Principles and Comparison of Stopping Rules for Turbo Decoders

Ivanov Yu.Yu.1, Kulyk A.Ya.2

1Postgraduate student of Institute of Automatics, Electronics and Computer Control Systems, Vinnytsia National Technical University, st. Khmelnytsky Shosse 95, Vinnytsia, Ukraine, YuraII@yandex.ru
2Prof., Ph.D., Prof. of Institute of Automatics, Electronics and Computer Control Systems, Vinnytsia National Technical University, st. Khmelnytsky Shosse 95, Vinnytsia, Ukraine, kulyk1960@mail.ru

Abstract — Decoders for turbo codes are iterative in nature, i.e., they have to perform a certain number of iterations before reaching a satisfactory degree of confidence regarding a frame to be decoded. This paper presents and compares some simple and efficient criterions for stopping the iteration process of turbo decoding.

Keywords: turbo code, stopping rules, iterative decoding process, implementation complexity.

I. INTRODUCTION

At the time of the latest technology information became the object of automated processing. The data transmission process in information systems is susceptible to errors, because any error in the calculations can breach them. To combat interferences in data transmission systems at all the stages error-correction coding is used, that provides reliability and credibility of transmitted information. The analysis allowed us to determine, that the most prominent achievement in the theory of error-correction coding in recent years is turbo code — the powerful algorithm of the modern communication systems. This code is the iterative probabilistic method of error-correcting coding with the reliable performance, which is very close to the C.E. Shannon theoretical limit [1]. Turbo codes are used to encode large volume information messages at the high speed with high error-correcting. The turbo codes evolution for the channels with AWGN and Rayleigh fadings have shown a very high potential of this coding technique at the low energy regions. These codes are used in practice in the most important areas, such as mobile and space satellite communications, digital television, DNA correcting [2].

The main problems are the lack of freeware distribution, low minimum distance, decoding algorithms complexity, early termination questions.

The work objective is to study the using of some stopping iterations criterions in turbo code systems.

The research subject is the mathematical algorithms for early termination in the turbo decoding process.

To achieve the work objective it is necessary to solve the following task: to analyze the work of stopping rules for turbo decoders in communication systems.

II. TURBO DECODING AND STOPPING RULES

The main principle of the classical turbo encoding is the usage of two parallel working constituent encoders, although you can use the arbitrary dimension encoders. The information block is encoded twice, the second time — after interleaving process, which helps to create the stream of independent errors (eliminate error packets). The perforation patterns are used for turbo codes to increase the data rate. The concept of “log-likelihood ratio” (LLR) is used in the decoding process. The decoded information (N bits) from the first $L_{e1}$ (second $L_{e2}$) decoder output is used as a
priori information \( L_{a2} \) for the second (first \( L_{a1} \)) decoder input in order to clarify the decoding result using some modifications of an iterative C. Berrou algorithm BCJR on maximum a posteriori probability. This is very efficient “soft” decoding algorithm for turbo codes, which minimizes the probability of bit error (bit-by-bit decoding). After exiting from the second decoder output the “soft” values of the extrinsic log-likelihood ratio is deinterleaved and “hard” decision on the transmitted data bit is accepted [1-2].

Mainly, an early stopping rules consists in a comparison of a measure, calculated after each iteration \( i \), with a \( \mu \) threshold. In the following, we present some of the existing main stopping rules [3-4].

1. **Sign-Change Ratio (SCR)**. In this criteria the iteration process will be stopped at the \( i \)-th iteration if the ratio \( C(i)/N \) is lower than threshold \( \mu \) (value between 0.005 and 0.01), where \( C(i) \) is the number of the sign differences between \( L_{c2k}^{-1} \) and \( L_{c2k}^i \).

2. **Sign-Difference Ratio (SDR)**. This criteria is a replica of SCR in which \( C(i) \) is calculated as the number of sign differences between \( L_{a2k}^i \) and \( L_{c2k}^i \).

3. **Hard Decision-Aided (HDA)**. Proceeding from the second criterion, the decoding process is stopped after iteration \( i \) for \( i \geq 2 \), if \( \text{sign}(L_{c2k}^i) = \text{sign}(L_{c2k}^{-1}) \).

4. **Mean Estimate (ME)**. After each iteration the mean \( M(i) \) of the absolute values of the LLRs is calculated, and the decoding process is stopped if
   \[
   M(i) = \frac{1}{N} \sum_{k=1}^{N} |L_{c2k}^i| > \mu .
   \]  

5. **Minimum LLR (mLLR)**. This rule stops the decoding process after iteration \( i \) for \( i \geq 2 \), if \( \min_{1 \leq k \leq N} |L_{c2k}^i| < \mu \).

6. **Sum-Reliability (SR)**. After each iteration \( i \) the sum \( S \) of the absolute values of the LLRs is calculated. And the decoding process is stopped after iteration \( i \) for \( i \geq 2 \), if \( S(i) \leq S(i-1) \).

7. **Cross Entropy (CE)**. Based on the Kullback-Leibler divergence – the CE between the distribution of the estimates at the outputs of the decoders at iteration \( i \) can be approximated by
   \[
   T(i) = \sum_{k=1}^{N} \frac{(L_{c2k}^i - L_{c2k}^{-1})^2}{e^{\mu |L_{c2k}^i|}} .
   \]  

The decoding process is stopped after iteration \( i \) for \( i \geq 2 \), if \( T(i)/T(1) < \mu \), where threshold \( \mu \) is around \( 10^{-3} \).

8. **CRC Rule**. A separate error-detection code, especially a cyclic redundancy check (CRC) code, can be concatenated as an outer code with an inner turbo code in order to flag erroneous decoded sequences. The decoding process is stopped after iteration \( i \) whenever the syndrome of the CRC is zero. The CRC code used for the CCSDS standard, with redundancy \( l = 16 \), has a minimum distance of at least \( d = 4 \) for all the turbo frame sizes recommended by the CCSDS and detects at least 99.9985 % of all frame errors.

Table I illustrates the implementation complexity for popular stopping rules.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Add</th>
<th>Mult</th>
<th>Memory</th>
<th>Impl. Comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>mLLR</td>
<td>2N</td>
<td>-</td>
<td>-</td>
<td>very low</td>
</tr>
<tr>
<td>SR</td>
<td>2N</td>
<td>-</td>
<td>1</td>
<td>very low</td>
</tr>
<tr>
<td>CRC</td>
<td>3N</td>
<td>-</td>
<td>-</td>
<td>very low</td>
</tr>
<tr>
<td>SDR</td>
<td>2N</td>
<td>1</td>
<td>-</td>
<td>low</td>
</tr>
<tr>
<td>SCR</td>
<td>2N</td>
<td>1</td>
<td>-</td>
<td>low</td>
</tr>
<tr>
<td>HDA</td>
<td>2N</td>
<td>-</td>
<td>1N \times 1 bit</td>
<td>medium</td>
</tr>
<tr>
<td>CRC</td>
<td>2N</td>
<td>1</td>
<td>1N</td>
<td>high</td>
</tr>
<tr>
<td>CE</td>
<td>3N</td>
<td>3N</td>
<td>2N + LUT</td>
<td>very high</td>
</tr>
</tbody>
</table>

9. **Magic GENIE**. The GENIE (optimum) criteria can be used in the case where the original information bits are known. Then, the iteration is stopped immediately after the frame is correctly decoded. This unrealizable criterion gives a limit for all possible criterions.

10. **Combiner Criteria**. This rule is a combination for these techniques.

To prevent an endless loop, if the stopping rules are never satisfied, decoding is finally stopped after a maximum of \( i = 6-10 \) iterations.

**Conclusions**

Stopping rules for turbo decoders can decrease the decoding average time, but sometimes increasing possibility of the maximum number of the imposed iterations. Such criterions must not alter the bit error rate (BER) performance obtained through the realization of all iterations. The utility of such iteration stopping criterions – compromise between the the elimination of the unnecessary iterations and the conservation of the BER performance. Consequently, the iteration control criteria mLLR, SR, and CRC are best suited for implementation.

**REFERENCES**


