CARRIER PHASE TRACKING AT LOW SIGNAL-TO-NOISE RATIO: A PERFORMANCE COMPARISON OF A PARITY-CODE-AIDED AND A PILOT-SYMBOL-ASSISTED APPROACH

Nele Noels(1), Mathieu Dervin(2), Marc Moeneclaey(1) and Marie-Laure Boucheret(3)

1 TELIN Department, Ghent University, St-Pietersnieuwstraat 41
B-9000 GENT, BELGIUM
{nnoels, mm} @telin.UGent.be

2 DRT Department, Alcatel Alenia Space, 26 av. J.F. Champollion, BP 1187
F-31037 Toulouse, FRANCE
Mathieu.Dervin@support-externe.alcatelaleniaspace.com

3 IRIT-ENSEEIHT-TéSA, 2 rue Charles Camichel, BP 7122
F-31071 Toulouse Cedex 7, FRANCE
marie-laure.boucheret@enseeiht.fr

INTRODUCTION

The last decade has seen the development of powerful channel codes such as turbo codes and low-density parity-check (LDPC) codes. The impressive bit error rate (BER) performance of the associated decoding processes implicitly assumes coherent detection, meaning that the carrier phase must be recovered accurately before the data is decoded. However, since the receiver usually operates at extremely low signal-to-noise ratio (SNR) values, conventional carrier synchronizers often fail to provide sufficiently accurate phase estimates. Numerous efforts to improve the carrier synchronization at low SNR have resulted in a myriad of different transceiver schemes; all based on the exploitation of some form of a priori information regarding the conveyed data during the phase estimation process. The resulting synchronization algorithms may be distinguished into two categories: category-I and category-II.

Category-I algorithms take advantage of the structure that the channel code enforces upon the data stream, and which can be considered as a kind of a priori information that is available at the receiver. Assuming that the carrier phase (after possible frequency correction) is essentially constant over a codeword, synchronizers that accept soft information from the channel decoder have been presented [1,2], which yield a mean square estimation error (MSEE) that is close to the Cramer-Rao lower bound at the normal operating SNR of the code. However, in the presence of phase variations due to phase noise, the carrier phase estimate must be updated at regular intervals during the codeword; when the channel decoder needs a long observation interval before decoding can start (as is the case with turbo codes and LDPC codes), obtaining soft information to aid the synchronizer is rather cumbersome. A practical solution to this problem was proposed in [3] for a classical turbo code: the synchronizer based on soft information from only one of the concatenated convolutional decoders (operating on a reduced number of observations) has shown to yield satisfactory performance. However, as the approach from [3] cannot be extended to more general codes (such as LDPC codes), researchers have investigated a second category of algorithms. *Category-II algorithms* completely ignore the underlying channel code, and use only additional redundancy that is introduced into the transmission scheme for the sake of synchronization. This approach permits to control the complexity and the update period of the synchronizer, but brings about a reduction in the power and bandwidth efficiency of the overall system.

In this paper we present a general formulation for *maximum likelihood* (ML) based *feedback* phase synchronizers. The category-I feedback phase synchronizer considered in [2,3] and the category-II feedback phase synchronizers considered in [2,4] can be derived as special cases of the proposed general approach. This gives new insights into the relation between the different strategies. A comparison of two category-II feedback phase synchronizers, causing the same reduction of power and bandwidth efficiency, is then considered. The first scheme [4] adds a short single-parity-check (SPC) code to aid synchronization; the second scheme [2] is based on the insertion of known pilot bits. Both schemes have recently been developed to track a time-varying carrier phase at very low SNR. In order to test the algorithms in realistic conditions, we base our study on the transmission system defined by the recently published DVB-S2 standard [5]. The feedback synchronizer is a second-order phase-locked loop (PLL) with damping factor equal to 0.707. We evaluate the synchronizer performance in the case of a constant carrier phase and in the presence of phase noise. The results indicate that the operating SNR determines which system (with SPC code or with pilot bits) performs best.

SYSTEM AND CHANNEL MODEL

Fig. 1 depicts the general block diagram of the system under consideration. A continuous stream of information bits is sent from the transmitter to the receiver. In order to protect the information against the detrimental effects of the channel, the bits are first channel encoded. This yields a coded bit stream $\mathbf{u}$. To aid synchronization some extra redundancy may be added to the bit stream $\mathbf{u}$ that leaves the channel encoder, yielding a bit stream $\mathbf{c}$. The bit stream $\mathbf{c}$ is then chopped into blocks of length $\log_2 M$. Each group of $\log_2 M$ bits is converted into a complex symbol, that is part of a M -point signaling constellation $\Omega = \{\omega_0, \omega_1, ..., \omega_{M-1}\}$, with average symbol energy $E_s = \frac{1}{M}\sum_{m=0}^{M-1} |\omega_m|^2$. The resulting stream of data symbols $\mathbf{a} = \{a(l)\}$ is filtered by a square-root Nyquist transmit pulse and transmitted over the channel. The transmitted signal is affected by a phase error and corrupted by additive white Gaussian noise (AWGN). The received signal is applied to a matched filter, and sampled at the optimal instants lT_S , where T_S denotes the symbol period. We obtain a sequence of matched filter output samples $\mathbf{r} = \{r(l)\}$ with

$$r(l) = a(l)e^{j\theta} + w(l), \quad l = 0,1,... \tag{1}$$

In (1), θ is the unknown carrier phase. The sequence $\{w(l)\}$ consists of independent zero-mean complex-valued Gaussian noise terms; $\mathrm{Re}[w(l)]$ and $\mathrm{Im}[w(l)]$ are statistically independent, and have a variance equal to $N_0/2$. Phase recovery is performed by a second-order PLL operating at an update rate $R_L = 1/(\nu T_S)$ which is an integer fraction of the symbol rate $1/T_S$. At instants iT_L, where $T_L = 1/R_L = \nu T_S$ denotes the update period, the phase error detector (PED) performs a suitable time-invariant nonlinear operation $X(.,.)$ on the ν matched filter output samples $\mathbf{r}_i = \{r(i\nu), r(i\nu+1), ..., r((i+1)\nu-1)\}$ and the current phase estimate $\hat{\theta}_i$, so that its output signal $x_i \equiv X(\mathbf{r}_i, \hat{\theta}_i)$ gives an indication about the sign and the magnitude of the instantaneous phase error $\phi_i = \theta - \hat{\theta}_i$. The PED output signal x_i is filtered by a linear time-invariant loop filter with transfer function $(\alpha(z-1)+\beta)/(z-1)$ in the z -domain, where the loop parameters α and β control the equivalent loop bandwidth B_L and the damping factor ς_L of the PLL. The sequence y_i at the loop filter output is used to update the phase estimate at the rate R_L, according to the following recursion: $\hat{\theta}_{i+1} = \hat{\theta}_i + y_i$.

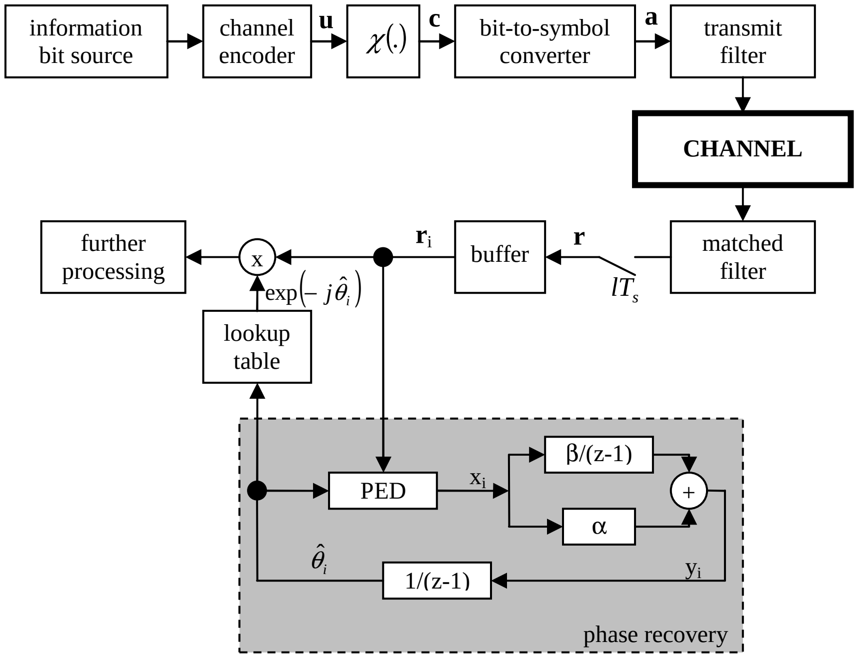

Fig. 1: Block diagram showing the transmission scheme and the receiver front-end. Phase recovery is performed with a second-order phase-locked loop (shaded area).

A UNIFIED FORMULATION OF ML-BASED FEEDBACK PHASE SYNCHRONIZERS

The probability density $p\left(\mathbf{r}\middle|\mathbf{a},\tilde{\theta}\right)$ of $\mathbf{r}$ resulting from (1), given the data sequence $\mathbf{a}$ and a trial value $\tilde{\theta}$ of the carrier phase, is given by

$$p\left(\mathbf{r}\middle|\mathbf{a},\tilde{\theta}\right) = C \cdot \prod_l F\left(r(l), a(l)e^{j\tilde{\theta}}\right) \tag{2}$$

with

$$F\left(r(l), a(l)e^{j\tilde{\theta}}\right) = \exp\left(-\frac{1}{N_0} \,|\, r(l) - a(l)e^{j\tilde{\theta}} \,|^2\right) \tag{3}$$

In (2), C is a factor not depending on $\mathbf{a}$ or $\tilde{\theta}$, and the multiplication is over all nonnegative values of l. The *likelihood function* $p\left(\mathbf{r}\middle|\tilde{\theta}\right)$ of the carrier phase is obtained by averaging $p\left(\mathbf{r}\middle|\mathbf{a},\tilde{\theta}\right)$ over the data symbols $\mathbf{a}$. Taking the logarithm then yields the *log-likelihood function* $L\left(\tilde{\theta}\right) = \ln\left(p\left(\mathbf{r}\middle|\tilde{\theta}\right)\right)$.

The symbol stream $\mathbf{a}$ usually consists of statistically independent sequences $\mathbf{a}_i$ of κ symbols long with a probability mass function $\Pr_{\mathbf{A}_i}[.]$; typically, $\mathbf{a}_i$ results from the bit-to-symbol conversion of the bit sequence $\mathbf{c}_i = \chi(\mathbf{u}_i)$, with $\mathbf{u}_i$ denoting the i-th codeword from the channel encoder and $\chi(.)$ representing the mapping function that introduces additional redundancy to aid the synchronization. This implies that the log-likelihood function $L\left(\tilde{\theta}\right)$ can be decomposed as follows: $L\left(\tilde{\theta}\right) = \sum_i L_i\left(\tilde{\theta};\kappa\right)$, with

$$L_i\left(\tilde{\theta};\kappa\right)=\ln\left(\sum_{\mathbf{a}_i} p(\mathbf{r}_i \mid \mathbf{a}_i,\tilde{\theta})\,\Pr_{\mathbf{A}_i}[\mathbf{a}_i]\right) \tag{4}$$

In (4), $\mathbf{r}_i$ denotes the matched filter output sample sequence corresponding to the i-th transmitted symbol sequence $\mathbf{a}_i$ and $p(\mathbf{r}_i \mid \mathbf{a}_i;\tilde{\theta})$ is given by (2) with l ranging from $i\kappa$ to $(i+1)\kappa-1$.

From a similar reasoning as in [2], the derivative $L'\left(\tilde{\theta}\right)$ of the log-likelihood function with respect to $\tilde{\theta}$ can then be manipulated into the following form: $L'\left(\tilde{\theta}\right)=\sum_i L_i'\left(\tilde{\theta};\kappa\right)$, where

$$L_i'\left(\tilde{\theta};\kappa\right)=\frac{2}{N_0}\sum_{l=i\kappa}^{(i+1)\kappa-1}\mathrm{Im}\left[\mu^*\left(l;\mathbf{r}_i,\tilde{\theta}\right)r(l)e^{-j\tilde{\theta}}\right] \tag{5}$$

and

$$\mu\left(l;\mathbf{r}_i,\tilde{\theta}\right)=E\left[a(l)\middle|\mathbf{r}_i,\tilde{\theta}\right]=\sum_{m=0}^{M-1}\Pr\left[a(l)=\omega_m\middle|\mathbf{r}_i,\tilde{\theta}\right]\omega_m \tag{6}$$

is the a posteriori expectation of the symbol $a(l)$ conditioned on $\mathbf{r}_i$ and $\tilde{\theta}$. In (5), $\Pr\left[a(l)=\omega_m\middle|\mathbf{r}_i,\tilde{\theta}\right]$ denotes the marginal a posteriori probability (APP) of the l-th data symbol $a(l)$, and $\Omega=\{\omega_0,\omega_1,...,\omega_{M-1}\}$ is the set of constellation points. By analogy with the estimation of continuous-valued parameters, the conditional mean $\mu\left(l;\mathbf{r}_i,\tilde{\theta}\right)$ can be considered as the *soft* (since taking values in some continuous result space) minimum mean square error *decision* regarding $a(l)$ based upon the received samples $\mathbf{r}_i$ and a trial value $\tilde{\theta}$ of the carrier phase.

The *ML estimate* of the carrier phase is the value of $\tilde{\theta}$ that makes zero the derivative of the log-likelihood function. Taking into account that in general the decoder must observe the entire sequence $\mathbf{r}_i$ before the soft decisions regarding any of the symbols $a(i\kappa)$, ..., $a((i+1)\kappa-1)$ can be computed, it is natural to take the update period equal to the symbol sequence duration, i.e., $T_L=\kappa T_S$ or $\nu=\kappa$, and to define the PED output at instant iT_L proportional to the right hand side of (5), with $\tilde{\theta}$ replaced by the estimate $\hat{\theta}_i$ and $\kappa=\nu$: e.g.

$$x_i=\frac{N_0}{2E_s\nu}L_i'\left(\hat{\theta}_i;\nu\right) \tag{7}$$

We speak of *soft-decision-directed* (SDD) PLL operation. The performance of the synchronizer is determined by the reliability of the soft symbol decisions $\mu\left(l;\mathbf{r}_i,\tilde{\theta}\right)$ feeding the PED, which in turn are affected by the channel code, the mapping $\chi(.)$, and the bit-to-symbol conversion rule. A performance limit is obtained by assuming in (6) $\mu\left(l;\mathbf{r}_i,\tilde{\theta}\right)\approx a(l)$ for all l. This corresponds to data-aided (DA) PLL operation, i.e. the phase estimate is updated once per symbol interval and the PED knows all data symbols in advance. When the phase to be estimated is constant, DA PLL operation yields

$$E[(\hat{\theta}-\theta)^2]=\frac{N_0B_LT_s}{E_s} \tag{8}$$

The right-hand side of (8) equals the modified Cramer-Rao bound (MCRB) [6], which is a lower bound on the achievable mean-square estimation error (MSEE) resulting from a PLL.

PRACTICAL ESTIMATION STRATEGIES

Straightforward application of the above procedure leads to an optimum category-I SDD PLL that fully exploits the redundancy due to channel coding. This is commonly referred to as (channel-)*code-aided* (CA) SDD PLL operation. However, the problem with many state-of-the-art channel coding schemes (turbo codes, LDPC codes) is that the sample sequences $\mathbf{r}_i$ involved in (6) become very long, yielding a low update rate of the carrier phase estimate, which makes it difficult to cope with phase variations.

A practical solution to this problem is to design the PED output according to a *modified* likelihood function that follows from modeling the channel encoder output $\mathbf{u}$ as a stream of independent and identically distributed (i.i.d.) random bits. Incorporating the resulting PED output in a PLL results in a category-II synchronizer that ignores the underlying (turbo or LDPC) channel code and exploits only the outer redundancy introduced by the one-to-one mapping $\mathbf{c}_i = \chi(\mathbf{u}_i)$.

An important issue that has yet to be addressed is the type of redundancy that is introduced in the transmission scheme to improve the synchronization. Ideally, the mapping rule $\chi(.)$ should provide a good compromise between power and bandwidth efficiency, computational complexity, and phase estimation accuracy. Our interest goes especially to mappings that keep the symbol sequence length κ relatively small. In the following we consider three types of category-II ML-based PLLs. The corresponding synchronization algorithms are derived under the assumption that $\mathbf{u}$ consists of i.i.d. bits, i.e., the operation of the channel encoder is ignored by the synchronizer.

PLL with Non-Code-Aided Operation

We speak of *non-code-aided* (NCA) operation when the PLL does not exploit any additional redundancy at all. This is equivalent with assuming that $\chi(\mathbf{u}_i) = \mathbf{u}_i$, for any κ. In that case, the symbols $a(l)$ are i.i.d., so that the APPs involved in (6) reduce to

$$\Pr[a(l) = \omega \,|\, \mathbf{r}_i, \hat{\theta}_i] \propto F\left(r(l), \omega e^{j\hat{\theta}_i}\right) \tag{9}$$

for $l \geq 0$ and $\omega \in \Omega$, and where $\propto$ denotes equality up to a normalization constant. As the APP of $a(l)$ depends on $r(l)$ rather than the entire $\mathbf{r}_i$, we can take the update rate of the NCA PLL equal to the symbol rate ($\nu = 1$), and use the PED output that is obtained by substituting (9) into (6), and onwards into (7) with $\nu = 1$.

PLL with Single-Parity-Check Code-Aided Operation

We speak of *single-parity-check code-aided* (SPC-CA) operation when the PLL exploits (only) the properties of a short $(k+1, k)$ SPC code that is added to the bits that leave the (turbo or LDPC) channel encoder. This is equivalent to assuming that $\chi(\mathbf{u}_i) = \{\mathbf{u}_i, \oplus_{\mathbf{u}_i}\}$, where $\mathbf{u}_i$ is i-th k-bit subsequence of $\mathbf{u}$ and $\oplus_{\mathbf{u}_i}$ denotes the modulo-2 sum of the bits in $\mathbf{u}_i$. To simplify the problem it is assumed that the ratio $\kappa = (k+1)/(\log_2 M)$, where M denotes the number of constellation points, is strictly positive and integer-valued. Letting $W(\omega)$ denote the modulo-2 sum of the bits corresponding to the symbol ω, the APPs of $a(l)$ can be exactly computed according to the following equation [4]

$$\Pr\left[a(l) = \omega \middle| \mathbf{r}_i, \hat{\theta}_i\right] \propto F\left(r(l), \omega e^{j\hat{\theta}_l}\right)\left\{1 + \left(1 - 2W(\omega)\right) \prod_{j=i\kappa}^{(i+1)\kappa-1} \frac{\left(1 - 2\psi_j\right)}{\left(1 - 2\psi_l\right)}\right\} \tag{10}$$

for $l \in [i\kappa, (i+1)\kappa - 1]$ and $\omega \in \Omega$. In (10), the quantities ψ_j in are given by

$$\psi_j = \sum_{\omega \in \Omega : W(\omega) = 1} F\left(r(j), \omega e^{j\hat{\theta}_l}\right) \tag{11}$$

where the summation is over all constellation points for which the modulo-2 sum of the corresponding bits equals one.

As the APP of $a(l)$ depends on the entire $\mathbf{r}_i$, the SPC-CA PLL operates at $1/\kappa$ times the symbol rate ($\nu = \kappa$), and uses the PED output that is obtained by substituting (10) into (6), and onwards into (7) with $\nu = \kappa$. It follows from (10)-(11) and (9) that SPC-CA PLL operation requires roughly κ times more computations per symbol interval than NCA PLL operation.

PLL with Pilot-Bit-Aided Operation

We speak of *pilot-bit-aided* (PBA) operation when the PLL knows some of the transmitted bits in advance. This is equivalent with assuming that $\chi(\mathbf{u}_i) = \{\mathbf{u}_i, p_0(i)\}$ where $\mathbf{u}_i$ is the i-th fixed-length subsequence of $\mathbf{u}$ and $p_0(i)$ denotes a deterministic pilot bit. Adding one pilot bit per k coded bits yields the same reduction of power and bandwidth efficiency as a $(k+1, k)$ SPC code. The major advantage of using pilot bits in stead of an SPC code is that the data symbols $a(l)$ remain statistically independent. To compute the APPs involved in (6) we have to distinguish between $(l+1)$ a multiple of κ, and $(l+1)$ not a multiple of κ, with $\kappa = (k+1)/(\log_2 M)$. We obtain

for $l \in [i\kappa, (i+1)\kappa - 2]$ and $\omega \in \Omega$, and

$$\Pr[a(l) = \omega \mid \mathbf{r}_i, \hat{\theta}_i] \propto \begin{cases} F\left(r(l), \omega e^{j\hat{\theta}_i}\right), & \text{for } \omega[\log_2 \mathrm{M} - 1] = p_0(i) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

for $l = (i+1)\kappa - 1$ and $\omega \in \Omega$. In (13), $\omega[n]$ denotes the n-th bit that is converted into the symbol ω. As the APP of $a(l)$ depends on $r(l)$ rather than the entire $\mathbf{r}_i$, the PBA PLL operates at the symbol rate ($\nu = 1$) and uses the PED output that is obtained by substituting (12)-(13) into (6), and onwards into (7) with $\nu = 1$. Comparing (12)-(13) with (9) it is easily verified that PBA PLL operation does not require more computations per symbol interval than NCA PLL operation.

NUMERICAL RESULTS

In this section we compare the above strategies by means of computer simulations. Results for SPC-CA and PBA operation are obtained with κ = 2, 3, 4. We consider Gray-mapped 4-PSK. The loop damping factor ς_L is set to 0.707.

Fig. 2 shows the ratio MSEE/MCRB assuming that the carrier phase is constant and the ratio of the normalized equivalent loop bandwidth $B_L T_S$ is 10^{-4}. Fig. 3 shows the MSEE in the presence of phase noise that satisfies the DVB-S2 spectral template defined in [5]; at each value of E_s/N_0 the loop bandwidth has been selected to minimize the MSEE. Both with and without phase noise, we observe that:

- For given E_s/N_0 and given κ, both the SPC-CA PLL and the PBA PLL yield a better synchronizer performance than the NCA PLL that does not require overhead.
- For given κ, the SPC-CA PLL yields better synchronizer performance than the PBA PLL at moderate to high SNR, whereas the opposite holds at low SNR.
- For given E_s/N_0, the synchronizer performance of both systems improves with decreasing κ.

We conclude that the operating SNR determines which system (with SPC code or with pilot bits) performs best. Table 1 shows the operating SNRs of the LDPC codes in the DVB-S2 standard [5], assuming perfect timing and carrier recovery, code words of 64800 bits, and 50 LDPC decoding iterations. Hence, for κ=2, using pilot symbols is better for the LDPC codes with a rate R ≤ 1/2, whereas using the SPC code is better for rates R >1/2.

Table 1. Operation signal-to-noise ratios proposed in the DVB-S2 standard. Guarantee a packet error rate (PER) below 10^{-7}.

signaling constellation and code rate	**E_s/N_0 required for a PER < 10^{-7}**
4-PSK, R=1/4	-2.35 dB
4-PSK, R=1/3	-1.24 dB
4-PSK, R=2/5	-0.30 dB
4-PSK, R=1/2	1.00 dB
4-PSK, R=3/5	2.23 dB
4-PSK, R=2/3	3.10 dB
4-PSK, R=3/4	4.03 dB
4-PSK, R=4/5	4.68 dB
4-PSK, R=5/6	5.18 dB
4-PSK, R=9/10	6.42 dB

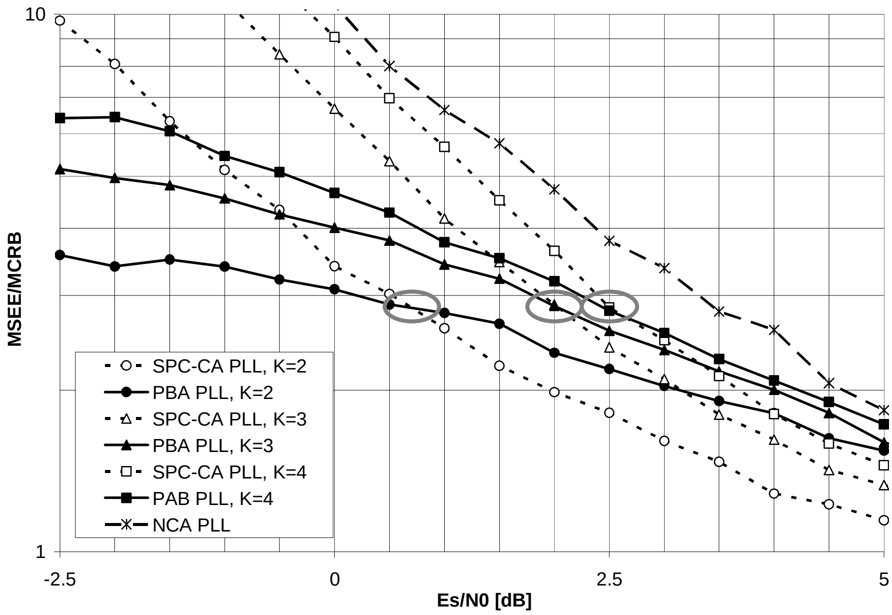

Fig. 2: MSEE/MCRB resulting from a non-code-aided (NCA), a single-parity-check code-aided (SPC-CA) and a pilot bit aided (PBA) second order phase locked loop (PLL) as a function of the SNR, assuming a normalized equivalent loop bandwidth of 10^{-4}, a damping factor of 0.707, a constant carrier phase and a 4-PSK signal constellation with Gray mapping.

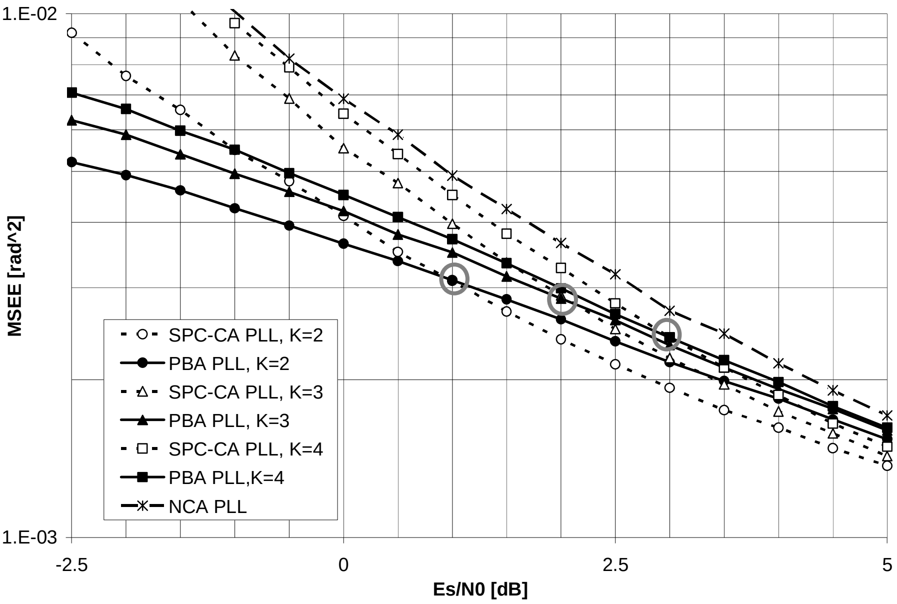

Fig. 3: MSEE resulting from a non-code-aided (NCA), a single-parity-check code-aided (SPC-CA) and a pilot bit aided (PBA) second order phase locked loop (PLL) as a function of the SNR, assuming an optimal equivalent loop bandwidth, a damping factor of 0.707, phase noise that satisfies the DVB-S2 spectral template defined in [5], and a 4-PSK signal constellation with Gray mapping.

ACKNOWLEDGEMENT

This work has been performed in the framework of the European Network NEWCOM (E.C. Contract no. 507325) en was supported by the Interuniversity Attraction Poles Program P5/11 – Belgian Science Policy.

REFERENCES

[1] N. Noels et al, “A Theoretical Framework for Soft Information Based Synchronization in Iterative (Turbo) Receivers,” *EURASIP J. on Wireless Communications and Networking*, vol. 2005, NO. 2, April 2005, pp. 117-129

[2] N. Noels, H. Steendam and M. Moeneclaey, "Performance Analysis of ML-Based Feedback Carrier Phase Synchronizers for Coded Signals" , accepted for publication in *IEEE Transactions on Signal Processing*

[3] C. Langlais and M. Helard, “Phase carrier recovery for turbo codes over a satellite link with the help of tentative decisions,” in *Proc. International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept. 2000, pp. 439-442

[4] M. Dervin, M.L. Boucheret, G. Mesnager and A. Ducasse, “Phase detection involving parity equations and suited to transmissions at low signal to noise ratio,” in *Proc. IEEE International Conference on Communications* (ICC), volume 3, pages 2080-2084, May 2005

[5] Digital Video Broadcasting (DVB) – Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News gathering and other Broadband Satellite Applications, ETSI EN 302 307 v1.1.1 (2004-01)

[6] A.N. D’Andrea, U. Mengali and R. Reggiannini, “The modified Cramer-Rao bound and its applications to synchronization parameters,” *IEEE Transactions on Communications*, vol. COM-24, pp. 1391-1399, Feb./Mar./Apr. 1994