TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Efficient stopping rule for turbo decoders

Citation for published version (APA):
Rovini, M., & Martinez, A. (2006). Efficient stopping rule for turbo decoders. *Electronics Letters*, *42*(4), 235-236. https://doi.org/10.1049/el:20064403

DOI:
10.1049/el:20064403

Document status and date:
Published: 01/01/2006

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 02. Jun. 2023

Efficient stopping rule for turbo decoders

M. Rovini and A. Martinez

A new stopping rule for turbo codes, the input-output consistency (IOC) check is presented. It is based on an extended maximum *a posteriori* algorithm that also outputs (hard) extrinsic estimates on the coded bits. In parallel, the hard decisions of the info bits are re-encoded and if the two sequences coincide, iterative decoding is stopped. Remarkably, IOC beats other known stopping rules in terms of error rate and convergence speed, closely approaching ideal 'Genie'-aided decoding.

Introduction: Turbo codes [1] achieve low error probabilities at rates very close to Shannon's capacity limit thanks to iterative decoding. In practical implementations, decoders run for a fixed number of iterations: this number is determined on the basis of a worst-case analysis, although only a few frames need the entire number of iterations. An early stopping criterion would not only save unnecessary elaborations and reduce decoding latency, but increase the decoder throughput as well, or even reduce the chip complexity when the decoder architecture performs statistical multiplexing [2].

Several stopping rules have been proposed in the past [3–6]. Among these, the cross-entropy (CE) criterion [3, 4] processes the decoder soft output, while its simpler versions sign change ratio (SCR) and hard-decision aided (HDA) operate on hard decisions [4]. SCR counts the changes in sign of the estimated sequence between two consecutive iterations and declares convergence when it is less than a certain threshold. Similarly, HDA terminates the iterative decoding when the hard decisions of two consecutive iterations coincide over the entire frame. A variation of SCR [5] monitors the Hamming distance of the hard decisions over two iterations to detect both early convergence and non-convergence. In [6] the mean of the absolute log-likelihood ratio (LLR) of the info bits is monitored and decoding stops if it is greater than a preset threshold.

In this Letter we present a new stopping criterion working with hard decisions and suitable for any generic iterative decoder. Crucial to this rule is the availability at the decoder of the maximum *a posteriori* (MAP) estimate of the channel code bits.

Extended MAP decoding: Turbo decoding is performed by two MAP decoders [1, 7], which, fed with the channel observations $\mathbf{r}$, iteratively refine the estimate of the LLR of the information bits u_k, defined as $L(u_k) \equiv \log(P(u_k=+1|\mathbf{r})/P(u_k=-1|\mathbf{r}))$, $k=0,1,\ldots,K-1$. In addition to the estimates of the information bits, the decoder must also calculate the *a posteriori* LLR of the coded bits c_n, denoted by $L_\beta(c_n)$. With a notation similar to [7], this is computed in the log-domain as

$$L_\beta(c_n) = \log\left(\frac{\sum_{c^+} \exp\{\alpha[s^S(e_k)] + \pi_\alpha[u(e_k)] + \beta[s^E(e_k)]\}}{\sum_{c^+} \exp\{\alpha[s^S(e_k)] + \pi_\alpha[u(e_k)] + \beta[s^E(e_k)]\}}\right) \quad (1)$$

where C^+ and C^- are the set of trellis transitions e_k with the coded bit $c_n(e_k)$ taking the values $+1$ and -1, respectively. With a notation borrowed from [7], we refer to the input *a priori* probability $P(x;\ I)$ ($\pi(x;\ I)$ in the log-domain) and to the output extrinsic probability $P(x;\ O)$($\pi(x;\ O)$ in the log-domain) of a random variable x, as α- and β-probability, $P_\alpha(x)$($\pi_\alpha(x)$) and $P_\beta(x)$($\pi_\beta(x)$), respectively. This is in line with the notation usually reserved for the state metrics, for which it holds also.

Note that (1) is not usually implemented in turbo decoders, where the two units only compute and exchange the estimates on the information bits, often referred to as extrinsic information. We refer to [7] for further details about (1).

Input-output consistency stopping rule: The proposed stopping rule can be seen as a generalisation of that in use with low-density parity-check (LDPC) codes [8]. An LDPC decoder may be instructed to halt when the messages coming from the bit nodes satisfy the whole set of (even) parity checks defining the code. This operation corresponds to re-encode the current estimates and check if they form a valid codeword, which is consistent with the LDPC encoder.

In analogy, our criterion performs a consistency check between the input and output bits of the decoder, and this check is used as a proxy for the valid estimation. Fig. 1 shows the block diagram of a generic iterative decoder equipped with the new input-output consistency (IOC) check.

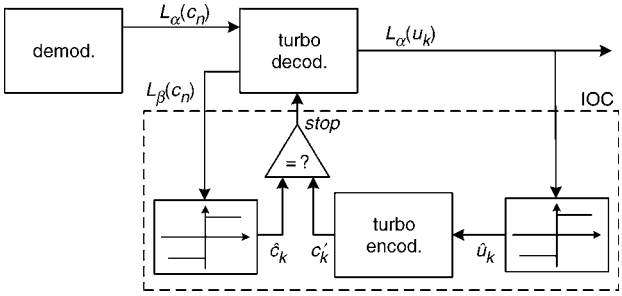

Fig. 1 *Input-output consistency check principle*

The hard-decision estimates $\hat{u}_k$, $k=0,\ldots,K-1$, output by the decoder are first re-encoded into the coded sequence c'_n and then compared with the sequence $\hat{c}_n$, $n=0,\ldots,N-1$ also output by the decoder. The latter are the hard decisions taken on the extrinsic *a posteriori* LLR $L_\beta(c_n)$ of the channel code bits c_n, $n=0,\ldots,N-1$. If the two sequences of re-encoded bits c'_n and channel extrinsic hard decisions $\tilde{C}_n$ coincide, then the decoder is considered to be in a stable state, and early convergence is declared.

Complexity estimate: We now give a conservative estimate of the overhead incurred in a hardware implementation of the IOC criterion. We use as our case study the UMTS turbo code, made up of two identical recursive systematic convolutional (RSC) codes with rate 1/2, 8 states and generators $[1,\ 15/13]_{oct}$.

The comparison of the two coded sequences only needs a single RSC encoder with no additional memories. At each half iteration, the decoded data $\hat{u}_k$ are re-encoded by the RSC and its outputs compared with $\hat{c}_n$ 'on-the-fly'. A single flag indicates whether the two coded sequences are the same at the end of each semi-iteration. This process is repeated in the next semi-iteration. If two consecutive semi-iterations have no errors, decoding stops.

Extra hardware is required for the computation of $L_\beta(C_n)$. This is upper bounded by the complexity of an extra LLR circuit for any parity bit, as the one used for the LLR on the extrinsic information $L_\alpha(u_k)$. The systematic bit s_n does not appear in this count, being its LLR easily obtained from the extrinsic information itself ($L_\beta(s_n) = L_\alpha(u_k) - L_\alpha(s_n) + L_\beta(u_k)$).

An LLR processor roughly counts $2(M-1)max^*$ and $4M+1$ sums, compared to $Mmax^*$ and M sums of a recursion unit (RU) [9], with M the number of states. Ignoring the impact of additions, which can be reused among units, and considering 3 RU as detailed in [9], the presence of one more LLR processor would bring the overall count of max^* to $7M-4$. For the UMTS turbo decoder, this means an increase in complexity of 37%. This number is a pessimistic estimate owing to the possibilities for hardware reuse among components. Furthermore, the impact of the state memory on the final complexity is neglected, which would reduce the overhead caused by the additional LLR circuit. Finally, in some applications, an extended MAP decoder may already be present, such as when iterations are performed over the demodulator/demapper at the channel output or with iterative removal of inter-symbol interference.

Performance with UMTS turbo code: As a test of the performance with IOC, we have applied it to the UMTS turbo code mentioned above. Fig. 2 shows the frame error rate (FER) and the average number of iterations for an UMTS code with rate 1/3, information block size $K=1504$ and BPSK modulation. In particular, IOC is compared to the following five alternatives: CE using two values of the convergence threshold, $\tau_1=0.0001$ (1) and $\tau_2=0.1$ (2), its approximations HDA (3) and SCR (4), and the Genie-aided stopping rule (5), which represents the optimal reference. To further reduce the number of operations, all rules are checked every half iteration, which can be done with no extra cost for IOC and CE. This is not the case of HDA and SCR, where extra hard-bit buffers are needed.

HDA and SCR declare convergence earlier than CE, but also experience a significant number of undetected errors, moving up the flattening in the FER curve. In opposition, IOC and CE with τ_1 stay

very close to the Genie, with only few undetected errors. In terms of speed however, IOC outperforms CE: on average they are 0.8 and 1.5 iterations above the Genie reference, respectively.

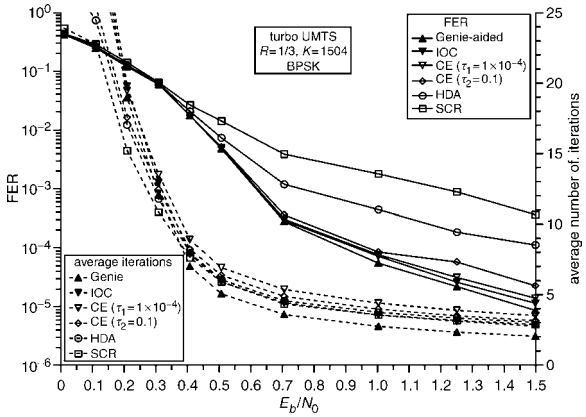

Fig. 2 *FER (right axes) and convergence speed (left axes) of IOC in comparison with others rules*

Performance of IOC at higher code rates is reported in Fig. 3. Particularly, we compared IOC with CE for code rate 1/2 and 3/4, block size 1504 and BPSK modulation. As a result of the previous analysis, the value τ_1 is used for CE. While no noticeable differences are shown in the FER curves, the convergence speed of IOC is confirmed to be around 0.8 and 0.5 iterations faster than CE at the two rates. Comparing Figs. 2 and 3, the result of puncturing is to reduce the number of undetected errors of IOC, which behaves exactly the same as the Genie in terms of FER.

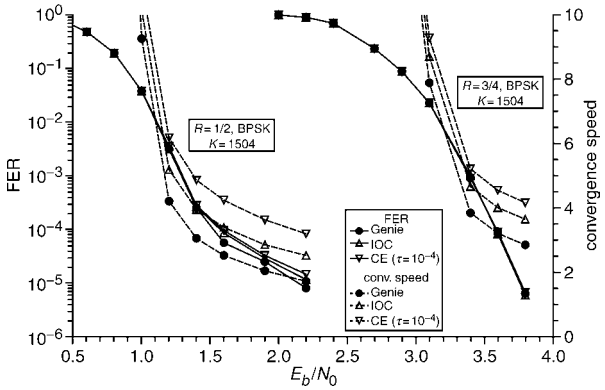

Fig. 3 *FER (right axes) and convergence speed (left axes) at rate 1/2 and 3/4 with BPSK modulation*

Conclusion: In this Letter we have presented the input-output consistency (IOC) check. This criterion allows the early stopping of the decoding operations when the estimates of the information bits output by the decoder coincide, after re-encoding, with the hard-decision estimates of the channel code bits. To adequately support IOC, the constituent decoders must also compute the extrinsic APP probabilities of the coded bits. These may be already available when iterations include the demodulator/demapper at the channel output. Computer simulations with a UMTS turbo code with rate 1/3 and $K = 1504$ showed that IOC outperforms CE in addition to its approximations HDA and SCR: it introduces fewer undetected errors, and its convergence speed is 0.5–0.8 iterations higher. Furthermore, undetected errors decrease with puncturing at higher rates.

Acknowledgments: We thank J. Sayir for inspiring discussions.

© IEE 2006
21 December 2005
Electronics Letters online no: 20064403
doi: 10.1049/el:20064403

M. Rovini (*Department of Information Engineering, University of Pisa, via G. Caruso, I-56100, Pisa, Italy*)

E-mail: massimo.rovini@iet.unipi.it

A. Martinez (*Department of Electrical Engineering, Technische Universiteit Eindhoven, 5600MB Den Dolech 2, Eindhoven, The Netherlands*)

References

1 Berrou, C., Glavieux, A., and Thitimajshima, P.: 'Near Shannon limit error-correcting coding and decoding: turbo-codes'. Proc. ICC 93, Geneva, Switzerland, 1993, pp. 1064–1070
2 Martinez, A., and Rovini, M.: 'Iterative decoders based on statistical multiplexing'. Proc. 3rd Int. Symp. on Turbo Codes and Related Topics, Brest, France, 2003, pp. 423–426
3 Hagenuaer, J., Offer, E., and Papke, L.: 'Iterative decoding of binary block and convolutional codes', *IEEE Trans. Inf. Theory*, 1996, **42**, (2), pp. 429–445
4 Shao, R.Y., Lin, S., and Fossorier, M.P.C.: 'Two simple stopping criteria for turbo codes', *IEEE Trans. Commun.*, 1999, **47**, (8), pp. 1117–1120
5 Taffin, A.: 'Generalised stopping criterion for iterative decoders', *Electron. Lett.*, 2003, **39**, pp. 993–994
6 Zhai, F., and Fair, I.J.: 'Techniques for early stopping and error detection in turbo decoding', *IEEE Trans. Commun.*, 2003, **51**, pp. 1617–1623
7 Benedetto, S., Divsalar, D., Montorsi, G., and Pollara, F.: 'Soft-input soft-output modules for the construction and distributed iterative decoding of code network', *Eur. Trans. Telecommun.*, 1998, **9**, (2), pp. 155–172
8 Gallager, R.: 'Low-density parity-check codes monograph' (MIT Press, 1963)
9 Boutillon, E., Gross, W., and Gulak, G.: 'VLSI architectures for the MAP algorithm', *IEEE Trans. Commun.*, 2003, **51**, (2), pp. 175–186