

Low complexity Turbo synchronization without initial carrier synchronization

U. Wasenmüller, C. Gimmler, and N. Wehn

Microelectronic Systems Design Research Group, University of Kaiserslautern, Erwin-Schrödinger 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) like turbo codes in an iterative way (outer loop), the communications performance can be increased. This principal is referred to as turbo synchronization. For turbo synchronization an initial estimate of phase and frequency offset is required. In this paper we study the case, where the initial carrier synchronization is omitted and an approach with trial frequencies is chosen. We present novel techniques to minimize the number of trial frequencies to be processed. The communications performance and effort of our method is demonstrated. Furthermore the implementation complexity of the whole system is shown on a Xilinx FPGA.

1 Introduction

Synchronization and channel decoding are vital parts of every digital receiver for wireless communication. The transmission over a wireless channel results in timing, frequency and phase offsets. In addition, the received symbols are corrupted by noise. 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 is the use of turbo codes. The turbo encoder delivers a stream of systematic bits and two parity bit streams by a recursive systematic convolutional encoding of the user data bits and an

interleaved version of the user data bits. Turbo code decoding is done in an iterative algorithm based on the maximum a posteriori principle. Communication systems with Turbo codes can operate at very low signal-to-noise ratios (SNR).

Frequency offset synchronization and phase offset synchronization (carrier synchronization) is typically performed only once before channel decoding. The variance of the phase and frequency estimation depends on the SNR as well as on the number of symbols available for estimation of the phase and frequency offset. The mentioned variances influence the decoder performance heavily; i.e. large variances lead to an unacceptable performance degradation of the used decoder.

In our paper we focus on carrier synchronization in conjunction with turbo code decoding. The method of joined iterative turbo code decoding and synchronization is called turbo synchronization. Turbo synchronization allows correct decoding also for larger variances of frequency and phase offset. This can be used to decrease the number of known symbols for synchronization purposes and thus increases the user data rate for a given bandwidth. However, also the advanced turbo synchronization technique needs an initial coarse carrier synchronization and can tolerate frequency offset estimation errors up to a small limit.

In the situation of unacceptable estimation errors of the initial carrier synchronization tentative decoding with several trial frequency offsets must be performed. The effort of the receiver depends now on the number of investigated trial frequency offsets. We demonstrate novel techniques for minimizing the number of trial frequency offsets to be processed. To prove the feasibility of the system approach an implementation of the whole system is presented. The key facts of the hardware architecture for achieving low complexity for the principally expensive usage of trial frequencies are illustrated. Furthermore we analyze the additional effort for the turbo decoder caused by turbo synchronization and processing of trial frequency offsets and demonstrate the



Correspondence to: U. Wasenmüller
(wasenmueller@eit.uni-kl.de)

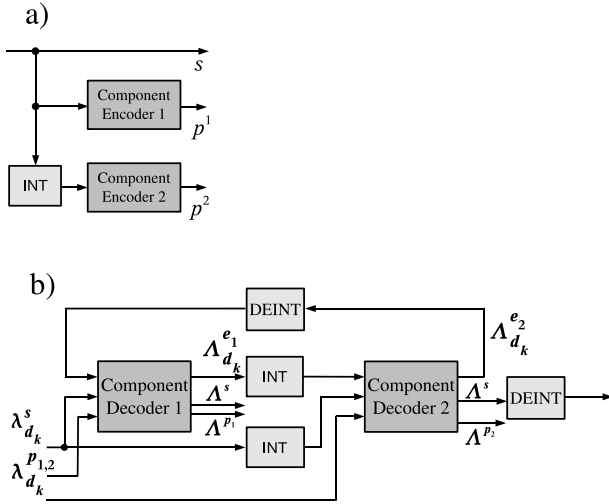


Fig. 1. (a) Turbo encoder, (b) Turbo decoder.

communications performance of the targeted communication system.

In Sect. 2 we summarize turbo encoding as well as the principle of turbo code decoding. In the following Sect. 3 we explain the iterative synchronization and the interrelation with the decoding algorithm. In Sect. 4 we give an short overview about the target communication system and explain the use of trial frequencies for frequency offset estimation. Our proposed method for minimizing the number of trials to be performed is presented. In Sect. 5 we analyze the communications performance of our proposed method. The hardware architecture is explained and hardware complexity of turbo synchronization without initial carrier synchronization is analyzed. Communications performance is determined by bit true models. Implementation complexities and results are given based on Xilinx devices. The paper is concluded in Sect. 6.

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 generally 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 Fig. 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 requirements.

A possible realization of a decoder for turbo codes is given in Fig. 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 $\Lambda_{d_k}^{e_1}$ and $\Lambda_{d_k}^{e_2}$ on the information bits. The iterative exchange of $\Lambda_{d_k}^{e_1}$ and $\Lambda_{d_k}^{e_2}$ between these component decoders is referred to as turbo principle. One (*full*) iteration is done if Decoder 1 and Decoder 2 have run once. If only one decoder has calculated new information, we call this one *half iteration*.

Both component decoders optimize the *maximum a posteriori probability* (MAP) criterion. However, in hardware implementations the suboptimal Max-Log MAP algorithm with *extrinsic scaling factor* (ESF) is more suitable. In comparison to the optimal algorithm the Max-Log MAP results in a performance loss of less than 0.2 dB (Robertson et al., 1995, 1997). Furthermore, the Max-Log MAP algorithm does not require knowledge of the SNR in contrast to the optimal Log MAP algorithm (Worm et al., 2000).

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 Λ^s , Λ^{p_1} , Λ^{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. We assume, that the steps of gain control, timing synchronization and burst detection are properly carried out before. The received sample sequence r is given in the complex baseband according to Eq. (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 unknown parameters of frequency offset f_o and phase offset Φ have to be estimated for every received sample sequence. These parameters are considered fixed during an estimation interval; the sample sequence has to be corrected accordingly to the estimated frequency offset \tilde{f}_o and estimated phase offset $\tilde{\Phi}$.

The synchronization with turbo synchronization is done in two main steps. Initially, a (coarse) carrier synchronization is performed. In case of larger variances of the estimation parameters a degradation in the decoding performance will occur. The principle of turbo synchronization is to improve the estimation parameters for synchronization (fine synchronization) with the additional use of tentative decoder decisions. The improved carrier synchronization is used to provide better input data for the decoding process. This process is done iteratively after each decoder iteration. We concentrate first on the step of fine synchronization, which will be used for turbo synchronization.

Frequency and phase offset can be optimally estimated on an unmodulated carrier. With the assumption, that the transmitted symbol sequence s of the burst is known the effect of the modulation by each transmitted symbol $s(l)$ can be removed by:

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

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.

The fine estimation of frequency and phase offset is based on the average phase $\tilde{\phi}_0$ of the front part and on the average phase $\tilde{\phi}_1$ of the rear part of the burst with a modulation removal by the estimated symbol sequence s_e . This is formally given by

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

With the two phase values of Eq. (3) the estimate of the frequency offset can be calculated with

$$\tilde{f}_0 = \frac{\arg(\tilde{\phi}_0 \cdot \tilde{\phi}_1^*)}{2\pi \cdot L} \quad (4)$$

The estimate of the phase offset is calculated with the help of Eq. (4)

$$\tilde{\phi} = \arg(\tilde{\phi}_0 + \tilde{\phi}_1) - L \cdot \tilde{f}_0 \cdot \pi \quad (5)$$

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 Λ^s of the systematic bits by default.



Fig. 2. Considered burst structure.

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. Provided that the k -th transmitted QPSK symbol $s(k)$ contains the bits $d_k^{(I)}$ and $d_k^{(Q)}$ of the code word the estimated symbol $s_e(k)$ is calculated as:

$$s_e(k) = \tanh \frac{\Lambda(d_k^{(I)})}{2} + j \tanh \frac{\Lambda(d_k^{(Q)})}{2} \quad (6)$$

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 to avoid throughput degradation by turbo synchronization. For the n -th iteration of the decoder the LLR values $\bar{\Lambda}^s$, $\bar{\Lambda}^{p1}$, and $\bar{\Lambda}^{p2}$ 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 Evaluation of a grid of trial frequency offsets

For using the principle of turbo synchronization an initial coarse carrier synchronization with a sufficiently small variance of the estimation parameters is needed. The methods for the carrier synchronization depend on the existing communications system. The burst structure of the targeted communication system is depicted in Fig. 2. The burst consists at the start and the end of a sequence of known symbols – start unique word and end unique word respectively. The unknown symbols in the mid of the burst correspond to the code word of the turbo code encoded information sequence. A lot of algorithms exist to estimate frequency and phase offset with known symbols (data aided estimation) or with the unknown data symbols (blind estimation) as described in Meyr et al. (1998); Mengali and D’Andrea (1997). The variance of the estimation parameters depends on the SNR, the number of symbols and the use of known or unknown symbols.

In our targeted communication system the variance of frequency offset estimation with data aided estimation methods as well as with blind estimation methods is not sufficiently small to perform afterwards a turbo synchronization with satisfying communications performance. This is caused by the targeted SNR range below zero dB and the low number of symbols (known and unknown).

Thus a grid of trial frequency offsets is tentatively used. These trial frequency offsets are evaluated by the decoding to decide on the best estimation. The effort is now determined by the number of trial frequencies to be processed.

To perform synchronization and decoding a set of trial frequency offsets \tilde{f}_i is used. For using this method it is required to estimate a corresponding phase offset $\tilde{\phi}$ for each trial frequency offset. Applying the trial frequency offset to the received sequence r is given by:

$$r_{\tilde{f}_i} = r(l) \cdot e^{-j(\tilde{f}_i \cdot l)} \quad l = 0, 1, \dots, L-1 \quad (7)$$

For maximum likelihood estimation of the phase offset the well known V&V algorithm (Viterbi and Viterbi, 1983) is used. A correlation of the modified received sequence $r_{\tilde{f}_i}$ with the unique word at the start and end of the burst is performed, which is given by:

$$k_{\tilde{f}_i} = \sum_{l \in UW} r_{\tilde{f}_i}(l) \cdot u_{UW}^*(l) \quad (8)$$

The estimation of the phase offset $\tilde{\phi}$ for the trial frequency offset is given by the argument of the phasor $k_{\tilde{f}_i}$ of Eq. (8):

$$\tilde{\phi}_{\tilde{f}_i} = \arg(k_{\tilde{f}_i}) \quad (9)$$

As mentioned a criterion is required to decide on the best frequency offset. The received sequence r will be corrected with the trial frequency \tilde{f}_i and the belonging phase offset $\tilde{\phi}$. Based on this new sequence the LLR values for the turbo decoder can be calculated. The decoder performs the iterations as described in Sect. 2. For a decision on the best trial frequency offset an estimation s_e of the transmitted symbol sequence is used. The estimate of the transmitted symbol sequence is achieved as described in Sect. 3. For every trial frequency the correlation of received symbol sequence r and the estimated transmitted symbol sequence s_e is used. Correlation is a measurement of similarity and thus the trial frequency producing the greatest similarity between the two mentioned sequences is chosen. The correlation operation is given by

$$c_{\tilde{f}_i} = \sum_{l=0}^{L-1} r_{\tilde{f}_i}(l) \cdot s_e^*(l) \quad (10)$$

The selection process is done with

$$\tilde{f} = \tilde{f}_k \quad \text{with} \quad c_{\tilde{f}_k} \geq c_{\tilde{f}_i} \quad \text{for all } i \quad (11)$$

For the selection process of \tilde{f} it is sufficient to use the sequence s_e in Eq. (11), which is achieved after one or two turbo decoder iterations.

The computational effort for the evaluation of trial frequencies is dominated by the total number of turbo decoder iterations which must be performed. Thus it is desirable to minimize the number of trial frequency offsets to be evaluated by the decoder. The calculated phasor $k_{\tilde{f}_i}$ for phase estimation in Eq. (8) can be used to exclude trial frequency offsets from consideration before the decoding step. The magnitude of the correlation is a measure for similarity and thus only trial frequency offsets f_i with

$$|c_{\tilde{f}_i}| \geq C_{Tr} \quad (12)$$

will be processed. It must be emphasized, that the best estimation of the frequency offset is not given by the maximum correlation value of Eq. (8).

5 Results

Our approach for synchronization and decoding without initial carrier synchronization is validated per software simulations with bit true models in sense of communications performance. In addition the architecture of the system is presented. The complexity of the components for implementation on Xilinx FPGA is shown and briefly analyzed.

5.1 Communications performance

The communications performance of our system is demonstrated for a burst with QPSK modulation. The start unique word and end unique word contains 40 and 24 symbols, respectively. The code word uses 1248 symbols; it is based on turbo encoding with rate 1/3 and 16 states. A grid of 61 trial frequencies is used for this burst type. The trial frequencies are chosen to cover the maximum relative frequency offset of 6×10^{-3} and to allow a fine synchronization by the principle of turbo synchronization.

The communications performance expressed as a frame error rate (FER) is shown in Fig. 3 for different scenarios in an additive white gaussian noise channel. The graph with label "Blind" shows the FER for a blind carrier synchronization. Use of turbo synchronization produces no visible improvement in the figure. It is evident, that the carrier estimation parameters cause a unacceptable performance. The graph with label "no threshold" and the graph with label "2800" coincide. In the targeted communication system a FER of 10^{-3} is specified at SNR of -0.7 dB for ideal synchronization and optimal decoding. Hence the achieved communications performance is comparable to a system with perfect knowledge of the synchronization parameters. The effort advantage of the threshold criterion can be checked by Fig. 4. Using this threshold only approximately 60% of the trial frequencies have to be processed.

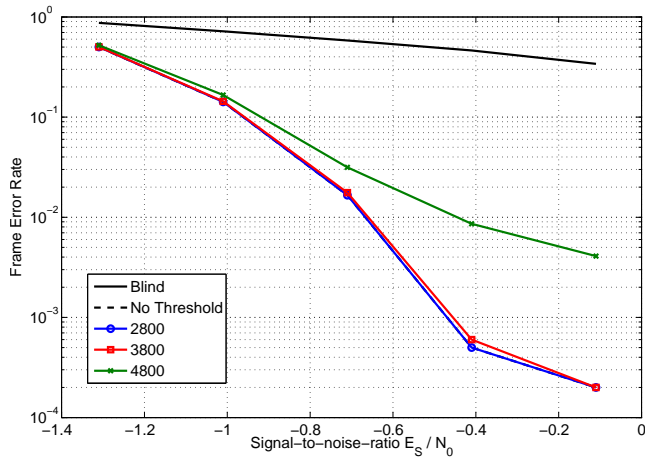


Fig. 3. Communications performance.

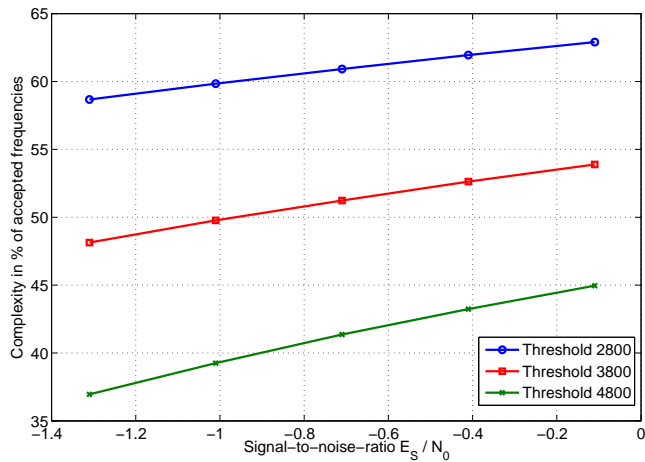


Fig. 4. Trial frequencies evaluated by decoder.

With the threshold parameter a trade off between communications performance and implementation performance can be regulated. The graphs with label “3800” and “4800” respectively show the FER performance for different values of the threshold. The computational effort for the different threshold values is shown in Fig. 4 as a percentage of the number of considered frequency offset values. For the threshold label “3800” approximately 50% of the trial frequencies have to be evaluated. It is possible to choose the threshold value for a defined SNR operation point. This is demonstrated with the graphs with label “3800”. For SNR values above -0.2 dB the performance is identical to the method without threshold. Depending on the exact value of the threshold about 40% to 60% of the trial frequencies can be excluded before the decoding step.

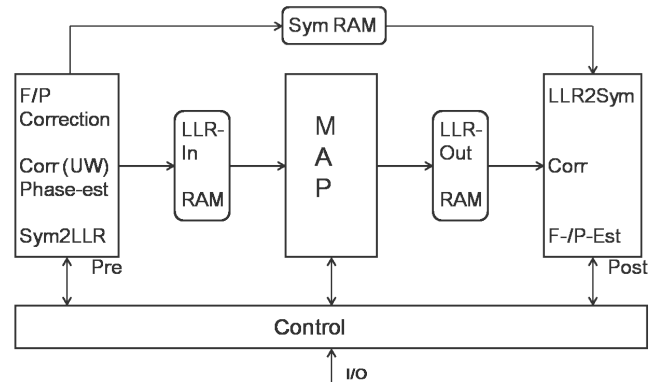


Fig. 5. Architecture.

Table 1. Key parameter of the turbo decoder.

Binary Turbo decoder	
Architecture	Serial MAP, 3 ACS units
Trellis states	16
Infoword size	128–5124 bits
Codeword size	128–15 372 bits
Code rates	$1/3 \dots 9/10$
Parallelism	1
Input quantization	6 bit
Algorithm	MaxLog-MAP
ESF	0.75
Max. iterations	8
Technology	FPGA
Clock frequency	223 MHz
Payload Thrpt. [Mbit/s]	4.67–14.02
Latency [μ s]	10.36–414.7
Payload Bit/Cycle	0.02–0.06

5.2 Architecture and implementation

The architecture of the system is depicted in Fig. 5. Objective of the architecture is to allow a hardware sharing of functionalities of turbo synchronization and trial frequency processing. Central part is the MAP component, which performs a half iteration of turbo code decoding. Component Pre includes the frequency and phase correction of the received sequence r , which is used for the step of fine synchronization in turbo synchronization as well as for processing of trial frequencies. The phase estimation according to Eq. (8) for the processing of trial frequencies is also included in component Pre as well as the determination, whether the actual trial frequency f_i will be excluded. The calculation of LLR values including in the demapping and depuncturing is used for providing the input data for the MAP component.

Unit Post carries out the central correlation operation according to Eqs. (3) and (10), respectively, which is used for fine synchronization as well as trial frequency processing. Both functionalities must use the transformation of LLR-Out

Table 2. Details of implementation complexity.

	MAP	Pre	Post	Ctrl	RAM	Sum
LUT	4372	2478	1974	909	1506	11 329
Xilinx Virtex-5 XC5VSX95T @ 223 MHz						
LUT	4372	2478	1974	909	1506	11 329
FF	2280	1243	1340	608	1064	6625
MAC	0	10	20	0	0	30
BRAMs	4	3	4	1	68	80

values to symbols as described for QPSK symbols in Eq. (6). The small part for frequency and phase estimation according to Eqs. (4) and (5) in component Post is used only for fine synchronization. All shown RAM blocks are double buffered to allow a parallel processing of the main components. Therefore different trial frequencies are processed in the components Pre, MAP and Post. The step of fine synchronization is carried out in component Pre and Post on the results of iteration n , while the MAP decoder executes the iteration $n + 2$.

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).

The resources for the components are presented in Table 2 for an implementation in a Xilinx Virtex-5 FPGA. Component Post is needed only for turbo synchronization and trial frequency evaluation respectively. Approximately half of the resources of component Pre are used for turbo synchronization and trial frequency evaluation, respectively.

6 Conclusions

In this paper we presented a novel method to reduce the complexity for turbo synchronization without initial carrier synchronization. To perform the decoding and synchronization steps without initial carrier synchronization a grid of trial frequencies is needed. The computational effort for processing of trial frequencies can be reduced by elimination of trial frequencies with a simple threshold comparison in the phase estimation step. Furthermore by applying turbo synchronization the grid of trial frequencies can be kept more coarse than with traditional decoding. The proposed hardware architecture allows a sharing of the components for evaluation of trial frequencies and for synchronization and decoding.

The components for trial frequency evaluation as well as for synchronization and turbo decoding can work concurrently. The mentioned features result in a low complexity system. The achieved communications performance is comparable to a system with perfect knowledge of the synchronization parameters.

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.
- May, M., Neeb, C., and Wehn, N.: Evaluation of High Throughput Turbo-Decoder Architectures, in: Proc. IEEE International Symposium on Circuits and Systems (ISCAS 2007), 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.
- Viterbi, A. J. and Viterbi, A. M.: Nonlinear Estimation of PSK Modulated Carrier Phase with Application to Burst Digital Transmission, IEEE Transactions on Information Theory, 32, 543–551, 1983.
- Worm, A., Hoeher, P., and Wehn, N.: Turbo-Decoding without SNR Estimation, IEEE Communications Letters, 4, 193–195, 2000.