

Optimum Frame Synchronization for Preamble-less Packet Transmission of Turbo Codes

Jian Sun and Matthew C. Valenti
Wireless Communications Research Laboratory
Lane Dept. of Comp. Sci. & Elect. Eng.
West Virginia University
Morgantown, WV 26506-6109
email: [jian,mvalenti]@csee.wvu.edu

Abstract—This paper introduces an optimum maximum a posteriori (MAP) frame synchronization method for packet-based turbo coded communication systems. The synchronizer maximizes the probability of frame synchronization by observing the received signal sequences. This method is based on the low-density parity-check properties of turbo codes and does not require insertion of sync words or preambles.

I. INTRODUCTION

Turbo codes were introduced by Berrou *et al* in 1993 [1]. While most work to date has viewed turbo codes from the literal perspective of being parallel concatenated recursive systematic convolutional (RSC) codes, Engdahl and Zigangirov provide an alternative way to view turbo codes as low density parity check (LDPC) codes [2]. The connection is established by the structure of convolutional codes. As early as 1973, Forney [3] suggested to transform truncated convolutional codes into linear block codes. Unlike usual LDPC codes that are defined on random sparse parity-check matrices, the linear block codes derived from turbo codes are highly structural, and in particular, they are quasi-cyclic. By “quasi-cyclic” we mean that the pattern in the parity-check matrix is repeated in the rows though the shift may be greater than one symbol.

Conventional frame synchronizers require insertion of sync words, or preambles. The correlation between the predefined sync word and received signal is calculated to determine the correct frame starting point. It is possible that the same or similar patterns of the sync word are present in the payload data. Hence the performance of synchronizers using sync words is constricted by the random data limit [4]. Besides, sync words consume signal energy. Thus insertion of sync words is not desirable for codes working at very low signal-to-noise ratio (SNR).

Both LDPC codes and turbo codes are attractive for their extraordinary error correction capability in low SNR environments. In order to fully achieve the potential capability, accurate frame synchronization is necessary. However, conventional frame synchronizers, which ignore the structure of the code, usually fail at low SNR. To improve frame acquisition performance, frame synchronization should be considered jointly with decoding [5] [6] [7] [8].

The low-density parity-check characteristics of LDPC and turbo codes enable us to examine if a valid codeword is

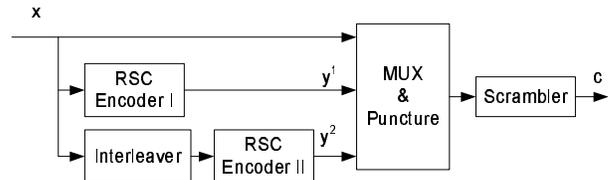


Fig. 1. Diagram of a turbo encoder.

received by using the parity-check equations [9]. In this work, an optimum frame synchronizer for packet-transmitted turbo codes is introduced. No additional preamble or sync words are required. The MAP frame synchronizer minimizes the probability of frame sync failure on a frame-by-frame base.

The remaining part of this paper is organized as follows. Section II introduces the parity-check properties of turbo codes. Section III describes the proposed frame synchronizer. Section IV presents simulation results. Finally, Section V concludes the work.

II. PARITY-CHECK CHARACTERISTICS OF TURBO CODES

A. Turbo encoder

Fig. 1 presents a diagram of a typical turbo encoder. A turbo encoder has two identical constituent RSC encoders. Encoder I uses x as its systematic input, while Encoder II uses an interleaved version of x as input. The interleaver size K is an important parameter of a turbo code which determines the length of a codeword. Usually an interleaver size $K > 1000$ is required for a powerful turbo code. The parity outputs y^1 and y^2 , together with x enter a multiplexer so that the bits are assembled into a codeword. In the multiplexer, some bits in y^1 and y^2 may be punctured in order to increase the code rate. A scrambler, also called a channel interleaver, permutes the codeword so that sequential symbols are interleaved. The permutation helps to combat burst errors which turbo codes are not good at dealing with. It also enables the frame synchronization technique proposed in this paper. All these components in the encoder determine the parity-check matrix H . The code structure is analyzed in the following.

B. Constituent RSC codes

We start with non-systematic convolutional (NSC) codes. If an NSC code has generating matrix $\mathbf{G}(D) = [g_1(D) \ g_2(D)]$, then its dual code is defined by the matrix $\mathbf{H}(D) = [g_2(D) \ g_1(D)]$. For example, let the octal representation of the generating polynomial be (7, 5), $\mathbf{G}(D) = [1 + D + D^2 \ 1 + D^2]$, the corresponding $\mathbf{H}(D) = [1 + D^2 \ 1 + D + D^2]$, and the matrix in numerical form is [10]

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ & & & & & & \cdots & & & & & \end{bmatrix}. \quad (1)$$

Most entries in \mathbf{H} are “0”. This sparseness of \mathbf{H} makes it resemble the parity check matrix of an LDPC code except that it is quasi-cyclic. Each NSC code has its equivalent recursive systematic convolutional (RSC) code. In the field of $\text{GF}(2)$, if the NSC code is $\mathbf{G}(D) = [g_1(D) \ g_2(D)]$, then the generating matrix of the RSC code is $\mathbf{G}_1(D) = [1 \ g_2(D)/g_1(D)]$. Because the code space remains the same, the \mathbf{H} matrix of the RSC code is the same as that of the NSC code. For a constituent RSC code in a turbo code, the dimension of \mathbf{H} is determined by K . Generally there are K parity-check equations and therefore K rows in \mathbf{H} before puncturing.

C. Puncturing

Puncturing is frequently used to increase the coding rate. The puncturer deletes some of the parity bits. Those columns in \mathbf{H} corresponding to these bits should not be included in any parity check equation. Puncturing reduces the number of parity-check equations, as well as the number of rows in the parity-check matrix. For example, if the puncturing rule is to delete every other parity bit, and the original codeword is $\mathbf{c} = [x_0 \ p_0 \ x_1 \ p_1 \ x_2 \ p_2 \ x_3 \ p_3 \ x_4 \ p_4]$, then the puncturing result is $\mathbf{c}' = [x_0 \ p_0 \ x_1 \ x_2 \ p_2 \ x_3 \ x_4 \ p_4]$. Using \mathbf{H} in (1) as an example, the new parity-check matrix is

$$\mathbf{H}' = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ & & & & & & \cdots & & & & & \end{bmatrix}. \quad (2)$$

The number of rows in \mathbf{H}' reduces to one half of the original \mathbf{H} . The density of the parity-check matrix is also increased after puncturing, especially the number of 1's in every row.

D. Permutation and interleaving

The parity-check matrix \mathbf{H} , as shown in (1), is quasi-cyclic. Therefore any two-bit shift of a valid codeword still satisfies all parity-check equations. This is undesirable for frame synchronization because the synchronizer needs to distinguish the correct frame starting point from other positions. Fortunately, the interleaver and the scrambler permute the bit sequence so

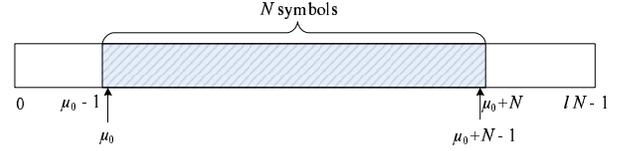


Fig. 2. The buffer structure.

that the resulting codeword is no longer cyclic. At the receiver, the bit sequence is rearranged to recover its original order.

Let \mathbf{c}_0 be the original codeword before the scrambler, and \mathbf{H}_0 be the corresponding parity-check matrix. The permutation is an elementary operation

$$\mathbf{c} = \mathbf{c}_0 \mathbf{P} \quad (3)$$

The new parity-check matrix is then $\mathbf{H} = \mathbf{P}^{-1} \mathbf{H}_0$.

III. FRAME SYNCHRONIZER

A. Packet Transmission

Packet-transmission in additive white Gaussian noise (AWGN) channel is considered. A frame of data is encoded, transmitted, and received. The signal is corrupted by additive white Gaussian noise. Let $\mathbf{d} = \{d_i\}$ denote the transmitted signal, and $\mathbf{w} = \{w_i\}$ be i.i.d. additive Gaussian noise. The received signal is

$$\mathbf{y} = \mathbf{d} + \mathbf{w}. \quad (4)$$

Assuming each symbol is sampled once with perfect symbol timing synchronization, the samples of the received signal are stored in a buffer as shown in Fig. 2. The location μ_0 where the codeword starts is unknown. It is assumed that the codeword is completely contained in the buffered samples. This assumption is valid if a coarse frame estimator is available, for example, by using the carrier power sensor as in [8]. The buffer size is lN , where N is the codeword length and $l > 1$ is the normalized observation window size. The problem is to estimate μ_0 , $0 \leq \mu_0 \leq lN - N$, from the whole frame of samples $\mathbf{y} = \{y_i\}$, $0 \leq i < lN$. If the estimate $\mu = \mu_0$, then frame synchronization is achieved. Otherwise, there is a failure.

B. Optimum Synchronization

The frame synchronizer examines \mathbf{y} against two hypotheses for each μ , $0 \leq \mu \leq lN - N$. Hypothesis \mathcal{H}_1 is that frame synchronization is achieved. The null hypothesis \mathcal{H}_0 is that there exist cycle slips of a few symbols. An optimum MAP estimator maximizes the probability of μ when receiving the samples \mathbf{y} , $\text{Pr}[\mu | \mathbf{y}]$.

The following components of the samples are taken into account, including the existence of blanks, code structure, and uncorrelated additive noise which is independent of the data. We use $d_i = 0$ to represent a blank, where no real data is transmitted, and $d_i = \pm\sqrt{E_s}$ the antipodal signal when BPSK modulation is used. E_s is the energy per symbol. $E_s = rE_b$, where r is the code rate, and E_b is the transmitted energy per

information bit. \mathcal{C} is the set of all valid codewords in GF(2),

$$\mathcal{C} = \{\mathbf{c} : \mathbf{c}\mathbf{H}^T = \mathbf{0}\}. \quad (5)$$

Let $\tilde{\mathcal{C}}$ denote the modulated version of \mathcal{C} . If $d_i = 0$, then $y_i = w_i$, which contains only noise with zero mean and variance $\sigma^2 = N_0/2$, where N_0 is the one-sided power spectral density of additive noise.

The frame synchronizer establishes a set of parity check equations according to \mathbf{H} . Each parity check equation is used to compute the probability that an even number of 1's have been transmitted for the subset of the bits that participate in the equation. Using Tanner's graphical representation [11], \mathbf{H} defines a bipartite graph where check nodes compute the probability of an even number of 1's in their adjacent variable nodes. In logarithm domain, the results are log-likelihood ratios (LLR).

The sum of LLR values is described by a random process $e(\mu)$, which is a function of μ because the LLR values change for every possible value of μ . When \mathcal{H}_1 is true, $e(\mu)$ has a positive mean. Otherwise, $e(\mu)$ has a zero mean. $e(\mu)$ is approximately Gaussian because of the large number of check nodes [12] [9]. When $\mu = \mu_0$, $e(\mu) \propto \mathcal{N}(Mm_c, \kappa Mm_c)$, where M is the number of check nodes, and m_c is the mean of LLR of one check node when an even number of 1's are present. κ is a coefficient greater than 2. The distribution of $e(\mu)$ is expressed as

$$f_e(x) = \frac{1}{\sqrt{2\pi\kappa Mm_c}} \exp\left(-\frac{[x - Mm_c]^2}{2\kappa Mm_c}\right). \quad (6)$$

When $\mu \neq \mu_0$, $e(\mu) \propto \mathcal{N}(0, \kappa Mm_c)$.

The a posteriori probability to be maximized is

$$\begin{aligned} \Pr[\mu | \mathbf{y}] &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{lN-N} \prod_{i=0}^{\mu-1} \exp\left(-\frac{y_i^2}{2\sigma^2}\right) \\ &\cdot \prod_{i=\mu+N}^{lN-1} \exp\left(-\frac{y_i^2}{2\sigma^2}\right) \\ &\cdot \Pr\left[\{d_\mu, \dots, d_{\mu+N-1}\} \in \tilde{\mathcal{C}} | \mathbf{y}\right] \end{aligned} \quad (7)$$

The two products in (7) account for the blanks in head and tail, where only noise is present. The last term on the right-hand side is the probability of receiving a valid codeword. The calculation of this last term requires a decoder. The block diagram of the optimum frame synchronizer is depicted in Fig. 3. In logarithm-domain, the frame synchronizer computes the log-likelihood function

$$L(\mu) = -\sum_{i=0}^{\mu-1} \frac{y_i^2}{2\sigma^2} - \sum_{i=\mu+N}^{lN-1} \frac{y_i^2}{2\sigma^2} - \frac{[e(\mu) - Mm_c]^2}{2\kappa Mm_c}. \quad (8)$$

The optimum estimate of μ in MAP sense is

$$\bar{\mu} = \arg \max_{\mu} \{L(\mu)\} \quad (9)$$

The synchronizer is simplified by modifying (8). When SNR

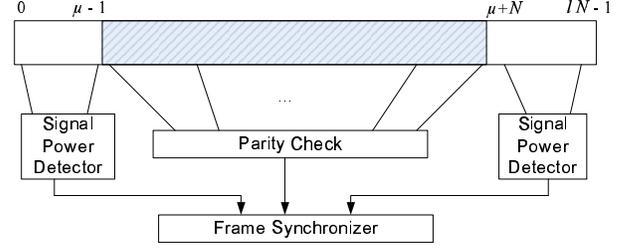


Fig. 3. Optimum MAP frame synchronizer.

is high, the conditional probability

$$\Pr[\{d_0, \dots, d_{\mu-2}\} = \mathbf{0} | d_{\mu-1} = 0] \approx 1, \quad (10)$$

therefore,

$$\Pr[\{d_0, \dots, d_{\mu-1}\} = \mathbf{0} | \mathbf{y}] \approx \Pr[d_{\mu-1} = 0 | \mathbf{y}]. \quad (11)$$

Likewise, we have

$$\Pr[\{d_{\mu+N}, \dots, d_{lN-1}\} = \mathbf{0} | \mathbf{y}] \approx \Pr[d_{\mu+N} = 0 | \mathbf{y}] \quad (12)$$

Therefore we obtain the high-SNR approximation as

$$L_{high}(\mu) = -\frac{y_{\mu-1}^2}{2\sigma^2} - \frac{y_{\mu+N}^2}{2\sigma^2} - \frac{[e(\mu) - Mm_c]^2}{2\kappa Mm_c}. \quad (13)$$

When SNR is low, the first two terms in (8) becomes constant because signals are "buried" in noise. Therefore the low-SNR approximation is valid

$$L_{low}(\mu) = -\frac{[e(\mu) - Mm_c]^2}{2\kappa Mm_c}. \quad (14)$$

Furthermore, if the occurrence of the events $\{\mu' | e(\mu') > Mm_c, \mu' \neq \mu_0\}$ is negligible, then the following likelihood function is viable,

$$L'_{low}(\mu) = e(\mu). \quad (15)$$

IV. SIMULATION STUDY

Packet transmission of turbo codes in AWGN channels is simulated with BPSK modulation. Two families of turbo codes are tested with random interleaver and scrambler. One family of codes has constraint length $k_c = 3$. The corresponding generating polynomial is (7, 5). The constraint length of the other family is $k_c = 4$ and the octal representation of the generating polynomial is (15, 13). The interleaver sizes considered are $K = 512$ and $K = 1024$ respectively. The likelihood function defined in (15) and $\kappa = (2W_r - 1)$ is used, where W_r is the row weight of \mathbf{H} . A sync failure is counted when the decision made by the frame synchronizer is not the same as the actual frame starting point.

Simulation results of rate 1/3 turbo codes are plotted in Fig. 4. The sync failure rate curves show that the failure rate is related to both the interleaver size and constraint length. Generally, the failure rate grows when the density of \mathbf{H} increases. Longer constraint length corresponds to an \mathbf{H} with higher density. The interleaver size determines the number of check nodes. If the weight of rows and columns in \mathbf{H} stays

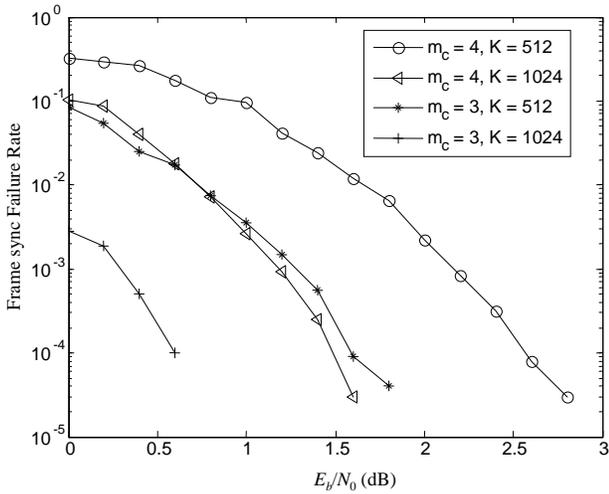


Fig. 4. Frame sync failure rate of turbo codes with code rate 1/3. Constraint lengths of constituent RSC codes are $k_c = 3$ and $k_c = 4$ respectively. Random interleavers and scramblers are used.

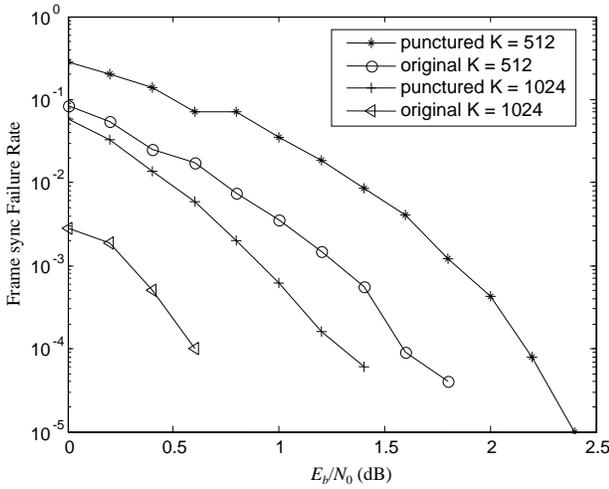


Fig. 5. Frame sync failure rate of punctured turbo codes and original turbo codes that are not punctured. The punctured codes have code rate 1/2. Constraint lengths of constituent RSC codes are $k_c = 3$. Random interleavers and scramblers are used.

the same, a greater interleaver size leads to lower density of \mathbf{H} . Therefore the frame sync failure rate is lower for codes with smaller constraint length and greater interleaver size. In all cases of interest, a frame sync failure rate lower than 10^{-4} is achieved when $E_b/N_0 < 2.5$ dB.

Fig. 5 compares the frame sync failure rate of punctured and original turbo codes that are not punctured. The constraint length is $k_c = 3$. Half of the parity bits are removed by the puncturer. Puncturing increases the density of \mathbf{H} . Hence it increases the sync failure rate as expected. Frame sync failure rates lower than 10^{-4} are achieved when $E_b/N_0 < 3$ dB.

Fig. 6 and Fig. 7 plot the frame error rates of turbo coded systems. The curves compare the systems with perfect frame synchronization to systems using MAP frame synchronizers. The turbo codes that generate Fig. 6 have $k_c = 3$. the proposed frame synchronizer has performance almost the same

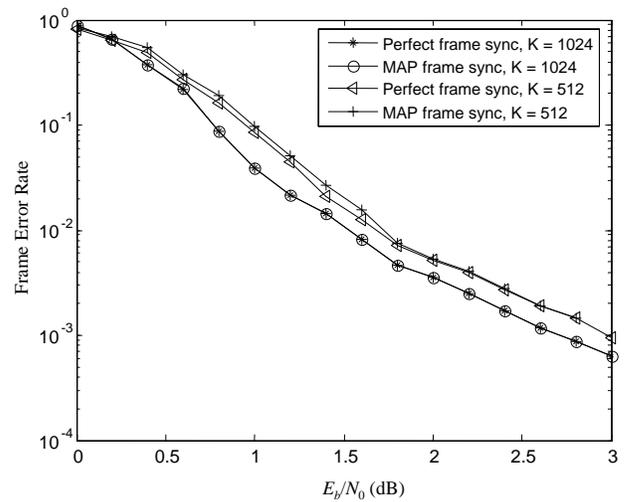


Fig. 6. Frame error rate of turbo codes with code rate 1/3. Constraint lengths of constituent RSC codes are $k_c = 3$. Random interleavers and scramblers are used.

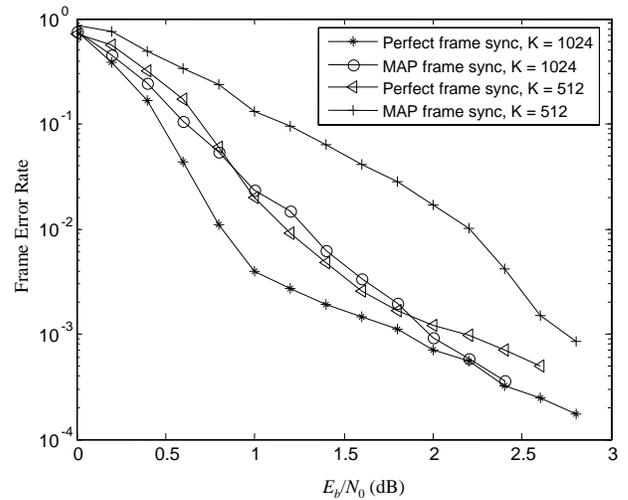


Fig. 7. Frame error rate of turbo codes with code rate 1/3. Constraint lengths of constituent RSC codes are $k_c = 4$. Random interleavers and scramblers are used.

as perfect frame synchronization.

The turbo codes that generate Fig. 7 have $k_c = 4$. When $K = 512$, the greatest gap between the curves is less than 1 dB and the curves converge when $E_b/N_0 > 2.5$ dB. When $K = 1024$, the largest gap between the curves is about 0.5 dB and converges when $E_b/N_0 > 2$ dB.

Fig. 8 shows the frame error rate performance of punctured turbo codes. Every other parity bits in \mathbf{y}^1 and \mathbf{y}^2 are punctured to increase the code rates to 1/2. The interleaver size is 1024. It is shown that punctured turbo codes are weaker than the original codes. The performance of the proposed frame synchronizer is also affected by puncturing. When $k_c = 4$, the greatest gap between the curves of the system using the proposed synchronizer and the system with perfect synchronization is about 1 dB and the curves converge when $E_b/N_0 > 3.4$ dB. When $k_c = 3$, the curves of the system

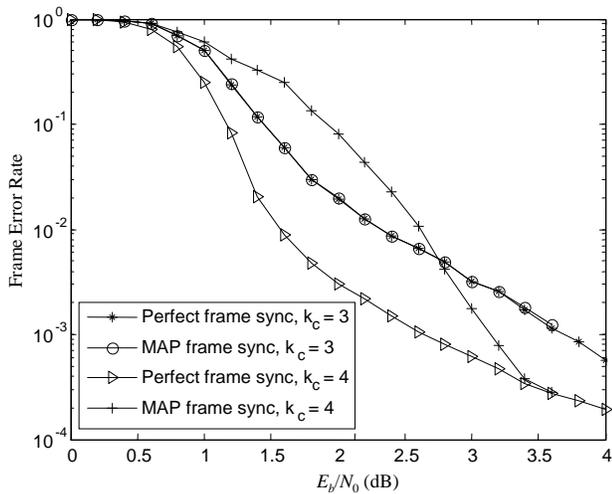


Fig. 8. Frame error rate of turbo codes with code rate 1/2, interleaver size = 1024. Random interleavers and scramblers are used.

using the proposed synchronizer and the system with perfect synchronization overlap. Hence when $k_c = 3$, the proposed frame synchronizer has negligible effects on the performance of the overall system.

V. CONCLUSIONS

An optimum frame synchronization technique for turbo coded packet-transmission system is proposed. The MAP method minimizes the frame sync failure rate. The coding structure of turbo codes is considered jointly with frame synchronization. The performance of the proposed frame synchronizer is determined by the sparseness of the parity-check matrix. The frame sync failure rate is lower for codes with smaller constraint length and greater interleaver size. Puncturing increases the density of the parity-check matrix, thus increasing the frame sync failure rate. The simulation

results show that frame sync failure rates lower than 10^{-4} are achieved at E_b/N_0 less than 3 dB for considered codes. The frame error rates of turbo coded systems using the proposed frame synchronizer are close to the rates of systems with perfect frame synchronization.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes(1)," in *Proc., IEEE Int. Conf. on Commun.*, (Geneva, Switzerland), pp. 1064–1070, May 1993.
- [2] K. Engdahl and K. S. Zigangirov, "On the theory of low-density convolutional codes I," in *Probl. Peredach. Inform.*, vol. 35, pp. 12–28, Oct.-Nov.-Dec. 1999.
- [3] G. D. Forney, Jr., "Structural analysis of convolutional codes via dual codes," *IEEE Trans. Info. Theory*, vol. IT-19, pp. 512–518, July 1973.
- [4] G. L. Lui and H. H. Tan, "Frame synchronization for Gaussian channels," *IEEE Trans. Comm.*, vol. COM-35, pp. 818–829, August 1987.
- [5] P. Robertson, "Optimum frame synchronization of packets surrounded by noise with coherent and differentially coherent demodulation," in *Proc. International Conf. Comm.*, pp. 874–879, May 1994.
- [6] T. M. Cassaro and C. N. Georghiades, "Frame synchronization for coded systems over AWGN channels," *IEEE Trans. Comm.*, vol. 32, pp. 484–489, Mar. 2004.
- [7] S. Pietrobon, "Implementation and performance of a turbo/MAP decoder," *International Journal of Satellite Communications*, vol. 16, pp. 23–46, Jan.–Feb. 1998.
- [8] W. Matsumoto and H. Imai, "Blind synchronization with enhanced sum-product algorithm for low density parity-check codes," in *International Symposium on Wireless Personal Multimedia Communications*, vol. 3, pp. 966–970, Oct. 2002.
- [9] J. Sun and M. C. Valenti, *Synchronization for Capacity-Approaching Coded Communication Systems*. PhD thesis, West Virginia University, 2004.
- [10] R. Johannesson and K. S. Zigangirov, *Fundamentals of convolutional coding*. IEEE Press series on digital and mobile communication, IEEE Press, 1998.
- [11] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Info. Theory*, vol. IT-27, pp. 533–547, Sep. 1981.
- [12] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Info. Theo.*, vol. 47, pp. 657–670, Feb. 2001.