Joint Channel Estimation and Channel Decoding in Physical-Layer Network Coding Systems: An EM-BP Factor Graph Framework

Taotao Wang, *Student Member, IEEE*, and Soung Chang Liew, *Fellow, IEEE*

Abstract—This paper addresses the problem of joint channel estimation and channel decoding in physical-layer network coding (PNC) systems. In PNC, multiple users transmit to a relay simultaneously. PNC channel decoding is different from conventional multi-user channel decoding: specifically, the PNC relay aims to decode a network-coded message rather than the individual messages of the users. Although prior work has shown that PNC can significantly improve the throughput of a relay network, the improvement is predicated on the availability of accurate channel estimates. Channel estimation in PNC, however, can be particularly challenging because of 1) the overlapped signals of multiple users; 2) the correlations among data symbols induced by channel coding; and 3) time-varying channels. We combine the expectation-maximization (EM) algorithm and belief propagation (BP) algorithm on a unified factor-graph framework to tackle these challenges. In this framework, channel estimation is performed by an EM subgraph, and channel decoding is performed by a BP subgraph that models a virtual encoder matched to the target of PNC channel decoding. Iterative message passing between these two subgraphs allow the optimal solutions for both to be approached progressively. We present extensive simulation results demonstrating the superiority of our PNC receivers over other PNC receivers.

Index Terms—Physical-layer network coding, expectation-maximization, belief propagation, factor graph, message passing.

I. INTRODUCTION

RECENTLY, the research community has shown growing interest in a simple relay network in which two terminal nodes communicate via a relay. This network is referred to as the two-way relay channel (TWRC). Much of the interest in TWRC is on the exploitation of physical-layer network coding (PNC) [1], [2] to boost its throughput.

Ref. [1] showed that PNC could increase TWRC throughput by 100% compared with traditional relaying [2]. In TWRC operated with PNC, the two terminal nodes first transmit their

messages simultaneously to the relay. The relay then maps the overlapped signals to a network-coded message (e.g., bit-wise XOR of the messages of the terminal nodes) and broadcasts the network-coded message to the two terminal nodes. Each terminal node then extracts the message of the other terminal node by subtracting its own message from the network-coded message. Thus, the two terminal nodes exchange one message with each other in two time slots. With traditional relaying, four time slots are needed [1].

This throughput advantage of PNC, however, is predicated on the accurate estimate of the channels between the terminal nodes and the relay. For optimality, it is desirable to obtain the maximum *a posteriori* probability (MAP) channel estimates. This is, however, a particularly challenging task for PNC. The involved issues addressed by this paper are as follows:

- For reliable communication, we consider channel-coded PNC [2], [3]. Specifically, the source messages of the two terminal nodes are channel-coded into channel-coded messages before transmission. The signals received by the relay contain the overlapped channel-coded messages as well as overlapped preambles (training symbols) and pilots.
- PNC channel decoding is different from conventional multiuser channel decoding. The goal of the PNC channel decoding at the relay is to obtain a network-coded message rather than the two individual source messages [2], [3]. In other words the relay aims not to decode the two source messages, but to decode a network-coding function of the two source messages (in this paper, we assume the network-coding function is the bit-wise XOR of the two source messages).
- For successful decoding, accurate channel estimates are needed. For optimality, it is desirable to estimate the channels using not just the preambles and pilots, but also the data in the signals. This is because the data portion also contains useful information related to the channels.
- We are interested in time-varying channels in which the channel gains vary from symbol to symbol within a packet.

Overall, performing channel estimation and PNC channel decoding when (i) the signals are overlapping; (ii) the data symbols are correlated due to channel coding; and (iii) the channels are time-varying, is a particularly challenging task.

To tackle this challenge, this paper proposes and investigates a joint channel estimation and channel decoding framework.

We argue that directly trying to solve the MAP channel estimation problem and the channel decoding problem in a separate manner is not viable; a solution is found in a combined use of expectation-maximization (EM) algorithm and belief propagation (BP) algorithm that solves the two problems jointly in an iterative manner.

We implement the EM-BP computation as a message passing algorithm on a factor graph [4], [5], in which the component for channel estimation (implemented by EM) and the component for PNC channel decoding (implemented by BP) are interconnected. Through iterative message passing between the EM and BP components and iterative message passing between elements within the BP channel decoding component, the results of EM channel estimation and BP channel decoding improve progressively toward optimality.

Overall, there are three major contributions to this work:

1) This is the first work that applies the EM algorithm for joint channel estimation and channel decoding in PNC systems. Notably, our algorithmic framework includes a schema to deal with time-varying channels. To reduce computation complexity, we further extend our framework by replacing EM with its variant named space alternating generalized expectation-maximization (SAGE).

2) We outline a factor graph framework for iterative message passing algorithm based on the foundations of EM/SAGE and BP. This is the first time EM/SAGE-BP computation for a channel-coded communication system is fully implemented as a message passing algorithm on a factor graph (prior work either did not consider channel coding or did not use the factor graph schema). In particular, we explicitly establish this EM/SAGE-BP factor graph framework from a rigorous theoretical foundation. We remark that although our focus here is on PNC systems, this framework is also applicable to the conventional single-user system and multi-user system.

3) Through extensive computer simulations, we investigate the performance of our EM/SAGE-BP PNC receivers and compare them with other existing receivers in the literatures. The simulation results demonstrate the superiority of our receivers over other receivers, confirming the theoretical optimality of EM/SAGE-BP PNC.

A. Related Works

Theory of BP and Its Application in PNC:

BP, also known as the sum-product message passing algorithm, is a general algorithmic inference method for graph models [4], [6]. It has found great success in the decoding of powerful channel codes (e.g., Turbo codes and LDPC codes [7]) in point-to-point communications. BP was first applied to PNC channel decoding in [3], which puts forth the concept of 'virtual encoding'. The concept was further generalized in [8], [9]. Refs. [3], [8], [9] were followed by many other papers on channel-coded PNC [2]. The prior studies of PNC channel decoding mostly assume that the channels are perfectly known. In practice, these channels have to be estimated. PNC systems are multi-user systems in which there are multiple channel parameters, whose estimation is particularly challenging.

It is desired that we could jointly solve channel estimation and channel decoding by BP. Recently, [10] proposed a BP method for joint channel estimation and channel decoding in PNC systems. The direct application of BP to channel estimation, however, requires summations (integrations) over continuous channel variables, which are computationally intensive. To reduce computation complexity, a moment matching (MM) technique is used in [10]. Specifically, the original Gaussian-mixture messages in the BP algorithm are approximated by Gaussian messages that have the same first and second moments. However, as a consequence of the approximation by MM, the optimality cannot be guaranteed. By contrast, in this paper, instead of using a pure BP method for both channel estimation and channel decoding, we use an EM-BP method that obviates the need for computation-intensive integration without sacrificing optimality.

Theory of EM and Its Application in Single-user Systems:

Ref. [11] first proposed EM as a general iterative algorithm for finding the maximum likelihood (ML) estimates of parameters in statistical model with hidden variables that cannot be observed directly. A small extension allows the finding of the MAP estimates also [12]. A variant of EM named SAGE was proposed in [13] to increase the convergence rate. Recently, [14], [15] proposed a framework to implement EM computation as a message passing algorithm on a factor graph. Application for communications systems (and in particular, channel-coded communications systems) was not its main target. It is not clear that the factor graph representation in [14], [15] is applicable to the communication problem of interest to us here. With respect to our contribution 2) listed above, in this paper we fill in the missing details and show specifically how a factor graph representation can be used in our problem. In addition, we extend the framework of EM message passing over factor graphs to SAGE message passing.

Papers [11]–[15] concern EM/SAGE in general. There have also been many investigations on the application of EM/SAGE in communication systems specifically. Refs. [16]–[22] for example, applied EM to the problem of joint channel estimation and detection/decoding in single-user systems. Refs. [23], [24] applied SAGE to the same problem. The work [22] directly applied the results in [14], [15] to implement EM message passing for joint channel estimation and detection in a single-user system. However, the incorporation of channel codes by [22] is heuristic and is not in accordance with the rigorous theoretical derivation in Section III of our paper here.

EM Application in Multi-user Systems:

Refs. [25]–[28] applied EM to joint channel estimation and multi-user detection in CDMA systems. Channel coding was not considered. Ref. [29] incorporated channel coding. However, the proposed scheme performs serial interference cancelation (SIC) and tries to decode the individual messages of different users using separate channel decoders. Decoding individual messages is an overkill for PNC systems and may lead to suboptimal performance (we will provide results showing this in Section V). The application of SAGE in multi-user systems for joint channel estimation and detection can be found in [26], [30], [31], where [26], [30] assume CDMA systems and [31] assumes OFDMA systems.

Overall, there has been little multi-user EM work that

incorporates channel coding. Furthermore, with respect to our contribution 1) listed above, it is worth re-emphasizing that PNC channel decoding [3], [8], [9] is different from conventional multiuser channel decoding [29], because PNC aims to channel-decode the overlapped received signals into a network-coded message [3], [8], [9] rather than the individual messages of different users.

The rest of this paper is organized as follows. Section II describes our system model. In section III, we apply EM-BP to PNC and derive the update equations for EM-BP message passing. Section IV applies SAGE-BP to PNC. Section V presents the simulation results. Section VI concludes this paper.

Notations: We denote matrices by bold capital letters, vectors by bold small letters, and scalars by regular letters throughout this paper. All vectors are column vectors. $\mathbf{I}_r$ denotes the $r \times r$ identity matrix and $\mathbf{0}_r$ denotes the $r \times r$ matrix with all zero elements. $\mathbf{A}^T$, $\mathbf{A}^H$, $\mathbf{A}^{-1}$ and $\det(\mathbf{A})$ denote the transpose, the conjugate transpose, the inverse and the determinant of $\mathbf{A}$, respectively. $\mathcal{CN}(\mathbf{x}:\mathbf{m},\mathbf{K}) \triangleq \frac{1}{\pi^r \det(\mathbf{K})} \exp\left[-(\mathbf{x}-\mathbf{m})^H \mathbf{K}^{-1}(\mathbf{x}-\mathbf{m})\right]$ denotes the probability density function (PDF) of an r-dimension complex Gaussian random variable $\mathbf{x}$ with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{K}$. $\|\mathbf{x}\|$ is the Euclidean norm of a vector $\mathbf{x}$. $(\cdot)^*$ denotes the conjugate of a complex number. $|C|$ is the cardinality of a set C. $x_{i:j}$ is a set containing the elements in a sequence x indexed from i to j, inclusively. Finally, $\otimes$ denotes the Kronecker product, and $\oplus$ denotes the exclusive-or operation.

II. SYSTEM MODEL

We consider a two-phase PNC transmission scheme for TWRC consisting of an uplink phase and a downlink phase. In the uplink phase, two terminal nodes A and B transmit packets to a relay node R simultaneously. From the overlapped signals received from A and B, R constructs a network-coded packet and broadcasts it to A and B in the downlink phase. From the network-coded packet, A(B) then recovers the packet of B(A) using its self information [2].

This paper focuses on the uplink phase; the problem of reliably transmitting the network-coded packet in the downlink phase is similar to that in a conventional point-to-point link. We assume A and B have one transmit antenna, and R has one receive antenna. In the uplink phase, the received signal at R in the ith symbol duration can be expressed as

$$y_i = h_{A,i} x_{A,i} + h_{B,i} x_{B,i} + n_{R,i} = \mathbf{h}_i^T \mathbf{x}_i + n_{R,i} \quad (1)$$

where $x_{A,i}$ ($x_{B,i}$) is the ith transmitted symbol of node A(B); h_i^A (h_i^B) is the ith fading coefficient of the channel between A(B) and R; $n_{R,i}$ is the complex white Gaussian noise sample with zero mean and variance N_0; $\mathbf{h}_i \triangleq [h_i^A, h_i^B]^T$; and $\mathbf{x}_i \triangleq [x_{A,i}, x_{B,i}]^T$. A block diagram of the system model is shown in Fig. 1, where $\{s_{A,j}\}$ and $\{s_{B,j}\}$ are the source information bits from nodes A and B. The transmitted symbols $\{x_{A,i}\}$ and $\{x_{B,i}\}$ are generated after channel encoding, interleaving, constellation mapping and pilot insertions at the transmitters. In this work, we assume that A and B use the same channel
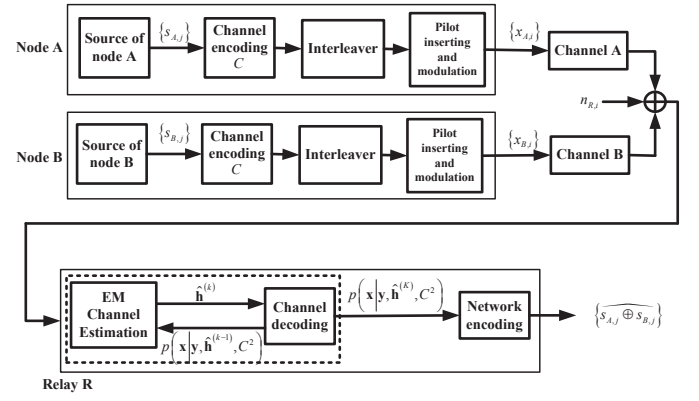

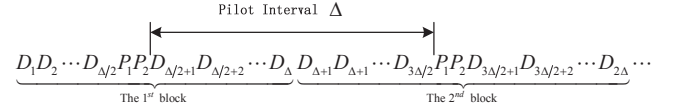Fig. 1.   The system model of uplink phase in TWRC.

Fig. 2.   The frame structure employed by the two user nodes.

encoder[1] (the valid set of codewords is denoted by C) and the same interleaver when mapping their source bits $\{s_{A,j}\}$ and $\{s_{B,j}\}$ to transmitted symbols. Pilot symbols are inserted periodically among coded data symbols. The assumed frame structure is shown in Fig. 2 where P and D represent the pilot symbols and coded data symbols, respectively. Each frame consists of l data symbols, divided into l/Δ blocks. Each block has Δ data symbols and two pilot symbols. We refer to Δ as the pilot interval.[2] The total frame length is $L = l + 2(l/\Delta)$ symbols.

We assume time-varying Rayleigh fading channels. The time-varying channel gains $\{h_{A,i}\}$ and $\{h_{B,i}\}$ are modeled as two independent first-order Gauss-Markov processes [32]:

$$\begin{aligned} h_{A,i} &= \alpha h_{A,i-1} + \sqrt{1-\alpha^2} z_{A,i} \\ h_{B,i} &= \alpha h_{B,i-1} + \sqrt{1-\alpha^2} z_{B,i} \end{aligned} \quad (2)$$

where $\{z_{A,i}\}$ and $\{z_{B,i}\}$ are independent white complex Gaussian processes with zero mean and variances σ_A^2 and σ_B^2 for all i. The parameter α is a correlation coefficient used for modeling how fast the channel varies with time (i.e., it is related to the channel coherence time) [32]. The distributions of $h_{A,i}$ and $h_{B,i}$ are zero mean complex Gaussian with variances σ_A^2 and σ_B^2, respectively. Therefore, the amplitude of every element of them is Rayleigh distributed, and the phase is uniformly distributed.

III. APPLICATION OF EM-BP TO PNC

A. Objectives of EM PNC Receiver

Let $\mathbf{h} \triangleq \{\mathbf{h}_i\}$ be the set containing channel gains of all time. Similarly, let $\mathbf{x}$ be the set of all transmitted symbols $\{\mathbf{x}_i\}$, and $\mathbf{y}$ is the set of all received signals $\{y_i\}$. To the relay, both $\mathbf{h}$ and $\mathbf{x}$ are unknowns to be estimated and decoded.

[1]Discussion on how to deal with two nodes with different channel encoders can be found in Section IV.

[2]Simulation results on the impact of the pilot interval on system performance can be found in Section V.

In a conventional receiver, $\mathbf{h}$ is first estimated, followed by the decoding of codewords $\mathbf{x}$. Pilots, corresponding to known $\mathbf{x}_i$ at specific positions i, are used for the estimation of $\mathbf{h}$. After that, $\mathbf{h}_i$ for the positions occupied by data are estimated by interpolation. This estimate of $\mathbf{h}$ is then substituted into (1) for the decoding of the data $\mathbf{x}_i$. This estimate of $\mathbf{h}$ makes use of only the pilot parts, and does not exploit useful information contained in the data part of $\mathbf{x}$.

In our PNC receiver design, we wish to make use of the pilots as well as the data parts of $\mathbf{x}$ in the channel estimation process. In particular, observations $\mathbf{y}$ at all positions i and the knowledge of the used channel code will be employed to estimate $\mathbf{h}$. A possible strategy for our channel estimation and channel decoding problem in PNC receiver is as follows:

Step 1 (channel estimation): Find MAP estimate $\widehat{\mathbf{h}}_{\mathrm{MAP}} = \arg\max_{\mathbf{h}} \left\{ \log p\left(\mathbf{h}\,\middle|\,\mathbf{y}, C^2\right) \right\} = \arg\max_{\mathbf{h}} \left\{ \log \sum_{\mathbf{x}} p\left(\mathbf{x}, \mathbf{h}\,\middle|\,\mathbf{y}, C^2\right) \right\}$

Step 2 (channel decoding): Given $\widehat{\mathbf{h}}_{\mathrm{MAP}}$, find $p\left(\mathbf{x}\,\middle|\,\widehat{\mathbf{h}}_{\mathrm{MAP}}, \mathbf{y}, C^2\right)$

Step 3 (network coding): Compute the network-coded source message $\left\{ \widehat{s_{A,j} \oplus s_{B,j}} \right\}$ based on the channel decoding output from Step 2 [2].

This is the PNC receiver with optimal channel estimation. Some remarks on the receiver are as follows:

1) A subtlety here in Step 2 for the PNC system is that C^2 is the code constraint by the 'virtual channel encoder' which takes the original information source symbols from nodes A and B $\{s_{A,j}, s_{B,j}\}$ as inputs, and outputs $\{\mathbf{x}_i\}$ as coded symbols (see [2] for details).

2) If the channel coefficients were perfectly known (as assumed in previous works [3], [8], [9]), then Step 1 would not be needed. Step 2 and Step 3 then form the so-called Channel-decoding-Network-Coding (CNC) process, an essence of channel-coded PNC systems [2], [3]. Compared to conventional channel decoding, the goal of CNC is not to decode the individual source messages of A and B, but to decode a network-coded message that mixes the two source messages (we refer the interested readers to [2] and references therein for details on CNC). If the MAP estimation in Step 1 could be achieved, then Step 2 and Step 3 could be implemented using the conventional CNC methods, substituting $\widehat{\mathbf{h}}_{\mathrm{MAP}}$ as the channel coefficients [2], [3]. Unfortunately, this is not viable because the exact MAP estimate of $\mathbf{h}$ is difficult due to the complexity of the computation of $\sum_{\mathbf{x}} p\left(\mathbf{h}, \mathbf{x}\,\middle|\,\mathbf{y}\right)$. A difficulty, for example, is that the symbols in $\mathbf{x}$ are correlated due to channel coding; in addition, signals of the two terminal nodes are overlapped in $\mathbf{y}$.

To facilitate the design of PNC receiver, this paper makes use of the EM algorithm for channel estimation. Specifically, EM tries to find $\widehat{\mathbf{h}}_{\mathrm{MAP}}$ iteratively rather than attacking the problem directly. The objective of EM is still to obtain the MAP estimate of $\mathbf{h}$ as in Step 1. However, EM combines Step 1 and Step 2 in an iterative manner to refine the estimate of $\mathbf{h}$ and the decoding of the network-coded message. In the following, we first describe the procedure of the EM computation and then present its implementation as a message passing algorithm on a factor graph.

In the terminology of EM, $\mathbf{y}$ is the observed data, $\mathbf{x}$ is the hidden data, the pair $(\mathbf{x}, \mathbf{y})$ is the complete data, and $\mathbf{h}$ is the unknown parameter. The kth iteration of EM consists of an E-step (expectation) and an M-step (maximization) as follows [12]:

E-step: Given the previous estimate $\widehat{\mathbf{h}}^{(k-1)}$, compute the conditional expectation

$$Q\left(\mathbf{h}\,\middle|\,\widehat{\mathbf{h}}^{(k-1)}\right) = \sum_{\mathbf{x}} p\left(\mathbf{x}\,\middle|\,\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right) \log p\left(\mathbf{y}, \mathbf{x}|\mathbf{h}, C^2\right) \tag{3}$$

M-step: Then, compute $\widehat{\mathbf{h}}^{(k)}$ by

$$\widehat{\mathbf{h}}^{(k)} = \arg\max_{\mathbf{h}} \left[Q\left(\mathbf{h}\,\middle|\,\widehat{\mathbf{h}}^{(k-1)}\right) + \log p(\mathbf{h}) \right] \tag{4}$$

The E-step in (3) can be broken down as follows. First, compute $p\left(\mathbf{x}\,\middle|\,\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)$ from $\mathbf{y}$ and $\widehat{\mathbf{h}}^{(k-1)}$. This computation of $p\left(\mathbf{x}\,\middle|\,\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)$ is similar to Step 2 above, with $\widehat{\mathbf{h}}^{(k-1)}$ replacing $\widehat{\mathbf{h}}_{\mathrm{MAP}}$. If the algorithm were to stop at iteration $k-1$, we could simply go to Step 3 to obtain the network-coded message based on $p\left(\mathbf{x}\,\middle|\,\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)$. Otherwise, the E-step continues and uses $p\left(\mathbf{x}\,\middle|\,\mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)$ to compute the $Q\left(\mathbf{h}\,\middle|\,\widehat{\mathbf{h}}^{(k-1)}\right)$ in (3). After that, the M-step finds a new estimate of $\mathbf{h}$ as in (4). In this way, the computation procedure jointly estimates the channels $\mathbf{h}$ and decodes the data $\mathbf{x}$, refining the solutions from one iteration to the next. This process is shown in the lower half of Fig. 1, which depicts our PNC receiver. An explanation on why EM works can be found in our technical report [33], where we use an argument based on K-L divergence to interpret the physical meaning of the EM algorithm.

B. Implementation of EM-BP PNC on Factor Graph

Although EM can iteratively find the MAP estimate, the computations of the E-step (3) and M-step (4) are still nontrivial. We next establish a framework that implement the EM computation as a message passing algorithm on a factor graph, where we can rigorously combine EM message passing for channel estimation with BP message passing for channel decoding. This framework allows us to more clearly see the intricacies of the EM computation, pointing to a systematic and practical way to implement it.

Refs [14], [15] showed how to transform a generic EM computation to a factor graph implementation. It is not clear, however, that the assumptions in [14], [15] on the functional forms of the parameter and variable probabilities are valid for our specific problem. Here, we give a theoretical derivation tailored to channel-coded communication systems.

A key to factor graph implementation is to factorize $p\left(\mathbf{y}, \mathbf{x}|\mathbf{h}, C^2\right)$ in (3) and $p(\mathbf{h})$ in (4). For $p\left(\mathbf{y}, \mathbf{x}|\mathbf{h}, C^2\right)$, we write

$$p\left(\mathbf{y}, \mathbf{x}|\mathbf{h}, C^2\right) = p\left(\mathbf{y}\,|\,\mathbf{x}, \mathbf{h}\right) p\left(\mathbf{x}\,\middle|\,C^2\right) = \frac{I_{C^2}(\mathbf{x}) \prod_i p(y_i|\mathbf{x}_i, \mathbf{h}_i)}{|C^2|} \tag{5}$$

where $I_{C^2}(\mathbf{x})$ is a indicator function defined as: $I_{C^2}(\mathbf{x}) = 1$ if $\mathbf{x} \in C^2$; $I_{C^2}(\mathbf{x}) = 0$ if $\mathbf{x} \notin C^2$, where we have assumed all

codewords are equally likely. Note that we have used (1) to justify the factorization of $p\left(\mathbf{y}\,|\mathbf{x},\mathbf{h}\right)$ into $\prod_i p\left(y_i\,|\mathbf{x}_i,\mathbf{h}_i\right)$ in (5), where y_i is independent of y_j for all $i \neq j$ given $\mathbf{x}$ and $\mathbf{h}$ thanks to their independent noise components. Substituting (5) into the Q function in (3) and dropping the term $-\log\left|C^2\right|$, which is independent of $\mathbf{h}$ and therefore does not matter as far as the M-step is concerned, we have

$$Q\left(\mathbf{h}\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)$$
$$= \sum_i \sum_{\mathbf{x}_i} \log p\left(y_i\,|\mathbf{x}_i,\mathbf{h}_i\right) \underbrace{\sum_{\mathbf{x}_{1:i-1},\mathbf{x}_{i+1:L}} p\left(\mathbf{x}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)}_{\triangleq p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)}$$

(6)

where $p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$ is the *a posteriori probability* (APP)[3] that can be computed using BP (sum-product) message passing algorithm for channel decoding on the factor graph [4] of the given channel code C^2, with a fixed channel $\widehat{\mathbf{h}}^{(k-1)}$. We define the symbol-wise Q function as

$$Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right) \triangleq \sum_{\mathbf{x}_i} \log p\left(y_i\,|\mathbf{x}_i,\mathbf{h}_i\right) p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$$

(7)

With complex Gaussian white noise, the above $\log p\left(y_i\,|\mathbf{x}_i,\mathbf{h}_i\right)$ as a function of the variables $\mathbf{x}_i$ and $\mathbf{h}_i$ can be obtained in closed form as $-\left\|y_i - \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\|^2\big/N_0$. We see that once $p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$ is computed by BP channel decoding, $Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)$ as a function of $\mathbf{h}_i$ can be obtained by the weighted sum of $p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)$ with weight $-\left\|y_i - \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\|^2\big/N_0$ over different possible values of $\mathbf{x}_i$. The overall Q function is the sum of symbol-wise Q functions:

$$Q\left(\mathbf{h}\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right) = \sum_{i=1}^L Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)$$

(8)

Using (8), the M-step in (4) is equivalent to

$$\widehat{\mathbf{h}}^{(k)} = \arg\max_{\mathbf{h}}\left(p\left(\mathbf{h}\right)\cdot\prod_{i=1}^L e^{Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}\right)$$

(9)

To see what will happen in the M-step, let us employ the Gauss-Markov channel model (2) and factorize $p\left(\mathbf{h}\right)$ as

$$p\left(\mathbf{h}\right) = \prod_{i=1}^L p\left(\mathbf{h}_i\,|\mathbf{h}_{i-1}\right)$$

(10)

where $p\left(\mathbf{h}_1\,|\mathbf{h}_0\right) = p\left(\mathbf{h}_1\right) = \mathcal{CN}\left(\mathbf{h}_1:\mathbf{0},\mathbf{Q}\right)$ with $\mathbf{Q} \triangleq diag\left(\left[\sigma_A^2,\sigma_B^2\right]\right)$ is the priori information of $\mathbf{h}_1$, and $p\left(\mathbf{h}_i\,|\mathbf{h}_{i-1}\right) = \mathcal{CN}\left(\mathbf{h}_i:\mathbf{m}_{\mathbf{h}_i|\mathbf{h}_{i-1}},\mathbf{K}_{\mathbf{h}_i|\mathbf{h}_{i-1}}\right)$ for $i \geq 2$ is the Markovian transition probability defined by (2). Specifically, $\mathbf{m}_{\mathbf{h}_i|\mathbf{h}_{i-1}} = \alpha\mathbf{h}_{i-1}$, $\mathbf{K}_{\mathbf{h}_i|\mathbf{h}_{i-1}} = \left(1-\alpha^2\right)\mathbf{Q}$ for $i \geq 2$. Substituting (10) into (9) and after some manipulations, we observe that finding $\widehat{\mathbf{h}}^{(k)}$ amounts to finding

$$\widehat{\mathbf{h}}_i^{(k)} = \arg\max_{\mathbf{h}_i} f\left(\widehat{\mathbf{h}}_{1:i-1}^{(k)},\mathbf{h}_i,\widehat{\mathbf{h}}_{i+1:L}^{(k)}\right), \forall i$$

(11)

where

[3]If $\mathbf{x}_i$ is a vector of coded symbols, its APP is given by the soft channel decoder. If it is a vector of pilot symbols, this probability is either 0 or 1, given by the *a priori* information available at the receiver about the pilots.

$$f\left(\widehat{\mathbf{h}}_{1:i-1}^{(k)},\mathbf{h}_i,\widehat{\mathbf{h}}_{i+1:L}^{(k)}\right)$$
$$= \max_{\mathbf{h}_{1:i-1},\mathbf{h}_{i+1:L}}\left\{\prod_{j=1}^L p\left(\mathbf{h}_j\,|\mathbf{h}_{j-1}\right)\prod_{j=1}^L e^{Q_j\left(\mathbf{h}_j\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}\right\}$$
$$= \underbrace{\max_{\mathbf{h}_{1:i-1}}\left\{p\left(\mathbf{h}_1\,|\mathbf{h}_0\right)\prod_{j=1}^{i-1}\left(e^{Q_j\left(\mathbf{h}_j\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}p\left(\mathbf{h}_j\,|\mathbf{h}_{j-1}\right)\right)\right\}}_{(1)} \times$$
$$\underbrace{e^{Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}}_{(2)} \times \underbrace{\max_{\mathbf{h}_{i+1:L}}\left\{\prod_{j=i+1}^L\left(e^{Q_j\left(\mathbf{h}_j\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}p\left(\mathbf{h}_j\,|\mathbf{h}_{j-1}\right)\right)\right\}}_{(3)}$$

(12)

We can now solve the M-step by a message passing algorithm that implements the max-product rule [4], [5] on a factor graph. We construct the factor graph corresponding to (10), and treat $e^{Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}$ (used in (12)) as the input message to the variable node of $\mathbf{h}_i$. The message passing algorithm is a bidirectional algorithm consisting of forward and backward message passing. For each direction, the message passing mechanism is very similar to Viterbi algorithm (VA) for convolutional codes except that the variables $\{\mathbf{h}_i\}$ are continuous. As indicated in (12), there are three messages needed for the computation of $\widehat{\mathbf{h}}_i^{(k)}$: message (1) is the result of message passing in the forward direction; message (2) is the input message to $\mathbf{h}_i$; and message (3) is the result of message passing in the backward direction. At first sight, it may seem that to obtain the messages in (12) (in particular, in the computation associated with the max operation), we need to search over the continuous space of the variables. This would be highly complex. It turns out that that is not the case, as explained below.

Borrowing the jargon from [15], the term $e^{Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}$ is the ith EM message and we abbreviate it as

$$em_i = e^{Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)}.$$

For our application, this EM message has a Gaussian functional form, greatly facilitating message passing on the factor graph. To see the Gaussian form, we write

$$\exp\left(Q_i\left(\mathbf{h}_i\left|\widehat{\mathbf{h}}^{(k-1)}\right.\right)\right)$$
$$= \exp\left(-\sum_{\mathbf{x}_i} p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)\left\|y_i - \mathbf{h}_i^{\mathrm{T}}\mathbf{x}_i\right\|^2\big/N_0\right)$$
$$\propto \mathcal{CN}\left(\mathbf{h}_i:\mathbf{m}_{em_i},\mathbf{K}_{em_i}\right)$$

with mean vector and covariance matrix

$$\mathbf{m}_{em_i} = \left(\mathbf{K}_{\mathbf{x}_i} + \mathbf{m}_{\mathbf{x}_i}\mathbf{m}_{\mathbf{x}_i}^{\mathrm{H}}\right)^{-1}\mathbf{m}_{\mathbf{x}_i}y_i$$

(13a)

$$\mathbf{K}_{em_i} = \left(\mathbf{K}_{\mathbf{x}_i} + \mathbf{m}_{\mathbf{x}_i}\mathbf{m}_{\mathbf{x}_i}^{\mathrm{H}}\right)^{-1}N_0$$

(13b)

where

$$\mathbf{m}_{\mathbf{x}_i} = \sum_{\mathbf{x}_i} p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)\mathbf{x}_i$$

$$\mathbf{K}_{\mathbf{x}_i} = \sum_{\mathbf{x}_i} p\left(\mathbf{x}_i\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C^2\right.\right)\left(\mathbf{x}_i - \mathbf{m}_{\mathbf{x}_i}\right)\left(\mathbf{x}_i - \mathbf{m}_{\mathbf{x}_i}\right)^{\mathrm{H}}$$

are the mean and variance of $\mathbf{x}_i$ computed based on its APP. Therefore, the model underlying the maximization problem (9) is a pure linear Gaussian model: all the messages on the
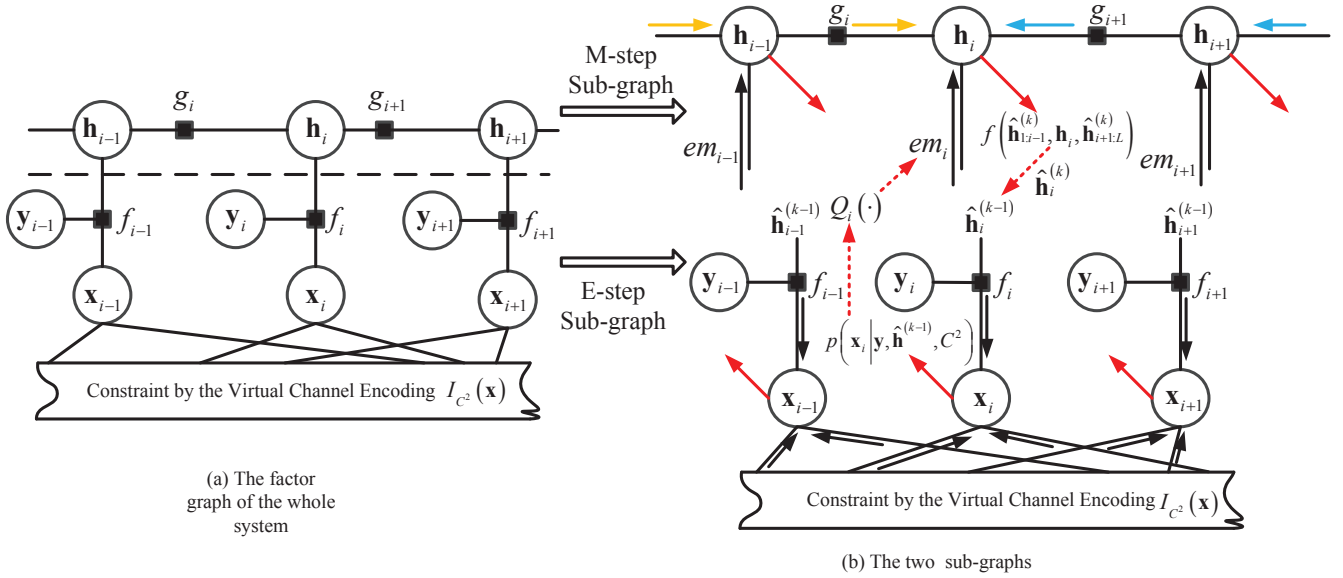
Fig. 3.   The illustration for the implementation of EM-BP PNC on factor graph: (a) The factor graph for the whole system; (b) The two sub-graphs.

factor graph turn out to be Gaussian functions of variables of interest. The maximization problem associated with a linear Gaussian model such as (9) can be solved by Gaussian message passing [5] that implements the max-product rule. For Gaussian message passing, we only need to track the mean vectors and covariance matrices of messages, since together they fully characterize the Gaussian distributions. This avoids searching over the whole continuous feasible region, allowing practical implementation of the message passing algorithm for the M-step.

We have now presented the big picture on how to implement EM channel estimation as a message passing algorithm on a factor graph and to combine it with BP message passing for channel decoding. Before going into the detailed derivation of the update equations for the EM-BP message passing, we make two important remarks:

1) We have shown above that, in channel-coded communication systems, the EM messages of the channels are Gaussian. As specified by (13), the mean and the variance of the ith EM message are linked to the APP of the ith transmitted symbol. This link connects the BP message passing for channel decoding with EM iterations.

2) The APP of $\mathbf{x}_i$ is a product of three parts:

$$p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right) = A \times$$
$$p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right) p\left(y_i \left| \mathbf{x}_i, \widehat{\mathbf{h}}^{(k-1)}\right.\right),$$
$$\text{(14)}$$

where A is a constant independent of $\mathbf{x}_i$, $p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right)$ is the extrinsic information sent from the BP message passing for channel decoding, and $p\left(y_i \left| \mathbf{x}_i, \widehat{\mathbf{h}}^{(k-1)}\right.\right)$ is the information provided by the observation y_i. When the channel code used has cycles in its factor graph (e.g., LDPC, Turbo codes), the BP message passing algorithm for channel decoding requires multiple

iterations to compute the extrinsic information $p\left(\mathbf{x}_i \left| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right)$ for all i [4]. Then, the APPs are updated according to (14). Meanwhile, each EM iteration requires one set of such newly computed APPs for the transmitted symbols. Therefore, the theoretical construct of EM requires us to carry out the multiple iterations of BP channel decoding to update the extrinsic information as soon as the channel has been updated to a new estimate.

Fig. 3 (a) presents the factor graph of the joint probability of our system, i. e. $p\left(\mathbf{y}, \mathbf{x}, \mathbf{h} \left| C^2\right.\right)$. On the factor graph, variable nodes $\{y_i, \mathbf{x}_i, \mathbf{h}_i\}$ are denoted by circles; and check nodes $\left\{f_i \triangleq p\left(y_i \left| \mathbf{x}_i, \mathbf{h}_i\right.\right), g_i \triangleq p\left(\mathbf{h}_i \left| \mathbf{h}_{i-1}\right.\right)\right\}$ are denoted by solid squares. Further, according to the factorization $p\left(\mathbf{y}, \mathbf{x}, \mathbf{h} \left| C^2\right.\right) = p\left(\mathbf{y}, \mathbf{x} | \mathbf{h}, C^2\right) p\left(\mathbf{h}\right)$, we split this overall factor graph into two parts:

E-step subgraph: The subgraph on the lower part of Fig. 3 (b) corresponds to the representation of $p\left(\mathbf{y}, \mathbf{x} | \mathbf{h}, C^2\right)$ through factorization (5). The E-step of EM is implemented by message passing on this subgraph, thus it is labeled as the E-step subgraph. Note that the subgraph that represents the constraint imposed by virtual channel encoding $I_{C^2}(\mathbf{x})$ is embedded inside the E-step subgraph. BP message passing for channel decoding is operating on the subgraph of $I_{C^2}(\mathbf{x})$. Thus, we can regard BP as being a sub-step within the E-step also.

M-step sub-graph: The subgraph on the upper part of Fig. 3 (b) corresponds to the representation of $p\left(\mathbf{h}\right)$ through factorization (10). The M-step of EM is implemented by message passing on this subgraph, thus it is labeled as the M-step subgraph.

In the following, we derive the message update equations for EM-BP PNC. We denote the message sent by a node x to a node y by $\mu_{x \rightarrow y}(\cdot)$. We perform the BP message passing algorithm for *virtual channel decoding* on the E-step subgraph to obtain the soft APP outputs $\left\{p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right.\right)\right\}$. The

BP message passing algorithm for virtual channel decoding can be derived directly by applying the sum-product rule on the factor graph that incorporates the constraints imposed by *virtual channel encoding* [2], [9] that models the simultaneous transmissions by terminal nodes. For virtual channel decoding within the kth EM iteration, the channel $\mathbf{h}$ from the M-step subgraph is fixed to the previous estimate $\widehat{\mathbf{h}}^{(k-1)}$. Thus, the message $\mu_{f_i \to \mathbf{x}_i}$, which initializes the virtual decoding, is given by

$$\mu_{f_i \to \mathbf{x}_i}(\mathbf{x}_i) = p\left(y_i \,|\, \mathbf{x}_i , \widehat{\mathbf{h}}_i^{(k-1)}\right)$$
$$= \mathcal{CN}\left(y_i : \left(\widehat{\mathbf{h}}_i^{(k-1)}\right)^{\mathrm{T}} \mathbf{x}_i, N_0\right) \quad (15)$$

for all i. The E-step of EM is completed by sending the EM messages em_i to the M-step subgraph after BP channel decoding, as indicated by the red dotted arrows in Fig. 3 (b). This is fulfilled by computing the mean vectors and covariance matrices of Gaussian EM messages as in (13) based on the channel decoding outputs $\left\{p\left(\mathbf{x}_i \,\big|\, \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)\right\}$.

Since the structure of the M-step subgraph is a tree, the M-step (9) is exactly implemented by a forward message passing (yellow arrows in upper sub-graph in Fig. 3 (b)) and a backward message passing (blue arrows in upper sub-graph in Fig. 3 (b)). This observation also coincides with the mathematical expression for message passing by equation (12). Due to the assumed Gauss-Markov channel model and the Gaussian feature of the incoming EM messages, all the messages on the M-step subgraph preserve Gaussianity. For forward message passing, it is required to compute the message

$$\mu_{g_i \to \mathbf{h}_i}(\mathbf{h}_i) = \max_{\mathbf{h}_{1:i-1}} \left\{ p(\mathbf{h}_1) \prod_{j=2}^{i} em_{j-1} p(\mathbf{h}_j \,|\, \mathbf{h}_{j-1}) \right\}$$

from $\mu_{g_{i-1} \to \mathbf{h}_{i-1}}(\mathbf{h}_{i-1})$ recursively. Given that $\mu_{g_{i-1} \to \mathbf{h}_{i-1}}(\mathbf{h}_{i-1}) \propto \mathcal{CN}\left(\mathbf{h}_{i-1} : \mathbf{m}_{\mathbf{h}_{i-1}}^f, \mathbf{K}_{\mathbf{h}_{i-1}}^f\right)$ and the Markovian property, we have the following update equations for forward message passing:

$$\mu_{g_i \to \mathbf{h}_i}(\mathbf{h}_i) = \max_{\mathbf{h}_{i-1}} \left\{ \mu_{g_{i-1} \to \mathbf{h}_{i-1}}(\mathbf{h}_{i-1}) \, em_{i-1} p(\mathbf{h}_i \,|\, \mathbf{h}_{i-1}) \right\}$$
$$\propto \mathcal{CN}\left(\mathbf{h}_i : \mathbf{m}_{\mathbf{h}_i}^f, \mathbf{K}_{\mathbf{h}_i}^f\right)$$

with

$$\mathbf{m}_{\mathbf{h}_i}^f = \alpha \mathbf{m}_{\mathbf{h}_{i-1}}^f + \alpha \mathbf{K}_{\mathbf{h}_{i-1}}^f \left(\mathbf{K}_{em_{i-1}} + \mathbf{K}_{\mathbf{h}_{i-1}}^f\right)^{-1} \\ \times \left(\mathbf{m}_{em_{i-1}} - \mathbf{m}_{\mathbf{h}_{i-1}}^f\right) \quad (16a)$$

$$\mathbf{K}_{\mathbf{h}_i}^f = \alpha^2 \mathbf{K}_{\mathbf{h}_{i-1}}^f + \left(1 - \alpha^2\right) \mathbf{Q} - \left(\alpha \mathbf{K}_{\mathbf{h}_{i-1}}^f\right) \\ \times \left(\mathbf{K}_{em_{i-1}} + \mathbf{K}_{\mathbf{h}_{i-1}}^f\right)^{-1} \left(\alpha \mathbf{K}_{\mathbf{h}_{i-1}}^f\right) \quad (16b)$$

The update equations (16) are essentially the famous Kalman one-step prediction equations. The backward update equations that compute message $\mu_{g_{i+1} \to \mathbf{h}_i}(\mathbf{h}_i)$ from $\mu_{g_{i+2} \to \mathbf{h}_{i+1}}(\mathbf{h}_{i+1}) \propto \mathcal{CN}\left(\mathbf{h}_{i+1} : \mathbf{m}_{\mathbf{h}_{i+1}}^b, \mathbf{K}_{\mathbf{h}_{i+1}}^b\right)$ are easily obtained similarly by symmetry; we therefore omit them at here. Finally, the message flowing out of variable node $\mathbf{h}_i$

(the red solid arrow in the upper sub-graph in Fig. 3 (b)) is the product of all the incoming messages

$$\mu_{\mathbf{h}_i \to out}(\mathbf{h}_i) = f\left(\widehat{\mathbf{h}}_{1:i-1}^{(k)}, \mathbf{h}_i, \widehat{\mathbf{h}}_{i+1:L}^{(k)}\right)$$
$$= \mu_{g_i \to \mathbf{h}_i}(\mathbf{h}_i) \times em_i \times \mu_{g_{i+1} \to \mathbf{h}_i}(\mathbf{h}_i)$$
$$\propto \mathcal{CN}\left(\mathbf{h}_i : \mathbf{m}_{\mathbf{h}_i}, \mathbf{K}_{\mathbf{h}_i}\right)$$

where the mean vector $\mathbf{m}_{\mathbf{h}_i}$ is used to update the new estimate of $\mathbf{h}_i$:

$$\widehat{\mathbf{h}}_i^{(k)} = \arg\max_{\mathbf{h}_i} \mu_{\mathbf{h}_i \to out}(\mathbf{h}_i)$$
$$= \mathbf{m}_{\mathbf{h}_i} = \mathbf{m}_{\mathbf{h}_i^-} + \mathbf{G}_{\mathbf{h}_i}\left(\mathbf{m}_{em_i} - \mathbf{m}_{\mathbf{h}_i^-}\right) \quad (17)$$

with

$$\mathbf{K}_{\mathbf{h}_i^-} = \left(\left(\mathbf{K}_{\mathbf{h}_i}^f\right)^{-1} + \left(\mathbf{K}_{\mathbf{h}_i}^b\right)^{-1}\right)^{-1}$$
$$\mathbf{m}_{\mathbf{h}_i^-} = \mathbf{K}_{\mathbf{h}_i^-}\left(\left(\mathbf{K}_{\mathbf{h}_i}^f\right)^{-1} \mathbf{m}_{\mathbf{h}_i}^f + \left(\mathbf{K}_{\mathbf{h}_i}^b\right)^{-1} \mathbf{m}_{\mathbf{h}_i}^b\right)$$
$$\mathbf{G}_{\mathbf{h}_i} = \mathbf{K}_{\mathbf{h}_i^-}\left(\mathbf{K}_{\mathbf{h}_i^-} + \mathbf{K}_{em_i}\right)^{-1}$$

This completes the M-step of EM. We then iterate back to the E-step with the new estimate.

C. Initialization and Termination of EM Iteration

EM iteration needs to be bootstrapped with a good initial point; otherwise there is no guarantee that the algorithm will converge to the global maximum [34]. For each block, we obtain the initial $\widehat{\mathbf{h}}^{(0)}$ by minimum mean square error (MMSE) estimation [35] using only the pilot symbols. Then, the channel coefficients within each data block are simply set to the estimated channel coefficient of the closest pilot.

We repeat the E-step and M-step iteratively. When the number of iterations k reaches a preset maximum limit K, we terminate the EM algorithm after obtaining the final channel estimate $\widehat{\mathbf{h}}^{(K)}$. Substituting $\widehat{\mathbf{h}}^{(K)}$ into the signal model (1) as the real channel $\mathbf{h}$, we could carry out a final round of channel decoding to obtain $p\left(\mathbf{x} \,\big|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right)$, where $\mathbf{x}$ is the overall codeword pair. The objective of this final channel decoding is consistent with Step 2 in Section III-A. However, instead of finding $p\left(\mathbf{x} \,\big|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right)$ for channel decoding, we choose to modify Step 2 by still employing BP for virtual channel decoding to find $p(\mathbf{x}_i | \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2)$ for each and every symbol pair $\mathbf{x}_i$. There are three reasons for this. First, we already have the virtual channel decoder at hand in the factor graph implementation of EM-BP PNC receiver. Second, for many advanced channel codes (e.g., LDPC, Turbo code), the decoding process finds $p(\mathbf{x}_i | \mathbf{y}, \widehat{\mathbf{h}}^{(K)} C^2)$ rather than $p\left(\mathbf{x} \,\big|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right)$, because finding $p\left(\mathbf{x} \,\big|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right)$ for all possible codewords is generally a difficult computation-intensive problem. Finding $p(\mathbf{x}_i | \mathbf{y}, \widehat{\mathbf{h}}^{(K)} C^2)$ for all i also may be treated as a sort of approximation of the original objective 2. Third, even if we could find $p\left(\mathbf{x} \,\big|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right)$, it would not be easy to obtain the ML network-coded message because the ML network-coded codeword is given by

$$\widehat{\mathbf{x}}_R = \widehat{\mathbf{x}_A \oplus \mathbf{x}_B} = \arg\max_{\mathbf{x}_R} \sum_{\substack{\mathbf{x}: \\ \mathbf{x}_A \oplus \mathbf{x}_B = \mathbf{x}_R}} p\left(\mathbf{x} \,\big|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right)$$

Algorithm 1 The EM-BP Message Passing Implementation of Joint Channel Estimation and Channel Decoding in PNC Systems

1: **Initialization:**
2: compute $\widehat{\mathbf{h}}^{(0)}$ by MMSE estimation from pilot symbols;
3: **for** $i = 1$ to K **do**
4:     **The E-step of EM**
5:       compute the inputs message to virtual decoding as (15), with the tentative channel estimate $\widehat{\mathbf{h}}^{(k-1)}$;
6:       compute the soft output APPs, $p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)$, from BP message passing for virtual channel decoding; the number of iterations in the virtual channel decoding is denoted by N_{cd1};
7:       compute the mean vectors and covariance matrices of the EM messages as (13);
8:     **The M-step of EM**
9:       perform the forward message passing for EM channel estimation as (16);
10:       perform the backward message passing for EM channel estimation similarly to (16);
11:       compute the new channel estimate $\widehat{\mathbf{h}}^{(k)}$ as (17);
12: **end for**
13: **Termination:**
14: perform BP message passing for the final virtual channel decoding, with the final channel estimate $\widehat{\mathbf{h}}^{(K)}$; the number of iterations in the virtual channel decoding is denoted by N_{cd2};
15: compute the network-coded information symbols as (18).

where $\mathbf{x}_A \triangleq \{x_{A,i}\}$ and $\mathbf{x}_B \triangleq \{x_{B,i}\}$ are the codeword transmitted by nodes A and B. The complexity for the decoding of $\widehat{\mathbf{x}}_R$ is exponential in the length of codewords. By contrast, the ML network-coded symbol is much easier to find from $p(\mathbf{x}_i|\mathbf{y}, \widehat{\mathbf{h}}^{(K)} C^2)$. It is given by

$$\widehat{x_{A,i} \oplus x_{B,i}} = \arg\max_x \sum_{\substack{\mathbf{x}_i: \\ x_{A,i} \oplus x_{B,i} = x}} p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right).$$

To summarize, with the final channel estimate $\widehat{\mathbf{h}}^{(K)}$, we can use the BP message passing for virtual channel decoding in the factor graph to compute the final decoding results $p(\mathbf{x}_i|\mathbf{y}, \widehat{\mathbf{h}}^{(K)} C^2)$ and $p\left(s_{A,j}, s_{B,j} \,\middle|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right)$ at the same time. Then, the network-coded source message is obtained by

$$\widehat{s_{A,j} \oplus s_{B,j}} = \arg\max_s \sum_{\substack{s_{A,j}, s_{B,j}: \\ s_{A,j} \oplus s_{B,j} = s}} p\left(s_{A,j}, s_{B,j} \,\middle|\, \mathbf{y}, \widehat{\mathbf{h}}^{(K)}, C^2\right) \tag{18}$$

for all j. After that, the relay channel-encodes the network-coded source message and broadcasts the channel-coded message to nodes A and B in the downlink phase. Obtaining the estimate of the network-coded message from $p\left(s_{A,j}, s_{B,j} \,\middle|\, \mathbf{y}, C^2\right)$, which is in turn decoded by imagining a virtual encoder as the source of the information, is referred to as the joint CNC process in [2], [3]. Whereas [2], [3] assumes the channel coefficients are perfectly known, here we assume the channel coefficients are unknown and need

to be estimated as part of the joint CNC process. Our EM-BP factor graph framework seamlessly bridges the channel estimation process and the joint CNC process. We summarize the EM-BP message passing implementation of joint channel estimation and channel decoding in PNC systems in Algorithm 1, where N_{cd1} denotes the number of iterations for the BP channel decoding within the EM iteration; and N_{cd2} denotes the number of iterations for the final BP channel decoding after the termination of EM.

IV. THE APPLICATION OF SAGE-BP TO PNC

SAGE sequentially updates a subset of parameters. Doing so essentially decomposes a higher dimension problem into several lower dimension sub-problems, since only a subset of the parameters are estimated and updated according to the EM mechanism each time. It can be shown that the sequential updates of subsets of parameters always still guarantee convergence [13]. In this section, we apply the theory of SAGE to the problem of joint channel estimation and channel decoding in PNC systems. We also extend the framework of EM-BP message passing over factor graph to SAGE-BP message passing. The motivation for introducing SAGE is to reduce the complexity of the channel estimation part of our framework. To our best knowledge, this is the first attempt to apply SAGE for channel estimation in PNC systems. In fact, in the context of general estimation problems, this is probably the first attempt that integrates the use of SAGE and BP on a unified factor graph.

In our setting, it is natural to break up the problem of estimating $\mathbf{h}$ into two sub-problems, estimation of $\mathbf{h}_A$ and estimation of $\mathbf{h}_B$. That is, we update the channel of node A, $\mathbf{h}_A \triangleq \{h_{A,i}\}$ and the channel of node B, $\mathbf{h}_B \triangleq \{h_{B,i}\}$ separately and alternatively. When updating one channel, we keep the other channel fixed. We formulate the kth SAGE iteration as a two-stage process.

The 1st stage of the kth iteration: $\mathbf{h}_B$ is fixed to $\widehat{\mathbf{h}}_B^{(k-1)}$; and $\mathbf{h}_A$ is updated by

$$\widehat{\mathbf{h}}_A^{(k)} = \arg\max_{\mathbf{h}_A} \left(p(\mathbf{h}_A) \cdot \prod_{i=1}^{L} e^{Q_{A,i}\left(\mathbf{h}_{A,i}\middle|\widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}\right)} \right) \tag{19}$$

where
$$Q_{A,i}\left(\mathbf{h}_{A,i}\middle|\widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}\right) = \sum_{\mathbf{x}_i} \log p\left(y_i \middle| \mathbf{x}_i, \mathbf{h}_{A,i}, \widehat{\mathbf{h}}_{B,i}^{(k-1)}\right) p\left(\mathbf{x}_i \middle| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right) \tag{20}$$

is the Q function of $\mathbf{h}_{A,i}$.

The 2nd stage of the kth iteration: $\mathbf{h}_A$ is fixed to $\widehat{\mathbf{h}}_A^{(k)}$; and $\mathbf{h}_B$ is updated by

$$\widehat{\mathbf{h}}_B^{(k)} = \arg\max_{\mathbf{h}_B} \left(p(\mathbf{h}_B) \cdot \prod_{i=1}^{L} e^{Q_{B,i}\left(\mathbf{h}_{B,i}\middle|\widehat{\mathbf{h}}_A^{(k)}, \widehat{\mathbf{h}}_B^{(k-1)}\right)} \right) \tag{21}$$

where

$$Q_{B,i}\left(\mathbf{h}_{B,i}\middle|\widehat{\mathbf{h}}_A^{(k)}, \widehat{\mathbf{h}}_B^{(k-1)}\right) = \sum_{\mathbf{x}_i} \log p\left(y_i \middle| \mathbf{x}_i, \widehat{\mathbf{h}}_{A,i}^{(k)}, \mathbf{h}_{B,i}\right) p\left(\mathbf{x}_i \middle| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right) \tag{22}$$

is the Q function of $\mathbf{h}_{B,i}$.

	EM-BP PNC	SAGE-BP PNC
channel estimation	$O\left(2\left(5N_u^3 + 4N_u^2 + 3N_u\right)L\right)$	$O\left(24N_uL\right)$
computing EM messages	$O\left(\left(N_u^3 + \left(N_m^{N_u} + 3\right)N_u^2 + \left(2N_m^{N_u} + 1\right)N_u + 2N_m^{N_u} - 2\right)L\right)$	$O\left(N_u\left(2N_m^{N_u} + 2N_m + 1\right)L\right)$
channel decoding	$O\left(N_{cd1}N_m^{N_u}\left(2N_m^{N_u} - 1\right)l\right)$	$O\left(N_uN_{cd1}N_m^{N_u}\left(2N_m^{N_u} - 1\right)l\right)$

The computations of Q functions (20) and (22) for all i correspond to the E-step; the maximization problems (19) and (21) correspond to the M-step. We still employ BP channel decoding to compute the APPs used in the Q functions of SAGE. The framework of message passing on a factor graph is still applicable to SAGE-BP. We describe the key steps on how to transform the SAGE-BP computation onto a factor graph in the following.

SAGE breaks up the EM channel estimation into two sub-problems. Each sub-problem is solved by message passing on a factor subgraph. For the M-step of SAGE, the factor subgraph that solves (19) ((21)) only consists of the channel variables of terminal node A (B): $\{h_{A,i}\}$ ($\{h_{B,i}\}$). The message passing algorithms for computing $\widehat{\mathbf{h}}_A^{(k)}$ and $\widehat{\mathbf{h}}_B^{(k)}$ are initialized with the incoming EM messages $\left\{em_{A,i} \triangleq e^{Q_{A,i}(\cdot)}\right\}$ and $\left\{em_{B,i} \triangleq e^{Q_{B,i}(\cdot)}\right\}$, respectively. It can be proven that Gaussianity is preserved. Namely, substituting (20) into $em_{A,i} = e^{Q_{A,i}(\cdot)}$ leads to the following Gaussian expression:

$$em_{A,i} = \exp\left(Q_{A,i}\left(h_{A,i}\left|\widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}\right.\right)\right)$$
$$\propto \mathcal{CN}\left(h_{A,i} : m_{x_{A,i}^*}y_i - m_{x_{A,i}^* \cdot x_{B,i}}\widehat{h}_{B,i}^{(k-1)}, N_0\right) \quad (23)$$

where $x_{A,i}^*$ is the conjugate of $x_{A,i}$, and $m_{x_{A,i}^*}$, $m_{x_{A,i}^* \cdot x_{B,i}}$ are given by (24) shown at the top of the next page. Based on this Gaussian expression for $\{em_{A,i}\}$ and the Gauss-Markov channel model of $\{h_{A,i}\}$ in (2), we can solve the M-step of the first stage (19) by a Gaussian message passing algorithm that is almost similar to the one derived in Section III for EM, with the difference that the state space here only includes the channel of one terminal node. Thus, all the length-2 channel vectors, in the algorithm, are reduced to scalars. Similarly, we can derive the Gaussian expression for $\{em_{B,i}\}$ and solve the M-step of the second stage (21) by the same Gaussian message passing algorithm. We round off the discussion of SAGE-BP PNC with the following remarks:

1) In (24), $p\left(x_{A,i}\left|\mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right.\right)$ is the decoding result for the transmitted symbol of node A, $x_{A,i}$; $p\left(x_{A,i}^* \cdot x_{B,i}\left|\mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2\right.\right)$ can be regarded as the decoding result for the variable $x_{A,i}^* \cdot x_{B,i}$. Based on these decoding results, the posterior means of $x_{A,i}^*$ and $x_{A,i}^* \cdot x_{B,i}$ are computed according to (24) and used in (23). The intuitive interpretation of the expression for the mean of the Gaussian expression in (23) is that it corresponds to an interference cancelation process for the target channel.

2) Comparing the EM messages in (13) and (23), the message passing implementation of EM-BP needs to compute the mean vector and the covariance matrix of the transmitted symbol pair; that of SAGE-BP needs to compute the mean of the symbol transmitted from the target channel and the cross-correlation between the two symbols $x_{A,i}, x_{B,i}$.

3) We compare the complexities of PNC in the following. Let us just focus on the algorithms within the EM/SAGE-BP iteration loop, since the complexities of initialization and termination are the same for EM-BP and SAGE-BP. The results are summarized in Table I. We denote the number of elements in the channel vector $\mathbf{h}_i$ by N_u. For our TWRC system, $N_u = 2$. We denote the size of the modulation by N_m (e. g., $N_m = 2$ for BPSK and $N_m = 4$ for QPSK). For EM-BP, the computation of (16) needs $O\left(5N_u^3 + 4N_u^2 + 3N_u\right)$ operations.[4] Therefore, the complexity of channel estimation in EM-BP is $O\left(2\left(5N_u^3 + 4N_u^2 + 3N_u\right)L\right)$, where the factor 2 is due to one forward and one backward message passing, L is the frame length. By counting the operations involved in (13), we can figure out that the complexity of computing the EM messages for EM-BP is $O\left(\left(N_u^3 + \left(N_m^{N_u} + 3\right)N_u^2 + \left(2N_m^{N_u} + 1\right)N_u + 2N_m^{N_u} - 2\right)L\right)$. For SAGE-BP, there is no matrix operation, and the computation of (16) becomes $O\left(12\right)$. Since we perform once bidirectional message passing for each node, the complexity of channel estimation in SAGE-BP is $O\left(24N_uL\right)$. The complexity of computing EM messages for SAGE-BP as in (23) is $O\left(N_u\left(2N_m^{N_u} + 2N_m + 1\right)L\right)$. From the above results, we can see that SAGE-BP simplifies the complexity of channel estimation by removing the need for matrix inversions and multiplications. Now, let us look at the channel decoding part. EM-BP updates $\widehat{\mathbf{h}}_A^{(k)}$ and $\widehat{\mathbf{h}}_B^{(k)}$ simultaneously; and requires one virtual channel decoding in each iteration. By contrast, SAGE-BP requires two virtual channel decodings, one before the update of $\widehat{\mathbf{h}}_A^{(k)}$ and one before the update of $\widehat{\mathbf{h}}_B^{(k)}$. That is, each time one of the channels is updated, the APPs of $\mathbf{x}_i$ will need to be re-computed before the other channel is updated. To analyze the complexity of virtual channel decoding, let us consider the regular repeat accumulate (RA) code [36] (used in our simulations). The codeword length is l, and there is no operation of channel decoding on the $L - l$ pilots symbols in each frame. The BP message passing requires $O\left(N_m^{N_u}\left(2N_m^{N_u} - 1\right)\right)$

---

[4]Strictly speaking, the complexities of matrix computations are different for different computing algorithm. Here, we assume that the inversion of an $n \times n$ matrix requires $O\left(n^3\right)$ computations. The product of an $n \times m$ matrix and an $m \times p$ matrix needs $O\left(nmp\right)$ computations for our analysis. Some fast algorithms of matrix computations can reduce these complexities to some extent.

$$m_{x_{A,i}^*} = \sum_{x_{A,i}} \left(\sum_{x_{B,i}} p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right) \right) x_{A,i}^* = \sum_{x_{A,i}} p\left(x_{A,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right) x_{A,i}^*$$

$$m_{x_{A,i}^* \cdot x_{B,i}} = \sum_{x_{A,i}^* \cdot x_{B,i}} \left(\sum_{\mathbf{x}_i : \, x_{A,i}^* \cdot x_{B,i} = x} p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right) \right) x_{A,i}^* \cdot x_{B,i} \qquad (24)$$

$$= \sum_{x_{A,i}^* \cdot x_{B,i}} p\left(x_{A,i}^* \cdot x_{B,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right) x_{A,i}^* \cdot x_{B,i}$$

computations per check node. There are l check nodes on the factor graph of the RA code. We perform N_{cd1} iterations for each round of channel decoding. It follows that the complexity of channel decoding in EM-BP is $O\left(N_{cd1}N_m^{N_u}\left(2N_m^{N_u} - 1\right)l\right)$; the complexity of channel decoding in SAGE-BP is $O\left(N_u N_{cd1}N_m^{N_u}\left(2N_m^{N_u} - 1\right)l\right)$, where the factor $N_u = 2$ is due to the two virtual channel decodings in each SAGE iteration. Therefore, SAGE-BP simplifies channel estimation but adds complexity to channel decoding. One simple method to maintain the same complexity in the channel decodings of SAGE-BP and EM-BP is to make the N_{cd1} in SAGE-BP equal to half of the N_{cd1} in EM-BP. This will cause SAGE-BP to suffer some performance loss. We will study this by simulations in the next section.

4) The decoding results $p\left(x_{A,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right)$, and $p\left(x_{A,i}^* \cdot x_{B,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right)$ are both obtained based on the outputs of BP message passing for virtual channel decoding (see (24)). Another way to do this is to employ parallel interference cancelation (PIC) [37], [38] to compute $p\left(x_{A,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right)$ $\left(p\left(x_{B,i} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right) \right)$ and then the joint probability $p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right)$. Thus, we can still compute the decoding result for $x_{A,i}^* \cdot x_{B,i}$ from $p\left(\mathbf{x}_i \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(k-1)}, \widehat{\mathbf{h}}_B^{(k-1)}, C^2 \right. \right)$. To compute these decoding results, PIC employs two single-user channel decoders for the two terminal nodes. And it requires iterative message passing between the two single-user channel decoders, besides the iterative message passing within each of the channel decoder. Since PIC is well implemented by BP message passing [38], we can also incorporate it into our factor graph framework for SAGE-BP. This method is referred to as SAGE-BP PIC. After the termination of SAGE iteration, SAGE-BP PIC obtains $\left\{ p\left(s_{A,j} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(K)}, \widehat{\mathbf{h}}_B^{(K)}, C^2 \right. \right) \right\}$, $\left\{ p\left(s_{B,j} \left| \mathbf{y}, \widehat{\mathbf{h}}_A^{(K)}, \widehat{\mathbf{h}}_B^{(K)}, C^2 \right. \right) \right\}$ from the final channel decoding. Then, we have, from the rule of sum-product algorithms, $p(s_{A,j}, s_{B,j}|\mathbf{y}, \widehat{\mathbf{h}}_A^{(K)}, \widehat{\mathbf{h}}_B^{(K)}, C^2) \approx p(s_{A,j}|\mathbf{y}, \widehat{\mathbf{h}}_A^{(K)}, \widehat{\mathbf{h}}_B^{(K)}, C^2) \times p(s_{B,j}|\mathbf{y}, \widehat{\mathbf{h}}_A^{(K)}, \widehat{\mathbf{h}}_B^{(K)}, C^2)$ for all j, from which we can perform network coding as in (18). Since PIC employs single-user decoding whose complexity does not increase exponentially with the number of nodes N_u (as virtual channel decoding does), SAGE-BP PIC can make our framework scalable with the number of nodes (if we want to extend our treatment

to beyond TWRC). However, we will see in Section V that the performance of SAGE-BP PIC is not as good as SAGE-BP PNC. The reason is that there is a small cycle between node $x_{A,i}$ and node $x_{B,i}$ for each i in the factor graph of PIC, and these small cycles degrade the performance of the BP algorithm. On the factor graph of virtual channel decoding, we cluster $x_{A,i}$ and $x_{B,i}$ together as one node $\mathbf{x}_i$. Since $x_{A,i}$ and $x_{B,i}$ now become a single variable node, the edges connecting them disappear. By this clustering technique [4], the small cycles between $x_{A,i}$ and $x_{B,i}$ are removed from the factor graph of virtual channel decoding.

5) We comment here that if nodes A and B employ different channel encoders, a corresponding virtual channel decoding does not exist. Then, we can only apply a channel decoding method for MUD systems (such as PIC) to compute the APP of $x_{A,i}$ and the APP of $x_{B,i}$, respectively. By replacing the virtual channel decoding with PIC channel decoding, we can deal with the set-up of different terminal nodes using different channel encoders under the EM/SAGE-BP framework.

6) Since the complexities of both SAGE for channel estimation and BP for PIC channel decoding are scalable with the number of nodes, we can apply SAGE-BP PIC to systems where collisions of more than two signals are possible [39], [40]. The study of this application awaits future work.

V. SIMULATION RESULTS

In this section, we perform computer simulations to evaluate the performances of the proposed schemes. We assume the channels of both terminal nodes have the same average power $\sigma_A^2 = \sigma_B^2$. Unless stated otherwise, the channel correlation coefficient α is set to 0.99. The regular RA code with coding rate $1/3$ is employed. In the case of BPSK modulation, each frame has 1024 information bits (thus, 3072 coded modulated data symbols); and in the case of QPSK modulation, each frame has 2048 bits (also 3072 coded modulated data symbols). We insert two pilots every Δ data symbols. Unless stated otherwise, the pilot interval Δ is set to 16 (this corresponds to a 11.1% pilot load). The two terminal nodes adopt orthogonal pilots, wherein $P_1 = 1, P_2 = 1$ for node A and $P_1 = 1, P_2 = -1$ for node B. All simulation results presented here are obtained by averaging over 10^5 pairs of frames. The signal to noise ratio (SNR) is defined as E_s/N_0 where E_s is the energy per coded bit. Specifically, for coding rate $1/3$, $E_s = E_b/3$ where E_b is energy per source bit.

A. Performance of EM-BP PNC

First, we investigate the performance of the EM-BP PNC receiver. We evaluate the BER of the network-coded messages and the mean square error (MSE) of the estimated channels. The results of the PNC receiver using just a one-shot MMSE channel estimation (this is equivalent to our EM-BP PNC receiver with $K = 0$) and the ideal PNC receiver with the full channel state information (CSI) will be given as benchmarks.

Fig. 4 presents the BER results of the three receivers: the EM-BP PNC, the MMSE PNC, and the Full-CSI PNC; and Fig. 5 presents their MSE results for the estimated channel. The BPSK modulation is used for all receivers. The number of channel decoding iterations in the MMSE PNC receiver and the Full-CSI PNC receiver is denoted by N_{cd}. Recall that for the EM-BP PNC, K is the number of EM iterations, N_{cd1} is the number of iterations for the virtual channel decoding in the E-step of each EM iteration; and N_{cd2} is the number of iterations in the final virtual channel decoding at the conclusion of all EM iterations. Since we use the MMSE channel estimation to initialize the EM-BP algorithm, we expect the EM-BP algorithm to give more accurate channel estimation than the one-shot MMSE channel estimation. This is confirmed by our simulation results in Fig. 5. In particular, the channel estimation accuracy in EM-BP improves progressively with the number of iterations. We can also observe that the first EM iteration can already extract most of the gain in MSE. Our simulations also indicates that the EM-BP PNC algorithm has almost converged after $K = 5$ iterations. These MSE improvements by EM-BP PNC are reflected into BER results. Comparing the BER results of EM-BP PNC receiver and MMSE PNC with $N_{cd} = 6$ in Fig. 4, we can see that there is a 4 dB gain by EM-BP PNC just after the first EM iteration ($K = 1$). There is a 6 dB gain after EM has converged ($K = 5$). Furthermore, the BER result of EM-BP PNC at $K = 5$ can approach the BER of the Full-CSI PNC very well.

For a fairer comparison, let us examine the performance of EM-BP PNC with $N_{cd1} = N_{cd2} = 6$, $K = 5$, and the performance of MMSE PNC receiver with $N_{cd} = 36$: i.e., the total numbers of channel decoding iterations are the same in the two cases. We observe that for BER, EM-BP PNC receiver has around 4 dB gain over MMSE PNC. The observed error floor in both BER and MSE as SNR increases are due to the time-varying property of the channel, which is also analyzed and reported in [41]. Essentially, even if the receiver noise is zero, the channel randomness in between pilots induces uncertainty that cannot be removed entirely regardless of the SNR. Effectively, the channel randomness in between pilots is a source of noise besides the thermal circuit noise in the receiver.

Fig. 6 and Fig. 7 present the BER and MSE results of EM-BP PNC when QPSK modulation is used. The results of EM-BP PNC with BPSK are also shown as benchmarks. For the data here, $N_{cd1} = N_{cd2} = 6$. We can observe that the BER and MSE results for QPSK are slightly worse than that for BPSK. The denser constellation map in QPSK makes the channel estimation tougher. Thus, the MSEs for QPSK are larger than for BPSK. Moreover, the denser constellation

makes the channel decoding more sensitive to the channel estimation error. The performance gap in the BER results of the BPSK and QPSK is about 0.5-1 dB in the low SNR regime ($E_s/N_0 < 6$ bB), and about 2 dB in the high SNR regime ($E_s/N_0 > 6$ bB). Since the performance trends of BPSK and QPSK are the same, we just focus on the performance of BPSK hereinafter for simplicity.

We now investigate the impact of channel correlation coefficient α on the BER performance of EM-BP PNC. For virtual channel decoding, we also set $N_{cd1} = N_{cd2} = 6$. The BER results of EM-BP PNC versus pilot interval Δ for various α are shown in Fig. 8. The operating SNR E_s/N_0 is 6 dB. As α decreases from 0.99 to 0.97, the channel varies faster and the BER gets worse. The BER is more sensitive to the pilot load as α gets smaller. When the pilot interval Δ goes from 2 to 32, the BER in a channel with $\alpha = 0.99$ ranges from 10^{-6} to 10^{-5}; however, the BER in a channel with $\alpha = 0.97$ ranges from 10^{-6} to 10^{-1}. To maintain the performance in an environment of rapidly varying channel, we can insert more pilot symbols or increase the number of EM iterations (K). For example, the BER of $\alpha = 0.97$ and $K = 5$ can approach the BER of $\alpha = 0.99$ and $K = 1$ when $\Delta \leq 16$.

In Section III, we derived the proposed message passing algorithm for channel estimation using the first-order Gauss-Markov channel model. Indeed, the Gauss-Markov channel model is an approximation of the real physical channel. Here, we conduct a simulation using a more realistic mobile channel. Our aim is to demonstrate the robustness of the channel-estimation message passing when there is a mismatch between the channel model and the actual realized physical channel. All the set-ups of the simulation are the same as the previous cases except that the actual channel gains are generated according to the Clarke's channel model [42]. The normalized maximum Doppler spread in the Clarke's channel model is set to 0.005. Following the relation between the Doppler spread and the correlation coefficient α established in [43], we set the correlation coefficient to $\alpha = 0.989$ and use it for channel-estimation message passing. We evaluate the MSEs of the channel estimation outputs, and present the results in Fig. 9. We can see that despite the channel mismatch, our channel estimation message passing still works well. We can still observe its MSE gains over the one-shot MMSE channel estimation. Comparing the MSE results of the Charke's channel in Fig. 9 with the MSE results of the Gauss-Markov channel in Fig. 5, we can see that there is no obvious performance difference between them. For the Clarke's channel, besides the MSE improvement, the EM-BP PNC receiver can also yield BER improvement over the one-shot MMSE channel estimation method. We omit the BER results here to conserve space.

B. Performance of SAGE-BP PNC

We now investigate the performance of the SAGE-BP PNC receiver. We first focus on SAGE-BP PNC with $N_{cd1} = N_{cd2} = 6$. The BER results are shown in Fig. 10, where the BER results of EM-BP PNC with $N_{cd1} = N_{cd2} = 6$ are also presented as benchmarks. It can be observed that SAGE-BP PNC has the same performance as EM-BP PNC. Both SAGE-BP PNC and EM-BP PNC converge to the same performance
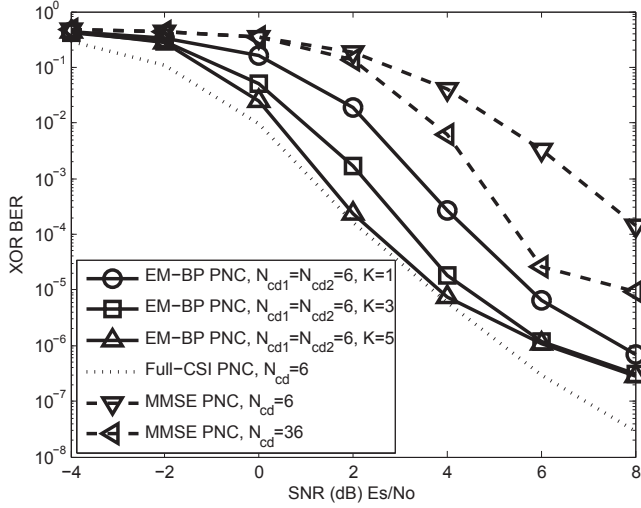
Fig. 4. The BER results of the EM-BP PNC receiver with BPSK modulation.
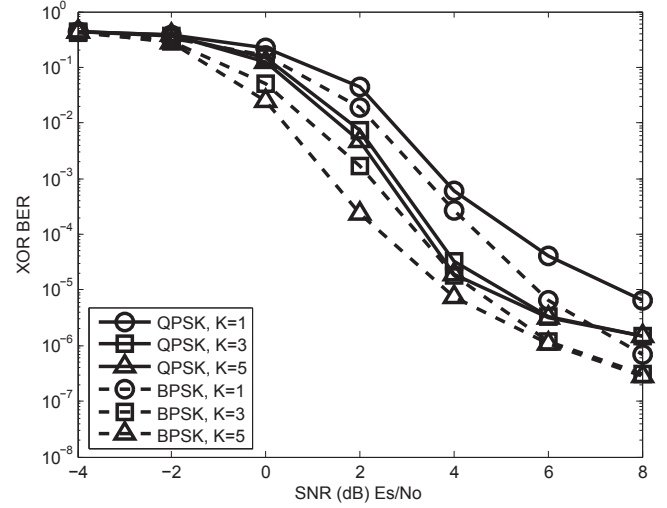

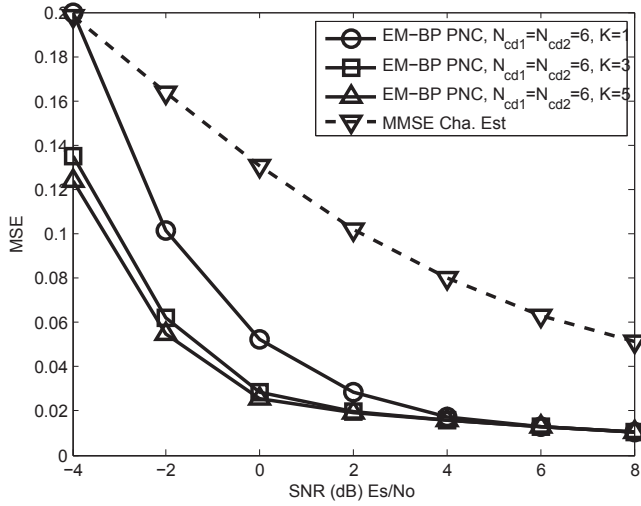Fig. 6. The BER results of the EM-BP PNC receiver with QPSK modulation.

Fig. 5. The MSE results for the estimated channels of the EM-BP PNC receiver with BPSK modulation.
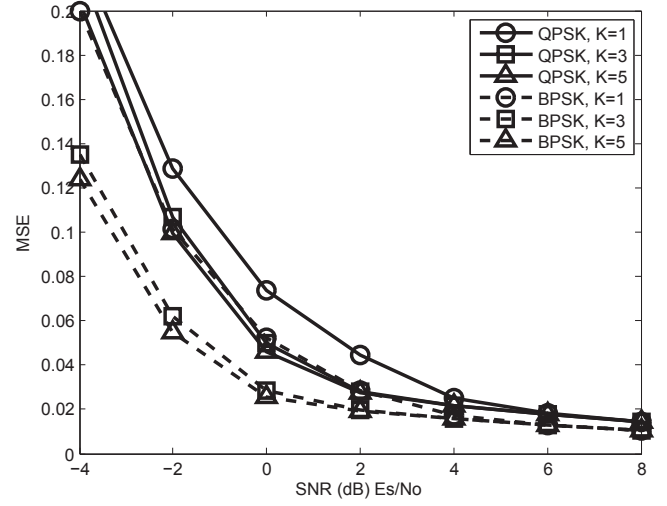

Fig. 7. The MSE results for the estimated channels of the EM-BP PNC receiver with QPSK modulation.

after $K = 5$ iterations. This is consistent with the fundamental theory of SAGE [13]. Substituting the simulation parameters $(N_{cd1} = 6, N_u = 2, N_m = 2, l = 8L/9)$ into Table I, we can find that per EM-BP iteration needs $O(333L)$ computations; per SAGE-BP iteration needs $O(369L)$ computations. The complexity of SAGE-BP now is slightly higher than that of EM-BP due to the more complex channel decoding in SAGE-BP.

For SAGE-BP PNC, since the two-dimension estimation problem of $\mathbf{h} = \{\mathbf{h}_A, \mathbf{h}_B\}$ in the M-step of EM has been decomposed into two one-dimension estimation sub-problems of $\mathbf{h}_A$ and $\mathbf{h}_B$, the complexity of its channel estimation is smaller than that of EM-BP PNC. To 'equalize' the complex-ities of the channel decoding of SAGE-BP PNC and EM-BP PNC, we set $N_{cd1} = 3, N_{cd2} = 6$ for SAGE-BP PNC, i.e., the total number of iterations for BP channel decoding in the E-step after both $\mathbf{h}_A$ and $\mathbf{h}_B$ are updated is $2 \times N_{cd1} = 6$; and $N_{cd1} = 6, N_{cd2} = 6$ for EM-BP PNC. Recall that the complexity of channel estimation in SAGE-BP PNC is smaller than that in EM-BP PNC (see the complexities in Table I).

Since we equalize the complexities of the channel decoding processes of EM-BP and SAGE-BP (by halving the number of channel decoding iterations in each SAGE-BP iteration), the overall complexity of SAGE-BP PNC (per SAGE-BP iteration needs $O(219L)$ computations) is now smaller than that of EM-BP PNC because of its less complex channel estimation. From the results in Fig. 10, we can see that the performance of SAGE-BP PNC with $N_{cd1} = 3, N_{cd2} = 6$ is not as good as that of EM-BP PNC with $N_{cd1} = 6, N_{cd2} = 6$.

We next compare the performances of SAGE-BP PIC and SAGE-BP PNC. We report the BER and MSE results in Fig. 11 and Fig. 12, respectively. We denote the number of iterations for the message passing between the two single-user channel decoders in PIC by P. For SAGE-BP PIC, we can increase P and N_{cd1} (N_{cd2}) to allow convergence of the PIC channel decoding. From our simulation results, PIC converges with $N_{cd1} = N_{cd2} = 18, P = 2$. When we continue to increase these numbers of iterations, no observable improvement on performance can be obtained. For SAGE-BP PNC, we can increase N_{cd1} (N_{cd2}) to make the virtual channel
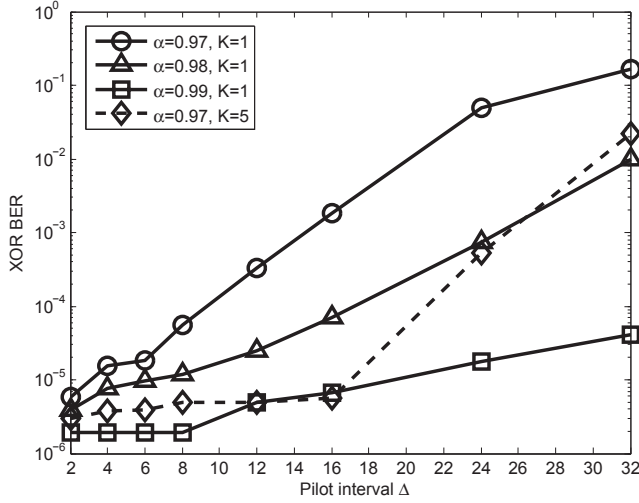
Fig. 8. The impact of the channel correlation coefficient α on the BER performance of the EM-BP PNC receiver.
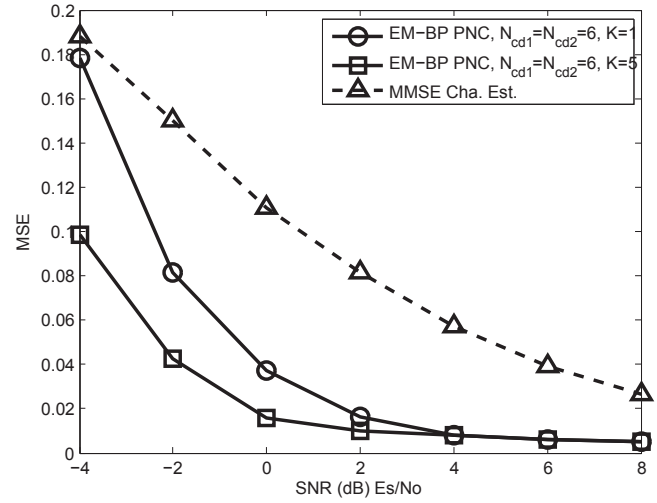

Fig. 9. The MSE results for the estimated channels of the EM-BP PNC receiver under the Clarke's channel (with normalized maximum Doppler spread 0.005).

decoding converge. Virtual channel decoding converges with $N_{cd1} = N_{cd2} = 18$. From the BER results in Fig. 11, we can see that the BER of SAGE-BP PIC is not as good as that of SAGE-BP PNC. Specifically, the error floor of SAGE-BP PIC in the high SNR regime is higher than that of SAGE-BP PNC. As explained earlier, this is because there are small cycles in the factor graph of PIC. The APPs of transmitted symbols computed by PIC is not as accurate as the ones computed by virtual channel decoding for PNC. Therefore, we expect the SAGE channel estimation based on the worse APPs will result in worse estimate results. From the MSE results in Fig. 12, we observe that with one SAGE iteration ($K = 1$), SAGE-BP PIC indeed has worse MSE than SAGE-BP PNC. When we increase the number of iterations to allow SAGE-BP PNC and SAGE-BP PIC to converge ($N_{cd1} = N_{cd2} = 18$, $P = 2$, $K = 5$), SAGE-BP PIC can obtain the same MSE as SAGE-BP PNC. However, even with the same MSE, the final channel decoding results of SAGE-BP PIC now is still worse than that of SAGE-BP PNC (see the BERs in Fig. 11). This is because the APPs computed by the PIC process are still not as accurate as those computed by the virtual channel decoding process. In other words, as far as the channel estimation is concerned, the performances of both schemes are comparable, but the virtual channel decoding in SAGE-BP PNC gives better estimates of the network-coded symbols.

C. Comparison with Other Receiver Architectures

In this subsection, we compare our EM-BP PNC receiver with other receiver architectures [10], [21], [22], [29].

We first investigate the performances of two different strategies on how to combine EM with BP channel decoding: our EM-BP strategy and the strategy suggested in [21], [22]. Although [21], [22] do not investigate PNC systems, we can extrapolate their strategy for PNC application. Before we compare the performance results of the two strategies, let us explain a subtle point in the rigorous EM-BP framework because this is where the strategy in [21], [22] deviates from this framework. We note that the extrinsic information
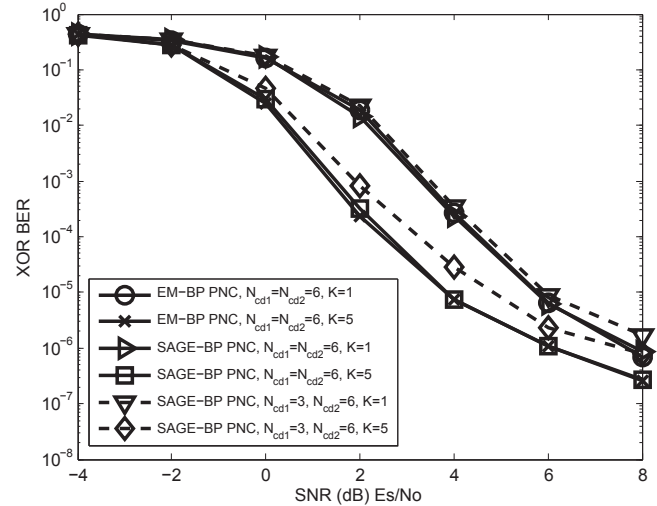
Fig. 10. The BER results of the SAGE-BP PNC receiver.

used in (14) is a function of the channel estimate $\widehat{\mathbf{h}}^{(k)}$. According to the theoretical framework of EM, upon a new channel estimate, we should immediately run BP channel decoding to compute new extrinsic information to update the APPs. The new APPs are then used to compute the next channel estimate. Our EM-BP strategy conforms to the above operational sequence. In the following discussion, when we refer to the strategy in [21], [22], we mean the strategy as applied to PNC. In the strategy of [21], [22], each iteration of BP channel decoding is followed by several EM iterations for channel estimation using the same extrinsic information from that single iteration of BP channel decoding. For example, at EM iteration $k = k_1$, $\widehat{\mathbf{h}}^{(k_1-1)}$ was obtained. After one iteration of BP channel decoding, we obtain the new extrinsic information $p\left(\mathbf{x}_i \big| \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2\right)$ for all i. For several subsequent EM iterations for channel estimation, indexed by $k = k_1 + 1, \cdots, k_2 - 1$, BP channel decoding will not be performed at all. EM updates a new channel estimate $\widehat{\mathbf{h}}^{(k)}$ in each iteration using an approximate APP of $\mathbf{x}_i$ rather than
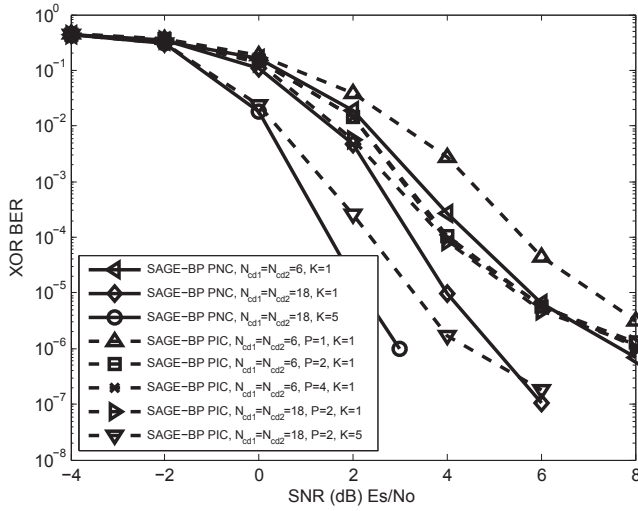
Fig. 11. The BER results of the SAGE-BP PIC/PNC receiver.

Fig. 12. The MSE results of the SAGE-BP PIC/PNC receiver.

(14). Specifically,

$$p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right) \approx$$
$$A \cdot p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2\right) p\left(y_i \,\middle|\, \mathbf{x}_i, \widehat{\mathbf{h}}^{(k-1)}\right).$$

In particular, an approximation is made on the extrinsic information:

$$p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k-1)}, C^2\right)$$
$$\approx p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2\right).$$

Obviously, this operation dose not correctly compute the APP of $\mathbf{x}_i$ needed by EM for iterations $k = k_1 + 1, \cdots, k_2 - 1$. Moreover, if there are cycles in the factor graph of the channel code (e. g. the RA code used in our simulation experiments), the computation of $p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}_{1:i-1}, \mathbf{y}_{i+1:L}, \widehat{\mathbf{h}}^{(k_1-1)}, C^2\right)$ with just one BP channel decoding iteration will not accurate either. This strategy is referred to as SP-EM in [21], where SP stands for sum-product. To avoid confusion, we rename it as multiple-EM-single-BP scheme here to represent the fact that there are multiple iterations of EM for each iteration of BP. Our EM-BP is single-EM-multiple-BP in that sense.

The BER results of EM-BP PNC and multiple-EM-single-BP PNC are shown in Fig. 13; their MSE results are shown in Fig. 14. For EM-BP PNC, there are N_{cd1} iterations for BP channel decoding after each EM iteration; and totally K EM iterations are performed (therefore, altogether there are K EM iterations and $N_{cd1}K$ BP channel decoding iterations). For multiple-EM-single-BP PNC PNC, there are K EM iterations after each for BP channel decoding iteration; and totally N iterations for BP channel decoding are performed prior to termination (therefore, altogether there are NK EM iterations and N BP channel decoding iterations). After termination, we perform additional $N_{cd2} = 18$ iterations for BP channel decoding in both strategies. The approximate APPs in multiple-EM-single-BP as explained above causes it to deviate from the principle of EM algorithm. The results in Fig. 13 and Fig. 14 show that compared with EM-BP PNC, multiple-EM-single-BP PNC exhibits worse performances and converges to an inferior operating point. From our simulation results
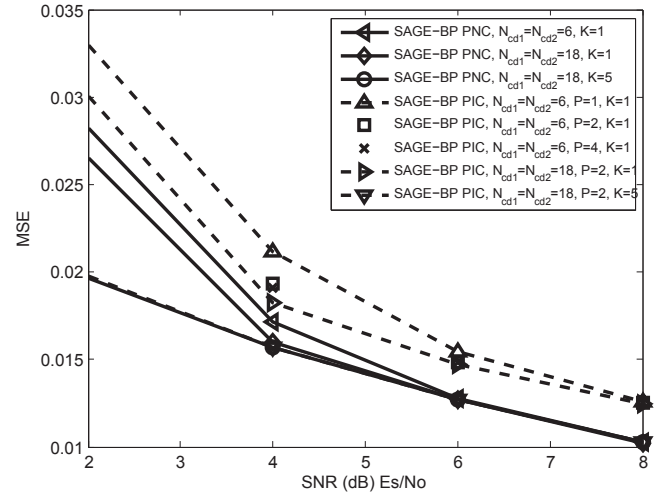
in Fig. 13 and Fig. 14, we see that no further improvement on the performance of multiple-EM-single-BP PNC can be obtained when N is increased from 5 to 7. In other words, even if we make N very large to equalize the channel decoding complexity of multiple-EM-single-BP PNC with that of EM-BP PNC, the performance of multiple-EM-single-BP PNC will still be worse than that of EM-BP PNC.

Next, we compare our EM-BP PNC receiver and the BP-MM receiver proposed in [10] (see Section I for the overview of [10]). In EM-BP PNC, EM is employed to accomplish the task of channel estimation; and BP is employed to accomplish the task of channel decoding. EM can find $\widehat{\mathbf{h}}_{MAP}$ when it converges to the global optimal, and in this case the result of the final channel decoding in EM-BP PNC is $p\left(\mathbf{x}_i \,\middle|\, \widehat{\mathbf{h}}_{MAP}, \mathbf{y}, C^2\right)$ for all i. This is the target of EM-BP. For EM, the convergence to the global optimal can always be guaranteed by a good initial point [34]. We can also finish the tasks of both channel estimation and channel decoding using BP alone, as in [10]. In this case, the final target of channel decoding is $p\left(\mathbf{x}_i \,\middle|\, \mathbf{y}, C^2\right) = \int p\left(\mathbf{x}_i, \mathbf{h} \,\middle|\, \mathbf{y}, C^2\right) d\mathbf{h}$, which is different from the target of EM-BP. However, the integration over continues variables required by BP channel estimation is computationally infeasible. BP-MM uses MM as an approximation to circumvent the need for explicit integration (i.e., integration of the approximate Gaussian distribution can be obtained in closed form) [10]. However, as a consequence of the approximation by MM, the optimality of BP cannot be guaranteed (i.e., even if the algorithm converges to the global optimal, it will not be the global optimal associated with the original non-Gaussian distribution). The BER results of BP-MM are shown in Fig. 15, where M denotes the number of iterations between channel estimation and channel decoding in BP-MM; and N_{cd} denotes the number of channel decoding iterations in BP-MM. Comparing the BERs of EM-BP PNC and BP-MM, EM-BP PNC is worse in the low SNR regime and it is better in the high SNR regime. Specifically, when both receivers have converged ($M = K = 5$, $N_{cd} = N_{cd1} = N_{cd2} = 18$), EM-BP PNC outperforms BP-MM by 1dB at the BER of 10^{-6}. We believe this gap could be due to the
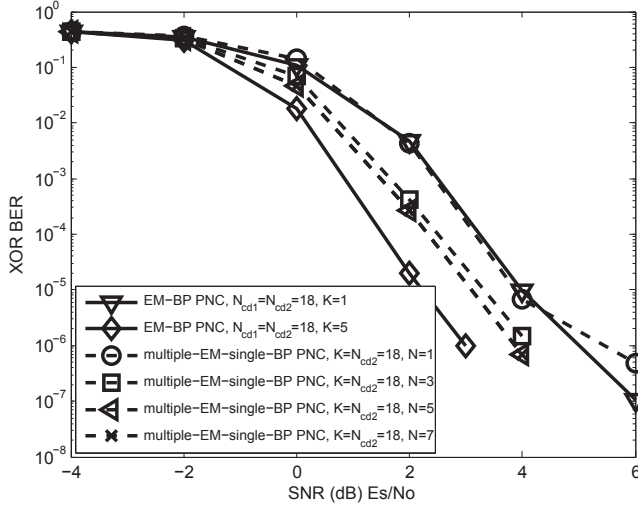
Fig. 13.   Comparing the different strategies on how to combine EM with BP channel decoding: the BER results.

Fig. 15.   Comparing the proposed EM-BP PNC receiver with other receiver architectures.
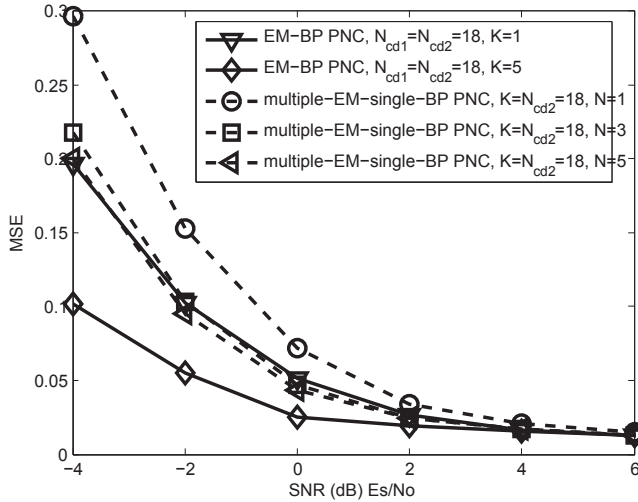

Fig. 14.   Comparing the different strategies on how to combine EM with BP channel decoding: the MSE results

approximation by MM.

As summarized in Section I, there is an EM approach for joint channel estimation and channel decoding in multi-user CDMA systems, proposed in [29]. In [29], channel decoding is implemented by MMSE SIC with separate single-user channel decoders for data from different users; and channel estimation is implemented by EM using APPs obtained from the single-user channel decoders. We compare EM-BP PNC with an EM SIC receiver modified from that in [29]. In the modified EM SIC receiver, there is no despreading operation, since we focus on PNC systems rather than CDMA systems. Specifically, given the channel estimate $\widehat{\mathbf{h}}^{(k-1)}$ from the last EM iteration, we directly employ MMSE SIC to compute the APP of the ith symbol transmitted by a individual node: $p\left(x_{A,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$, $p\left(x_{B,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$, for all i. MMSE SIC [38] is a technique in which the MMSE estimation on the signal of one node is subtracted from the received signal; then channel decoding is performed on the remaining signal using the standard sum-product algo-
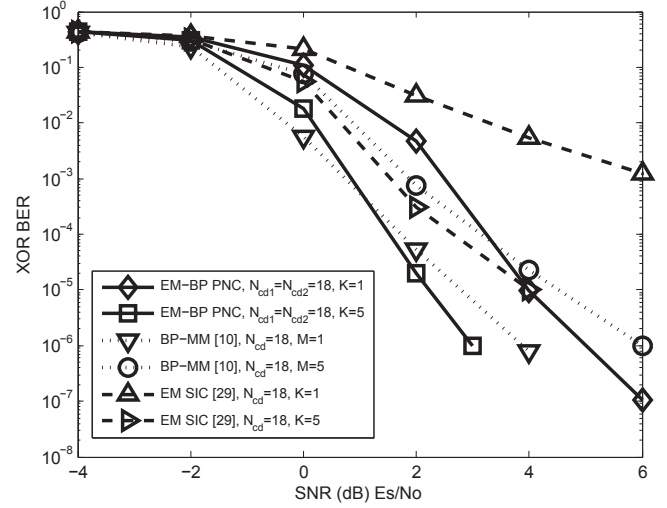
rithm to obtain the APP of the other node. Based on these APPs, the channel estimation is implemented by the EM message passing as proposed in our paper here. After the final round of channel decoding, we obtain $\{\widehat{s}_{A,j}\}$, $\{\widehat{s}_{B,j}\}$ from the two single-user channel decoders. We then perform network coding as $\{\widehat{s}_{A,j}\oplus\widehat{s}_{B,j}\}$. Compared with EM-BP PNC, there are two drawbacks to this EM SIC receiver: 1) the APPs $p\left(x_{A,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$, $p\left(x_{B,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$, are only approximately computed by MMSE SIC (even without considering the cycles in the factor graph of channel coding) because of the use of the aforementioned MMSE signal cancelation rather than the use of the strict sum-product formalism to link the computations of $p\left(x_{A,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$ and $p\left(x_{B,i}\left|\mathbf{y},\widehat{\mathbf{h}}^{(k-1)},C\right.\right)$ together; and 2) its single-user channel decoding is not optimal for PNC. The BER results of EM SIC are shown in Fig. 15. when both receivers have converged ($K = 5$, $N_{cd} = N_{cd1} = N_{cd2} = 18$), we see that EM-BP PNC outperforms EM SIC by around 2dB at BER 10^{-5}.

VI. CONCLUSION

We have proposed an EM-BP factor-graph framework tailored for solving the problem of joint channel estimation and channel decoding in PNC systems. This framework consists of EM message passing for channel estimation and BP message passing for channel decoding. The output of one forms the input of the other, and vice versa, so that the results of channel estimation and channel decoding can be refined in an iterative and progressive manner. A salient feature of our framework is the use of 'virtual channel decoding' to ensure optimal performance for PNC systems. Furthermore, we show that the EM messages in our factor-graph framework are Gaussian messages that can be characterized by their means and variances only, and this greatly reduces computation complexity. We refer to the receiver based on this framework as EM-BP PNC.

Our simulation results indicate that the BER of EM-BP PNC can approach that of an ideal PNC receiver with perfect CSI.

In addition, EM-BP PNC outperforms other receivers in terms of BER and MSE. Beyond PNC, we believe the EM-BP factor graph framework proposed in this work can also be used to construct receivers with superior performance in conventional single-user and multi-user systems.

REFERENCES

[1] S. Zhang, S. Liew, and P. Lam, "Hot topic: physical-layer network coding," in *Proc. 2006 ACM Mobicom*, pp. 358–365.

[2] S. Liew, S. Zhang, and L. Lu, "Physical-layer network coding: tutorial, survey, and beyond," *Physical Commun.*, vol. 6, pp. 4–42, Mar. 2013.

[3] S. Zhang and S. Liew, "Channel coding and decoding in a relay system operated with physical-layer network coding," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 788–796, May 2009.

[4] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[5] H. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, June 2007.

[6] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring Artificial Intelligence in the New Millennium*, vol. 8, pp. 236–239, 2003.

[7] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[8] D. Wubben and Y. Lang, "Generalized sum-product algorithm for joint channel decoding and physical-layer network coding in two-way relay systems," in *Proc. 2010 IEEE Glob. Telecomm. Conf.*, pp. 1–5.

[9] L. Lu and S. Liew, "Asynchronous physical-layer network coding," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 819–831, Feb. 2012.

[10] F. Lehmann, "Joint channel estimation and decoding for trellis-coded MIMO two-way relay networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 8, pp. 1455–1468, Aug. 2013.

[11] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[12] M. Gupta and Y. Chen, *Theory and Use of the EM Algorithm*. Now Pub, 2011, vol. 3.

[13] J. A. Fessler and A. O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Process.*, vol. 42, no. 10, pp. 2664–2677, Oct. 1994.

[14] J. Dauwels, S. Korl, and H. Loeliger, "Expectation maximization as message passing," in *Proc. 2005 IEEE Int. Symp. on Inf. Theory*, pp. 583–586.

[15] J. Dauwels, A. Eckford, S. Korl, and H. Loeliger, "Expectation maximization as message passing–part 1: principles and Gaussian messages," Arxiv preprint arXiv:0910.2832, pp. 1–14, 2009.

[16] C. Georghiades and J. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Commun.*, vol. 45, no. 3, pp. 300–308, Mar. 1997.

[17] E. Chiavaccini and G. Vitetta, "MAP symbol estimation on frequency-flat Rayleigh fading channels via a Bayesian EM algorithm," in *Proc. 2001 IEEE Int. Conf. Comm.*, pp. 1057–1061.

[18] C. Cozzo and B. Hughes, "Joint channel estimation and data detection in space-time communications," *IEEE Trans. Commun.*, vol. 51, no. 8, pp. 1266–1270, Aug. 2003.

[19] 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. Netw.*, vol. 2005, no. 2, pp. 117–129, 1900.

[20] D. Torrieri, E. Ustunel, S. Min, and D. Kang, "Iterative CDMA receiver with EM channel estimation and turbo decoding," in *Proc. 2006 IEEE Military Communications Conference*, pp. 1–6.

[21] C. Herzet, V. Ramon, and L. Vandendorpe, "A theoretical framework for iterative synchronization based on the sum–product and the expectation-maximization algorithms," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1644–1658, May 2007.

[22] Q. Guo and D. Huang, "EM-based joint channel estimation and detection for frequency selective channels using Gaussian message passing," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 4030–4035, Aug. 2011.

[23] J. Ylioinas and M. Juntti, "Iterative joint detection, decoding, and channel estimation in turbo-coded MIMO-OFDM," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1784–1796, Apr. 2009.

[24] E. Panayirci, H. Senol, and H. Poor, "Joint channel estimation, equalization, and data detection for OFDM systems in the presence of very high mobility," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4225–4238, Aug. 2010.

[25] U. Fawer and B. Aazhang, "A multiuser receiver for code division multiple access communications over multipath channels," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 1556–1565, Feb./Mar./Apr. 1995.

[26] A. Kocian and B. Fleury, "EM-based joint data detection and channel estimation of DS-CDMA signals," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1709–1720, Oct. 2003.

[27] S. Wu, U. Mitra, and C. Kuo, "Iterative joint channel estimation and multiuser detection for DS-CDMA in frequency-selective fading channels," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3261–3277, July 2008.

[28] A. Assra, W. Hamouda, and A. Youssef, "EM-based joint channel estimation and data detection for MIMO-CDMA systems," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1205–1216, Mar. 2010.

[29] M. Kopbayashi, J. Boutros, and G. Caire, "Successive interference cancellation with SISO decoding and EM channel estimation," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 8, pp. 1450–1460, Aug. 2001.

[30] A. Kocian, I. Land, and B. H. Fleury, "Joint channel estimation, partial successive interference cancellation, and data decoding for DS-CDMA based on the SAGE algorithm," *IEEE Trans. Commun.*, vol. 55, no. 6, pp. 1231–1241, June 2007.

[31] M.-O. Pun, M. Morelli, and C.-C. J. Kuo, "Iterative detection and frequency synchronization for OFDMA uplink transmissions," *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 629–639, Feb. 2007.

[32] C. Tan and N. Beaulieu, "On first-order Markov modeling for the Rayleigh fading channel," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2032–2040, Dec. 2000.

[33] T. Wang and S. C. Liew, "Joint channel estimation and channel decoding in physical-layer network coding systems: an EM-BP factor graph framework." Available: http://arxiv.org/abs/1307.4612, Tech. Rep., 2013.

[34] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[35] M. Biguesh and A. Gershman, "Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, Mar. 2006.

[36] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for turbo-like codes," in *Proc. 1998 Allerton Conf. on Communication, Control and Computing*, pp. 201–210.

[37] X. Wang and V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, July 1999.

[38] J. Boutros and G. Caire, "Iterative multiuser joint decoding: unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, July 2002.

[39] G. Cocco, C. Ibars, D. Gunduz, and O. del Rio Herrero, "Collision resolution in slotted Aloha with multi-user physical-layer network coding," in *Proc. 2011 IEEE Veh. Tech. Conf.*, pp. 1–4.

[40] J. Goseling, M. Gastpar, and J. H. Weber, "Physical-layer network coding on the random access channel," in *Proc. 2013 IEEE Int. Symp. on Inf. Theory*, pp. 2339–2343.

[41] Y. Zhu, D. Guo, and M. Honig, "A message-passing approach for joint channel estimation, interference mitigation, and decoding," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 6008–6018, Dec. 2009.

[42] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[43] M. Medard, "The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the channel," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 933–946, Mar. 2000.

Taotao Wang received his B.S. Degree in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2008, and received his M.S. Degree in information and signal processing from Beijing University of Posts and Telecommunications, Beijing, China, in 2011. From Oct. 2010 to Jul. 2011, he was working as a research assistant with the Institute of Network Coding at the Chinese University of Hong Kong, Hong Kong. Now he is a Ph.D. student at the Department of Information Engineering of the Chinese University of Hong Kong. His main research interests are wireless communications and networking, statistical signal processing. His most recent research focus is on the application of network coding at the physical-layer of wireless communication systems. He is a recipient of the Hong Kong Ph.D. Fellowship.

Soung Chang Liew received his S.B., S.M., E.E., and Ph.D. degrees from the Massachusetts Institute of Technology. From 1984 to 1988, he was at the MIT Laboratory for Information and Decision Systems, where he investigated Fiber-Optic Communications Networks. From March 1988 to July 1993, he was at Bellcore (now Telcordia), New Jersey, where he engaged in Broadband Network Research. He has been a Professor at the Department of Information Engineering, the Chinese University of Hong Kong, since 1993. He is an Adjunct Professor at Peking University and Southeast University, China.

Prof. Liew's research interests include wireless networks, Internet protocols, multimedia communications, and packet switch design. Prof. Liew's research group won the best paper awards in IEEE MASS 2004 and IEEE WLN 2004. Separately, TCP Veno, a version of TCP to improve its performance over wireless networks proposed by Prof. Liew's research group, has been incorporated into a recent release of Linux OS. In addition, Prof. Liew initiated and built the first inter-university ATM network testbed in Hong Kong in 1993. More recently, Prof. Liew's research group pioneers the concept of Physical-layer Network Coding (PNC).

Besides academic activities, Prof. Liew is active in the industry. He co-founded two technology start-ups in Internet Software and has been serving as a consultant to many companies and industrial organizations. He is currently consultant for the Hong Kong Applied Science and Technology Research Institute (ASTRI), providing technical advice as well as helping to formulate R&D directions and strategies in the areas of Wireless Internet working, Applications, and Services.

Prof. Liew is the holder of nine U.S. patents and a Fellow of IEEE, IET and HKIE. He currently serves as Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and Ad Hoc and Sensor Wireless Networks. He is the recipient of the first Vice-Chancellor Exemplary Teaching Award in 2000 and the Research Excellence Award in 2013 at the Chinese University of Hong Kong. Publications of Prof. Liew can be found in www.ie.cuhk.edu.hk/soung.