Low-Cost Code-Aided ML Timing Recovery from Turbo-Coded QAM Transmissions

Faouzi Bellili*, Souheib Ben Amor†, Achref Methenni†, and Sofiène Affes†
*The Edward S. Rogers Sr. Dept. of Electrical and Computer Engineering, University of Toronto,
10 King's College Road, Toronto, ON, M5S 3G4, Canada.
†INRS-EMT, 800, De La Gauchetière West, Suite 6900, Montreal, QC, H5A 1K6, Canada.
Emails: faouzi.bellili@utoronto.ca, {souheib.ben.amor, methenni, affes}@emt.inrs.ca

***Abstract*—In this paper, we propose a new code-aided (CA) maximum likelihood (ML) approach for time synchronization in turbo-coded systems. The time delay estimate is refined at each turbo iteration owing to the increasingly accurate estimates for the log-likelihood ratios (LLRs) of the coded bits. The refined time delay estimate is then used by the matched filter (MF) in order to provide the soft-input soft-output (SISO) decoders with more reliable symbol-rate samples for the next turbo iteration. Simulation results show the remarkable performance improvements of CA estimation against the traditional non-data-aided (NDA) estimation scheme. Moreover, the new CA ML estimator (MLE) enjoys significant advantage in computational complexity over existing ML CA solutions.**

***Keywords*—Time synchronization, turbo codes, soft decoding, maximum likelihood.**

I. Introduction

Turbo codes along with high-order quadrature amplitude modulations (QAMs) play a crucial role in the current and future wireless communications. The widespread adoption of turbo codes is in part sustained by their ability to operate in the near-Shannon limit even under adverse SNR conditions [2]. Yet, the performance of these powerful error-correcting codes is prone to severe degradations if the system is not accurately synchronized in time, phase or frequency. Notably, time synchronization aims to estimate and compensate for the unknown time delay introduced by the channel so as to provide the decision device with symbol-rate samples of the lowest possible inter-symbol interference (ISI) corruption [3].

The problem of timing recovery for linearly-modulated transmissions has been heavily investigated in the literature and the vast majority of existing time delay estimators (TDEs) operate with *complete unawareness* of the code structure (see [4-9] and references therein). The latters, referred to as NDA TDEs, suffer from severe performance degradations under harsh SNR conditions since no *a priori* knowledge about the transmitted symbols is exploited during the estimation process. Alternatively, data-aided (DA) methods are more accurate and entail reduced computational burden. However, they require the transmission of a known pilot sequences which in turn limits the whole throughput of the system.

CA estimation lends itself as a middle ground scheme between NDA and DA estimations wherein the *soft information* extracted from the decoder is exploited during the estimation process at each turbo iteration. In fact, the two SISO decoders provide increasingly reliable information about the transmitted data from one turbo iteration to another; namely *a posteriori* LLRs of the code bits and their extrinsic information. According to the turbo principle, the latter are exchanged between the two SISO decoders until achieving a steady state wherein the *a posteriori* LLRs are used as decision metrics for data detection. In a nutshell, CA estimation consists in leveraging those soft outputs, by embedding the timing recovery task into the decoding process, in an attempt to improve the performance of the estimator and vice versa.

A number of CA timing recovery algorithms have been proposed over the last decade [10-15] and, to the best of the authors' knowledge, only two approaches are based on the ML criterion. The first ML solution [11] relies on the expectation-maximization (EM) algorithm and the second one [13] is a combined sum-product (SP) and EM algorithm approach. The SP-EM ML estimator offers indeed significant performance improvements over the original EM-based estimator but at the cost of increased computational complexity. In the SP-EM ML approach, an EM iteration loop is required under each turbo iteration wherein the algorithm switches between the so-called expectation step (E-STEP) and maximization step (M-STEP).

In this paper, we propose a new ML approach for CA timing recovery in turbo-coded systems. The new ML-based technique eliminates completely the need for the EM iteration loop under each turbo iteration. In other

Work supported by the Discovery Grants Program and a Discovery Accelerator Supplement Award of NSERC and the NSERC CREATE PERSWADE Research Training Program on Wireless <www.create-perswade.ca>. Most recent results in this line of work, disclosed herein and in a journal version [1], have never been published nor presented previously in any conference venue.

words, the new LF needs to be maximized only once per-turbo iteration after being updated by the associated *a priori* LLRs that are acquired from the output of the SISO decoders[1]. Consequently, the proposed CA timing recovery algorithm offers significant improvements in computational complexity over existing ML CA solutions.

We organize the rest of this paper as follows. In section II, we present the system model. In section III, we derive the closed-form expression of the log-likelihood function (LLF). In section IV, we introduce the new CA ML time delay estimator. In section V, we assess its performance by computer simulations. Finally, we draw out some concluding remarks in section VI.

Some of the common notations will be used throughout this paper. Vectors and matrices are represented in lower- and upper-case bold fonts, respectively. $\mathbf{I}_N$ denote, the $N \times N$ identity matrix. The shorthand notation $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{R})$ is used to denote the fact that the vector $\mathbf{x}$ follows a normal (i.e., Gaussian) distribution with mean $\mathbf{m}$ and auto-covariance matrix $\mathbf{R}$. In addition, $\{.\}^H$ and $\{.\}^T$ refer to the Hermitian and transpose of any vector or matrix, respectively. The operators $\Re\{.\}$ and $\Im\{.\}$ denote, respectively, the real and imaginary parts of any complex number. The operators $\{.\}^*$ and $|.|$ return its conjugate and its amplitude, respectively, and j is the pure complex number that verifies $j^2 = -1$. The Kronecker and Dirac delta functions are denoted, respectively, as $\delta_{m,n}$ and $\delta(t)$. We will also denote the probability mass function (PMF) for discrete random variables (RVs) by $P[.]$ and the probability density function (pdf) for continuous RVs by $p[.]$. The statistical expectation is denoted as $\mathbb{E}\{.\}$ and the notation $\triangleq$ is used for definitions.

II. System Model

Consider a turbo-coded system where a binary sequence of information bits is fed into a turbo encoder with two identical recursive and systematic convolutional codes (RSCs). The two RSCs are concatenated in parallel via an inner interleaver Π_1. The resulting code bits are injected into a puncturer which selects an appropriate combination of the parity bits in order to operate at the desired code rate R. The entire code bit sequence is scrambled with an outer interleaver, Π_2, then divided into K subgroups of $2p$ bits each (for some integer $p \geq 1$). The k^{th} subgroup of code bits, $b_1^k b_2^k \cdots b_l^k \cdots b_{2p}^k$, is conveyed by a symbol $a(k)$ that is selected from a fixed alphabet, $\mathcal{C}_p = \{c_0, c_1, \cdots, c_{M-1}\}$, of a $M-$ary (with $M = 2^{2p}$) QAM constellation (i.e., square-QAM). Each point, $c_m \in \mathcal{C}_p$, is mapped onto a unique sequence of $\log_2(M) = 2p$ bits denoted here as $\bar{b}_1^m \bar{b}_2^m \cdots \bar{b}_l^m \cdots \bar{b}_{2p}^m$, with respect to the Gray coding mechanism. The symbol c_m is selected to convey the k^{th} code bits subgroup [i.e., $a(k) = c_m$] if and only if $b_l^k = \bar{b}_l^m$ for $l = 1, 2, \cdots, 2p$. We also define the *a priori* LLR of the l^{th} code bit, b_l^k, conveyed by $a(k)$ as follows:

$$L_l(k) \triangleq \ln\left(\frac{P[b_l^k = 1]}{P[b_l^k = 0]}\right). \quad (1)$$

Taking into account the fact that $P[b_l^k = 0] + P[b_l^k = 1] = 1$, it can be shown that:

$$P[b_l^k = \bar{b}_l^m] = \frac{1}{2\cosh\big(L_l(k)/2\big)} e^{(\bar{b}_l^m - 1)\frac{L_l(k)}{2}}, \quad (2)$$

All the symbols, $\{a(k)\}_{k=1}^K$, are then pulse-shaped leading to the following transmitted signal:

$$x(t) = \sum_{k=1}^{K} a(k)\, h(t - kT), \quad (3)$$

with T being the symbol duration and $h(t)$ a unit-energy square-root raised cosine filter. We define the Nyquist pulse $g(t)$ obtained from the filter $h(t)$ as follows:

$$g(t) = \int_{-\infty}^{+\infty} h(x)h(t + x)dx, \quad (4)$$

which satisfies, at the same time, the first Nyquist criterion [3]:

$$g(nT) = \begin{cases} 0, & \text{for any integer } n \neq 0, \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

At the receiver side, we obtain the following (delayed) continuous-time received signal before matched filtering:

$$y(t) = \sqrt{E_s}\; x(t - \tau) \; + \; w(t), \quad (6)$$

where E_s refers to the transmit signal energy and τ is the unknown time delay parameter that needs to be estimated in this paper. In addition, $w(t)$ is a complex additive white Gaussian noise (AWGN) whose real and imaginary parts are independent and each having variance σ^2. The SNR of the channel is given by:

$$\rho \triangleq \frac{E_s}{N_0} = \frac{E_s}{2\sigma^2}. \quad (7)$$

When devising ML-type approaches, finding the LLF of the system is mandatory. This requires marginalizing the conditional LF over the constellation alphabet. For NDA estimators, the *a priori* information about the transmitted symbols is not available. Therefore, the latter are usually assumed to have equal *a priori* probabilities (APPs), i.e., $\forall\, c_m \in \mathcal{C}_p$, we have:

$$P[a(k) = c_m] = \frac{1}{M} \qquad \text{for} \;\; k = 1, 2, \cdots, K. \quad (8)$$

[1]Note that the *a priori* LLRs can also be deduced from LDPC-coded systems if they are decoded using turbo-like processing. There, the check nodes (C-nodes) and variable nodes (V-nodes) [16] play the very same role as SISO decoders in turbo-coded systems.

III. Derivation of the LLF

As widely known, the set of finite-energy signals:

$$\mathcal{L}^2_{\mathbb{R}} = \left\{ s(t) \text{ such that } \int_{\mathbb{R}} |s(t)|^2 dt < +\infty \right\},$$

form an infinite-dimension Hilbert subspace [17] for which one can find an orthonormal basis $\{\varphi_n(t)\}_n$ and an inner product as follows:

$$\langle s_1(t), s_2(t)\rangle = \int_{\mathbb{R}} s_1(t) s_2(t)^* dt, \ \ \forall\, s_1(t), s_2(t) \in \mathcal{L}^2_{\mathbb{R}}.$$

Therefore, an exact discrete representation for any continuous-time signal $s(t) \in \mathcal{L}^2_{\mathbb{R}}$ requires an infinite-dimensional vector, $\boldsymbol{s}$, that contains its expansion coefficients, $\left\{s_n = \langle s(t), \varphi_n(t)\rangle\right\}_n$, in the basis $\{\varphi_n(t)\}_n$. To avoid this problem, we first consider the truncated N-dimensional representation vectors:

$$\boldsymbol{y}_N = [y_1, y_2, \ldots, y_N]^T, \tag{9}$$
$$\boldsymbol{w}_N = [w_1, w_2, \ldots, w_N]^T, \tag{10}$$
$$\boldsymbol{x}_N(\tau) = [x_1(\tau), x_2(\tau), \ldots, x_N(\tau)]^T. \tag{11}$$

Those vectors contain the orthogonal projection coefficients of $y(t)$, $w(t)$, and $x(t-\tau)$, respectively, over the first N basis functions $\{\varphi_n(t)\}_{n=1}^N$ (for any $N \geq 1$), i.e.:

$$y_n = \big\langle y(t), \varphi_n(t)\big\rangle = \int_{\mathbb{R}} y(t)\varphi_n(t)^* dt, \tag{12}$$
$$w_n = \big\langle w(t), \varphi_n(t)\big\rangle = \int_{\mathbb{R}} w(t)\varphi_n(t)^* dt, \tag{13}$$
$$x_n(\tau) = \big\langle x(t-\tau), \varphi_n(t)\big\rangle = \int_{\mathbb{R}} x(t-\tau)\varphi_n(t)^* dt, \tag{14}$$

Plugging (6) and (12) in (14), it follows that:

$$\boldsymbol{y}_N = \sqrt{E_s}\, \boldsymbol{x}_N(\tau) + \boldsymbol{w}_N. \tag{15}$$

It can be shown that the noise coefficients, $\{w_n\}_{n=1}^N$, are Gaussian-distributed and that $\boldsymbol{w}_N \sim \mathcal{N}(\mathbf{0}_N, 2\sigma^2 \mathbf{I}_N)$. Therefore, the pdf of the vector $\boldsymbol{y}_N$ in (15) conditioned on the sequence of transmitted symbols, $\boldsymbol{a} = [a(1), a(2), \ldots, a(K)]^T$ and parametrized by τ is given by:

$$p(\boldsymbol{y}_N|\boldsymbol{a};\tau) = \prod_{n=1}^{N} \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{1}{2\sigma^2}\left|y_n - \sqrt{E_s} x_n(\tau)\right|^2\right\}. \tag{16}$$

Note here that, the transmitted symbols are indeed involved in (16) via the coefficients $\{x_n(\tau)\}_n$. By neglecting the terms that do not depend on τ in (16), we obtain the following *truncated* LF:

$$\Lambda(\boldsymbol{y}_N|\boldsymbol{a};\tau) = \exp\left\{\frac{\sqrt{E_s}}{\sigma^2}\sum_{n=1}^{N}\Re\left\{y_n x_n(\tau)^*\right\} - \frac{E_s}{2\sigma^2}\sum_{n=1}^{N}\left|x_n(\tau)\right|^2\right\}. \tag{17}$$

The conditional LF which incorporates all the information contained in the *non-truncated* vector $\boldsymbol{y}$ $\big[$or equivalently the received continuous-time signal $y(t)\big]$, is obtained by making N tend to infinity in (17). By doing so and using the Plancherel equality, the *conditional* LF can written as:

$$\Lambda(\boldsymbol{y}|\boldsymbol{a};\tau) = \exp\left\{\frac{\sqrt{E_s}}{\sigma^2}\int_{\mathbb{R}}\Re\{y(t)x(t-\tau)^*\}dt - \frac{E_s}{2\sigma^2}\int_{\mathbb{R}}|x(t-\tau)|^2 dt\right\}.$$

Now, replacing the transmitted signal $x(t)$ by its expression given in (3), and exploiting the fact that the shaping pulse, $g(t)$, in (4) verifies the first-order Nyquist criterion (5), it can be shown that:

$$\Lambda(\boldsymbol{y}|\boldsymbol{a};\tau) = \prod_{k=1}^{K} \Omega_\tau\big(a(k), y(t)\big), \tag{18}$$

where

$$\Omega_\tau\big(a(k), y(t)\big) \triangleq \exp\left\{\frac{\sqrt{E_s}}{\sigma^2}\int_{\mathbb{R}}\Re\{y(t)a(k)^*\}h(t-kT-\tau)dt - \frac{E_s}{2\sigma^2}\left|a(k)\right|^2\right\}.$$

The *unconditional* LF, $\Lambda(\boldsymbol{y};\tau)$, is obtained by averaging (18) over all possible transmitted symbol sequences of size K, i.e., $\Lambda(\boldsymbol{y};\tau) = \mathbb{E}_{\boldsymbol{a}}\{\Lambda(\boldsymbol{y}|\boldsymbol{a};\tau)\}$ leading to:

$$\Lambda(\boldsymbol{y};\tau) = \sum_{\boldsymbol{c}_i \in \mathcal{C}_p^K} P[\boldsymbol{a} = \boldsymbol{c}_i]\Lambda(\boldsymbol{y}|\boldsymbol{a} = \boldsymbol{c}_i;\tau). \tag{19}$$

Under coded digital transmissions, a simplifying assumption that is used in CA estimation practices postulates that the transmitted symbols are independent (cf. [10-15] and references therein) in spite of the statistical dependence between the coded bits, it follows that:

$$P\big[\boldsymbol{a} = \boldsymbol{c}_i\big] = \prod_{k=1}^{K} P\big[a(k) = \boldsymbol{c}_i(k)\big]. \tag{20}$$

Plugging (18) and (20) back into (19), it can be shown that:

$$\Lambda(\boldsymbol{y};\tau) = \prod_{k=1}^{K}\sum_{c_m \in \mathcal{C}_p} P\big[a(k) = c_m\big]\Omega_\tau\big(c_m, y(t)\big). \tag{21}$$

Therefore, the *unconditional* log-likelihood function (LLF), defined as $\mathcal{L}(\boldsymbol{y};\tau) \triangleq \ln\big(\Lambda(\boldsymbol{y};\tau)\big)$, is given by:

$$\mathcal{L}(\boldsymbol{y};\tau) = \sum_{k=1}^{K} \ln\Big(\bar{\Omega}_k\big(\tau, y(t)\big)\Big), \tag{22}$$

where $\bar{\Omega}_k\big(\tau, y(t)\big)$ is the average value of $\Omega_\tau\big(a(k), y(t)\big)$ over the constellation alphabet, i.e.:

$$\bar{\Omega}_k\big(\tau, y(t)\big) \triangleq \sum_{c_m \in \mathcal{C}_p} P\big[a(k) = c_m\big]\Omega_\tau\big(c_m, y(t)\big). \tag{23}$$

For ease of notations, we will henceforth no longer show the dependence of $\bar{\Omega}_k\big(\tau, y(t)\big)$ on the received signal,

$y(t)$, and denote it simply as $\bar{\Omega}_k(\tau)$. Moreover, it is shown in [1], that we obtain the following much useful factorization for $\bar{\Omega}_k(\tau)$:

$$\bar{\Omega}_k(\tau) = 4\beta_k F_{k,2p}\big(u_k(\tau)\big) F_{k,2p-1}\big(v_k(\tau)\big), \qquad (24)$$

where

$$\beta_k \triangleq \prod_{l=1}^{2p} \frac{1}{2\cosh\big(L_l(k)/2\big)}. \qquad (25)$$

The function $F_{k,q}(.)$ involved in (24) is given by:

$$F_{k,q}(x) = \sum_{i=1}^{2^{p-1}} \theta_{k,q}(i) e^{-\rho[2i-1]^2 d_p^2} \cosh\left(\frac{\sqrt{E_s}[2i-1]d_p}{\sigma^2}x + \frac{L_q(k)}{2}\right), (26)$$

in which the index q is either $2p$ or $2p-1$ depending on the context and d_p refers to half the minimum inter-symbol distance whose expression is given in [19, 20]. Both $\theta_{k,2p}(i)$ and $\theta_{k,2p-1}(n)$ are given by:

$$\theta_{k,2p}(i) \triangleq \prod_{l=1}^{p-1} e^{(2\bar{b}_{2l}^{(i)}-1)\frac{L_{2l}(k)}{2}}, \qquad (27)$$

$$\theta_{k,2p-1}(n) \triangleq \prod_{l=1}^{p-1} e^{(2\bar{b}_{2l-1}^{(n)}-1)\frac{L_{2l-1}(k)}{2}}. \qquad (28)$$

The two variables $u_k(\tau)$ and $v_k(\tau)$, involved in (24), denotes the matched-filtered *real* and *imaginary* parts of the received signal, respectively, i.e.:

$$u_k(\tau) = \int_{-\infty}^{+\infty} \Re\{y(t)\} h(t-kT-\tau)dt, \qquad (29)$$

$$v_k(\tau) = \int_{-\infty}^{+\infty} \Im\{y(t)\} h(t-kT-\tau)dt. \qquad (30)$$

Now, by using (24) back into (22) and dropping the constant term $4\beta_k$ which does not depend on τ, the useful LLF develops into:

$$\mathcal{L}(\boldsymbol{y};\tau) = \sum_{k=1}^{K} \ln\Big(F_{k,2p}\big(u_k(\tau)\big)\Big) + \sum_{k=1}^{K} \ln\Big(F_{k,2p-1}\big(v_k(\tau)\big)\Big). \ (31)$$

IV. New Time Delay CA ML Estimator

As mentioned previously, the timing recovery task is embedded into the turbo iteration loop. But in order to initiate the turbo decoding process itself, some preliminary digital symbol-rate samples are required. The latter can be obtained at the output of the MF (corrected with $\widehat{\tau}_{\text{ML-NDA}}$) where $\widehat{\tau}_{\text{ML-NDA}}$ is the NDA MLE for the TD parameter obtained as:

$$\widehat{\tau}_{\text{ML-NDA}} = \underset{\tau}{\operatorname{argmax}}\, \mathcal{L}^{(0)}(\tau), \qquad (32)$$

with $\mathcal{L}^{(0)}(.)$ being the NDA LLF deduced directly from (31) by setting[2] $L_l(k) = 0$ for all l and k, i.e.:

$$\mathcal{L}^{(0)}(\tau) = \sum_{k=0}^{K-1} \Big[\ln\big(F\big(u_k(\tau)\big)\big) + \ln\big(F\big(v_k(\tau)\big)\big)\Big], \ (33)$$

Here, $F(.)$ is expressed as follows:

$$F(x) = \sum_{i=1}^{2^{p-1}} e^{-\rho N_a d_p^2 [2i-1]^2} \cosh\left(\frac{2S[2i-1]\sqrt{N_a}d_p}{\sigma^2}x\right).$$

Note here that we adopt an iterative scheme to maximize $\mathcal{L}^{(0)}(\tau)$ with respect to τ in (32). Further details about the iterative technique will be provided at the end of this section. Note also that $u_k(\tau)$ and $v_k(\tau)$ involved in (32) are the real and imaginary components of a discrete-time MF output that is obtained as follows. At the receiver side, $y(t)$ is upsampled using a sampling period $T_s < T/(1+\alpha)$ with α being the roll-off factor to yield:

$$y_l \triangleq y(lT_s) = \sqrt{E_s} \sum_{k=1}^{K} a(k)\, h(lT_s - kT - \tau) + w(lT_s).$$

These high-rate samples are then passed through a MF to obtain the symbol-rate samples:

$$y_k(\tau) = y_l \star h(lT_s - kT - \tau) = \sum_l y_l\, h(lT_s - kT - \tau)dt,$$

from which we obtain $u_k(\tau) = \Re\{y_k(\tau)\}$ and $v_k(\tau) = \Im\{y_k(\tau)\}$ which are then used in (33). Once $\widehat{\tau}_{\text{ML-NDA}}$ is acquired, the corresponding sequence of symbol-rate samples:

$$\mathbf{y}(\widehat{\tau}_{\text{ML-NDA}}) = \big[y_1(\widehat{\tau}_{\text{ML-NDA}}), y_2(\widehat{\tau}_{\text{ML-NDA}}), \ldots, y_K(\widehat{\tau}_{\text{ML-NDA}})\big]^T,$$

is passed to the soft demapper in order to find the so-called *bit likelihoods*:

$$\Lambda_l(k) \triangleq \ln\left(\frac{p\big[\mathbf{y}(\tau)\big|b_l^k = 1\big]}{p\big[\mathbf{y}(\tau)\big|b_l^k = 0\big]}\right). \qquad (34)$$

By exchanging the *extrinsic* information between the two SISO decoders, the *a posteriori* LLRs of the code bits:

$$\Upsilon_l(k) = \ln\left(\frac{P\big[b_l^k = 1\big|\mathbf{y}(\tau)\big]}{P\big[b_l^k = 0\big|\mathbf{y}(\tau)\big]}\right), \qquad (35)$$

are updated iteratively according to the turbo principle. We denote their values at the r^{th} turbo iteration as $\Upsilon_l^{(r)}(k)$. The convergence is achieved after R turbo iterations wherein $\Upsilon_l^{(R)}(k) \approx \Upsilon_l(k)$, for every l and k. In addition, the signs of those *a posteriori* LLRs are used to detect the transmitted bits. Yet, owing to the well-known Bayes' formula, we have:

$$P\big[b_l^k = 1\big|\mathbf{y}(\tau)\big] = \frac{p\big[\mathbf{y}(\tau)\big|b_l^k = 1\big] P\big[b_l^k = 1\big]}{p\big[\mathbf{y}(\tau)\big]}, \qquad (36)$$

[2]In the NDA case, the has no *a priori* information about the bits is available, i.e., $P[b_l^k = 0] = P[b_l^k = 1] = 1/2$ and thus $L_l(k) = 0$ for all l and k.

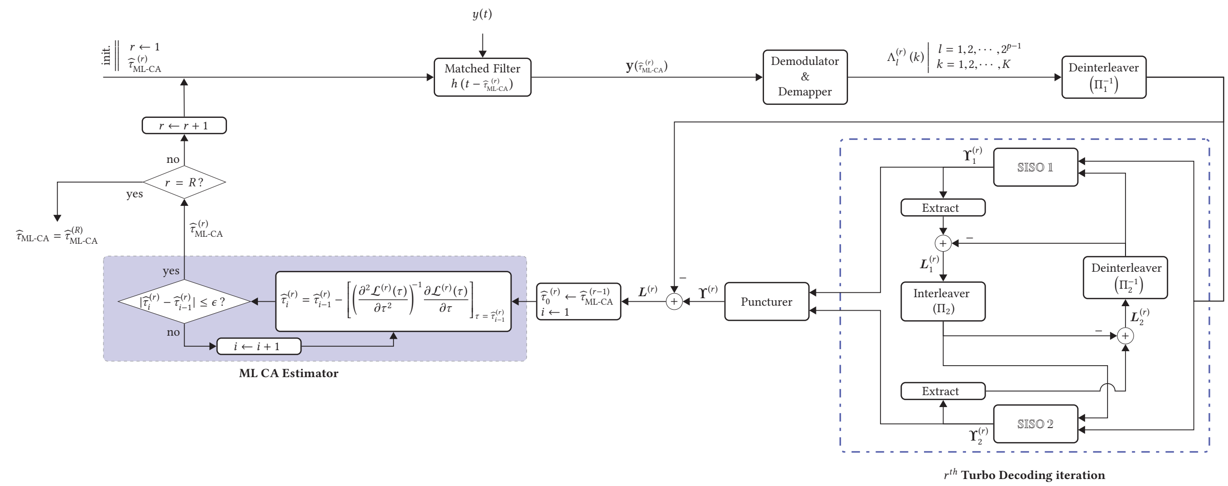

Fig. 1. Flowchart of the new CA TD ML technique.

and

$$P\big[b_l^k = 0\big|\mathbf{y}(\tau)\big] = \frac{p\big[\mathbf{y}(\tau)\big|b_l^k = 0\big]P\big[b_l^k = 0\big]}{p[\mathbf{y}(\tau)]}. \tag{37}$$

Therefore, by applying the natural logarithm to the ratio of (36) and (37), it follows:

$$L_l(k) = \Upsilon_l(k) - \Lambda_l(k) \approx \Upsilon_l^{(R)}(k) - \Lambda_l(k). \tag{38}$$

In other words, the required *a priori* LLRs of the code bits can be easily extracted from their steady-state *a posteriori* LLRs and $\Lambda_l(k)$ already computed by the *soft* demapper prior to data decoding. To exploit the output of the decoder and better re-synchronize the system, we modify (38) as follows:

$$L_l^{(r)}(k) = \Upsilon_l^{(r)}(k) - \Lambda_l^{(r-1)}(k). \tag{39}$$

By doing so, we obtain a more refined TD estimate, $\widehat{\tau}^{(r)}_{\text{ML-CA}}$ at the end of each r^{th} turbo iteration. Note here that $\Lambda_l^{(r-1)}(k)$ are the bit likelihoods that are obtained after re-synchronizing the system with $\widehat{\tau}^{(r-1)}_{\text{ML-CA}}$, i.e., the TD estimate corresponding to the previous turbo iteration. These bit likelihoods are fed to the SISO decoders to obtain an updated version of the *a posteriori* LLRs, $\Upsilon_l^{(r)}(k)$, at the r^{th} iteration. The refined TD MLE is thereof obtained as:

$$\widehat{\tau}^{(r)}_{\text{ML-CA}} = \underset{\tau}{\operatorname{argmax}}\ \mathcal{L}^{(r)}(\tau). \tag{40}$$

The CA LLF, $\mathcal{L}^{(r)}(\tau)$, involved in (40) is given by:

$$\mathcal{L}^{(r)}(\tau) = \sum_{k=1}^{K}\ln\Big(F_{k,2p}^{(r)}\big(u_k(\tau)\big)\Big) + \ln\Big(F_{k,2p-1}^{(r)}\big(v_k(\tau)\big)\Big),$$

in which $F_{k,q}^{(r)}(.)$ is given by:

$$F_{k,q}^{(r)}(x) = \sum_{i=1}^{2^{p-1}}\theta_{k,q}^{(r)}(i)e^{-\rho d_p^2[2i-1]^2}\cosh\left(\frac{\sqrt{E_s}[2i-1]d_p}{\sigma^2}x + \frac{L_q^{(r)}(k)}{2}\right),$$

with, $\widehat{\theta}_{k,2p}^{(r)}(i)$ and $\widehat{\theta}_{k,2p-1}^{(r)}(i)$ are also obtained by using $L_l^{(r)}(k)$ instead of $L_l(k)$ in (27) and (28), respectively.

Note that we still have to provide details about the maximization procedure of the NDA and CA LLFs. Actually, since these LLFs are expressed in a closed-form, they can be maximized using any iterative technique such as the Newton-Raphson algorithm:

$$\widehat{\tau}_i^{(r)} = \widehat{\tau}_{i-1}^{(r)} - \left[\left(\frac{\partial^2\mathcal{L}^{(r)}(\tau)}{\partial\tau^2}\right)^{-1}\frac{\partial\mathcal{L}^{(r)}(\tau)}{\partial\tau}\right]_{\tau = \widehat{\tau}_{i-1}^{(r)}}, \tag{41}$$

with $\widehat{\tau}_i^{(r)}$ is the TD update related to the i^{th} Newton-Raphson iteration. The algorithm converge to the CA TD MLE, $\widehat{\tau}^{(r)}_{\text{ML-CA}}$, once the criterion $|\widehat{\tau}_i^{(r)} - \widehat{\tau}_{i-1}^{(r)}| \leq \epsilon$ is met during the r^{th} turbo loop. It is worth noting that the Newton-Raphson algorithm itself is iterative in nature. Consequently, it requires a reliable initial estimate, $\widehat{\tau}_0^{(r)}$, to ensure its convergence to the global maximum of the underlying objective function. At each r^{th} turbo iteration, the algorithm is initialized by $\widehat{\tau}_0^{(r)} = \widehat{\tau}^{(r-1)}_{\text{ML-CA}}$. At the very first turbo iteration, however, we use the NDA MLE, $\widehat{\tau}_{\text{ML-NDA}}$, obtained in (32) as initial guess. The latter is the result of maximizing $\mathcal{L}^{(0)}(\tau)$ using the Newton-Raphson algorithm with initial guess obtained by broad line search over τ. For better illustration, Fig. 1 depicts the architecture of the newly proposed CA ML timing recovery algorithm.

V. Simulation Results

In this section, we provide some simulation results of the new CA ML TDE performance. We also analyze its computational complexity and compare it to that of the existing NDA and CA approaches. We consider an encoder with two identical RSCs concatenated in parallel. Those RSCs are equipped with the generator polynomials (1,0,1,1) and (1,1,0,1), and a systematic rate $R_0 = \frac{1}{2}$ each. The output of the turbo encoder is punctured in order to achieve the desired code rate R. For the tailing bits, the size of the RSC encoders memory is fixed to 4. We also consider a root-raised-cosine (RRC) signal with a roll-off factors $\alpha = 0.2$. The QPSK and 16-QAM, are adopted as two examples of square-QAM constellations. We assess the performance of the new TD CA ML estimator using the normalized (by T^2) mean square error (NMSE) as a performance measure:

$$\text{NMSE} \quad = \quad \frac{1}{T^2}\frac{\sum_{m=1}^{M_c}\left(\widehat{\tau}_{\text{ML-CA}}^{[m]} - \bar{\tau}\right)^2}{M_c}, \qquad (42)$$

where $\widehat{\tau}_{\text{ML-CA}}^{[m]}$ is the estimate of τ generated from the m^{th} Monte-Carlo run for $m = 1, 2, \ldots, M_c$.

In Figs. 2 and 3, we plot the NMSE of the new estimator obtained from $M_c = 5000$ Monte-Carlo trials, and benchmark the resulting performance curves against the corresponding CA Cramer-Rao lower bounds (CRLBs) of [18]. We also compare the new CA ML TDE against conventional NDA and CA techniques. In the NDA scenario, we consider both non-ML [8] and ML [9] benchmark solutions. In the CA scenario, however, we gauge the proposed CA estimator against two existing CA ML-type approaches, namely the Decision-Directed ML estimator introduced in [11] and SP-EM of [13].

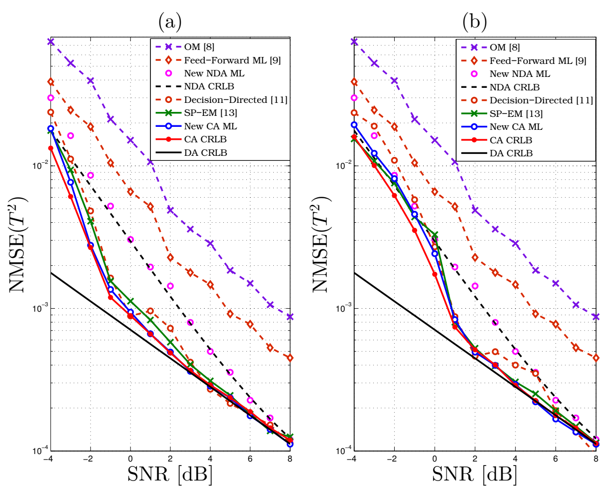

Fig. 2. NMSE of the estimators vs. the SNR, QPSK, roll-off $= 0.2$, for two different coding rates: (a) $R = 1/2$, and (b) $R = 1/3$.

As seen in Figs. 2 and 3, the performance of the conventional NDA algorithms is lower bounded by the NDA CRLB. Using the decoder output, however, the new estimator breaks this barrier and almost reaches the CA CRLB over the entire practical SNR range confirming thereby its statistical efficiency in practice. More so, it even coincides with the DA CRLB at high SNR values. In spite of its clear superiority over the conventional NDA techniques, the proposed CA estimator exhibits almost the same performance as the existing CA approaches at high SNRs, with a slight advantage at small SNRs. However, it has a great advantage owing to the huge computational savings it provides against the

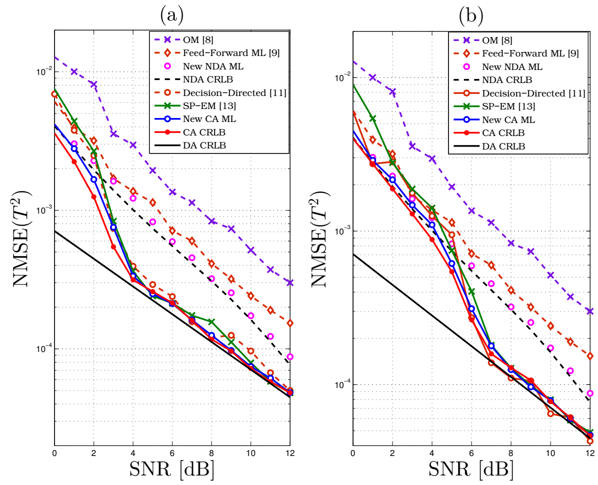

Fig. 3. NMSE of all the estimators vs. the SNR, 16-QAM, roll-off $= 0.2$, for two different coding rates: (a) $R = 1/2$, and (b) $R = 1/3$.

existing CA approaches. Indeed, Fig. 4 depicts the computational complexity of all the considered techniques (both NDA and CA). Clearly, the proposed CA ML estimator enjoys a remarkable computational advantage against the existing CA estimators while featuring the same if not better estimation performance as already seen in Fig 3. As expected, the NDA estimators entail smaller computational cost but they perform quite poorly in terms of estimation accuracy when compared to the CA schemes. From this perspective, we conclude that

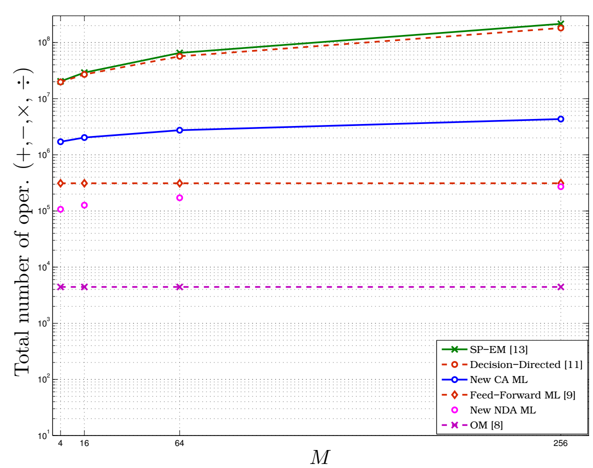

Fig. 4. Computational complexities of the estimators vs. the modulation order (M).

the proposed CA ML TDE provides the best trade-off between estimation performance and computational complexity as compared to state-of-the-art techniques.

VI. Conclusion

In this paper, we developed a new CA ML time delay estimator that is able to achieve the potential performance gains predictable by the CA CRLBs. Simulations results confirmed that the proposed algorithm is statistically efficient since it almost coincides with the CA CRLB. At high SNR values, it even reaches the DA CRLB, the ultimate bound that could be obtained if all the transmitted bits were perfectly known to the receiver beforehand. The new estimator also exhibits a remarkable advantage in terms of computational complexity as compared to the most powerful CA ML-type algorithms from the open literature.

References

[1] F. Bellili, A. Methenni, S. Ben Amor, S. Affes, and A. Stéphenne, "Time synchronization of turbo-coded square-QAM-modulated transmissions: code-aided ML estimator and closed-form Cramér-Rao lower bounds", accepted for publication in *IEEE Trans. Vehic. Techno.*, May 2017, to appear.

[2] C. Berrou, "The ten-year old turbo codes are entering into service", *IEEE J. Commun. Mag.*, vol. 41, no. 8, pp. 110-116, Aug. 2003.

[3] U. Mengali and A. N. D'andrea, *Synchronization Techniques for Digital Receivers,* New York: Plenum, 1997.

[4] J. Riba, J. Sala, and G. Vazquez, "Conditional maximum likelihood timing recovery: Estimators and bounds," *IEEE Trans. Signal Process.*, vol. 49, no. 4, pp. 835-850, Apr. 2001.

[5] A. Masmoudi, F. Bellili, S. Affes, and A. Stéphenne, "A non-data-aided maximum likelihood time delay estimator using importance sampling," *IEEE Trans. Signal Process.*, vol. 59 no. 10, pp. 4505-4515, Oct. 2011.

[6] A. Masmoudi, F. Bellili, S. Affes, and A. Stéphenne, "A new importance-sampling-based non-data-aided maximum likelihood time delay estimator," in *Proc. of IEEE WCNC'11.*, Cancun, Mexico, Mar. 2011, pp. 1682-1687.

[7] W. Gappmair, "Symbol-timing recovery with modified Gardner detectors," in *Proc. of 2nd International Symposium on Wireless Communication Systems*, Siena, Italy, Sep. 2005, pp. 831-834.

[8] M. Oerder and H. Meyr, "Digital filter and square timing recovery," *IEEE Trans. Commun.*, vol. 36, no. 5, pp. 605-612, Mar. 1988.

[9] M. Moeneclaey and G. de Jonghe, "Tracking performance comparisons of two feedforward ML-oriented carrier-independent NDA symbol synchronizers," *IEEE Trans. Commun.,* vol. 40, pp. 1423-1425, Sept. 1992.

[10] X. Man, H. Zahi, J. Yang, and E. Zhang, "Improved code-aided symbol timing recovery with large estimation range for LDPC-coded systems", *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 1008-1011, Apr. 2013.

[11] C. Herzet, H. Wymeersch, M. Moeneclaey, and L. Vandendorpe, "On maximum-likelihood timing synchronization," *IEEE Trans. Commun.*, vol. 55, no. 6, pp. 1116-1119, June 2007.

[12] J. Sun and M. C. Valenti, "Joint synchronization and SNR estimation for turbo codes in AWGN channels," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1136-1144, July 2005.

[13] C. Herzet, V. Ramon, and L. Vandendorpe, "A theoretical framework for iterative synchronization based on the sum-product and the expectation-maximization algorithms," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1644 - 1658, May 2007.

[14] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moenedaey, M. Luke, and L. Vandendorpe, "Turbo synchronization: an EM algorithm interpretation," in *Proc. of IEEE ICC*, Anchorage, Alaska, May 2003, pp. 2933-2937.

[15] N. Noels, H. Wymeersch, H. Steendam, and M. Moeneclaey, "True Cramér-Rao bound for timing recovery from a bandlimited linearly modulated waveform with unknown carrier phase and frequency," *IEEE Trans. Comm.,* vol. 52, no. 3, pp. 473-383, Mar. 2004.

[16] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.

[17] S. Mallat, *A Wavelet Tour of Signal Processing*, San Diego, CA: Academic, 1998.

[18] F. Bellili, A. Methenni, S. Ben Amor, and S. Affes, "Code-aided time synchronization of turbo-coded square-QAM-modulated transmissions: closed-form Cramer-Rao lower bounds," in *Proc. IEEE GLOBECOM*, San Diego, CA, USA, Dec. 2015.

[19] F. Bellili, A. Methenni, and S. Affes, "Closed-form CRLBs for SNR estimation from turbo-coded BPSK-, MSK-, and square-QAM-modulated signals," *IEEE Trans. Signal Process.,* vol. 58, no. 15, pp. 4018-4033, Aug. 2014.

[20] F. Bellili, A. Methenni, and S. Affes, "Closed-form CRLBs for CFO and phase estimation from turbo-coded square-QAM-modulated transmissions," *IEEE Trans. Wireless Commun.,* vol. 14, no. 5, pp. 2513-2531, May 2015.