On Maximum-Likelihood Timing Synchronization

Cédric Herzet*, Henk Wymeersch, Marc Moeneclaey and Luc Vandendorpe*
TELIN Dept.,Ghent University, St.Pietersnieuwtraat 41, B-9000 Gent
*Communications Laboratory, Université Catholique de Louvain,
Pl. du Levant 2, B-1348 Louvain-la-Neuve
e-mail: {herzet,vandendorpe}@tele.ucl.ac.be, {hwymeers,mm}@telin.ugent.be

***Abstract*— In this paper we address the issue of symbol timing recovery for a coded burst transmission system. As direct Maximum Likelihood (ML) estimation is intractable, we resort to the Expectation-Maximization (EM) algorithm, in order to derive a receiver which iterates between data detection and synchronization. Conventional Data-Aided (DA) and Decision-Directed (DD) synchronizers can be interpreted as special cases of the proposed algorithm. The EM-based technique takes into account code properties and is especially well suited to scenarios where conventional schemes fail to provide the detector with a reliable timing estimate. The performance of the proposed algorithm is compared with conventional techniques through computer simulations, both in terms of Mean-Square Estimation Error (MSEE) and Bit Error Rate (BER).**

I. Introduction

In order to perform coherent data detection, a receiver has to estimate a number of synchronization parameters. In particular, timing recovery is an essential operation in digital communication systems [1]. Conventionally, timing synchronization for burst transmission is achieved through non-data-aided (NDA) estimators [2], commonly derived from the maximum-likelihood (ML) criterion. These estimators are acceptable for uncoded system, but may lead to unreliable estimates when state-of-the-art error-correcting codes are employed (this is due to the low-SNR operating point). In order to improve the estimator's performance, the underlying code may be exploited.

This idea has received some attention from the technical community in the last few years: reference [3], for instance, proposes to combine soft-decision directed carrier phase estimation with turbo decoding. In [4] the carrier phase synchronizer is embedded into a maximum a posteriori decoder and exploits the extrinsic information generated at each turbo iteration. In [5] soft information from the decoder is fed to a Mueller and Muller timing error detector in an iterative fashion. Another approach can be found in [6]: delay estimation is performed through soft-bit combining. A general framework for iterative code-aided (CA) synchronization was presented in [7], [8], based on on the expectation-maximization (EM) algorithm [9]. This principle was first applied to delay estimation for a transmission based on bit-interleaved coded modulation in [10], though no comparisons with conventional estimators were carried out.

In this paper, we will focus on the issue of symbol timing recovery for turbo coded systems. The proposed synchronizer will be derived from the EM algorithm and compared (through computer simulations) to a conventional NDA estimator.

II. System Model

As depicted in Fig. 1, we consider a burst transmission system where a sequence of L information bits is encoded by a channel encoder with code rate R. The resulting bits sequence is mapped to a signaling constellation $\mathcal{A}$ of size M and shaped by a unit energy square-root raised-cosine pulse $u(t)$ with roll-off α. The resulting baseband signal is then given by

$$s(t) = \sum_{k=0}^{K-1} a_k u(t - kT), \tag{1}$$

where $a_k \in \mathcal{A}$ is the k-th transmitted symbol, T is the symbol duration and K is the number of symbols in the burst. After propagation through an AWGN channel with delay τ, the received signal can be written as

$$r(t) = \sum_{k=0}^{K-1} a_k\, u(t - kT - \tau) + n(t), \tag{2}$$

where $n(t)$ is the complex envelope of an AWGN process with passband two sided power spectral density $N_0/2$. In coherent detection, the propagation delay τ needs to be estimated before data detection can take place.

III. Code-aided estimation

A. ML timing estimation

If we denote by $\mathbf{r}$ a vector representation of $r(t)$, the ML estimate of τ is given by

$$\begin{aligned}\hat{\tau}_{ML} &= \arg\max_{\tau} p(\mathbf{r}\,|\tau) \\ &= \arg\max_{\tau} \sum_{\mathbf{a}} p(\mathbf{r}\,|\tau, \mathbf{a})\, p(\mathbf{a}),\end{aligned} \tag{3}$$

where the sum in (3) goes over all possible transmitted sequence with a priori distribution $p(\mathbf{a})$. Generally, $p(\mathbf{a}) = 1/M^{KR}$ when $\mathbf{a}$ is a codeword and zero otherwise (i.e., a uniform prior within the code-space). As the summation in (3) gives rise to a computational burden that is exponential in K, ML estimation as defined in (3) cannot be used in practice.

B. The Expectation-Maximization Algorithm

The Expectation-Maximization algorithm, first introduced by Dempster, Laird and Rubin in [9], is an iterative method which enables the computation of maximum-likelihood estimates. This algorithm is actually well-suited to situations where the direct computation of the ML estimate is made intractable by the presence of unknown data. More formally, let Θ denote a parameter vector to be estimated, $\mathbf{r}$ an observation vector and $\mathbf{b}$ a nuisance vector. Defining the so-called *complete data* $\mathbf{x} = \left[\mathbf{r}^T, \mathbf{b}^T\right]^T$, the EM-algorithm iteratively executes the following two steps, starting from an initial estimate $\hat{\Theta}^{(0)}$: the Expectation step (Eq. 4) and the Maximization step (Eq. 5):

$$Q\left(\Theta \left| \hat{\Theta}^{(n)}\right.\right) = \int_{\mathbf{x}} p\left(\mathbf{x} \left| \mathbf{r}, \hat{\Theta}^{(n)}\right.\right) \ln p\left(\mathbf{x} \left| \Theta\right.\right) d\mathbf{x} \quad (4)$$

$$\hat{\Theta}^{(n+1)} = \arg\max_{\Theta} Q\left(\Theta \left| \hat{\Theta}^{(n)}\right.\right). \quad (5)$$

A crucial property of the EM algorithm is that the log-likelihood function $L(\Theta) = \log p(\mathbf{r}|\Theta)$ of successive estimates increases at each iteration as long as a stationary point (i.e., a global maximum, a local maximum or a saddle point) has not been reached [11]. The EM algorithm converges towards the global maximum of the log-likelihood function $\hat{\Theta}_{ML}$ provided that the initial estimate $\hat{\Theta}^{(0)}$ is sufficiently close to this maximum; otherwise, convergence to a different stationary point may occur.

C. Iterative ML Timing Estimation

We now make use of the EM algorithm to estimate the propagation delay τ. Let us define the complete data set as $\mathbf{x} = [\mathbf{r}^T, \mathbf{a}^T]^T$. Taking into account that the propagation delay τ is independent of the unknown transmitted sequence $\mathbf{a}$, the $\mathcal{Q}$-function in (4) reduces to

$$\mathcal{Q}\left(\tau \left| \hat{\tau}^{(n)}\right.\right) = \sum_{\mathbf{a}} p\left(\mathbf{a} \left| \mathbf{r}, \hat{\tau}^{(n)}\right.\right) \ln p\left(\mathbf{r} \left| \mathbf{a}, \tau\right.\right), \quad (6)$$

where[1]

$$\ln p\left(\mathbf{r} \right| \mathbf{a}, \tau) \propto -\int_{-\infty}^{+\infty} \left| r(t) - \sum_k a_k u(t - kT - \tau)\right|^2 dt$$
$$\propto \sum_{k=0}^{K-1} \Re\left\{ a_k^* \int_{-\infty}^{+\infty} r(t) u^*(t - kT - \tau) dt \right\}. \quad (7)$$

Substituting (7) into (6), we come up with the following Maximization step:

$$\hat{\tau}^{(n+1)} = \arg\max_{\tau} \sum_{k=0}^{K-1} \Re\left\{ \eta_k^*(\hat{\tau}^{(n)}) y(kT + \tau)\right\}, \quad (8)$$

where $y(kT + \tau)$ is the output of the matched filter at the instant $kT + \tau$, i.e.

$$y(kT + \tau) = \int_{-\infty}^{\infty} r(t) u^*(t - kT - \tau) dt \quad (9)$$

and

$$\eta_k(\hat{\tau}^{(n)}) = \sum_{a \in \mathcal{A}} a \times p\left(a_k = a \right| \mathbf{r}, \hat{\tau}^{(n)}\left.\right) \quad (10)$$

[1] $\propto$ stands for proportional to.

denotes the a posteriori expectation of the data symbol a_k, and is a weighted average of all possible constellation points. The marginal a posteriori probabilities (APPs) of the data symbols $p\left(a_k \right| \mathbf{r}, \hat{\tau}^{(n)}\left.\right)$, required to evaluate the Expectation-step, are delivered by a MAP decoder, operating according to the sum-product algorithm [12] (as in turbo and LDPC codes). The resulting estimator will be denoted EM CA (for Code-Aided). It is important to note that:

- When some of the data symbols are pilot/training symbols, the corresponding $p\left(a_k \right| \mathbf{r}, \hat{\tau}^{(n)}\left.\right)$ revert to Dirac distributions. Hence, the code-aided EM approach is an elegant extension of training-based algorithms, and can be seen as a soft-decision directed method.
- If we replace $\eta_k(\hat{\tau}^{(n)})$ by

$$\eta_k(\hat{\tau}^{(n)}) = \arg\max_{a \in \mathcal{A}} p\left(a_k = a \right| \mathbf{r}, \hat{\tau}^{(n)}\left.\right) \quad (11)$$

  we make hard instead of soft decisions w.r.t. the data symbols. This results in a Decision-Directed (DD) estimator. DD estimators are quite popular and provide a convenient (though ad-hoc) way of exploiting code-properties. Of course, at sufficiently high SNR, $p\left(a_k = a \right| \mathbf{r}, \hat{\tau}^{(n)}\left.\right)$ will tend to a Dirac distribution, so that (10) and (11) will yield the same performance. We denote this estimator by DD CA.
- It is also possible to use the EM algorithm or the DD algorithm without exploiting the code properties. In this case the marginal a posteriori probabilities $p\left(a_k = a \right| \mathbf{r}, \hat{\tau}^{(n)}\left.\right)$ in (10) and (11) are replaced by

$$p\left(a_k = a \right| \mathbf{r}, \hat{\tau}^{(n)}\left.\right) = \frac{p\left(y\left(kT + \hat{\tau}^{(n)}\right) \right| a_k = a, \hat{\tau}^{(n)}\left.\right)}{\sum_{\tilde{a} \in \mathcal{A}} p\left(y\left(kT + \hat{\tau}^{(n)}\right) \right| a_k = \tilde{a}, \hat{\tau}^{(n)}\left.\right)} \quad (12)$$

  which corresponds to the assumption that all M^K data symbol sequences $\mathbf{a}$ of length K have the same prior distribution. We denote the resulting estimators as EM NCA and DD NCA.

A major drawback of the proposed estimation algorithm is its computational overhead: for each EM iteration APPs need to be computed. When the detector itself is iterative (such as happens with turbo codes and LDPC codes), computation of the APPs is itself an iterative process. To reduce the computational complexity, we resort to a technique known as *embedded* estimation [13]: after each EM update of $\hat{\tau}$, we perform only a single decoding iteration, but maintain state information within the decoder from one EM iteration to the next. Hence, the EM and decoding iterations are intertwined, drastically reducing the computational overhead related to the EM estimation process.

IV. Performance Results

Simulation Set-up

In this section we evaluate the performance of different timing estimation algorithms. Performance is measured in terms of Mean Squared Estimation Error (i.e., MSEE=$E\left\{|\tau - \hat{\tau}|^2\right\}$). The MSEE is benchmarked against the Modified Cramer Rao Bound (MCRB), which is a lower bound for the MSEE for any unbiased estimator [14]. Additionally, we measure the BER performance, and compare it against a receiver with perfect knowledge of the propagation delay (denoted 'perfect sync'). We use a turbo encoder which consists of two rate $1/2$ RSC encoders with polynomial generators $(37, 33)_8$, separated by an interleaver. The turbo encoder output is punctured so that the overall code rate is $1/2$. The roll-off factor is set to $\alpha = 0.1$. Transmitted frames consist of 512 BPSK symbols. The timing offset is changed for each new transmitted frame and is assumed to be uniformally distributed in $[-0.5T, 0.5T]$. The receiver operates as follows (see Fig. 1): received signal $r(t)$ is passed through an ideal lowpass filter with bandwidth $1/(2T_s)$, sampled at a rate $1/T_s$, and passed through a discrete-time matched filter. The resulting samples are used to find $\hat{\tau}$, an estimate of τ. At that point, matched filter outputs at time instants $kT + \hat{\tau}$, for $k = 0, \ldots, K-1$ are computed by means of an interpolator. The iterative decoder performs 15 iterations to detect the data. As we will use embedded estimation, this corresponds to 15 EM iterations.

We will consider the following delay estimation algorithms: EM CA, EM NCA, DD CA and DD NCA. Of course, each of these algorithms requires an initial estimate, $\hat{\tau}^{(0)}$. For this purpose, we will use the well-known NDA Oerder&Meyr (O&M) estimator [15]. The Maximization-step (8) will be carried out by means of a Newton Raphson method.

Numerical results

Fig. 2 represents the performance of the different estimation algorithms in terms of MSEE. We consider both the case in which the CA synchronizers are initialized with an estimate delivered by an O&M estimator and the case in which the initial estimate is computed by an EM NCA iterative synchronizer (itself initialized by an O&M). The curve labeled "EM CA init EM NCA & O&M" for example represents the performance of an EM CA synchronizer initialized by an EM NCA estimator which is itself initialized by and O&M. Note that the MSEE of the NCA and O&M synchronizers do not depend on the code structure. The synchronization operation may be improved by considering an iterative EM or DD NCA estimator instead of the conventional O&M estimator. Let us now focus on the performance of the CA EM-based timing estimator. Since the turbo codes operate at low SNRs, we may notice that the exploitation of the code structure in the synchronizer leads to a gain in term of MSEE between the NCA and CA synchronizer performance. Observe that the EM CA initialized with an EM NCA synchronizer in Fig. 2 achieves the MCRB for $E_b/N_o \simeq 1.5$ dB. The importance of the initialization procedure may be emphasized by comparing the curves labeled "EM CA init O&M" and "EM CA init EM NCA & O&M". Indeed, we may observe that the synchronizer initialized with the EM NCA estimator exhibits better performance than the one simply initialized with the O&M. This may be explained by the fact that a good initial estimate will help the convergence of both the turbo decoder and the EM synchronizer.

Another view is offered in Fig. 3, where we display the BER achieved by the considered synchronizers versus E_b/N_o-ratio. At $E_b/N_o = 2.5$dB we observe that the EM CA synchronizer initialized by the O&M leads to worse performance than a synchronization with the EM NCA synchronizer. Yet, looking at the MSEE achieved by the two synchronizers we see that the EM CA synchronizer enables a better estimate quality than the EM NCA estimator. The difference in the BER achieved by the synchronizers may explained by the fact that the extrinsic information exchanged between the two constituent decoders may be degraded by a possible bad initial estimate. This bad initialization may prevent the overall turbo-decoder convergence even if the timing estimate is improved through the iterations. The EM NCA synchronizer which delivers a good initial estimate to the turbo decoder enables therefore to achieve better BER than an EM CA which initializes the turbo decoding with a bad-quality estimate. Considering now DD synchronizers we notice in Fig. 2 that the use of hard decision instead of soft decision does not lead to a significant degradation of the MSEE. We also observe in Fig. 3 that the use of DD synchronizers instead of EM synchronizers does not lead very large degradation of the BER. The quality of the initial estimate seems then to be the most important factor in the convergence behavior of the turbo system.

V. Conclusions

We proposed a timing synchronizer based on the expectation-maximization (EM) algorithm which is able to exploit code properties. Such a synchronizer may be applied in scenarios where conventional techniques fail. The synchronizer can operate in both a code-aided (CA) and Non-Code-Aided (NCA) mode. Both the CA and NCA mode approach the modified Cramer-Rao bound for large SNRs. At low SNRs, the CA synchronizer enables an improvement of the estimation quality in terms of MSEE with respect to an EM NCA estimator. Furthermore, the classical decision-directed (DD) estimators were shown to be an approximation of the proposed EM-based synchronizer. The use of hard decisions instead of soft decisions does not seem to lead to significant degradations of the performance. However, the MSEE of the DD synchronizer was shown by simulation to be lower-bounded by the MSEE of the EM synchronizer. We showed that the conventional O&M NDA synchronizer may give rise to unacceptable degradation of the BER for low roll-off factors. The proposed iterative estimators may be applied in this case to improve the estimation quality and consequently circumvent large BER degradations. Finally, we pointed out that the initial estimate delivered to an iterative receiver such as a turbo decoder is a crucial factor for the overall convergence. A poor initial estimate may prevent the turbo decoder to converge even if the estimation quality is further improved.

Acknowledgment

This work has been supported by the Interuniversity Attraction Poles Program P5/11 - Belgian Science Policy and the NEWCOM Network of Excellence, under the EU 6th Framework Program.

REFERENCES

[1] H. Meyr, M. Moeneclaey, and S. Fetchel. *Digital Communication Receivers : Synchronization, Channel Estimation and Signal Processing*. Wiley Series in Telecommunications and Signal Processing, USA, 1998.

[2] M. Morelli, A. D'Andrea and U. Mengali. "Feedforward ML-based timing estimation with PSK signals". *IEEE Trans. Comm.*, 1(3):80–82, May 1997.

[3] V. Lottici and M. Luise. "Carrier phase recovery for turbo-coded linear modulations". In *Proc. International Conference on Communications (ICC)*, New York, April 2002.

[4] W. Oh and K. Cheun. "Joint decoding and carrier phase recovery algorithm for turbo codes". *IEEE Comm. Letters*, 5(9):pp.375–377, September 2001.

[5] A.R. Nayak, J.R. Barry and S.W. McLaughlin. "Joint timing recovery and turbo equalization for coded partial response channels". *IEEE Trans. Magnetics*, 38(5):2295–2297, Sept. 2002.

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

[7] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise and L. Vandendorpe. "Turbo-synchronization: an EM algorithm interpretation". In *Proc. IEEE International Conference on Communications (ICC)*, Anchorage, May 2003.

[8] N. Noels, V. Lottici, A. Dejonghe, H. Steendam, M. Moeneclaey, M. Luise , L. Vandendorpe. "A Theoretical Framework for Soft Information Based Synchronization in Iterative (Turbo) Receivers". *EURASIP Journal on Wireless Communications and Networking JWCN, Special issue on Advanced Signal Processing Algorithms for Wireless Communications*, 2005(2):117–129, April 2005.

[9] 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.

[10] C. Herzet, V. Ramon, L. Vandendorpe and M. Moeneclaey. "EM algorithm-based timing synchronization in turbo receivers". In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, April 2003.

[11] C. F. J. Wu. "On the convergence properties of the EM algorithm". *Ann. Statistics*, 11(1):pp. 95–103, 1983.

[12] F. Kschischang, B. Frey and H.-A. Loeliger. "Factor graphs and the sum-product algorithm". *IEEE Trans. Inform. Theory*, 47(2):pp.498–519, February 2001.

[13] V. Lottici and M. Luise. "Embedding carrier phase recovery into iterative decoding of turbo-coded linear modulations". *IEEE Transactions on Communications*, 52(4):661–669, April 2004.

[14] A. D'Andrea, U. Mengali and R. Reggiannini. "The Modified Cramer-Rao Bound and Its Application to Synchronization Problems". *IEEE Trans. Comm.*, 42(2/3/4):1391–1399, February/March/April 1994.

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

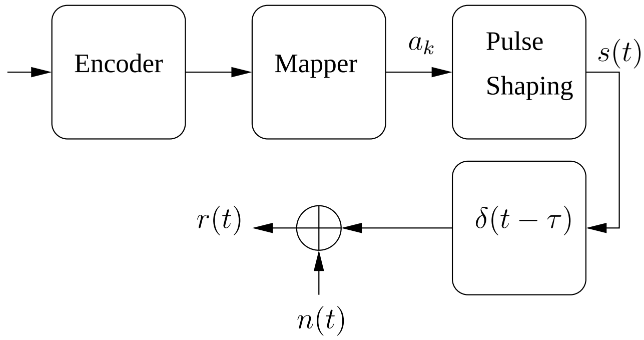

Fig. 1. Transmitter and receiver front end

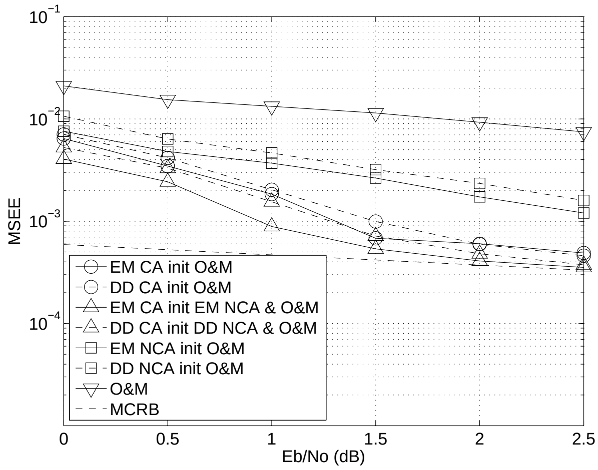

Fig. 2. MSEE achieved by the considered timing estimators vs E_b/N_o for turbo code and at roll-off 0.1.

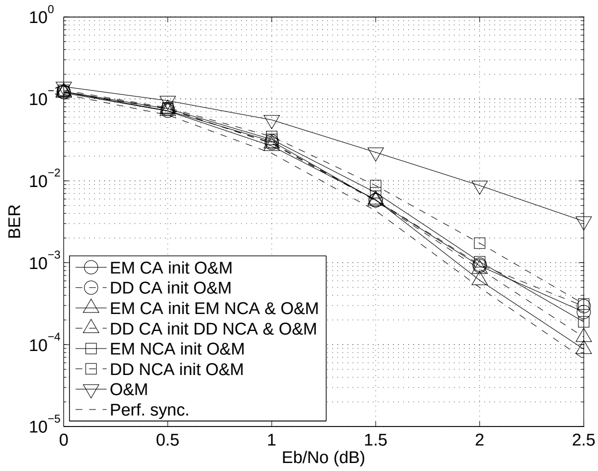

Fig. 3. BER vs E_b/N_o achieved by turbo code synchronized with different timing estimators and with roll-off=0.1.