Efficient stopping rule for turbo decoders

Rovini, M.; Martinez Vicente, A.

Published in:
Electronics Letters

DOI:
10.1049/el:20064403

Published: 01/01/2006

Citation for published version (APA):
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 r, iteratively refine the estimate of the LLR of the information bits $u_k$, defined as $L(u_k) \equiv \log(p(u_k = +1|r)/p(u_k = -1|r))$, $k = 0,1, \ldots ,K - 1$. 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 $L(c_n)$. With a notation similar to [7], this is computed in the log-domain as

$$L(c_n) = \log \left( \sum_{e_n} \exp \left[ 2 \log \left( \frac{P(\bar{e}_n)}{P(e_n)} \right) + \beta \left( \frac{P(\bar{e}_n)}{P(e_n)} \right) \right] \right)$$

(1)

where $C^+$ and $C^-$ are the set of trellis transitions $e_n$ with the coded bit $c_n$ taking the values $+1$ and $-1$, respectively. With a notation borrowed from [7], we refer to the input a priori probability $P(x_k)$; $\pi(c_k)$ in the log-domain) and to the output extrinsic probability $P(x_k; \theta)$, $O(\pi(x_k; \theta))$ in the log-domain) of a random variable $x_k$ as $\alpha$- and $\beta$-probability, $P(x_k|\pi(x_k))$ and $P(\pi(x_k)|x_k)$, 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 in low-density parity-check (LDPC) codes [5]. 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.

**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 $L(c_n)$. 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 $L(u_k)$. The systematic bit $s_n$ does not appear in this count, being its LLR easily obtained from the extrinsic information itself ($L(s_n) = L(u_n) - L(c_n) - L(s_n)$).

An LLR processor roughly counts $2(M-1)\max^* + 4M+1$ sums, compared to $M\max^*$ 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, $t_1 = 0.0001$ (1) and $t_2 = 0.1$ (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 $t_1$ 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 other 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 $t_1$ 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

Electronics Letters online no: 20064403
doi: 10.1049/el:20064403

M. Rovini (Department of Information Engineering, University of Pisa, via G. Caruso, 1-56100, Pisa, Italy)
E-mail: massimo.rovini@iet.unipi.it

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

References


