Carrier Phase Tracking From Turbo and LDPC Coded Signals Affected By a Frequency Offset

Nele Noels, *Student Member, IEEE*, Heidi Steendam, *Member, IEEE*, and Marc Moeneclaey, *Fellow, IEEE*

***Abstract*— This contribution considers carrier phase estimation from *coded* signals, affected by a frequency offset at *low* operating signal-to-noise ratio (SNR). We derive a maximum likelihood (ML) based iterative code-aided feedback phase-tracker suited for receivers with iterative maximum-a-posteriori (MAP) detection. Simulations indicate that the proposed synchronizer is considerably more robust against frequency offsets than a feedforward synchronizer.**

***Index Terms*— Synchronization, code-aided estimation, phase tracking, feedback.**

I. Introduction

The impressive BER performance of powerful codes implicitly assumes coherent detection, i.e., the carrier phase must be recovered accurately before data detection. Synchronization for encoded systems is yet a very challenging task since the receiver usually operates at extremely low SNR values. A first strategy (*separate detection and decoding*) is to estimate the carrier phase, rotate the received signal accordingly, and apply the rotated signal to exactly the same decoder as for coherent detection (see [1] and the references therein); [1] and [2] appear to be the only papers considering a nonzero frequency offset. Another strategy (*combined detection and decoding*) is to modify the decoder operation by taking into account the phase statistics or using per survivor phase estimates inside the decoder (see [3] and references therein). The latter strategy will not be considered in this contribution, because its computational complexity is higher than for the former strategy.

The iterative feedforward algorithm proposed in [1] converges to the true ML carrier phase and frequency estimates, provided that the frequency offset is small as compared to the inverse of the data burst length. In [2], feedback phase estimation with a second-order update loop has been adopted to cope with a carrier frequency offset. The major shortcoming of [2] is the lack of a mathematical foundation for the choice of the synchronizer. This contribution considers the feedback counterpart of [1]. Unlike in [2], the proposed algorithm is derived from the ML criterion. We obtain an iterative soft-decision-directed phase-tracker suited for application in a receiver with iterative MAP decoding. The proposed synchronizer is shown to tolerate frequency offsets that the feedforward synchronizer from [1] cannot handle.

Manuscript received March 23, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. Ernesto Conte. This work was supported by the Interuniversity Attraction Poles (IAP) Program P5/11, Begian Science Policy, Fedral Office for Scientific, Technical and Cultural Affairs, Belgium, and the EU 6FP IST Network of Excellence in Wireless Communications (NEWCOM), a project funded by the European Community.

The authors are with the Telecommunications and Information Processing Dept., Ghent University, Gent, Belgium (e-mail: {nnoels, hs, mm, hb@telin.UGent.be).

Digital Object Identifier 10.1109/LCOMM.2005.10020.

II. ML-Based Phase Tracking for Coded Signals

Consider a linear modulation, additive white Gaussian noise (AWGN) w_k, a random carrier phase θ and an unknown but deterministic static frequency offset F. Assuming perfect timing information at the receiver, the matched filter output samples at the correct decision instants t=kT are given by

$$r_k = a_k exp(j\theta_k) + w_k, \quad k \in I = \{0, ..., L-1\} \tag{1}$$

where a_k is the k-th transmitted symbol and $\theta_k = \theta + 2\pi F k T$. The data symbols $\{a_k\}$ are obtained from the encoding of information bits and a proper mapping on an M-fold signal constellation $S = \{s_0,\ s_1,\ ...,\ s_{M-1}\}$. Pilot symbols may also be inserted in $\{a_k\}$. For $k \in I_p \subset I$, the symbol a_k denotes a pilot symbol. For $k \in I_d = I|I_p$, a_k denotes a data symbol.

The proposed iterative (i.e., iteration between the decoder and the synchronizer) phase-tracker is shown in the shaded area of Fig. 1. The phase error detector (PED) output $x_k^{(i)}$ at instant kT provides an indication of the phase estimation error $(\theta_k - \hat{\theta}_k^{(i)})$, where $\hat{\theta}_k^{(i)}$ denotes the estimate of θ_k during the i-th iteration. The PED is incorporated in a type-II phase-locked loop (PLL), which is known to yield zero steady-state phase error in the presence of a nonzero frequency offset [4]. The associated loop filter has a transfer function [a(z-1)+b]/(z-1), with a and b determining the damping factor ζ and the loop bandwidth $B_L T$ of the PLL [4]. The quantity $\hat{F}_k^{(i)}$ in Fig. 1 can be interpreted as an estimate of F at instant kT in the i-th iteration [4]. In the i-th iteration the phase and frequency estimates are updated according to the following recursion:

$$\begin{pmatrix} \hat{\theta}_{k+\alpha}^{(i)} \\ 2\pi\alpha\hat{F}_{k+\alpha}^{(i)}T \end{pmatrix} = \begin{pmatrix} \hat{\theta}_k^{(i)} + 2\pi\alpha\hat{F}_k^{(i)}T + a x_k^{(i)} \\ 2\pi\alpha\hat{F}_{k+\alpha}^{(i)}T + b x_k^{(i)} \end{pmatrix} \tag{2}$$

for $k \in I$. In (2), α=1 and α=-1 correspond to a forward recursion and a backward recursion, respectively. Conventionally, only forward recursions are used in PLLs, but we will also consider backward recursions (see further).

The PED is designed according to the ML criterion. From a similar reasoning as in [1], the derivative of the log-likelihood function, related to (1), is given by

$$\frac{d}{d\tilde{\theta}_k} ln(p(\mathbf{r};\tilde{\boldsymbol{\theta}})) = Im\left[A(\mathbf{r},\tilde{\boldsymbol{\theta}}) r_k e^{-j\tilde{\theta}_k}\right] \tag{3}$$

where $\mathbf{r} = (r_0 r_1 ... r_{L-1})$, and $\tilde{\boldsymbol{\theta}} = \left(\tilde{\theta}_0\tilde{\theta}_1...\tilde{\theta}_{L-1}\right)$ is a vector of trial phase values. For $k \in I_d$, $A_k(\mathbf{r},\tilde{\boldsymbol{\theta}})$ in (3) denotes the a

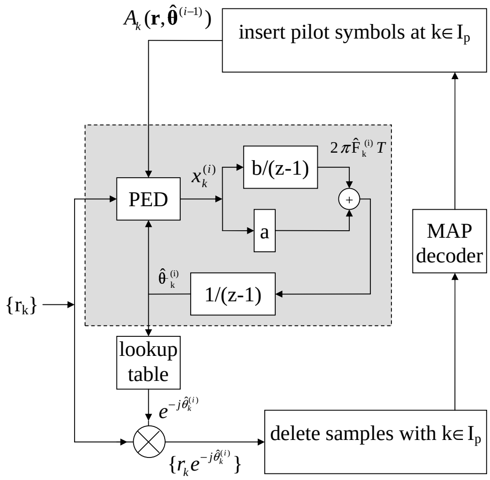

Fig. 1. Receiver with code-aided iterative phase-tracking.

posteriori mean (conditioned on $\mathbf{r}$ and $\tilde{\boldsymbol{\theta}}$) of the coded symbol a_k, i.e., $A_k(\mathbf{r},\tilde{\boldsymbol{\theta}}) = \sum_{m=0}^{M-1} Pr[a_k = s_m|\mathbf{r},\tilde{\boldsymbol{\theta}}]s_m$. It can be considered as a soft decision (SD) regarding a_k, based upon the received vector $\mathbf{r}$ and the phase vector $\tilde{\boldsymbol{\theta}}$. The a posterior probabilities $Pr[a_k = s_m|\mathbf{r},\tilde{\boldsymbol{\theta}}]$ can be efficiently computed by means of a MAP decoding algorithm. For $k \in I_p$, $A_k(\mathbf{r},\tilde{\boldsymbol{\theta}})$ simply equals a_k. As the ML phase estimate makes zero the derivative of the log-likelihood function, we are tempted to define the PED output at instant kT as the right-hand side of (3), with $\tilde{\boldsymbol{\theta}}$ replaced by the estimate $\hat{\boldsymbol{\theta}} = \left(\hat{\theta}_0\hat{\theta}_1...\hat{\theta}_{L-1}\right)$. However, at instant kT, the estimates $\hat{\theta}_j$ with j > k are not yet available. This problem is circumvented by performing *iterated* estimation: during the i-th iteration, the PED output at instant kT is given by

$$x_k^{(i)} = Im\left[A_k^{(i)*} r_k e^{-j\hat{\theta}_k^{(i)}}\right], \quad (i > 0) \tag{4}$$

where the SDs $A_k^{(i)} = A_k\left(\mathbf{r},\hat{\boldsymbol{\theta}}^{(i-1)}\right)$ needed in the i-th iteration are computed from the phase estimate vector $\hat{\boldsymbol{\theta}}^{(i-1)}$ obtained during the (i-1)-th iteration. The phase estimate vector $\hat{\boldsymbol{\theta}}^{(0)}$ needed to start the iterations is obtained by performing (2),(4) with i = 0, but utilizing SDs that disregard the code properties ('*non-code-aided*' recursion, as opposed to a '*code-aided*' recursion that takes code properties into account). It is easily verified from (1) that $A_k(\mathbf{r},\tilde{\boldsymbol{\theta}})$ depends only on r_k and $\tilde{\theta}_k$ (instead of $\mathbf{r}$ and $\tilde{\boldsymbol{\theta}}$) when assuming uncoded symbols; hence iteration i=0 requires no phase estimates from a previous iteration. More specifically, one obtains: $A_k(\mathbf{r},\tilde{\boldsymbol{\theta}}) = C\ \sum_{m=0}^{M-1} exp\left(-\frac{E_s}{N_0}\left|r_k e^{-j\tilde{\theta}_k} - s_m\right|^2\right) s_m$.

At each iteration, the PLL must be initialized with proper phase and frequency estimates. At iteration i=0, we apply a non-code-aided forward recursion initialized with $(\hat{\theta}_0^{(0)}, \hat{F}_0^{(0)})$. However, as these initial estimates are usually not accurate, an acquisition transient occurs at iteration i=0, during which the phase error may assume large values. Assuming that the transient is no longer present near the end of the burst, we can avoid using the inaccurate phase estimates during the transient of iteration i=0 for computing SDs of the coded symbols needed in iteration i=1, by applying for i=1 a non-code-aided backward recursion, initialized by the (assumed accurate) estimates $(\hat{\theta}_L^{(0)}, \hat{F}_L^{(0)})$ at the end of iteration i=0. For iterations i=2,3,..., code-aided recursions are used, forward for even i and backward otherwise, with each recursion initialized by the estimates obtained at the end of the previous iteration.

The interaction between the MAP decoder and the synchronizer is shown in Fig. 1. Strictly speaking, the iterative (turbo or LDPC) decoders are required to converge at each synchronizer iteration. However, to reduce the overall computational complexity, the synchronizer iterations are merged with the decoder iterations, i.e., after each synchronizer iteration only one decoder iteration is performed (without resetting extrinsic probabilities). The details of this procedure are outlined in [1]. In general, considering more than one internal decoder iteration does not help the overall decoding process.

III. Numerical Results and Discussion

We consider a rate r=1/3 turbo code consisting of the parallel concatenation of two identical non-recursive systematic convolutional codes with generator polynomials $(21)_8$ and $(37)_8$ in octal notation, separated by a pseudo-random interleaver of size N=3333 bits. The mapping is BPSK; a preamble and postamble of 32 pilot symbols each are added. Hence, L=10063. The nominal operating point is E_b/N_0 = 0.4 dB (in absence of pilot symbols), yielding a bit error rate (BER) of about 10^{-4} when synchronization is perfect. Results not shown here indicate that the corresponding root mean square (r.m.s) phase error should not exceed 5 degrees, in order to yield a negligible BER degradation. A total of 10 iterations of the forward-backward iterative feedback synchronizer are carried out. We take for $\hat{\theta}_0^{(0)}$ a data-aided phase estimate obtained from the preamble, and set $\hat{F}_0^{(0)} = 0$. We use the postamble to correct $\hat{\theta}_L^{(0)}$ for *cycle slips* [4] during iteration i=0, before starting iteration i=1. The loop filter parameters are selected such that $\zeta = 1/\sqrt{2}$ and $B_L T = 2.5\ 10^{-3}$, yielding an r.m.s. phase error of 4.7 degrees, when the SDs would equal the true data symbols[1] [4].

The performance of the feedback synchronizer is assessed in terms of BER versus $\frac{E_b}{N_0} = \frac{E_s}{N_0}\frac{L}{N}$ (see Fig. 2). For the sake of comparison we also show the BER for a perfectly synchronized system (both with and without pilot symbols), and the BER as obtained with the code-aided feedforward estimator from [1] operating on the same signal and with identical initialization. We observe that there is a critical value F_{max} of |F|, below which the synchronizers yield a negligible BER degradation as compared to the perfectly synchronized receiver (taking into account the power loss of 10log(rL/N)≊0.03dB caused by pilot symbol insertion), and above which the synchronizers break down. We have $F_{max}^{(FB)}T \approxeq B_L T/3 \approxeq 8\ 10^{-4}$, and $F_{max}^{(FF)}T \approxeq 3/(8L) \approxeq 3.75\ 10^{-5}$. Hence, as compared to the FF synchronizer, the FB synchronizer can handle about 20 times larger frequency offsets.

[1]The SDs are close to the true data symbols when the synchronizer/decoder iterations have converged.

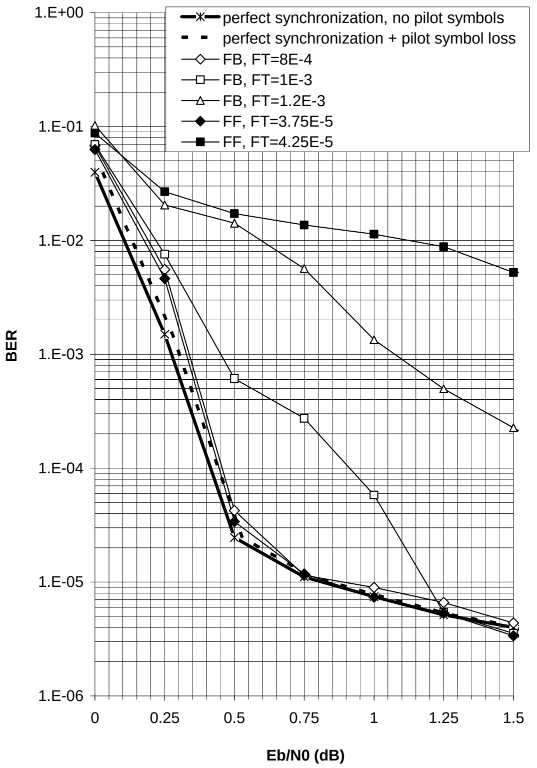

Fig. 2. BER performance of turbo code with feedforward or feedback carrier synchronization.

The FB algorithm can be considered succesfull if in the vast majority of the situations the acquisition transient is short as compared to the data burst duration LT. When the acquisition transient exceeds LT, the estimates $\hat{\theta}_L^{(0)}$ and $\hat{F}_L^{(0)}$ at the end of iteration i=0 are inaccurate; this yields cycle slips during iteration i=1, resulting in large phase estimation errors, which, in turn, prevent correct convergence of the joint detection and synchronization process in subsequent iterations. As the mean acquisition time increases with the frequency offset FT, the FB algorithm breaks down for FT > $F_{max}^{(FB)}T$. This problem can be solved (at the expense of additional computations) by carrying out 2 + floor(L_{acq}/L) (instead of 2) non-code-aided iterations, with L_{acq} denoting the duration (in symbol intervals) of the acquisition.

IV. Conclusions

We have considered carrier phase estimation in coded systems in the presence of a nonzero frequency offset. We propose an ML-based code-aided iterative feedback type-II phase-tracker which exploits the code structure without requiring modification of the standard decoder. When applied to a receiver with iterative MAP detection/decoding, the proposed phase estimation/compensation scheme yields very low additional complexity. Simulation results show that our code-aided feedback synchronizer can cope with frequency offsets that its feedforward counterpart from [1] cannot handle.

References

[1] N. Noels, V. Lottici, A. Dejonghe, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, “A theoretical framework for soft information-based synchronization in iterative (turbo) receivers," *EURASIP J. Wireless Commun. and Networking*, Special Issue on Advanced Signal Processing Algorithms for Wireless Communications, vol. 2005, pp. 117-129, Apr. 2005.

[2] W. Oh and K. Cheun, “Joint decoding and carrier recovery algorithms for turbo codes," *IEEE Commun. Lett.*, vol. 6, pp. 375-377, Sept. 2001.

[3] G. Ferrari, A. Anastasopoulos, G. Colavolpe, and R. Raheli, “Adaptive iterative detection for the phase-uncertain channel: limited-tree-search versus truncated-memory detection," *IEEE Trans. Veh. Technol.*, vol. 53, pp. 433-441, Mar. 2004.

[4] H. Meyr and G. Ascheid, *Synchronization in Digital Communications*. New York: Wiley, 1990