

# Iterative Carrier Phase Synchronization for Low-Density Parity-Check Coded Systems

Heidi Steendam, Nele Noels and Marc Moeneclaey  
DIGCOM research group, TELIN Dept., Ghent University  
Sint-Pietersnieuwstraat 41, 9000 GENT, BELGIUM  
E-mail : {hs,nnoels,mm}@telin.rug.ac.be

**Abstract**—In this paper, we consider the effect of a carrier phase offset on the performance of a low-density parity-check (LDPC) coded QAM modulated system. We investigate an ML-based carrier phase synchronization algorithm, that makes use of the posterior probabilities of the data symbols, provided by the iterative decoder. The resulting carrier phase synchronizer is an extension, to LDPC coded systems, of the iterative phase estimator for turbo coded systems, presented in [1], and has negligible BER performance degradation as compared to the ideally synchronized system.

## I. INTRODUCTION

Low-density parity-check (LDPC) codes were first introduced by Gallager as a class of linear error-correcting block codes, of which the check matrix is sparse. Gallager showed that random LDPC codes can perform close to the Shannon capacity limit when the block length increases [2]. However, as no practical decoding algorithm was available that could achieve the near-Shannon performance, the LDPC codes were forgotten. Only recently, with the discovery of turbo codes [3], the LDPC codes were rediscovered. A decoding algorithm for LDPC codes, similar to that for turbo codes, has been proposed [4]. In this decoding algorithm, we aim to compute the marginal posterior probability that a received bit is erroneous, given the information of the check matrix and the syndrome. This computation of the marginal posterior probability is done in an iterative way, by exchanging the belief that the information is correct [5]. With this iterative decoding algorithm, it has been shown in [6], [7], [8] that the Shannon limit can be approached.

However, the excellent performance of the LDPC codes was based on the assumption of ideal coherent detection, i.e. the carrier reference must be estimated accurately before data detection. This assumption may not be realistic in a system without pilot tone, as the system is operating at signal-to-noise ratios (SNR) that are extremely low. In this paper, we show that the iterative decoder is quite sensitive to carrier phase offsets, such that accurate carrier phase recovery is necessary to realize the excellent performance of the LDPC codes. To counteract the effect of the carrier phase offset on the performance of the iterative decoder, we propose a simple carrier phase synchronization algorithm, that is based on the ML approach. In this algorithm, the (iterative) estimation of the carrier phase is based on the soft decisions of the data provided by the iterative detector at each iteration. In this way, blind reliable synchronization is performed, and almost ideal coherent detection is obtained, even at very low SNR, as required in the LDPC coded system.

In section II, we briefly outline the LDPC coded system in the presence of a carrier phase offset. The effect of the carrier phase offset on the performance of the iterative decoder is discussed. The blind soft-decision directed carrier phase synchronizer is derived in section III, and we present the relevant numerical results in section IV. The conclusions are drawn in section V.

## II. THE LDPC CODED SYSTEM IN THE PRESENCE OF CARRIER PHASE OFFSETS

The conceptual block diagram of the LDPC coded system in the presence of a carrier phase offset is shown in figure 1. The bit sequence to be transmitted is first split into blocks  $\mathbf{b}$  of  $k$  bits, and fed to the  $(n, k)$  LDPC code, where  $n$  is the length of the code word. The linear transformation that converts the information word  $\mathbf{b}$  into the code word  $\mathbf{c}$  is characterized by the generator matrix  $\mathbf{G}$ . The corresponding check matrix  $\mathbf{H}$  is sparse, i.e. it contains a low density of ones. The resulting code word is Gray-mapped onto the QAM symbols  $\{a_k\}$  and transmitted over the channel. The channel adds white Gaussian noise  $\{w_k\}$ , with uncorrelated real and imaginary parts, each having a variance  $\sigma^2$ . Further, the transmitted symbols are affected by a carrier phase offset  $\theta$ . At the receiver, we obtain the sequence

$$r_k = a_k e^{j\theta} + w_k. \quad (1)$$

The symbols  $\{a_k\}$  have average energy per symbol equal to 1.

First, we consider the case that the carrier phase offset is not corrected at the receiver. The sequence (1) is applied to the decoder. Let us define the set of code bits  $\ell$  that participate in the  $m$ -th syndrome bit by  $\mathcal{L}(m) \equiv \{\ell : \mathbf{H}_{m,\ell} = 1\}$ , and the set of syndrome bits  $m$  in which the  $\ell$ -th code bit participates by  $\mathcal{M}(\ell) \equiv \{m : \mathbf{H}_{m,\ell} = 1\}$ , where  $\mathbf{H}$  denotes the check matrix. The decoder iteratively updates two quantities [4]:  $\Lambda q_{m,\ell} = \ln(q_{m,\ell}^1/q_{m,\ell}^0)$  and  $\Lambda r_{m,\ell} = \ln(r_{m,\ell}^1/r_{m,\ell}^0)$ , where  $q_{m,\ell}^x$  is the probability that the  $\ell$ -th code bit has the value  $x$ , given the information obtained from the syndromes other than the  $m$ -th syndrome. The quantity  $r_{m,\ell}^x$  is the probability of the  $m$ -th syndrome bit being zero, conditioned on the  $\ell$ -th code bit having a value  $x$ , and assuming that the other bits have a separable distribution given by the probabilities  $\{q_{m,\ell'} : \ell' \in \mathcal{L}(m) \setminus \ell\}$ . Further, the algorithm computes  $\Lambda q_\ell = \ln(q_\ell^1/q_\ell^0)$ , where  $q_\ell^x$  is the pseudoposterior probability that the code bit has value  $x$ . The iterative decoding algorithm is summarized in table I. The algorithm initializes the quantities  $q_{m,\ell}^x$  to the prior probability that the corresponding code bit  $\ell$  has value

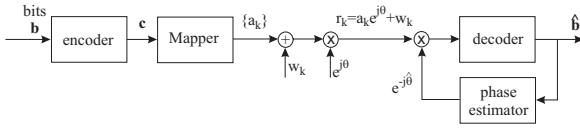


Fig. 1. Block diagram of LDPC coded system

TABLE I  
THE ITERATIVE DECODING ALGORITHM [3]

Initialization	
	initialize $\Lambda q_{m,\ell} = \Lambda p_\ell = \ln \left( \frac{p_\ell^1}{p_\ell^0} \right)$
Iteration	
Horizontal step	
	update $\Lambda r_{m,\ell} = \ln \left( \frac{1 - \prod_{\ell' \in \mathcal{L}(m) \setminus \ell} -\tanh \frac{\Lambda q_{m,\ell'}}{2}}{1 + \prod_{\ell' \in \mathcal{L}(m) \setminus \ell} -\tanh \frac{\Lambda q_{m,\ell'}}{2}} \right)$
Decoding	
	compute $\Lambda q_\ell = \Lambda p_\ell + \sum_{m \in \mathcal{M}(\ell)} \Lambda r_{m,\ell}$
	select $\hat{c}_\ell = \begin{cases} 0 & \Lambda q_\ell < 0, \\ 1 & \Lambda q_\ell \geq 0. \end{cases}$
Vertical step	
	update $\Lambda q_{m,\ell} = \Lambda p_\ell + \sum_{m' \in \mathcal{M}(\ell) \setminus m} \Lambda r_{m',\ell}$

$x$ , i.e.  $p_\ell^x$ . For the considered system, the prior probability that the corresponding code bit  $\ell$  equals 1, is given by

$$p_\ell^1 = \frac{\sum_{a_k: c_\ell=1} e^{-\frac{1}{2\sigma^2}|r_k - a_k|^2}}{\sum_{\forall a_k} e^{-\frac{1}{2\sigma^2}|r_k - a_k|^2}}, \quad (2)$$

and  $p_\ell^0 = 1 - p_\ell^1$ , where  $\sigma^{-2} = 2E_s/N_0$ . After each iteration, an updated version  $\hat{\mathbf{c}}$  of the code word is computed, based on the pseudoposterior probabilities, and the syndrome is computed to check whether  $\hat{\mathbf{c}}$  is a valid code word. The algorithm stops if a valid code word is found, or if the maximum allowed number of iterations is reached, in which case the algorithm reports that the decoding was not successful.

We evaluate the performance of the iterative decoder by means of simulations, as an analytical approach is too complex. We consider the case of a rate  $R = 1/2$  LDPC encoder with block length  $n = 1000$ . The maximum number of iteration steps is set to 20. A 4-QAM modulated system is considered. Figure 2 shows the bit error rate (BER) as function of  $E_b/N_0$ , where  $E_b$  denotes the energy transmitted per information bit, for different values of  $\theta$ . In figure 3, the BER is shown as function of the angle  $\theta$ , for different values of  $E_b/N_0$ . As we observe, the presence of a carrier phase offset causes severe performance degradations. Hence, in order to achieve a performance that is close to the performance of the case of ideal coherent detection, the carrier phase offset must be accurately estimated and compensated for.

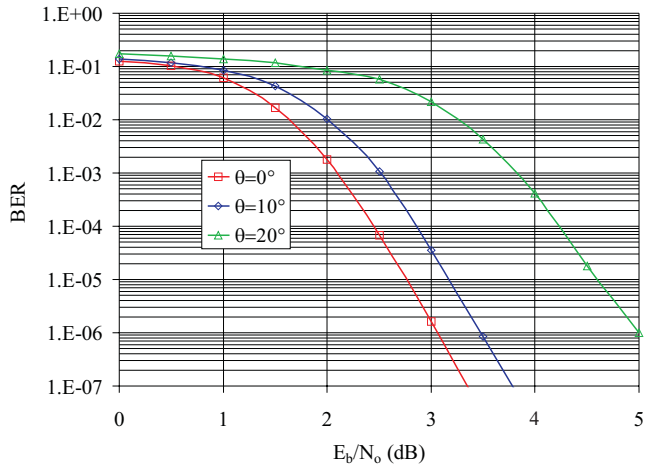


Fig. 2. BER performance of the iterative decoder versus  $E_b/N_0$

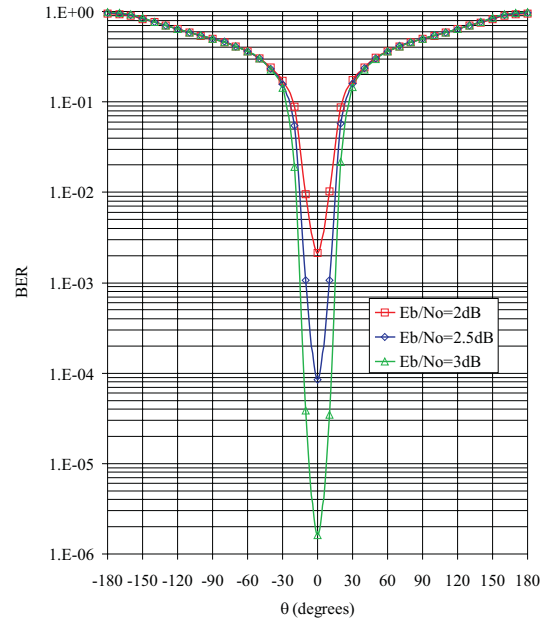


Fig. 3. BER performance of the iterative decoder versus carrier phase offset

### III. THE CARRIER PHASE SYNCHRONIZATION ALGORITHM

The carrier phase estimator to be proposed in this section iteratively maximizes the log-likelihood function  $p(\mathbf{r}|\theta)$ , by exploiting the posterior probabilities of the coded bits, that are provided by the iterative LDPC decoder. The resulting synchronizer can be viewed as an extension to LDPC codes of the synchronizer for turbo-coded systems from [1]. Whereas the latter synchronizer has been introduced as an ad hoc algorithm, it has been shown in [9] that its operation is mathematically equivalent to the expectation-maximization (EM) algorithm [10], [11], [12] for iteratively maximizing the log-likelihood function. In the following, we adopt the line of thought from

[9].

The received vector  $\mathbf{r}$  (1) depends of the data symbols  $\mathbf{a}$  with prior probability  $P(\mathbf{a})$ , the noise vector  $\mathbf{w}$  and the carrier phase offset  $\theta$ , which has to be estimated. The ML estimate  $\hat{\theta}$  of  $\theta$  is obtained as the solution of

$$\hat{\theta} = \arg \max_{\theta} \ln p(\mathbf{r}|\theta), \quad (3)$$

where

$$p(\mathbf{r}|\theta) = \int_{\mathbf{a}} P(\mathbf{a}) p(\mathbf{r}|\theta, \mathbf{a}) d\mathbf{a} \quad (4)$$

is the likelihood function, after averaging over the data symbols  $\mathbf{a}$ . In [9], it is shown that the solution of (3) satisfies

$$\int_{\mathbf{a}} P(\mathbf{a}|\mathbf{r}, \theta) \frac{\partial}{\partial \theta} \ln p(\mathbf{r}|\theta, \mathbf{a}) d\mathbf{a} \Big|_{\theta=\hat{\theta}} = 0. \quad (5)$$

As the carrier phase appears in both term of the integrand, the solution of (5) is hard to find. Therefore, we try to solve (5) iteratively. In the proposed algorithm, a sequence of estimates  $\hat{\theta}^{(k)}$  is produced by solving

$$\int_{\mathbf{a}} P(\mathbf{a}|\mathbf{r}, \theta^{(k-1)}) \frac{\partial}{\partial \theta} \ln p(\mathbf{r}|\theta, \mathbf{a}) d\mathbf{a} \Big|_{\theta=\hat{\theta}^{(k)}} = 0, \quad (6)$$

where the previous estimate  $\hat{\theta}^{(k-1)}$  is used to resolve the conditioning on  $\theta$  in the first factor of the integrand. This algorithm turns out to be equivalent with the EM algorithm, which converges to the ML estimate [10], [11], [12].

Considering (1), the vector  $\mathbf{r}$  conditioned on  $\mathbf{a}$  and  $\theta$  has a Gaussian distribution with average  $\mathbf{a}e^{j\theta}$  and covariance matrix  $\sigma^2 I$ , where  $I$  denotes the identity matrix. Neglecting the irrelevant terms independent of  $\mathbf{a}$  and  $\theta$ , we obtain

$$\ln p(\mathbf{r}|\theta, \mathbf{a}) = -\frac{1}{2\sigma^2} \sum_{i=1}^N |a_i e^{j\theta} - r_i|^2, \quad (7)$$

where  $N$  is the number of complex data symbols that correspond to a code word  $\mathbf{c}$  (i.e.  $N = \lceil n/\log_2 M \rceil$ , where  $M$  is the constellation size). Substituting (7) in (6), we obtain the estimate

$$\hat{\theta}^{(k)} = \arg(\mathbf{r}^T \tilde{\mathbf{a}}), \quad (8)$$

where

$$\tilde{\mathbf{a}} = \int_{\mathbf{a}} \mathbf{a} P(\mathbf{a}|\mathbf{r}, \theta^{(k-1)}) d\mathbf{a} \quad (9)$$

is the posterior average of the vector  $\mathbf{a}$  of the complex data symbols. The data symbols  $\mathbf{a}$  are related in a deterministic way to the code bits  $\mathbf{c}$ . Hence, obtaining the estimate (8) requires the knowledge of the joint statistics of the code bits, i.e.  $P(\mathbf{c}|\mathbf{r}, \theta)$ . An estimate of the joint statistics of the code bits can be obtained from the iterative decoding algorithm:

$$P(c_\ell = 1|\mathbf{r}, \theta) = \frac{e^{\Lambda_{q\ell}}}{1 + e^{\Lambda_{q\ell}}}, \quad (10)$$

and  $P(c_\ell = 0|\mathbf{r}, \theta) = 1 - P(c_\ell = 1|\mathbf{r}, \theta)$ . Denote  $\mathbf{c}_m = (c_{m,1}, \dots, c_{m,\log_2 M})$ ,  $m = 1, \dots, N$  as the set of code bits

TABLE II  
THE JOINT ITERATIVE CARRIER PHASE SYNCHRONIZATION AND  
DECODING ALGORITHM

Initialization	initialize $\Lambda_{q_{m,\ell}}$
Iteration	Horizontal step update $\Lambda_{r_{m,\ell}}$
Decoding	compute $\Lambda_{q_\ell}$ select $\hat{c}_\ell = \begin{cases} 0 & \Lambda_{q_\ell} < 0, \\ 1 & \Lambda_{q_\ell} \geq 0. \end{cases}$
Phase estimation	compute $\hat{\theta}$ using (8) update $\Lambda_{p_\ell}$
Vertical step	update $\Lambda_{q_{m,\ell}}$

that contribute to the data symbol  $a_m$ . The data symbol  $a_m$  can take  $M$  different values with probability

$$P(a_m|\mathbf{r}, \theta) = \prod_{\substack{i=1 \\ \mathbf{c}_m: a_m}}^{\log_2 M} P(c_{m,i}|\mathbf{r}, \theta), \quad (11)$$

where the set  $\mathbf{c}_m$  is mapped on the symbol  $a_m$  and  $P(c_{m,i}|\mathbf{r}, \theta)$  is given by (10). Note that in (11) it is tacitly assumed that the coded bits are statistically independent. From (11), the posterior average  $\tilde{\mathbf{a}}$  of  $\mathbf{a}$  can be written as

$$\tilde{\mathbf{a}}_m = \sum_{\ell=1}^M a_{m,\ell} \prod_{\substack{i=1 \\ \mathbf{c}_m: a_m}}^{\log_2 M} P(c_{m,i}|\mathbf{r}, \theta). \quad (12)$$

For example, in the 4-QAM case, with the even bits mapped on the real part of the data symbol and the odd bits on the imaginary part of the symbol, the posterior average  $\tilde{\mathbf{a}}$  is given by

$$\tilde{\mathbf{a}}_m = \frac{\sqrt{2}}{2} \left( \text{atan} \frac{\Lambda_{q_{2m}}}{2} + j \text{atan} \frac{\Lambda_{q_{2m+1}}}{2} \right). \quad (13)$$

The phase estimate (8) is used to rotate the received samples over the angle  $-\hat{\theta}^{(k)}$ , resulting in the sequence  $\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} e^{-j\hat{\theta}^{(k)}}$ . An updated version of the prior probabilities  $\Lambda_{p_\ell}$  is computed based on the rotated sequence  $\mathbf{r}^{(k)}$ , and is used to update the quantities  $\Lambda_{q_{m,\ell}}$ . Hence, in each iteration step, soft information is exchanged between the synchronizer and the decoder. After a number of iterations, the estimate of the carrier phase and the posterior probability of  $\mathbf{c}$  converge to their "true" value. The joint carrier phase synchronization and decoding algorithm is summarized in table II.

#### IV. SIMULATION RESULTS

To evaluate the performance of the proposed joint carrier phase synchronization and decoding algorithm, simulations have been carried out with the same LDPC code with rate  $R = 1/2$  and block length  $n = 1000$  as in section II. We study the mean  $E[\hat{\theta}]$  of the estimate  $\hat{\theta}$ , the BER and the root-mean

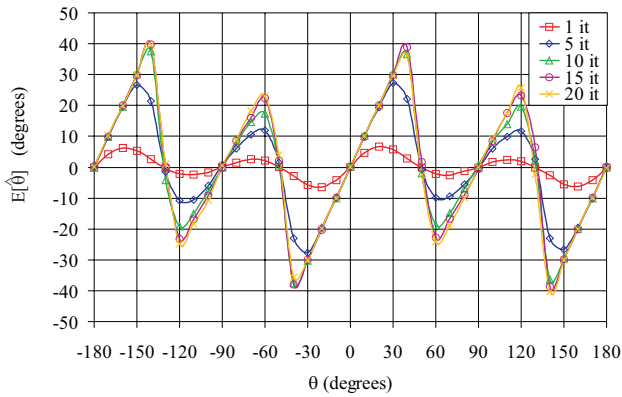


Fig. 4. Mean estimated  $\hat{\theta}$  for different number of iterations,  $E_b/N_0 = 2$  dB, 4-QAM

squared error (RMSE), in terms of the number of iteration steps and  $E_b/N_0$ . In the following figures, results are shown for 4-QAM. However, similar results have been obtained for 16-QAM.

In figures 4 and 5,  $E[\hat{\theta}]$  is shown as function of the carrier phase offset  $\theta$  for different number of iteration steps, at  $E_b/N_0 = 2$  dB, corresponding roughly to a BER= $10^{-3}$  in case of perfect synchronization, and for different values of  $E_b/N_0$ , for 20 iteration steps. As we observe, the proposed algorithm is not able to avoid the four fold ambiguity caused by the rotational symmetry of the QAM constellation. Further, the correspondence between the mean estimated  $\hat{\theta}$  and the carrier phase offset  $\theta$  increases with increasing number of iterations and increasing  $E_b/N_0$ , for carrier phase offsets in the range  $[-45^\circ, 45^\circ]$ , i.e. after correction of the carrier phase offset, only a small residual phase offset is present. Hence, the proposed algorithm provides good estimates for carrier phase offsets up to  $30^\circ - 40^\circ$ . To overcome the four-fold ambiguity of the algorithm, the input of the decoder must be compensated by a coarse estimate (e.g., resulting from a few pilot symbols) of the initial phase offset, such that the residual phase offset is within the tracking range of the proposed algorithm in the starting-up phase of the synchronizer. Alternatively, an ambiguous phase estimate can be detected from a nonzero syndrome, after which a multiple of  $\pi/2$  is added to the estimate until a zero syndrome is obtained. Further, the phase estimate of the previous code block can be used as a finer estimate of the phase offset to precorrect all signal samples of the next code block, in order to keep the residual phase offset in the vicinity of the origin. In this way, better estimates of the carrier phase offsets can be obtained by the algorithm.

In figure 6, the BER, resulting from the combination of synchronizer and decoder, is shown as function of the carrier phase offset  $\theta$  for different values of  $E_b/N_0$ , assuming 20 iteration steps. As we observe, the BER is essentially independent of the carrier phase offset and equal to the BER of the perfectly synchronized system, for carrier phase offsets in the range of  $[-30^\circ, 30^\circ]$ . Further, the BER is shown in figure

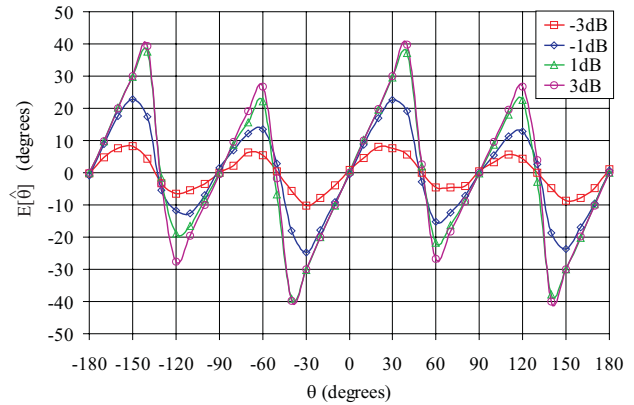


Fig. 5. Mean estimated  $\hat{\theta}$  for different values of  $E_b/N_0$ , 20 iterations, 4-QAM

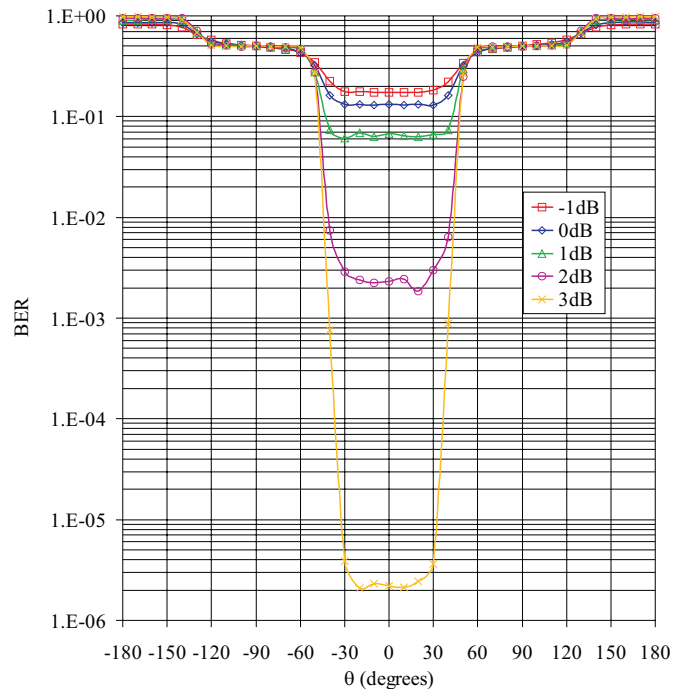


Fig. 6. BER performance for different values of  $E_b/N_0$ , 20 iterations, 4-QAM

7 as function of  $E_b/N_0$ , for different values of the number of iteration steps, and a carrier phase offset  $\theta = 20^\circ$ . We observe that the performance degradation is negligibly small as compared to the perfectly synchronized system for 20 iteration steps. These results confirm that the proposed algorithm can perform joint carrier phase synchronization and decoding with negligible BER performance degradation.

Finally, in figure 8, the RMSE is shown as function of  $E_b/N_0$  for different values of the number of iteration steps. For increasing  $E_b/N_0$ , the RMSE decreases; for given  $E_b/N_0$ , the RMSE decreases with increasing number of iteration steps. At high  $E_b/N_0$  (for 20 iteration steps,  $E_b/N_0 \geq 1.5$  dB),

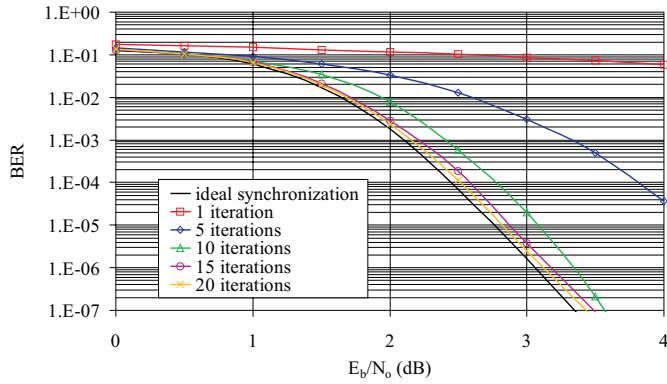


Fig. 7. BER performance versus  $E_b/N_0$  for different number of iterations,  $\theta = 20^\circ$ , 4-QAM

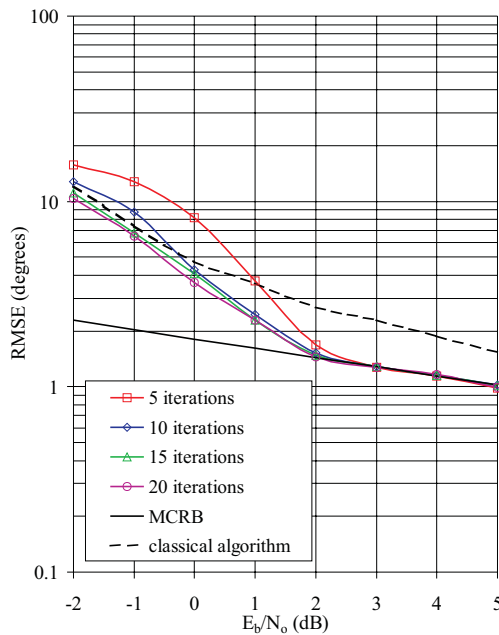


Fig. 8. RMSE versus  $E_b/N_0$ ,  $\theta = 20^\circ$ , 4-QAM

the curves essentially coincide with the modified Cramer-Rao lower bound (MCRB). This can be explained by observing that for high  $E_b/N_0$ , the posterior averages of the transmitted data symbols are close to the true symbols, so the performance of the algorithm reaches the performance of the data-aided carrier phase synchronizer. Further, the RMSE for the classical carrier phase estimator, corresponding to the estimate

$$\hat{\theta} = \frac{1}{4} \arg \left( \sum_{k=0}^{L-1} r_k^4 \right), \quad (14)$$

is shown, where  $L$  denotes the number of data symbols used to estimate the carrier phase. As observed, the RMSE for the classical phase estimator is much higher than the RMSE for the proposed algorithm.

## V. CONCLUSIONS

In this paper, we have considered the effect of a carrier phase offset on the performance of an LDPC coded QAM modulated system. We show that the performance of the LDPC coded system is severely degraded when the carrier phase offset is not compensated. To overcome the problem of the degradation caused by a carrier phase offset, we propose a simple blind soft-decision directed carrier phase synchronization algorithm, that is based on the ML approach. The algorithm iteratively exchanges soft information between the synchronizer and the decoder. The proposed algorithm introduces negligibly small performance degradations as compared to the ideally synchronized system for carrier phase offsets up to  $30^\circ$ . Larger carrier phase offsets can not be estimated effectively, because of the rotational symmetry of the QAM constellation. To counter this problem, we propose to compensate the input of the decoder with a coarse estimate of the initial phase offset when starting up the synchronizer, and to use the phase estimate of the previous code block to obtain a finer estimate of the carrier phase offset, which can be used to compensate the signal samples of the next code block. In this way, the residual phase offset is kept close to the origin, and better estimates for the phase are obtained.

## ACKNOWLEDGMENT

This work has been supported by the Interuniversity Attraction Poles Program - Belgian State - Federal Office for Scientific, Technical and Cultural Affairs.

## REFERENCES

- [1] V. Lottici, M. Luise, "Carrier Phase Recovery for Turbo-Coded Linear Modulations", *Proc. 2002 IEEE Int. Conference on Communications, ICC*, New York, NY, 2002
- [2] R.G. Gallager, "Low-Density Parity-Check Codes", *IRE Trans. Info. Theory*, IT-8, Jan 1962, pp. 21-28
- [3] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", *Proc. 1993 IEEE Int. Conference on Communications, ICC*, Geneva, Switzerland, 1993, pp. 1064-1070
- [4] D.J.C. MacKay, "Good Error Correcting Codes Based on Very Sparse Matrices", *IEEE Trans. on Information Theory*, Vol 45, no 2, 1999, pp. 399-431
- [5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufman Publishers, San Francisco, 1988
- [6] D.J.C. MacKay, R.M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes", *Electronics Letters*, Vol 32, no 18, Aug 1996, pp. 1645-1646
- [7] S.Y. Chung, G.D. Forney, T.J. Richardson, R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045dB of the Shannon Limit", *IEEE Communications Letters*, Vol 5, Feb 2001, pp. 58-60
- [8] T.J. Richardson, A. Shokrollahi, R. Urbanke, "Design of Capacity-Approaching Low-Density Parity-Check Codes", *IEEE Trans. on Information Theory*, Vol 47, Feb 2001, pp. 617-637
- [9] N. Noels et Al., "Blind ML Synchronization and Channel Gain Estimation for Iterative (Turbo) Receivers", submitted to *IEEE Trans. on Communications*
- [10] A. Dempster, N. Laird, D. Rubin, "Maximum-Likelihood from Incomplete Data Via the EM Algorithm", *J. Royal Statistics Soc., Ser. B*, vol. 39, no 1, Jan. 1977, pp. 1-38
- [11] T. Moon, "The Expectation-Maximization Algorithm", *IEEE Signal Proc. Mag.*, vol. 13, no 11, Nov. 1996, pp. 47-60
- [12] C. Georgiades, J.C. Han, "Sequence Estimation in the Presence of Random Parameters Via the EM Algorithm", *IEEE Trans. on Communications*, vol. 45, no 3, Mar. 1997, pp. 300-308