A Maximum-Likelihood Based Feedback Carrier Synchronizer for Turbo-Coded Systems

Nele Noels*, Heidi Steendam*, Marc Moeneclaey*, Herwig Bruneel+
Telecommunications and Information Processing Department (DIGCOM*, SMACS+ Group)
Ghent University
Gent, Belgium
{nnoels,hs,mm,hb}@telin.UGent.be

***Abstract*—This contribution considers the estimation of a *time-varying* carrier phase from coded signals at low operating SNR. ML estimation theory is used to derive an iterative code-aided feedback phase-tracker that is well suited for application in receivers with iterative MAP detection. Simulation results for a turbo-coded system indicate that, in the presence of carrier frequency offsets and phase noise that cannot be handled by a feedforward synchronizer, the proposed synchronizer yields only a small BER degradation as compared to a perfectly synchronized receiver.**

Carrier synchronization; phase locked loops; codes

I. INTRODUCTION

The impressive bit error rate (BER) performance of state-of-the-art 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 signal-to-noise ratio (SNR) values. Traditional phase estimation techniques may fail to cope with such a high noise environment. Evaluation of the Cramer-Rao bound (CRB) for carrier phase estimation in coded systems [1] has shown that synchronizers that exploit the code properties in the estimation process (so-called *code-aided* synchronizers) are potentially more accurate than synchronizers that do not exploit the code properties (so-called *non-code-aided* synchronizers) when operating on coded signals. The development of accurate code-aided phase estimation algorithms has received much attention in the recent technical literature.

In [1,2], we have proposed a *feedforward* (FF) carrier estimation technique for coded systems that exploits the knowledge about the underlying error-correcting code. In this FF structure synchronization is carried out iteratively. At each iteration, the current phase and frequency estimates are used by the decoder to produce soft information that is subsequently used by the synchronizer to update the phase and frequency estimate. We will refer to this scheme as “turbo”-FF (T-FF) synchronization in the sequel. At the normal operating SNR of coded systems, T-FF synchronization has been shown to perform very closely to a fictitious data-aided (DA) synchronizer that knows all transmitted symbols in advance, provided that sufficiently accurate initial estimates are available. Whereas the requirements on the initial phase estimate are easily met by a FF DA algorithm operating on a short pilot sequence that precedes the actual data, this is not the case for the initial frequency estimate. Convergence of the T-FF synchronizer requires that the difference between the frequency offset F and its initial estimate is small as compared to 1/LT, with T and L denoting the symbol interval and the number of transmitted symbols, respectively. For large L and low SNR (i.e., the common situation for turbo codes), obtaining such an accurate initial frequency estimate from a short pilot sequence is extremely hard because of the so-called threshold-phenomenon [1]. Therefore, the initial frequency estimate is set to zero, in which case T-FF carrier synchronization is a useful tool only for |FT| small as compared to 1/L.

In this contribution we develop a feedback (FB) phase estimator that exploits the code properties. The main advantage of FB algorithms as compared to FF algorithms is that they inherently have the ability to automatically track a slowly varying carrier phase. In [3-6], FB phase estimation has been adopted to cope with a carrier phase which is time-variant due to a frequency offset and/or phase noise. In [3], the receiver is preceded by an external “standard” FB loop, which runs once and effectively de-rotates the signal before it is processed by the “standard” decoder. In this simple scenario, the estimation process completely ignores the underlying code structure. The code-aided algorithm in [4] implicitly assumes that the phase variations are small over the frame duration (LT), and its ad-hoc derivation is far from optimum. The tentative decision-aided FB algorithm in [5] uses hard symbol decisions extracted from one of the two serial concatenated decoders of a turbo decoder and therefore loses a great part of the available information. The main drawback of [6] is that the standard decoder has to be modified considerably in order to embed phase estimation. In [7], algorithms are derived from a factor graph that includes both the code constraints and the statistics of a Wiener phase noise model. Unfortunately, the latter methods are only applicable if the carrier phase statistics are available. Here, we present a novel approach based on the iterative exchange of information between an additional phase tracking unit and a conventional decoder, so-called “turbo”-FB (T-FB) synchronization. The algorithm is derived from the ML criterion and may be viewed as the FB counterpart of the T-FF estimator presented in [1,2]. When applied to an iterative maximum a posteriori (MAP) decoder, it has a very low implementation complexity. Simulation results for a turbo-coded system show that the proposed T-FB carrier

This work was supported by the InteruniversityAttraction Poles Programme P5/11 – Belgian Science Policy

synchronizer can cope with phase noise and frequency offsets that the T-FF carrier synchronizer from [1,2] cannot handle.

II. A ML-Based Phase Tracking Synchronizer for Coded Transmission

A. Uni-directional T-FB algorithm

Consider a linear modulation, and a channel characterized by additive white Gaussian noise (AWGN) w_k, phase noise Θ_k and 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

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

where a_k is the k-th transmitted symbol, $\theta_k=\Theta_k+2\pi FkT$, and T denotes the symbol interval. The data symbols in the sequence $\{a_k\}$ are obtained from the encoding of a sequence of information bits and a proper mapping on a signal constellation. Pilot symbols may also be inserted in $\{a_k\}$.

From a similar reasoning as in [1], the derivative of the log-likelihood function, related to (1), is given by

$$\frac{d}{d\tilde{\theta}_k}\ln\left(p\left(\mathbf{r};\tilde{\boldsymbol{\theta}}\right)\right)=\mathrm{Im}\left[A_k^*(\mathbf{r},\tilde{\boldsymbol{\theta}})r_k e^{-j\tilde{\theta}_k}\right] \tag{2}$$

where

$$A_k(\mathbf{r},\tilde{\boldsymbol{\theta}}) = \begin{cases} a_k, & a_k \textit{ is pilot symbol} \\ \sum_{m=0}^{M-1}\Pr[a_k=\alpha_m \mid \mathbf{r},\tilde{\boldsymbol{\theta}}]\,\alpha_m, & \textit{otherwise}\end{cases} \tag{3}$$

and $(\alpha_0, \alpha_1, ..., \alpha_{M-1})$ denotes the set of M constellation points. The a posteriori mean $A_k(\mathbf{r},\tilde{\boldsymbol{\theta}})$ can be considered as a soft decision (SD) regarding a_k, based upon the received vector $\mathbf{r}$ and the phase vector $\tilde{\boldsymbol{\theta}}=(\tilde{\theta}_0,...,\tilde{\theta}_{L-1})$. In the case of coded transmission, the a posterior probabilities $\Pr[a_k=\alpha_m \mid \mathbf{r},\tilde{\boldsymbol{\theta}}]$ can be efficiently computed by means of a MAP soft decoding algorithm [8].

In a FB phase estimator derived from the ML criterion, the output signal x_k of the phase error detector (PED) at instant kT is given by (2), with $\tilde{\theta}_k$ replaced by the phase estimate $\hat{\theta}_k$. However a complication arises, as the SD A_k from (3) depends not only on $\hat{\theta}_k$ but rather on the entire phase estimate vector $\hat{\boldsymbol{\theta}}=(\hat{\theta}_0,...,\hat{\theta}_{L-1})$, which is not available at instant kT. This difficulty can be circumvented by considering iterative FB synchronization. During the i-th iteration, the PED output at instant kT is given by

$$x_k^{(i)} = \mathrm{Im}[\tilde{a}_k^{(i)*} r_k e^{-j\hat{\theta}_k^{(i)}}] \tag{4}$$

with $\tilde{a}_k^{(i)} = A_k(\mathbf{r},\hat{\boldsymbol{\theta}}^{(i-1)})$ from (3) and (i = 1, 2, ..., I). Note that during the i-th iteration the SDs $\tilde{a}_k^{(i)}$ are computed from the phase estimate vector $\hat{\boldsymbol{\theta}}^{(i-1)}$ obtained during the (i-1)-th iteration.

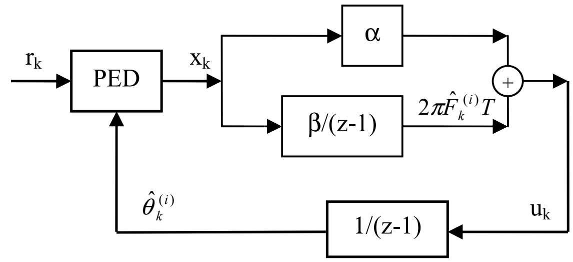

Figure 1. Block scheme of a discrete-time type-II phase locked loop

The PED is incorporated in a type-II phase-locked loop (PLL) [9]. The advantage of a type-II loop over a type-I loop is the zero steady-state phase error in the presence of a frequency offset. The state variables of the type-II loop during the i-th iteration are the phase estimate $\hat{\theta}_k^{(i)}$ and the frequency estimate $\hat{F}_k^{(i)}$:

$$\begin{pmatrix}\hat{\theta}_{k+1}^{(i)} \\ 2\pi\hat{F}_{k+1}^{(i)}T\end{pmatrix} = \begin{pmatrix}\hat{\theta}_k^{(i)} + 2\pi\hat{F}_k^{(i)}T + \alpha x_k \\ 2\pi\hat{F}_k^{(i)}T + \beta x_k\end{pmatrix} \tag{5}$$

where α and β are loop parameters that control the loop equivalent noise bandwidth B_L and damping factor ζ.

At each iteration, the PLL must be initialized with proper values of $\hat{\theta}_0^{(i)}$ and $\hat{F}_0^{(i)}$. Irrespective of the iteration index i, we take $\hat{F}_0^{(i)}$ = 0 and $\hat{\theta}_0^{(i)}$ = $\hat{\theta}_0$, where $\hat{\theta}_0$ is the DA phase estimate resulting from a short pilot sequence that precedes the actual data symbols.

Running the PLL during the i-th iteration using (4,5) yields the phase estimate vector $\hat{\boldsymbol{\theta}}^{(i)}$. The phase estimate vector $\hat{\boldsymbol{\theta}}^{(0)}$ needed to start the iterations is obtained by running the PLL with the following SDs:

$$\tilde{a}_k^{(0)} = C\sum_{m=0}^{M-1}\exp\left(-\frac{E_s}{N_0}\left|r_k e^{-j\hat{\theta}_k^{(0)}} - \alpha_m\right|^2\right)\alpha_m \tag{6}$$

where E_s = $E[|a_k|^2]$, N_0 = $E[|w_k|^2]$ and C is a normalization constant. The SD in (6) is obtained by disregarding the code structure, in which case the SD $\tilde{a}_k^{(0)}$ depends only on r_k and $\hat{\theta}_k^{(0)}$.

The aim of iterating with respect to i is to improve the reliability of the SDs $\tilde{a}_k^{(i)}$, which in turn improves the quality of the phase and frequency estimates. A performance limit of the synchronizer is obtained by assuming $\tilde{a}_k^{(i)}$ = a_k, which corresponds to DA FB operation (all symbols a priori known).

When applied to a turbo receiver with iterative MAP detection/decoding, the proposed phase estimation /compensation scheme yields very low additional complexity when the synchronizer iterations are merged with the decoder iterations [1,2]. I.e., after each synchronizer iteration one decoder iteration is performed without resetting extrinsic probabilities. This approach will be adopted in the simulations.

B. *Bi-directional iterative FB algorithm*

The drawback of the uni-directional T-FB carrier synchronizer described in section II.A is that in each iteration the presence of a nonzero frequency offset F gives rise to an acquisition transient, during which the phase error assumes large values. The duration of this transient is in the order of $1/B_L$, with B_L denoting the loop bandwidth.

This problem can be solved by applying bi-directional T-FB synchronization, where during odd and even iterations, the updating is performed in the forward (from the first symbol to the last) and backward (from the last symbol to the first) direction, respectively. The initial phase and frequency estimates for a given iteration equal the last phase estimate and minus the last frequency estimate from the previous iteration. In this way, an acquisition transient occurs only in the first iteration of the iterative FB synchronizer. In order to avoid that the SDs $\{\tilde{a}_k^{(1)} \mid k = 0, ..., L-1\}$ computed during iteration i=1 are affected by large phase transients, we perform a forward and a backward recursion of the non-code-aided PLL using (6) before starting the iterative process.

C. *Cycle slipping*

When the signal constellation is invariant under a rotation over an angle p, and the code structure is disregarded (e.g. during non-code-aided operation), the carrier synchronizer cannot distinguish between angles θ and θ + kp, with k = ±1, ±2,... As a result, the carrier synchronizer has infinitely many stable operation points, which are spaced by p. The same goes for code-aided operation. Although the rotational invariance is destroyed by the code structure, the usual p estimation ambiguity due to the symmetry of the constellation remains apparent during code-aided operation as can be found in [2]. Most of the time, the carrier phase estimate exhibits small random fluctuations about a stable operating point. Occasionally, noise or other disturbances push the estimate away from the current stable operating point, into the domain of attraction of a neighboring stable operating point. This phenomenon is called a cycle slip. After this, the estimate remains for a long time in the close vicinity of the new operating point, until the next slip occurs.

A cycle slips in the midst of a codeword will of coarse prevent error-free decoding. The probability of such an event increases exponentially with the phase error variance and linearly with the codeword length. Powerful error correcting codes usually work at low SNR and often have large block lengths; therefore, cycle slipping is a major performance limiting factor of our T-FB carrier synchronization system.

The occurrence and the direction of a slip can be detected by monitoring a known synchronization word (SW), which is, at regular intervals, inserted into the symbol stream to be transmitted. After a carrier cycle slip, the known symbols of the SW are found to have the wrong phase. The phase of all symbols following the synchronization word is corrected accordingly, before computing the SDs. Hence, the effect of the slip extends until the SW following the slip.

III. NUMERICAL RESULTS

We consider a rate 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 3333 bits. The encoder output is mapped onto a sequence of N_d=9999 BPSK symbols. Bursts are transmitted with a preamble of 32, and a postamble of 16 pilot symbols. A SW of 16 pilot symbols was inserted into the symbol stream once every 256 coded symbols. The total number of pilot symbols amounts N_p=672. The loop parameters from (5) are selected such that the loop bandwidth satisfies B_LT=0.0075 and the damping factor ζ equals 0.707, when the soft data decisions would equal the actual data symbols[1]. This choice of the loop bandwidth is a trade-off: small enough to reduce the effect of AWGN on the phase error, but large enough to track phase noise and to limit the acquisition time caused by a frequency error at the first iteration.

In order to evaluate the synchronizer's performance, two criteria are used; bit error rate (BER), and mean square error (MSE) of the phase estimate.

Figures 2 and 3 show the MSE as obtained with the uni-directional and the bi-directional T-FB synchronizer, respectively. The SNR equals $\frac{E_b}{N_0} = \frac{E_s}{N_0}\frac{1}{r}\frac{N_p + N_d}{N_d} = 1\,dB$, the frequency offset FT equals 10^{-3}, and Θ_k is modeled as a time-invariant uniformly distributed random variable in [-π,π[(no phase noise). For reasons of clarity we have chosen not to depict the MSE after iteration i=1 (performed in the forward direction). The quasi periodical behavior of the MSE (with period equal to 256+16) is a result of the slip detection/ correction process. Further, we observe that

- The MSE assumes very large values during the acquisition transient in each iteration of the uni-directional algorithm (Figure 2).
- The MSE assumes very large values during the acquisition transient in the initial forward recursion of the bi-directional algorithm (Figure 3, curve 'iteration i=0 forward'). Thanks to the additional backward initialization stage of the bi-directional T-FB synchronizer, this transient is not present in the final estimates $\{\hat{\theta}_k^{(0)}\}$, which are used by the decoder to compute the SDs $\{\tilde{a}_k^{(1)}\}$ during iteration i=1 (Figure 3, curve 'iteration i=0 backward').
- The MSE of the bi-directional T-FB algorithm exhibits a transient near the start of the data block in the forward (odd) iterations, and near the end of the block in backward (even) iterations. This is a result of (i) the reusing of the samples r_k,

[1] The soft decisions are close to the true data symbols when the synchronizer/decoder iterations have converged. During initial iterations the soft decisions are less reliable, which yields a smaller loop bandwidth and a lower damping factor.

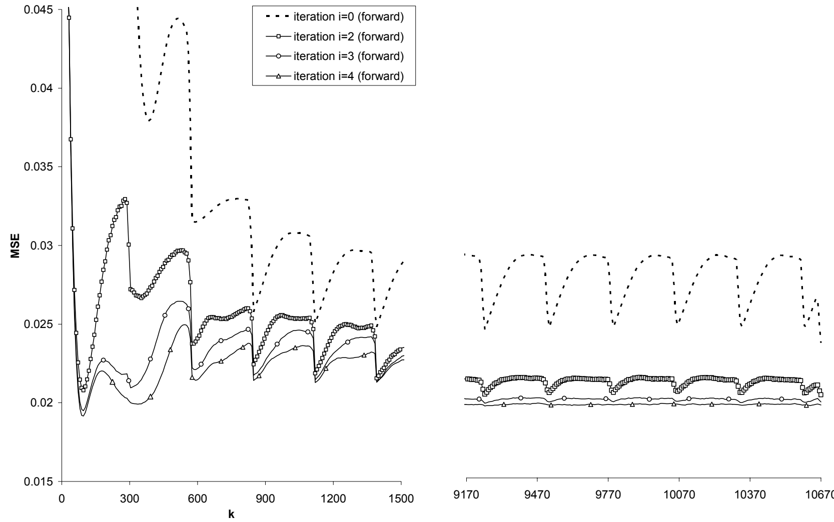

Figure 2. MSE obtained with a second-order uni-directional T-FB phase tracking loop as a function of the time index k

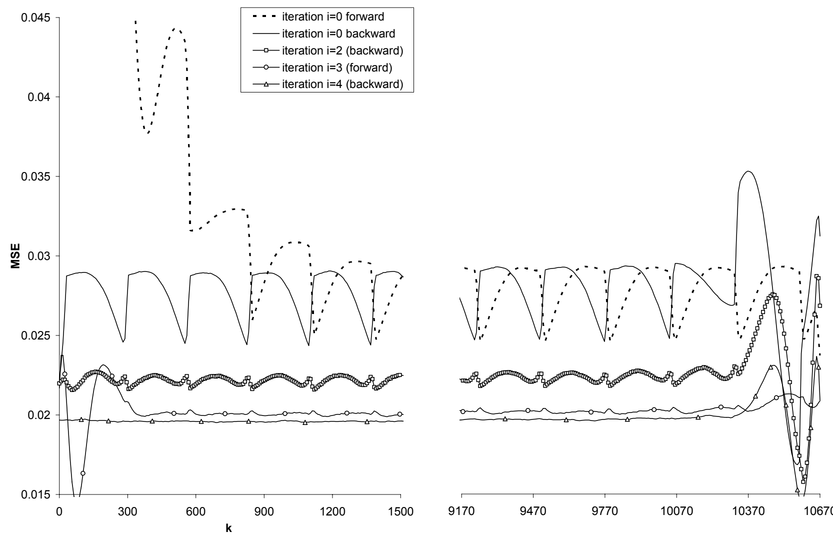

Figure 3. MSE obtained with a second-order bi-directional T-FB phase tracking loop as a function of the time index k

and (ii) the transition from a positive phase reference slope to a negative phase reference slope or vice versa. This effect can be circumvented by taking as final phase estimates the estimates at instants k = 0, ..., (L/2)-1 from the last forward iteration and those with indices k = L/2, ..., L-1 from the last backward iteration.

- The steady-state MSE of the T-FB algorithm is smaller than that of the non-code-aided FB algorithm used for initialization, and decreases as the number of iterations increases. This is because of the increased accuracy of the SDs. After convergence of the iterations, the MSE is very close to the steady state MSE for DA FB operation, i.e. N_0B_L/E_s=0.019, which illustrates the near-optimality of the proposed synchronizer.

In Figure 4, the performance of the bi-directional iterative T-FB synchronizer is assessed in terms of BER versus $\frac{\widetilde{E}_b}{N_0}=\frac{E_s}{N_0}\frac{1}{r}$.[2] A total of 10 synchronizer/decoding iterations are carried out. The phase noise is modeled as a discrete-time Wiener process $\Theta_k=\Theta_{k-1}+\vartheta_k$, where Θ_0 is a uniformly distributed random variable in [-π,π[, and { ϑ_k } are zero-mean

[2] Here, we make abstraction of the pilot symbol insertion loss of $-10\log\left(1+\frac{N_p+N_d}{N_d}\right)\cong 0.28$ dB. As a consequence, the gap between the BER of the perfectly synchronized system and the other BER curves is uniquely due to phase estimation/compensation inaccuracy.

statistically independent Gaussian increments with standard deviation σ_θ (typically about a few degrees). For the sake of comparison we also show in Figure 4 the BER for a perfectly synchronized system, and the BER as obtained with the code-aided FF joint phase and frequency estimator from [1] operating on the same signal and with identical initialization. In Figure 4, 'rms' stands for σ_θ (in degrees). We make the following observations:

- When the carrier phase is constant ((F, σ_θ) = (0, 0°)), the receiver with FF synchronization yields essentially the same BER as the perfectly synchronized receiver, and outperforms the receiver with FB synchronization by about 0.2 dB in the waterfall region of the turbo code. The BER degradation of the latter receiver can be made vanishingly small by reducing the loop bandwidth, because the carrier phase is time-invariant.

- The BER performance of the receiver with FF synchronization is considerably degraded in the presence of a moderately time-varying phase (such as (FT, σ_θ) = (5.10^{-5}, 0°) or (0, 0.36°)). As the phase estimate in the first iteration is constant over the frame, the time-variations of the actual phase give rise to unreliable SDs that cannot be improved in subsequent iterations.

- The receiver with FB synchronization can handle phase variations (such as (FT, σ_θ) = (10^{-3}, 0°) or (0, 1.08°)) that are too large for proper operation of the receiver with FF synchronization. As (FT, σ_θ) = (10^{-3}, 0°) and (FT, σ_θ) = (0, 0°) yield the same BER performance, it follows that the BER degradation as compared to perfect synchronization is caused by the additive-noise component of the phase error (frequency offset causes no steady-state phase error, because the PLL is type-II). A slightly larger degradation occurs for (FT, σ_θ) = (0, 1.08°), because a phase-noise component is added to the phase error.

IV. CONCLUSIONS

This contribution considers the estimation of a time-varying carrier phase in coded systems. We propose a new ML-based code-aided iterative FB type-II phase-tracker which optimally 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. As its derivation does not rely on the knowledge of the specific carrier phase statistics, the proposed algorithm exhibits a high robustness in the presence of a time-varying carrier phase. Simulation results show that our code-aided FB synchronizer can handle phase noise and frequency offsets with which its FF counterpart from [1,2] cannot cope.

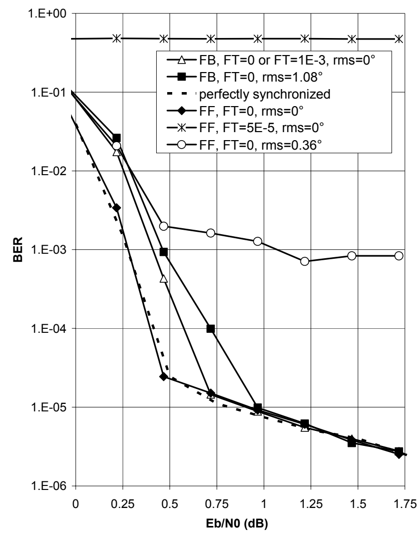

Figure 4. BER performance of turbo receiver with T-FF or T-FB carrier synchronization

REFERENCES

[1] Nele Noels, Heidi Steendam and Marc Moeneclaey, "On the Cramer-Rao lower Bound and the performance of synchronizers for (turbo) encoded systems," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless communications*, Lisbon, Portugal, July 11-14, 2004

[2] Nele Noels, Vincenzo Lottici, Antoine Dejonghe, Heidi Steendam, Marc Moeneclaey, Marco Luise and Luc Vandendorpe, "A Theoretical Framework for Soft Information Based Synchronization in Iterative (Turbo) Receivers," accpeted for publication in EURASIP *Journal on Wireless Communications and Networking*, special issue on Advanced Signal Processing Algorithms for Wireless Communications

[3] L. Lu and S. W. Wilson, "Synchronization of turbo-coded systems at low SNR," *in Proc. IEEE Int. Conf. Commun*., Atlanta, GA, 1998, pp. 428-432

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

[5] C. Langlais and M. Helard, "Phase carrier recovery for turbo codes over a sattelite link with the help of tentative deisions," *in Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, Sept. 2000, pp. 439-442

[6] A. Anastasopoulos and K.M.Chugg, " Adaptive iterative detection for phase tracking in turbo-coded systems," *IEEE Trans. Commun*., vol. 49, pp 2135-2144, Dec. 2001

[7] G. Colavolpe, A. Barbieri and G. Caire, "Iterative decoding in the presence of strong phase noise," submitted to *IEEE J. on Sel. Areas in Commun*., available at www. eurecom.fr/~caire

[8] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol IT-20, March 1974, pp. 248-287

[9] H. Meyr and G. Asheid, "*Synchronization In Digital Communications*," New York, Wiley, 1990