The Modified Max-Log-MAP Turbo Decoding Algorithm by Extrinsic Information Scaling for Wireless Applications

Mustafa Taskaldiran, Richard C.S. Morling, and Izzet Kale

Applied DSP and VLSI Research Group, Department of Electronic Systems, University of Westminster, London, W1W 6UW, United Kingdom

{m.taskaldiran, morling, kalei}@wmin.ac.uk

1. Introduction

In his 1948 paper [1], Shannon defined the limits of a communication system. He proved that there exists error correcting codes which can provide arbitrarily high reliability of transmission for information rates below the channel capacity. In spite of all efforts to find such error control codes, the gap between the Shannon limit and practice was still 2dB until 1993. A major advancement in the channel coding area was introduced by Berrou *et al* in 1993 by the advent of turbo codes [2]. Turbo codes have shown the best Forward Error Correction (FEC) performance known up to now. Turbo codes are revolutionary in the sense that they allow reliable data transmission within a half decibel of the Shannon Limit. At first, the extraordinary performance of turbo codes encountered some doubts by the communication community. However, their performance has been verified by many researchers in a short time after the emergence of turbo codes. A massive amount of research effort has been performed to facilitate the energy efficiency of turbo codes [3]. The superior performance of turbo codes has been studied and well understood [4]. As a result, turbo codes have been incorporated into many standards used by the NASA Consultative Committee for Space Data Systems (CCSDS), Digital Video Broadcasting (DVB), both Third Generation Partnership Project (3GPP) standards for IMT-2000, and Wideband CDMA which requires throughputs from 2 Mb/s to several 100 Mb/s.

The iterative nature of turbo-decoding algorithms increases their complexity compare to conventional FEC decoding algorithms. Two iterative decoding algorithms, Soft-Output-Viterbi Algorithm (SOVA) and Maximum A posteriori Probability (MAP) Algorithm require complex decoding operations over several iteration cycles. So, for real-time implementation of turbo codes, reducing the decoder complexity while preserving bit-error-rate (BER) performance is an important design consideration.

In this chapter, a modification to the Max-Log-MAP algorithm is presented. This modification is to scale the extrinsic information exchange between the constituent decoders. The remainder of this chapter is organized as follows: An overview of the turbo encoding and decoding processes, the MAP algorithm and its simplified versions the Log-MAP and Max-Log-MAP algorithms are presented in section 1. The extrinsic information scaling is introduced, simulation results are presented, and the performance of different methods to choose the best scaling factor is discussed in Section 2. Section 3 discusses trends and applications of turbo coding from the perspective of wireless applications.

1.1 Turbo Encoder

A generic structure for turbo encoding based on parallel concatenation of two Recursive Systematic Convolutional (RSC) encoders is given in Fig 1. Two identical RSC encoders produce the redundant data as parity bits. The input data stream and parity bits are combined in series to form the turbo coded word. The size of the input data word may vary from 40 bits to 5114 bits for UMTS [5] and take specified values such as 378, 570, and 20730 for CDMA2000 [6] turbo coding which are the two main standards of 3GPP and 3GPP2 respectively.

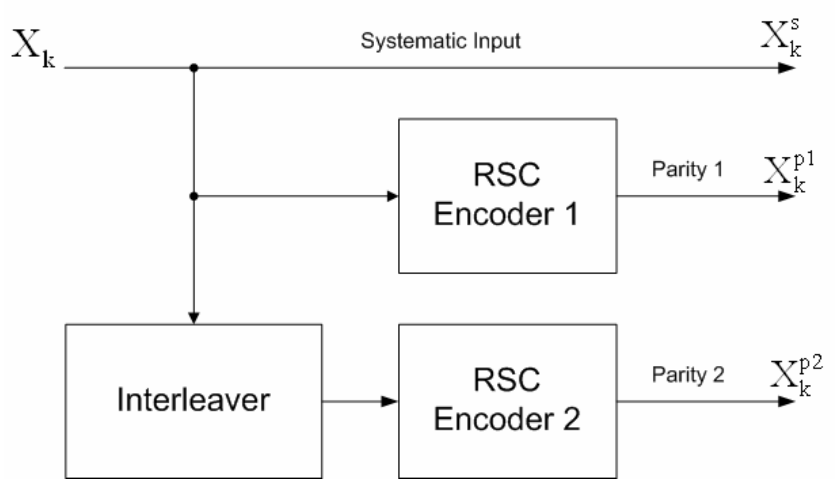

Fig. 1. Generic turbo encoder

The interleaver is the crucial part of turbo encoding as it shapes the weight distribution of the code in a way to produce low-weight code words. Opposite to their non-recursive counterparts, RCS encoders can only be terminated by certain terminating data sequences. The interleaver separating two RCS encoders prevents at least one of the encoders to terminate quickly. It is obvious that a data sequence terminating after a long period has a large Hamming distance and hence provides better error protection [3]. This improvement is called the *interleaver gain* which is one of the main reasons of the excellent performance of turbo codes [7].

The interleaver design also affects the turbo decoder performance by reducing the degree of correlation between the soft-output of each decoder which becomes the extrinsic information to the other decoder (Decoder1 & Decoder2 in Fig 2). As the degree of correlation between these two soft-information decreases the performance of the turbo decoder increases [8].

1.2. Turbo Decoder

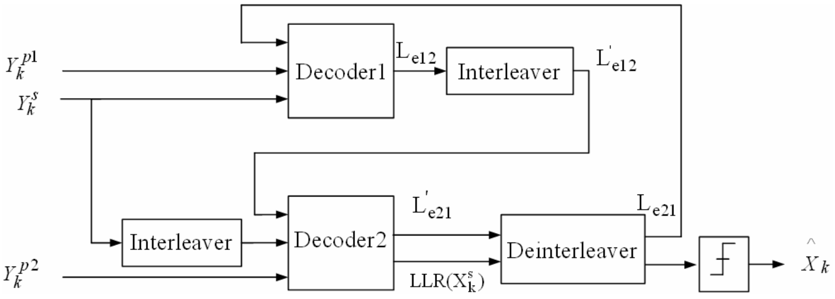

Fig. 2. Iterative Turbo Decoding

In a typical turbo decoding system (see Fig. 2), two decoders operate iteratively and pass their decisions to each other after each iteration. These decoders should produce soft-outputs to improve the decoding performance. Such a decoder is called a Soft-Input Soft- Output (SISO) decoder [9]. Each decoder operates not only on its own input but also on the other decoder's incompletely decoded output which resembles the operation principle of turbo engines. This analogy between the operation of the turbo decoder and the turbo engine gives this coding technique its name, "turbo codes" [10].

Turbo decoding process can be explained as follows: Encoded information sequence X_k is transmitted over an Additive White Gaussian Noise (AWGN) channel, and a noisy received sequence Y_k is obtained. Each decoder calculates the Log-Likelihood Ratio (LLR) for the k-th data bit d_k, as

$$L(d_k) = \log\left[\frac{P(d_k = 1 \mid Y)}{P(d_k = 0 \mid Y)}\right] \tag{1}$$

LLR can be decomposed into 3 independent terms, as

$$L(d_k) = L_{apri}(d_k) + L_c(d_k) + L_e(d_k) \tag{2}$$

where $L_{apri}(d_k)$ is the a-priori information of d_k, $L_c(x_k)$ is the channel-measurement, and $L_e(d_k)$ is the extrinsic information exchanged between the constituent decoders. Extrinsic information from one decoder becomes the a-priori information for the other decoder at the next decoding stage.

L_{e12} and L_{e21} in Figure 1 represent the extrinsic information from decoder1 to decoder2 and decoder2 to decoder1 respectively.

LLR computations can be performed by using one of the two main turbo decoding algorithms SOVA and MAP algorithms. The MAP algorithm seeks for the most likely data sequence whereas SOVA, which is a modified version of the Viterbi algorithm, seeks for the most likely connected path through the encoder trellis. The MAP algorithm is a more complex algorithm compared to SOVA. At high SNR, the performance of SOVA and MAP are almost the same. However, at low Signal-to-Noise Ratios (SNRs) MAP algorithm is superior to SOVA by 0.5 dB or more [9]. The following sections explain the MAP algorithm and its simplified versions Log-MAP and Max-Log-MAP algorithms.

1.3. The MAP Algorithm

The MAP algorithm is an optimal but computationally complex SISO algorithm. The Log-MAP and Max-Log-MAP algorithms are simplified versions of the MAP algorithm.

MAP algorithm calculates LLRs for each information bit as

$$L(d_k) = \ln\left[\frac{\sum\limits_{S_k}\sum\limits_{S_{k-1}} \gamma_1(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k)}{\sum\limits_{S_k}\sum\limits_{S_{k-1}} \gamma_0(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k)}\right] \tag{3}$$

where α is the forward state metric, β is the backward state metric, γ is the branch metric, and S_k is the trellis state at trellis time k. Forward state metrics are calculated by a forward recursion from trellis time $k=1$ to, $k=N$ where N is the number of information bits in one data frame. Recursive calculation of forward state metrics is performed as

$$\alpha_k(S_k) = \sum_{j=0}^{1} \alpha_{k-1}(S_{k-1})\gamma_j(S_{k-1}, S_k) \tag{4}$$

Similarly, the backward state metrics are calculated by a backward recursion from trellis time $k=N$ to, $k=1$ as

$$\beta_k(S_k) = \sum_{j=0}^{1} \beta_{k+1}(S_{k+1})\gamma_j(S_k, S_{k+1}) \tag{5}$$

Branch metrics are calculated for each possible trellis transition as

$$\gamma_i(S_{k-1},S_k)=A_k P(S_k \mid S_{k-1})\exp\left[\frac{2}{N_o}\left(y_k^s x_k^s(i)+y_k^p x_k^p(i,S_{k-1},S_k)\right)\right] \tag{6}$$

where $i=(0,1)$, A_k is a constant, x_k^s and x_k^p are the encoded systematic data bit and parity bit, and, y_k^s and y_k^p are the received noisy systematic data bit and parity bit respectively.

1.4. The Log-MAP Algorithm

To avoid complex mathematical calculations of MAP decoding, computations can be performed in the logarithmic domain. Furthermore, logarithm and exponential computations can be eliminated by the following approximation

$$\max{}^*(x,y) \triangleq \ln(e^x+e^y) = max(x,y) + \log(1+e^{-|y-x|}) \tag{7}$$

The last term in max*(.) operation can easily be calculated by using a look-up table (LUT).

So equations (3)-(6) become

$$\begin{aligned} L(d_k) = & \max_{(S_{k-1},S_k,1)}{}^*\left(\bar{\gamma}_1(S_{k-1},S_k)+\bar{\alpha}_{k-1}(S_{k-1})+\bar{\beta}_k(S_k)\right) \\ & -\max_{(S_{k-1},S_k,0)}{}^*\left(\bar{\gamma}_0(S_{k-1},S_k)+\bar{\alpha}_{k-1}(S_{k-1})+\bar{\beta}_k(S_k)\right) \end{aligned} \tag{8}$$

$$\bar{\alpha}_k(S_k) = \max_{(S_{k-1},i)}{}^*\left(\bar{\alpha}_{k-1}(S_{k-1})+\bar{\gamma}_i(S_{k-1},S_k)\right) \tag{9}$$

$$\bar{\beta}_k(S_k) = \max_{(S_k,i)}{}^*\left(\bar{\beta}_{k+1}(S_{k+1})+\bar{\gamma}_i(S_k,S_{k+1})\right) \tag{10}$$

$$\bar{\gamma}_i(S_{k-1},S_k)=\frac{2}{N_o}\left(y_k^s x_k^s(i)+y_k^p x_k^p(i,S_{k-1},S_k)\right)+\ln\left(P(S_k \mid S_{k-1})\right)+K \tag{11}$$

where K is a constant.

1.5. The Max-Log-MAP Algorithm

The correction function $\mathrm{f_c} = \log(1 + \mathrm{e}^{-|y-x|})$ in the $\max^*(.)$ operation can be implemented in different ways. The Max-log-MAP algorithm simply neglects the correction term and approximates the $\max^*(.)$ operator as

$$\ln(\mathrm{e}^{\mathrm{x}} + \mathrm{e}^{\mathrm{y}}) \approx \max(\mathrm{x,y}) \tag{12}$$

at the expense of some performance degradation.

This simplification eliminates the need for an LUT required to find the corresponding correction factor in the $\max^*(.)$ operation. The performance degradation due to this simplification is about 0.5dB compared to the Log-MAP algorithm [11].

2. Extrinsic Information Scaling

The extrinsic information exchanged between the constituent decoders can be scaled to improve the performance of turbo decoding [12-14]. With this modification equation (11) for branch metric calculations can be rewritten as

$$\bar{\gamma}_i(S_{k-1},S_k) = \frac{2}{N_o}\left(y_k^s x_k^s(i) + y_k^p x_k^p(i,S_{k-1},S_k)\right) + s_d \ln\left(P(S_k \mid S_{k-1})\right) + K \tag{13}$$

The only modification is the scaling factor s_d where $d = 1,2$ for decoder1 and decoder2 respectively.

Extrinsic information scaling has been proposed to compensate for the optimistic LLR calculations of SOVA [14]. A gain of 0.4 dB has been reported for a code of memory length 4 at BER of 10^{-4} [14]. Scaling factor modification has also been applied and tested on the Max-Log-Map algorithm. Authors of [12] have reported 0.2-0.4dB gain over the standard algorithm for 3GPP standards. They used a constant scaling factor of 0.7. In [13], scaling factor optimization for Max-Log-Map decoding is explained as mutual information combining which is the evolution of the information exchange between the two MAP decoders. The best scaling factors for each iteration were calculated for different SNRs by off-line computation. The performance difference between the modified Max-Log-Map and Log-Map was reported as 0.05 dB for UMTS-based turbo coding [13]. The performance improvement introduced by the scaling factor modification is explained as the correction of the accumulated bias due to maximum (max) operation in the Max-Log-Map algorithm [13].

2.1. Simulation Results

Table .1.Scaling factors for different SNRs and iterations for decoder1(D1) and decoder2 (D2) (R=1/3, interleaver length=1024,generator polynomial (13, 15) $_{oct}$)

Iterations	1		2		3		4		5		6	
E_b/N_o	D1	D2	D1	D2	D1	D2	D1	D2	D1	D2	D1	D2
0	0*	0.5	0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.7
0.25	0	0.6	0.6	0.6	0.6	0.6	0.6	0.7	0.6	0.7	0.7	0.7
0.50	0	0.6	0.6	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.7
0.75	0	0.6	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.7	0.7	0.8
1	0	0.6	0.7	0.8	0.8	0.7	0.8	0.6	0.5	0.7	0.4	0.8
1.25	0	0.7	0.9	0.7	0.5	0.9	0.5	0.8	0.4	0.8	0.3	1
1.5	0	0.7	0.7	0.8	0.5	1	0.5	0.9	0.4	0.9	0.4	0.8

*No extrinsic information from decoder 2 to decoder 1 for the 1st iteration.

Table 2. Scaling factors for different SNRs (R=1/3, interleaver length=5114, generator polynomial (13, 15) $_{oct}$)

E_b/N_o (dB)	0	0.25	0.5	0.75	1	1.25	1.5
Dec1	0.6	0.6	0.7	0.7	0.9	0.5	0.4
Dec2	0.6	0.7	0.7	0.8	0.8	0.8	0.8

A constant scaling factor over all SNRs and decoding iterations improves the Max-Log-MAP decoding significantly. The best scaling factors for different SNRs and decoding iterations are found by off-line computations. These scaling factors are obtained via simulations by choosing the scaling factors corresponding to the minimum BER.

A turbo code of rate R = 1/3, memory length $m = 3$, generator polynomial (13, 15) $_{oct}$ is simulated to obtain the best scaling factors for different SNRs and decoding iterations. Table 1 shows the best scaling factors for iterations 1 to 6 and SNR values of 0 dB to 1.5 dB. Table 2 shows the best scaling factors after 6 iterations only for different SNRs assuming a constant scaling factor for both decoders. The performance of the modified algorithm is compared with the standard Max-Log-Map and Log-Map algorithms. Fig. 3 and 4 show the BER performances of the Log-Map, the Max-Log-Map, and the modified Max-Log-Map with scaling factor 0.7 after 6 decoding iterations for interleaver lengths 5114 and 1024 respectively. A constant scaling factor (0.7) provides approximately 0.4 dB improvement over the standard Max-Log-Map algorithm at a BER of 10^{-4}. Table 3 shows BER values at E_b/N_o=1dB, for an interleaver length of 1024.

From the simulation results, it is observed that changing scaling factors for different SNRs/iterations or just for SNRs doesn't improve the decoding performance. As it is shown in Table 3, BER values for 3 different methods of scaling factor modification are almost identical. Although, changing scaling factors for SNRs (and decoding iterations) provides an improvement over the standard Max-Log-Map algorithm, this improvement is observed to be equal to the improvement obtained by a constant choice of the scaling factor. The performance difference between the Log-Map and the modified Max-Log-Map is around 0.1 dB as observed from simulations.

Table 3.BER values for each iteration at E_b/N_o=1dB (R=1/3, interleaver length=1024, generator polynomial (13, 15) $_{oct}$) *Sf: Scaling Factor

	BER				
Iteration	Log-Map	Max- Log-Map	Constant Sf*	Sf for SNRs	Sf for SNRs and Iterations
1	0.0396	0.0509	0.0467	0.0482	0.0462
2	0.0125	0.0301	0.0186	0.0239	0.0180
3	0.0034	0.0188	0.0067	0.0119	0.0064
4	0.0012	0.0132	0.0028	0.0068	0.0028
5	0.0006	0.0101	0.0016	0.0044	0.0016
6	0.0004	0.0087	0.0011	0.0034	0.0010

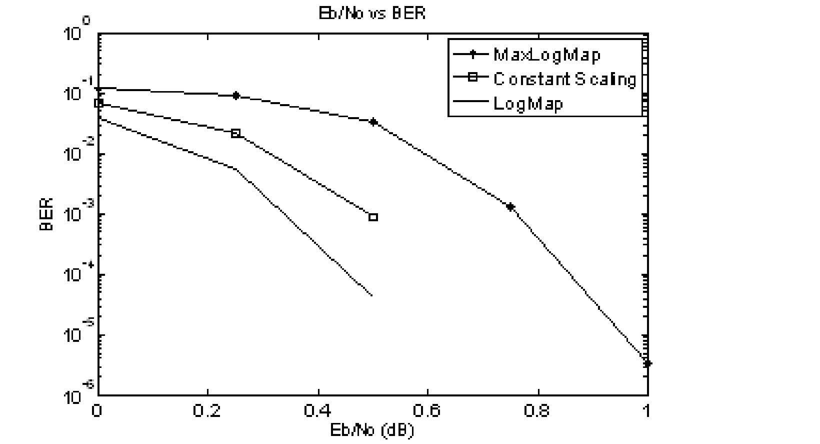

Fig. 3. BER versus E_b/N_o for Log-MAP, Max-Log-MAP and modified Max-Log-MAP with scaling factor = 0.7 (interleaver length 5114, 6 iterations).

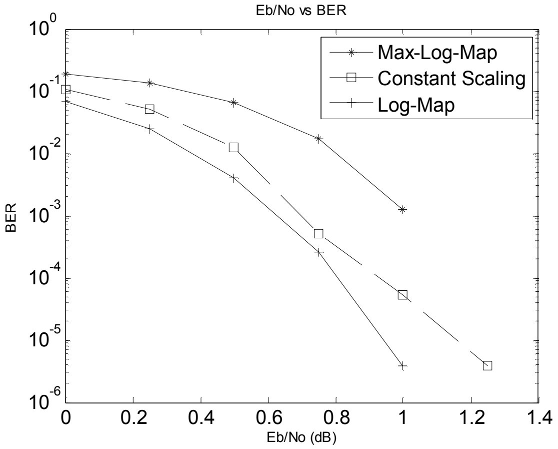

Fig. 4. BER versus E_b/N_o for Log-MAP, Max-Log-MAP and modified Max-Log-MAP with scaling factor = 0.7 (interleaver length 1024, 6 iterations).

3. Trends and Applications of Turbo Codes

Near Shannon-capacity performance of turbo codes have attracted many researchers to investigate the principles in which turbo coding is based on and to apply these principles for development of new codes. Turbo coding and iterative decoding principles have found practical applications just after their discovery [15].

An important class of binary linear block code, Low-Density Parity Check (LDPC) codes [16], or Gallager codes were originally invented in 1963 and found practical applications only after the emergence of turbo decoding principles. Turbo-like codes have been proposed to reduce the complexity of iterative turbo decoding while providing near Shannon-capacity performance [17]. Repeat Accumulate (RA) [18], Irregular Repeat Accumulate (IRA) [19], and Accumulate Repeat Accumulate (ARA) [20] codes are some of the recently invented turbo-like codes.

Turbo codes have been incorporated into many important wireless protocols from deep-space communications to mobile communications. They are used extensively in 3G mobile telephony standards. MediaFLO, terrestrial mobile television system from Qualcomm has integrated turbo codes as the error-correction algorithm. NASA missions such as Mars Reconnaissance Orbiter use turbo codes, as an alternative to Reed Solomon-Viterbi codes. The European Space Agency's first mission to the moon SMART-1, launched in September 2003, adopted turbo codes. IEEE 802.16, a wireless metropolitan network standard, also uses turbo coding. DVB-RCS, DVB-RCT, INMARSAT, EUTELSAT, and BRAN are other systems which use turbo codes as the channel coding algorithm.

Improving the coding gain obtained by channel coding enables communication systems to work at a lower power consumption and also lower cost while preserving the quality of communication. For deep space applications, 1 dB coding gain would save millions of dollars by reducing size and weight of the required antenna and equipment (like solar panels). For mobile cellular systems, power consumption of mobile handsets can be reduced and also the quality of service can be increased by increasing the possible number of users in the cell.

As turbo codes have already been adopted in important wireless communication standards like 3GPP and DVB-RCS, there are still research areas worth to investigate aiming to reduce the complexity, power dissipation and throughput of turbo decoders.

The most obvious of these is to improve the memory access of the turbo decoder. It has been shown that most of the energy in iterative decoding is consumed during memory access. Up to 70% of power saving, compared to conventional implementation, is possible by reducing the memory requirement [21]. The interleaver stage introduced at the encoder side complicates the memory addressing. New interleaver designs providing an easier memory access during iterative decoding have been proposed [15].

Longer interleaver lengths provide better error-protection. However, the latency and complexity introduced by the interleaver prevents the use of long interleaver lengths in real-time wireless applications. In addition, a complex algorithm is used to define the interleaver addresses of the turbo code interleaver in the 3GPP standard [5]. The easiest approach is to generate the addresses for a certain frame length and save them into a ROM. The maximum block size in 3GPP UMTS is 5114 which can be addressed by 13 bits. Instead of saving all address mappings into a ROM, a better approach to find interleaver addresses is to develop an interleaver address generator which is generic for all frame lengths [22].

Turbo codes have not only provided an excellent near Shannon-capacity performance but also they have inspired the development of new codes and rediscovery of some other high performance codes untouched for a long time, like LDPC codes. The massive research effort performed just after the invention of turbo codes and their rapid adaptation into many important communication standards show that the research and development in turbo codes and turbo-like codes will continue in the near future.

4. Conclusions

In this chapter, principles of turbo coding and its applications in wireless communications have been discussed. The emphasis has been given on an algorithm modification to improve the BER performance of the Max-Log-MAP algorithm which is the reduced complexity version of the Log-MAP algorithm.

The performance gap between the Max-Log-MAP and Log-MAP algorithms can be compensated by scaling the extrinsic information exchange between two constituent MAP decoders. This modification in the Max-Log-MAP algorithm can be implemented simply by multiplying the extrinsic information by a scaling factor. The modified Max-Log-MAP algorithm is simulated by choosing this scaling factor as a constant as well as choosing the best scaling factors for different SNRs and decoding iterations. Simulation results show that there is almost no performance gain when we adaptively change the scaling factor with different channel conditions and for different decoding iterations against keeping the scaling factor constant. On the other hand, a proper choice of the scaling factor provides 0.4 dB improvement for the Max-Log-MAP algorithm. This optimum constant scaling factor is found to be 0.7 from our simulations.

References

1. C.E. Shannon, A Mathematical Theory of Communication. Bell System Technical Journal, 1948. 27: p. 379-423, 623-656.
2. C. Berrou, A. Glavieux, and P. Thitimajshima, Near Shannon limit error-correcting coding and decoding: Turbo-codes. in Proc. ICC'93, 1993. 2: p. 1064 - 1070.
3. A. Burr, Turbo-codes: the ultimate error control codes? Electronics & Communication Engineering Journal, 2001. 13(4): p. 155 - 165.
4. S. Benedetto and G. Montorsi, Unveiling turbo codes: some results on parallel concatenated coding schemes. IEEE Transactions on Information Theory, 1996. 42(2): p. 409 - 428.
5. European Telecommunication Standards Institute, Universal Mobile Telecommunications System (UMTS): Multiplexing and channel coding (TDD),3GPP TS 25.222 version 7.3.0 Release 7, p. 18 - 23, May 2005.
6. Third Generation Partnership Project 2, Physical Layer Standard for cdma2000 Spread Spectrum Systems, C.S0002-D, version 2.0, p. 2.97-2.105, Sept.,2005.
7. M.C. Valenti and J. Sun, Turbo Codes. Handbook of Rf and Wireless Technologies, ed. F. Dowla. 2004: Newnes Press. 375 - 399.
8. H.R. Sadjadpour, N.J.A. Sloane, and G. Nebe, Interleaver Design for Turbo Codes. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, 2001. 19(5): p. 831-837.
9. B. Sklar, Digital Communications: Fundamentals and Applications. Second ed. Fundamentals of Turbo Codes. 2001: Prentice Hall.
10. C. Heegard and S.B. Wicker, Turbo Coding. 1 ed. 1999, Boston: Kluwer Academic Publisher.
11. M.C. Valenti and J. Sun, The UMTS Turbo code and an Efficient Decoder Implementation Suitable for Software-Defined Radios. International Journal of Wireless Information Networks, 2001. 8(4): p. 203-214.
12. J. Vogt and A. Finger, Improving the Max-Log-MAP turbo decoder. Electronics Letters, 2000(23): p. 1937 - 1939.
13. H. Claussen, H.R. Karimi, and B. Mulgrew, Improved max-log map turbo decoding using maximum mutual information combining. 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. 1: p. 424 - 428.
14. L. Papke, P. Robertson, and E. Villebrun, Improved decoding with the SOVA in a parallel concatenated (Turbo-code) scheme. IEEE International Conference on Communications, 1996. 1: p. 102 - 106.

15. B. Vucetic, et al., Recent Advances in Turbo Code Design and Theory. Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), 2007. 95(6): p. 1323 - 1344.
16. R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA : MIT Press, 1963.
17. A. Abbasfar, D. Divsalar, and K. Yao, A class of turbo-like codes with efficient and practical high-speed decoders. IEEE Military Communications Conference, MILCOM 2004, 2004. 1: p. 245 - 250.
18. D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for turbo-like codes," in Proc. Allerton Conf., 1998, pp. 201–210.
19. H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in Proc. 2nd Int. Symp. Turbo Codes, 2000, pp. 1–8.
20. A. Abbasfar, D. Divsalar, and K. Yao, Accumulate-Repeat-Accumulate Codes. IEEE Transactions on Communications, 2007. 55.
21. G. Masera, et al., VLSI architectures for turbo codes. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1999. 7(3): p. 369 - 379.
22. I. Ahmed and T. Arslan, A low energy VLSI design of random block interleaver for 3GPP turbo decoding. 2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings., 2006: p. 4.