Joint Estimation and Decoding for Sparse Channels via Relaxed Belief Propagation

Philip Schniter
Dept. of ECE, The Ohio State University, Columbus, OH 43210. (Email: schniter@ece.osu.edu)

***Abstract*—We consider the problem of communicating efficiently over an N-block Rayleigh-fading channel with a K-sparse L-length discrete-time impulse response (with $0<K<L<N$), where both the channel's coefficients and support are *unknown* to both the transmitter and receiver. For this, we propose to use bit-interleaved coded orthogonal frequency division multiplexing (OFDM) in conjunction with a novel belief-propagation (BP) based demodulation scheme. Unlike the conventional "decoupled" approach, where pilots are used to estimate the sparse channel, and the resulting channel estimate is used for data decoding, we perform *joint* sparse-channel estimation and data decoding. Simulations suggest that the proposed scheme performs remarkably well at both low and high-SNR—achieving, e.g., the channel capacity prelog factor—with reasonable complexity: $\mathcal{O}(NL)$.[1]**

I. Introduction

We consider the problem of communicating efficiently over an N-block Rayleigh-fading channel with a K-sparse discrete-time impulse response of length L (with $0<K<L<N$), when both channel coefficients and support are *unknown* to the transmitter and receiver. It was recently shown that, at high SNR, the ergodic capacity of this channel obeys [1]

$$C(\mathsf{SNR}) \;=\; \tfrac{N-K}{N}\log_2(\mathsf{SNR})+\mathcal{O}(1), \qquad (1)$$

where SNR denotes received signal-to-noise ratio. Interestingly, the prelog factor $\frac{N-K}{N}$ depends on the channel sparsity K and not the channel length L, even though the locations of the non-zero channel coefficients are unknown.

Pilot-aided transmission has emerged as a practical means of communicating over unknown channels, and a large body of work (see, e.g., the bibliography in [2]) has grown around the idea of leveraging sparsity to reduce the number of pilots, with the end goal of increasing spectral efficiency. The conventional approach to demodulation, which we refer to as "*compressed channel sensing*" (CSS) after [2], is a *decoupled* one: pilot knowledge is exploited for sparse-channel estimation, and the resulting channel estimate is used for data decoding. Based on provable performance guarantees from the theory of compressed sensing, it is widely accepted that $\mathcal{O}\big(K\,\mathrm{polylog}(L)\big)$ pilots are both necessary and sufficient for CCS [2].

From the capacity expression (1), we can see that any scheme allocating more than K degrees of freedom (per fading block) to pilots will be spectrally inefficient in the high-SNR regime. Thus, the CCS approach, which requires $\mathcal{O}\big(K\,\mathrm{polylog}(L)\big)>K$ pilots, falls short of its goal. The key question, then, is whether there exists a practical[2] scheme that achieves the capacity prelog factor in (1). In this paper, we propose a scheme that, empirically, appears to meet this goal.

The transmission scheme that we assume is a conventional one: bit-interleaved coded modulation (BICM) is combined with orthogonal frequency division multiplexing (OFDM) and a few carefully placed training bits. The proposed demodulation scheme is, to our knowledge, novel: we perform *joint* sparse-channel estimation and data decoding using *belief-propagation* (BP), exploiting recent advances in approximate BP [3], [4]. Simulations, using $N=1021$ subcarriers with channels of length $L=256$ and average sparsity $\mathrm{E}\{K\}=64$, suggest that our scheme behaves near-optimally in both low- and high-SNR regimes—significantly outperforming CCS—with reasonable complexity: $\mathcal{O}(NL)$.

II. System Model

We assume a total of N OFDM subcarriers, each modulated by a QAM symbol from a 2^M-ary unit-energy constellation $\mathbb{S}$. Of these subcarriers, N_{p} are dedicated as pilots,[3] and the remaining $N_{\mathsf{d}} \triangleq N-N_{\mathsf{p}}$ are used to transmit a total of M_t training bits and $M_{\mathsf{d}} \triangleq N_{\mathsf{d}}M-M_{\mathsf{t}}$ coded/interleaved data bits. To generate the latter, we encode M_{i} information bits using a rate-R coder, interleave them, and partition the resulting $M_{\mathsf{c}} \triangleq M_{\mathsf{i}}/R$ bits among an integer number $T \triangleq M_{\mathsf{c}}/M_{\mathsf{d}}$ of OFDM symbols. The resulting scheme has a spectral efficiency of $\eta \triangleq M_{\mathsf{d}}R/N$ information bits per channel use (bpcu).

In the sequel, we use $s^{(k)} \in \mathbb{S}$ for $k \in \{1,\dots,2^M\}$ to denote the k^{th} element of the QAM constellation, and $\boldsymbol{c}^{(k)} \triangleq (c_1^{(k)},\dots,c_M^{(k)})^{\mathsf{T}}$ to denote the corresponding bits as defined by the symbol mapping. Likewise, we use $s_i[t]\in\mathbb{S}$ for the QAM symbol transmitted on the i^{th} subcarrier of the t^{th} OFDM symbol and $\boldsymbol{c}_i[t] \triangleq (c_{i,1}[t],\dots,c_{i,M}[t])^{\mathsf{T}}$ for the coded/interleaved bits corresponding to that symbol. We use $\boldsymbol{c}[t] \triangleq (\boldsymbol{c}_0[t],\dots,\boldsymbol{c}_{N-1}[t])^{\mathsf{T}}$ to denote the coded/interleaved bits in the t^{th} OFDM symbol and $\boldsymbol{c} \triangleq (\boldsymbol{c}[1],\dots,\boldsymbol{c}[T])^{\mathsf{T}}$ to denote the entire (interleaved) codeword. The elements of $\boldsymbol{c}$ that are apriori known as pilot or training bits will be referred to as $\boldsymbol{c}_{\mathsf{pt}}$. The remainder of $\boldsymbol{c}$ is determined from the information bits $\boldsymbol{b}=(b_1,\dots,b_{M_{\mathsf{i}}})^{\mathsf{T}}$ by coding/interleaving.

We assume the standard model for the received value on subcarrier i of OFDM symbol t:

$$y_i[t] \;=\; s_i[t]z_i[t]+v_i[t], \qquad (2)$$

[1]This work has been supported in part by NSF grant CCF-1018368.

[2]In [1], a scheme that achieves the prelog factor in (1) was proposed, but its complexity grows exponentially with the fading-block length N.

[3]For our relaxed-BP decoder, we recommend $N_{\mathsf{p}}=0$; see Section IV.

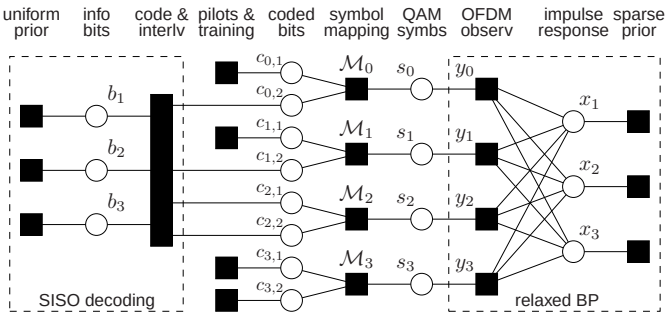

Fig. 1. Factor graph of the joint estimation/decoding problem for a toy example with $M_{\mathsf{i}} = 3$ information bits, $N_{\mathsf{p}} = 1$ pilot subcarrier (at subcarrier index $i = 3$), $M_{\mathsf{t}} = 2$ training bits, $M = 2$ bits per QAM symbol, $N = 4$ OFDM subcarriers, and channel impulse response length $L = 3$.

where $z_i[t] \in \mathbb{C}$ is the i^{th} subcarrier's gain and $\{v_i[t]\}$ is circular white Gaussian noise with variance μ^v. The subcarrier gains $\boldsymbol{z}[t] \triangleq (z_0[t], \ldots, z_{N-1}[t])^{\mathsf{T}}$ are related to the channel impulse response $\boldsymbol{x}[t] \triangleq (x_0[t], \ldots, x_{L-1}[t])^{\mathsf{T}}$ via $z_i[t] = \sum_{j=0}^{L-1} \Phi_{ij} x_j[t]$, where $\Phi_{ij} = e^{-\sqrt{-1}\frac{2\pi}{N}ij}$ can be recognized as the $(i,j)^{th}$ element of the N-DFT matrix $\boldsymbol{\Phi}$. We assume a block fading channel, so that $\{\boldsymbol{x}[t]\}_{t=1}^{T}$ are i.i.d. To simplify the development, we assume that $T = 1$ in the sequel (but not in the simulations) and drop the "$[t]$" notation for brevity.

We assume that the impulse response $\{x_i\}_{i=0}^{L-1}$ is sparse and we model sparsity using an i.i.d Bernoulli-Gaussian prior:

$$p_{X_j}(x) \;=\; \lambda_j \mathcal{CN}(x; 0, \mu_j) + (1 - \lambda_j)\delta(x), \qquad (3)$$

where $\mathcal{CN}(x; a, b) \triangleq (\pi b)^{-1} \exp(-b^{-1}|x - a|^2)$ denotes the complex-Gaussian pdf, $\delta(\cdot)$ the Dirac delta, and $\lambda_j \triangleq \Pr\{X_j \neq 0\}$ and $\mu_j \triangleq \mathrm{var}\{X_j\}$ denote apriori sparsity-rate and variance, respectively. Assuming that the channel is energy-preserving with an exponential delay-power profile, we have $\mu_j = 2^{-j/L_{\mathsf{hpd}}} / (\sum_{r=0}^{L-1} \lambda_r 2^{-r/L_{\mathsf{hpd}}})$, where L_{hpd} denotes the half-power delay.

III. Joint Estimation/Decoding via Relaxed BP

Our goal is to infer the information bits $\boldsymbol{b}$, given the OFDM observations $\boldsymbol{y} \triangleq (y_0, \ldots, y_{N-1})^{\mathsf{T}}$ and the pilot/training bits $\boldsymbol{c}_{\mathsf{pt}}$, but in the absence of channel state information. In particular, we aim to maximize the posterior pmf $p(b_m \,|\, \boldsymbol{y}, \boldsymbol{c}_{\mathsf{pt}})$ of each bit. Given the model of Section II, this posterior can be decomposed into a product of factors as follows:

$$p(b_m \,|\, \boldsymbol{y}, \boldsymbol{c}_{\mathsf{pt}}) \;=\; \sum_{\boldsymbol{b}_{-m}} p(\boldsymbol{b} \,|\, \boldsymbol{y}, \boldsymbol{c}_{\mathsf{pt}}) \;\propto\; \sum_{\boldsymbol{b}_{-m}} p(\boldsymbol{y} \,|\, \boldsymbol{b}, \boldsymbol{c}_{\mathsf{pt}}) p(\boldsymbol{b}) \quad (4)$$

$$= \int_{\boldsymbol{x}} \sum_{\boldsymbol{c}} \sum_{\boldsymbol{s}} \sum_{\boldsymbol{b}_{-m}} p(\boldsymbol{y} \,|\, \boldsymbol{s}, \boldsymbol{x}) p(\boldsymbol{x}) p(\boldsymbol{s} \,|\, \boldsymbol{c}) p(\boldsymbol{c} \,|\, \boldsymbol{b}, \boldsymbol{c}_{\mathsf{pt}}) p(\boldsymbol{b})$$

$$= \int_{\boldsymbol{x}} \prod_{j=0}^{L-1} p(x_j) \sum_{\boldsymbol{s}} \prod_{i=0}^{N-1} p(y_i | s_i, \boldsymbol{x}) \sum_{\boldsymbol{c}} p(s_i | \boldsymbol{c}_i) \sum_{\boldsymbol{b}_{-m}}$$

$$\times\, p(\boldsymbol{c} | \boldsymbol{b}, \boldsymbol{c}_{\mathsf{pt}}) \prod_{m=1}^{M_{\mathsf{i}}} p(b_m), \qquad (5)$$

where "$\propto$" denotes equality up to a scaling and $\boldsymbol{b}_{-m} \triangleq (b_1, \ldots, b_{m-1}, b_{m+1}, \ldots, b_{M_{\mathsf{i}}})^{\mathsf{T}}$. The factorization (5) is illustrated by the *factor graph* in Fig. 1, where the round nodes represent random variables and the square nodes represent the factors of the posterior identified in (5).

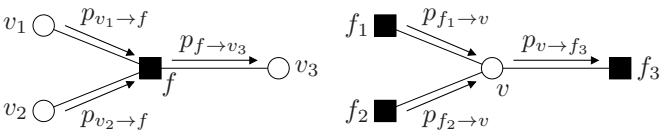

Fig. 2. Examples of belief propagation among nodes of a factor graph.

A. Background on Belief Propagation

Although exact evaluation of the posteriors $\{p(b_m \,|\, \boldsymbol{y}, \boldsymbol{c}_{\mathsf{pt}})\}$ is computationally impractical for the problem sizes of interest, these posteriors can be approximately evaluated using *belief propagation* (BP) [5] on the loopy factor graph in Fig. 1. In BP, beliefs take the form of pdfs/pmfs that are propagated among nodes of the factor graph via the *sum/product algorithm*:

1) If factor node $f(v_1, \ldots, v_A)$ is connected to variable nodes $\{v_a\}_{a=1}^{A}$, then the belief passed from f to v_b is $p_{f \to v_b}(v_b) \propto \int_{\{v_a\}_{a \neq b}} f(v_1, \ldots, v_A) \prod_{a \neq b} p_{v_a \to f}(v_a)$, given the beliefs $\{p_{v_a \to f}(\cdot)\}_{a \neq b}$ previously passed to f.
2) If variable node v is connected to factor nodes $\{f_1, \ldots, f_B\}$, then the belief passed from v to f_a is $p_{v \to f_a}(v) \propto \prod_{b \neq a} p_{f_b \to v}(v)$, given the beliefs $\{p_{f_b \to v}(\cdot)\}_{b \neq a}$ previously passed to v.
3) If variable node v is connected to factor nodes $\{f_1, \ldots, f_B\}$, then the posterior on v is the product of all arriving beliefs, i.e., $p(v) \propto \prod_{b=1}^{B} p_{f_b \to v}(v)$.

Figure 2 helps to illustrate the first two rules.

When the factor graph contains no loops, BP yields exact posteriors after only two rounds of message passing (i.e., forward and backward). With loops, however, convergence to the exact posteriors is not guaranteed. That said, there exist many problems to which loopy BP has been successfully applied, including LDPC decoding [6] and compressed sensing [4], [7]. Our work not only leverages these past successes, but unites them through the framework of turbo equalization [8].

B. Background on Relaxed Belief Propagation

A sub-problem of particular interest to us is the estimation of a non-Gaussian vector $\boldsymbol{x}$ that is linearly mixed to form $\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{x}$ and subsequently observed through componentwise non-Gaussian "measurement channels" $\{p_{Y_i|Z_i}(y_i|z_i)\}_{i=0}^{N-1}$. In our case (3) specifies the non-Gaussian prior on $\boldsymbol{x}$ and (2), with uncertainty in s_i, yields the non-Gaussian channel. This sub-problem yields the factor graph shown within the right dashed box in Fig. 1, where the nodes "y_i" represent the measurement channels and the rightmost nodes represent prior on $\boldsymbol{x}$.

Building on prior multiuser detection work by Guo and Wang [3], Rangan recently proposed a so-called *relaxed BP* scheme [4] that yields asymptotically exact posteriors as $N, L \to \infty$ [7]. The main ideas behind relaxed BP are the following. First, although the beliefs flowing leftward from the nodes $\{x_j\}$ are clearly non-Gaussian, the corresponding belief about $z_i = \sum_{j=0}^{L-1} \Phi_{ij} x_j$ can be accurately approximated as Gaussian, when L is large, using the central limit theorem. Moreover, to calculate the parameters of this distribution (i.e.,

definitions:

$$p_{Z_i|Y_i}(z|y;\hat{z},\mu^z) \;=\; \frac{p_{Y_i|Z_i}(y|z)\,\mathcal{CN}(z;\hat{z},\mu^z)}{\int_{z'} p_{Y_i|Z_i}(y|z')\,\mathcal{CN}(z';\hat{z},\mu^z)} \quad \text{(D1)}$$

$$F_{\mathsf{out},i}(y,\hat{z},\mu^z) \;=\; \int_z z\, p_{Z_i|Y_i}(z|y;\hat{z},\mu^z) \quad \text{(D2)}$$

$$\mathcal{E}_{\mathsf{out},i}(y,\hat{z},\mu^z) \;=\; \int_z |z - F_{\mathsf{out},i}(y,\hat{z},\mu^z)|^2\, p_{Z_i|Y_i}(z|y;\hat{z},\mu^z) \quad \text{(D3)}$$

$$p_{Q_j}(q;\hat{q},\mu^q) \;=\; \frac{p_{X_j}(q)\,\mathcal{CN}(q;\hat{q},\mu^q)}{\int_{q'} p_{X_j}(q')\,\mathcal{CN}(q';\hat{q},\mu^q)} \quad \text{(D4)}$$

$$F_{\mathsf{in},j}(\hat{q},\mu^q) \;=\; \int_q q\, p_{Q_j}(q;\hat{q},\mu^q) \quad \text{(D5)}$$

$$\mathcal{E}_{\mathsf{in},j}(\hat{q},\mu^q) \;=\; \int_q |q - F_{\mathsf{in},j}(\hat{q},\mu^q)|^2\, p_{Q_j}(q;\hat{q},\mu^q) \quad \text{(D6)}$$

initialize:

$$\forall i,j: \hat{x}_{ij}(1) \;=\; \hat{x}_j(1) \;=\; \int_x x\, p_{X_j}(x) \quad \text{(I1)}$$

$$\forall j: \mu^x_j(1) \;=\; \int_x |x - \hat{x}_j(1)|^2 p_{X_j}(x) \quad \text{(I2)}$$

for $n = 1, 2, 3, \ldots$

$$\forall i: \mu^z_i[n] \;=\; \textstyle\sum_{j=0}^{L-1} |\Phi_{ij}|^2 \mu^x_j[n] \quad \text{(R1)}$$

$$\forall i: \hat{z}_i[n] \;=\; \textstyle\sum_{j=0}^{L-1} \Phi_{ij}\hat{x}_{ij}[n] \quad \text{(R2)}$$

$$\forall i,j: \hat{z}_{ij}[n] \;=\; \hat{z}_i[n] - \Phi_{ij}\hat{x}_{ij}[n] \quad \text{(R3)}$$

$$\forall i: \mu^e_i[n] \;=\; \mathcal{E}_{\mathsf{out},i}(y_i,\hat{z}_i[n],\mu^z_i[n]) \quad \text{(R4)}$$

$$\forall i,j: \hat{e}_{ij}[n] \;=\; F_{\mathsf{out},i}(y_i,\hat{z}_i[n],\mu^z_i[n]) - \Phi_{ij}\hat{x}_{ij}[n]\mu^e_i[n]/\mu^z_i[n] - \hat{z}_{ij}[n] \quad \text{(R5)}$$

$$\forall i: \mu^u_i[n] \;=\; \big(1 - \mu^e_i[n]/\mu^z_i[n]\big)^{-1}\mu^z_i[n] \quad \text{(R6)}$$

$$\forall i,j: \hat{u}_{ij}[n] \;=\; \big(1 - \mu^e_i[n]/\mu^z_i[n]\big)^{-1}\hat{e}_{ij}[n] \quad \text{(R7)}$$

$$\forall j: \mu^q_j[n] \;=\; \big(\textstyle\sum_{i=0}^{N-1} |\Phi_{ij}|^2/\mu^u_i[n]\big)^{-1} \quad \text{(R8)}$$

$$\forall j: \hat{q}_j[n] \;=\; \mu^q_j[n] \textstyle\sum_{i=0}^{N-1} \big(\Phi^*_{ij}\hat{u}_{ij}[n]/\mu^u_i[n]\big) \quad \text{(R9)}$$

$$\forall j: \mu^x_j[n+1] \;=\; \mathcal{E}_{\mathsf{in},j}(\hat{q}_j[n],\mu^q_j[n]) \quad \text{(R10)}$$

$$\forall j: \hat{x}_j[n+1] \;=\; F_{\mathsf{in},j}(\hat{q}_j[n],\mu^q_j[n]) \quad \text{(R11)}$$

$$\forall i,j: \hat{x}_{ij}[n+1] \;=\; \hat{x}_j[n+1] - \big(\Phi^*_{ij}\hat{u}_{ij}[n]/\mu^u_i[n]\big)\mu^x_j[n+1] \quad \text{(R12)}$$

end

TABLE I
THE RELAXED-BP ALGORITHM

its mean and variance), only the mean and variance of each x_j are needed. Thus, it suffices to pass only means and variances leftward from each x_j node. It is similarly desirable to pass only means and variances rightward from each measurement node. Although the exact rightward flowing beliefs would be non-Gaussian (due to the non-Gaussian assumption on the measurement channels $p_{Y_i|Z_i}$), relaxed-BP approximates them as Gaussian using a 2nd-order Taylor series, and passes only the resulting means and variances. A further simplification employed by relaxed BP is to approximate the *differences* among the outgoing means/variances of each left node, and the incoming means/variances of each right node, using Taylor series. The relaxed-BP algorithm[4] is summarized in Table I. Assuming (D1)-(D6) can be calculated efficiently (as we show below), the complexity of relaxed-BP is $\mathcal{O}(NL)$.

C. Joint Estimation/Decoding via Relaxed BP

Here we detail our application of relaxed-BP to joint estimation/decoding, frequently referring to the factor graph in Fig. 1. Note that, since our factor graph is loopy, there exists considerable freedom in the belief propagation schedule. We choose to propagate beliefs from the left to the right and back again, several times, stopping as soon as the beliefs converge. Below, we detail each step of the process.

At the very start, we know nothing about the info-bits (i.e., $\Pr\{b_m = 1\} = \frac{1}{2}\ \forall m$). Thus, we take the initial bit beliefs flowing rightward out of the coding/interleaving block to be uniform (i.e., $p_{c_{i,m}\to\delta_i}(c) = \frac{1}{2}\ \forall c$ for the info (i,m)). Meanwhile, we know the pilot/training bits perfectly, and so we set $p_{c_{i,m}\to\delta_i}(c)$ for those (i,m) equal to either 0 or 1, depending on the assigned value of the pilot/training bits.

Next we propagate the coded-bit beliefs rightward into the symbol mapping nodes. The symbol mapping is deterministic, with factors of the form $p(s^{(k)} \,|\, \boldsymbol{c}^{(l)}) = \delta_{k-l}$, using Kronecker delta notation. From the sum/product algorithm, the message passed rightward from symbol mapping node "$\mathcal{M}_i$" is

$$p_{\mathcal{M}_i\to s_i}(s^{(k)}) \;=\; \prod_{m=1}^{M} p_{c_{i,m}\to\mathcal{M}_i}(c^{(k)}_m), \quad (6)$$

which is then copied forward as the message passed rightward from node s_i (i.e., $p_{\mathcal{M}_i\to s_i}(s^{(k)}) = p_{s_i\to y_i}(s^{(k)})$). For brevity, we use $\beta^{(k)}_i \triangleq p_{s_i\to y_i}(s^{(k)})$ in the sequel.

The belief $\{\beta^{(k)}_i\}_{k=1}^{2^M}$ that propagates rightward into the OFDM observation node "y_i" determines the i^{th} "measurement channel" $p_{Y_i|Z_i}(y|z)$ used in relaxed BP. In particular, (2) implies a Gaussian-mixture channel of the form

$$p_{Y_i|Z_i}(y|z) \;=\; \sum_{k=1}^{2^M} \beta^{(k)}_i\, \mathcal{CN}(y; s^{(k)}z; \mu^v), \quad (7)$$

From (7), it can be shown (after a bit of algebra) that the quantities in (D2)-(D3) of Table I become

$$F_{\mathsf{out},i}(y,\hat{z},\mu^z) \;=\; \hat{z} + \hat{e}_i(y,\hat{z},\mu^z) \quad (8)$$

$$\mathcal{E}_{\mathsf{out},i}(y,\hat{z},\mu^z) \;=\; \sum_{k=1}^{2^M} \xi^{(k)}_i(y,\hat{z},\mu^z)\Big(\frac{\mu^v}{|s^{(k)}|^2}\frac{|s^{(k)}|^2\mu^z}{|s^{(k)}|^2\mu^z + \mu^v} + \big|\hat{e}_i(y,\hat{z},\mu^z) - \hat{e}^{(k)}(y,\hat{z},\mu^z)\big|^2\Big) \quad (9)$$

for

$$\xi^{(k)}_i(y,\hat{z},\mu^z) \;\triangleq\; \frac{\beta^{(k)}_i \mathcal{CN}(y; s^{(k)}\hat{z}, |s^{(k)}|^2\mu^z + \mu^v)}{\sum_{k'} \beta^{(k')}_i \mathcal{CN}(y; s^{(k')}\hat{z}, |s^{(k')}|^2\mu^z + \mu^v)} \quad (10)$$

$$\hat{e}^{(k)}(y,\hat{z},\mu^z) \;\triangleq\; \Big(\frac{y}{s^{(k)}} - \hat{z}\Big)\frac{|s^{(k)}|^2\mu^z}{|s^{(k)}|^2\mu^z + \mu^v} \quad (11)$$

$$\hat{e}_i(y,\hat{z},\mu^z) \;\triangleq\; \sum_{k=1}^{2^M} \xi^{(k)}_i(y,\hat{z},\mu^z)\, \hat{e}^{(k)}(y,\hat{z},\mu^z), \quad (12)$$

where $\xi^{(k)}_i(y_i,\hat{z},\mu^z)$ can be interpreted as the posterior on s_i under the channel model $z_i \sim \mathcal{CN}(\hat{z},\mu^z)$. Likewise, from (3), it can be shown that the quantities (D5)-(D6) take the form

$$F_{\mathsf{in},j}(\hat{q},\mu^q) \;=\; \frac{\gamma_j(\hat{q},\mu^q)}{\alpha_j(\hat{q},\mu^q)} \quad (13)$$

$$\mathcal{E}_{\mathsf{in},j}(\hat{q},\mu^q) \;=\; |\gamma_j(\hat{q},\mu^q)|^2\frac{\alpha_j(\hat{q},\mu^q) - 1}{[\alpha_j(\hat{q},\mu^q)]^2} + \frac{\nu_j(\mu^q)}{\alpha_j(\hat{q},\mu^q)}, \quad (14)$$

for

$$\alpha_j(\hat{q},\mu^q) \;\triangleq\; 1 + \frac{1-\lambda_j}{\lambda_j}\frac{\mu_j}{\nu_j(\mu^q)}\exp\!\left(-\frac{|\gamma_j(\hat{q},\mu^q)|^2}{\nu_j(\mu^q)}\right) \quad (15)$$

$$\gamma_j(\hat{q},\mu^q) \;\triangleq\; \frac{\nu_j(\mu^q)}{\mu^q}\hat{q} \quad (16)$$

$$\nu_j(\mu^q) \;\triangleq\; \frac{\mu^q\mu_j}{\mu^q + \mu_j}. \quad (17)$$

[4]To be precise, the relaxed BP algorithm in Table I is an extension of that proposed in [4]. Table I handles *complex* Gaussian distributions and *non-identically distributed* signal and measurement channels.

Using (8)-(17), we iterate the relaxed-BP algorithm in Table I until it converges. In doing so, we generate (a close approximation to) the conditional-mean (i.e., nonlinear MMSE) estimates of the sparse-channel impulse-response coefficients $\{x_j\}$, as well as their conditional variances $\{\mu_j^x\}$, given the observations $\{y_i\}$ and the soft symbol estimates $\{\beta_i^{(k)}\}$. Conveniently, relaxed-BP also returns (a close approximation to) the conditional-mean estimates $\{\hat{z}_i\}$ of the subchannel gains $\{z_i\}$, as well as their conditional variances $\{\mu_i^z\}$.

After several rounds of message passing within the relaxed-BP sub-graph, beliefs are passed leftward out of that sub-graph. The sum/product algorithm implies that the belief propagating leftward from the y_i node takes the form

$$p_{s_i \leftarrow y_i}(s) \propto \int_z \mathcal{CN}(y_i; sz, \mu^v)\, \mathcal{CN}(z; \hat{z}_i, \mu_i^z) \tag{18}$$

$$= \mathcal{CN}(y_i; s\hat{z}_i, |s|^2 \mu_i^z + \mu^v), \tag{19}$$

where the relaxed-BP outputs $(\hat{z}_i, \mu_i^z)$ play the role of "soft channel estimates." Likewise, $p_{\mathcal{M}_i \leftarrow s_i}(s) = p_{s_i \leftarrow y_i}(s)$.

Next, the belief passed leftward from the symbol-mapping node $\mathcal{M}_i$ to the bit node $c_{i,m}$ takes the form

$$p_{c_{i,m} \leftarrow \mathcal{M}_i}(c)$$

$$\propto \sum_{k=1}^{2^M} \sum_{\boldsymbol{c}: c_m = c} p(s^{(k)} \,|\, \boldsymbol{c})\; p_{\mathcal{M}_i \leftarrow s_i}(s^{(k)}) \prod_{m' \neq m} p_{c_{i,m'} \rightarrow \mathcal{M}_i}(c_{m'})$$

$$= \sum_{k: c_m^{(k)} = c} p_{\mathcal{M}_i \leftarrow s_i}(s^{(k)}) \frac{\prod_{m'=1}^{M} p_{c_{i,m'} \rightarrow \mathcal{M}_i}(c_{m'}^{(k)})}{p_{c_{i,m} \rightarrow \mathcal{M}_i}(c)} \tag{20}$$

$$= \frac{1}{p_{c_{i,m} \rightarrow \mathcal{M}_i}(c)} \sum_{k: c_m^{(k)} = c} p_{\mathcal{M}_i \leftarrow s_i}(s^{(k)}) p_{\mathcal{M}_i \rightarrow s_i}(s^{(k)}). \tag{21}$$

Of course, $p_{c_{i,m} \leftarrow \mathcal{M}_i}(c)$ does not need to be evaluated for any pair (i,m) corresponding to a pilot or training bit.

Finally, we pass messages leftward into the code/interleave block. In doing so, we are essentially feeding extrinsic soft bit estimates to a soft-input/soft-output (SISO) decoder/deinterleaver, where they are treated as priors. Since SISO decoding is by now a mature topic, we refer the reader elsewhere for details (e.g., [9]). The extrinsic outputs of the SISO decoder are then re-interleaved and passed rightward from the code/interleave block to begin another round of belief propagation on the overall factor graph in Fig. 1. These outer ("turbo") iterations continue until either the decoder detects no bit errors, the soft bit estimates have converged, or a maximum number of iterations has elapsed.

IV. Numerical Results

In this section, we present numerical results that compare our proposed BP-based joint estimator/decoder to the CCS approach as well as to several reference schemes.

We used the following procedure for CCS. First, we generated a LASSO channel estimate $\hat{\boldsymbol{x}}[t]$ using pilot-subcarriers. To implement LASSO, we used SPGL1 [10] with a genie-optimized tuning parameter. We then computed $\hat{\boldsymbol{z}}[t] = \boldsymbol{\Phi}\hat{\boldsymbol{x}}[t]$, from which we calculated the (genie-aided empirical) variance $\hat{\mu}_i^z[t] \triangleq \|\hat{\boldsymbol{z}}[t] - \boldsymbol{z}[t]\|_2^2/N$. From this "soft" channel estimate, we computed soft coded-bit estimates using the procedure described for BP and fed them to the SISO deinterleaver/decoder to yield info-bit estimates. Due to the genie-aided steps, the performance attained by CCS may be somewhat optimistic.

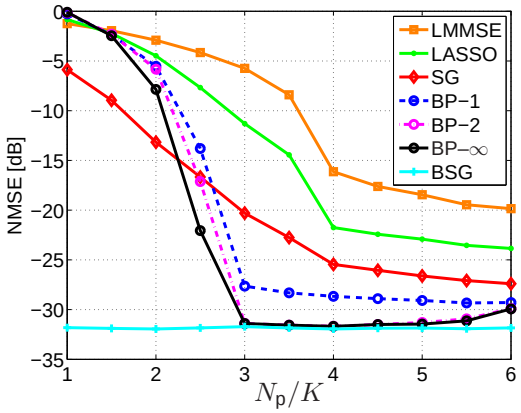

Fig. 3. Channel estimation NMSE versus pilot ratio N_p/K, for SNR = 20dB, $M_\mathsf{t} = 0$, $\eta = 3$ bpcu, and 64-QAM.

We now describe several reference schemes, all of which use the CCS procedure described above but with different channel estimators. The first uses traditional linear MMSE (LMMSE) estimation. Since LMMSE does not exploit channel sparsity, we expect it to be outperformed by LASSO. We also consider MMSE-optimal estimation under the *support-aware genie* (SG), which yields a CCS performance upper-bound. Finally, we consider MMSE-optimal estimation under a *bit- and support-aware genie* (BSG). Here, in addition to the channel support being known, all N symbols are known during the channel estimation step. Clearly, this reference upper-bounds the performance of *any* implementable decoder.

For all of our results, we used irregular LDPC codes with length $\approx$ 10000 and average column weight 3, generated (and decoded) using the software [11]. In converting bits to symbols, we used multilevel Gray-mapping. For OFDM, we used $N = 1021$ subcarriers, since prime N ensures that square/tall submatrices of $\boldsymbol{\Phi}$ will be full-rank. The N_p pilot subcarriers, when used, were spaced uniformly and modulated with QAM symbols chosen uniformly at random. The M_t training bits, when used, were chosen uniformly at random and assigned to the most significant bits of uniformly spaced data subcarriers.

For all of our results, we used a length $L = 256$ channel with $\lambda = 1/4$, yielding $\lambda N = 64 = \mathrm{E}\{K\}$ non-zero taps on average. All results are averaged over $T = 100$ OFDM symbols.

Figure 3 plots channel estimation NMSE $\triangleq \|\hat{\boldsymbol{x}}[t] - \boldsymbol{x}[t]\|_2^2/\|\boldsymbol{x}[t]\|_2^2$ versus the pilot ratio N_p/K at SNR = 20dB. As expected, LASSO's NMSE falls between that of LMMSE and SG, and all three decrease monotonically with N_p. Even after a single turbo iteration, BP significantly outperforms LASSO, and—perhaps surprisingly—the SG (when $N_\mathsf{p}/K \geq 3$). The reason for the latter is that, while the SG uses only N_p subcarriers, BP makes use of all N subcarriers, although the

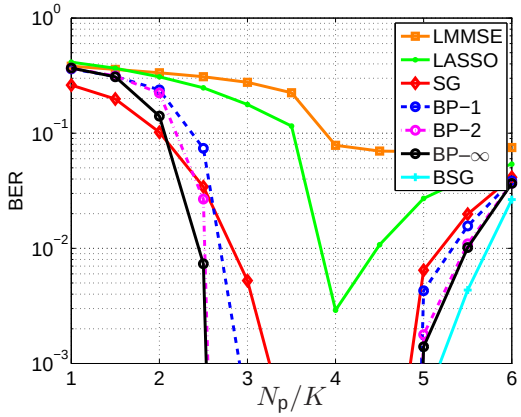

Fig. 4. BER versus pilot ratio N_p/K, for SNR $= 20$dB, $M_t = 0$, $\eta = 3$ bpcu, and 64-QAM.

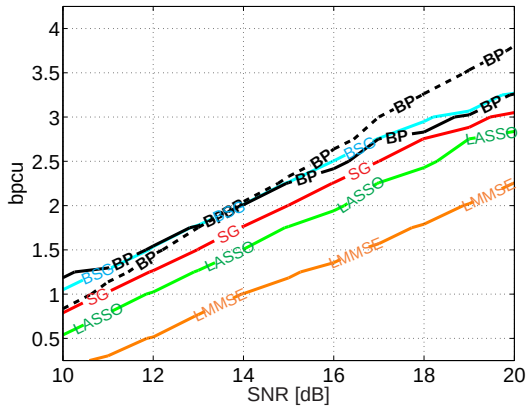

Fig. 5. BER $= 0.001$-achieving spectral efficiency $\eta_{0.001}$ versus SNR. The solid traces used $N_p/K = 4$, $M_t = 0$, and 64-QAM, while the dashed trace used $N_p = 0$, $M_t = MK$, and 256-QAM.

$N_d = N - N_p$ data subcarriers are extremely "noisy" at first, due to complete lack of symbol knowledge. After only 2 turbo iterations, BP learns the data subcarriers well enough that its NMSE is only slightly higher than that of the BSG (which knows all data subcarriers perfectly). The fact that these BP estimates are nearly as good as BSG's support-aware estimates attests to the near-optimal estimation ability of relaxed BP.

Figure 4 plots BER versus the pilot ratio N_p/K at SNR $=$ 20dB and a fixed spectral efficiency of $\eta = 3$ bpcu. The curves have a "notched" shape because, as N_p increases, the code rate R must decrease to maintain a fixed value of η. Thus, while an increase in N_p can make channel estimation easier, the reduction in R makes data decoding more difficult. For CCS, Fig. 4 indicates that $N_p = 4K = L$ is optimal, which is interesting because, when $N_p \geq L$, the channel sensing is not actually "compressed." The SG and BP curves show a similar notch-like shape, although their notches are much wider.

Figure 5 plots $\eta_{0.001}$ versus SNR, where $\eta_{0.001}$ is the spectral efficiency (in bpcu) that yielded BER $= 0.001$. The solid-line traces correspond to $N_p = 4K = L$ pilots and $M_t = 0$ training bits and 64-QAM, as suggested by Fig. 4. These traces all display the anticipated high-SNR scaling law

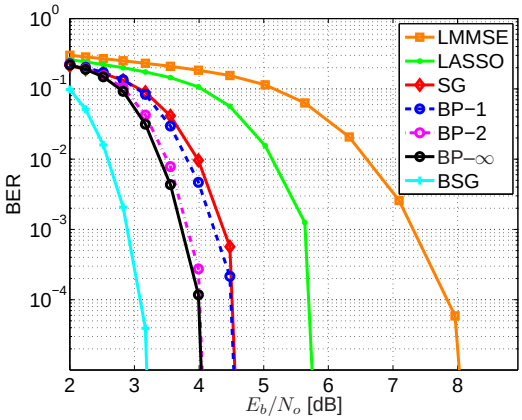

Fig. 6. BER versus $E_b/N_o (=\text{SNR}/\eta)$, for $N_p/K = 4$, $M_t = 0$, $\eta = 0.5$ bpcu, and 4-QAM.

$\frac{N-N_p}{N}\log_2(\text{SNR}) + \mathcal{O}(1)$, differing only in the $\mathcal{O}(1)$ term. Although, for this setup, BP performs on par with BSG, neither attains the desired channel-capacity prelog-factor of $\frac{N-K}{N} = \frac{15}{16}$. It turns out that this shortcoming is due to the choice $(N_p, M_t) = (L, 0)$, which was chosen with CCS in mind. In fact, experiments not shown here confirmed that, at high SNR, BP performs best with $(N_p, M_t) = (0, MK)$. This latter configuration, in conjunction with 256-QAM, yields the dashed $\eta_{0.001}$-vs-SNR in Fig. 5, which—remarkably—does attain the optimal prelog-factor, $\frac{N-K}{N}$.

Figure 6 plots BER versus $E_b/N_o (=\text{SNR}/\eta)$ over a much lower range of SNR. Experiments (not shown) confirmed that CCS favors $(N_p, M_t) = (L, 0)$ in this SNR range as well, and so this configuration was used to keep CCS competitive, while being suboptimal for BP. Still, we see from Fig. 6 that BP, after only two turbo iterations, beats LASSO by 1.8dB and remains only 0.8dB away from the BSG.

REFERENCES

[1] A. P. Kannu and P. Schniter, "On communication over unknown sparse frequency-selective block-fading channels," *arXiv:1006.1548*, June 2010.

[2] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, pp. 1058–1076, June 2010.

[3] D. Guo and C.-C. Wang, "Random sparse linear systems observed via arbitrary channels: A decoupling principle," in *Proc. IEEE Int. Symposium Inform. Theory*, (Nice, France), pp. 946–950, June 2007.

[4] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," *arXiv:1001.2228v2*, May 2010.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[6] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. New York: Cambridge University Press, 2003.

[7] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," *arXiv:1010.5141*, Oct. 2010.

[8] R. Koetter, A. C. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Process. Mag.*, vol. 21, pp. 67–80, Jan. 2004.

[9] T. J. Richardson and R. L. Urbanke, *Modern coding theory*. New York: Cambridge University Press, 2009.

[10] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Scientific Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[11] I. Kozintsev, "Matlab programs for encoding and decoding of LDPC codes over GF(2^m)." http://www.kozintsev.net/soft.html.