

# Analysis of Iteration Control for Turbo Decoders in Turbo Synchronization Applications

Timo Lehnigk-Emden, Uwe Wasenmüller, Christina Gimmler, and Norbert Wehn

Microelectronic Systems Design Research Group, University of Kaiserslautern,  
Erwin-Schroedinger Str., 67663 Kaiserslautern, Germany

**Abstract.** Wireless data transmission results in frequency and phase offsets of the signal in the receiver. In addition, the received symbols are corrupted by noise. Therefore, synchronization and channel coding are vital parts of each receiver in digital communication systems. By combining the phase and frequency synchronization with an advanced iterative channel decoder (inner loop) e.g. turbo codes in an iterative way (outer loop), the communications performance can be further increased. This principle is referred to as turbo synchronization. The energy consumption and the peak throughput of the system depend on the number of iterations for both loops. An advanced iteration control can decrease the mean number of needed iterations by detecting correctly decoded blocks. This leads to a dramatic energy saving or to an increase of throughput. In this paper we present a new stopping criterion for decodable blocks for turbo decoding in interrelation with turbo synchronization. Furthermore the implementation complexity of the turbo decoder is shown on a Xilinx FPGA.

---

## 1 Introduction

Synchronization and channel decoding are essential parts of every digital receiver for wireless communication. Task of the synchronization is to present data bits to the channel decoder, where the negative influences of timing, frequency and phase offset are eliminated. A well known advanced scheme for channel coding are turbo codes (Berrou et al., 1993). Turbo code decoding is done in an iterative algorithm based on the maximum a posteriori principle. The turbo decoder produces extrinsic information for the systematic bits, which are used in the next iteration. Communications performance, throughput and power consumption of a turbo code

decoder depend on the number of iterations. Thus it is desirable to find criteria for the iteration control without negative impact on the communications performance.

Synchronization is typically performed only once before channel decoding. The method of joint iterative turbo code decoding and synchronization is called turbo synchronization (Godtmann et al., 2006). Turbo synchronization allows to decrease the number of pilot symbols for synchronization purposes and thus increases the user data rate for a given bandwidth. In turbo synchronization the turbo decoder must additionally calculate new a posteriori information on the parity bits for the next synchronization refinement. The essential calculation of the extrinsic information of the parity bits in turbo synchronization provides the possibility to use this additional information for iteration control. State-of-the-art iteration controls for turbo decoder cannot check for a valid codeword without adding additional redundancy bits. Therefore they can stop the decoder for invalid codewords too. Our novel iteration control is based on a check, whether the results of the actual turbo decoding iteration represent a valid codeword without extra redundancy bits.

In this paper we apply turbo synchronization for the iterative synchronization of phase and frequency offset as well as iteration control. In Section 2 we summarize turbo encoding as well as the principle of turbo code decoding. In the following Section 3 we explain the iterative synchronization and the interrelation with the decoding algorithm. In Section 4 we categorize the state-of-the-art methods for iteration control and present our new iteration control method. In Section 5 we analyse the communications performance of turbo synchronization. We thoroughly compare our advanced iteration control to state-of-the-art taking communications performance and hardware performance into account. Communications performance is determined by bit true models. Implementation complexities and results are given based on Xilinx devices.

## 2 Turbo Codes

With the introduction of binary turbo codes by Berrou in 1993 (Berrou et al., 1993) near optimum error correction became possible. Due to these error correction capabilities, binary and duo-binary turbo codes allow for low frame error rates (FER) at a low signal-to-noise ratio (SNR), outperforming the widely used convolutional codes. Because of this advantage turbo codes are now part of a large number of communication standards.

Turbo codes in general consist of a serial or parallel concatenation of two codes, so called component codes, and an interleaver. While the first component code encodes the information in the original order, the second one gets the information in a permuted order, see Figure 1a). In all standards convolutional codes are used as component codes.

Decoding of turbo codes is an iterative process where probabilistic information is exchanged between component decoders (Berrou, 2003). Iterative decoding implies a big challenge with respect to low latency and high throughput decoders.

A possible realization of a decoder of turbo codes is given in Figure 1b). The two component decoders that decode the two component codes are connected via interleaver and deinterleaver. They use *log likelihood ratios* (LLR)  $\lambda_{d_k}^s, \lambda_{d_k}^{p_1}$  and  $\lambda_{d_k}^{p_2}$  of the systematic and parity information to compute the extrinsic information  $\Lambda_{d_k}^{e1}$  and  $\Lambda_{d_k}^{e2}$  on the information bits. The iterative exchange of  $\Lambda_{d_k}^{e1}$  and  $\Lambda_{d_k}^{e2}$  between these component decoders is referred to as turbo principle. One (*full*) iteration is done if decoder one and decoder two have run once. If only one decoder has calculated new information, we call this one *half iteration*.

Both component decoders perform a *maximum a posteriori probability* (MAP) decoding. However the suboptimal Max-Log MAP algorithm with *extrinsic scaling factor* (ESF) is more suitable for implementations. In comparison to the optimal algorithm the Max-Log MAP results in a performance loss below 0.2 dB (Robertson et al., 1995)(Robertson et al., 1997). Moreover it was shown in (Worm et al., 2000) that, when employed in turbo decoding, one does not require knowledge of the SNR.

The Max-Log MAP algorithm consists of a forward and a backward recursion. It computes for each possible information or parity bit  $d_k$  an *a posteriori probability* (APP) LLR  $\Lambda^s, \Lambda^{p_1}, \Lambda^{p_2}$ .

## 3 Turbo Synchronization

The synchronization consists of the estimation of the unknown parameters of timing, frequency and phase offset, and the elimination of all possible negative influences introduced by these parameters. We focus on the frequency and phase synchronization of bursts with linear modulation (e.g. QPSK,16-QAM) in conjunction with turbo decoding.

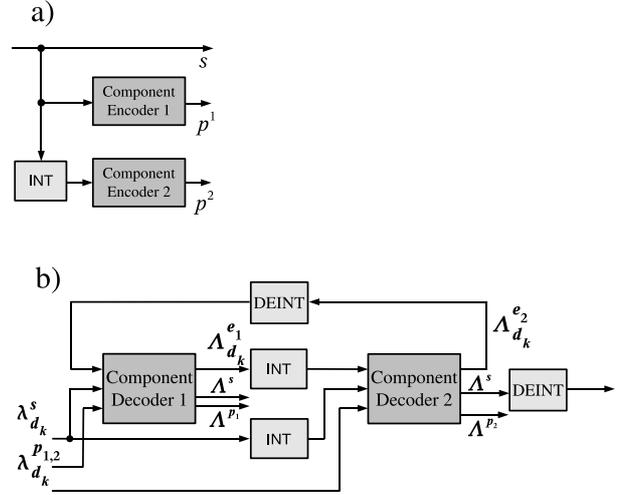


Fig. 1. a) Turbo Encoder, b) Turbo Decoder

We assume, that the steps of gain control, timing and burst detection are properly carried out before. The received sample sequence  $r$  is given in the complex baseband according to Equation 1:

$$r(l) = s(l) \cdot e^{j(2\pi f_o l + \Phi)} + n(l) \quad l = 0, 1, \dots, L - 1 \quad (1)$$

The sample sequence  $r$  with  $L$  elements is based on modulation symbols  $s(l)$  with one sample per symbol and symbol duration  $T$ , and is disturbed by a noise sequence  $n$ . The frequency offset  $f_o$  and phase offset  $\Phi$  have to be estimated and corrected. They are considered fixed during an estimation interval. The synchronization is done in two main steps. Initially, a coarse synchronization is carried out. Afterwards, fine synchronization is done iteratively with the additional use of tentative decoder decisions after each decoder iteration. The methods for the coarse synchronization step depend on the existing communications system. The choice of method is influenced by the maximum frequency offset, the number of symbols per burst as well as the number and placement of known symbols.

We concentrate on the step of fine synchronization, which will be used for turbo synchronization. With the assumption, that all symbols of the burst are known the effect of the modulation by  $s(l)$  can be removed by:

$$\tilde{r}(l) := r(l) \cdot s^*(l) \quad l = 0, 1, \dots, L - 1 \quad (2)$$

A lot of algorithms exist to estimate frequency and phase offset for an unmodulated carrier as described in Equation 2 (Meyr et al., 1998; Mengali and D'Andrea, 1997). However it must be considered, that usually the symbols of the burst are unknown or only some symbols, used for supporting the burst detection or supporting the coarse synchronization are known. Thus we replace the transmitted symbol sequence  $s$  by an estimated symbol sequence  $s_e$ . The estimation of

the transmitted symbol sequence is provided by the turbo decoder. Furthermore, the fine synchronization runs in parallel to the decoder iterations to avoid throughput degradation by turbo synchronization; thus a fast and simple algorithm is required. The fine estimation of frequency and phase offset is based on the average phase of the front part and rear part of the burst with a modulation removal by the estimated symbol sequence  $s_e$ . This is formally given by

$$Z(k) := \sum_{l=0+k \cdot L/2}^{L/2-1+k \cdot L/2} r(l) \cdot s_e(l) \quad k = 0, 1 \quad (3)$$

With the two phase values of Equation 3 the estimate of the frequency offset can be calculated with

$$\tilde{f}_0 = \frac{\arg(Z(1) \cdot Z^*(0))}{2\pi \cdot L} \quad (4)$$

The estimate of the phase offset is calculated with the help of Equation 4

$$\tilde{\phi} = \arg(Z(1) + Z(0)) - L \cdot \tilde{f}_0 \cdot \pi \quad (5)$$

The first decoder iteration is based on the LLR values  $\lambda_{d_k}^s$ ,  $\lambda_{d_k}^{p1}$  and  $\lambda_{d_k}^{p2}$  calculated with the symbols of the coarse synchronized received sequence. For the iterative fine synchronization an estimate of the transmitted symbols is used, which is produced by the turbo code decoder after each iteration. The estimate of the transmitted symbols is gathered by the APP LLR of the decoder. A turbo code decoder computes APP LLR values  $\Lambda^s$  of the systematic bits by default. In turbo synchronization applications the decoder must additionally calculate the APP LLR values  $\Lambda^{p1,2}$  for the parity bits.

To reduce the effect of using some erroneous reference symbols in the fine synchronization, soft values are used for the reference symbols. The values for the quadrature components of the estimated symbols are calculated by a *tanh* operation on the APP LLR. With the tentative soft values of systematic bits and parity bits the sequence  $s_e$  is generated and the described fine synchronization process can be carried out.

The received sequence  $r$  is corrected with the new estimates of frequency and phase offset. A new synchronized received sequence  $r$  is calculated after each full decoder iteration. Turbo decoding and fine synchronization run in parallel in our architecture. For the  $n$ -th iteration of the decoder the LLR values  $\Lambda^s$ ,  $\Lambda^{p1}$ , and  $\Lambda^{p2}$  of the transmitted bits are calculated on base of the fine synchronized sequence, which used the APP LLR values of the  $(n-2)$ -th iteration. A discussion of the effects of the schedule regarding the update of LLR values can be found in (Alles et al., 2007).

## 4 Iterations Control

Latency and energy consumption strongly depend on the number of iterations in the decoding process. Thus, reducing the number of iterations in the decoding process is an urgent issue. Instead of using a fixed number of iterations, the number of iterations has to be controlled by intelligent iteration control mechanisms. An efficient iteration control allows to detect correctly decoded blocks while decoding. Thus, a good iteration control is key to optimize energy consumption and reduce the mean latency. In (Thul et al., 2002) it was shown that iteration control is the most efficient technique for energy saving in a turbo decoder system without sacrificing communications performance.

### 4.1 State-of-the-Art

**Cyclic Redundancy Check (CRC) Rule.** A separate error-detection code, such as a CRC, can be concatenated as an outer code with an inner turbo code in order to flag erroneously decoded sequences. The decoding process is stopped after iteration  $i$  whenever the syndrome of the CRC is zero. The false alarm rate in sense of recognizing a erroneous block as correct is a delta to zero. One disadvantage of the CRC is an additional code rate loss. However, in many systems like UMTS LTE the CRC code is available.

**Hard decision Rules.** The idea behind these algorithms presented in (Wu et al., 2000) and (Shao et al., 1999) is to follow the changes of the hard decisions of either the extrinsic information, the APP information or the decoded bits within one or two iterations. The disadvantage of these rules are that they are not able to decide if the decoded codeword is a valid codeword.

**Soft decision Rules** Other approaches observe the soft LLR or APP values in the MAP decoders. With these values, several metrics can be calculated, e.g. the mean LLR value. For more details the reader is referred to (Wu et al., 2000)(Hagenauer et al., 1996) and (Matache et al., 2000). All soft decision rules have in common, that thresholds have to be defined for aborting the decoding process. In addition, there is no guarantee that the decision is correct. The choice of the thresholds is always a trade-off between false alarm rate and detection rate of correctly decoded blocks.

### 4.2 Re-Encoding approach

Our approach is to check whether the actually decoded information represent a valid codeword or not. One possibility to realize this, is to re-encode the hard decoded information bits and to compare the parity bits from the decoder and encoder. If a valid codeword is detected the decoder can be stopped.

The decoder calculates LLR values of the information and parity bits. The hard decision information bits are given to a turbo encoder which is identical the the encoder at the transmitter side, for re-encoding. The encoder produces new par-

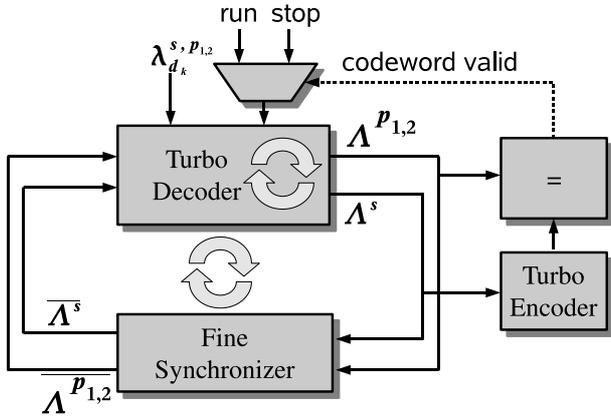


Fig. 2. Turbo Decoder with fine synchronization and iteration control

ity bits which are compared to the hard decision parity bits calculated by the two MAP decoders. If there is no difference the decoder has found a valid codeword. Further iterations make no sense. The decoder can be stopped immediately, see Figure 2.

For implementation the check can be done in parallel to the MAP decoding which leads to no throughput degradation of the MAP decoder. In parallel to the MAP processing the encoder encodes the systematic information and compares the parity of the current MAP decoder. A valid codeword is detected if the systematic bits do not change within two successive half iterations and if the two parity bit sets do not differ. In this case the decoder and synchronization loops can be stopped and the system is ready for a new codeword.

### 5 Results

The new approach for iteration control is validated per software simulations in sense of communications performance. In addition the implementation complexity for the turbo synchronization with iteration control on a Xilinx FPGA is presented and compared to the non iteration control solution.

#### 5.1 Communications Performance

The new iteration control detects correctly decoded blocks and stops the decoder. This approach has no impact on the communications performance, which is shown by simulations in Figure 3. The figure shows the results for a block length of 1056 and 1136 information bits for the code rates 0.443 and 0.8 respectively. For the lower code rate code 16-QAM gray modulation is used for the higher rate QPSK respectively. Down to a frame error rate (FER) of  $10^{-3}$  the iteration control has no effect on the communications performance, the curves are congruent. In addition, the turbo synchronization gains up to 0.25 dB compared to the initial synchronization only.

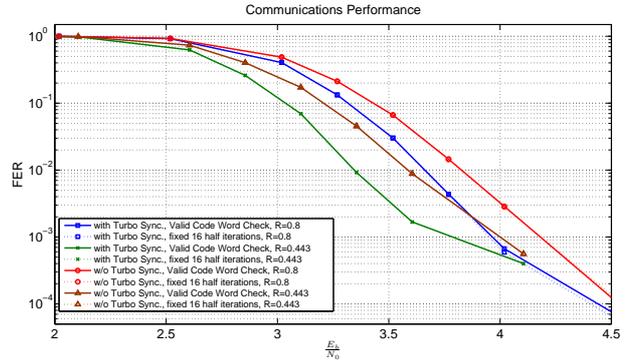


Fig. 3. Comparison communications performance - with and w/o iteration control; 16 decoding half iterations

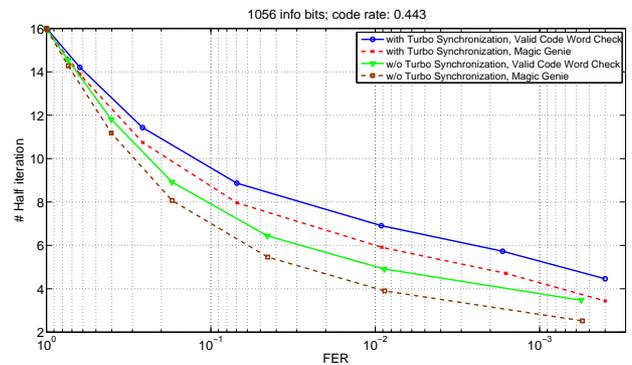
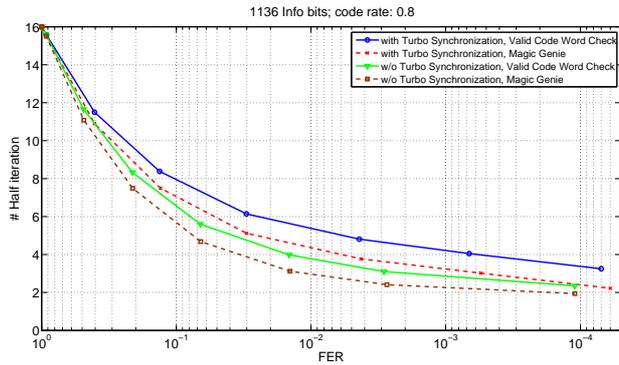


Fig. 4. Iteration Control: Magic Genie vs. Re-encoding approach - with and w/o turbo synchronization; max 16 decoding half iterations; code rate 0.443

An optimal stopping criterion for decodable blocks, we call this “magic-genie”, can stop the decoder at once if the decoded information bits are correct. A comparison of the needed iterations between the “magic-genie” criterion and the re-encoding approach is shown in Figure 4 for the code with rate 0.443 and in Figure 5 for the code with the rate 0.8, respectively.

The re-encoding approach is close to the magic-genie criterion. The maximum difference is one half iteration. For lower FER the criterion can save up to 80% of the iterations. That means the energy consumption of the whole system can be decreased dramatically or the peak throughput can be increased. In addition it can be seen, that 50% of iterations can be saved at a frame error rate below  $10^{-1}$  for high rate code. The turbo decoder with turbo synchronization operates at a lower SNR for the same FER than without turbo synchronization, see Figure 3. Therefore it needs up to two more half iterations in average. In addition the new criterion can be used as quality indicator of the channel. A statistic over several blocks can be seen as a rough channel estimation.



**Fig. 5.** Iteration Control: Magic Genie vs. Re-encoding approach - with and w/o turbo synchronization; max 16 decoding half iterations; code rate 0.8

	Binary Turbo Decoder
<b>Architecture</b>	Serial MAP, 3 ACS units
<b>Trellis States</b>	8
<b>Infoword Size</b>	128 - 5124 bits
<b>Codeword Size</b>	128 - 15372 bits
<b>Code Rates</b>	$\geq 1/3$
<b>Parallelism</b>	1
<b>Input Quantization</b>	6 bit
<b>Algorithm</b>	MaxLog-MAP
<b>ESF</b>	0.75
<b>Max. Iterations</b>	8
<b>Technology</b>	FPGA
<b>Clock Frequency</b>	233 MHz
<b>Payload Thrpt.[Mbit/s]</b>	4.67 - 14.02
<b>Latency [<math>\mu</math>s]</b>	10.36 - 414.7
<b>Payload Bit/Cycle</b>	0.02 - 0.06

**Table 1.** Key parameter of the turbo decoder

## 5.2 Hardware Implementation

The implementation complexity is analysed on a Xilinx Virtex-5 FPGA. Three versions have been implemented. First the turbo decoder without the extra parity LLR calculation unit which is needed for the turbo synchronization. The second one with the parity LLR calculation unit. The third version extends the second one with the new iteration control. The architecture of the turbo decoder is a state-of-the-art SMAP architecture with three recursion units which run in parallel. The key parameters are summarized in Table 1. For more details the reader is referred to (May et al., 2007).

Table 2 shows a comparison of implementation complexity of the three cases. The extra parity calculation and the encoder for the iteration control utilize additional 57 slices, i.e. 4%. That means that the hardware overhead is negligible in this case.

	Turbo Decoder	Turbo Decoder	Turbo Decoder
<b>calc. parity LLR</b>	no	yes	yes
<b>iteration control</b>	no	no	yes
Xilinx Virtex-5 XC5VSX95T @ 233 MHz			
<b>LUT</b>	4,432	4,495	4,513
<b>FF</b>	2,255	2,283	2,297
<b>Slices</b>	1,427	1,477	1,484
<b>BRAMs</b>	4	4	4

**Table 2.** Comparison implementation complexity

## 6 Conclusions

In this paper we presented a new iteration control for turbo decoding in combination with turbo synchronization for turbo decoders. With turbo synchronization a communications performance gain of 0.25 dB for the low and high rate code can be achieved. The stopping criterion is based on a re-encoding approach to detect a valid codeword during the decoding process. Stopping the decoding loop with this criterion saves up to 80% of iterations and energy without losing any communications performance. The hardware overhead for the stopping criterion is below 5% of the whole turbo decoder.

## References

- Alles, M., Lehnigk-Emden, T., Wasenmüller, U., and Wehn, N.: Implementation Issues of Turbo Synchronization with Duo-Binary Turbo Decoding, in: Proc. 19th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2007, Athens, Greece, 2007.
- Berrou, C.: The Ten-Year-Old Turbo Codes are Entering into Service, *IEEE Communications Magazine*, 41, 110–116, 2003.
- Berrou, C., Glavieux, A., and Thitimajshima, P.: Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes, in: Proc. 1993 International Conference on Communications (ICC '93), pp. 1064–1070, Geneva, Switzerland, 1993.
- Godtmann, S., Pollok, A., Hadaschik, N., Steinert, W., Ascheid, G., and Meyr, H.: Joint Iterative Synchronization and Decoding Assisted by Pilot Symbols, in: IST Mobile & Wireless Communications Summit, Myconos, Greece, 2006.
- Hagenauer, J., Offer, E., and Papke, L.: Iterative Decoding of Binary Block and Convolutional Codes, *IEEE Transactions on Information Theory*, 42, 429–445, 1996.
- Matache, A., Dolinar, S., and Pollara, F.: Stopping Rules for Turbo Decoders, TMO Progress Report 42–142, [http://tda.jpl.nasa.gov/progress\\_report/](http://tda.jpl.nasa.gov/progress_report/), Jet Propulsion Laboratory, Pasadena, California, 2000.
- May, M., C.Neeb, and Wehn, N.: Evaluation of High Throughput Turbo-Decoder Architectures, in: IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, USA, 2007.
- Mengali, U. and D'Andrea, A.: Synchronization Techniques for Digital Receivers, Plenum Publishing Corporation, New York, 1997.

- Meyr, H., Moeneclaey, M., and Fechtel, S. A.: *Digital Communication Receivers*, John Wiley & Sons Inc, 1998.
- Robertson, P., Villebrun, E., and Hoeher, P.: A Comparison of Optimal and Sub-Optimal MAP decoding Algorithms Operating in the Log-Domain, in: *Proc. 1995 International Conference on Communications (ICC '95)*, pp. 1009–1013, Seattle, Washington, USA, 1995.
- Robertson, P., Hoeher, P., and Villebrun, E.: Optimal and Sub-Optimal Maximum a Posteriori Algorithms Suitable for Turbo Decoding, *European Transactions on Telecommunications (ETT)*, 8, 119–125, 1997.
- Shao, R. Y., Lin, S., and Fossorier, M. C. P.: Two Simple Stopping Criteria for Turbo Decoding, *IEEE Transactions on Communications*, 47, 1117–1120, 1999.
- Thul, M. J., Vogt, T., Gilbert, F., and Wehn, N.: Evaluation of Algorithm Optimizations for Low-Power Turbo-Decoder Implementations, in: *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, pp. 3101–3104, Orlando, Florida, USA, 2002.
- Worm, A., Hoeher, P., and Wehn, N.: Turbo-Decoding without SNR Estimation, *IEEE Communications Letters*, 4, 193–195, 2000.
- Wu, Y., Woerner, B. D., and Ebel, W. J.: A Simple Stopping Criterion for Turbo Decoding, *IEEE Communications Letters*, 4, 258–260, 2000.