ML frame synchronization for turbo and LDPC codes

Henk Wymeersch, Marc Moeneclaey
DIGCOM research group, TELIN Dept., Ghent University
Sint-Pietersnieuwstraat 41, 9000 GENT, BELGIUM
E-mail: {hwymeers,mm}@telin.UGent.be

Abstract

This contribution introduces a novel maximum-likelihood (ML) frame synchronization technique for coded systems that exploits the code properties by accepting soft information from the MAP decoder. Simulation results are presented for turbo and LDPC codes, and are compared to performance results of a frame synchronizer that does not use the code properties. We show that code-aided frame synchronization is required for turbo codes and LDPC codes, in order to avoid either very long pilot sequences or a considerable overall BER degradation.

1 Introduction

In packet-based communications frames arrive at the receiver with random inter-arrival times. This implies the start of each individual packet must be determined. This problem is referred to as frame synchronization. The most common approach is inserting known symbols in the (unknown) data sequence. A Maximum Likelihood (ML) or Maximum A Posteriori (MAP) method may then be used to determine the frame starting point. Assuming an additive white gaussian noise (AWGN) channel and uncoded random data, the ML rule reduces [1,2] to the well-known maximum correlation rule when the SNR is low.

The frame synchronization problem is an example of *hypothesis testing*, requiring an exhaustive search. As a frame synchronization failure gives rise to the loss of entire packet, its probability of occurrence should be made sufficiently small. This is different from other types of synchronization, such as symbol synchronization and carrier synchronization, which are *estimation problems*, with small synchronization errors giving rise to small BER degradations.

Generally data is not transmitted *per se*, but encoded by means of a channel encoder with error-correction capabilities to protect against transmission errors. In this contribution we consider turbo codes and low-density parity-check (LDPC) codes. Turbo codes were introduced in [3] and have been widely investigated ever since. One of the key aspects of these codes is that they consist of two convolutional encoders separated by an interleaver. A MAP decoder decodes each of the constituent codes in an iterative fashion, with exchange of soft information between constituent decoders which operate according to the BCJR [4] algorithm. LDPC codes were originally investigated in [5]. Iterative decoding is performed using the sum-product algorithm [6].

In order to combine frame synchronization and decoding, various approaches have been proposed: [7] uses a list-based synchronizer and makes the pilot sequence part of the codeword, thus forcing the coder in a sequence of known states. The decoder verifies this sequence to determine whether or not frame synchronization is achieved. In [8] the so-called *path surface metric*, based on on the forward and backward metrics in the BCJR algorithm, is used for frame synchronization. The properties of this metric change when the decoder is not synchronized. Yet another approach is mentioned in [9], where it was observed that a frame synchronization failure reduces the amplitude of the so-called extrinsic log-likelihood ratios as compared to a synchronized decoder. In [10], termination symbols in the convolutional codes are taken into account in deriving the ML frame position.

In this paper we introduce a code-aided frame-synchronizer which accepts feedback from the decoder in the form of a posteriori probabilities. This frame synchronizer operates according to the ML principle, and therefore minimizes the probability of a frame synchronization failure. We show through simulation that the performance of this frame synchronizer for turbo codes and LDPC codes is superior as compared to more traditional frame synchronizers that make no use of the code properties.

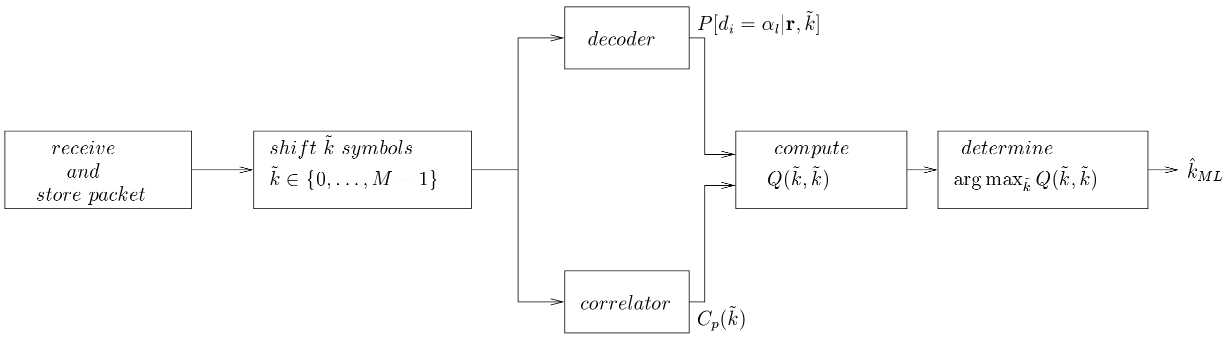

Figure 1: proposed frame synchronizer

2 Frame synchronization for AWGN channels

2.1 System description

We assume data symbols transmitted in frames ($\mathbf{a}$) of N symbols. Frames consist of a pilot sequence ($\mathbf{p}$) of length L and a coded data sequence ($\mathbf{d}$) of length $N-L$. The pilot sequence does not necessarily belong to the same alphabet as the data sequence. Some form of energy detection is assumed to roughly determine the arrival of a burst. The exact arrival time is assumed to be unknown up to a number of symbol intervals (M). The interval between bursts exceeds MT (where $1/T$ is the symbol rate) and contains only noise. This permits frames to be treated independently. Since this contribution does not deal with symbol and carrier synchronization, these are assumed to be perfect.

The received signal, at the input of the matched filter can be written as

$$r(t) = \sum_{n=0}^{N-1} a_n p(t - \tau - nT) + n(t) \qquad (1)$$

where $\mathbf{a} = [\mathbf{p}\,\mathbf{d}]$ is the vector of transmitted symbols and $\tau = kT$ is the propagation delay, $k \in \{0, \ldots, M-1\}$. The transmit pulse, $p(t)$, is a square-root cosine-roll-off unit energy transmit pulse with roll-off α and one-sided bandwidth $B = (1+\alpha)/(2T)$; $n(t)$ is a complex AWGN process with spectral density N_0/E_s, with E_s the energy per symbol. An M-PSK constellation is assumed though the results can easily be extended to other constellation types. Let us denote by $\mathbf{r}$ a random vector obtained by expanding $r(t)$ onto a suitable basis.

2.2 ML estimation and the EM algorithm

Assume we want to estimate a (discrete or continuous) parameter vector $\mathbf{b}$ from an observation vector $\mathbf{r}$ in the presence of a so-called nuisance vector $\mathbf{a}$. As these nuisance parameters are of no interest in the estimation process, we integrate them out. A maximum likelihood estimate of $\mathbf{b}$ maximizes the log-likelihood function:

$$\hat{\mathbf{b}}_{ML} = \arg\max_{\tilde{\mathbf{b}}} \left\{ \ln p\left(\mathbf{r} \middle| \tilde{\mathbf{b}}\right) \right\} \qquad (2)$$

where

$$p\left(\mathbf{r} \middle| \tilde{\mathbf{b}}\right) = \int_{\mathbf{a}} p\left(\mathbf{r} \middle| \mathbf{a}, \tilde{\mathbf{b}}\right) p(\mathbf{a})\, d\mathbf{a}.$$

Often $p\left(\mathbf{r} \middle| \tilde{\mathbf{b}}\right)$ is difficult to calculate. The EM-algorithm [11] is a method which iteratively solves (2). Defining the complete data $\mathbf{x} = [\mathbf{r}, \mathbf{a}]$, the EM-algorithm breaks up in two parts: the expectation part (Eq. 3) and the maximization part (Eq. 4):

$$Q\left(\tilde{\mathbf{b}}, \hat{\mathbf{b}}^{(n)}\right) = \int_{\mathbf{x}} p\left(\mathbf{x} | \mathbf{r}, \hat{\mathbf{b}}^{(n)}\right) \ln p(\mathbf{x} | \tilde{\mathbf{b}}) d\mathbf{x} \qquad (3)$$

$$\hat{\mathbf{b}}^{(n+1)} = \arg\max_{\tilde{\mathbf{b}}} \left\{ Q\left(\tilde{\mathbf{b}}, \hat{\mathbf{b}}^{(n)}\right) \right\}. \qquad (4)$$

It can be shown that $\hat{\mathbf{b}}^{(n)}$ converges to the ML estimate when the initial estimate is sufficiently close to the ML estimate. In the special case that $\mathbf{b}$ and $\mathbf{a}$ are independent, the estimation step becomes:

$$Q\left(\tilde{\mathbf{b}}, \hat{\mathbf{b}}^{(n-1)}\right) = \int_{\mathbf{a}} p\left(\mathbf{a} \middle| \mathbf{r}, \hat{\mathbf{b}}^{(n-1)}\right) \ln p\left(\mathbf{r} \middle| \mathbf{a}, \tilde{\mathbf{b}}\right) d\mathbf{a}. \qquad (5)$$

2.3 EM frame synchronization

The goal of our frame synchronizer is to determine the frame starting position (k) by means of the ML criterion, which is known to minimize the frame synchronization error probability $P\left[\hat{k} \neq k\right]$:

$$\hat{k}_{ML} = \arg\max_{\tilde{k}\in U}\left(\ln p\left(\mathbf{r}|\,\tilde{k}\right)\right) \tag{6}$$

where U is the set containing the M possible frame starting points, which are assumed to be equiprobable. Let us define the complete data as $\mathbf{x} = [\mathbf{r}, \mathbf{d}]$. Taking (1) into account, we obtain, with $\sigma^2 = N_0/\left(2E_s\right)$

$$\begin{aligned}
&\ln p\left(\mathbf{r}|\,\tilde{k}, \mathbf{d}\right) \\
&\propto -\frac{1}{2\sigma^2}\int_{-\infty}^{+\infty}\left|r\left(t\right) - \sum_n a_n p\left(t - \tilde{k}T - nT\right)\right|^2 dt \\
&\propto \sum_n \Re\left\{\int_{-\infty}^{+\infty} r^*\left(t\right) a_n p\left(t - \tilde{k}T - nT\right) dt\right\}.
\end{aligned}$$

Since $\mathbf{d}$ and k are independent, Eq. (5) then becomes

$$\begin{aligned}
&Q\left(\tilde{k}, \hat{k}\right) \\
&= E_{\mathbf{d}}\left[\ln p\left(\mathbf{r}|\,\tilde{k}, \mathbf{d}\right)\middle|\,\hat{k}, \mathbf{r}\right] \\
&= \sum_n \Re\left\{E_{\mathbf{d}}\left[a_n|\,\hat{k}, \mathbf{r}\right] y^*_{n+\tilde{k}}\right\}
\end{aligned}$$

where $y_{n+\tilde{k}} = \int_{-\infty}^{+\infty} r\left(t\right) p^*\left(t - \tilde{k}T - nT\right) dt$ is the matched filter output corresponding to time instant $nT + \tilde{k}T$. We break up $Q\left(\tilde{k}, \hat{k}\right)$ as follows:

$$Q\left(\tilde{k}, \hat{k}\right) = C_p\left(\tilde{k}\right) + C_d\left(\tilde{k}, \hat{k}\right) \tag{7}$$

where

$$C_p\left(\tilde{k}\right) = \sum_{i=0}^{L-1} \Re\left\{y^*_{i+\tilde{k}} p_i\right\} \tag{8}$$

and

$$C_d\left(\tilde{k}, \hat{k}\right) = \sum_{i=0}^{N-1} \Re\left\{y^*_{i+L+\tilde{k}} \mu_i\left(\mathbf{r}, \hat{k}\right)\right\} \tag{9}$$

wherein

$$\mu_i\left(\mathbf{r}, \hat{k}\right) = \sum_{\{\alpha_l\}} P\left[d_i = \alpha_l\,\middle|\,\mathbf{r}, \hat{k}\right]\alpha_l \tag{10}$$

denotes the a posteriori average of the data symbol d_i. Here $\{\alpha_l\}$ is the set of constellation points. $\mu_i\left(\mathbf{r}, \hat{k}\right)$ can be interpreted as a soft symbol decision: it is a weighted average of all possible constellation points. The symbol posterior probabilities in (10) can be provided by the MAP decoder. A conventional maximum correlation frame synchronizer would only calculate the terms $C_p\left(\tilde{k}\right)$, and determine the frame starting point according to:

$$\hat{k} = \arg\max_{\tilde{k}} C_p\left(\tilde{k}\right). \tag{11}$$

Eq. (7) can be interpreted as a generalized correlation rule: $\left(y_{\tilde{k}}, \ldots, y_{\tilde{k}+L-1}, y_{\tilde{k}+L}, \ldots, y_{\tilde{k}+L+N-1}\right)$ is correlated with $\left(p_0, \ldots, p_{L-1}, \mu_0, \ldots, \mu_{N-1}\right)$.

2.4 Solving convergence problems

According to the formulation (3)-(4), we should start with an initial frame position estimate $\hat{k}^{(0)}$, perform an exhaustive search to obtain a refined estimate $\hat{k}^{(1)}$, and further continue this procedure until convergence occurs. However, depending on the initial estimate, the EM algorithm might converge to a local instead of the global maximum of the likelihood function [12]. This convergence problem can be avoided by noticing that the ML estimate satisfies $\hat{k}^{(n+1)} = \hat{k}^{(n)} = \hat{k}_{ML}$. Hence, the ML frame position estimate satisfies the following relation:

$$\hat{k}_{ML} = \arg\max_{\tilde{k}} Q\left(\tilde{k}, \tilde{k}\right). \tag{12}$$

The corresponding synchronizer is shown in Fig. 1. It performs a single exhaustive search according to (12). The advantages with respect to the formulation (3)-(4) are that convergence to a local maximum is avoided and only one exhaustive search is required.

3 Performance results

The synchronization performance of any frame synchronizer is determined by the frame synchronization error rate, FSER, being the fraction of frames for which the frame synchronization fails. Denoting by $BER(M)$ the BER after frame synchronization when the frame starting position is unknown up to M positions, then $BER(0)$ is the BER under perfect frame synchronization. The resulting $BER(M)$ can be approximated by [7]:

$$\begin{aligned}
BER(M) &\approx BER(0)\left(1 - \text{FSER}\right) + \frac{\text{FSER}}{2} \\
&\approx BER(0)\left(1 + \frac{1}{2}\frac{\text{FSER}}{BER(0)}\right)
\end{aligned} \tag{13}$$

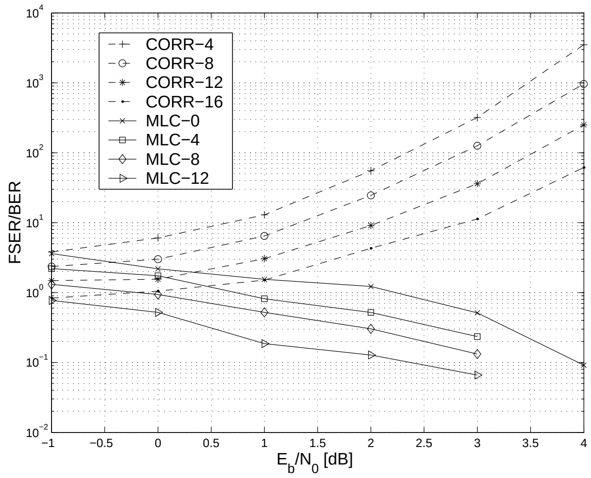

Figure 2: normalized FSER for a turbo code

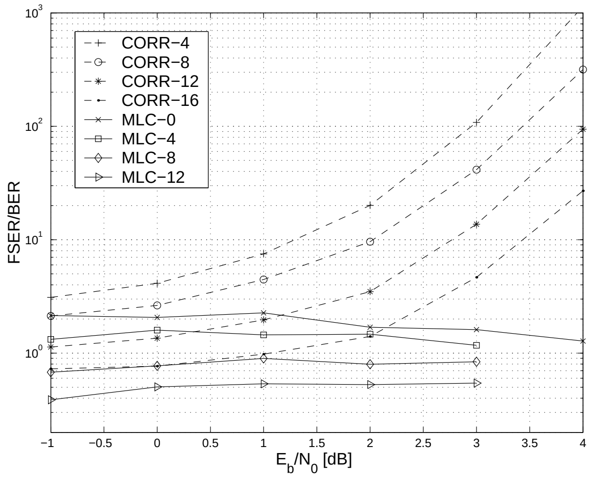

Figure 3: normalized FSER for an LDPC code

since in a non-synchronized frame roughly 50% of the decoded bits will be in error. In order to achieve low overall BER degradation we need to keep FSER around or below $BER(0)$. It is important to note that this implies that FSER must get smaller when the BER decreases. Consequently a better code requires a better frame synchronizer. Therefore a fair performance measure to evaluate frame synchronizers is the ratio FSER/$BER(0)$, which should be around or below 1. We call this ratio the normalized FSER. In the remainder we abbreviate $BER(0)$ by BER.

To compare the proposed ML code-aided frame synchronizer scheme to more conventional frame synchronizers, we have carried out computer simulations for turbo and LDPC coding, assuming BPSK mapping. The constituent convolutional codes of the turbo code are systematic and recursive with rate $1/2$, generator polynomials $(5,7)_8$ and constraint length 3. The turbo code consists of the parallel concatenation of two such encoders, which yields an overall code rate of $1/3$. The length of the information word is 45 bits, yielding a codeword of 135 BPSK symbols. The LDPC code (from [13]) is a rate $1/3$ code with an information sequence consisting of 32 bits. The number of decoding iterations for the turbo code and the LDPC code are 10 and 200, respectively. Although we have performed simulations for codes with short block lengths, it should be noted that the proposed scheme also works for more powerful codes working at very low SNR, as these codes inherently provide accurate soft symbol decisions, $\mu_i\left(\mathbf{r}, \hat{k}\right)$.

The ratio E_b/N_0 varies between -1 dB and $+4$ dB. The pilot sequence has a length varying from 0 to 16 symbols. Note that the energy per transmit symbol does not depend on the length of the pilot sequence. We assume the start of the frame to be unknown up to $M = 9$ symbol intervals. No optimization of the pilot sequence has been performed in term of its correlation properties; instead a new random pilot sequence is used for each burst. The performance is evaluated based on the normalized frame synchronization error rate (FSER/BER).

We have determined the frame synchronization performance resulting from the correlation frame synchronization algorithm (11) and our new ML frame synchronization algorithm (12) for coded data. The corresponding results are referred to as CORR-L and MLC-L in Figs. 2 and 3, where L denotes the number of pilot symbols.

The frame synchronizer performance results for the turbo and the LDPC code are shown in Fig. 2 and Fig. 3, respectively. As expected, the performance improves when the number of pilot symbols increases. We observe that the ratio FSER/BER (and, therefore, also the degradation of the BER caused by frame synchronization failures) increases with increasing E_b/N_0 when the code properties are ignored by the frame synchronizer, whereas the frame synchronizer that exploits the code properties exhibits the opposite behavior. Hence, the use of the latter algorithm is mandatory to avoid large BER degradations. It is interesting to note that when no pilot symbols are transmitted, the code-aided frame synchronizer still performs fairly well.

4 Conclusions and remarks

In this paper we have introduced an ML code-aided frame synchronizer for packet transmission, which exploits both the presence of the pilot sequences and the soft information provided by the MAP-decoder. We have evaluated its performance for both turbo codes and LDPC codes in terms of the ratio FSER/BER, and made a comparison with frame synchronizers that do not exploit code properties. We have shown that the latter frame synchronizers give rise to a large BER degradation caused by occasional synchronization failures, whereas the BER degradation is acceptable for the code-aided frame synchronizer, even when no pilot sequence is present. The proposed synchronizer thus increases the spectral efficiency of the transmission scheme.

Denoting the computation time per iteration of the MAP decoder by D and the number of decoder iterations by I, the total computational complexity of the proposed synchronizer is proportional to MDI. To decrease this complexity several approaches are possible:

- reduce I: the number of decoding iterations may be reduced (as far as frame synchronization is concerned)

- reduce M: a list-based synchronizer can be devised as follows. The correlator from Eq. (8) provides a list of the $M_l < M$ most likely values for k. The decoder works only with the shorter list

- reduce D: a suboptimal decoding algorithm may be used for frame synchronization, such as Max-Log-MAP or SOVA [14]

- an ad-hoc threshold may be used: once $Q\left(\tilde{k}, \tilde{k}\right)$ is above a certain value, the frame is assumed to be synchronized.

The impact of these complexity reductions on the frame synchronizer performance is a topic for further research.

The ML code-aided frame synchronization algorithm has been derived under the assumption that the considered frame is preceded and followed by noise only. The algorithm can be modified in a straightforward way to handle the case where the considered frame is preceded and/or followed by other frames.

In this paper we have assumed perfect carrier and symbol synchronization. In practice, carrier and symbol synchronization can be accomplished before frame synchronization by means of non-data-aided algorithms [15–17]. When frame synchronization has been obtained, the carrier and symbol synchronization estimates can be further refined by exploiting the code properties [18, 19].

Acknowledgement

This work has been supported by the Interuniversity Attraction Poles Program P5/11- Belgian State - Federal Office for Scientific, Technical and Cultural Affairs.

References

[1] J.L. Massey. "Optimum frame synchronization". *IEEE Trans. Comm.*, com-20(2):pp. 115–119, April 1972.

[2] G.L. Lui and H.H. Tan. "Frame synchronization for gaussian channels". *IEEE Trans. Comm.*, com-35(8):pp. 818–829, Aug. 1987.

[3] C. Berrou, A. Glavieux and P. Thitimajsima. "Near Shannon limit error-correcting coding and decoding". In *Proc. IEEE International Conference on Communications (ICC)*, pages 1064–1070, Geneva, Switzerland, May 1993.

[4] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv. "Optimal decoding of linear codes for minimising symbol error rate". *IEEE Trans. Inform. Theory*, 20:pp. 284–287, March 1974.

[5] R.G. Gallager. "Low density parity-check codes". *IRE Trans. Inform. Theory*, IT-8:pp. 21–29, Jan. 1962.

[6] D.J.C. MacKay. "Good error-correcting codes based on very sparse matrices". *IEEE Trans. Inform. Theory*, 45(2):pp. 399–431, March 1999.

[7] M.K. Howlader and B.D. Woerner. "Decoder-assisted frame synchronization for packet transmission". *IEEE journal on selected areas in comm.*, 19(12):pp. 2331–2345, Dec. 2001.

[8] J. Sodha. "Turbo code frame synchronization". *Signal Processing Journal, Elsevier*, 82:pp. 803–809, 2002.

[9] B. Mielczarek and A. Svensson. "Timing error recovery in turbo coded systems on AWGN channels". *IEEE Trans. Comm.*, 50(10):pp. 1584–1592, Oct. 2002.

[10] P. Robertson. "Improving frame synchronization when using convolutional codes". In *Proc. IEEE GLOBECOM*, pages 1606–1611, Houston, 1993.

[11] A.P. Dempster, N.M. Laird and D.B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". *Journal of the Royal Statistical Society*, 39(1):pp. 1–38, 1977. Series B.

[12] P. Spasojevic and C.N. Georghiades. "On the (non) convergence of the EM algorithm for discrete parameter estimation". In *Proc. Allerton Conference*, Monticello, Illinois, Oct. 2000.

[13] D.J.C. MacKay. "Online database of low-density parity check codes". http://www.inference.phy.cam.ac.uk/mackay /codes/data.html.

[14] J. Hagenauer and P. Hoeher. "A Viterbi algorithm with soft-decision outputs and its applications". In *Proc. IEEE Globecom*, pages 1680–1686, vol. 3, November 1989.

[15] H. Meyr, M. Moeneclaey, S.A. Fechtel. *"Synchronization, Channel Estimation, and Signal Processing"*, volume 2 of *"Digital Communication Receivers"*. John Wiley & Sons, 1997.

[16] A.J. Viterbi and A.M. Viterbi. "Nonlinear Estimation of PSK-modulated Carrier Phase with Application to Burst Digital Transmission". *IEEE Trans. Inform. Theory*, IT-29:pp. 543–51, Aug. 1983.

[17] M. Oerder and H. Meyr. "Digital filter and square timing recovery". *IEEE Trans. Comm.*, 36:605–611, May 1988.

[18] C. Herzet, R. Valéry, L. Vandendorpe and M. Moeneclaey. "EM algorithm-based timing synchronization in turbo receivers". In *Proc. ICASSP*, Hong Kong, April 2003.

[19] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M Luise and L. Vandendorpe. "Turbo-synchronization: an EM algorithm approach". In *Proc. IEEE ICC*, Anchorage, May 2003.