}\right)$$

The LLR is a measure of the probability that, given a received soft input *y'*, the message bit m_i associated with a transition in the trellis is 1 or 0. If the events are equiprobable then the output is 0, but any tendency for m_i towards 1 or 0 will result in positive or negative values of Λi.

It is simplest to view the decoding process as 2 stages: initialising the decoder and decoding the sequence. The demodulator output contains the soft values of the sequence x' and the parity bits p_1' and p_2'. These are used to initialise the decoder, as shown in Figure 3a. The interleaved sequence is sent to decoder 2, while the sequence derived from x' is sent to decoder 1 and presented to decoder 2 through an interleaver. This re-sequences bits from streams x*'* and p_1' so that bits generated from the same bit in x are presented simultaneously to decoder 2, whether from x, p_1' or p_2'.

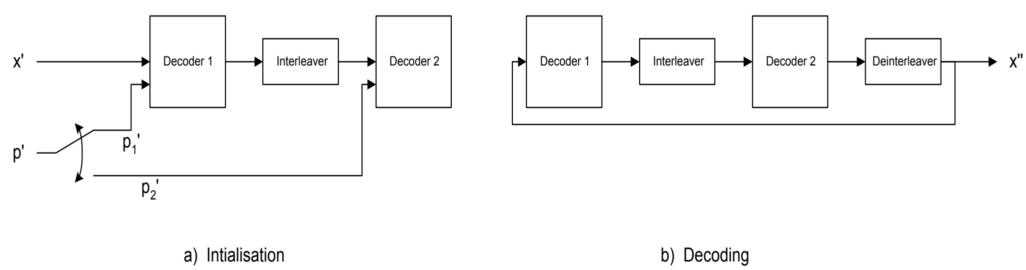

Figure 3 Turbo Decoder

The decoder may have some knowledge the probability of the transmitted signal, for example it may know that some messages are more likely than others. This *a priori* information assists the decoder, which adds information gained from the decoding process forming the *a posteriori* output. The decoder uses all this information to make its best estimate of the received sequence. The output is then de-interleaved and presented back to decoder 1, which makes its best estimate. Further iterations through decoders 1 and 2, with associated interleaving and de-interleaving, refine the estimate until a final version of the block, x'', is presented at the output. This process is shown in Figure 3b.

Decoding Algorithms

The two main types of decoder are Maximum A Posteriori (MAP) and the Soft Output Viterbi Algorithm[4] (SOVA). MAP looks for the most likely symbol received, SOVA looks for the most likely sequence. Both MAP and SOVA perform similarly at high Eb/No. At low Eb/No MAP has a distinct advantage, gained at the cost of added complexity.

MAP was first proposed by Bahl[5] et al and was selected by Berrou[2] et al as the optimal decoder for turbo codes. MAP looks for the most probable value for each received bit by calculating the conditional probability of the transition from the previous bit, given the probability of the received bit. The focus on transitions, or state changes within the trellis, makes LLR a very suitable probability measure for use in MAP.

SOVA is very similar to the standard Viterbi algorithm used in hard demodulators. It uses a trellis to establish a surviving path but, unlike its hard counterpart, compares this with the sequences that were used to establish the non-surviving paths. Where surviving and non-surviving paths overlap the likelihood of that section being on the correct path is reinforced. Where there are differences, the likelihood of that section of the path is reduced. At the output of each decoding stage the values of the bit sequence are scaled by a channel reliability factor, calculated from the likely output sequence, to reduce the probability of over-optimistic soft outputs. The sequence and its associated confidence factors are then presented to the interleaver for further iterations. After the prescribed number of iterations, the SOVA decoder will output the sequence with the maximum likelihood.

Performance of Turbo Codes

Berrou's impressive BER of 10^{-5} at Eb/No of 0.7 dB does not tell the whole story. To achieve this required 18 iterations and a block length of 65532 data bits. If this channel supported a speech coder at 4.8 Kbit/s it would take 15 seconds to receive a block. This latency would be impractical for telephony, but would be more than adequate for receiving images from a deep space probe.

Most of the assessments of turbo code performance have resulted from simulation. In the ideal environment of a simulation, free of the constraints of real systems, its is possible to produce highly impressive results. To apply turbo codes to real systems requires acceptance of real world constraints such as latency and computing power.

Wang[6] has explored the performance of codes with parameters set to values that are more practical. He determined that the performance of turbo codes was influenced by four main factors: the number of iterations, constraint length, interleaver design and puncturing. His simulations were based on use of the MAP decoder and constraint lengths of 3-5. He concluded that:

- Ten decoding iterations are adequate.
- A constraint length of greater than 3 adds little coding gain for short block sizes (around 100)
- Random interleavers provide the best all round performance.
- The use of 1/2 rate punctured codes degrades the BER performance by only 0.5 to 0.7 dB relative to 1/3 rate unpunctured codes.

Wang also observed that there is a performance floor around Eb/No of 3dB, typically yielding BER in the region 10^{-6} to 10^{-8}. The reasons for this floor are the subject of current research.

While there is considerable material reporting on the optimum performance of turbo codes, surprisingly little material exists reporting on the performance of turbo codes in practical scenarios. Clearly, exploring the lower limits of turbo code performance can provide an insight into their practical limitations. Real decoders need to provide the best BER from the worst channel in the shortest time. A realistic implementation would have low bandwidth, and thus use punctured codes, short block sizes, few iterations and the lowest Eb/No capable of supporting the required service. With this in mind some additional simulations were undertaken as part of this assignment to explore code performance in realistic implementations.

Simulations

MATLAB routines for simulating turbo codes are available on the Internet[7], with the proviso that they are only used for educational purposes. In exploring the performance of turbo codes several simulations were run on a PC. For the purposes of this simulation a punctured turbo code at rate R=1/2 was used. The data block length was 400 bits, and a MAP decoder was used in the simulation. The results shown at Figure 4 are the BER vs Eb/No curves for different numbers of iterations from n=1, 2, 5 and 10. A BPSK channel was assumed.

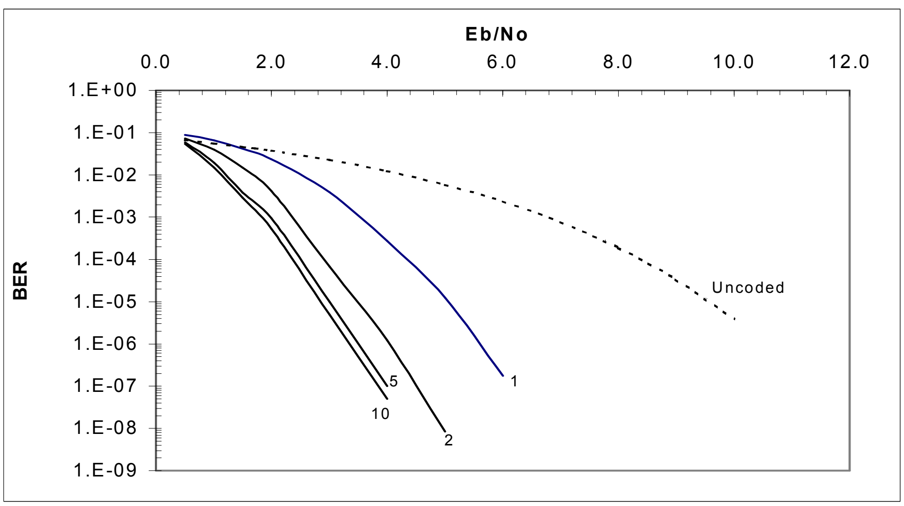

Figure 4 BER for Turbo Code

It can be seen that BERs of the order of 10^{-5} are achievable with Eb/No >3 dB with modest numbers of iterations. A typical coding gain of Eb/No of 7dB, relative to an uncoded channel, was observed at a BER of 10^{-5}. Figure 4 infers that the BER should improve with each iteration, so a series of simulations were run to evaluate the improvement. It can be seen from Figure 5 that the first few iterations yield the most significant improvements in BER for any given Eb/No. Thereafter the results appear to converge onto a BER for each value of Eb/No. It is apparent that there is a trade off to be made between the number of iterations, processing power, and Eb/No when seeking a given BER.

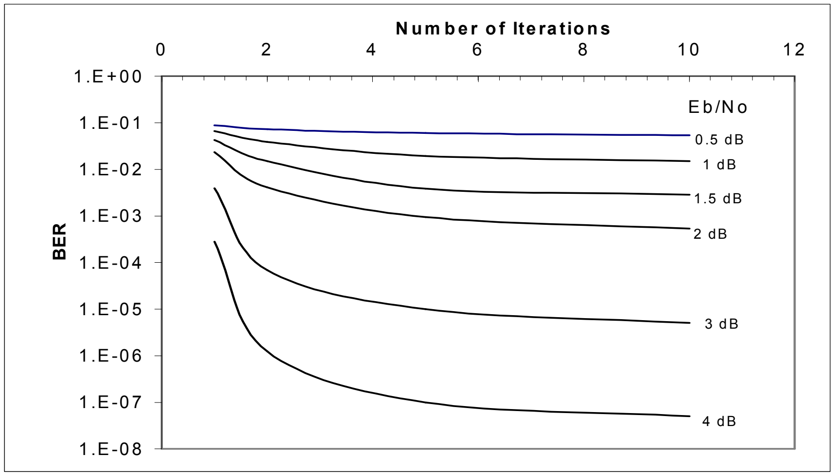

Figure 5 BER vs Eb/No as number of iterations varies

A final simulation was run to compare the performance of the MAP and SOVA decoders, particularly at low values of Eb/No. The number of iterations was set at 5, since Figure 4 and Figure 5 indicated that further iterations would yield marginal improvements. The results, shown at Figure 6, confirm that MAP is about 0.5 dB better than SOVA at low values of Eb/No.

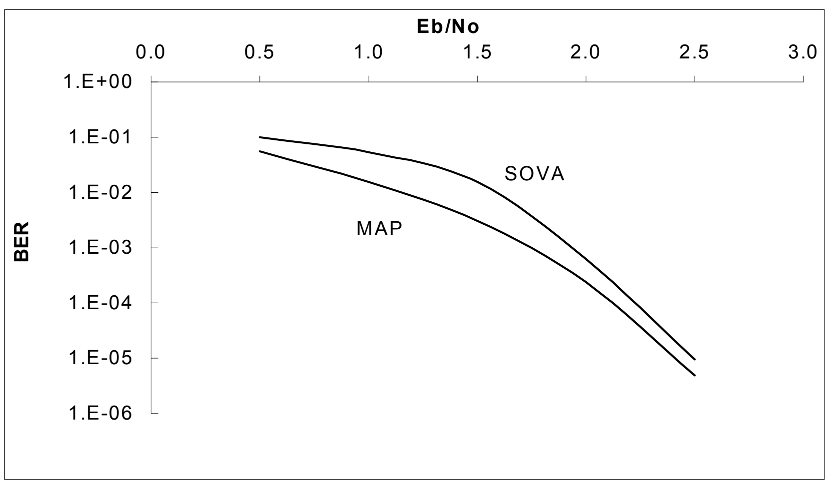

Figure 6 Comparison of MAP and SOVA at Low Eb/No

Published Results

Efficient and effective coding is not an end unto itself; the performance of the modulation scheme has equal importance to the communications channel. The ideal combination would be a spectrally efficient modulation scheme combined with a robust coding scheme.

Shortly after their original paper on turbo codes, Berrou and Glavieux joined with Le Goff to examine the integration of turbo codes with high spectrally efficient modulation schemes[8]. In a highly readable paper they examined the association between turbo codes and multi-level modulation schemes, QAM and MPSK, over Gaussian and Rayliegh channels. The coding gains obtained for an AWGN channel are at Figure 7.

Turbo code rate	½	2/3	3/4	2/3
Modulation	16 QAM	8 PSK	16 QAM	64 QAM
Spectral Efficiency (bits/Hz)	2	2	3	4
Coding gain at 10^{-6} over uncoded modulation	6.0 dB	5.5 dB	7.8 dB	5.8 dB
Coding gain at 10^{-6} over 64 state TCM	2.4 dB	1.9 dB	2.6 dB	2.2 dB

Figure 7 Coding Gains for AWGN Channels

The graphs in the paper show consistently lower BER being predicted for turbo coded 16 QAM than for 64 TCM at BERs below 10^{-3}, for both Gaussian and Rayliegh channels.

Chang and Wei[9] proposed methods for integrating turbo codes with q-ary modulation schemes, focussing on 4-ary (16 QAM) and 8-ary (64 QAM) modulation. Gray code mapping was used. The basic principle, which others seem to have followed, is to code, interleave, puncture and decode at symbol boundaries. For rate R = 1/2, 16 QAM, they reported a 1dB improvement over the corresponding binary (BPSK) turbo code at a BER of 10^{-5}. For a rate R=2/3, 64 QAM, code they reported a gain of about 0.6 dB at a BER of 10^{-5}, observing that the code was approaching the error floor.

Turbo Code Research & Development

It is obvious that turbo codes have the potential to make a significant contribution to communications systems, particularly those that operate with a low Eb/No. The last 10 years have seen turbo codes move from theory, through simulation, to the emergence of the first products. The key enabling technology has been the emergence of electronic devices capable of implementing the required number of operations per second.

The universities who take an active role in research and publications in this field include Virginia, South Australia, Notre Dame, Surrey, Southampton, Kiel, the Technical University of Denmark, Caltech and Cornell. JPL are also actively involved in research into turbo codes, and some of their staff are prolific authors.

Research

A search of the IEEE and IEE websites reveals a rich vein of papers, reflecting the intensity of research into turbo codes. Initial research was focussed on establishing the fundamental properties of turbo codes and their performance envelope. Typical areas of research included:

- applying turbo codes to different modulation schemes[9, 10]
- establishing the factors that affect code performance[6, 11, 12]
- exploring the effects of interleaver design[13]
- types of decoders[14, 15, 16]

It is generally concluded that turbo codes perform very well when compared to other convolutional and block codes, particularly when combined with multilevel constellations. The fundamental properties of turbo codes are now well understood. There is considerable potential for further research into the complex trade-offs between code mapping in constellations, coder and decoder design, and interleaver design. The latter seems to play an important role in turbo code performance, and has some affect on the observed error floor; this is the subject of ongoing research.

Shoemake et al[17] have looked at 8-PSK modulation schemes, focussing on the constellation mapping. They observed the lack of an error floor on their simulation, and concluded that the constellation mapping and good interleaver design were critical to code performance.

More recently, the focus of published papers moved towards the application of turbo codes to real situations. Obvious applications include the protection of radio channels, where power limiting or fading can cause low signal to noise ratios. Typical areas of research are mobile satellite communications[18], HF data systems[19], jammed channels[20], and on channels subject to multipath fading[21].

Numerous recently published papers[22] have addressed a number of mainstream research areas of interleaver design, modifications to the Bahl decoder, and performance issues. Others have looked outside the mainstream of research at issues such as codes for low data rates, diversity radio systems, and codes for use in memory-limited decoders.

The Proceedings for the MILCOM 1999 Conference[23] contain some interesting papers and concepts, including the implementation and performance issues for software radios, and code performance under jamming.

Development

Manufacturers are looking seriously at the advantages of turbo codes over the well-established Viterbi and Reed-Solomon (RS) codes. The initial emphasis has been on producing chipsets to allow manufacturers to implement turbo codes in their hardware. High-speed electronics has aided the development of chips with sufficient processing power to implement turbo decoders, currently at data rates of a few tens of Mbit/s. AHA and Comatlas have produced turbo code chipsets at 36 and 40 Mbit/s respectively, while Small World Comms have an FPGA for turbo decoding up to 90 Mbit/s.

These enabling technologies have helped to bring the first generation of turbo code based equipment to the marketplace. Comtech have launched a satellite modem that uses a rate R = 3/4 turbo code claiming significant bandwidth and BER improvements over modems using Viterbi and RS codes. Other major satellite modem manufacturers such as Comsat Labs are introducing turbo codes into their product. Alantro are developing turbo code firmware for such diverse roles as satellite links and hard disk drives.

The mobile phone industry is looking at turbo codes to provide error correction for third generation handsets. There have been many recent papers on the subject of implementing turbo codes in CDMA systems for UMTS. Turbo codes can operate at reduced power levels offering improved safety and extended battery life, both of which are important to the mobile phone user.

Recommended Reading

This short paper has given an introduction to the subject and its terminology. Material referred to and used in its preparation is listed at the end of this paper. The author would highly recommend "Digital Communications Fundamentals and Applications (second edition)" by Bernard Sklar as the next step in learning about turbo codes. This excellent textbook contains a chapter on turbo codes which takes the reader further than this paper; the remainder of the book sets coding in the wider context of the communications channel.

Conclusions

Turbo codes are a class of convolution code which exhibit the properties of large block codes through the use of recursive coders. Coder performance is heavily dependent on the design of the interleaver, which must ensure adequate weight for at least one of the codes. Soft decoders are used with turbo codes to allow the a posteriori probability to be passed between decoder iterations. The MAP decoder is generally preferred because it offers the same performance as SOVA at 0.5dB lower value of Eb/No. This performance edge is achieved at the cost of increased complexity.

A half rate turbo coded BPSK channel can offer coding gains of 7dB over an uncoded channel at a BER of 10^{-5}. The coding gain depends on the number of iterations; typically 5 to 10 iterations generate most of the improvement. Turbo code performance has been simulated for a number of high order constellations, including 8PSK, 16 QAM and 64QAM, and the importance of code mapping within the constellation has been recognised. The reasons for the observed error floor are not yet fully understood, but indications are that it is linked in some way to interleaver design.

Research is now beginning to be directed towards applying turbo codes to resolve real communications problems. There are many potential applications for turbo codes, particularly in the field of radio communications. With suitable chipsets becoming available the first products are beginning to be marketed. The superior performance offered by turbo codes ensures that they have a good future in information systems.

Several useful introductory texts have been identified in researching this paper and are recommended to newcomers to the subject[24, 25, 26].

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 & Thitimajshima P. "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes". IEEE 1993.

[3] Andersen JD and Zyablov VV, "Interleaver Design for Turbo Coding", Proceedings of the International Symposium on Turbo Codes, September 1997.

[4] Berrou C, Adde P, Angui E & Faudeil S. "A Low Complexity, Soft Output, Viterbi Decoder Architecture". Proceedings IEEE. International Conference on Communications. May 1993.

[5] Bahl LR, Cocke J, jelinek F & Rajiv J.. "Optimal Decoding of Linear Codes for Minimising Symbol Error Rate". IEEE Transactions on Information Theory, March 1974.

[6] Wang CC. "On the Performance of Turbo Codes". IEEE, 1998.

[7] http://www.ee.vt.edu/yufei/turbo.html

[8] Le Goff S, Glavieux A, Berrou C. "Turbo Codes and High Spectral Efficiency Modulation". IEEE 1994.

[9] Chang Y, Wei VK. "Q-ary Turbo Codes with QAM Modulations". IEEE 1996.

[10] Fazel K, Papke L. "Combined Multilevel Turbo Codes with 8-PSK modulation". IEEE 1995.

[11] Divsalar D, Dolinar S, McEliece RJ & Pollara F. "Performance Analysis of Turbo Codes". IEEE 1995.

[12] Dolinar S, Divsalar D and Pollara F. "Turbo Code Performance as a Function of Block Size". IEEE 1998.

[13] Yuan J, Vucetic B & Feng W. "Combined turbo codes and interleaver design". IEEE 1998.

[14] Jung P and Nasshan MM. "Comprehensive Comparison of Turbo Code Decoders". IEEE 1995.

[15] Fossorier MPC, Burkert F, Lin s & Hagenauer J. "On the Equivalence Between SOVA and max-Log-MAP Decoders". IEEE Communications letters 1998.

[16] Ambroze A, Wade G & Tomlinson M. "Iterative MAP Decoding for Serial Concatenated Comnvolutional Codes". IEE Proceedings on Communications, April 1998.

[17] Shoemake MB, Heegard C & Rossin E. "Turbo Codes for High Order Constellations". IEEE,1998.

[18] Fonseka JP. "Applications of Turbo Codes in Satellite Mobile Systems". Electronics letters January 1999.

[19] Coulton P, Honary A, Darnell M & Wicker SB. "Application of Turbo Codes to HF Data Transmission". IEE Conference on HF Radio Systems and Techniques". July 1997.

[20] Jordan MA. "Turbo Code Performance in Partial Band Jamming". IEEE 1998.

[21] Li KL and Cheung SW. "Performance of Turbo Codes with fading Compensation in Multipath Channels". IEEE 1999.

[22] IEEE/IEE Electronic Library Online

[23] Various authors. "Military Communications Conference Proceedings, 1999. MILCOM 1999". IEEE

[24] Valenti MC. "An Introduction to Turbo Codes".

[25] Battail, G. "A Conceptual Framework for Understanding Turbo Codes". IEEE Journal on Selected Areas in Communications, 1998.

[26] Sklar B. "Turbo Code Concepts made easy, or How I learned to Concatenate and Reiterate". IEEE 1997.