Joint Channel-Estimation and Equalization of Single-Carrier Systems via Bilinear AMP

Peng Sun, Zhongyong Wang, and Philip Schniter, *Fellow, IEEE*

***Abstract*—We propose a novel soft-input soft-output equalizer for single-carrier transmissions over unknown frequency-selective block-fading channels. Our equalizer leverages the recently proposed parametric bilinear generalized approximate message passing algorithm for joint channel-estimation and symbol-detection, and exploits fast Fourier transform (FFT)-processing to achieve a per-symbol complexity that grows only logarithmically in the channel delay-spread. Furthermore, it supports the use of Gaussian mixture models to support the approximately sparse nature of wideband wireless channel responses. Numerical experiments, conducted using physically motivated Saleh–Valenzuela channel models, show that the proposed approach achieves channel normalized mean square error and bit error rate that are significant improved over existing turbo frequency-domain equalization approaches for unknown channels. Additional experiments show that the proposed scheme facilitates much higher spectral efficiencies than sparse deconvolution methods based on convex relaxation.**

***Index Terms*—Single-carrier block transmission, joint channel estimation and equalization, turbo equalization, approximate message passage.**

I. Introduction

Single-carrier (SC) block transmission with frequency domain equalization (FDE) [1], [2] is an attractive technology for communications across frequency-selective channels, especially for short-burst communications. Compared to orthogonal frequency division multiplexing (OFDM) [3], SC-FDE has similar performance and complexity but much lower peak-to-average power ratio (PAPR), which relaxes requirements on power-amplifier linearity and thus enables the use of more efficient and/or cheaper amplifiers. SC-FDE is also less sensitive to frequency offsets and phase noise [4], [5].

The SC-FDE receiver's goal is to recover the information bits from the received samples, which are corrupted by unknown frequency-selective (FS) fading and additive noise. To facilitate this task, it is common to transmit known pilot symbols [6]. This approach is justified by the fact that pilot-aided transmission can attain the maximum spectral efficiency achievable with unknown FS Raleigh fading [7].

To infer the information bits, conventional SC-FDE receivers perform channel estimation, equalization, and decoding in a *decoupled* manner. That is, pilots are first used to estimate the channel, the channel estimate is then used to infer the data symbols (i.e., "equalization"), and the symbol estimates are finally used to infer the information bits (i.e., "decoding"). Computationally, this approach is very efficient: fast Fourier transforms (FFTs) can be used for channel estimation and equalization, so that the per-symbol estimation complexity grows only *logarithmically* with the channel delay spread L. The decoupling of these tasks, however, leads to suboptimal decoding performance. Consequently, many *joint* equalization/estimation/coding schemes have been proposed, most of which are iterative in nature.

A. Turbo Equalization

Turbo equalization [8], [9] is a well-known example of iterative (joint) detection and decoding. There, information is repeatedly exchanged between a soft-input soft-output (SISO) equalizer and a SISO decoder. This information takes the form of "extrinsic log-likelihood ratios (LLRs)" on the coded bits that determine each symbol. Belief propagation (BP) [10], and in particular the sum-product algorithm (SPA) [11], explains why information should be exchanged in this way [9].

When the channel response is known, exact SISO equalization [8], [9] can be accomplished using the trellis-based BCJR algorithm[1] [12]. But its complexity grows as $O(A^L)$, where A is the number of coded bits per quadrature amplitude modulation (QAM) symbol, which is impractical for large A or L. The soft-output Viterbi [13] and max-log-MAP [14] approximations of BCJR also have $O(A^L)$ complexity. A substantial reduction in complexity can be obtained by reducing the number of trellis states via, e.g., delayed decision-feedback sequence estimation [15], [16]. Even more complexity reduction can be achieved by using *linear* processing in place of trellis processing, which can be justified by modeling the symbols as independent Gaussian during SISO equalization. Many such schemes have been

Manuscript received October 2, 2017; revised February 6, 2018; accepted February 22, 2018. Date of publication March 9, 2018; date of current version April 17, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsung-Hui Chang. The work of P. Sun and Z. Wang was supported by the National Natural Science Foundation of China under Grant NSFC 61571402. The work of P. Schniter was supported by the National Science Foundation under Grant CCF-1527162. *(Corresponding author: Philip Schniter.)*

P. Sun is with the School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China and also with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: sun.1771@osu.edu).

Z. Wang is with the School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: iezywang@zzu.edu.cn).

P. Schniter is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: schniter.1@osu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2018.2812720

[1] Also known as the forward-backward algorithm or symbol-MAP detection.

proposed under different names, such as "soft inter-symbol interference (ISI) cancellation" and "soft decision feedback equalization (DFE)." See, e.g., [17]–[22].

The above discussion pertains to known-channel SISO equalization. When the channel response is *unknown*, the complexity of exact LLR computation grows as $O(A^M)$, where M is the block length [23]. Trellis-based approximations have been proposed (e.g., [24]–[26]) but they remain impractical for large L. Thus, most practical strategies *iterate* between channel estimation (using symbol estimates) and SISO equalization (using a channel estimate), as illustrated in Fig. 1(a). For channel estimation, it has been proposed to use expectation maximization (EM) [15], [27], [28], decision-directed least-squares (LS) [29], linear minimum mean-squared error (LMMSE) estimation [30], or adaptive strategies such as least mean square (LMS), recursive least square (RLS), or Kalman filtering [31]–[33]. As an alternative, linear equalizer coefficients can designed directly, without explicit channel estimation (e.g., [34]).

Among the works cited above, the per-symbol complexity of SISO equalization grows at least linearly in the number of equalizer coefficients (and thus L), and often quadratically due to the use of matrix inversion to design the equalizer and/or estimate the channel. This stands in contrast to conventional SC-FDE reception, whose per-symbol complexity grows only *logarithmically* in L due to the use of FFTs for FDE and frequency domain (FD) channel estimation. The use of FFTs in SC block-transmission systems is facilitated by the use of guard sequences (e.g., cyclic prefix (CP), zero prefix, unique word (UW)) for block separation, which allows FS channel propagation to be modeled as circular convolution [1], [2].

Motivated by the possibility of "fast" (i.e., $O(\log L)$ per-symbol complexity) processing, a number of SC-FDE SISO equalization schemes have been proposed, including [35]–[38]. However, matrix inversion is used for FDE design in [36] and channel-estimation in [37], so neither is fast overall. Furthermore, among the fast SC-FDE channel-estimation/equalization schemes, we are not aware of any that can exploit the sparsity exhibited by wideband FS channel responses [39]. As we show in the sequel, exploiting such sparsity can lead to large gains in performance.

Just as the SPA explains the messages passed between the SISO equalizer and decoder in known-channel turbo equalization [9], the SPA also provides a principled framework for SISO equalization of an unknown channel. A direct application of the SPA under FS Rayleigh-fading channels is, however, impractical: it suggests passing messages with a Gaussian-mixture form, where the mixture order grows exponentially in the number of iterations [40]. Thus, for practical joint channel-estimation/equalization, the SPA must be approximated.

B. Approximate Message Passing

"Approximate message passing" (AMP) [41] is a principled approximation framework for the SPA. AMP leverages the central limit theorem and other Taylor expansions to obtain computationally efficient algorithms with predictable performance. In particular, AMP's large-system behavior is rigorously characterized by a scalar state evolution whose fixed points, when unique, attain MMSE performance [42]. Originally, AMP was developed for sparse *linear* estimation problems [43], [44]. In this form, AMP is directly applicable to sparse FS channel estimation (assuming independent channel taps with known distribution), as well as SISO equalization, as was established in several earlier works (e.g., [45]–[47]) before the name "AMP" was crystallized. Subsequently, AMP was extended to non-independent priors in [48], facilitating its application to clustered-sparse channel estimation [49]. Meanwhile, AMP was extended to generalized-linear estimation problems [50], which enabled its application to joint channel-estimation/SISO-equalization of OFDM [49]. Methods to deal with unknown priors were subsequently developed in [51].

To our knowledge, there has been no work applying AMP to joint channel-estimation/SISO-equalization of SC transmissions over FS channels, mainly due to limitations in the AMP methodology. However, a "parametric bilinear generalized AMP" (PBiGAMP), which tackles generic *bilinear* estimation problems, was recently proposed [52] and analyzed [53].

C. Contributions

In this paper, we will show that PBiGAMP offers an attractive method for SISO equalization of SC block transmissions over unknown FS channels. When used in a turbo configuration [48] with an appropriate channel model and EM-learning of the channel statistics [51], PBiGAMP leads to a SC-FDE receiver that i) has $O(\log L)$ per-symbol complexity, ii) can learn and leverage channel sparsity, and iii) performs close to oracle bounds. Our claims are supported by numerical experiments with realistic wireless channel models of the Saleh-Valenzuela (SV) form [54] with parameters motivated by physical channel-sounding experiments [39].

Although we focus on single-antenna systems and time-invariant channels in this paper, our approach can be straightforwardly extended to multiple antenna systems, as well as time-varying channels (e.g., through the use of basis-expansion models [55]). Similarly, although we focus on SC systems, our approach can be straightforwardly extended to discrete Fourier transform (DFT)-precoded systems like [56]. We leave these extensions to future work.

The paper is organized as follows. In Section II, we present our SC block transmission model and SV channel model, as well as a Gaussian-mixture channel model that is compatible with PBiGAMP. In Section III, after a brief introduction to belief propagation and PBiGAMP, we propose our PBiGAMP-based SISO equalizer and explain how to integrate it into a turbo receiver. We also describe methods to learn the statistical parameters of our channel model. In Section IV, we detail several benchmark schemes that will be used in our numerical comparisons. They include traditional turbo SC-FDE schemes like [37] and [38], as well as the recently proposed convex SparseLift method from [57]. In Section V, we report numerical results, and in Section VI we conclude.

Notation—We use boldface uppercase letters like $\boldsymbol{B}$ to denote matrices and boldface lowercase letters like $\boldsymbol{b}$ to denote vectors,

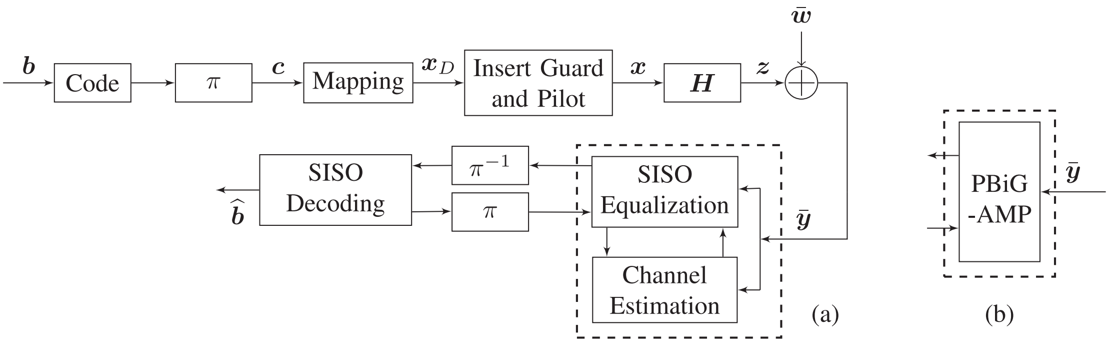

Fig. 1. The block diagram of SC block transmission and turbo reception, where π and π^{-1} represent interleaving and deinterleaving, respectively, and $\boldsymbol{H}$ represents channel propagation. Diagram (a) shows reception using a SISO equalizer that iterates between symbol and channel estimation. By replacing the dashed box in (a) with the dashed box in (b), we obtain the proposed receiver, which uses PBiGAMP for joint symbol and channel estimation.

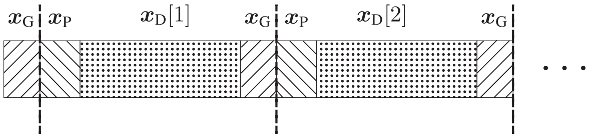
$\boldsymbol{x}_\text{G}$
$\boldsymbol{x}_\text{P}$
$\boldsymbol{x}_\text{D}[1]$
$\boldsymbol{x}_\text{G}$
$\boldsymbol{x}_\text{P}$
$\boldsymbol{x}_\text{D}[2]$
$\boldsymbol{x}_\text{G}$
$\cdots$

Fig. 2. The transmitted block structure (showing $K = 2$ blocks).

where b_i represents the ith element of $\boldsymbol{b}$. The notation $[\boldsymbol{B}]_{i,j}$ extracts the ith row and jth column from the matrix $\boldsymbol{B}$. $\boldsymbol{I}_M$ denotes the $M \times M$ identity matrix, $\mathbf{1}_M$ denotes the M-length vector of ones, and $\mathbf{0}_M$ denotes the M-length vector of zeros. $\text{Diag}(\boldsymbol{b})$ is the diagonal matrix formed from the vector $\boldsymbol{b}$. $\boldsymbol{F}_N$ is the $N \times N$ unitary discrete Fourier transform (DFT) matrix, $\boldsymbol{F}_N^{1:L}$ is the matrix formed by the first L columns of $\boldsymbol{F}_N$, $\boldsymbol{f}_N^i$ is the ith column of $\boldsymbol{F}_N$, and f_N^{ij} is the $(i+1, j+1)$th element of $\boldsymbol{F}_N$. For matrices and vectors, $(\cdot)^\mathsf{T}$ denotes transpose, $(\cdot)^\mathsf{H}$ denotes conjugate transpose, and $(\cdot)^*$ denotes conjugate. Likewise, $\odot$, $\oslash$, and $|\cdot|^{\odot 2}$ denote elementwise multiplication, division, and absolute-value squared, respectively. Finally, the probability density function (pdf) of a multivariate complex Gaussian random vector $\boldsymbol{x}$ with mean $\widehat{\boldsymbol{x}}$ and covariance $\boldsymbol{\Sigma}$ will be denoted by $\mathcal{CN}(\boldsymbol{x}; \widehat{\boldsymbol{x}}, \boldsymbol{\Sigma})$.

II. System Model

A. SC Block Transmission Model

We consider a SC block transmission system, where the kth block takes the form $\boldsymbol{x}[k] = [x_0[k], x_1[k], \ldots, x_{M-1}[k]]^\mathsf{T} = [\boldsymbol{x}_\text{P}^\mathsf{T}, \boldsymbol{x}_\text{D}[k]^\mathsf{T}, \boldsymbol{x}_\text{G}^\mathsf{T}]^\mathsf{T}$, with pilot sequence $\boldsymbol{x}_\text{P} \in \mathbb{C}^{N_\text{P}}$, data sequence $\boldsymbol{x}_\text{D} \in \mathcal{S}^{N_\text{D}}$, and guard sequence $\boldsymbol{x}_\text{G} \in \mathbb{C}^{N_\text{G}}$ (see Fig. 2). Here, $\mathcal{S}$ denotes the 2^A-ary symbol alphabet, e.g., QAM. This block structure covers CP transmission, where $\boldsymbol{x}_\text{G}$ is constructed from the last N_G data symbols of the next $\boldsymbol{x}_\text{D}$, UW transmission, where $\boldsymbol{x}_\text{G} = \boldsymbol{x}_\text{P}$ is invariant across blocks, and zero-padded (ZP) transmission, where $\boldsymbol{x}_\text{G} = \mathbf{0}_{N_\text{G}}$. (See [1] for a review of CP, UW, and ZP.)

The data sequences $\boldsymbol{x}_\text{D}[k]$ are constructed as follows. First, N_b information bits $\boldsymbol{b} = [b_1, \ldots, b_{N_\text{b}}]^\mathsf{T}$ are coded and interleaved to yield the coded bits $\boldsymbol{c} \in \{0,1\}^{AKN_\text{D}}$, where the code rate is $R \triangleq \frac{N_\text{b}}{AKN_\text{D}}$. The coded bits are then partitioned into KN_D groups of A bits, $\boldsymbol{c} = [\boldsymbol{c}_0^\mathsf{T}, \ldots, \boldsymbol{c}_{KN_\text{D}-1}^\mathsf{T}]^\mathsf{T}$, where each group $\boldsymbol{c}_n = [c_{n,1}, \ldots, c_{n,A}]^\mathsf{T}$ determines the value of one data symbol. The KN_D data symbols are then partitioned into K blocks of N_D symbols, $\{\boldsymbol{x}_\text{D}[k]\}_{k=1}^K$, where $\boldsymbol{x}_\text{D}[k] = [x_{N_\text{P}}[k], \ldots, x_{N_\text{P}+N_\text{D}-1}[k]]^\mathsf{T}$.

The blocks $\{\boldsymbol{x}[k]\}_{k=1}^K$ are modulated using the pulse shape $g_\text{t}(\tau)$, yielding the baseband transmission waveform

$$a(t) = \sum_{k=1}^{K} \sum_{m=0}^{M-1} x_m[k] g_\text{t}(t - mT - (k-1)MT), \tag{1}$$

where T is the baud interval. The waveform $a(t)$ is then upconverted, transmitted through a noisy and frequency selective channel, and downconverted, yielding the received baseband signal

$$\bar{y}(t) = \int_{\tau_\text{min}}^{\tau_\text{max}} g(t,\tau) a(t-\tau)\, \mathrm{d}\tau + \bar{w}(t), \tag{2}$$

where $\bar{w}(t)$ is Gaussian noise with flat power spectral density $\sigma_{\bar{w}}^2$ and $g(t,\tau)$ is the baseband-equivalent channel impulse response at time t and lag τ, supported on the interval $[\tau_\text{min}, \tau_\text{max}]$. The receiver filters $\bar{y}(t)$ with $g_\text{r}(\tau)$ and samples the result at integer multiples of T seconds, yielding

$$\bar{y}_m[k] = \int_{-\infty}^{\infty} \bar{y}(t) g_\text{r}(mT + (k-1)MT - t)\, \mathrm{d}t, \tag{3}$$

where again k is the block index and m is the sample index within the block. Let us define the effective channel response

$$h(t,\tau) \triangleq (g_\text{t} \star g \star g_\text{r})(t,\tau), \tag{4}$$

where $\star$ denotes convolution in the τ domain. Assuming that $h(t,\tau)$ is invariant over each time interval $t \in [kMT, (k+1)MT)$ and supported on the lags $\tau \in [0, LT)$, we can write

$$\bar{y}_m[k] = \sum_{l=0}^{L-1} x_{m-l}[k] h_l[k] + \bar{w}_m[k] \tag{5}$$

for $k = 1, \ldots, K$ and $m = 0, \ldots, M-1$, where

$$h_l[k] \triangleq h((k-1)MT, lT), \tag{6}$$

and where $x_q[k]|_{q<0} = x_{q+M}[k-1]$. Assuming $(g_\text{t} \star g_\text{r})(\tau)$ is a Nyquist pulse, $\bar{w}_m[k]$ are i.i.d. Gaussian with variance $\sigma_{\bar{w}}^2$.

When $\boldsymbol{x}_\text{G}$ is identical across blocks (as in UW or ZP systems) and $N_\text{G} \geq L-1$, the linear convolution in (5) reduces to circular convolution[2] with period M. Defining $\bar{\boldsymbol{y}}[k] \triangleq [\bar{y}_0[k], \ldots, \bar{y}_{M-1}[k]]^\mathsf{T}$, $\boldsymbol{h}[k] \triangleq [h_0[k], \ldots, h_{L-1}[k]]^\mathsf{T}$, and $\bar{\boldsymbol{w}}[k] \triangleq [\bar{w}_0[k], \ldots, \bar{w}_{M-1}[k]]^\mathsf{T}$, we have

$$\bar{\boldsymbol{y}}[k] = \boldsymbol{H}[k]\boldsymbol{x}[k] + \bar{\boldsymbol{w}}[k], \tag{7}$$

where $\boldsymbol{H}[k]$ is the $M \times M$ circulant matrix with first column $[\boldsymbol{h}[k]^\mathsf{T}\ \boldsymbol{0}_{M-L}^\mathsf{T}]^\mathsf{T}$. The receiver converts to the frequency domain via

$$\boldsymbol{y}[k] \triangleq \boldsymbol{F}_M \bar{\boldsymbol{y}}[k]/\sqrt{M}, \tag{8}$$

after which $\boldsymbol{H}[k] = \boldsymbol{F}_M^\mathsf{H} \mathrm{Diag}(\sqrt{M}\boldsymbol{F}_M^{1:L}\boldsymbol{h}[k])\boldsymbol{F}_M$ implies that

$$\boldsymbol{y}[k] = \mathrm{Diag}(\boldsymbol{F}_M^{1:L}\boldsymbol{h}[k])\boldsymbol{F}_M\boldsymbol{x}[k] + \boldsymbol{w}[k], \tag{9}$$

where $\boldsymbol{w}[k] \triangleq \boldsymbol{F}_M \bar{\boldsymbol{w}}[k]/\sqrt{M}$ is i.i.d. Gaussian with variance $\sigma_w^2 = \sigma_{\bar{w}}^2/M$.

B. Saleh-Valenzuela Model for the Continuous-Time Channel

We assume that the continuous-time channel impulse responses $g(t,\tau)$ obey the SV model [54]

$$g(t,\tau) = \sum_{c=1}^{C}\sum_{v=1}^{V} g_{v,c} e^{j\phi_{v,c}} \delta(\tau - T_c - \tau_{v,c}), \tag{10}$$

where C is the number of clusters, V is the number of paths per cluster, T_c is the delay of the cth cluster, $\tau_{v,c}$ are relative path delays, $g_{v,c} \geq 0$ are path amplitudes, $\phi_{v,c} \in [0, 2\pi)$ are path phases, and $\delta(\cdot)$ is the Dirac delta. Here, $C, V, \{T_c, \{\tau_{v,c}, g_{v,c}, \phi_{v,c}\}_{v=1}^{V}\}_{c=1}^{C}$ are random variables with prescribed distributions, as detailed in Appendix A. We assume block fading, where $g(t,\tau)$ is fixed during the time intervals $t \in [kMT, (k+1)MT)$ and is i.i.d. across k.

C. Gaussian-Mixture Model for the Discrete-Time Channel

The SV model is accurate (see, e.g., [39]) and so we will use it to generate channel realizations for the simulations in Section V. But it is not easy to exploit for channel estimation. For example, the number of paths, VC, in the SV model (10) is typically much larger than the number of parameters, L, in the discrete-time model $\boldsymbol{h}[k]$. So, we find it more efficient to model the coefficients $\boldsymbol{h}[k]$ directly. Due to the block-fading assumption, we will estimate the channel separately for each block k, and so we drop the block index "$[k]$" for brevity.

We propose to use a D-state Gaussian-mixture model (GMM) for the channel coefficients h_l, which takes the form

$$p(h_l) = \sum_{d=1}^{D} \lambda_{l,d}\, \mathcal{CN}(h_l; 0, \nu_{l,d}), \tag{11}$$

where $\nu_{l,d} > 0$ are variances, $\lambda_{l,d} \geq 0$ are weights (such that $\sum_d \lambda_{l,d} = 1 \forall l$), and a mean of zero has been assumed. We note that a 2-state GMM was proposed in [49], while here we consider a D-state GMM.

The GMM model (11) can be considered a simplification of that proposed in [49], which used a hidden Markov model (HMM) to couple the coefficients h_l across lag l, thus promoting clustered sparsity. We will compare the performance of the GMM (11) with the GMM/HMM from [49] in Section V.

III. Turbo Equalization With PBiGAMP

Our goal is to infer the information bits $\boldsymbol{b}$ from the frequency-domain observations $\boldsymbol{y} \triangleq [\boldsymbol{y}[1]^\mathsf{T}, \ldots, \boldsymbol{y}[K]^\mathsf{T}]$ under the SC block-transmission model from Section II-A and the GMM channel model from Section II-C. In particular, our goal is to compute the marginal posterior probabilities $\{p(b_i|\boldsymbol{y})\}_{i=1}^{N_\mathrm{b}}$, which take the form

$$p(b_i|\boldsymbol{y}) = \sum_{\boldsymbol{b}_{-i}} p(\boldsymbol{b}|\boldsymbol{y}) = \sum_{\boldsymbol{b}_{-i}} \frac{p(\boldsymbol{y}|\boldsymbol{b})p(\boldsymbol{b})}{p(\boldsymbol{y})} \propto \sum_{\boldsymbol{b}_{-i}} p(\boldsymbol{y}|\boldsymbol{b}) \tag{12}$$

$$= \sum_{\boldsymbol{b}_{-i},\boldsymbol{x},\boldsymbol{c}} \int_{\boldsymbol{h}} p(\boldsymbol{y}|\boldsymbol{h},\boldsymbol{x})p(\boldsymbol{h})p(\boldsymbol{x}|\boldsymbol{c})p(\boldsymbol{c}|\boldsymbol{b}) \tag{13}$$

$$= \sum_{\boldsymbol{b}_{-i},\boldsymbol{c}} p(\boldsymbol{c}|\boldsymbol{b}) \prod_{k=1}^{K} \sum_{\boldsymbol{x}[k]} \int_{\boldsymbol{h}[k]} p(\boldsymbol{y}[k]|\boldsymbol{h}[k],\boldsymbol{x}[k]) \tag{14}$$

$$\times \left[\prod_{l=0}^{L-1} p(h_l[k])\right]\left[\prod_{n=0}^{N_\mathrm{D}-1} p(x_{N_\mathrm{P}+n}[k]|\boldsymbol{c}_{(k-1)N_\mathrm{D}+n})\right]$$

where $\boldsymbol{b}_{-i} \triangleq [b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_{N_\mathrm{b}}]^\mathsf{T}$. Above, (12) follows from Bayes rule and the assumption that the information bits $\boldsymbol{b}$ are uniformly distributed; (13) follows from the dependency relationships among the random vectors $\boldsymbol{y}$, $\boldsymbol{h} \triangleq [\boldsymbol{h}[1]^\mathsf{T}, \ldots, \boldsymbol{h}[K]^\mathsf{T}]^\mathsf{T}$, $\boldsymbol{x} \triangleq [\boldsymbol{x}[1]^\mathsf{T}, \ldots, \boldsymbol{x}[K]^\mathsf{T}]^\mathsf{T}$, $\boldsymbol{c}$, and $\boldsymbol{b}$; and (14) follows from the additive white Gaussian noise (AWGN) assumption on $\boldsymbol{y}$, the block-fading GMM model on $\boldsymbol{h}$, and the bit-to-symbol mapping. In particular,

$$p(\boldsymbol{y}[k]|\boldsymbol{h}[k],\boldsymbol{x}[k]) = \mathcal{CN}\big(\boldsymbol{y}[k]; \mathrm{Diag}(\boldsymbol{F}_M^{1:L}\boldsymbol{h}[k])\boldsymbol{F}_M\boldsymbol{x}[k], \sigma_w^2\boldsymbol{I}_M\big) \tag{15}$$

follows from (9), and $p(x_{N_\mathrm{P}+n}[k]|\boldsymbol{c}_{(k-1)N_\mathrm{D}+n})$ is determined by the bit-to-symbol mapping.

Equation (14) can be visualized using the bipartite factor graph in Fig. 3, where the open circles represent the variable nodes and the solid rectangles represent the factors in (14). For interpretability, the factor graph has been partitioned into two subgraphs: the left subgraph corresponds to SISO decoding, and the right subgraph to SISO equalization with an unknown channel.

A. Belief Propagation

The posterior bit marginals $\{p(b_i|\boldsymbol{y})\}_{i=1}^{N_\mathrm{b}}$ can in principle be computed from (14), but doing so is intractable for complexity reasons. An alternate approach is to perform BP using the SPA [11], where messages are passed along the edges of the factor graph in Fig. 3. These messages come in the form of pmfs for discrete-valued variables like $b_i, c_{n,a}, x_n$ and pdfs for continuous variables like h_l. If there were no loops (or "cycles") in the graph, BP would compute exact marginals. But since there are loops in Fig. 3, BP computes only approximate marginals, since exact computation is actually NP hard [59]. Still, we expect

[2] Likewise, in CP systems, when $N_\mathrm{G} \geq L-1$, (5) reduces to circular convolution with period $M - N_\mathrm{G}$. However, due to the advantages of UW and ZP over CP [58], we focus on the former.

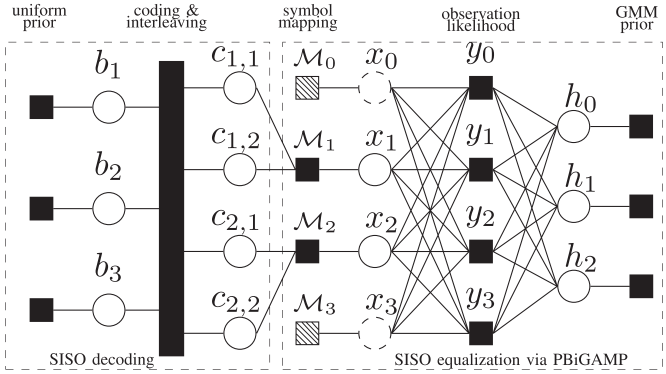

Fig. 3. The factor graph corresponding to (14) for a toy example with $K = 1$ blocks, $N_{\mathrm{b}} = 3$ information bits, 4 coded and interleaved bits, $A = 2$ bits/symbol, $N_{\mathrm{D}} = 2$ data symbols, one pilot symbol x_0, one guard symbol x_3, block length $M = 4$, and $L = 3$ channel taps. For simplicity, the block index "$[k]$" has been suppressed. Node y_m stands for $p(y_m|\boldsymbol{x}, \boldsymbol{h})$ and node $\mathcal{M}_i$ denotes the bit-to-symbol mapping.

good results from loopy BP, as it has been used successfully in many problems, such as turbo decoding [60], low-density parity-check (LDPC) decoding [61], turbo equalization [9], inference on Markov random fields [62], multiuser detection [45], and compressive sensing [43].

That said, for the factor graph in Fig. 3, and the SISO equalization subgraph in particular, exact implementation of the SPA is intractable. This is because the messages to and from the h_l nodes take the form of Gaussian mixtures with a mixture order than grows exponentially as the iterations progress. One might consider passing Gaussian approximations of those problematic SPA messages, an approach known as expectation propagation (EP) [63]. But since there are ML edges between the $\{h_l\}$ and $\{y_m\}$ nodes in Fig. 3, the per-symbol complexity of EP is $O(L)$ and not the desired $O(\log L)$. Also, the fixed-points of EP are not well understood.

As discussed in Section I, many approximate methods have been proposed for SISO equalization of SC block transmissions over unknown FS channels. But, to our knowledge, few have the $O(\log L)$ per-symbol complexity of conventional SC-FDE and none are able to handle a GMM channel prior. In the sequel, we propose a new method, based on PBiGAMP, that has both of these desirable features. Fig. 1(b) illustrates the proposed system diagram.

B. Review of PBiGAMP

PBiGAMP [52] is an efficient scheme to compute approximate marginal posteriors on independent random variables $\{\mathsf{x}_n\}_{n=0}^{N-1}$ and $\{\mathsf{h}_l\}_{l=0}^{L-1}$ from observations $\boldsymbol{y} = [y_0, \ldots, y_{M-1}]^{\mathsf{T}}$ under a likelihood model of the form

$$p(\boldsymbol{y}|\boldsymbol{z}) = \prod_{m=1}^{M} p_{\mathsf{y}_m|\mathsf{z}_m}(y_m|z_m) \tag{16a}$$

$$\mathsf{z}_m = \sum_{l=0}^{L-1}\sum_{n=0}^{N-1} \mathsf{x}_n z_m^{(n,l)} \mathsf{h}_l, \tag{16b}$$

TABLE I
THE PBiGAMP ALGORITHM FROM [52]

```
definitions:
  $p_{\mathsf{z}_m|\mathsf{p}_m}(z\,|\,\widehat{p};\nu^p) \triangleq \frac{p_{\mathsf{y}_m|\mathsf{z}_m}(y_m\,|\,z)\,\mathcal{CN}(z;\widehat{p},\nu^p)}{\int_{z'} p_{\mathsf{y}_m|\mathsf{z}_m}(y_m\,|\,z')\,\mathcal{CN}(z';\widehat{p},\nu^p)}$   (D1)
  $p_{\mathsf{h}_l|\mathsf{r}_l}(h\,|\,\widehat{r};\nu^r) \triangleq \frac{p_{\mathsf{h}_l}(h)\,\mathcal{CN}(h;\widehat{r},\nu^r)}{\int_{h'} p_{\mathsf{h}_l}(h')\,\mathcal{CN}(h';\widehat{r},\nu^r)}$   (D2)
  $p_{\mathsf{x}_n|\mathsf{q}_n}(x\,|\,\widehat{q};\nu^q) \triangleq \frac{p_{\mathsf{x}_n}(x)\,\mathcal{CN}(x;\widehat{q},\nu^q)}{\int_{x'} p_{\mathsf{x}_n}(x')\,\mathcal{CN}(\mathsf{x}';\widehat{q},\nu^q)}$   (D3)
initialization:
  $\forall m : \widehat{s}_m[0] = 0$   (I1)
  $\forall n,l :$ choose $\widehat{x}_n[1], \nu_n^x[1], \widehat{h}_l[1], \nu_l^h[1]$   (I2)
for $t = 1, \ldots T_{\max}$
  $\forall m,n : \widehat{z}_m^{(n,*)}[t] = \sum_{l=0}^{L-1} z_m^{(n,l)}\widehat{h}_l[t]$   (R1)
  $\forall m,l : \widehat{z}_m^{(*,l)}[t] = \sum_{n=0}^{N-1} \widehat{x}_n[t] z_m^{(n,l)}$   (R2)
  $\forall m : \widehat{z}_m^{(*,*)}[t] = \sum_{n=0}^{N-1} \widehat{x}_n[t]\widehat{z}_m^{(n,*)}$ or $\sum_{l=0}^{L-1} \widehat{h}_l[t]\widehat{z}_m^{(*,l)}[t]$   (R3)
  $\forall m : \bar{\nu}_m^p[t] = \sum_{n=0}^{N-1} \nu_n^x[t]|\widehat{z}_m^{(n,*)}[t]|^2$
      $+ \sum_{l=0}^{L-1} \nu_l^h[t]|\widehat{z}_m^{(*,l)}[t]|^2$   (R4)
  $\forall m : \nu_m^p[t] = \bar{\nu}_m^p[t] + \sum_{n=0}^{N-1} \nu_n^x[t] \sum_{l=0}^{L-1} \nu_l^h[t]|\widehat{z}_m^{(n,l)}[t]|^2$   (R5)
  $\forall m : \widehat{p}_m[t] = \widehat{z}_m^{(*,*)}[t] - \widehat{s}_m[t-1]\bar{\nu}_m^p[t]$   (R6)
  $\forall m : \nu_m^z[t] = \mathrm{var}\{\mathsf{z}_m\,|\,\mathsf{p}_m = \widehat{p}_m[t]; \nu_m^p[t]\}$   (R7)
  $\forall m : \widehat{z}_m[t] = \mathrm{E}\{\mathsf{z}_m\,|\,\mathsf{p}_m = \widehat{p}_m[t]; \nu_m^p[t]\}$   (R8)
  $\forall m : \nu_m^s[t] = (1 - \nu_m^z[t]/\nu_m^p[t])/\nu_m^p[t]$   (R9)
  $\forall m : \widehat{s}_m[t] = (\widehat{z}_m[t] - \widehat{p}_m[t])/\nu_m^p[t]$   (R10)
  $\forall l : \nu_l^r[t] = \Big(\sum_{m=0}^{M-1} \nu_m^s[t]|\widehat{z}_m^{(*,l)}[t]|^2\Big)^{-1}$   (R11)
  $\forall l : \widehat{r}_l[t] = \widehat{h}_l[t] + \nu_l^r[t]\sum_{m=0}^{M-1} \widehat{s}_m[t]\widehat{z}_m^{(*,l)*}[t]$
      $-\nu_l^r[t]\widehat{h}_l[t]\sum_{m=0}^{M-1} \nu_m^s[t]\sum_{n=0}^{N-1} \nu_n^x[t]|z_m^{(n,l)}|^2$   (R12)
  $\forall n : \nu_n^q[t] = \Big(\sum_{m=0}^{M-1} \nu_m^s[t]|\widehat{z}_m^{(n,*)}[t]|^2\Big)^{-1}$   (R13)
  $\forall n : \widehat{q}_n[t] = \widehat{x}_n[t] + \nu_n^q[t]\sum_{m=0}^{M-1} \widehat{s}_m[t]\widehat{z}_m^{(n,*)*}[t]$
      $-\nu_n^q[t]\widehat{x}_n[t]\sum_{m=0}^{M-1} \nu_m^s[t]\sum_{l=0}^{L-1} \nu_l^h[t]|z_m^{(n,l)}|^2$   (R14)
  $\forall l : \nu_l^h[t+1] = \mathrm{var}\{\mathsf{h}_l\,|\,\mathsf{r}_l = \widehat{r}_l[t]; \nu_l^r[t]\}$   (R15)
  $\forall l : \widehat{h}_l[t+1] = \mathrm{E}\{\mathsf{h}_l\,|\,\mathsf{r}_l = \widehat{r}_l[t]; \nu_l^r[t]\}$   (R16)
  $\forall n : \nu_n^x[t+1] = \mathrm{var}\{\mathsf{x}_n\,|\,\mathsf{q}_n = \widehat{q}_n[t]; \nu_n^q[t]\}$   (R17)
  $\forall n : \widehat{x}_n[t+1] = \mathrm{E}\{\mathsf{x}_n\,|\,\mathsf{q}_n = \widehat{q}_n[t]; \nu_n^q[t]\}$   (R18)
end
```

where $z_m^{(n,l)}$ are known. Throughout this subsection, we typeset random variables in san-serif font (e.g., y_m) and non-random variables in serif font (e.g., y_m) for clarity. Note that, in (16), z_m can be interpreted as noiseless bilinear measurements of the random vectors $\mathbf{x} \triangleq [\mathsf{x}_0, \ldots, \mathsf{x}_{N-1}]^{\mathsf{T}}$ and $\mathbf{h} \triangleq [\mathsf{h}_0, \ldots, \mathsf{h}_{L-1}]^{\mathsf{T}}$, and $p_{\mathsf{y}_m|\mathsf{z}_m}(y_m|z_m)$ can be interpreted as a noisy measurement channel. Applications of (16) include matrix compressive sensing, self-calibration, blind deconvolution, and joint channel/symbol estimation.

The PBiGAMP algorithm is stated in Table I. There, the priors on x_n and h_l are denoted by $p_{\mathsf{x}_n}(x_n)$ and $p_{\mathsf{h}_l}(h_l)$, respectively. The approximate marginal posteriors are denoted by $p_{\mathsf{x}_n|\mathsf{q}_n}(x_n|\widehat{q}_n;\nu_n^q)$ and $p_{\mathsf{h}_l|\mathsf{r}_l}(h_l|\widehat{r}_l;\nu_l^r)$, and specified in lines (D2)-(D3), where $\widehat{q}_n, \nu_n^q, \widehat{r}_l, \nu_l^r$ are quantities computed iteratively by PBiGAMP.

The PBiGAMP algorithm was derived in [52] as a computationally tractable approximation of the SPA for (16), under the assumption that $z_m^{(n,l)}$ are independent realizations of a zero-mean Gaussian random variable. In the large-system limit (i.e., $M, N, L \to \infty$ with fixed N/M and L/M) this approximation becomes exact. In [53], PBiGAMP was analyzed using the replica method from statistical physics. There it was shown that, in the large-system limit, the macroscopic performance of PBiGAMP is predicted by a scalar state-evolution. This state evolution was studied in detail for the case of i.i.d. Bernoulli-Gaussian x_n and h_l and found to exhibit a sharp phase-transition behavior. In particular, for some combinations of measurement rates N/M and L/M and sparsity rates on x_n and h_l, PBiGAMP converges to the MMSE estimates of $\boldsymbol{x}$ and $\boldsymbol{h}$. For more difficult combinations of rates, PBiGAMP does not yield accurate estimates, but it is conjectured that no other polynomial-time method will yield accurate estimates for those rates.

C. SISO Equalization via PBiGAMP

We now detail how PBiGAMP can be applied to SISO equalization of SC block transmissions over unknown FS channels. The first step is to specialize the PBiGAMP model (16) to the SC block-transmission model (9). In doing so, we omit the block index notation "$[k]$" because PBiGAMP can be applied to each block separately, as implied by the factorization (14). Rewriting (9) as

$$y_m = \sum_{n=0}^{M-1} \sum_{l=0}^{L-1} x_n f_M^{mn} f_M^{ml} h_l + w_m \tag{17}$$

for $m = 0, \ldots, M-1$, where $f_M^{mn} = e^{-\mathrm{j}\frac{2\pi}{M}mn}/\sqrt{M}$ is the $(m+1, n+1)$th element of the unitary M-DFT matrix, we see that the PBiGAMP quantities in (16) become (with $N = M$)

$$p_{\mathsf{y}_m|\mathsf{z}_m}(y_m|z_m) = \mathcal{CN}(y_m; z_m, \sigma_w^2) \tag{18}$$

$$z_m^{(n,l)} = f_M^{mn} f_M^{ml}. \tag{19}$$

PBiGAMP's prior on h_l takes the form of the GMM in (11). For data-symbol indices $n \in \{N_\mathrm{P}, \ldots, N_\mathrm{P} + N_\mathrm{D} - 1\}$, PBiGAMP's prior on x_n takes the form of the pdf

$$p_{\mathsf{x}_n}(x_n) = \sum_{j=1}^{2^A} \gamma_{n,j} \delta(x_n - s^{(j)}), \tag{20}$$

where $\delta(\cdot)$ is the Dirac delta, $\{s^{(1)}, \ldots, s^{(2^A)}\} = \mathcal{S}$ is the data-symbol alphabet, and $\gamma_{n,j} = \Pr\{\mathsf{x}_n = s^{(j)}\}$ denotes the prior data-symbol pmf. Although the data symbols x_n are discrete, PBiGAMP treats them as random variables in $\mathbb{C}$. For pilot indices $n \in \{0, \ldots, N_\mathrm{P} - 1\}$ and guard indices $n \in \{N_\mathrm{P} + N_\mathrm{D}, \ldots, M-1\}$, the PBiGAMP prior $p_{\mathsf{x}_n}(x_n)$ is trivial because the pilots and guards are known.

The data-symbol pmf is determined by the coded-bit priors $\Pr\{\mathsf{c}_{n,a} = c_a^{(j)}\}$ coming from the SISO decoder:

$$\Pr\{\mathsf{x}_n = s^{(j)}\} = \sum_{j'=1}^{2^A} \Pr\{\mathsf{x}_n = s^{(j)}, \mathbf{\mathsf{c}}_n = \boldsymbol{c}^{(j')}\} \tag{21}$$

$$= \sum_{j'=1}^{2^A} \underbrace{\Pr\{\mathsf{x}_n = s^{(j)} | \mathbf{\mathsf{c}}_n = \boldsymbol{c}^{(j')}\}}_{\delta_{j-j'}} \Pr\{\mathbf{\mathsf{c}}_n = \boldsymbol{c}^{(j')}\} \tag{22}$$

$$= \Pr\{\mathbf{\mathsf{c}}_n = \boldsymbol{c}^{(j)}\} = \prod_{a=1}^{A} \Pr\{\mathsf{c}_{n,a} = c_a^{(j)}\}, \tag{23}$$

where $\boldsymbol{c}^{(j)} = [c_1^{(j)}, \ldots, c_A^{(j)}]^\mathsf{T} \in \{0,1\}^A$ is the coded-bit sequence corresponding to the symbol value $s^{(j)}$ and δ_j is the Kronecker delta sequence.

Given the above, we are now in a position to apply PBiGAMP from Table I. For brevity, we omit the iteration index "$[t]$." To start, (19) and the definition of f_M^{mn} imply that the quantities in lines (R1)-(R3) of Table I become

$$\widehat{\boldsymbol{z}}^{(n,*)} = \mathrm{Diag}(\boldsymbol{f}_M^n) \boldsymbol{F}_M^{1:L} \widehat{\boldsymbol{h}} \tag{24}$$

$$\widehat{\boldsymbol{z}}^{(*,l)} = \mathrm{Diag}(\boldsymbol{f}_M^l) \boldsymbol{F}_M \widehat{\boldsymbol{x}} \tag{25}$$

$$\widehat{\boldsymbol{z}}^{(*,*)} = \mathrm{Diag}(\boldsymbol{F}_M \widehat{\boldsymbol{x}}) \boldsymbol{F}_M^{1:L} \widehat{\boldsymbol{h}}. \tag{26}$$

The quantities in lines (R4)-(R6) then become

$$\bar{\boldsymbol{\nu}}^p = |\boldsymbol{F}_M^{1:L} \widehat{\boldsymbol{h}}|^{\odot 2} \left(\frac{1}{M} \sum_{n=0}^{N-1} \nu_n^x \right) + |\boldsymbol{F}_M \widehat{\boldsymbol{x}}|^{\odot 2} \left(\frac{1}{M} \sum_{l=0}^{L-1} \nu_l^h \right)$$

$$\boldsymbol{\nu}^p = \bar{\boldsymbol{\nu}}^p + \underbrace{\left(\frac{1}{M} \sum_{n=0}^{M-1} \nu_n^x \right)}_{\triangleq\, \nu^x} \underbrace{\left(\frac{1}{M} \sum_{l=0}^{L-1} \nu_l^h \right)}_{\triangleq\, \nu^h} \mathbf{1}_M \tag{27}$$

$$\widehat{\boldsymbol{p}} = (\boldsymbol{F}_M \widehat{\boldsymbol{x}}) \odot (\boldsymbol{F}_M^{1:L} \widehat{\boldsymbol{h}}) - \widehat{\boldsymbol{s}} \odot \bar{\boldsymbol{\nu}}^p, \tag{28}$$

where $\odot$ denotes elementwise multiplication of two vectors and $|\cdot|^{\odot 2}$ denotes element-wise absolute-value squared. Note that, since the individual terms in $\{\nu_n^x\}$ and $\{\nu_l^h\}$ are not used, it suffices to track the average quantities ν^x and ν^h.

Due to the Gaussian form of the likelihood in (18), the posterior mean and variance computations in (R8) and (R7) reduce to

$$\widehat{\boldsymbol{z}}_m = \boldsymbol{\nu}^z \odot (\boldsymbol{y}/\sigma_w^2 + \widehat{\boldsymbol{p}} \oslash \boldsymbol{\nu}^p) \tag{29}$$

$$\boldsymbol{\nu}^z = \sigma_w^2 \boldsymbol{\nu}^p \oslash (\sigma_w^2 + \boldsymbol{\nu}^p), \tag{30}$$

where $\oslash$ denotes elementwise division of two vectors.

Given the form of $\widehat{\boldsymbol{z}}^{(n,*)}$ and $\widehat{\boldsymbol{z}}^{(*,l)}$ in (24)-(25), the quantities in (R11) and (R13) become

$$\boldsymbol{\nu}^r = \mathbf{1}_L M / (\boldsymbol{\nu}^{s\mathsf{T}} |\boldsymbol{F}_M \widehat{\boldsymbol{x}}|^{\odot 2}) \triangleq \nu^r \mathbf{1}_L \tag{31}$$

$$\boldsymbol{\nu}^q = \mathbf{1}_M M / (\boldsymbol{\nu}^{s\mathsf{T}} |\boldsymbol{F}_M^{1:L} \widehat{\boldsymbol{h}}|^{\odot 2}) \triangleq \nu^q \mathbf{1}_M, \tag{32}$$

where it suffices to track the scalars ν^r and ν^q. It can then be shown that the quantities in (R12) and (R14) become

$$\widehat{\boldsymbol{r}} = \widehat{\boldsymbol{h}} + \nu^r \boldsymbol{F}_M^{1:L\,\mathsf{H}} ((\boldsymbol{F}_M \widehat{\boldsymbol{x}})^* \odot \widehat{\boldsymbol{s}}) - \nu^s \nu^x \nu^r \widehat{\boldsymbol{h}} \tag{33}$$

$$\widehat{\boldsymbol{q}} = \widehat{\boldsymbol{x}} + \nu^q \boldsymbol{F}_M^\mathsf{H} ((\boldsymbol{F}_M^{1:L} \widehat{\boldsymbol{h}})^* \odot \widehat{\boldsymbol{s}}) - \nu^s \nu^h \nu^q \widehat{\boldsymbol{x}} \tag{34}$$

for

$$\nu^s \triangleq \frac{1}{M} \sum_{m=0}^{M-1} \nu_m^s. \tag{35}$$

To see this, we can start with (R12) to get

$$\widehat{r}_l = \widehat{h}_l + \nu_l^r \sum_{m=0}^{M-1} \widehat{s}_m \widehat{z}_m^{(*,l)*} - \nu_l^r \widehat{h}_l \sum_{m=0}^{M-1} \nu_m^s \sum_{n=0}^{N-1} \nu_n^x |z_m^{(n,l)}|^2$$

$$= \widehat{h}_l + \nu^r [\widehat{\boldsymbol{z}}^{(*,l)}]^\mathsf{H} \widehat{\boldsymbol{s}} - \nu^r \nu^s \nu^x \widehat{h}_l \tag{36}$$

$$= \widehat{h}_l + \nu^r (\boldsymbol{F}_M \widehat{\boldsymbol{x}})^\mathsf{H} \mathrm{Diag}(\boldsymbol{f}_M^l)^* \widehat{\boldsymbol{s}} - \nu^r \nu^s \nu^x \widehat{h}_l \tag{37}$$

$$= \widehat{h}_l + \nu^r (\boldsymbol{f}_M^l)^\mathsf{H} \big((\boldsymbol{F}_M \widehat{\boldsymbol{x}})^* \odot \widehat{\boldsymbol{s}} \big) - \nu^r \nu^s \nu^x \widehat{h}_l \tag{38}$$

where in (36) we used the facts that $\nu_l^r = \nu^r\ \forall l$ and $|z_m^{(n,l)}| = 1/M\ \forall m, n, l$ and the definitions of ν^x from (27) and of ν^s

from (35). For (37), we used (25). Since $(\boldsymbol{f}_M^l)^{\mathsf{H}}$ is the lth row of $\boldsymbol{F}_M^{\mathsf{H}}$, stacking (38) vertically for $l = 0, \ldots, L-1$ yields (33). A similar derivation yields (34).

Lines (R15)-(R16) compute the mean and variance of the approximate posterior on h_l defined in (D2). As a result of the GMM prior (11), this approximate posterior is [51]

$$p_{\mathsf{h}_l\,|\,\mathsf{r}_l}(h_l\,|\,\widehat{r}_l;\nu_l^r) = \sum_{d=1}^{D}\bar{\lambda}_{l,d}\mathcal{CN}\left(h_l;\frac{\nu_{l,d}\widehat{r}_l}{\nu_{l,d}+\nu_l^r},\frac{\nu_{l,d}\nu_l^r}{\nu_{l,d}+\nu_l^r}\right) \tag{39}$$

$$\bar{\lambda}_{l,d} = \frac{\lambda_{l,d}\mathcal{CN}(\widehat{r}_l;0,\nu_{l,d}+\nu_l^r)}{\sum_{d'=1}^{D}\lambda_{l,d'}\mathcal{CN}(\widehat{r}_l;0,\nu_{l,d'}+\nu_l^r)}, \tag{40}$$

which is also a GMM. The corresponding mean and variance follow straightforwardly as

$$\widehat{h}_l = \sum_{d=1}^{D}\bar{\lambda}_{l,d}\frac{\nu_{l,d}\widehat{r}_l}{\nu_{l,d}+\nu_l^r} \tag{41}$$

$$\nu_l^h = \sum_{d=1}^{D}\bar{\lambda}_{l,d}\left(\frac{\nu_{l,d}\nu_l^r}{\nu_{l,d}+\nu_l^r} + \left|\frac{\nu_{l,d}\widehat{r}_l}{\nu_{l,d}+\nu_l^r}\right|^2\right) - |\widehat{h}_l|^2. \tag{42}$$

Finally, lines (R17)-(R18) compute the mean and variance of the approximate posterior on x_n defined in (D3). As a result of the discrete prior (23), the approximate posterior is

$$p_{\mathsf{x}_n\,|\,\mathsf{q}_n}(x_n\,|\,\widehat{q}_n;\nu_n^q) = \sum_{j=1}^{2^A}\bar{\gamma}_{n,j}\delta(x_n - s^{(j)}) \tag{43}$$

$$\bar{\gamma}_{n,j} = \frac{\Pr\{\mathsf{x}_n = s^{(j)}\}\mathcal{CN}\big(s^{(j)};\widehat{q}_n,\nu_n^q\big)}{\sum_{j'=1}^{2^A}\Pr\{\mathsf{x}_n = s^{(j')}\}\mathcal{CN}\big(s^{(j')};\widehat{q}_n,\nu_n^q\big)}, \tag{44}$$

which is again a discrete distribution with support on $\mathcal{S}$. The posterior mean and variance of x_n follow straightforwardly as

$$\widehat{x}_n = \sum_{j=1}^{2^A}\bar{\gamma}_{n,j}s^{(j)} \tag{45}$$

$$\nu_n^x = \sum_{j=1}^{2^A}\bar{\gamma}_{n,j}|s^{(j)} - \widehat{x}_n|^2. \tag{46}$$

Note that $\{\bar{\gamma}_{n,j}\}_{j=1}^{2^A}$ is the posterior pmf on x_n. It can be converted to posterior pmfs on the coded bits $\{\mathsf{c}_{n,a}\}_{a=1}^{A}$ via

$$\Pr\{\mathsf{c}_{n,a} = 1\,|\,\widehat{q}_n\} = \sum_{j=1\ldots2^A\,|\,c_a^{(j)}=1}\Pr\{\boldsymbol{\mathsf{c}}_n = \boldsymbol{c}^{(j)}\,|\,\widehat{q}_n\} \tag{47}$$

$$= \sum_{\substack{j\,=\,1\ldots2^A \\ c_a^{(j)}\,=\,1}}\sum_{j'=1}^{2^A}\underbrace{\Pr\{\boldsymbol{\mathsf{c}}_n = \boldsymbol{c}^{(j)}|\mathsf{x}_n = s^{(j')}\}}_{\delta_{j-j'}}\underbrace{\Pr\{\mathsf{x}_n = s^{(j')}\,|\,\widehat{q}_n\}}_{\bar{\gamma}_{n,j'}} \tag{48}$$

$$= \sum_{j=1\ldots2^A\,|\,c_a^{(j)}=1}\bar{\gamma}_{n,j}. \tag{49}$$

TABLE II
SISO Equalization via PBiGAMP

```
inputs:
    symbol priors $\{\gamma_{n,j}\}$, AWGN variance $\sigma_w^2$,
    GMM-state priors $\{\lambda_{l,d}\}$, GMM variances $\{\nu_{l,d}\}$,
    maximum iterations $T_{\max}$, initial channel estimate $(\widehat{\boldsymbol{h}}_{\text{init}}, \nu_{\text{init}}^h)$
outputs:
    symbol posteriors $\{\bar{\gamma}_{n,j}[T_{\max}]\}$, GMM-state posteriors $\{\bar{\lambda}_{l,d}[T_{\max}]\}$
initialization:
        $\widehat{\boldsymbol{x}}[1] = [\boldsymbol{x}_{\text{P}}^{\mathsf{T}}, \mathbf{0}_{N_{\text{D}}}^{\mathsf{T}}, \boldsymbol{x}_{\text{G}}^{\mathsf{T}}]^{\mathsf{T}},\ \nu^x[1] = \frac{N_{\text{D}}}{M}$
        $\widehat{\boldsymbol{h}}[1] = \widehat{\boldsymbol{h}}_{\text{init}},\ \nu^h[1] = \nu_{\text{init}}^h,\ \widehat{\boldsymbol{s}}[0] = \mathbf{0}_M$
for $t = 1, \ldots T_{\max}$
        $\underline{\widehat{\boldsymbol{h}}}[t] = \boldsymbol{F}_M^{1:L}\widehat{\boldsymbol{h}}[t]$    (B1)
        $\underline{\widehat{\boldsymbol{x}}}[t] = \boldsymbol{F}_M\widehat{\boldsymbol{x}}[t]$    (B2)
        $\bar{\boldsymbol{\nu}}^p[t] = |\underline{\widehat{\boldsymbol{h}}}[t]|^{\odot 2}\nu^x[t] + |\underline{\widehat{\boldsymbol{x}}}[t]|^{\odot 2}\nu^h[t]$    (B3)
        $\boldsymbol{\nu}^p[t] = \bar{\boldsymbol{\nu}}^p[t] + \nu^x[t]\nu^h[t]\mathbf{1}_M$    (B4)
        $\widehat{\boldsymbol{p}}[t] = \underline{\widehat{\boldsymbol{h}}}[t] \odot \underline{\widehat{\boldsymbol{x}}}[t] - \widehat{\boldsymbol{s}}[t-1] \odot \bar{\boldsymbol{\nu}}^p[t]$    (B5)
        $\boldsymbol{\nu}^z[t] = \sigma_w^2\boldsymbol{\nu}^p[t] \oslash (\sigma_w^2 + \boldsymbol{\nu}^p[t])$    (B6)
        $\widehat{\boldsymbol{z}}[t] = \boldsymbol{\nu}^z[t] \odot (\boldsymbol{y}/\sigma_w^2 + \widehat{\boldsymbol{p}}[t] \oslash \boldsymbol{\nu}^p[t])$    (B7)
        $\boldsymbol{\nu}^s[t] = \big(\mathbf{1}_M - \boldsymbol{\nu}^z[t] \oslash \boldsymbol{\nu}^p[t]\big) \oslash \boldsymbol{\nu}^p[t]$    (B8)
        $\nu^s[t] = (\mathbf{1}_M^{\mathsf{T}}\boldsymbol{\nu}^s[t])/M$    (B9)
        $\widehat{\boldsymbol{s}}[t] = \big(\widehat{\boldsymbol{z}}[t] - \widehat{\boldsymbol{p}}[t]\big) \oslash \boldsymbol{\nu}^p[t]$    (B10)
        $\nu^r[t] = M/\big(\boldsymbol{\nu}^s[t]^{\mathsf{T}}|\underline{\widehat{\boldsymbol{x}}}[t]|^{\odot 2}\big)$    (B11)
        $\widehat{\boldsymbol{r}}[t] = \widehat{\boldsymbol{h}}[t] + \nu^r[t]\Big(\boldsymbol{F}_M^{1:L\,\mathsf{H}}\big(\underline{\widehat{\boldsymbol{x}}}[t]^* \odot \widehat{\boldsymbol{s}}[t]\big) - \nu^s[t]\nu^x[t]\widehat{\boldsymbol{h}}[t]\Big)$    (B12)
        $\nu^q[t] = M/\big(\boldsymbol{\nu}^s[t]^{\mathsf{T}}|\underline{\widehat{\boldsymbol{h}}}[t]|^{\odot 2}\big)$    (B13)
        $\widehat{\boldsymbol{q}}[t] = \widehat{\boldsymbol{x}}[t] + \nu^q[t]\Big(\boldsymbol{F}_M^{\mathsf{H}}\big(\underline{\widehat{\boldsymbol{h}}}[t]^* \odot \widehat{\boldsymbol{s}}[t]\big) - \nu^s[t]\nu^h[t]\widehat{\boldsymbol{x}}[t]\Big)$    (B14)
    $\forall l, d : \bar{\lambda}_{l,d}[t] = \frac{\lambda_{l,d}\mathcal{CN}(\widehat{r}_l[t];0,\nu_{l,d}+\nu_l^r[t])}{\sum_{d'=1}^{D}\lambda_{l,d'}\mathcal{CN}(\widehat{r}_l[t];0,\nu_{l,d'}+\nu_l^r[t])}$    (B14)
    $\forall l : \widehat{h}_l[t+1] = \sum_{d=1}^{D}\bar{\lambda}_{l,d}[t]\frac{\nu_{l,d}\widehat{r}_l[t]}{\nu_{l,d}+\nu_l^r[t]}$    (B15)
        $\nu^h[t+1] = \sum_{l=0}^{L-1}\Big[\sum_{d=1}^{D}\bar{\lambda}_{l,d}[t]\Big(\frac{\nu_{l,d}\nu_l^r[t]}{\nu_{l,d}+\nu_l^r[t]} + \Big|\frac{\nu_{l,d}\widehat{r}_l[t]}{\nu_{l,d}+\nu_l^r[t]}\Big|^2\Big)$
                $- |\widehat{h}_l[t+1]|^2\Big]\frac{1}{M}$    (B16)
    $\forall n, j : \bar{\gamma}_{n,j}[t] = \frac{\gamma_{n,j}\mathcal{CN}(s^{(j)};\widehat{q}_n[t],\nu_n^q[t])}{\sum_{j'=1}^{2^A}\gamma_{n,j'}\mathcal{CN}(s^{(j')};\widehat{q}_n[t],\nu_n^q[t])}$    (B17)
    $\forall n : \widehat{x}_n[t+1] = \sum_{j=1}^{2^A}\bar{\gamma}_{n,j}[t]s^{(j)}$    (B18)
        $\nu^x[t+1] = \frac{1}{M}\sum_{n=0}^{M-1}\sum_{j=1}^{2^A}\bar{\gamma}_{n,j}[t]\big|s^{(j)} - \widehat{x}_n[t+1]\big|^2$    (B19)
end
```

The PBiGAMP-based SISO equalization procedure is summarized in Table II.[3] Its complexity is dominated by the DFT-matrix multiplies in lines (B1), (B2), (B12), and (B14), which consume $O(M\log M)$ operations per block, or $O(\log M)$ per symbol, when an FFT is used. This complexity is often stated as "$O(\log L)$" in the literature, because the block length M is chosen proportional to the delay spread L. The complexity of all other lines in Table II consumes only $O(M)$ operations per block, or $O(1)$ per symbol.

D. Turbo Equalization

As described in Section III-A, we compute (approximate) posterior marginal bit probabilities $\{p(b_i|\boldsymbol{y})\}_{i=1}^{N_{\text{b}}}$ using the SPA, which is the usual approach to "turbo equalization" [9]. However, because exact SPA is intractable for the SISO-equalization subgraph in Fig. 3, we propose to use the PBiGAMP approximation described in Section III-B. We now detail the remaining steps in the SPA, for completeness.

Roughly speaking, we pass messages on the factor graph in Fig. 3 from the left to the right and back again. One such forward-backward pass will be referred to as a "turbo iteration."

[3] For simplicity, Table II assumes that the pilot and guards are in $\mathcal{S}$.

During a single turbo iteration, we alternate SISO equalization using PBiGAMP and SISO decoding using an off-the-shelf decoder/interleaver (see, e.g., [61], [64]). The SPA dictates that we pass "extrinsic" information between nodes on the graph, and hence between the subgraphs in Fig. 3. For a discrete random variable, the extrinsic message is a pmf formed by dividing the posterior pmf by the prior pmf. Details are given below.

On a typical turbo iteration, extrinsic information on the coded bits $\mathsf{c}_{n,a}$ is passed from the SISO decoder to PBiGAMP, where it is treated as prior information in (23) to determine the symbol priors $\gamma_{n,j}$. PBiGAMP is then run to convergence, generating the symbol posteriors $\bar{\gamma}_{n,j}$. The symbol posteriors are used in (49) to determine the coded-bit posteriors, which are then converted to extrinsic form and passed to the SISO decoder. The SISO decoder accepts this extrinsic information from PBiGAMP, treating it as a prior on the coded bits. It then computes posteriors on the coded bits, converts them to extrinsic form, and passes them to PBiGAMP for the next turbo iteration.

E. Learning the Channel Prior

The GMM denoising (41)-(42) assumes knowledge of the GMM prior (11). We now suggest two ways to learn this prior in practice.

If the receiver has successfully decoded transmissions over many channel realizations from a statistically common ensemble, then it has access to many (estimated) channel realizations $\{\widehat{\boldsymbol{h}}\}$. From these realizations, one could fit the parameters of a GMM prior using, say, the standard EM-based approach to fitting a GMM [65, p. 435].

In the case where very few (e.g., one) data blocks are available to learn the prior, the above approach may not be appropriate. As an alternative, an "EM-GM-AMP"-like approach could be used. First consider the simple case where the channel coefficients $\{\mathsf{h}_l\}_{l=0}^{L-1}$ are modeled as identically distributed, in which case the statistical parameters in (11) reduce to the GMM weights and variances $\{\lambda_d, \nu_d\}_{d=1}^{D}$. The paper [51] showed how these GMM parameters can be learned from the observations $\boldsymbol{y}$ using a combination of EM and AMP, and [52] showed how EM can be combined with PBiGAMP in a similar manner. More generally, one could partition the coefficients $\{\mathsf{h}_l\}_{l=0}^{L-1}$ into subsets and learn different parameters for each subset. In Section V, we investigate the performance of this EM-GM-PBiGAMP method for the Saleh-Valenzuela channels described in Section II-B.

IV. Benchmark Schemes

In this section, we briefly review several other approaches to SISO equalization that will be used as benchmarks.

A. TD-DFE-LS

First we review the "time-domain (TD) decision-feedback equalization (DFE) with least-squares (LS) channel estimation" scheme from [37]. At turbo iteration $i = 1, 2, \ldots$, soft symbol estimates $\widetilde{\boldsymbol{x}}^{(i)}$ are constructed as

$$\widetilde{\boldsymbol{x}}^{(i)} = \boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}^{(i)}\boldsymbol{y} - \boldsymbol{G}^{(i)}\widehat{\boldsymbol{x}}^{(i-1)}, \tag{50}$$

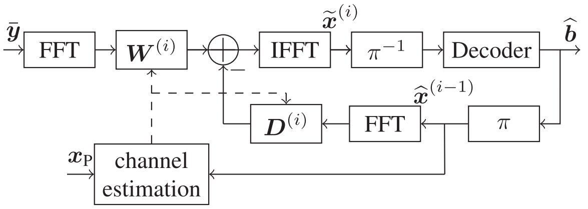
$\bar{\boldsymbol{y}}$
FFT
$\boldsymbol{W}^{(i)}$
$-$
IFFT
$\widetilde{\boldsymbol{x}}^{(i)}$
π^{-1}
Decoder
$\widehat{\boldsymbol{b}}$
$\widehat{\boldsymbol{x}}^{(i-1)}$
$\boldsymbol{D}^{(i)}$
FFT
π
$\boldsymbol{x}_{\mathrm{P}}$
channel estimation

Fig. 4. Implementation of the TD-DFE-LS and FD-DFE-LS approaches. Here, i denotes the turbo iteration and both $\boldsymbol{W}^{(i)}$ and $\boldsymbol{D}^{(i)}$ are diagonal matrices.

where $\boldsymbol{W}^{(i)}$ is a diagonal (frequency-domain forward filter) matrix, $\boldsymbol{G}^{(i)}$ is a circulant (time-domain feedback filter) matrix, and $\widehat{\boldsymbol{x}}^{(i-1)}$ are the symbol means computed from the most recent decoder outputs, as in (45). The first iteration uses $\widehat{\boldsymbol{x}}^{(0)} \triangleq \boldsymbol{0}$. The coefficients in $\boldsymbol{W}^{(i)}$ and $\boldsymbol{G}^{(i)}$ are designed to approximately minimize the MSE of $\widetilde{\boldsymbol{x}}^{(i)}$ under the constraint $\text{Diag}(\boldsymbol{G}^{(i)}) = \boldsymbol{0}$, a time-domain channel estimate $\widehat{\boldsymbol{h}}$, and an estimate of the error variance in $\widehat{\boldsymbol{h}}$. Decision-directed LS channel estimation is used (except for the first turbo iteration, where only pilots are used), and the estimation error variance is approximated by assuming perfect decoding. Because $\boldsymbol{G}^{(i)}$ is circulant, (50) can be implemented in the frequency domain with $O(\log L)$ per-symbol complexity, as illustrated in Fig. 4 with diagonal $\boldsymbol{D}^{(i)} = \boldsymbol{F}\boldsymbol{G}^{(i)}\boldsymbol{F}^{\mathsf{H}}$. Given a channel estimate, the design of $\boldsymbol{G}^{(i)}$ costs $O(L)$ operations per symbol via circulant matrix embedding. But LS channel estimation raises the per-symbol complexity to $O(L^2)$.

B. FD-DFE-LS

Next we review the "frequency-domain (FD) DFE with LS channel estimation" from [38]. Like the TD-DFE-LS, the FD-DFE-LS can be implemented as shown in Fig. 4. But, unlike the TD-DFE-LS, the (diagonal, approximately MMSE) feedforward and feedback filter matrices $\boldsymbol{W}^{(i)}$ and $\boldsymbol{D}^{(i)}$ are designed directly in the frequency domain. The FD-DFE-LS channel estimation procedure is also different: a decision-directed LS estimate is computed in the frequency domain, converted to the TD via inverse FFT (IFFT), truncated to length L, and converted back to the FD via FFT. For the first turbo iteration, the channel estimate is computed using only pilots. In total, the per-symbol complexity of channel estimation, filter design, and symbol estimation is $O(\log L)$ for FD-DFE-LS, which is much more efficient than TD-DFE-LS. However, the channel estimates of FD-DFE-LS are of lower quality, and their error variance is not used in FD-DFE-LS filter design.

C. SparseLift

SparseLift [57] is a recently proposed convex method that is applicable to joint sparse-channel-estimation/equalization. Its goal is to jointly recover the vector $\boldsymbol{x}$ and the sparse vector $\boldsymbol{h}$ from noisy observations $\boldsymbol{y} = \text{Diag}(\boldsymbol{B}\boldsymbol{h})\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are known. It works by first rewriting $\boldsymbol{y} = \boldsymbol{\Phi}\text{vec}(\boldsymbol{h}\boldsymbol{x}^{\mathsf{T}}) + \boldsymbol{w}$ with $\boldsymbol{\Phi} \triangleq [\text{Diag}(\boldsymbol{a}_1)\boldsymbol{B} \cdots \text{Diag}(\boldsymbol{a}_M)\boldsymbol{B}]$, where $\boldsymbol{a}_m$ is the mth column of $\boldsymbol{A}$. It then uses LASSO [66] to estimate sparse

$\text{vec}(\boldsymbol{h}\boldsymbol{x}^{\mathsf{T}})$ from $\boldsymbol{y}$. The estimate of $\text{vec}(\boldsymbol{h}\boldsymbol{x}^{\mathsf{T}})$ is then reshaped into a matrix and its principal rank-one component is extracted, yielding an estimate of $\boldsymbol{h}\boldsymbol{x}^{\mathsf{T}}$, which implies estimates of $\boldsymbol{h}$ and $\boldsymbol{x}$ up to a scalar ambiguity. An analysis of the noiseless case shows that, under sufficiently sparse $\boldsymbol{h}$, sufficiently tall $\boldsymbol{A}$, and sufficiently random $\boldsymbol{B}$, SparseLift exactly recovers $\boldsymbol{h}\boldsymbol{x}^{\mathsf{T}}$ with high probability [57] . Further analysis shows that SparseLift is robust to the presence of bounded noise [57].

The computational complexity of SparseLift is dominated by its LASSO stage. For LASSO, efficient algorithms like SPGL1 [67] can be used, which iterate matrix-vector multiplications with $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^{\mathsf{H}}$ and soft thresholding steps.

Equation (9) suggests that a direct application of SparseLift to SC channel-estimation/equalization would use $\boldsymbol{A} = \boldsymbol{F}_M$, $\boldsymbol{B} = \boldsymbol{F}_M^{1:L}$, and $\boldsymbol{x} = [\boldsymbol{x}_{\text{P}}^{\mathsf{T}}, \boldsymbol{x}_{\text{D}}^{\mathsf{T}}, \boldsymbol{x}_{\text{G}}^{\mathsf{T}}]^{\mathsf{T}}$. Our experiments with this approach, and several variations, are discussed in Section V-C.

V. Numerical Results

We now present numerical results comparing the proposed PBiGAMP scheme with the benchmark schemes discussed in Section IV. As a reference, we also consider the performance of PBiGAMP with perfect channel-state information (PCSI). In this latter case, PBiGAMP reduces to AMP. The application of AMP methods to soft symbol estimation can be traced back to early works like [45]–[47].

Unless otherwise noted, the following setup was used in our simulations. Recalling the SC block-transmission model from Section II-A, $N_{\text{b}} = 512$ information bits were coded at rate $R = 1/3$ by an irregular low-density parity-check (LDPC) code with average column weight 3. The 1540 coded bits were randomly interleaved and Gray-mapped to 16-QAM (i.e., $A = 4$), resulting in $N_{\text{D}} = 385$ data symbols. The data-symbol sequence $\boldsymbol{x}_{\text{D}}[1]$ was merged with a $N_{\text{P}} = 64$-length pilot sequence $\boldsymbol{x}_{\text{P}}$ and an $N_{\text{G}} = 63$-length guard sequence $\boldsymbol{x}_{\text{G}}$ to yield a single (i.e., $K = 1$) transmitted block $\boldsymbol{x}[1] = [\boldsymbol{x}_{\text{P}}^{\mathsf{T}}, \boldsymbol{x}_{\text{D}}[1]^{\mathsf{T}}, \boldsymbol{x}_{\text{G}}^{\mathsf{T}}]^{\mathsf{T}}$ of length $M = 512$. The spectral efficiency of the resulting scheme is thus 1 bit per channel use (bpcu). A Chu sequence [68] was used for the pilots, and two forms of guard were considered: zero-padding (ZP), where $\boldsymbol{x}_{\text{G}} = \boldsymbol{0}$; and unique-word (UW), where $\boldsymbol{x}_{\text{G}}$ repeats the last N_{G} symbols in $\boldsymbol{x}_{\text{P}}$.

The transmission block $\boldsymbol{x}[1]$ (see Fig. 2 for an illustration) was then modulated using a square-root raised cosine (SRRC) pulse with parameter 0.5, propagated through a noisy continuous-time channel, demodulated using a SRRC, and sampled at the baud rate, as in (1)–(5). Channel realizations were generated using the Saleh-Valenzuela model (10) with parameters specified in Appendix A, among them $L = 64$.

With PBiGAMP's GMM prior (11), we have several choices for how it is configured. First, we must choose the GMM order $D \geq 1$. Next, we must choose whether $\{\mathsf{h}_l\}$ are identically distributed or whether the GMM parameters vary with the lag l. Also, we must choose whether the coefficients $\{\mathsf{h}_l\}$ are treated as independent or coupled through a HMM, as in [49]. Finally, we must decide whether the hyper-parameters are learned from a large training corpus of 20 000 realizations of $\{\boldsymbol{h}\}$, or a the current measured block $\boldsymbol{y}$ using EM-GM-PBiGAMP. Our experiments will investigate the following combinations of such choices:

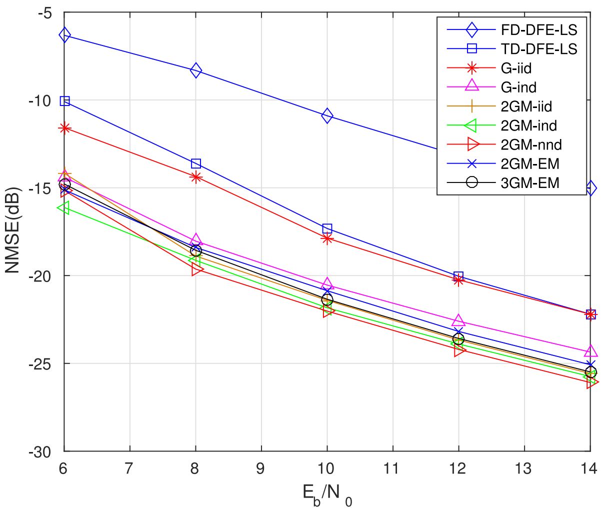

Fig. 5. Channel estimate NMSE versus E_b/N_0 for UW transmission.

- *G-iid:* $\{\mathsf{h}_l\}$ are modeled as i.i.d. zero-mean Gaussian. The variance is learned from a large training corpus $\{\boldsymbol{h}\}$.
- *2GM-iid:* $\{\mathsf{h}_l\}$ are modeled as i.i.d. from a zero-mean GMM with $D = 2$ components. The weights and variances are learned from a large training corpus $\{\boldsymbol{h}\}$.
- *2GM-EM:* $\{\mathsf{h}_l\}$ are modeled as i.i.d. from a GMM with $D = 2$ components. The weights and variances are learned from each $\boldsymbol{y}$ using the EM-GM-PBiGAMP approach from Section III-E.
- *3GM-EM:* Like 2GM-EM but with a $D = 3$-term GMM.
- *G-ind:* $\{\mathsf{h}_l\}$ are modeled as independent but non-identically distributed zero-mean Gaussian. The PDP $\{\nu_l\}$ is learned from a large training corpus $\{\boldsymbol{h}\}$.
- *2GM-ind:* $\{\mathsf{h}_l\}$ are modeled as independent but non-identically distributed from a zero-mean GMM with $D = 2$ components. The GMM parameters $\{\lambda_{l,1}, \nu_{l,1}, \nu_{l,2}\}_{l=0}^{L-1}$ are learned from a large training corpus $\{\boldsymbol{h}\}$.
- *2GM-nnd:* $\{\mathsf{h}_l\}$ are non-identical and non-independent across l, and obey a GMM/HMM with $D = 2$ components, as in [49]. The GMM/HMM parameters are learned from a large training corpus $\{\boldsymbol{h}\}$, as in [49].

Unless otherwise mentioned, all turbo methods were run for 20 iterations using the SISO LDPC decoder from [69].

A. Performance With UW Transmissions

We first investigate the performance of UW systems, where the pilot sequence is repeated as a guard sequence.

Fig. 5 shows the channel estimate's normalized mean squared error (NMSE) $\mathbb{E}\{\|\boldsymbol{h} - \widehat{\boldsymbol{h}}\|^2 / \|\boldsymbol{h}\|^2\}$ versus E_b/N_0 for the various schemes under test. There we see that the NMSE of TD-DFE-LS is significantly better than that of FD-DFE-LS, although—as explained earlier—this comes at the cost of $O(L^2)$ versus $O(\log L)$ implementation complexity. We also see that the NMSE of G-iid PBiGAMP is only slightly better than TD-DFE-LS. We attribute their similar performances to their use of

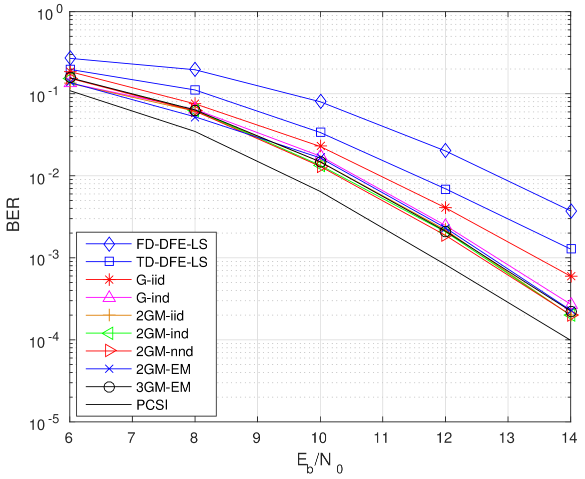

Fig. 6. BER versus E_b/N_0 for UW transmission and PBiGAMP with different channel models.

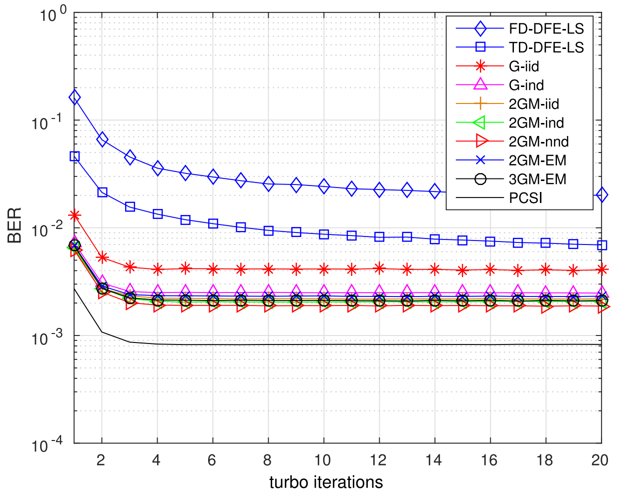

Fig. 7. BER versus turbo iteration for UW transmission at $E_b/N_0 = 12$ dB.

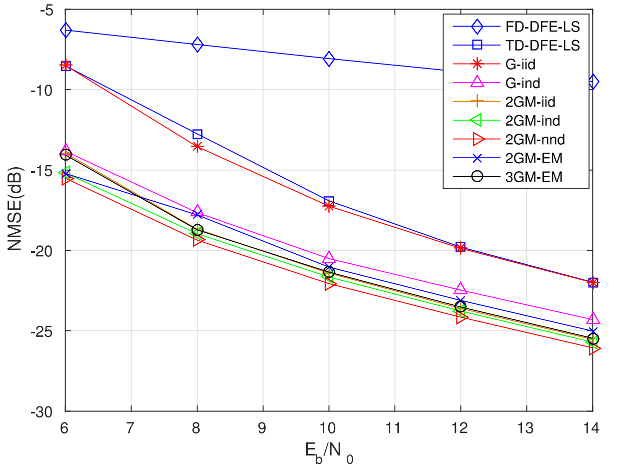

Fig. 8. Channel estimate NMSE versus E_b/N_0 for ZP transmission.

similarly simple channel models: TD-DFE-LS essentially uses no channel prior, while PBiGAMP's i.i.d. zero-mean Gaussian prior is parameterized by a single scalar variance. Still, PBiGAMP is advantageous in that it costs $O(\log L)$ while TD-DFE-LS costs $O(L^2)$.

Looking at the remainder of the NMSE traces in Fig. 5 for $E_b/N_0 \geq 8$ dB, we see that PBiGAMP's performance increases with each step in the sequence G-ind, 2GM-EM, 3GM-EM, 2GM-iid, 2GM-ind, 2GM-nnd. Here we observe two key trends. First, *as the model gets more sophisticated, the NMSE performance improves.* This is seen when going from 2GM-EM to 3GM-EM, and also when going from 2GM-iid to 2GM-ind to 2GM-nnd. In both cases, though, the improvement is less than 1 dB. Second, *sparse models are very effective.* This is seen when comparing G-ind (which has L parameters learned from a large training corpus) to 2GM-EM (which has 3 parameters learned without any training): the simpler 2GM-EM model exploits sparsity and works better as a result.

Fig. 6 shows bit error rate (BER) versus E_b/N_0 for the various schemes under test. There we first see that the BER of TD-DFE-LS is significantly better than that of FD-DFE-LS, although at the cost of $O(L^2)$ versus $O(\log L)$ complexity. Next we see that the BER of G-iid PBiGAMP is noticeably better than that of TD-DFE-LS, even though it costs only $O(\log L)$. Finally, the remainder of the PBiGAMP traces are significantly better than TD-DFE-LS and not far from the PCSI lower bound. Among the different PBiGAMP channel models, the ordering of the BER results matches that of the channel NMSE results, but the spread in BER is rather small.

Fig. 7 shows BER versus turbo iteration at $E_b/N_0 = 12$ dB. There we see that the PBiGAMP schemes converge after only 3-4 iterations. The FD-DFE-LS and TD-DFE-LS schemes take about 20 iterations to converge, suggesting further computational advantages for PBiGAMP.

B. Performance with ZP Transmissions

We now investigate the performance of ZP systems, which use zero-valued guard symbols. Here we modified the FD-DFE-LS and TD-DFE-LS schemes to use LMMSE channel estimation, instead of LS channel estimation, during the first turbo iteration, since this improved their performance.

Fig. 8 shows the channel estimate's NMSE versus E_b/N_0 for the various schemes under test. Similar to the case of UW guards, for ZP guards, we see that the NMSE of TD-DFE-LS is significantly better than that of FD-DFE-LS, while the NMSE of G-iid PBiGAMP is similar to that of TD-DFE-LS. Meanwhile, the NMSEs of the other PBiGAMP variations are significantly better than that of TD-DFE-LS, and the ranking among the PBiGAMP variations matches that observed for UW guards.

Comparing the ZP results in Fig. 8 to the UW results in Fig. 5, we see that the NMSEs of PBiGAMP and TD-DFE-LS are similar for the two guard types, while the NMSE of FD-DFE-LS is worse with ZP, especially at larger values of E_b/N_0. We conjecture a reason for this behavior below.

Fig. 9 shows BER versus E_b/N_0 for the various schemes under test. Similar to the case of UW guards, for ZP guards, we see that the BER of TD-DFE-LS is significantly better than

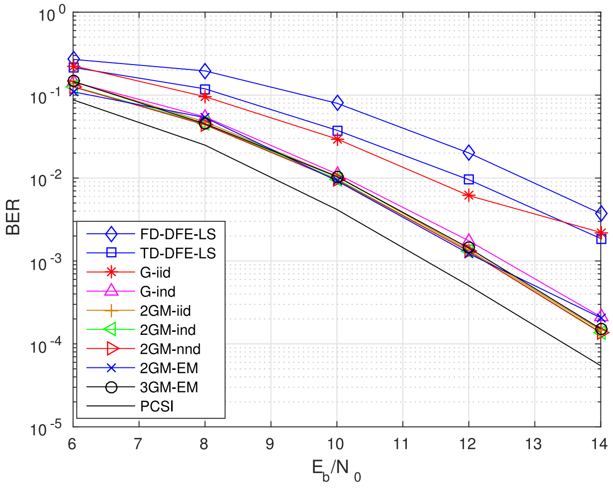

Fig. 9. BER versus E_b/N_0 for ZP transmission and PBiGAMP with different channel models.

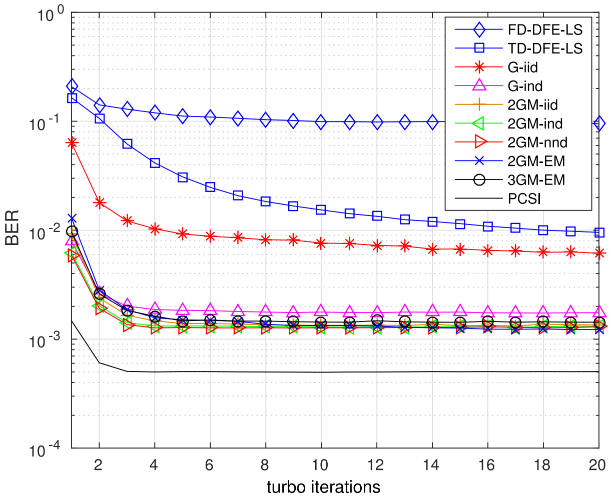

Fig. 10. BER versus turbo iteration for ZP transmission at $E_b/N_0 = 12$ dB.

that of FD-DFE-LS, while the BER of G-iid PBiGAMP is similar to that of TD-DFE-LS. Meanwhile, the BERs of the other PBiGAMP variations are significantly better than that of TD-DFE-LS, and not far from the PCSI lower bound. Among the different PBiGAMP variations, the BER ordering matches that of the NMSE ordering in Fig. 8, but the BER spread is rather small.

Comparing the ZP results in Fig. 9 to the UW results in Fig. 6, we see that the BERs for TD-DFE-LS, PBiGAMP, and PCSI are all better in the ZP case. We attribute this behavior to the fact that the ZP transmissions waste no energy on the guard symbols (which are thrown away) leading to potentially improved symbol recovery. However, the channel is less "visible" in the ZP case, thus requiring a more sophisticated method to recover it. For example, in the UW case, the channel can be easily estimated from any consecutive pair of guard and pilot sequences by forming and inverting an $N_\text{P} \times N_\text{P}$ circulant pilot matrix. No such approach exists in the ZP case. Of course, if those symbols can be accurately estimated, then the interference diminishes, but this never happens in the case of FD-DFE-LS.

Fig. 10 shows BER versus turbo iteration at $E_b/N_0 = 12$ dB. Here again we see that the PBiGAMP schemes converge after only 3-4 iterations. FD-DFE-LS converges after $\approx$10 iterations to a solution of poor quality, while TD-DFE-LS takes at least 20 iterations to converge.

C. Comparison to SparseLift

As described in Section IV-C, SparseLift aims to recover (up to a scalar ambiguity) $\boldsymbol{x}$ and sparse $\boldsymbol{h}$ from noisy measurements $\boldsymbol{y} = \text{Diag}(\boldsymbol{Bh})\boldsymbol{Ax} + \boldsymbol{w}$, and it performs provably well [57] for sufficiently tall $\boldsymbol{A}$ and random $\boldsymbol{B}$.

Equation (9) suggests that a direct application of SparseLift to SC channel-estimation/equalization would use $\boldsymbol{A} = \boldsymbol{F}_M$, $\boldsymbol{B} = \boldsymbol{F}_M^{1:L}$, and $\boldsymbol{x} = [\boldsymbol{x}_\text{P}^\mathsf{T}, \boldsymbol{x}_\text{D}^\mathsf{T}, \boldsymbol{x}_\text{G}^\mathsf{T}]^\mathsf{T}$. Our experiments with this approach were unsuccessful, however, probably because $\boldsymbol{A}$ was not tall. Thus, to make $\boldsymbol{A}$ as tall as possible, we tried ZP transmission with a single scalar-valued pilot $x_\text{P} = 1$. This pilot can be used to resolve the scalar ambiguity on the estimates $\boldsymbol{h}$ and $\boldsymbol{x}$. In this case, $\boldsymbol{A} = \boldsymbol{F}_M^{1:(M-2L+1)}$ and $\boldsymbol{x} = [1, \boldsymbol{x}_\text{D}^\mathsf{T}]^\mathsf{T}$. But here again, SparseLift was unsuccessful, probably due to the lack of randomness in $\boldsymbol{A}$ and $\boldsymbol{B}$.

Next, we tried randomly precoding the data symbols. That is, we constructed a time-domain transmitted block of the form $[\begin{smallmatrix}\boldsymbol{Gx}\\ \boldsymbol{0}_{L-1}\end{smallmatrix}]$, where $\boldsymbol{G} \in \mathbb{R}^{(M-L+1)\times(N_\text{D}+1)}$ was drawn i.i.d. $\{\pm 1/\sqrt{N_\text{D}+1}\}$ and $\boldsymbol{x} = [1, \boldsymbol{x}_\text{D}^\mathsf{T}]^\mathsf{T}$ included a scalar pilot, as before. In this case, SparseLift with $\boldsymbol{A} = \boldsymbol{F}_M^{1:M-L+1}\boldsymbol{G}$ was successful in recovering scaled estimates of $\boldsymbol{x}$ and $\boldsymbol{h}$. The scaling ambiguity was then resolved using the pilot, and the data-symbol estimates were passed to the LDPC coder, via (49), to obtain bit estimates.

Note that this precoding scheme cannot be considered as a "single carrier" scheme, since each transmitted sample is a linear combination of many QAM symbols. Likewise, as a result of the precoding, there is no clear way to exploit FFTs for complexity reduction, thus preventing us from considering the approach as "frequency-domain equalization." Finally, there is no clear way to incorporate prior knowledge of the symbols, so the approach cannot be considered as "SISO." Still, this precoding framework allows us to compare SparseLift—one of a new generation of convex blind-deconvolution algorithms—to more traditional approaches to channel-estimation/equalization. We believe that this is an important comparison since wireless communications is among the principal applications touted for SparseLift [57].

In benchmarking SparseLift, there remains the question of what precoding rate to choose. Recall that, for a fixed spectral efficiency, the precoding rate $(N_\text{D}+1)/(M+L-1)$ could be increased if the LDPC coding rate R was decreased. We thus tested SparseLift over a range of precoding rates, i.e., a range of N_D.

Fig. 11 plots BER versus spectral efficiency (in bpcu) at SNR $= 12$ dB. To change the spectral efficiency, we varied the LDPC code rate R (by varying the number of information bits N_b) while leaving all other modulation parameters (e.g., K, M, N_D, A) fixed. The figure suggests that, for any given spectral

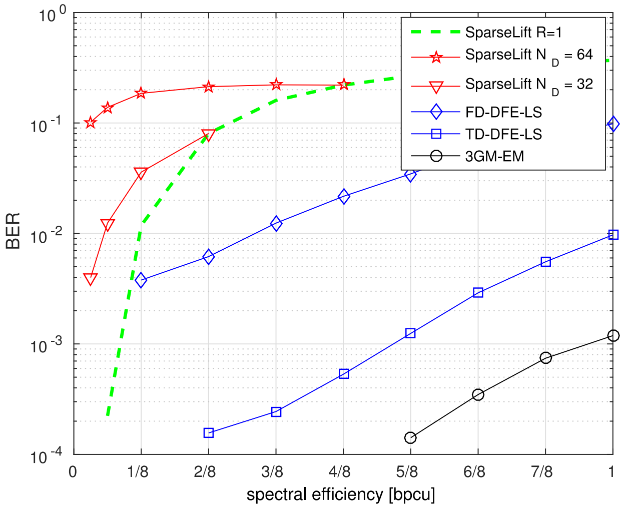

Fig. 11. BER versus spectral efficiency (in bpcu) at SNR $=$ 12 dB with ZP transmission. SparseLift used random precoding $\boldsymbol{G}$ and a single pilot $\boldsymbol{x}_{\text{P}} = 1$, whereas the other approaches used the same Chu pilots as in Figs. 8–10. To construct each solid curve, the LDPC code rate R was varied while the other modulation parameters were fixed. The star-marked red curve used $N_{\text{D}} = 32$ while the triangle-marked red curve used $N_{\text{D}} = 64$. The greed dashed curve was constructed by varying N_{D} at the LDPC coding rate of $R = 1$, i.e., no LDPC coding.

efficiency, SparseLift works best with LDPC code rate $R = 1$, i.e., when no LDPC coding is used. But, even with $R = 1$, SparseLift performs much worse than all of the other schemes under test.

VI. Conclusion

In this paper, we proposed a new SISO equalizer for SC transmissions over unknown frequency-selective block-fading channels. Our equalizer leverages the recently proposed PBiGAMP algorithm for joint channel-estimation and symbol-detection and exploits FFTs to achieve a per-symbol complexity that grows only logarithmically in the channel delay-spread L. A D-state GMM channel prior was used to exploit the approximately sparse nature of realistic wireless channels, with statistics learned via the EM-GM-AMP approach [51]. Our methodolody also supports the more sophisticated clustered-sparse GMM/HMM prior from [49], although the details are omitted due to space limitations.

Numerical experiments conducted using realistic Saleh-Valenzuela channel realizations and SRRC pulse-shaping suggest that the proposed approach achieves channel NMSE and BER that is significantly improved over existing turbo FDE approaches like [37] and [38]. Additional experiments showed that the proposed scheme facilitates much higher spectral efficiencies than SparseLift, a recently proposed method for sparse deconvolution based on convex relaxation.

The method we proposed could be straightforwardly extended in several directions. One extension is to multi-antenna systems. Another is to systems with fast-fading channels, where basis expansion could be used to model time variation. Yet another is to DFT-precoded systems, like those in [56].

Appendix A
Saleh-Valenzuela Channel Parameters

Based on the 3–10 GHz experiments in [39], we assumed the following regarding the parameters of the Saleh-Valenzuela model (10).

- The cluster arrival times are a Poisson process with rate Λ, i.e., $p(T_c|T_{c-1}) = \Lambda \exp(-\Lambda(T_c - T_{c-1}))$. The initial cluster delay $T_1 \geqslant \tau_{\min}$, as seen by the receiver, is a function of the timing synchronization algorithm.
- The component arrivals are a mixture of two Poisson processes: $p(\tau_{v,c}|\tau_{v-1,c}) = \beta\lambda_1 \exp(-\lambda_1(\tau_{v,c} - \tau_{v-1,c})) + (1-\beta)\lambda_2 \exp(-\lambda_2(\tau_{v,c} - \tau_{v-1,c}))$ with $\tau_{0,c} = 0$.
- The component energies obey
$$\mathbb{E}\{|g_{v,c}|^2\} = \frac{\exp\left(-T_c/\Gamma - \tau_{(v,c)}/\gamma\right)}{\gamma\left[(1-\beta)\lambda_1 + \beta\lambda_2 + 1\right]}, \qquad (51)$$
where Γ is the cluster decay time constant and γ is the intra-cluster decay time constant.
- The amplitudes $\{g_{v,c}\}$ are independent and identically distributed (i.i.d.) Nakagami with m-factors randomly generated via i.i.d. $m \sim \mathcal{N}(m_0, \widehat{m}_0^2)$.
- The phases $\{\phi_{v,c}\}$ are i.i.d. uniform on $[0, 2\pi)$.
- The number of clusters C is Poisson distributed with mean $\bar{C}$, i.e., $p(C) = (\bar{C})^C \exp(-\bar{C})/C!$.
- The number of components per cluster V is set large enough to yield a desired modeling accuracy.

Beyond the above specifications, we used:

- $V = 100$ components per cluster,
- pulses $g_{\text{t}}(\tau)$ and $g_{\text{r}}(\tau)$ that are square-root raised cosine (SRRC) designs with parameter 0.5,
- a system bandwidth of $T^{-1} = 64$ MHz,
- $L = 64$ channel taps,
- an initial delay of $T_1 = L_{\text{pre}}T + \widetilde{T}_0$, where $L_{\text{pre}} = 4$ and $\widetilde{T}_0$ is exponentially distributed with mean T, i.e., $p(\widetilde{T}_0) = \Gamma_0 \exp(-\Gamma_0 \widetilde{T}_0)$ with $\Gamma_0 = 1/T$. Here, we chose L_{pre} so that $\{g_l\}_{l=0}^{L_{\text{pre}}}$ captures the "pre-cursor" energy contributed by the pulse shape, while Γ_0 models a positive synchronization uncertainty. The "cursor" tap index is thus $l = L_{\text{pre}} + 1$.

References

[1] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Commun. Mag.*, vol. 40, no. 4, pp. 58–66, Apr. 2002.

[2] F. Pancaldi, G. M. Vitetta, R. Kalbasi, N. Al-Dhahir, M. Uysal, and H. Mheidat, "Single-carrier frequency domain equalization," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 37–56, Sep. 2008.

[3] L. J. Cimini, Jr., "Analysis and simulation of a digital mobile radio channel using orthogonal frequency division multiplexing," *IEEE Trans. Commun.*, vol. 33, no. 7, pp. 665–765, Jul. 1985.

[4] T. Pollet, M. V. Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 191–193, Feb./Apr. 1995.

[5] G. Carron, R. Ness, L. Deneire, L. V. Perre, and M. Engels, "Comparison of two modulation techniques using frequency domain processing for in-house networks," *IEEE Trans. Consum. Electron.*, vol. 47, no. 1, pp. 63–72, Feb. 2001.

[6] L. Tong, B. M. Sadler, and M. Dong, "Pilot-assisted wireless transmissions," *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 12–25, Nov. 2004.

[7] H. Vikalo, B. Hassibi, B. Hochwald, and T. Kailath, “On the capacity of frequency-selective channels in training-based transmission schemes,” *IEEE Trans. Signal Process.*, vol. 52, no. 9, pp. 2572–2583, Sep. 2004.

[8] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, “Iterative correction of intersymbol interference: Turbo equalization,” *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sep./Oct. 1995.

[9] R. Koetter, A. C. Singer, and M. Tüchler, “Turbo equalization,” *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 67–80, Jan. 2004.

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

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

[12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.

[13] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” in *Proc. IEEE Global Telecommun. Conf.*, Dallas, TX, USA, Nov. 1989, pp. 1680–1686.

[14] P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” in *Proc. IEEE Int. Conf. Commun.*, 1995, pp. 1009–1013.

[15] A. O. Berthet, B. S. Ünal, and R. Visoz, “Iterative decoding of convolutionally encoded signals over multipath Rayleigh fading channels,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1729–1743, Sep. 2001.

[16] N. Nefedov, M. Pukkila, R. Visoz, and A. O. Berthet, “Iterative data detection and channel estimation for advanced TDMA systems,” *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 141–144, Apr. 2003.

[17] A. Glavieux, J. Laot, and J. Labat, “Turbo-equalization over a frequency-selective channel,” in *Proc. Int. Symp. Turbo Codes*, 1997, pp. 96–102.

[18] S. Ariyavisitakul and Y. Li, “Joint coding and decision feedback equalization for broadband wireless channels,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 9, pp. 1670–1678, Dec. 1998.

[19] Z. Wu and J. Cioffi, “Turbo decision aided equalization for magnetic recording channels,” in *Proc. IEEE Global Telecommun. Conf.*, 1999, pp. 733–738.

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

[21] M. Tüchler, R. Koetter, and A. C. Singer, “Turbo equalization: Principles and new results,” *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.

[22] R. R. Lopes and J. R. Barry, “The soft-feedback equalizer for turbo equalization of highly dispersive channels,” *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 783–788, May 2006.

[23] A. Anastasopoulos, K. M. Chugg, G. Colavolpe, G. Ferrari, and R. Raheli, “Iterative detection for channels with memory,” *Proc. IEEE*, vol. 95, no. 6, pp. 1272–1294, Jun. 2007.

[24] B. D. Hart and S. Pasupathy, “Innovations-based MAP detection for time-varying frequency-selective channels,” *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1507–1519, Sep. 2000.

[25] A. Anastasopoulos and K. M. Chugg, “Adaptive soft-input soft-output algorithms for iterative detection with parametric uncertainty,” *IEEE Trans. Commun.*, vol. 48, no. 10, pp. 1638–1649, Oct. 2000.

[26] L. M. Davis, I. B. Collings, and P. Hoeher, “Joint MAP equalization and channel estimation for frequency-selective and frequency-flat fast-fading channels,” *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2106–2114, Dec. 2001.

[27] G. K. Kaleh and R. Vallet, “Joint parameter estimation and symbol detection for linear or nonlinear unknown channels,” *IEEE Trans. Commun.*, vol. 42, no. 7, pp. 2406–2413, Jul. 1994.

[28] R. R. Lopes and J. R. Barry, “The extended-window channel estimator for iterative channel-and-symbol estimation,” *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 2, pp. 92–95, 2005.

[29] M. Sandell, C. Luschi, P. Strauch, and R. Yan, “Iterative channel estimation using soft decision feedback,” in *Proc. IEEE Global Telecommun. Conf.*, Nov. 1998, pp. 3728–3733.

[30] M. Nicoli, S. Ferrara, and U. Spagnolini, “Soft-iterative channel estimation: Methods and performance analysis,” *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2993–3006, Jun. 2007.

[31] E. Baccarelli and R. Cusani, “Combined channel estimation and data detection using soft statistics for frequency-selective fast-fading digital links,” *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 424–427, Apr. 1998.

[32] S. Song, A. C. Singer, and K.-M. Sung, “Soft input channel estimation for turbo equalization,” *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2885–2894, Oct. 2004.

[33] R. Otnes and M. Tüchler, “Iterative channel estimation for turbo equalization of time-varying frequency-selective channels,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1918–1923, Nov. 2004.

[34] C. Laot, A. Glavieux, and J. Labat, “Turbo equalization: Adaptive equalization and channel decoding jointly optimized,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1744–1752, Sep. 2001.

[35] M. Tüchler and J. Hagenauer, “Linear time and frequency domain turbo equalization,” in *Proc. IEEE Veh. Technol. Conf.*, 2001, pp. 1449–1453.

[36] F. Pancaldi and G. M. Vitetta, “Block channel equalization in the frequency domain,” *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 463–471, Mar. 2005.

[37] B. Ng, C. Lam, and D. Falconer, “Turbo frequency domain equalization for single-carrier broadband wireless systems,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 759–767, Feb. 2007.

[38] F. Coelho, R. Dinis, and P. Montezuma, “Joint detection and channel estimation for block transmission schemes,” in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2010, pp. 1765–1770.

[39] A. F. Molisch, “Ultrawideband propagation channels—Theory, measurement, and modeling,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1528–1545, Sep. 2005.

[40] A. P. Worthen and W. E. Stark, “Unified design of iterative receivers using factor graphs,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 843–849, Feb. 2001.

[41] A. Montanari, “Graphical models concepts in compressed sensing,” in *Compressed Sensing: Theory and Applications*, Y. C. Eldar and G. Kutyniok, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[42] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[43] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing,” *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[44] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing: I. Motivation and construction,” in *Proc. IEEE Inf. Theory Workshop Inf. Theory*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[45] 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, Jul. 2002.

[46] T. Tanaka and M. Okada, “Approximate belief propagation, density evolution, and neurodynamics for CDMA multiuser detection,” *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 700–706, Feb. 2005.

[47] A. Montanari and D. Tse, “Analysis of belief propagation for non-linear problems: The example of CDMA (or: How to prove Tanaka’s formula),” in *Proc. IEEE Inf. Theory Workshop Inf. Theory*, 2006, pp. 160–164.

[48] P. Schniter, “Turbo reconstruction of structured sparse signals,” in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2010, pp. 1–6.

[49] P. Schniter, “A message-passing receiver for BICM-OFDM over unknown clustered-sparse channels,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 8, pp. 1462–1474, Dec. 2011.

[50] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” in *Proc. IEEE Int. Symp. Inf. Theory.*, Aug. 2011, pp. 2168–2172.

[51] J. P. Vila and P. Schniter, “Expectation-maximization Gaussian-mixture approximate message passing,” *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[52] J. T. Parker and P. Schniter, “Parametric bilinear generalized approximate message passing,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 1–14, Jun. 2016.

[53] C. Schülke, P. Schniter, and L. Zdeborová, “Phase diagram of matrix compressed sensing,” *Phys. Rev. E*, vol. 94, no. 6, Dec. 2016, Art no. 062 136.

[54] A. Saleh and R. A. Valenzuela, “A statistical model for indoor multipath propagation,” *IEEE J. Sel. Areas Commun.*, vol. 5, no. 2, pp. 128–137, Feb. 1987.

[55] P. Schniter, S.-J. Hwang, S. Das, and A. P. Kannu, “Equalization of time-varying channels,” in *Wireless Communications over Rapidly Time-Varying Channels*, F. Hlawatsch and G. Matz, Eds. New York, NY, USA: Academic, 2011, pp. 321–352.

[56] N. Benvenuto, R. Dinis, D. Falconer, and S. Tomasin, “Single carrier modulation with nonlinear frequency domain equalization: An idea whose time has come—again,” *Proc. IEEE*, vol. 98, no. 1, pp. 69–96, Jan. 2010.

[57] S. Ling and T. Strohmer, “Self-calibration and biconvex compressive sensing,” *Inverse Problems*, vol. 31, no. 11, 2015, Art. no. 115002.

[58] L. Deneire, B. Gyselinchx, and M. Engels, "Training sequence versus cyclic prefix—A new look on single carrier communication," *IEEE Commun. Lett.*, vol. 5, no. 7, pp. 292–294, Jul. 2001.

[59] G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artif. Intell.*, vol. 42, pp. 393–405, 1990.

[60] 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.

[61] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2003.

[62] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int. J. Comput Vis.*, vol. 40, no. 1, pp. 25–47, Oct. 2000.

[63] T. Minka, "A family of approximate algorithms for Bayesian inference," Ph.D. dissertation, Dept. Comp. Sci. Eng., MIT, Cambridge, MA, USA, Jan. 2001.

[64] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. New York, NY, USA: Cambridge Univ. Press, 2009.

[65] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2007.

[66] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

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

[68] D. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 531–532, Jul. 1972.

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

Peng Sun received the B.S. degree in communication engineering from Zhengzhou University, Zhengzhou, China, in 2012. He is currently working toward the Ph.D. degree in the School of Information and Engineering, Zhengzhou University, Zhengzhou, China. In 2015–2017, he was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA. His research interests include signal processing and communication theory.

Zhongyong Wang received the B.S. and M.S. degrees in automatic control from Harbin Shipbuilding Engineering Institute, Harbin, China, in 1986 and 1988, respectively, and the Ph.D. degree in automatic control theory and application from Xi'an Jiaotong University, Xi'an, China, in 1998. Since 1988, He has been with the Department of Electronics, Zhengzhou University, Zhengzhou, China, as a Lecturer. From 1999 to 2002, he was an Associate Professor, and in 2002, he was promoted to Professor with the Department of Communication Engineering. His research interests include numerous aspects within embedded systems, signal processing, and communication theory.

Philip Schniter (S'92–M'93–SM'05–F'14) received the B.S. and M.S. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 1992 and 1993, respectively, and the Ph.D. degree in electrical engineering from Cornell University, Ithaca, NY, USA, in 2000. From 1993 to 1996, he was with the Tektronix Inc., Beaverton, OR, USA, as a Systems Engineer. After receiving the Ph.D. degree, he joined the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA, where he is currently a Professor. In 2008–2009, he was a Visiting Professor with Eurecom, Sophia Antipolis, France, and with Supelec, Gif-sur-Yvette, France. In 2016–2017, he was a Visiting Professor with Duke University, Durham, NC, USA. His research interests include signal processing, wireless communications, and machine learning. He was the recipient of the NSF CAREER award in 2002, the IEEE Signal Processing Society Best Paper Award in 2016, and the Qualcomm Faculty Award in 2017. He is currently on the IEEE Sensor Array and Multichannel Technical Committee and the IEEE Computational Imaging Special Interest Group.