VITERBI ALGORITHM FOR ITERATIVE DECODING OF PARALLEL CONCATENATED CONVOLUTIONAL CODES

Bogdan Cristea

LMS Romania SRL
I. Slavici Street, No. 15A, 500398, Braşov, Romania
phone: +40268310101
email: bogdan.cristea@lmsintl.com
web: http://www.lmsintl.com

"Transilvania" University of Braşov
Eroilor Blvd., No. 29, 500036, Braşov, Romania
phone: +40748644800
email: cristeab@ieee.org
web: http://www.unitbv.ro

ABSTRACT

A low complexity Soft-Input Soft-Output (SISO) module, based on Viterbi algorithm, is proposed. This module is suitable for turbo decoding of Parallel Concatenated Convolutional Codes (PCCCs). The interest of this approach resides on the fact that the Viterbi algorithm is used unmodified, still accepting as its inputs soft information. The hard output of the Viterbi algorithm is used to compute the extrinsic information based on the knowledge of systematic and *a priori* information. Also, a set of scaling factors is used to improve the quality of the extrinsic information. The scaling factors are obtained based on a semi-analytical method. The convergence properties and performance of the proposed turbo decoder are evaluated using EXtrinsic Information Transfer (EXIT) charts and Bit Error Rate (BER) simulations.

1. INTRODUCTION

PCCCs are a well known topic of research and have been selected as channel coding technique in several wireless communications standards (e.g. Long Term Evolution [1]). The classical approach for iterative decoding of PCCCs uses the log Maximum A Posteriori (MAP) algorithm [2], whose complexity is much larger than the complexity of the Viterbi algorithm [11]. The increase in complexity of the MAP algorithm allows to compute soft (extrinsic) information for each data bit which is exchanged between two SISO modules in an iterative fashion [3]. In order to further lower the complexity of the MAP algorithm, several approaches have been proposed: max log MAP algorithm, threshold MAP algorithm [5], etc. All these variants have in common the fact that the original structure of the MAP algorithm is kept (with forward and backward recursions), but the number of needed operations and/or trellis states is reduced.

A different approach for low complexity turbo decoding is represented by the Soft Output Viterbi Algorithm (SOVA) [6]. In SOVA the Viterbi algorithm is modified so that the extrinsic information is obtained from metric difference between the surviving and competitor paths. Further, an update mechanism is used to compute the extrinsic information for surviving paths. This allows to obtain soft information from the Viterbi algorithm at the expense of a higher complexity than the original Viterbi algorithm. A modified version of SOVA has been shown to be equivalent to max log MAP algorithm [4]. Further improvements allowed to obtain performace close to log MAP algorithm when the target BER is moderately low [12].

In [7] a novel low complexity turbo decoder is proposed, based on an unmodified Viterbi algorithm. The novelty of this approach comes from the fact that the soft information, needed by the next stage of the turbo receiver, is computed from the hard output of the Viterbi algorithm. Also, both intrinsic information of systematic bits and *a priori* information of informational bits are used to compute an extrinsic information of data bits to be passed to the next stage of the iterative decoding process.

This paper follows and extends the results presented in [7] by providing a theoretical framework for the proposed SISO Viterbi algorithm. Also, a semi-analytical method for computing some scaling coefficients, used to compute the extrinsic information, is proposed based on EXIT charts. The rest of the paper is organized as follows: section 2 describes the SISO module based on Viterbi algorithm, followed by the presentation of some simulation results in section 3. Section 4 ends the paper with conclusions and directions of further research.

2. SOFT-INPUT SOFT-OUTPUT MODULE BASED ON VITERBI ALGORITHM

This section presents a low complexity algorithm based on Viterbi algorithm used for turbo decoding of PCCCs.

The encoder is represented by two Recursive and Systematic Convolutional Codes (RSCCs) separated by an interleaver [3]. Let u_n, $n \in \{0, 1, \dots, N-1\}$, be the sequence of informational bits. The sequence of informational bits enters unmodified into the upper RSCC, while the input of the lower RSCC is represented by the interleaved sequence, with N the interleaver length. The output of the encoder is represented by the systematic bits, $v_n^{(s)} = u_n$ and by the parity bits from both RSCCs: $v_n^{(p)_{up}}$ and $v_n^{(p)_{low}}$. Then, the encoded bits are Binary Phase Shift Keying (BPSK) modulated and sent into an Aditive White Gaussian Noise (AWGN) channel:

$$y_n = (1 - 2v_n) + \eta_n \tag{1}$$

where v_n is the coded bit (systematic or parity bit) and η_n is an AWGN with zero mean and variance σ^2.

The turbo decoder uses two SISO modules, exchanging soft information along several iterations (Fig. 1) [3]. The input to the upper SISO module is represented by intrinsic information of systematic bits defined as:

$$L(v_n^{(s)}; I) = \ln \frac{p(y_n^{(s)}/u_n = 1)}{p(y_n^{(s)}/u_n = 0)} \tag{2}$$

$$= -\frac{2}{\sigma^2} y_n^{(s)} \tag{3}$$

© EURASIP, 2010 ISSN 2076-1465

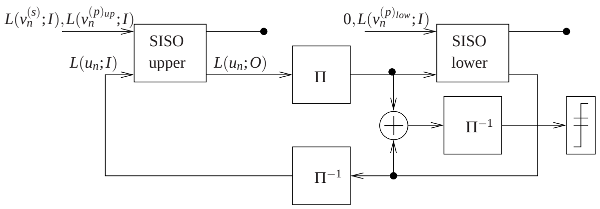


Figure 1: Turbo decoder for PCCC

where $p(y_n^{(s)}/u_n = 1)$ is the conditional probability density function of the received symbol, $y_n^{(s)}$, knowing the systematic bit $u_n = 1$. The second equality comes from the fact that the additive noise has a Gaussian distribution. Similarly, it can be shown that the intrinsic information of parity bits from upper RSCC is: $L(v_n^{(p)_{up}};I) = -\frac{2}{\sigma^2} y_n^{(p)_{up}}$. This represents the second input of the upper SISO module. The third input is represented by the *a priori* information of informational bits defined as:

$$L(u_n;I) = \ln \frac{P(u_n = 1)}{P(u_n = 0)} \tag{4}$$

where $P(u_n = 1)$ is the *a priori* probability of the informational bit $u_n = 1$. This soft information is computed by the lower SISO module during the previous iteration.

The output of the upper SISO module is the extrinsic information of informational bits, $L(u_n;O)$, used, after interleaving, as a priori information at the input of the lower SISO module.

Also, the lower SISO module has as input the intrinsic information of parity bits from the lower RSCC, $L(v_n^{(p)_{low}};I)$. The lower SISO module produces an extrinsic information of informational bits used as *a priori* information in the next iteration by the upper SISO module.

In the following only the upper SISO module shall be discussed, since the lower SISO module has exactly the same functionality (except that no intrinsic information of systematic bits is available at its input). Also, in order to simplify the notation, the superscript ^{up} shall be dropped.

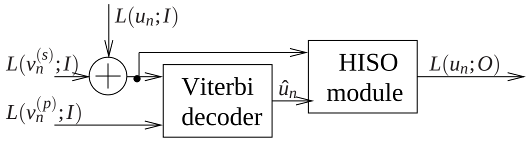


Figure 2: SISO module based on Viterbi algorithm for RSCC

The proposed SISO module is based on a Viterbi decoder, accepting as its inputs soft information and a Hard-Input Soft-Output (HISO) module, having at one of its inputs the hard output of the Viterbi decoder, $\hat{u}_n$, (Fig. 2). These two modules shall be detailed in the following subsections.

2.1 Viterbi decoder with soft information as its inputs

In order to use the *a priori* information of informational bits, the metric used for path selection by the Viterbi algorithm is defined as [9]:

$$\begin{aligned} P(\mathbf{y},\mathbf{v}) &= p(\mathbf{y}/\mathbf{v})P(\mathbf{v}) && (5) \\ &= p(\mathbf{y}/\mathbf{v})P(\mathbf{u}) && (6) \end{aligned}$$

where $\mathbf{y} = \left[y_0^{(s)} \quad y_0^{(p)} \quad y_1^{(s)} \quad y_1^{(p)} \quad \cdots \quad y_{N-1}^{(s)} y_{N-1}^{(p)} \right]$ is the sequence of received symbols, $\mathbf{v}$ is the sequence of systematic and parity bits and $\mathbf{u}$ is the sequence of informational bits. The second equality comes from the fact that the probability of the sequence at the output of the encoder, $P(\mathbf{v})$, equals the probability of the sequence at the input of the encoder, $P(\mathbf{u})$.

Using in (6) the fact that the received symbols are independent (since the additive noise is white) and based on the expressions of intrinsic (3) and *a priori* (4) information, the metric used by the Viterbi algorithm can be rewritten as:

$$\begin{aligned} &P(\mathbf{y},\mathbf{v}) = \Pi_{n=0}^{N-1} K_n \\ &\exp\left(u_n (L(v_n^{(s)};I) + L(u_n;I)) + v_n^{(p)} L(v_n^{(p)};I) \right) \end{aligned} \tag{7}$$

where K_n is a constant depending only on the current instant, n. Note that, since the transition between trellis states depends on the informational (systematic) and parity bits, the path selection is influenced only by the argument of the exponential function (7). Thus, the Viterbi decoder can be fed directly with soft information (the intrinsic information of systematic and parity bits). Further, since the encoder is systematic, the *a priori* information can be added to the intrinsic information of systematic bits before being fed to the Viterbi decoder (Fig. 2).

So, when the encoder is systematic, the classical Viterbi decoder [11] can be used unmodified, still accepting as its input soft information. The output of the Viterbi decoder is the hard decoded sequence of informational bits, used by the HISO module in order to compute the extrinsic information of informational bits.

2.2 Hard-Input Soft-Output module

The algorithm used by the HISO module represents the main contribution of our paper. This algorithm follows and extends the ideas presented in [7].

In order to compute the extrinsic information of informational bits, $L(u_n;O)$, from the hard output of the Viterbi algorithm, the following expression is used:

$$L(u_n;O) = \Lambda(u_n) - L(u_n;I) \tag{8}$$

where $\Lambda(u_n)$ is the Logarithm of Likelihood Ratio (LLR) of informational bits defined as

$$\begin{aligned} \Lambda(u_n) &= \ln \frac{P(u_n = 1/\hat{u}_n)}{P(u_n = 0/\hat{u}_n)} && (9) \\ &= (2\hat{u}_n - 1) \ln \frac{1 - P_n(e)}{P_n(e)} && (10) \end{aligned}$$

and $P_n(e)$ is the Bit Error Probability (BEP), defined as

$$\begin{aligned} P_n(e) =& P(\hat{u}_n = 1/u_n = 0)P(u_n = 0) + \\ & P(\hat{u}_n = 0/u_n = 1)P(u_n = 1) \end{aligned} \tag{11}$$

Expression (8) can be proven using (9), (4) and the Bayes' rule.

In order to prove (10) the BEP is rewritten as (11):

$$P_n(e) = \begin{cases} P(\hat{u}_n = 0/u_n = 1)P(u_n = 1) & \text{if } \hat{u}_n = 0 \\ P(\hat{u}_n = 1/u_n = 0)P(u_n = 0) & \text{if } \hat{u}_n = 1 \end{cases} \tag{12}$$

Using Bayes' rule, (9) can be rewritten as:

$$\Lambda(u_n) = \ln \frac{P(\hat{u}_n/u_n=1)P(u_n=1)}{P(\hat{u}_n) - P(\hat{u}_n/u_n=1)P(u_n=1)} \tag{13}$$

If $\hat{u}_n = 0$, with (12) the above equation becomes:

$$\Lambda(u_n) = \ln \frac{P_n(e)}{1-P_n(e)} \tag{14}$$

Using similar develoments for $\hat{u}_n = 1$, the equation (10) is completely proven.

Note that a result similar to (10) has been obtained in [7], but instead of using the BEP at each discrete instant, $P_n(e)$, the BEP for the entire interleaver is used. However, in [7] no formal proof is provided for this result.

Thus, using the hard output from the Viterbi algorithm, $\hat{u}_n$, and the knowledge of the BEP, $P_n(e)$, the extrinsic information of informational bits, $L(u_n;O)$, can be computed by the HISO module.

However, the BEP is not available at the decoder side and some online approximation is needed for BEP computation. Since there are several methods for online BEP estimation for the entire interleaver [8], we choose to approximate the BEP at each instant by the BEP for the entire interleaver:

$$P_n(e) \approx P(e) \tag{15}$$

where $n \in \{0,1,\ldots,N-1\}$. Note that this approach allows to have different BEPs for different interleavers.

Also, note that with (15) the LLR of informational bits (10) has a constant absolute value for the entire interleaver length. This effect will worsen the performance of the turbo decoder due to the poor quality of the extrinsic information (8). A similar effect can be observed if the LLR of informational bits, obtained using the MAP algorithm, is limited using some threshold value. Thus, in order to improve the quality of the soft information, as provided by the HISO module, the following approximation is used (10):

$$\Lambda(u_n) \approx \begin{cases} c(2\hat{u}_n-1)\ln\frac{1-P(e)}{P(e)}, H(L(u_n;I)+L(v_n^{(s)};I)) = \hat{u}_n \\ e(2\hat{u}_n-1)\ln\frac{1-P(e)}{P(e)}, H(L(u_n;I)+L(v_n^{(s)};I)) \neq \hat{u}_n \end{cases} \tag{16}$$

where the scaling factors c and e are positive real values and $H(x)$ is the Heaviside step function: $H(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$. The approximation (16) has been proposed in [7] and allows to introduce additional information into the estimated LLR of informational bits based on the difference between the estimated bits, $\hat{u}_n$, and the soft information of informational bits, $L(u_n;I)+L(v_n^{(s)};I)$.

In order to approximate the BEP for the entire interleaver, $P(e)$, the simplest approach uses the flipped bit error rate [7]:

$$P(e) = \frac{1}{N}\sum_{n=0}^{N-1}\left|H(L(u_n;I)+L(v_n^{(s)};I)) - \hat{u}_n\right| \tag{17}$$

More complex approximations for $P(e)$ can be found in [8]. Our simulations showed that the best results can be obtained using (17), so in the following only this approximation is used.

Further, the scaling factors, c and e, can be computed as the ratio between the expected absolute value of the true LLR, as computed by the MAP algorithm, $\mathrm{E}[|\Lambda(u_n)|]$, and the absolute value of the approximated LLR (16):

$$c = \frac{\mathrm{E}[|\Lambda(u_n)|]}{\left|\ln\frac{1-P(e)}{P(e)}\right|} \tag{18}$$

where the expectation, $\mathrm{E}[|\Lambda(u_n)|]$, is taken over the set of indices n for which $H(L(u_n;I)+L(v_n^{(s)};I)) = \hat{u}_n$ (for matching bits). In a similar manner, the scaling factor e is computed over the set of indices for which $H(L(u_n;I)+L(v_n^{(s)};I)) \neq \hat{u}_n$ (for nonmatching bits).

The method represented by (18) is briefly mentioned in [7], but is not developed and instead the scaling factors are obtained empirically. Thus, the method for computing the scaling factors can be seen as another contribution of our paper.

It is worth to emphasize that the scaling factors computed with (18) depend not only on the Signal to Noise Ratio (SNR) at the input of the turbo decoder, but also on the iteration number in the turbo decoder. However, by simulation it can be shown that a single set of scaling factors is enough to achieve the best performance of the turbo decoder with the proposed SISO module.

3. SIMULATION RESULTS

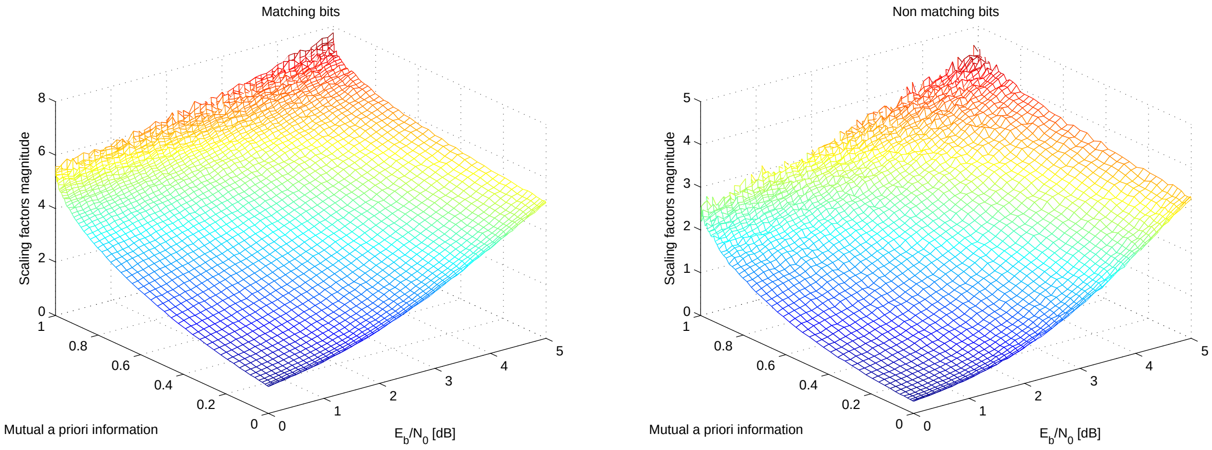
A PCCC of coding rate $R = 1/3$ is simulated. The two RSCCs are identical and have generator polynomials $[1\ 15/13]$ in octal form .

In order to compute the scaling factors used in (16) we use the ratio between the true LLR, as provided by the MAP algorithm, and the LLR computed using the proposed SISO module (18). The magnitude of the scaling factors as a function of both mutual *a priori* information [10] and SNR, $\frac{E_b}{N_0}$, is shown in Fig. 3a and 3b. The scaling factors should have increased values when the mutual *a priori* information (or the number of iterations) and the SNR increase.

Further, EXIT charts [10] are used in order to estimate the convergence threshold, expressed as a SNR, at which the iterative decoding process converges (Fig. 4). The turbo decoder uses the proposed SISO module with varying scaling factors as a function of both mutual *a priori* information and SNR. The EXIT chart is obtained from the transfer characteristics of the upper and lower SISO modules. The upper SISO module uses at its input the received systematic and parity bits and the *a priori* information, while the lower SISO module uses only the received parity bits and the *a priori* information. As proposed in [10], the *a priori* information corresponds to systematic bits and is represented by the realisations of an independent Gaussian variable.

From Fig. 4 it can be seen that at about $\frac{E_b}{N_0} = 1.4\ dB$ there is a tunnel between the transfer characteristics of the two SISO modules, allowing the convergence of the iterative decoding algorithm. This represents the convergence threshold for the proposed turbo decoder and can be used to select a single set of scaling factors.

Based on this result, a single set of scaling factors can be selected at zero mutual *a priori* information. This choice is justified since a zero mutual *a priori* information corresponds



(a)

(b)

Figure 3: Scaling factors magnitude as a function of mutual *a priori* information and SNR for (a)matching bits (b)non matching bits

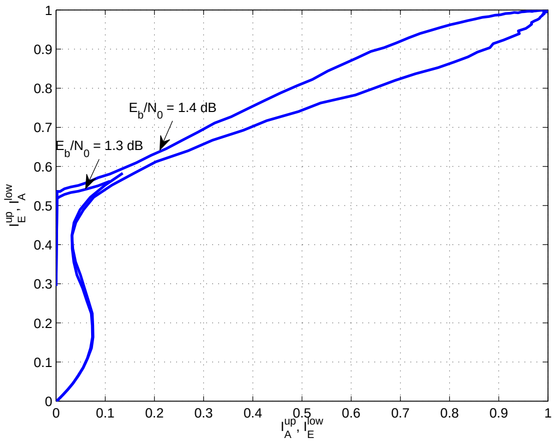


Figure 4: EXIT diagram for different values of SNR

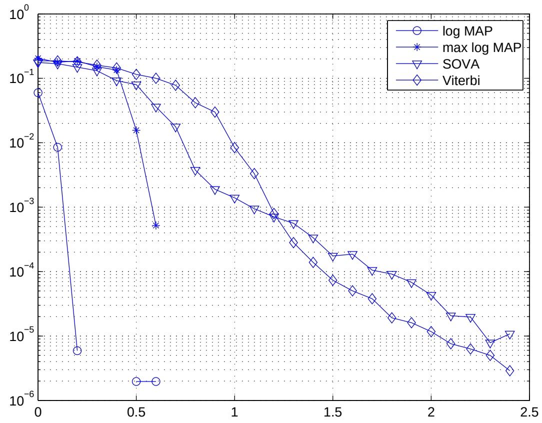


Figure 5: Performance of the turbo decoder with different decoding algorithms

to the beginning of the iterative decoding process, while the convergence threshold ensures the convergence of the turbo decoder. With these choices, the following scaling factors are obtained:

$$c = 1.6114 \tag{19}$$

$$e = 0.5859 \tag{20}$$

The performance of the turbo decoder with the above scaling factors is presented in Fig. 5. Also, for comparison purposes, the performance of log MAP, max log MAP and SOVA are also depicted. The SOVA is used unmodified, thus the poor performance at a SNR above 0.8 dB.

The SISO module based on Viterbi algorithm has the lowest complexity among the four algorithms, but this comes with a performance penality at low SNR. Note also from Fig. 5 that the predicted convergence threshold, $\frac{E_b}{N_0} = 1.4\ dB$, is confirmed by the BER courve for SISO module based on Viterbi algorithm. This shows that a single set of scaling coefficients is enough to achieve good performance of the turbo decoder at the predicted convergence threshold.

It is also worth to emphasize that the results presented in Fig. 5 were obtained with an interleaver length $N = 2^{14}$ bits, while the EXIT charts from Fig. 4 have been obtained with an interleaver length $N = 10^5$. Thus, small performance improvement can be obtained, especially at low SNR, with a careful choice of the scaling factors, depending on the interleaver length [7].

So, the scaling factors can be obtained using a semi-analytical method by computing the ratio between the true LLR, as provided by the MAP algorithm, and the LLR computed by the proposed HISO module. Only a single set of scaling factors is needed, regardless of the current iteration (mutual *a priori* information) and SNR.

4. CONCLUSION

A low complexity SISO module based on Viterbi algorithm has been presented, suitable for turbo decoding of PCCCs. Since the encoders are systematic, the classical Viterbi algorithm can be used with no modifications, still accepting at its inputs soft information. Further, it has been shown how the extrinsic information can be obtained from the hard output of the Viterbi algorithm, using also the knowledge of systematic and *a priori* information. By approximating the BEP at the current instant with the BEP of the entire interleaver, a practical method for extrinsic information computation has been provided. In order to improve the quality of the extrinsic information, two scaling factors have been used. A semi-analytical method has been proposed for computing the scaling factors, based on the ratio between the expected absolute value of the true LLR and the absolute value of the approximated LLR. It has also been shown that a single set of scaling factors is enough to achieve the best performance of the proposed turbo decoder, regardless of the iteration number and SNR.

However, the performance of the proposed turbo decoder is far from the performance of the best known turbo decoder, based on the log MAP algorithm. Thus, future reseach should focus on a hybrid turbo decoder, using both Viterbi and log MAP algorithms. Also, generalizations of the hybrid turbo decoder for other types of encoders (e.g. serial concatenations) is another direction of future research.

5. ACKNOWLEDGMENT

The author would like to thank LMS Romania SRL, Braşov, Romania, for their financial support.

REFERENCES

[1] 3GPP. 3rd generation partnership project; technical specification group radio access network; multiplexing and channel coding (fdd). Technical report, 3GPP, Dec. 2009. 3GPP TS 25.212 V9.1.0 (2009-12).

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, pages 284–287, Mar. 1974.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. A soft-input soft-output APP module for iterative decoding of concatenated codes. *IEEE Communications Letters*, 1(1):22–24, Jan. 1997.

[4] M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer. On the equivalence between SOVA and max-log-MAP decodings. *IEEE Communications Letters*, 2(5):137–139, May 1998.

[5] V. Franz and J. B. Anderson. Concatenated decoding with a reduced-search BCJR algorithm. *IEEE Journal on Selected Areas in Communications*, 16(2), Feb. 1998.

[6] J. Hagenauer and P. Hoeher. A Viterbi algorithm with soft-decision outputs and its applications. *Global Telecommunications Conference*, 1989.

[7] M. Kerner and O. Amrani. Iterative decoding using optimum Soft Input - Hard Output module. *IEEE Transactions on Communications*, 57(7):1881–1885, July 2009.

[8] N. Letzepis and A. Grant. Bit error rate estimation for turbo decoding. *IEEE Transactions on Communications*, 57(3):585–590, March 2009.

[9] S. Lin and D. J. Costello Jr. *Error Control Coding*. Pearson Prentice Hall, second edition, 2004.

[10] S. ten Brink. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Transactions on Communications*, 49(10):1727–1737, Oct. 2001.

[11] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, Apr 1967.

[12] Z. Wang and K. K. Parhi. High performance, high throughput turbo/SOVA decoder design. *IEEE Transactions on Communications*, 51(4):570–579, April 2003.