Near Optimum Low Complexity Smoothing Loops for Dynamical Phase Estimation-Application to BPSK Modulated Signals

Jianxiao Yang, Benoit Geller

Near-MAP Smoothing Loops for Code Aided QAM Dynamical Carrier Phase Estimation

Names

***Abstract*—This paper proposes a near-maximum *a posteriori* smoothing algorithm for the dynamical carrier phase estimation of coded quadrature amplitude modulation (QAM) signals. This low-complexity near-optimum smoothing is obtained by averaging phase-locked loops (PLLs) with possibly the aid of the decoded *a posteriori* information. The proposed code-aided smoothing algorithm performs near the off-line Bayesian and hybrid Cramer-Rao bounds (BCRBs and HCRBs) of interest. It has a gain of several decibels compared to the conventional on-line loop and is able to track frequency offsets.**

***Index Terms*—Bayesian Cramer-Rao Bound (BCRB), Code-Aided (CA), Data-Aided (DA), Dynamical Phase Estimation, Hybrid Cramer-Rao Bound (HCRB), Maximum A Posteriori (MAP), Non-Data-Aided (NDA), Smoothing Phase-locked Loops (S-PLL)**

I. INTRODUCTION

Modern communications systems have to cope with more and more stressing conditions (low signal-to-noise ratios (SNRs), high data rate). As phase estimation is processed at the front-end of digital receivers, phase errors rapidly degrade the overall performance of communication systems and consequently phase synchronization has become one of the most critical tasks that a receiver has to operate. To solve this challenging phase synchronization problem, many signal processing techniques have been proposed. Among Bayesian methods, [1] proposed a non-coherent method based on a truncated memory, [2]-[4] executed message passing algorithms based on factor graphs and the algorithms proposed in [5],[6] are based on the variational Bayesian methods. [7] composed a partical filtering with a phase-locked loop technique and found some interest to partical filtering when there are abrupt changes in the phase trajectory. A BCJR based Gaussian sum smoother was proposed in [21] for convolutional turbo code aided dynamical phase estimation, and was derived by extending the original BCJR algorithm [26] to the case of joint phase estimation and decoding. However, in the general case, Bayesian methods are far more complex to implement than deterministic methods and thus deterministic phase algorithms have been employed so far in real systems. The most commonly used deterministic methods are based on the EM algorithm [8],[9] or on gradient-like phase locked loops (PLL) techniques. PLL are traditionally known as low-cost algorithms that lead to a good asymptote performance for on-line estimation [10]. Their performance can even be improved at low SNRs within the turbo-receiver concept [11]-[13]. Rather than considering actual on-line estimation for which estimated values only depend on past observations, one may also consider off-line estimators involving both past and future observations; such a data block approach is a very natural way to proceed, as modern systems use error correcting codes. Bayesian and hybrid Cramer-Rao bounds (BCRBs and HCRBs), associated to this dynamical phase synchronization problem, have been considered in some recent contributions [14]-[17] and they allow to measure the performance of such algorithms; they clearly show the superiority of the off-line approach, compared to the on-line approach. Contrarily to the performance of [11]-[13] limited by the on-line bound, [17] analyzed a smoothing PLL (S-PLL) algorithm based on two on-line PLLs that was proposed without any performance evaluation in [18]. [17] derived the S-PLL algorithm for a non-coded BPSK system and it is well known that it is not difficult to achieve synchronization in such a context. [19] and [20] extended the study of the S-PLL algorithm to other telecommunication formats.

In this paper, we extend the analysis of the S-PLL to the useful case in practice of channel encoded QAM modulated systems. We derive our algorithm from the definition of the code-aided (CA) a posteriori probabilities (APPs) and we then compare its performance with Cramer-Rao bounds of interest. However, we have to point out that the computation complexity of the true maximum *a posteriori* (MAP) algorithm for the CA scenario exponentially increases with the block length and is impossible to achieve in practice. Nevertheless, the near-MAP algorithm derived via some approximations is able to reach the lower bound over a very wide SNR range. Moreover, we prove the asymptotic convergence of the algorithm towards the Bayesian Cramer-Rao bound illustrate. We also illustrate the respective gains brought by the on-line/off-line and code-aided/non-data-aided scenarios.

The rest of the paper is structured as follows. The system model is described in Section II. After briefly reviewing the MAP estimation principle in Section III, a code-aided near-MAP estimation framework is derived in Section IV. In Section V, we prove the asymptotic convergence of the algorithm. Some practical considerations are then discussed in the next section and numerical results are presented in section VII, where we illustrate the algorithm performances and compare them to the Cramer-Rao bounds of interest. Finally, in Section VIII, some concluding remarks are made.

II. System Model

We consider the transmission of symbols s_l belonging to a constellation set $\mathbf{S}_M = \{\tilde{s}_1, \cdots, \tilde{s}_M\}$ (such as M-PSK or M-QAM) over an additive white Gaussian noise (AWGN) channel, additionally affected by some carrier phase offsets θ_l. The sequence of symbols are stacked in the symbols vector $\mathbf{s} = [s_1, \cdots, s_L]^T$ and the phase offsets are stacked in the vector $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_L]^T$. Assuming that the timing recovery is perfect, the sampled baseband signal $\mathbf{y} = [y_1, \cdots, y_L]^T$ does not suffer from any inter-symbol interference (ISI) and can thus be written as

$$y_l = s_l e^{j\theta_l} + n_l = (a_l + jb_l) e^{j\theta_l} + n_l, \tag{1}$$

where at time *l*, s_l is the l^{th} transmitted complex symbol, θ_l is the perturbing phase that must be estimated and the last term n_l in the right-hand side of (1) is a circular Gaussian noise with zero-mean and variance σ_n^2.

For the code-aided system, we assume that K bits are encoded into a codeword $\mathbf{c} = \{c_1, c_2, \cdots, c_N\} \in \{\tilde{\mathbf{c}}_v \mid v = 1, \cdots, 2^K\}$ of N bits which are further mapped as a constellation vector $\tilde{\mathbf{s}}(\tilde{\mathbf{c}}_v) = \{\tilde{s}_1(\tilde{\mathbf{c}}_v), \cdots, \tilde{s}_L(\tilde{\mathbf{c}}_v)\}$. In this scenario, the independent and identically distributed (*i.i.d.*) law among the transmitted symbols does not hold any more, instead, the *i.i.d.* law holds among all the 2^K elements of the codebook. So the conditional probability density based on the known phase vector $\boldsymbol{\theta}$ can be written as

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \sum_{v=1}^{2^K} p(\mathbf{y} \mid \mathbf{c} = \tilde{\mathbf{c}}_v, \boldsymbol{\theta})\, p(\mathbf{c} = \tilde{\mathbf{c}}_v) = \sum_{v=1}^{2^K} p(\mathbf{y} \mid \mathbf{s} = \tilde{\mathbf{s}}(\tilde{\mathbf{c}}_v), \boldsymbol{\theta})\, p(\mathbf{s} = \tilde{\mathbf{s}}(\tilde{\mathbf{c}}_v)) = \left(\frac{1}{\pi\sigma_n^2}\right)^L \sum_{v=1}^{2^K} \frac{1}{2^K} \prod_{l=1}^{L} \exp\left\{-\frac{|\tilde{s}_l(\mathbf{c}_v)|^2 + |y_l|^2}{\sigma_n^2}\right\} \exp\left\{2\frac{\mathrm{Re}\{y_l \tilde{s}_l^*(\mathbf{c}_v) e^{-j\theta_l}\}}{\sigma_n^2}\right\}. \tag{2}$$

We now consider the phase model. Due to the imperfections of the oscillator, the receiver suffers from jitter. Moreover, due to the lack of knowledge about the transmitter's clock, there exists a constant frequency shift in the receiver's clock. This results on a classical Brownian phase model with a linear drift

$$\theta_l = \theta_{l-1} + \xi + w_l \,, \tag{3}$$

where at time l , w_l is Gaussian distributed with zero mean and variance σ_w^2, ξ is the unknown constant frequency offset (linear drift) and θ_l is the unknown phase offset. This model [20]-[22] has been widely used to describe the behavior of practical oscillators for which the frequency is randomly perturbed. The corresponding conditional probability can be written as

$$p(\theta_l \mid \theta_{l-1}) = \frac{1}{\sqrt{2\pi}\sigma_\theta} \exp\left\{ -\frac{(\theta_l - \theta_{l-1} - \xi)^2}{2\sigma_\theta^2} \right\}. \tag{4}$$

III. MAP Estimation and Conditional Gradient Ascent Algorithm

The maximum *a posteriori* (MAP) estimation of the phase vector $\boldsymbol{\theta}$ given the observation $\mathbf{y}$ is given by

$$\hat{\boldsymbol{\theta}} \triangleq \arg\max_{\boldsymbol{\theta}} \left[\ln p(\mathbf{y},\boldsymbol{\theta}) \right] = \arg\max_{\boldsymbol{\theta}} \left[\ln p(\mathbf{y} \mid \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) \right], \tag{5}$$

where $p(\mathbf{y},\boldsymbol{\theta})$ may also be expressed as

$$p(\mathbf{y},\boldsymbol{\theta}) = \sum_{\mathbf{s}} p(\mathbf{y},\boldsymbol{\theta},\mathbf{s}) \,. \tag{6}$$

Computing $\hat{\boldsymbol{\theta}}$ is usually not a trivial problem and one can consider the following methods.

First-order Optimality Condition: The solution of (5) must also satisfy the following necessary condition of optimality

$$\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y},\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = 0 \,, \tag{7}$$

where $\nabla_{\boldsymbol{\theta}} = \left[\frac{\partial}{\partial\theta_1}, \cdots, \frac{\partial}{\partial\theta_L} \right]^T$. (7) defines therefore L necessary conditions of optimality.

Gradient Ascent Optimization: Consider the following iterative algorithm

1. Pick $l \in \{1, \cdots, L\}$
2. Update $\hat{\boldsymbol{\theta}}$ as follows

$$\hat{\theta}_m^{(i+1)} = \hat{\theta}_m^{(i)} + \gamma \cdot \frac{\partial}{\partial\theta_m} \ln p\left(\mathbf{y}, \left\{\hat{\boldsymbol{\theta}}^{(i)}\right\}\right), \tag{8}$$

$$\hat{\theta}_l^{(i+1)} = \hat{\theta}_l^{(i)}, \quad l \neq m \tag{9}$$

where $\gamma > 0$.

3. Repeat 1 and 2 until convergence.

It is easy to see that this algorithm is nothing but a conditional gradient ascent algorithm. By definition, $\hat{\boldsymbol{\theta}}^f$ is a fixed point of this algorithm *iff*

$$\frac{\partial}{\partial \theta_m} \ln p\left(\mathbf{y}, \hat{\boldsymbol{\theta}}^f\right) = 0, \quad \forall m \tag{10}$$

i.e., any stationary point of $\ln p(\mathbf{y}, \boldsymbol{\theta})$ is a fixed point of the algorithm. Comparing with (7), we therefore see that the MAP estimate $\hat{\boldsymbol{\theta}}$ is necessarily a fixed point of the above algorithm.

IV. Derivation of the Near-MAP Code Aided Algorithm

For this particular problem of estimating time-varying phase offsets, the model in (1)-(4) implies that

$$\ln p(\mathbf{y} \mid \boldsymbol{\theta}) = \ln\left(\sum_{v=1}^{2^K} p(\mathbf{y} \mid \mathbf{c} = \tilde{\mathbf{c}}_v, \boldsymbol{\theta}, \xi)\, p(\mathbf{c} = \tilde{\mathbf{c}}_v)\right) = \ln\left(\sum_{v=1}^{2^K}\left(\prod_{l=1}^{L} p\left(y_l \mid s_l = \tilde{s}_l(\tilde{\mathbf{c}}_v), \theta_l, \xi\right)\right) p(\mathbf{c} = \tilde{\mathbf{c}}_v)\right), \tag{11}$$

$$\ln p(\boldsymbol{\theta}) = \ln p(\theta_1) + \sum_{l=2}^{L} \ln p(\theta_l \mid \theta_{l-1}). \tag{12}$$

Based on this model, we now derive the explicit expression of the error-term used by the gradient ascent algorithm

$$\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y}, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y} \mid \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} + \nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \tag{13}$$

Taking (1)-(4) into account, we get for any index *m* such that $1 < m < L$, the error-term considered in the gradient ascent algorithm (see details of the calculus derived in the Appendix)

$$\left[\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y}, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m = \left[\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y} \mid \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m + \left[\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m, \tag{14}$$

where $$\left[\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y} \mid \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m = \frac{\partial \ln p(\mathbf{y} \mid \boldsymbol{\theta})}{\partial \theta_m} = \sum_{v=1}^{2^K} \Pr(\mathbf{c} = \tilde{\mathbf{c}}_v \mid \mathbf{y}, \boldsymbol{\theta}, \xi) \frac{\partial \ln\left(y_m \mid s_m = \tilde{s}_m(\tilde{\mathbf{c}}_v), \theta_m, \xi\right)}{\partial \theta_m} \tag{15}$$

and $$\left[\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m = \frac{\partial \ln p(\boldsymbol{\theta})}{\partial \theta_m} = \frac{\theta_{m-1} + \theta_{m+1} - 2\theta_m}{\sigma_\theta^2}. \tag{16}$$

By setting the error-term to zero, we further obtain the following expressions

$$\theta_m = \begin{cases} \theta_2 + \sigma_\theta^2 \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \dfrac{\partial \ln\left(y_1 \mid s_1 = \tilde{s}_1\left(\tilde{\mathbf{c}}_v\right), \theta_1, \xi\right)}{\partial \theta_1} - \xi, & m = 1 \\ \dfrac{1}{2}\left(\theta_{m-1} + \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \dfrac{\partial \ln\left(y_m \mid s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right), \theta_m, \xi\right)}{\partial \theta_m}\right) + \dfrac{1}{2}\left(\theta_{m+1} + \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \dfrac{\partial \ln\left(y_m \mid s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right), \theta_m, \xi\right)}{\partial \theta_m}\right), & 1 < m < L \\ \theta_{L-1} + \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \dfrac{\partial \ln\left(y_L \mid s_L = \tilde{s}_L\left(\tilde{\mathbf{c}}_v\right), \theta_L, \xi\right)}{\partial \theta_L} + \xi, & m = L. \end{cases} \tag{17}$$

All these derivatives are too hard to use in practice because they all involve an exponential number of sums. We thus propose the following method to find a good approximation of these derivatives. From (17), the terms containing an exponential number of sums can be further expressed as follows

$$\begin{aligned} &\sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \frac{\partial \ln\left(y_m \mid s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right), \theta_m, \xi\right)}{\partial \theta_m} \\ &\qquad = \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v, s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right) \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \frac{\partial \ln p\left(y_m \mid s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right), \theta_m, \xi\right)}{\partial \theta_m} \\ &\qquad = \sum_{v=1}^{2^K} \sum_{i=1}^{M} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v, s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right) \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \delta_{\mathbf{c} = \tilde{\mathbf{c}}_v}\left(s_m - \tilde{s}_i\right) \frac{\partial \ln p\left(y_m \mid s_m = \tilde{s}_i, \theta_m, \xi\right)}{\partial \theta_m}, \end{aligned} \tag{18}$$

where

$$\delta_{\mathbf{c} = \tilde{\mathbf{c}}_v}\left(s_m - \tilde{s}_i\right) = \begin{cases} 1, & \text{if } s_m = \tilde{s}_i \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Thus

$$\begin{aligned} &\sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \frac{\partial \ln\left(y_m \mid s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right), \theta_m, \xi\right)}{\partial \theta_m} \\ &\qquad = \sum_{i=1}^{M} \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v, s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right), s_m = \tilde{s}_i \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \frac{\partial \ln p\left(y_m \mid s_m = \tilde{s}_i, \theta_m, \xi\right)}{\partial \theta_m} \\ &\qquad = \sum_{i=1}^{M} \Pr\left(s_m = \tilde{s}_i \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \frac{2 \operatorname{Im}\left\{y_m \tilde{s}_i^* e^{-j\theta_m}\right\}}{\sigma_n^2}, \end{aligned} \tag{20}$$

where

$$\Pr\left(s_m = \tilde{s}_i \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) = \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v, s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right), s_m = \tilde{s}_i \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) = \sum_{v=1}^{2^K} \Pr\left(\mathbf{c} = \tilde{\mathbf{c}}_v, s_m = \tilde{s}_m\left(\tilde{\mathbf{c}}_v\right) \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right) \delta_{\mathbf{c} = \tilde{\mathbf{c}}_v}\left(s_m - \tilde{s}_i\right). \tag{21}$$

The term $\Pr\left(s_m = \tilde{s}_i \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right)$ in (21) is exactly the a posteriori probability (APP) of the code-aided scenario. Generally, the calculation of $\Pr\left(s_m = \tilde{s}_i \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right)$ is NP hard due to 2^K sums involved. However, if we consider the term $\Pr\left(s_m = \tilde{s}_i \mid \mathbf{y}, \boldsymbol{\theta}, \xi\right)$ as the decoded information, there do exist tractable decoding algorithms for approximate and even accurate computations. For example, the belief propagation (BP) algorithm [27], [28] allow us evaluate the approximate value of $\Pr\left(s_m=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)$ for turbo codes and low-density parity-check (LDPC) codes. Inserting (20) into (17), we can further modify our algorithm as below

$$\theta_m=\begin{cases}\theta_2+\dfrac{2\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_1=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_1\tilde{s}_i^*e^{-j\theta_1}\right\}-\xi, & m=1\\ \dfrac{1}{2}\left(\theta_m^{(F)}+\theta_m^{(B)}\right), & 1<m<L\\ \theta_{L-1}+\dfrac{2\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_L=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_L\tilde{s}_i^*e^{-j\theta_L}\right\}+\xi, & m=L,\end{cases}\tag{22}$$

where the terms defined as

$$\theta_m^{(F)}\triangleq\theta_{m-1}+\frac{\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_m=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_m\tilde{s}_i^*e^{-j\theta_m}\right\}\tag{23}$$

and $$\theta_m^{(B)}\triangleq\theta_{m+1}+\frac{\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_m=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_m\tilde{s}_i^*e^{-j\theta_m}\right\}\tag{24}$$

can be regarded as first order PLL outputs aided with the decoded information $\left\{\Pr\left(s_m=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\right\}$, $\theta_m^{(F)}$ and $\theta_m^{(B)}$ being respectively updated in the increasing (Forward) and decreasing (Backward) time directions. The physical meaning of (22) can thus be interpreted as follows; for the first position $m=1$, the MAP estimator is estimated from θ_2 with a backward PLL and for the last position $m=L$, the MAP estimator is estimated from θ_{L-1} with a forward PLL; for the middle range positions $1<m<L$, the MAP phase estimator averages the values of the forward and of the backward PLLs. Since in practice it is impossible to know the actual phase values θ_{m-1}, θ_m, and θ_{m+1} in the forward and backward loops defined in (25)-(26), we replace these true phase values by their estimates

$$\begin{cases}\hat{\theta}_m^{(F)}=\hat{\theta}_{m-1}^{(F)}+\dfrac{\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_m=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_m\tilde{s}_i^*e^{-j\hat{\theta}_{m-1}^{(F)}}\right\},\\ \hat{\theta}_m^{(B)}=\hat{\theta}_{m+1}^{(B)}+\dfrac{\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_m=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_m\tilde{s}_i^*e^{-j\hat{\theta}_{m+1}^{(B)}}\right\}.\end{cases}\tag{25}$$

Then the so-called smoothing PLL (or S-PLL) [17],[19] can be written as

$$\hat{\theta}_m^{(F/B)}\triangleq\begin{cases}\hat{\theta}_2^{(B)}+\dfrac{2\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_1=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_1\tilde{s}_i^*e^{-j\hat{\theta}_2^{(B)}}\right\}-\hat{\xi}, & m=1,\\ \dfrac{1}{2}\left(\hat{\theta}_m^{(F)}+\hat{\theta}_m^{(B)}\right), & 2\le m\le L-1,\\ \hat{\theta}_{L-1}^{(F)}+\dfrac{2\sigma_\theta^2}{\sigma_n^2}\sum_{i=1}^{M}\Pr\left(s_L=\tilde{s}_i \mid \mathbf{y},\boldsymbol{\theta},\xi\right)\operatorname{Im}\left\{y_L\tilde{s}_i^*e^{-j\hat{\theta}_{L-1}^{(F)}}\right\}+\hat{\xi}, & m=L.\end{cases}\tag{26}$$

Note that for a BPSK non coded system, (26) just reduces to equation (18) of [17]. The expression “Smoothing loops” stems from the estimation procedure; the S-PLL encompasses one classical forward PLL and one backward

PLL working in the reverse time direction towards the beginning of the block. One can note that this algorithm reminds Kalman smoothing valid for linear Gaussian problems. After that both loops respectively finish their estimations on the whole block, the final operation just globally averages the estimated phases of the forward and the backward loops as displayed by Fig. 1 and commented in the next section.

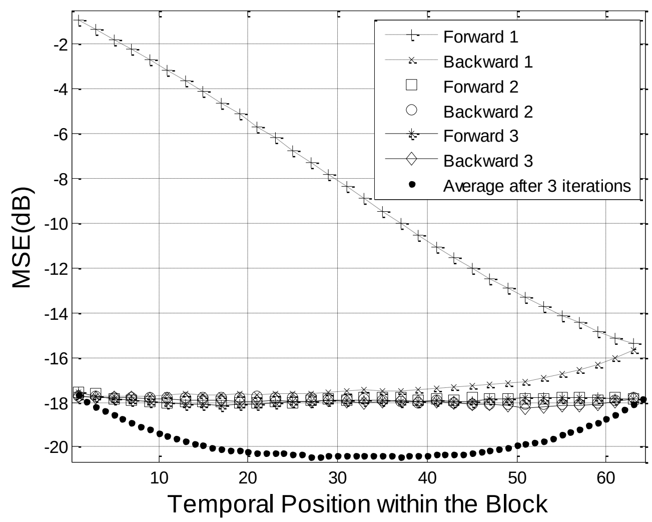

Fig. 1 MSEs of the various estimation schemes along time

V. Performance Study

In this section, we relate the performance of the proposed algorithm to the Bayesian Cramer-Rao bound (BCRB) associated to the estimation of θ_m. We first briefly recall the main expressions of the Cramer-Rao bounds. We then derive a linear model for the error term, which is valid for moderate-to-high SNR and for small phase errors. Finally, we show that the proposed algorithm asymptotically converges towards the BCRB, as long as our linear model is valid.

Cramer-Rao bound: One traditionally distinguishes between two kinds of Cramer-Rao bounds (CRBs), *i.e.* the standard and the Bayesian CRB respectively dedicated to deterministic and Bayesian parameters. The standard CRB lower-bounds the conditional mean square error (MSE) of any unbiased estimator, *i.e.*,

$$E_{\mathbf{y}|\boldsymbol{\theta}}\left\{\left(\hat{\theta}_m(\mathbf{y})-\theta_m\right)^2\right\}=SCRB(\theta_m). \tag{27}$$

It is defined as the inverse of the Fisher information matrix and can be expressed as

$$\mathbf{F}(\theta_m) \triangleq E_{\mathbf{y}|\boldsymbol{\theta}}\left\{-\frac{\partial^2 \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m^2}\right\} = E_{\mathbf{y}|\boldsymbol{\theta}}\left\{\left(\frac{\partial \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m}\right)^2\right\}. \tag{28}$$

Note that in the above definition of the SCRB, we have assumed that all $\theta_{l\neq m}$ are perfectly known. Note also that the SCRB is in general a function of the particular realization of θ_m. In the particular model we consider, due to the rotational invariance, it is however easy to show that the SCRB is independent on θ_m, i.e., $SCRB(\theta_m) = \text{const}$.

The BCRB is a lower bound on the mean squared error which can be achieved by any estimator, *i.e.*,

$$E_{\mathbf{y},\boldsymbol{\theta}}\left\{\left(\hat{\theta}_m(\mathbf{y}) - \theta_m\right)^2\right\} \geq BCRB(\theta_m). \tag{29}$$

The BCRB is equal to the inverse of the Bayesian Fisher information matrix, which is defined as

$$\mathbf{B}(\theta_m) \triangleq -E_{\mathbf{y},\boldsymbol{\theta}}\left\{\frac{\partial^2 \ln p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_m^2}\right\} = -E_{\mathbf{y},\boldsymbol{\theta}}\left\{\frac{\partial^2 \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m^2}\right\} - E_{\mathbf{y},\boldsymbol{\theta}}\left\{\frac{\partial^2 \ln p(\boldsymbol{\theta})}{\partial\theta_m^2}\right\} = E_{\mathbf{y},\boldsymbol{\theta}}\left\{\left(\frac{\partial \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m}\right)^2\right\} + E_{\mathbf{y},\boldsymbol{\theta}}\left\{\left(\frac{\partial \ln p(\boldsymbol{\theta})}{\partial\theta_m}\right)^2\right\}. \tag{30}$$

Linear approximation of the error term: We derive here a "small-phase error/high SNR" model and successively consider approximations of the terms in (14), *i.e.*, (15) and (16).

1) Approximation of (15): if $\hat{\boldsymbol{\theta}} \simeq \boldsymbol{\theta}$ is valid, the first term in (14) can be approximated by its first-order Taylor expansion

$$\frac{\partial \ln p(\mathbf{y}\,|\,\hat{\boldsymbol{\theta}})}{\partial\theta_m} \simeq \frac{\partial \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m} + \frac{\partial^2 \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m^2}\left(\hat{\theta}_m - \theta_m\right). \tag{31}$$

Moreover, if we assume a sufficiently high SNR, the last equation becomes

$$\begin{aligned}\frac{\partial \ln p(\mathbf{y}\,|\,\hat{\boldsymbol{\theta}})}{\partial\theta_m} &\simeq E_{\mathbf{y}|\boldsymbol{\theta}}\left\{\frac{\partial^2 \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m^2}\right\}\left(\hat{\theta}_m - \theta_m\right) + \frac{\partial \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial\theta_m}\\ &\triangleq g_m(\boldsymbol{\theta})\left(\hat{\theta}_m - \theta_m\right) + h_m(\boldsymbol{\theta}),\end{aligned} \tag{32}$$

where in standard PLL schemes, the parameters $g_m(\boldsymbol{\theta})$ and $h_m(\boldsymbol{\theta})$ are usually referred as the gain and the noise of the loop, respectively [14]. It turns out that these parameters can be related to the SCRB of the problem. In particular, we have (by definition)

$$g_m(\boldsymbol{\theta}) = -SCRB^{-1}(\theta_m) \tag{33}$$

and with a direct calculus

$$\begin{cases} E_{\mathbf{y}}\left\{h_m(\boldsymbol{\theta})\right\} = \int p(\mathbf{y}\,|\,\boldsymbol{\theta}) \dfrac{\partial \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial \theta_m} d\mathbf{y} = \int p(\mathbf{y}\,|\,\boldsymbol{\theta}) \dfrac{1}{p(\mathbf{y}\,|\,\boldsymbol{\theta})} \dfrac{\partial p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial \theta_m} d\mathbf{y} = \dfrac{\partial}{\partial \theta_m} \int p(\mathbf{y}\,|\,\boldsymbol{\theta}) d\mathbf{y} = 0 \quad (\text{as} \int p(\mathbf{y}\,|\,\boldsymbol{\theta}) d\mathbf{y} = 1), \\ E_{\mathbf{y}}\left\{\left(h_m(\boldsymbol{\theta})\right)^2\right\} = \int p(\mathbf{y}\,|\,\boldsymbol{\theta}) \left(\dfrac{\partial \ln p(\mathbf{y}\,|\,\boldsymbol{\theta})}{\partial \theta_m}\right)^2 d\mathbf{y} = SCRB^{-1}(\theta_m). \end{cases} \tag{34}$$

In other words, the smaller the SCRB, the higher (the amplitude of) the loop gain and the smaller the loop noise. These connections between the loop gain and noise and the SCRB will be useful in the characterization of the performance of the proposed algorithm.

2) Approximation of (16): similarly assuming $\hat{\boldsymbol{\theta}} \simeq \boldsymbol{\theta}$, the second term in (14) may be approximated as

$$\left[\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m = \frac{-2\hat{\theta}_m + \theta_{m-1} + \theta_{m+1}}{\sigma_\theta^2}. \tag{35}$$

Using the model described by equation (3), we obtain

$$\left[\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m = \frac{-2\hat{\theta}_m + \theta_m - w_l + \theta_m + w_{l+1}}{\sigma_\theta^2} = \frac{-2\hat{\theta}_m + 2\theta_m - w_l + w_{l+1}}{\sigma_\theta^2}. \tag{36}$$

Putting for any index *m*, (32) and (36) into (13), we obtain the following linear model

$$\begin{aligned} \left[\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y},\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m &\simeq g_m(\boldsymbol{\theta})\left(\hat{\theta}_m - \theta_m\right) + h_m(\boldsymbol{\theta}) + \frac{-2\hat{\theta}_m + 2\theta_m - w_l + w_{l+1}}{\sigma_\theta^2} \\ &= \left(g_m(\boldsymbol{\theta}) - \frac{2}{\sigma_\theta^2}\right)\left(\hat{\theta}_m - \theta_m\right) + h_m(\boldsymbol{\theta}) + \frac{1}{\sigma_\theta^2}\left(w_{l+1} - w_l\right) \\ &= -BCRB^{-1}(\theta_m)\left(\hat{\theta}_m - \theta_m\right) + h_m(\boldsymbol{\theta}) + \frac{1}{\sigma_\theta^2}\left(w_{l+1} - w_l\right), \end{aligned} \tag{37}$$

where the last equality comes from (30) and the fact that

$$-E_{\boldsymbol{\theta}}\left\{\frac{\partial^2 \ln p(\boldsymbol{\theta})}{\partial \theta_m^2}\right\} = \frac{2}{\sigma_\theta^2}. \tag{38}$$

Performance study: Here, we show that, as long as the linear model (37) is valid, the proposed algorithm converges towards the BCRB. By definition, the estimate after convergence must be such that $\left[\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{y},\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\right]_m = 0$ for any *m*. Using (37), this is equivalent to the following condition

$$BCRB^{-1}(\theta_m)\left(\hat{\theta}_m - \theta_m\right) = h_m(\boldsymbol{\theta}) + \frac{1}{\sigma_\theta^2}\left(w_{l+1} - w_l\right). \tag{39}$$

Squaring both sides and taking the expectation with respect to $\mathbf{y}$ and $\boldsymbol{\theta}$, we obtain

$$\left(BCRB^{-1}(\theta_m)\right)^2 E_{\mathbf{y},\boldsymbol{\theta}}\left\{\left(\hat{\theta}_m - \theta_m\right)^2\right\} = E_{\mathbf{y},\boldsymbol{\theta}}\left(\left(h_m(\boldsymbol{\theta})\right)^2\right) + \frac{1}{\sigma_\theta^4} E_{\boldsymbol{\theta}}\left(\left(w_{l+1}\right)^2\right) + \frac{1}{\sigma_\theta^4} E_{\boldsymbol{\theta}}\left(\left(w_l\right)^2\right) + \frac{2}{\sigma_\theta^2} E_{\mathbf{y},\boldsymbol{\theta}}\left(w_{l+1} h_m(\boldsymbol{\theta})\right) - \frac{2}{\sigma_\theta^2} E_{\mathbf{y},\boldsymbol{\theta}}\left(w_l h_m(\boldsymbol{\theta})\right) . \tag{40}$$

Now, the five terms in the right-hand side of (40) are respectively equal to

$$E_{\mathbf{y},\boldsymbol{\theta}}\left(\left(h_m(\boldsymbol{\theta})\right)^2\right) = SCRB^{-1}(\theta_m) \quad \text{(see equation (34))} \tag{41}$$

$$E_{\boldsymbol{\theta}}\left(\left(w_{l+1}\right)^2\right) = E_{\boldsymbol{\theta}}\left(\left(w_l\right)^2\right) = \sigma_\theta^2\,, \tag{42}$$

$$E_{\mathbf{y},\boldsymbol{\theta}}\left(w_{l+1}h_m(\boldsymbol{\theta})\right) = E_{\mathbf{y},\boldsymbol{\theta}}\left(w_l h_m(\boldsymbol{\theta})\right) = 0\,, \tag{43}$$

Substituting (41)-(43) into (40), we finally obtain

$$E_{\mathbf{y},\boldsymbol{\theta}}\left\{\left(\hat{\theta}_m - \theta_m\right)^2\right\} = BCRB(\theta_m). \qquad \text{Q.E.D.} \tag{44}$$

This asymptotic result will be illustrated in section VII.

VI. Comments on the S-PLL Algorithm

Before illustrating the algorithm performance, we address several practical considerations. In order to overcome the initial transient problem, the S-PLL can be initialized as heuristically proposed by Cochran [29] and depicted on Fig. 1. Without any a priori knowledge on the initial phase, a forward loop $\hat{\theta}_m^{(F)}$ runs, as traditionally, from the beginning of the block with an arbitrary initial value, towards the end of the considered block (see on Fig. 1 the curve labeled as “Forward 1”). Then a backward PLL $\hat{\theta}_m^{(B)}$ is initialized with the final value estimated by the forward PLL and is recursively updated from the end to the beginning of the considered block, running in the reverse time direction (see on Fig. 1 the curve labeled as “Backward 1”). This process can then be iterated several times, *i.e.* the estimation error at the end of the previous backward loop can be used as the estimation error at the beginning of the next recursion (“Forward 2”), and globally $N_{\text{S-PLL}}$ forward and backward recursions can sequentially be proceeded till the convergence of the loops. Note that though this procedure (see (25)) is efficient to get rid of the initial transient, it however only leads to on-line performance. The on-line algorithm can be further improved at low SNR within the turbo-receiver framework [12],[13] but it cannot exceed the on-line Cramer-Rao bounds. To take advantage of the clear superiority of the off-line bounds compared to the on-line bounds [14],[15],[30],[31], it is absolutely necessary to average the forward and the backward estimates as described by equation (26).

In the previous initial forward-backward recursions, the S-PLL algorithm is initialized with symbols having an equal a posteriori probability because the soft decoding process has not yet been performed at this stage. The phases of the received symbols are then updated with the initial phase outputs of the S-PLL algorithm (after $N_{\text{S-PLL}}$ forward and backward recursions) and a soft decoding can then be operated. As depicted on the right hand sight of Fig. 2, some symbol a posteriori probabilities can then be obtained from the soft decoder's output and fed back to the S-PLL algorithm which then proceeds a more accurate phase estimation. However, one might rather choose to only proceed non-code-aided off-line synchronization. In this case it is sufficient to process the $N_{\text{S-PLL}}$ initial forward and backward recursions without the need for feeding back the soft decoder's output. Results comparing the coded and non-coded off-line scheme will be displayed in the following paragraph. Also, similarly to turbo decoding, one might divide each received symbol blocks into sub-blocks; a S-PLL algorithm can then be deployed in parallel on each of the smaller blocks and this can shorten the processing time.

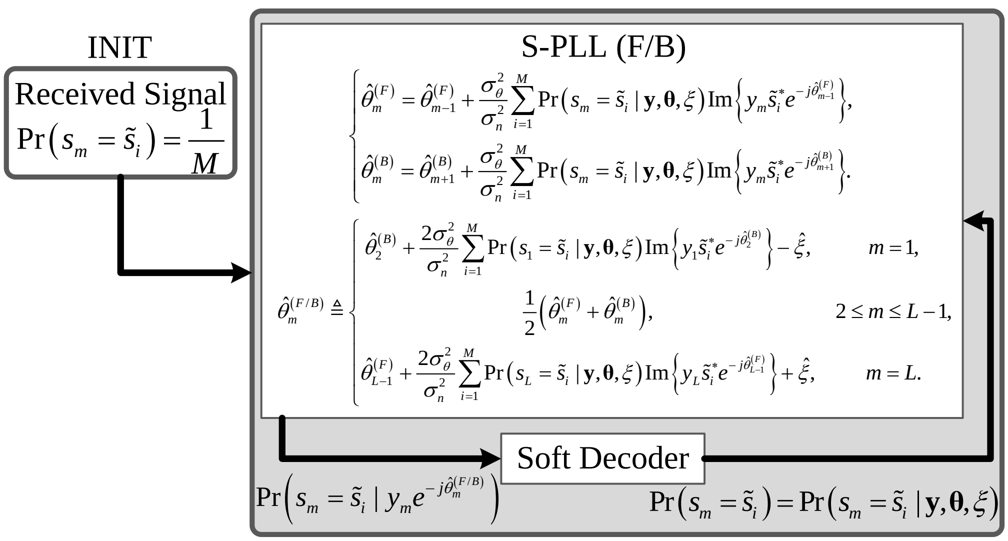

Fig. 2 The simulation platform

Finally we give a brief analysis about the algorithm implementation complexity. The classical on-line PLL has a very low gradient-like complexity and has been employed in real systems for several decades. The complexity price for the off-line improvement is only twice that of the on-line algorithm due to combining two traditional PLLs. In addition to the traditional memorization of L symbols, we need to store $2L$ phase values. For the non-coded off-line scenario, we need to proceed $N_{\text{S-PLL}}$ forward-backward recursions ($N_{\text{S-PLL}} = 3$ in practice) which involves a very reasonable delay.

VII. Simulation Results

We assume that messages are encoded by the recursive systematic rate-$1/2$ turbo code with generator polynomials $G_1 = 37_{OCT}$ and $G_2 = 21_{OCT}$; the N bit codewords are then interleaved by a S-random interleaver [32] and mapped by a conventional Gray mapper. After passing through the phase uncertain channel, blocks of L complex-valued symbols (BPSK, QAM) are finally received by the carrier recovery block such as depicted by Fig. 2. MSEs are evaluated over 10^5 Monte Carlo trials and the corresponding curves are noted "Sim" in the following figures. Moreover, we also use the following notations: "On-Line" means that the MSE is the conventional forward estimation without any backward estimation, and "Off-Line" means that the MSE is measured after $N_{\text{S-PLL}} = 3$ forward and backward iterations.

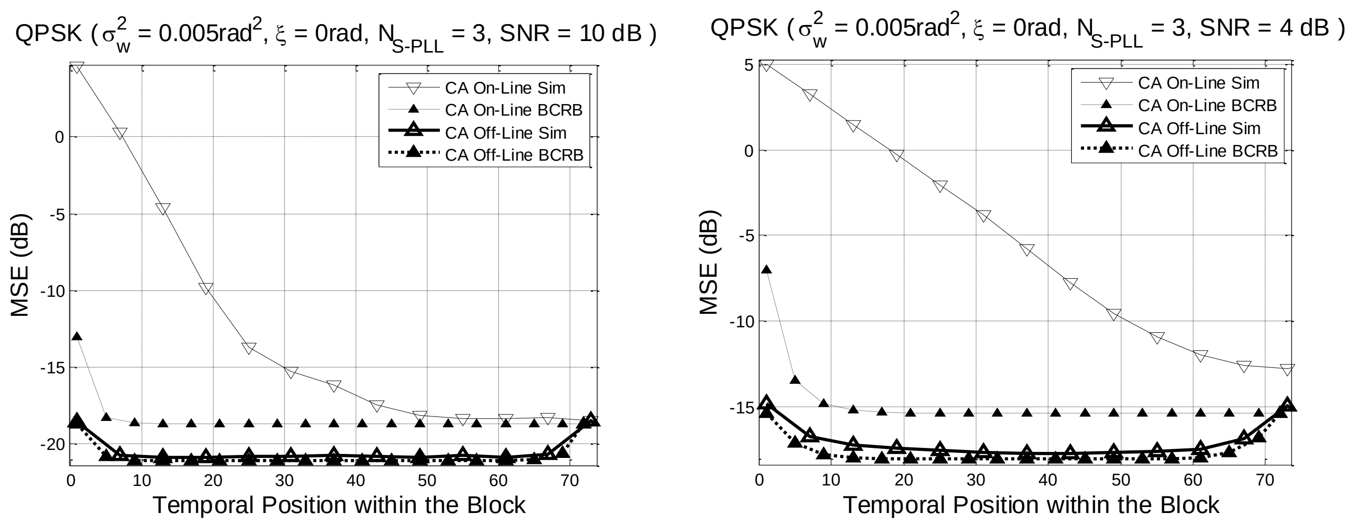

Fig. 3 MSE curves for the various positions in the block at two different SNRs.

Figure 3 compares different on-line and off-line MSE performance to Bayesian CRBs [30],[31] for a block of L=72 coded QPSK modulated signals. For the on-line scenario, the MSE lowers down towards the on-line BCRB as more and more samples are processed. However, the off-line estimator performance is always better than the on-line estimator performance. This is because the off-line estimator is able to reach the off-line BCRB which is a few dBs under the on-line BCRB. As also illustrated by figure 4 and figure 5, DA and NDA curves would lead to similar conclusions, the non-coded performance being itself inferior to the coded-scheme, especially at low SNR.

We now display various results where $L = 864$ for QPSK and $L = 1152$ for 16QAM. Instead of looking at the different positions for some given SNRs, we now display the phase estimation error curves as function of the SNR in the central position of the block.

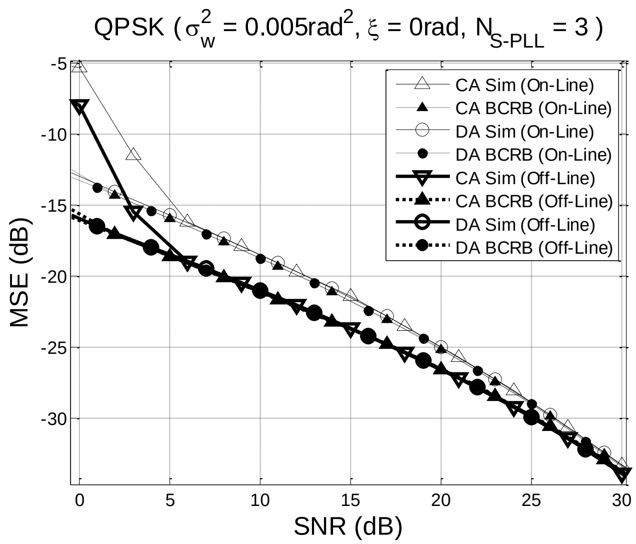

Fig. 4 QPSK performance in the center position versus SNR with no linear drift

When the linear drift $\xi = 0$, both the on-line and off-line algorithms can reach the corresponding BCRBs and work very well in a large SNR range (see figure 4). At high SNR, each current observation is reliable enough to estimate the current phase shift and there is neither a gain with a block proceeding nor any phase synchronization problem. At more realistic average to low SNRs, as it was already depicted on figure 3 for any position, the off–line approach allows to benefit from a gain of several dBs. The reason for this result is that one observation is not sufficient to correctly estimate the symbols while a block of more observations should be used to improve the estimation performance. At low SNR, when the code is not efficient anymore, the CA curves leave the DA curves. The superiority of the off-line bound is even enhanced when there is a linear drift as illustrated on the following figure 5.

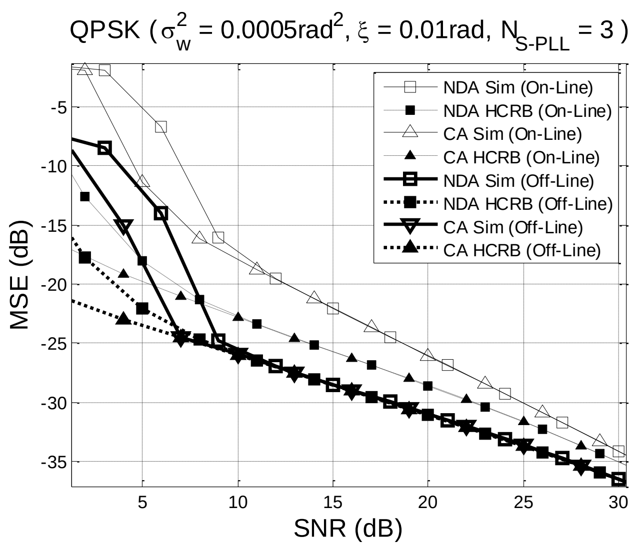

Fig. 5 QPSK performance in the center position versus SNR, with a linear drift $\xi \neq 0$

The conclusions for figure 5 with a linear drift are similar to figure 4 but we can specify three more facts. First, the MSEs must now be compared to Hybrid CRBs because there is a (large) deterministic linear drift ξ and all the curves are poorer than the curves obtained with no linear drift. Second, for SNRs spanning from mid-range to low range, the on-line scheme is not able to reach anymore its HCRB; this comes from the classical fact that a traditional PLL becomes biased for larger and larger linear drift (see for instance [17]). Third, the CA MSEs leave their asymptote regime at lower SNRs than the NDA curves and this might be useful for terminals forced to work at low SNRs.

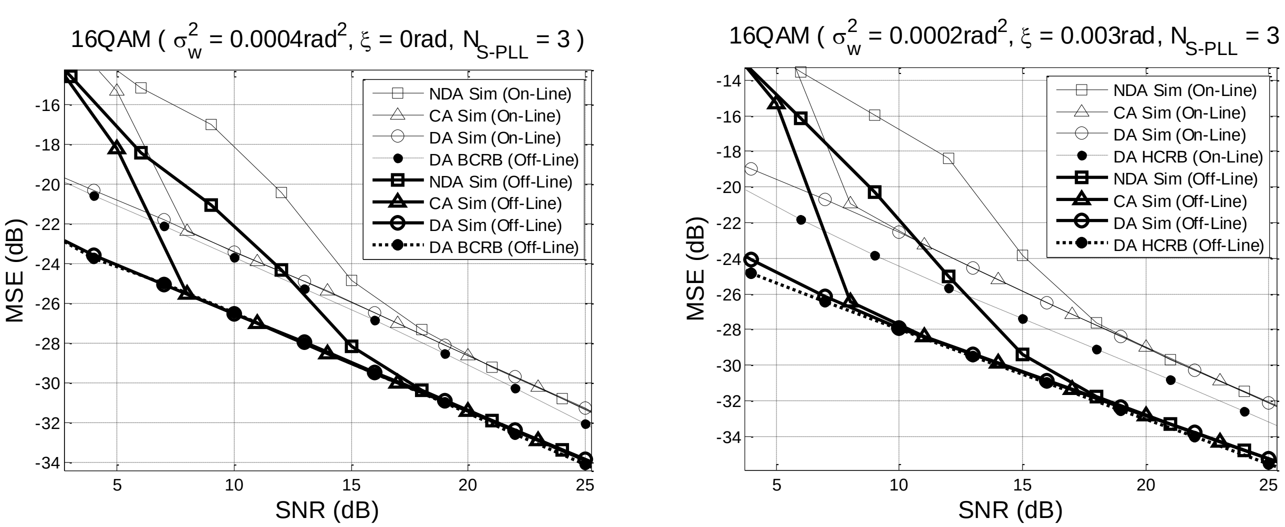

Fig. 6 16QAM performance in the center position versus SNR for two different linear drifts

Similar results, are obtained for a 16QAM modulated signal at larger SNRs with (right part of Figure. 6) or without (left part of Figure. 6) a linear drift. The advantage of using an off-line scenario is predominant for all the synchronizing schemes. As the SNR becomes lower and lower, the NDA curves leave the CRBs first, before the CA curves and finally the DA curves. In this SNR range, the error control coding becomes critical to improve the estimation performance; however, once again, the main result from this figure is that for most of the SNRs, though it has a lower complexity, the off-line NDA SPLL is generally able to give better (or similar) results than the on-line CA PLL.

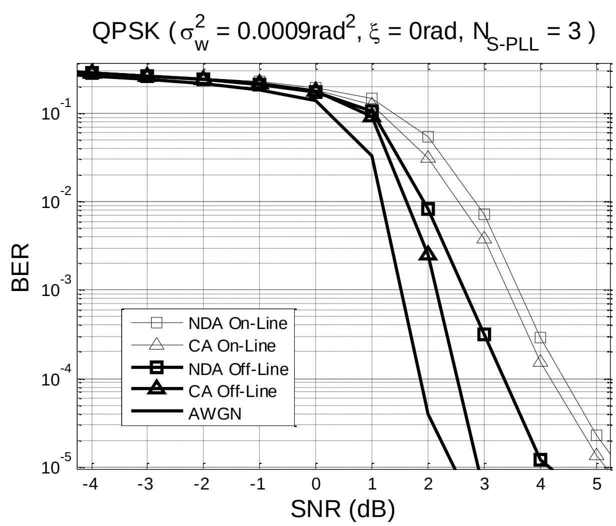

Fig. 7 Influence of the phase recovery on BER curves

Finally, figure 7 represents the BER curves for various scenarios (off-line/on-line, NDA/CA) for a QPSK modulated signal. Clearly, the off-line scenario is superior to its on-line counterpart; also, the CA scheme allows to gain up to 1 dB compared to the NDA scheme for the off-line scenario. We note with interest that a CA off-line scheme is able to work at about 0.5 dB from the perfect phase recovery scheme (AWGN channel).

VIII. CONCLUSION

In this paper, we derived a near-MAP phase estimator for generally coded and modulated symbols which is made out of two first order PLLs possibly aided by some decoded *a posteriori* information. The MSE performance of the S-PLL provides a gain of several decibels when compared to a classical forward on-line algorithm; it can reach the Cramer-Rao bounds of interest over a wide range of SNRs and we proved the asymptotic convergence. Moreover, it is to be high-lighted that low-complexity NCA off-line schemes often give better results than the classical turbo recovery (CA on-line PLLs). Smoothing loops do not suffer from a poor transient behavior and are robust against frequency offsets. They are easy to implement, especially compared to CA (such as turbo) phase recovery schemes and thus, they could be very useful in practice.

REFERENCES

[1] G. Colavolpe, G. Ferrari, and R. Raheli, "Noncoherent iterative (Turbo) decoding," *IEEE Trans. Commun.*, vol. 48, pp. 1488–1498, Sept. 2000.

[2] G. Colavolpe, A. Barbieri, and G. Caire, "Algorithms for iterative decoding in the presence of strong phase noise," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 1748–1757, Sept. 2005.

[3] J. Dauwels and H.-A. Loeliger, "Joint decoding and phase estimation: An exercise in factor graphs," *in Proc. IEEE ISIT'03*, p. 231, Yokohama, Japan, June 29-July 4, 2003.

[4] J. Dauwels and H.-A. Loeliger, "Phase estimation by message passing," *in Proc. IEEE ICC'04,* pp. 523–527, New York, USA, June 20-24, 2004.

[5] M. Nissila and S. Pasupathy, "Adaptive Iterative Detectors for Phase-Uncertain Channels via Variational Bounding," *IEEE Trans. on Commun.*, vol. 57, pp.716-725, Mar. 2009.

[6] D. D. Lin and T. J. Lim, "The variational inference approach to joint data detection and phase noise estimation in OFDM," *IEEE Trans. on Signal Processing,* vol. 55, pp. 1862-1874, May 2007.

[7] P.O. Amblard, J.M. Brossier, and E. Moisan, "Phase tracking: what do we gain from optimality? Particle filtering versus phase-locked loops," *Signal Processing,* vol. 83, pp. 151-167, Oct. 2003.

[8] M. J. Nissila, S. Pasupathy, and A. Mammela, "An EM approach to carrier phase recovery in AWGN channel," *in Proc. IEEE ICC'01*, pp. 2199–2203, Helsinki, Finland, June 11-15, 2001.

[9] 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 ICC’03,* pp. 2933–2937, Anchorage, Alaska, May 11-15, 2003.

[10] F. M. Gardner, *Phaselock Techniques,* 3rd ed., Wiley-Interscience, 2005.

[11] L. Zhang and A. Burr, “Iterative Carrier Phase Recovery suited for Turbo-Coded systems,” *IEEE Trans. on Wireless Commun.,* vol. 3, pp. 2267-2276, Nov. 2004.

[12] N. Noels, H. Steendam and M. Moeneclaey, "Performance Analysis of ML-Based Feedback Carrier Phase Synchronizers for Coded Signals," *IEEE Transactions on Signal Processing,* vol. 55, pp. 1129-1136, March 2007.

[13] N. Noels, H. Steendam, M. Moeneclaey, "Effectiveness Study of Code-Aided and Non-Code-Aided ML-Based Feedback Phase Synchronizers," *in Proc. IEEE ICC'06,* Istanbul, Turkey, June 11-15, 2006.

[14] S. Bay, C. Herzet, J.P. Barbot, J. M. Brossier, and B. Geller, "Analytic and Asymptotic Analysis of Bayesian Cramer–Rao Bound for Dynamical Phase Offset Estimation," *IEEE Transactions on Signal Processing,* vol. 56, pp. 61-70, Jan. 2008.

[15] S. Bay, B. Geller, A. Renaux, J.P. Barbot, and J.M. Brossier, “On the Hybrid Cramer-Rao bound and its application to dynamical phase estimation,” *IEEE Signal Processing Letters,* vol. 15, pp. 453-456, 2008.

[16] J. Yang and B. Geller, “Bayesian and Hybrid Cramer-Rao Bounds for the Carrier Recovery under Dynamic Phase Uncertain Channels,” *IEEE Trans. on Signal Processing,*pp. 667-680, Feb. 2011.

[17] J. Yang and B. Geller, “Near Optimum Low Complexity Smoothing Loops for Dynamical Phase Estimation - Application to BPSK Modulated Signals,” *IEEE Trans. on Signal Processing,* pp. 3704-3711, Sept. 2009.

[18] B. Geller, J. P. Barbot, J.M. Brossier, C. Vanstraceele, Procédé d’estimation de la phase et du gain de données d’observation transmises sur un canal de transmission. French patent Fr2005002301. International extension WO2006032768 on the 30th March 2006.

[19] J. Yang, B. Geller, C. Herzet, and J.M. Brossier, “Smoothing PLLs for QAM Dynamical Phase Estimation,” *in Proc. IEEE Inter. Conf. on Commun. 2009, ICC'09,* Dresden, 14-18 June 2009.

[20] N. Noels, M. Moeneclaey, F. Simoens, and D. Delaruelle, “A Low-Complexity Iterative Phase Noise Tracker for coded Bit-Interleaved CPM Signals in AWGN,” IEEE Trans. on Signal Processing, vol.59, pp. 4271-4284, Sept. 2011.

[21] F. Lehmann, “A Gaussian Sum Approach to Blind Carrier Phase Estimation and Data Detection in Turbo Coded Transmissions,” IEEE Trans. on Communication, vol. 57, pp. 2619-2632, Sept. 2009.

[22] J. A. McNeill, *“Jitter in ring oscillators,”* Ph.D. dissertation, Boston University, 1994.

[23] A. Demir, A. Mehrotra, and J. Roychowdhury, “Phase noise in oscillators: a unifying theory and numerical methods for characterization,” *IEEE Trans. Circuits Syst. I,* vol. 47, pp. 655–674, May 2000.

[24] ETSI EN 302 307 “Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications,” June 2004.

[25] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[26] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Trans. on Info. Theory,* vol. IT-20(2), pp. 284-287, Mar, 1974.

[27] J. Pearl, “Reverend Bayes on inference engines: A distributed hierarchical approach,” *in Proc. American Association of Artificial Intelligence National Conference on AI,* pp.133-136, Pittsburgh, PA. 1982.

[28] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (Revised Second Printing),* San Francisco, CA: Morgan Kaufmann, 1988.

[29] A. Cochran, Carrier Phase Synchronization by Reverse Playback. International patent PCT US 1997/022067, WO1998024210 on the 4th Jun. 1998.

[30] J. Yang, B. Geller, and A. Wei, “Bayesian and Hybrid Cramer-Rao Bounds for QAM Dynamical Phase Estimation,” *in Proc. IEEE Signal Processing, ICASSP'09,* Taipei, 19-24 April 2009.

[31] J. Yang, B. Geller, and A. Wei, “Approximate Expressions for Cramer-Rao Bounds of Code Aided QAM Dynamical Phase Estimation,” *in Proc. IEEE Inter. Conf. on Commun. 2009, ICC'09,* Dresden, 14-18 June 2009.

[32] D. Divsalar and F. Pollara, “Turbo codes for PCS applications,” *in Proc. ICC’95,* pp. 54-59, Seattle, WA, June 18-22, 1995.

APPENDIX

DERIVATIONS OF ERROR TERMS

The first derivative of $\ln p(\mathbf{y},\boldsymbol{\theta})$ can be calculated from (11)-(12) ; for $1<m<L$, we thus get $\frac{\partial \ln p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_m}$

$$\begin{aligned}
\frac{\partial \ln p(\mathbf{y},\boldsymbol{\theta})}{\partial\theta_m} &= \frac{\partial \ln\left(\sum_{v=1}^{2^K}\left(\prod_{l=1}^{L} p\left(y_l \mid s_l=\tilde{s}_l(\tilde{\mathbf{c}}_v),\theta_l,\xi\right)\right)p(\mathbf{c}=\tilde{\mathbf{c}}_v)\right)}{\partial\theta_m} - \frac{(\theta_m-\theta_{m-1}-\xi)}{\sigma_\theta^2} + \frac{(\theta_{m+1}-\theta_m-\xi)}{\sigma_\theta^2} \\
&= \frac{\sum_{v=1}^{2^K}\left(\prod_{l=1,l\neq m}^{L} p\left(y_l \mid s_l=\tilde{s}_l(\tilde{\mathbf{c}}_v),\theta_l,\xi\right)\frac{\partial p\left(y_m \mid s_m=\tilde{s}_m(\tilde{\mathbf{c}}_v),\theta_m,\xi\right)}{\partial\theta_m}\right)p(\mathbf{c}=\tilde{\mathbf{c}}_v)}{\sum_{v=1}^{2^K}\left(\prod_{l=1}^{L} p\left(y_l \mid s_l=\tilde{s}_l(\tilde{\mathbf{c}}_v),\theta_l,\xi\right)\right)p(\mathbf{c}=\tilde{\mathbf{c}}_v)} - \frac{2\theta_m}{\sigma_\theta^2} + \frac{\theta_{m-1}}{\sigma_\theta^2} + \frac{\theta_{m+1}}{\sigma_\theta^2} \\
&= \frac{\sum_{v=1}^{2^K}\left(\prod_{l=1}^{L} p\left(y_l \mid s_l=\tilde{s}_l(\tilde{\mathbf{c}}_v),\theta_l,\xi\right)\right)p(\mathbf{c}=\tilde{\mathbf{c}}_v)}{p(\mathbf{y}\mid\boldsymbol{\theta},\xi)}\frac{\frac{\partial p\left(y_m \mid s_m=\tilde{s}_m(\tilde{\mathbf{c}}_v),\theta_m,\xi\right)}{\partial\theta_m}}{p\left(y_m \mid s_m=\tilde{s}_m(\tilde{\mathbf{c}}_v),\theta_m,\xi\right)} - \frac{2\theta_m}{\sigma_\theta^2} + \frac{\theta_{m-1}}{\sigma_\theta^2} + \frac{\theta_{m+1}}{\sigma_\theta^2} \\
&= \frac{\sum_{v=1}^{2^K} p(\mathbf{y}\mid\mathbf{c}=\tilde{\mathbf{c}}_v,\boldsymbol{\theta},\xi)\,p(\mathbf{c}=\tilde{\mathbf{c}}_v)}{p(\mathbf{y}\mid\boldsymbol{\theta},\xi)}\frac{\partial \ln\left(y_m \mid s_m=\tilde{s}_m(\tilde{\mathbf{c}}_v),\theta_m,\xi\right)}{\partial\theta_m} - \frac{2\theta_m}{\sigma_\theta^2} + \frac{\theta_{m-1}}{\sigma_\theta^2} + \frac{\theta_{m+1}}{\sigma_\theta^2} \\
&= \frac{\sum_{v=1}^{2^K} p(\mathbf{c}=\tilde{\mathbf{c}}_v,\mathbf{y}\mid\boldsymbol{\theta},\xi)}{p(\mathbf{y}\mid\boldsymbol{\theta},\xi)}\frac{\partial \ln\left(y_m \mid s_m=\tilde{s}_m(\tilde{\mathbf{c}}_v),\theta_m,\xi\right)}{\partial\theta_m} - \frac{2\theta_m}{\sigma_\theta^2} + \frac{\theta_{m-1}}{\sigma_\theta^2} + \frac{\theta_{m+1}}{\sigma_\theta^2} \\
&= \sum_{v=1}^{2^K} \Pr(\mathbf{c}=\tilde{\mathbf{c}}_v\mid\mathbf{y},\boldsymbol{\theta},\xi)\frac{\partial \ln\left(y_m \mid s_m=\tilde{s}_m(\tilde{\mathbf{c}}_v),\theta_m,\xi\right)}{\partial\theta_m} + \frac{\theta_{m-1}+\theta_{m+1}-2\theta_m}{\sigma_\theta^2},
\end{aligned} \tag{45}$$

where $\Pr(\mathbf{c}=\tilde{\mathbf{c}}_v\mid\mathbf{y},\boldsymbol{\theta},\xi)=\frac{p(\mathbf{c}=\tilde{\mathbf{c}}_v,\mathbf{y}\mid\boldsymbol{\theta},\xi)}{p(\mathbf{y}\mid\boldsymbol{\theta},\xi)}$ is the normalized a posteriori probability (APP) of the codeword $\tilde{\mathbf{c}}_v$.

Assuming that we do not have any priori information about θ_1 (*i.e.* $\frac{\partial \ln p(\theta_1)}{\partial\theta_1}=0$), similarly to obtaining (46), we can easily obtain the first derivatives for $m=1$ and $m=L$

$$\frac{\partial \ln p(\mathbf{r},\boldsymbol{\theta})}{\partial\theta_1} = \sum_{v=1}^{2^K}\Pr(\mathbf{c}=\tilde{\mathbf{c}}_v\mid\mathbf{y},\boldsymbol{\theta},\xi)\frac{\partial \ln\left(y_1 \mid s_1=\tilde{s}_1(\tilde{\mathbf{c}}_v),\theta_1,\xi\right)}{\partial\theta_1} + \frac{(\theta_2-\theta_1-\xi)}{\sigma_\theta^2} \tag{47}$$

$$\frac{\partial \ln p(\mathbf{r},\boldsymbol{\theta})}{\partial\theta_L} = \sum_{v=1}^{2^K}\Pr(\mathbf{c}=\tilde{\mathbf{c}}_v\mid\mathbf{y},\boldsymbol{\theta},\xi)\frac{\partial \ln\left(y_L \mid s_L=\tilde{s}_L(\tilde{\mathbf{c}}_v),\theta_L,\xi\right)}{\partial\theta_L} - \frac{(\theta_L-\theta_{L-1}-\xi)}{\sigma_\theta^2}, \tag{48}$$

where the exact expression $\frac{\partial \ln p(y_l\mid\theta_l,\xi,s_l)}{\partial\theta_l}$ can be obtained from (49)

$$\frac{\partial \ln p(y_l\mid\theta_l,\xi,s_l)}{\partial\theta_l} = \frac{2\,\mathrm{Im}\left\{y_l s_l^* e^{-j\theta_l}\right\}}{\sigma_n^2}. \tag{50}$$