

# Phase Estimation by Message Passing

Justin Dauwels and Hans-Andrea Loeliger

Dept. of Information Technology and Electrical Engineering, ETH Zürich

CH-8092 Zürich, Switzerland

Email: {dauwels, loeliger}@isi.ee.ethz.ch

**Abstract**—The problem of phase estimation in a “turbo receiver” is considered for two different channel models. Several message passing algorithms for phase estimation are derived from the factor graph of the channel models: (1) straight sum-product, applied to a quantized phase model; (2) LMS-type gradient methods; (3) a particle filter. All considered algorithms are suitable for use in a “turbo receiver” with joint iterative decoding and phase estimation.

## I. INTRODUCTION

We consider algorithms for phase estimation in a communications receiver, with special emphasis on “turbo receivers” using joint iterative phase estimation and decoding. Coded channel input symbols are transmitted in frames of  $L$  symbols. We consider channels of the form

$$Y_k = X_k e^{j\Theta_k} + N_k, \quad (1)$$

where  $X_k$  is the channel input symbol at time  $k \in \{1, 2, \dots, L\}$ ,  $Y_k$  is the corresponding received symbol,  $\Theta_k$  is the unknown phase, and  $N_k$  is complex white Gaussian noise with (known) variance  $2\sigma_N^2$ , i.e.,  $\sigma_N^2$  per dimension. For the sake of definiteness, we assume that the channel input symbols  $X_k$  are M-PSK symbols and are protected by a low-density parity check (LDPC) code.

We will consider two different models for the evolution of the phase  $\Theta_k$ :

**Constant Phase:**  $\Theta_k = \Theta$ , an unknown constant.

**Random Walk:**

$$\Theta_k = (\Theta_{k-1} + W_k) \bmod 2\pi, \quad (2)$$

where  $W_k$  is white Gaussian noise with known variance  $\sigma_W^2$ .

The extension to more sophisticated phase noise models (e.g., [1]–[4]), with or without additional frequency offset can also be handled, but is not treated in this paper.

Several turbo-synchronization algorithms that can deal with constant phase rotations [5]–[10] have been proposed recently. A turbo-synchronization algorithm for a stochastic phase model was presented in [11]. However, no such algorithm seems to have been proposed for the random-walk phase model.

The algorithms we propose are approximations of the symbol-wise MAP (maximum a posteriori) decoder:

$$\hat{X}_{k,\text{MAP}} = \underset{x_k}{\operatorname{argmax}} \int_0^{2\pi} \dots \int_0^{2\pi} p(x_k, \mathbf{y}, \boldsymbol{\theta}) d\boldsymbol{\theta} \quad (3)$$

$$= \underset{x_k}{\operatorname{argmax}} \int_0^{2\pi} \dots \int_0^{2\pi} \sum_{\mathbf{x} \text{ with } x_k \text{ fixed}} p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) d\boldsymbol{\theta}, \quad (4)$$

where  $\mathbf{X} \triangleq (X_1, X_2, \dots, X_L)$ ,  $\mathbf{Y} \triangleq (Y_1, Y_2, \dots, Y_L)$  and  $\boldsymbol{\Theta} \triangleq (\Theta_1, \Theta_2, \dots, \Theta_L)$ . The function  $p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$  stands for the joint probability function of  $\mathbf{X}$ ,  $\mathbf{Y}$ , and  $\boldsymbol{\Theta}$ ; it is a probability density function (pdf) in  $\boldsymbol{\theta}$  and  $\mathbf{y}$  and a probability mass function (pmf) in  $\mathbf{x}$ . Note that Equations (3) and (4) involve averaging over the phase  $\boldsymbol{\Theta}$ . Message passing algorithms to approximately compute (3) may be obtained by the following procedure:

- 1) The probability function  $p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})$  is represented by a factor graph [12] [13].
- 2) Message types are chosen and message update rules are computed.
- 3) A message update schedule is chosen.

In Step 2, finite-alphabet variables (such as  $X_k$ ) are handled by the standard sum-product rule [12] [13]. For continuous-valued variables (such as  $\Theta_k$ ) however, the sum-product rule leads to intractable integrals, which can be approximated in several ways; each such approximation corresponds to a certain message type and results in a different algorithm. In some cases, we will obtain an approximation of the entire posterior distribution  $p(\boldsymbol{\theta} | \mathbf{y})$ ; in other cases, we will obtain only an estimate  $\hat{\boldsymbol{\Theta}}$ .

This paper is structured as follows. In Section II, we explain the factor graphs. The various message passing algorithms are described in Section III. Simulation results are presented in Section IV.

## II. FACTOR GRAPHS OF THE SYSTEM

We use Forney-style factor graphs (FFG), where nodes (boxes) represent factors and edges represent variables. A tutorial introduction to such graphs is given in [12].

The system described in Section I is easily translated into the FFG of Figure 1, which represents the factorization of the joint probability function of all variables. The upper part of the graph is the indicator function of the LDPC code, with parity check nodes in the top row that are “randomly” connected to equality constraint nodes (“bit nodes”). The function  $f$  is the deterministic mapping  $f : (B_k^{(1)}, \dots, B_k^{(\log_2 M)}) \mapsto X_k$

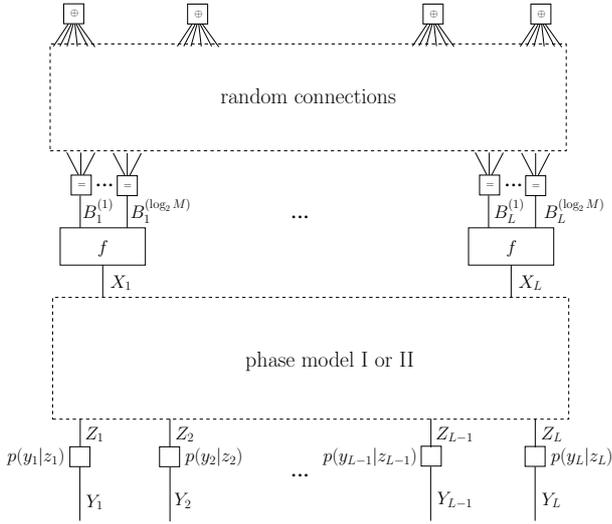


Fig. 1. FFG of LDPC code and the channel model.

of the bits  $B_k^{(1)}, \dots, B_k^{(\log_2 M)}$  to the symbol  $X_k$ . The nodes labelled “ $f$ ” (“bit mapper nodes”) correspond to the factors

$$\delta_f \left( b_k^{(1)}, \dots, b_k^{(\log_2 M)}, x_k \right) \quad (5)$$

$$\triangleq \begin{cases} 1, & \text{if } f \left( b_k^{(1)}, \dots, b_k^{(\log_2 M)} \right) = x_k; \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The bottom row of the graph represents the factors  $p(y_k|z_k) \triangleq (2\pi\sigma_N^2)^{-1} e^{-\|y_k - z_k\|^2 / 2\sigma_N^2}$ .

The “phase model” in Figure 1 is detailed in Figures 2 and 3. In these figures,  $S_k$  is defined as  $S_k \triangleq e^{j\Theta_k}$  and  $Z_k$  is defined as  $Z_k \triangleq X_k S_k$ . The top row of nodes (“multiply nodes”) in Figures 2 and 3 represents the factors  $\delta(z_k - x_k S_k)$ . The function  $g$  is the deterministic mapping  $g : \Theta_k \mapsto S_k$  of the phase  $\Theta_k$  to  $S_k$ ; the nodes labelled “ $g$ ” in Figures 2 and 3 represent the factors  $\delta(s_k - e^{j\theta_k})$ . The equality constraint nodes inside the dotted box in Figure 2 impose the constraint  $\Theta_k = \Theta$ ,  $\forall k$ ; the box can be viewed as a compound equality constraint node. In Figure 3, the nodes labelled  $p(\theta_k|\theta_{k-1})$  (“phase noise nodes”) represent the factors  $p(\theta_k|\theta_{k-1}) \triangleq (2\pi\sigma_W^2)^{-1/2} \sum_{n \in \mathbb{Z}} e^{-((\theta_k - \theta_{k-1}) + n2\pi)^2 / 2\sigma_W^2}$ .

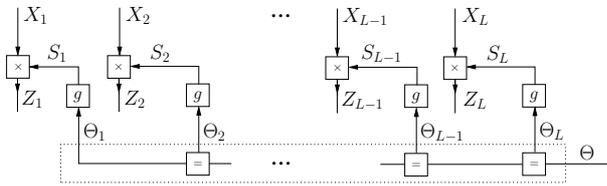


Fig. 2. FFG of the constant-phase model.

### III. MESSAGE-PASSING ALGORITHMS

#### A. Sum-Product Message Update Rules

We now apply the sum-product algorithm [12] to the FFG in Figure 1, where the FFG of the phase model is detailed in

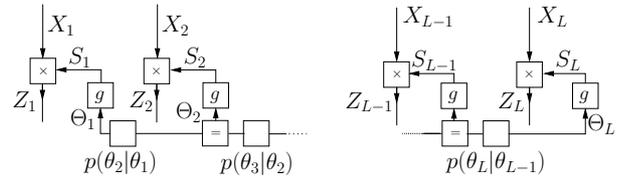


Fig. 3. FFG of the random-walk phase model.

Figures 2 and 3. The message passing schedule is as follows. First, the bottom row in Figure 1 is updated, i.e., messages are sent from the received symbols  $Y_k$  towards the phase model. Then, we alternate between

- 1) A sequence of horizontal (left to right and left to right) sweeps through the graph of the phase model (Figures 2 or 3).
- 2) Several iterations of the LDPC decoder.

Additionally, one can iterate between the LDPC decoder and the bit mapper nodes  $f$ .

The computation of the messages out of the bit mapper nodes and inside the graph of the LDPC code is standard [12] [13] [14]; we therefore only consider the computation of the messages inside, and out of, the graph of the two phase models.

Straightforward application of the sum-product algorithm [12] [13] to the graph of the two phase models results in the following update rules:

- **Equality Constraint Node** (see Figure 4(a);  $\zeta, \chi$  and  $\xi \in [0, 2\pi)$ )

$$\mu_{\square \rightarrow \xi}(\xi) = \gamma \mu_{\zeta \rightarrow \square}(\xi) \mu_{\chi \rightarrow \square}(\xi) \quad (7)$$

- **Multiply Node** (see Figure 4(b);  $\xi$  takes values on the unit circle,  $\chi$  is an M-PSK symbol and  $\zeta \in \mathbb{C}$ )

$$\mu_{\square \rightarrow \xi}(\xi) = \gamma \sum_{\chi} \mu_{\chi \rightarrow \square}(\chi) \mu_{\zeta \rightarrow \square}(\chi \xi) \quad (8)$$

$$\mu_{\square \rightarrow \chi}(\chi) = \gamma \oint_{\text{unit circle}} \mu_{\xi \rightarrow \square}(\xi) \mu_{\zeta \rightarrow \square}(\chi \xi) d\xi \quad (9)$$

- **Phase Noise Node** (see Figure 4(c);  $\chi$  and  $\xi \in [0, 2\pi)$ )

$$\mu_{p \rightarrow \xi}(\xi) = \gamma \int_0^{2\pi} \mu_{\chi \rightarrow p}(\chi) p(\xi|\chi) d\chi \quad (10)$$

- **Phase Mapper Node** (see Figure 4(d);  $\chi \in [0, 2\pi)$  and  $\xi$  takes values on the unit circle)

$$\mu_{g \rightarrow \chi}(\chi) = \mu_{\xi \rightarrow g}(e^{j\chi}) \quad (11)$$

$$\mu_{g \rightarrow \xi}(\xi) = \mu_{\chi \rightarrow g}(\arg \xi), \quad (12)$$

where  $\arg \xi$  stands for the argument of  $\xi$ .

In Equations (7)–(10),  $\gamma$  is a normalization factor. In Figures 2 and 3, the multiply nodes are connected to the phase mapper nodes by  $S_k$  edges; as a consequence, by combining (9) and (12), one can rewrite the update rule for the messages

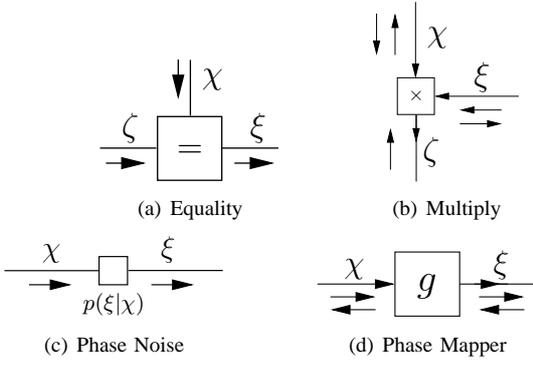


Fig. 4. Nodes in the graph of the phase models.

$\mu_{\boxtimes \rightarrow X_k}(x_k)$  out of the multiply nodes along the  $X_k$  edges as

$$\mu_{\boxtimes \rightarrow X_k}(x_k) = \gamma \int_0^{2\pi} \mu_{\Theta_k \rightarrow g}(\theta_k) \mu_{Z_k \rightarrow \boxtimes}(x_k e^{j\theta_k}) d\theta_k, \quad (13)$$

which is more practical to implement than (9). Similarly, one can rewrite the update rule for the messages  $\mu_{g \rightarrow \Theta_k}(\theta_k)$  out of the phase mapper nodes along the  $\Theta_k$  edges as

$$\mu_{g \rightarrow \Theta_k}(\theta_k) = \gamma \sum_{x_k} \mu_{X_k \rightarrow \boxtimes}(x_k) \mu_{Z_k \rightarrow \boxtimes}(x_k e^{j\theta_k}). \quad (14)$$

Note that according to the sum-product rule, the messages  $\mu_{\Theta_k \rightarrow g}(\theta_k)$  in Figure 2 are computed as

$$\mu_{\Theta_k \rightarrow g}(\theta_k) = \prod_{i \neq k} \mu_{g \rightarrow \Theta_i}(\theta_k), \quad (15)$$

i.e., the product of all messages arriving at the equality constraint node, except the one that arrives along the edge  $\Theta_k$ . However, it is more convenient to approximate the messages  $\mu_{\Theta_k \rightarrow g}(\theta_k)$  by the product of *all* messages arriving at the equality constraint node:

$$\mu_{\Theta_k \rightarrow g}(\theta_k) \approx \prod_{i=1}^L \mu_{g \rightarrow \Theta_i}(\theta_k) \quad (16)$$

$$= \mu_{\square \rightarrow \Theta}(\theta_k). \quad (17)$$

This leads in practice to the same result, since  $L$ , the number of messages in the product (15) and (16), is typically large (between 100 and 10.000).

We apply several methods to approximate the integrals in Equations (10) and (13): numerical integration, gradient methods and particle filtering. Each approximation leads to a different message passing algorithm for phase estimation.

### B. Numerical Integration

In this approach, we replace the integrals in Equations (10) and (13) by finite sums. Using the rectangular integration rule, the integral in the right hand side of Equation (13) for example is replaced by the sum  $\sum_{m=0}^{n-1} \mu_{\Theta_k \rightarrow g}(2\pi m/n) \mu_{Z_k \rightarrow \boxtimes}(x_k e^{j2\pi m/n})$ . Accordingly, the messages are represented by piecewise constant functions. This approach is equivalent to straightforwardly

applying the sum-product algorithm to a quantized phase model. Other integration schemes could also be used, such as the trapezoidal rule, which corresponds to representing the messages as piecewise linear functions.

### C. Gradient Methods

It is common to represent a message  $\mu(x)$  by the value  $\hat{X} \triangleq \operatorname{argmax}_x[\mu(x)]$ . The message  $\mu(x)$  is then approximated by the Dirac delta  $\delta(x - \hat{x})$ . In many cases, the expression  $\hat{X} \triangleq \operatorname{argmax}_x[\mu(x)]$  cannot be evaluated analytically. However, the value  $\hat{X}$  can often be determined numerically by means of search methods such as for example gradient methods. The latter methods are only applicable if the required derivatives can be evaluated; least mean square (LMS) gradient methods are based on first-order derivatives; techniques such as (scaled) conjugate gradient (SCG) require the (exact or approximative) computation of second-order derivatives.

We have developed LMS-style gradient-based algorithms for phase estimation. The gradient method for the constant-phase model searches the  $\operatorname{argmax}$  of the function  $\mu_{\square \rightarrow \Theta}(\theta)$  by performing the following steps:

- 1) Choose an initial phase estimate, which may be obtained from a non-data aided algorithm as for example the M-law [15].
- 2) The compound equality constraint node broadcasts the current phase estimate  $\hat{\Theta}^{\text{old}}$  to all multiply nodes  $\left. \frac{d \log \mu_{g \rightarrow \Theta_k}(\theta)}{d\theta} \right|_{\theta=\hat{\theta}^{\text{old}}}$ .
- 3) At the compound equality constraint node, a new phase estimate is computed according to the rule

$$\hat{\theta}^{\text{new}} = \hat{\theta}^{\text{old}} + \lambda \left. \frac{d \log \mu_{\square \rightarrow \Theta}(\theta)}{d\theta} \right|_{\theta=\hat{\theta}^{\text{old}}} \quad (18)$$

$$= \hat{\theta}^{\text{old}} + \lambda \sum_{k=1}^L \left. \frac{d \log \mu_{g \rightarrow \Theta_k}(\theta)}{d\theta} \right|_{\theta=\hat{\theta}^{\text{old}}}. \quad (19)$$

- 4) Iterate the steps 2 to 4 as often as desired.

The step size parameter  $\lambda$  in Equations (18) and (19) may be decreased as the number of iterations grows.

For the random-walk phase model, one alternates between forward and backward sweeps in the graph of the phase model. The forward and backward sweep result in the phase estimates  $\hat{\Theta}_k^{\text{FW}}$  and  $\hat{\Theta}_k^{\text{BW}}$  respectively. In the forward sweep, the following steps are performed sequentially:

- 1) Choose an initial phase estimate  $\hat{\Theta}_1^{\text{FW}}$ ; in the first forward sweep,  $\hat{\Theta}_1^{\text{FW}}$  can be initialized by an estimate obtained from a non-data aided algorithm; if a backward sweep has been performed previously, then  $\hat{\Theta}_1^{\text{FW}} = \hat{\Theta}_1^{\text{BW}}$ .
- 2) For  $k = 2, \dots, L$ 
  - a) The equality constraint node  $k$  receives the estimate  $\hat{\Theta}_{k-1}^{\text{FW}}$  from the phase noise node at its left side and subsequently computes the estimates  $\hat{\Theta}_k^{\text{FW}}$

according to the rule:

$$\hat{\theta}_k^{FW} = \hat{\theta}_{k-1}^{FW} + \lambda \left. \frac{d \log \mu_{g \rightarrow \Theta_k}(\theta)}{d\theta} \right|_{\theta = \hat{\theta}_{k-1}^{FW}}. \quad (20)$$

- b) The equality constraint node  $k$  sends the estimate  $\hat{\theta}_k^{FW}$  to the phase noise node at its right side, which sends this estimate (unchanged) to the equality constraint node  $k + 1$ .

In the backward sweep, the estimates  $\hat{\Theta}_k^{BW}$  are updated from right to left in a similar fashion. The estimate  $\hat{\Theta}_L^{BW}$  is initialized as  $\hat{\Theta}_L^{BW} = \hat{\Theta}_L^{FW}$ , where  $\hat{\Theta}_L^{FW}$  has been computed in the previous forward sweep. After a sufficient number—typically between 10 and 100—of forward and backward sweeps, the final estimates  $\hat{\Theta}_k$  are computed as the average of the estimates  $\hat{\Theta}_k^{FW}$  and  $\hat{\Theta}_k^{BW}$  computed in the last forward and backward sweep respectively.

#### D. Particle Filtering

A probability distribution can be represented by a list of samples (also referred to as “particles”) from the distribution. This data type is the foundation of the particle filter [16]. In this approach, integrals (such as in the right hand side of Equations (10) and (13)) are replaced by weighted averages over the list of samples. We have developed particle filtering techniques that in contrast to existing methods, do not represent the whole distribution as a list of samples. Instead, the particle methods we propose are intended to locate the argmax of unimodal distributions. In the following, we explain how we applied these methods for estimating the phase in the two phase models presented in Section I.

In the graph of the constant-phase model (see Figure 2), the message  $\mu_{\square \rightarrow \Theta}(\theta)$  out of the compound equality constraint node along the  $\Theta$  edge is represented as a list of samples  $\hat{\theta}_i$ ,  $i = 1, 2, \dots, M$ . The latter are obtained as follows:

- 1) Initialize the list of samples. This initial list may be generated by taking  $M/L$  samples from each distribution  $\mu_{g \rightarrow \Theta_k}(\theta_k)$ .
- 2) The compound equality constraint node broadcasts the list of samples  $\hat{\theta}_i^{\text{old}}$  to all multiply nodes.
- 3) The multiply nodes reply with the values  $\mu_{g \rightarrow \Theta_k}(\hat{\theta}_i^{\text{old}})$ .
- 4) At the equality constraint node, the weights  $w_i$  are computed as

$$\tilde{w}_i = \prod_{k=1}^L \mu_{g \rightarrow \Theta_k}(\hat{\theta}_i^{\text{old}}) \quad (21)$$

$$w_i = \gamma \tilde{w}_i, \quad (i = 1, 2, \dots, M) \quad (22)$$

where  $\gamma^{-1} = \sum_{i=1}^M \tilde{w}_i$ .

- 5) The list of samples is updated according to the rule

$$\hat{\theta}_i^{\text{new}} = \arg \left[ (1 - \varepsilon) e^{j\hat{\theta}_i^{\text{old}}} + \varepsilon e^{j\bar{\theta}} \right] \quad (i = 1, 2, \dots, M), \quad (23)$$

where  $\bar{\theta}$  is the (weighted) average of the list of samples

$$\bar{\theta} = \arg \sum_{i=1}^M w_i e^{j\hat{\theta}_i^{\text{old}}}. \quad (24)$$

- 6) Iterate the steps 2 to 5 as often as one wishes.

The parameter  $\varepsilon$  should be sufficiently small (e.g.,  $\varepsilon = 0.1$ ). After performing the above steps, the samples in each list are located in close vicinity of the argmax of the distribution  $\mu_{\square \rightarrow \Theta}(\theta)$ .

In the graph of the random-walk phase model (see Figure 3), the messages  $\mu_{\Theta_k \rightarrow g}(\theta_k)$  are represented by lists of samples  $\hat{\theta}_{k,i}$ ,  $k = 1, 2, \dots, L$  and  $i = 1, 2, \dots, M$ . The latter are obtained as follows:

- 1) Initialize the lists of samples. These initial lists may be all identical; they may be generated by taking  $M/L$  samples from each distribution  $\mu_{g \rightarrow \Theta_k}(\theta_k)$ .
- 2) Set the weights  $w_{1,i}^{FW}$  and  $w_{L,i}^{BW}$  equal to 1.
- 3) The weights  $\tilde{w}_{k,i}^{FW}$  are computed by the following forward recursion:

$$\tilde{w}_{k,i}^{FW} = w_{k-1,i}^{FW} \cdot \mu_{g \rightarrow \Theta_{k-1}}(\hat{\theta}_{k-1,i}) \quad (25)$$

$$w_{k,i}^{FW} = \gamma \tilde{w}_{k,i}^{FW}, \quad (26)$$

where  $\gamma^{-1} = \sum_{i=1}^M \tilde{w}_{k,i}^{FW}$ .

- 4) The weights  $\tilde{w}_{k,i}^{BW}$  are computed by the backward recursion:

$$\tilde{w}_{k,i}^{BW} = w_{k+1,i}^{BW} \cdot \mu_{g \rightarrow \Theta_{k+1}}(\hat{\theta}_{k+1,i}) \quad (27)$$

$$w_{k,i}^{BW} = \gamma \tilde{w}_{k,i}^{BW}, \quad (28)$$

where  $\gamma^{-1} = \sum_{i=1}^M \tilde{w}_{k,i}^{BW}$ .

- 5) The weights  $\tilde{w}_{k,i}$  are computed:

$$\tilde{w}_{k,i} = w_{k,i}^{FW} \cdot w_{k,i}^{BW} \quad (29)$$

$$w_{k,i} = \gamma \tilde{w}_{k,i}, \quad (30)$$

where  $\gamma^{-1} = \sum_{i=1}^M \tilde{w}_{k,i}$ .

- 6) The lists of samples are updated according to the rule

$$\hat{\theta}_{k,i}^{\text{new}} = \arg \left[ (1 - \varepsilon) e^{j\hat{\theta}_{k,i}^{\text{old}}} + \varepsilon e^{j\bar{\theta}_k} \right] \quad (i = 1, 2, \dots, M), \quad (31)$$

where  $\bar{\theta}_k$  is the (weighted) average of the list of samples

$$\bar{\theta}_k = \arg \sum_{i=1}^M w_{k,i} e^{j\hat{\theta}_{k,i}^{\text{old}}}. \quad (32)$$

- 7) Iterate the steps 2 to 6 as often as desired.

After a sufficient number of iterations, the samples  $\hat{\theta}_{k,i}$  in list  $k$  are located in close vicinity of the argmax of the distribution  $\mu_{\Theta_k \rightarrow g}(\theta_k)$ .

## IV. RESULTS AND DISCUSSION

We performed simulations of the proposed algorithms for joint decoding and phase estimation. We used a fixed rate 1/2 LDPC code of length 100 that was randomly generated and was not optimized for the channel at hand. The factor graph of the code does not contain cycles of length four. The degree of all bit nodes equals 3; the degrees of the check nodes are distributed as follows: 1, 14, 69 and 16 check nodes have degree 4, 5, 6 and 7 respectively. The symbol constellation was Gray-encoded 4-PSK. We iterated three times between the LDPC decoder and the phase estimator,

each time with hundred iterations inside the LDPC decoder. We did not iterate between the LDPC decoder and the mapper. The phase ambiguity was resolved by testing the  $M$  possible hypotheses. In the particle filtering algorithms, the messages were represented as lists of 100 samples; in the numerical integration algorithms, the phase was uniformly quantized over  $n = 100$  levels. In the gradient methods, we optimized the (constant) step size parameter  $\lambda$ . In Figures 5 and 6, the phase synchronizers are compared in terms of mean squared (phase) estimation error (MSEE) and the decoded frame error rate (FER) as a function of the SNR respectively; the figures show that the particle filter algorithms have the best performance, followed by the algorithms based on numerical integration and then by the gradient methods—the more accurate the phase messages are represented, the better the performance.

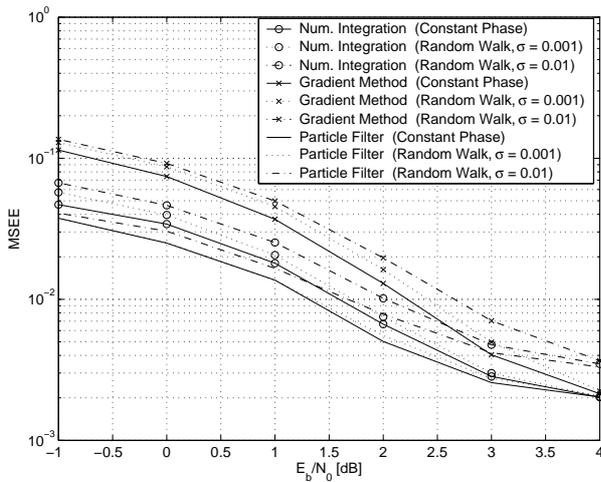


Fig. 5. Mean squared estimation error.

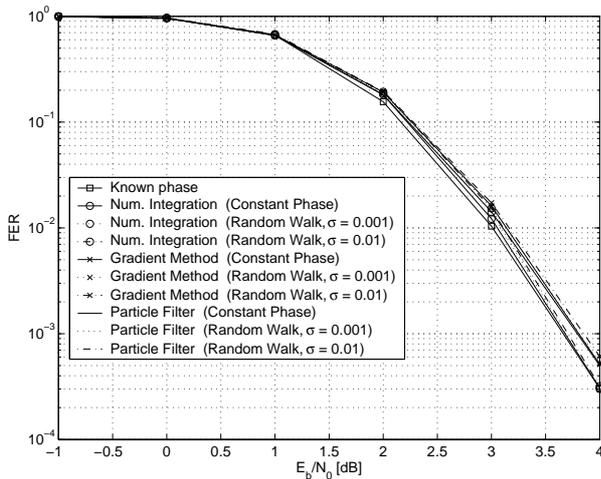


Fig. 6. Frame error rate.

The gradient methods have the lowest complexity, since the messages are represented by only a few parameters. The algo-

rithms based on numerical integration are much more complex. As is well known, numerical integration becomes infeasible in high-dimensional systems. The particle methods are complex as well, but they scale better with the dimensionality of the system.

## V. CONCLUSION

We have presented several message passing algorithms, applicable to several channel models, for joint decoding and phase estimation. In contrast to prior and parallel work on such schemes by other authors, we have emphasized the systematic exploration of such algorithms starting from a factor graph of the channel model.

## ACKNOWLEDGMENT

This project was supported in part by the Swiss National Science Foundation grant 200021-101955.

We heavily used the collection of C programs for LDPC codes by Radford M. Neal from <http://www.cs.toronto.edu/~radford/ldpc.software.html>.

## REFERENCES

- [1] D. Lee, "Analysis of Jitter in Phase-Locked Loops," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 11, Nov. 2002, pp. 704–711.
- [2] Q. Huang, "Phase Noise to Carrier Ratio in LC Oscillators" *IEEE Trans. Circuits Syst. I*, vol. 47, no. 7, Jul. 2000, pp. 965–980.
- [3] T. Lee and A. Hajimiri, "Oscillator Phase Noise: A Tutorial" *IEEE J. Solid-State Circuits*, vol. 35, no. 3, Mar. 2000, pp. 326–336.
- [4] 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, no. 5, May 2000, pp. 655–674.
- [5] A. Burr and L. Zhang, "Application of Turbo-Principle to Carrier Phase Recovery in Turbo Encoded Bit-Interleaved Coded Modulation System," *Proc. 3rd International Symposium on Turbo Codes and Related Topics*, Brest, France, 1–5 Sept., 2003, pp. 87–90.
- [6] I. Sutskever, S. Shamai and J. Ziv, "A Novel Approach to Iterative Joint Decoding and Phase Estimation," *Proc. 3rd International Symposium on Turbo Codes and Related Topics*, Brest, France, 1–5 Sept., 2003, pp. 83–86.
- [7] H. Steendam, N. Noels, M. Moeneclaey, "Iterative Carrier Phase Synchronization for Low-Density Parity-Check Coded Systems", *International Conference on Communications 2003*, Anchorage, Alaska, May 11–15, 2003, pp. 3120–3124.
- [8] N. Noels et al., "Turbo synchronization : an EM algorithm interpretation", *International Conference on Communications 2003*, Anchorage, Alaska, May 11–15, 2003, pp. 2933–2937.
- [9] V. Lottici, M. Luise, "Carrier Phase Recovery for Turbo-Coded Linear Modulations", *International Conference on Communications 2002*, New York, NJ, 28 April–2 May, 2002, vol. 3, pp. 1541–1545.
- [10] R. Nuriyev and A. Anastasopoulos, "Analysis of joint iterative decoding and phase estimation for the noncoherent AWGN channel, using density evolution," *Proc. 2002 IEEE Information Theory Workshop*, Lausanne, Switzerland, June 30 – July 5, 2002, p. 168.
- [11] B. Mielczarek and A. Svensson, "Phase offset estimation using enhanced turbo decoders," *IEEE International Conference on Communications, ICC 2002*, vol. 3, 28 April–2 May, 2002, pp. 1536–1540.
- [12] H. -A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, Jan. 2004, pp. 28–41.
- [13] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.
- [14] H.-A. Loeliger, "Some Remarks on Factor Graphs", *Proc. 3rd International Symposium on Turbo Codes and Related Topics*, 1–5 Sept., 2003, pp. 111–115.
- [15] M. Meyr, M. Moeneclaey and S. A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation and Signal Processing*, New York, John Wiley & Sons, 1998, pp. 246–249.
- [16] A. Doucet, N. de Freitas and N. J. Gordon, eds., *Sequential Monte Carlo Methods in Practice*, New York, Springer-Verlag, 2001.