Markov Chain Monte Carlo Detectors for Channels With Intersymbol Interference

Rong-Hui Peng, Rong-Rong Chen, *Member, IEEE*, and Behrouz Farhang-Boroujeny, *Senior Member, IEEE*

***Abstract*—In this paper, we propose novel low-complexity soft-in soft-out (SISO) equalizers using the Markov chain Monte Carlo (MCMC) technique. We develop a bitwise MCMC equalizer (b-MCMC) that adopts a Gibbs sampler to update one bit at a time, as well as a group-wise MCMC (g-MCMC) equalizer where multiple symbols are updated simultaneously. The g-MCMC equalizer is shown to outperform both the b-MCMC and the linear minimum mean square error (MMSE) equalizer significantly for channels with severe amplitude distortion. Direct application of MCMC to channel equalization requires sequential processing which leads to long processing delay. We develop a parallel processing algorithm that reduces the processing delay by orders of magnitude. Numerical results show that both the sequential and parallel processing MCMC equalizers perform similarly well and achieve a performance that is only slightly worse than the optimum maximum *a posteriori* (MAP) equalizer. The MAP equalizer, on the other hand, has a complexity that grows exponentially with the size of the memory of the channel, while the complexity of the proposed MCMC equalizers grows linearly.**

***Index Terms*—Equalization, intersymbol interference, Markov chain Monte Carlo, soft-in soft-out detection.**

I. Introduction

THE increasing demand for high speed wireless products has motivated a significant amount of research to combat the intersymbol interference (ISI) resulting from multipath transmission. Early developments date back to the 1960s and 1970s when symbol-by-symbol linear and decision feedback equalizers were developed [1], [2]. These traditional methods face the problem of noise enhancement, in the case of linear equalizers, or error propagation, in the case of decision feedback equalizers. Furthermore, these methods are based on hard decisions and thus cannot benefit from the modern coding techniques where by making use of soft information one can approach the channel capacity. Significant improvement in system performance can be achieved by joint processing of equalization and channel decoding. Such systems operate based on turbo principles where soft information are exchanged between a soft-in soft-out (SISO) equalizer and a channel decoder. The optimal maximum *a posteriori* (MAP) equalizer can be used to find the best bit/symbol stream that matches the received signal in the presence of prior information provided by the channel decoder. However, the computational complexity of the MAP equalizer, even with the use of efficient implementations such as the BCJR algorithm [3] is exponential with respect to the length of the channel impulse response and the constellation size and thus may be prohibitive in many cases.

To resolve this problem, low complexity equalizers have been developed and remain an active area of research. The first work to reduce the complexity of the MAP equalizer for a turbo equalization system is due to [4], where the soft-output Viterbi algorithm (SOVA) is used for the implementation of the SISO equalizer. In [5], a low-complexity SISO equalizer is implemented with an adaptive soft interference canceler based on linear filters. A SISO turbo equalizer based on the minimum mean square error (MMSE) criteria is proposed [6] and [7]. This turbo MMSE equalizer performs better than the SIC approach of [5] and is widely used due to its excellent performance. Trellis-based approaches that prune the insignificant branches of trellis to reduce the complexity of the BCJR algorithm have also been successfully developed. Examples of such algorithms are breadth-first algorithms such as the M-best BCJR (M-BCJR) and the threshold-based BCJR (T-BCJR) of [8], and the depth-first algorithm such as the list-sequential (LISS) algorithm of [9] and [10].

In this paper, we develop a novel low complexity approximation to the MAP equalizer based on the Markov chain Monte Carlo (MCMC) simulation principles [11]. The MCMC simulation is a mathematical tool that may be used to draw samples from an arbitrary and possibly unknown distribution. The key point that makes the MCMC attractive to SISO equalization is the fact that, unlike the BCJR algorithm, its computational complexity does not grow exponentially with the channel memory. The BCJR algorithm uses a complete sample set of trellis states to achieve optimal detection, while the other trellis-based approaches [8] trade the use of an incomplete set of trellis states for suboptimal performance. As noted in [8], to ensure good detection performance, the trellis states chosen should be those that have significant contribution to the soft information that one seeks. The SISO MCMC equalizer proposed in this paper follows a similar strategy. It uses an statistical method to search for a *small* (to keep the complexity low) but *important* (to achieve good performance) sample set containing important samples that match the best with received signals.

The contributions of this work are summarized as follows:

1) We develop new MCMC detectors for ISI channels that can achieve near optimal MAP performance at low complexity. Such detectors operate either bit-wise (b-MCMC), or group-wise (g-MCMC) over groups of G symbols. Previous work in the literature considers only b-MCMC [12], or symbol-wise MCMC (s-MCMC) [13], and the latter is

Manuscript received June 30, 2009; accepted November 14, 2009. First published December 18, 2009; current version published March 10, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Milica Stojanovic. This work is supported in part by the NSF under Grants ECS-0547433 and ECS-0524720. The material in this paper was presented in part at the IEEE International Conference on Communications (ICC), Dresden, Germany, June 14–19, 2009.

The authors are with the Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT 84112 USA (e-mail: peng@ece.utah.edu; rchen@ece.utah.edu; farhang@ece.utah.edu).

Digital Object Identifier 10.1109/TSP.2009.2038958

1053-587X/$26.00 © 2010 IEEE

a special case of the proposed g-MCMC with $G = 1$. For ISI channels with moderate or severe amplitude distortion, the b-MCMC or s-MCMC is shown to be much inferior to the g-MCMC with $G = 2$ or $G = 4$.

2) We provide a systematic comparison between the proposed MCMC detectors and the SISO turbo MMSE equalizer [7] over several ISI channels with different levels of amplitude distortion. It is shown that for channels with mild distortion, all the detectors considered in this work perform closely to the optimal MAP detector. However, as the level of amplitude distortion increases, the g-MCMC $(G \geq 2)$ performs about 1.5–2 dB better than the MMSE detector.
3) We propose to use the QRD-M algorithm [14] in the implementation of g-MCMC to moderate its computational complexity.
4) We develop a parallel implementation of the MCMC detector for ISI channels which significantly reduces the processing delay of its sequential counterpart without degradation of performance.

The earlier works on the application of MCMC detector to MIMO detection were based on s-MCMC [15]–[17]. In [18], where an implementation of the MCMC detector in a custom chip was studied, it was noted that b-MCMC is more appealing to hardware implementation. More recently, a comparison of b-MCMC and s-MCMC for achieving capacity of MIMO channels has been presented in [19]. It reveals that for MIMO channels b-MCMC performs as good as s-MCMC, thus the conclusion was drawn that b-MCMC should be used because of its simplicity. Our study in this paper reveals that this conclusion is not applicable to channels with ISI. Simulation studies show that in channels with a small *condition number* (where the channel condition number is defined as the ratio of the maximum over minimum spectral density at the channel output) the b-MCMC performs as good as the s-MCMC and g-MCMC and all perform very close to the optimal MAP detector. However, as the channel condition number increases, the b-MCMC begins to fail, and in channels with higher condition number, the s-MCMC may also fail. In channels with a large condition number, only g-MCMC performs close to the MAP detector. It is also worth noting that the condition number defined here is a simple way of quantifying the channel amplitude distortion by a single number. A larger condition number means a higher amplitude distortion.

In a recent work, Kashif *et al.* [13] have also explored the use of MCMC for SISO equalization. The emphasis of [13] is on non-linear channels with very short memory—the channels considered have a memory of two samples. It also explores a number of MCMC detection algorithms from the literature, including the one presented in [16]. In this paper, we explore linear channels with longer memory length. Also, based on the results in [13] and our findings, we note that the least complex and most stable MCMC algorithm reported so far is the one proposed in [16]. We thus concentrate on the use of this algorithm.

While in this work we consider bit/symbol-wise MAP equalization, another important class of equalization techniques has been investigated extensively in the literature which performs maximum likelihood sequence estimation (MLSE). The basis of such techniques is the Viterbi algorithm (VA) which is used to search for the most likely transmitted sequence (path) with the minimum cost in a trellis. The complexity of the VA algorithm is determined by the number of states in the trellis, which grows exponentially with the length of the channel impulse response and the size of the signal constellation. To reduce algorithm complexity, the generalized Viterbi algorithm (GLA) developed in [20] operates on a trellis with a smaller constraint length and invokes list decoding to select a list of $S > 1$ survivors connecting to each trellis state. The reduced-state sequence estimation (RSSE) proposed in [21] constructs a trellis with fewer states based on partitioning a set of channel states. In [22], per-survivor processing (PSP) techniques are developed to approximate MLSE based on the idea of canceling residual ISI on the basis of the particular survivor sequence, i.e., in a per-survivor fashion. The PSP techniques are applicable to reduced-complexity MLSE algorithms [20], [21] and to channels with unknown parameters. Breadth-first MLSE under given structural and complexity constraints are discussed in [23]. In [24], a statistical approach based on particle filter is investigated for ISI channels. It proposes a grouping technique of particle trajectories to reduce algorithm complexity which resembles merging paths in trellis decoding. A large number of particle trajectories, however, is still required to approach the performance of the MAP detector. Detailed comparisons of the above approaches with the MCMC equalizers proposed in this work, however, are beyond the scope of this paper.

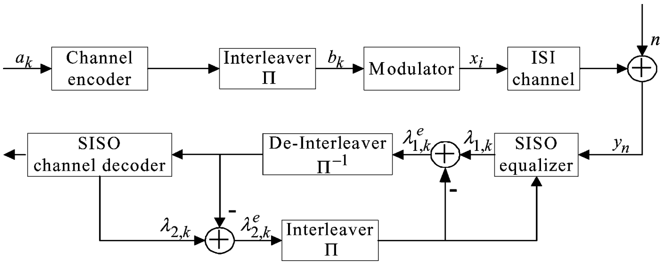


Fig. 1. A block diagram of the system model that employs joint iterative equalization and channel decoding.

This paper is organized as follows. In Section II, we introduce the system model and the optimum MAP equalization. Section III includes detailed descriptions of the proposed MCMC equalizers. Simulation results are presented in Section IV and the conclusions are drawn in Section V.

In this paper, the following notations are used. Vectors are denoted by lowercase bold letters and are in column form. Matrices are denoted by uppercase bold letters. The superscripts T is used to denote matrix or vector transpose. To differentiate between bit and symbol indices, k is consistently used for bits and i is used for the symbols. The notations $P(\cdot)$ and $p(\cdot)$ denote the probability mass function of discrete random variables and the probability density function of continuous random variables, respectively.

II. System Description

We consider the communication system depicted in Fig. 1. A sequence of binary information bits $\{a_k\}$ is first encoded by a channel encoder of rate R. The coded bits are passed through an interleaver with the output $\{b_k\}$. We assume a packetized data transmission system in which each packet consists of M_bN coded bits. The modulator maps each set of M_b bits of $\{b_k\}$ to a data symbol x_i to form the transmit signal sequence $\{x_n\}$. This sequence is passed through an ISI channel to generate output

$\{y_n\}$. At the receiver, SISO equalization and channel decoding are performed in a turbo loop. The SISO equalizer computes the log-likelihood ratio (LLR) of the coded bits, $\{\lambda_{1,k}\}$, based on the extrinsic information from the channel decoder, $\{\lambda^{\mathrm{e}}_{2,k}\}$. The extrinsic (i.e., the new portion) of LLR values from the SISO equalizer given by $\lambda^{\mathrm{e}}_{1,k} = \lambda_{1,k} - \lambda^{\mathrm{e}}_{2,k}$ are passed to the channel decoder for channel decoding. The decoder generates improved extrinsic LLR values $\{\lambda^{\mathrm{e}}_{2,k}\}$ to be passed back to the equalizer. After several outer iterations of information exchange between the SISO equalizer and channel decoder, the channel decoder makes a decision on the transmitted uncoded bits $\{a_k\}$ based on $\{\lambda_{2,k}\}$.

A. ISI Channel Model

The ISI channel is characterized by the equation

$$y_n = \sum_{l=0}^{L} h_l x_{n-l} + z_n, \quad \text{for } n = 0, 1, \cdots, N-1+L \tag{1}$$

where n is the time index, $\{y_n\}_{n=0}^{N-1+L}$ is the received signals, $\{h_l\}_{l=0}^{L}$ is the channel impulse response, $\{x_n\}_{n=0}^{N-1}$ is the transmitted symbol sequence, $\{z_n\}_{n=0}^{N-1+L}$ are independent and identically distributed (i.i.d.) complex Gaussian noise samples with zero-mean and a variance of $2\sigma^2$. While we transmit N symbols $\{x_n\}_{n=0}^{N-1}$, due to the channel memory, we take into account the received signal samples $\{y_n\}_{n=N}^{N-1+L}$ because they also depend on the transmitted sequence. For notational convenience, in (1), we let $x_{-L} = \cdots = x_{-1} = 0$ and $x_N = \cdots = x_{N-1+L} = 0$. We define vectors $\mathbf{y} = [y_0, y_1, \cdots, y_{N-1+L}]^T$ and $\mathbf{x} = [x_0, x_1, \cdots, x_{N-1}]^T$. Also, let $\mathbf{x}_{j-L:j} = [x_{j-L}, x_{j-L+1}, \cdots, x_j]^T$. Due to finite memory of the channel, we have

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \prod_{j=0}^{N-1+L} p(y_j|\mathbf{x}_{j-L:j}) \\ &= \prod_{j=0}^{N-1+L} \exp\left(-\frac{1}{2\sigma^2}\left|y_j - \sum_{l=0}^{L} h_l x_{j-l}\right|^2\right) \\ &= \exp\left\{\sum_{j=0}^{N-1+L}\left(-\frac{1}{2\sigma^2}\left|y_j - \sum_{l=0}^{L} h_l x_{j-l}\right|^2\right)\right\}. \end{aligned} \tag{2}$$

For each bit vector $\mathbf{b} = \{b_0, b_1, \cdots, b_{M_bN-1}\}$, we denote the corresponding symbol vector by $\mathbf{x}(\mathbf{b}) = \{x_0, \cdots, x_{N-1}\}$. When no confusion arises, we simplify the notation and use $\mathbf{x}$ instead of $\mathbf{x}(\mathbf{b})$. Since there is a one-to-one mapping between $\mathbf{b}$ and $\mathbf{x}$, we let $p(\mathbf{y}|\mathbf{b}) = p(\mathbf{y}|\mathbf{x})$ and use them interchangeably.

B. Optimal MAP Equalizer

Given $\mathbf{y}$, the extrinsic LLR value of a particular bit b_k is given by

$$\begin{aligned} \lambda^e_{1,k} &= \ln\frac{P(b_k = 0|\mathbf{y}, \boldsymbol{\lambda}^e_2)}{P(b_k = 1|\mathbf{y}, \boldsymbol{\lambda}^e_2)} - \lambda^e_{2,k} \\ &= \ln\frac{\sum\limits_{\mathbf{b}:b_k=0} P(\mathbf{b}|\mathbf{y}, \boldsymbol{\lambda}^e_2)}{\sum\limits_{\mathbf{b}:b_k=1} P(\mathbf{b}|\mathbf{y}, \boldsymbol{\lambda}^e_2)} - \lambda^e_{2,k} \\ &= \ln\frac{\sum\limits_{\mathbf{b}:b_k=0} \frac{p(\mathbf{y}|\mathbf{b},\boldsymbol{\lambda}^e_2)P(\mathbf{b}|\boldsymbol{\lambda}^e_2)}{p(\mathbf{y}|\boldsymbol{\lambda}^e_2)}}{\sum\limits_{\mathbf{b}:b_k=1} \frac{p(\mathbf{y}|\mathbf{b},\boldsymbol{\lambda}^e_2)P(\mathbf{b}|\boldsymbol{\lambda}^e_2)}{p(\mathbf{y}|\boldsymbol{\lambda}^e_2)}} - \lambda^e_{2,k} \\ &= \ln\frac{\sum\limits_{\mathbf{b}:b_k=0} p(\mathbf{y}|\mathbf{b})P(\mathbf{b}|\boldsymbol{\lambda}^e_2)}{\sum\limits_{\mathbf{b}:b_k=1} p(\mathbf{y}|\mathbf{b})P(\mathbf{b}|\boldsymbol{\lambda}^e_2)} - \lambda^e_{2,k}. \end{aligned} \tag{3}$$

Since

$$\lambda^e_{2,j} = \ln\frac{P(b_j = 0)}{P(b_j = 1)},$$

we have $P(b_j = 0|\lambda^e_{2,j}) = f_j \cdot \exp(\lambda^e_{2,j}/2)$ and $P(b_j = 1|\lambda^e_{2,j}) = f_j \cdot \exp(-\lambda^e_{2,j}/2)$, where $f_j = [\exp(\lambda^e_{2,j}/2) + \exp(-\lambda^e_{2,j}/2)]^{-1}$. Hence, we can write

$$P(b_j|\lambda^e_{2,j}) = f_j \cdot \exp\left((-1)^{b_j}\lambda^e_{2,j}/2\right), \quad \text{for } b_j = 0, 1.$$

Furthermore, we make the independence assumption such that

$$\begin{aligned} P(\mathbf{b}|\boldsymbol{\lambda}^e_2) &= \prod_{j=0}^{M_bN-1} P(b_j|\lambda^e_{2,j}) \\ &= \left[\prod_{j=0}^{M_bN-1} \exp\left((-1)^{b_j}\lambda^e_{2,j}/2\right)\right]\left[\prod_{j=0}^{M_bN-1} f_j\right]. \end{aligned} \tag{4}$$

We then substitute (4) into (3) to obtain (5), shown at the bottom of the page.

$$\begin{aligned} \lambda^e_{1,k} &= \ln\frac{\sum\limits_{\mathbf{b}:b_k=0}\left\{p(\mathbf{y}|\mathbf{b})\left[\prod_{j=0}^{M_bN-1}\exp\left((-1)^{b_j}\lambda^e_{2,j}/2\right)\right]\left[\prod_{j=0}^{M_bN-1} f_j\right]\right\}}{\sum\limits_{\mathbf{b}:b_k=1}\left\{p(\mathbf{y}|\mathbf{b})\left[\prod_{j=0}^{M_bN-1}\exp\left((-1)^{b_j}\lambda^e_{2,j}/2\right)\right]\left[\prod_{j=0}^{M_bN-1} f_j\right]\right\}} - \lambda^e_{2,k} \\ &= \ln\frac{\sum\limits_{\mathbf{b}:b_k=0}\left\{p(\mathbf{y}|\mathbf{b})\left[\prod_{j=0}^{M_bN-1}\exp\left((-1)^{b_j}\lambda^e_{2,j}/2\right)\right]\right\}}{\sum\limits_{\mathbf{b}:b_k=1}\left\{p(\mathbf{y}|\mathbf{b})\left[\prod_{j=0}^{M_bN-1}\exp\left((-1)^{b_j}\lambda^e_{2,j}/2\right)\right]\right\}} - \lambda^e_{2,k} \\ &= \ln\frac{\sum\limits_{\mathbf{b}:b_k=0}\exp\left[-\frac{1}{2\sigma^2}\sum_{j=0}^{N-1+L}\left|y_j - \sum\limits_{l=0}^{L} h_l x_{j-l}\right|^2 + \sum_{j=0}^{M_bN-1}(-1)^{b_j}\lambda^e_{2,j}/2\right]}{\sum\limits_{\mathbf{b}:b_k=1}\exp\left[-\frac{1}{2\sigma^2}\sum_{j=0}^{N-1+L}\left|y_j - \sum\limits_{l=0}^{L} h_l x_{j-l}\right|^2 + \sum_{j=0}^{M_bN-1}(-1)^{b_j}\lambda^e_{2,j}/2\right]} - \lambda^e_{2,k}. \end{aligned} \tag{5}$$

The summations in (5) are over a total of 2^{NM_b-1} combinations of $\mathbf{b}$. For typical values of M_bN which are in the order of at least few hundreds, this clearly is a prohibitive complexity. By utilizing the trellis structure of the ISI channel, the BCJR algorithm [3] may be used to reduce this complexity significantly. However, even this reduced complexity grows exponentially with the multiplication of the length of the channel $L+1$ and the number of bits per symbol M_b. We use MCMC to reduce this complexity to a more affordable level.

III. MCMC EQUALIZER

The complexity of the optimum SISO (i.e., MAP) equalizer comes from the exponential growth of the number of combinations of $\mathbf{b}$ which leads to 2^{NM_b-1} summation terms in the numerator and in the denominator of (5). However, we note that in practice there is always only a small subset of $\mathbf{b}$ that contributes significantly to the final results of the summations. Let us call such subset the *important set*. An ideal important set is the set that includes only the significant probability terms under the summations in (5). The MCMC is a search method that finds the desired significant terms by browsing through the choices of $\mathbf{b}$ in an efficient manner. We refer to these choices as *samples*. By increasing the number of samples of MCMC, one can improve the detector performance. In other word, MCMC allows one to trade between the complexity and performance.

A. Bit-Wise MCMC Equalizer

The MCMC equalizer, in general, is implemented in two steps. In the first step, the Gibbs sampler is used to generate a collection of sample vectors $\mathbf{b}^{(n)}$, for $n = 0, 1, \cdots, I$, which forms the important sample set $\mathcal{I}$. In the second step, the *a posteriori* LLR values are computed using the samples in $\mathcal{I}$. The details, for b-MCMC, are as follows.

1) Gibbs Sampler: During the nth iteration, starting with $\mathbf{b}^{(n-1)} = \{b_0^{(n-1)}, \cdots, b_{M_bN-1}^{(n-1)}\}$, $\mathbf{b}^{(n)}$ is generated bit by bit sequentially. Assuming that b_0 through b_{k-1} have been updated during the nth iteration, given by $b_0^{(n)}, \cdots b_{k-1}^{(n)}$, and bits $\{b_{k+1}, \cdots, b_{M_bN-1}\}$ have been last updated during the $(n-1)$th iteration, denoted by $b_{k+1}^{(n-1)}, \cdots b_{M_bN-1}^{(n-1)}$. Conditioned upon

$$\bar{\mathbf{b}}_k = \left\{b_0^{(n)}, \cdots, b_{k-1}^{(n)}, b_{k+1}^{(n-1)}, \cdots, b_{M_bN-1}^{(n-1)}\right\},$$

the selection of the bit $b_k^{(n)}$ is based on the conditional probability distribution $\{\gamma_a, a = 0, 1\}$, where

$$\gamma_a = P\left(b_k = a|\bar{\mathbf{b}}_k, \mathbf{y}, \lambda_{2,k}^e\right).$$

For each $a = 0$, 1, we define

$$\mathbf{b}^a = \left\{b_0^{(n)}, \cdots, b_{k-1}^{(n)}, a, b_{k+1}^{(n-1)}, \cdots, b_{M_bN-1}^{(n-1)}\right\}.$$

We let $\mathbf{x}^a$ denote the symbol vector corresponding to $\mathbf{b}^a$. Assuming that b_k is mapped to the ith symbol, we have

$$\begin{aligned}\gamma_a &= P\left(b_k = a|\bar{\mathbf{b}}_k, \mathbf{y}, \lambda_{2,k}^e\right) \propto p(\mathbf{y}|\mathbf{b}^a)P\left(b_k = a|\lambda_{2,k}^e\right)\\ &= p(\mathbf{y}|\mathbf{x}^a)P\left(b_k = a|\lambda_{2,k}^e\right)\\ &= \prod_{j=0}^{N+L-1} p\left(y_j|\mathbf{x}_{j-L:j}^a\right) P\left(b_k = a|\lambda_{2,k}^e\right)\\ &= \left\{\prod_{j=0}^{i-1} p\left(y_j|\mathbf{x}_{j-L:j}^a\right) \prod_{j=i+L+1}^{N+L-1} p\left(y_j|\mathbf{x}_{j-L:j}^a\right)\right\}\\ &\quad \times \prod_{j=i}^{i+L} p\left(y_j|\mathbf{x}_{j-L:j}^a\right) P\left(b_k = a|\lambda_{2,k}^e\right). \end{aligned} \tag{6}$$

When $j \leq i-1$ or $j \geq i+L+1$, the vector $\mathbf{x}_{j-L:j}^a$ is independent of a. Thus, (6) is simplified to

$$\begin{aligned}\gamma_a &\propto \prod_{j=i}^{i+L} p\left(y_j|\mathbf{x}_{j-L:j}^a\right) P\left(b_k = a|\lambda_{2,k}^e\right)\\ &= C \cdot \exp\left\{\sum_{j=i}^{i+L}\left(-\frac{1}{2\sigma^2}\left|y_j - \sum_{l=0}^{L} h_l x_{j-l}^a\right|^2\right)\right.\\ &\qquad\qquad \left. + (-1)^a \lambda_{2,k}^e/2\right\} \end{aligned} \tag{7}$$

where C is a scaling constant to ensure that $\gamma_0 + \gamma_1 = 1$.

Algorithm 1: b-MCMC Equalizer

```
Input: Prior LLR for transmitted symbols(from the decoder)
Output: Extrinsic LLR for channel decoder
Use bit-wise Gibbs sampler to generate sample set
    // D parallel Markov chain
  repeat
    Generate initial sequence $\mathbf{b}^{(0)}$
    // I iterations
    for $n \leftarrow 1$ to $I$ do
      Generate $b_0^{(n)}$ from the distribution
        $P(b_0 = b|b_1^{(n-1)}, b_2^{(n-1)}, \cdots, b_{M_bN-1}^{(n-1)}, \mathbf{y}, \boldsymbol{\lambda}_2^e)$
      Generate $b_1^{(n)}$ from the distribution
        $P(b_1 = b|b_0^{(n)}, b_2^{(n-1)}, \cdots, b_{M_bN-1}^{(n-1)}, \mathbf{y}, \boldsymbol{\lambda}_2^e)$
        $\vdots$
       Generate $b_{M_bN-1}^{(n)}$ from the distribution
         $P(b_{M_bN-1} = b|b_0^{(n)}, b_1^{(n)}, \cdots, b_{M_bN-2}^{(n)}, \mathbf{y}, \boldsymbol{\lambda}_2^e)$
       Save $\mathbf{b}^{(n)}$ into the important set $\mathcal{I}$
  until D times;

Compute the LLR
  for $k \leftarrow 0$ to $M_bN-1$ do
    Construct $I_{i_1:i_2}^{k,0}$ and $\mathcal{I}_{i_1:i_2}^{k,1}$.
    Compute extrinsic LLR for $b_k$ using (8)

```

2) Computing the a posteriori LLR Values: Assume that the Gibbs sampler produces the important sample set $\mathcal{I}$. Each element in $\mathcal{I}$ is a bit vector of length M_bN. Assume that bit k is mapped to symbol x_i. Then, the received signals that are affected by b_k are $\mathbf{y}_{i:i+L}$. Since $\mathbf{y}_{i:i+L}$ depends only on bits $\{b_l, i_1 = M_b(i-L) \leq l \leq M_b(i+L+1)-1 = i_2\}$, we find that when computing the output LLR for b_k, it is sufficient to truncate each sequence in $\mathcal{I}$ to take into account only bits $\{b_l, i_1 \leq l \leq i_2\}$. We denote the set that contains the truncated sequences by $\mathcal{I}_{i_1:i_2}$. For each $0 \leq k \leq M_bN-1$, we construct a larger set $\mathcal{I}^k_{i_1:i_2}$ which includes all sequences in $\mathcal{I}_{i_1:i_2}$, together with new sequences that are obtained by flipping the kth bit of each sequence in $\mathcal{I}_{i_1:i_2}$. Repetitious sequences are removed from $\mathcal{I}^k_{i_1:i_2}$. Furthermore, we let $\mathcal{I}^{k,0}_{i_1:i_2}$ and $\mathcal{I}^{k,1}_{i_1:i_2}$ denote sequences in $\mathcal{I}^k_{i_1:i_2}$ whose kth bit equals 0 and 1, respectively. The LLR value for bit b_k is then computed as

$$\lambda^e_{1,k} = \ln \frac{\sum\limits_{\mathbf{b}_{i_1:i_2} \in \mathcal{I}^{k,0}_{i_1:i_2}} p\left(\mathbf{y}_{i:i+L} | \mathbf{b}_{i_1:i_2}\right) \prod\limits_{l=i_1}^{i_2} P\left(b_l | \lambda^e_{2,l}\right)}{\sum\limits_{\mathbf{b}_{i_1:i_2} \in \mathcal{I}^{k,1}_{i_1:i_2}} p\left(\mathbf{y}_{i:i+L} | \mathbf{b}_{i_1:i_2}\right) \prod\limits_{l=i_1}^{i_2} P\left(b_l | \lambda^e_{2,l}\right)} - \lambda^e_{2,k}. \tag{8}$$

The performance of the MCMC equalizer is dependent on the quality of samples in the important set. In practice, Gibbs sampler may require many iterations to converge to its stationary distribution. This is called burn-in period. As a result, including the burn-in period in the implementation of the Gibbs sampler may increase the complexity significantly. In [16], it has been shown empirically that the formulas such as (8) still work well if the stationary distribution of the underlying Markov chain is replaced by a uniform distribution over the significant samples (see [16, Fig. 2]). To obtain samples with this uniform distribution, it has also been noted in [16] that a set of parallel Gibbs samplers with no burn-in period and small number of iterations are more effective than using a single Gibbs sampler with many iterations. We have followed this implementation in this paper.

B. *Group-Wise MCMC Equalizer*

Our numerical results in Section IV show that while for channels with mild amplitude distortion the b-MCMC equalizer performs well, for channels with moderate or severe amplitude distortion, performance of the b-MCMC equalizer is compromised. This is because under such channel conditions, the Gibbs sampler suffers from slow-mixing problem when it gets stuck in certain states and cannot move freely. To improve the mixing rate and hence the speed of convergence, one can perform *group updating*, which updates a group of symbols simultaneously from their joint conditional distribution. It is shown in [25] and [26] that group updating indeed mix faster than ordinary single-updating scheme. Hence, we propose a group-wise MCMC equalization (g-MCMC) algorithm to deal with channels with moderate or severe amplitude distortion.

In g-MCMC, we group every G symbols $\mathbf{x}_{i:i+G-1} := \{x_i, \cdots, x_{i+G-1}\}$ together and update these symbols simultaneously. Assume that symbols $\{x_0, \cdots, x_{i-1}\}$ have been updated during the nth iteration, given by $x_0^{(n)}, \cdots x_{i-1}^{(n)}$, and symbols $\{x_{i+G}, \cdots, x_{N-1}\}$ have been last updated during the $(n-1)$th iteration, denoted by $x_{i+G}^{(n-1)}, \cdots x_{N-1}^{(n-1)}$. Conditioned upon $\{x_0^{(n)}, \cdots x_{i-1}^{(n)}, x_{i+G}^{(n-1)}, \cdots x_{N-1}^{(n-1)}\}$, we want to generate a random sample vector corresponding to $\mathbf{x}_{i:i+G-1}$ according to some probability distribution. We define the full sample space $\mathcal{S}_f = \{\mathbf{u}^1, \cdots, \mathbf{u}^r\}$ that contains all possible values of $\mathbf{x}_{i:i+G-1}$, where $r = 2^{M_bG}$. We compute the *a posteriori* probability distribution $\{\gamma_l, l = 1, \cdots, r\}$ where

$$\gamma_l = P\left(\mathbf{x}_{i:i+G-1} = \mathbf{u}^l | x_0^{(n)}, \cdots x_{i-1}^{(n)}, x_{i+G}^{(n-1)}, \cdots x_{N-1}^{(n-1)}, \mathbf{y}, \boldsymbol{\lambda}^e_2\right).$$

For each $l = 1, \cdots, r$, let

$$\mathbf{x}^l = \left(x_0^{(n)}, \cdots x_{i-1}^{(n)}, \mathbf{u}^l, x_{i+\mathrm{G}}^{(n-1)}, \cdots x_{N-1}^{(n-1)}\right).$$

By removing $\mathbf{u}^l$ (consisting of symbols i to $i+G-1$) from $\mathbf{x}^l$, we obtain

$$\bar{\mathbf{x}}_{i:i+G-1} = \left(x_0^{(n)}, \cdots x_{i-1}^{(n)}, x_{i+\mathrm{G}}^{(n-1)}, \cdots x_{N-1}^{(n-1)}\right).$$

We then have

$$\begin{aligned}
\gamma_l &= P\left(\mathbf{x}_{i:i+\mathrm{G}-1} = \mathbf{u}^l | \bar{\mathbf{x}}_{i:i+G-1}, \mathbf{y}, \boldsymbol{\lambda}^e_2\right) \\
&\propto P(\mathbf{y}|\mathbf{x}^l) P\left(\mathbf{x}_{i:i+G-1} = \mathbf{u}^l | \boldsymbol{\lambda}^e_2\right) \\
&= \prod_{j=0}^{N+L-1} p\left(y_j | \mathbf{x}^l_{j-L:j}\right) P\left(\mathbf{x}_{i:i+G-1} = \mathbf{u}^l | \boldsymbol{\lambda}^e_2\right) \\
&= \left\{ \prod_{j=0}^{i-1} p\left(y_j | \mathbf{x}^l_{j-L:j}\right) \prod_{j=i+L+G}^{N+L-1} p\left(y_j | \mathbf{x}^l_{j-L:j}\right) \right\} \\
&\quad \times \prod_{j=i}^{i+L+G-1} p\left(y_j | \mathbf{x}^l_{j-L:j}\right) P\left(\mathbf{x}_{i:i+G-1} = \mathbf{u}^l | \boldsymbol{\lambda}^e_2\right).
\end{aligned} \tag{9}$$

When $j \leq i-1$ or $j \geq i+L+G$, the vector $\mathbf{x}^l_{j-L:j}$ is independent of $\mathbf{u}^l$, thus is independent of l. Hence, (9) can be simplified as

$$\begin{aligned}
\gamma_l &\propto \prod_{j=i}^{i+G-1+L} p\left(y_j | \mathbf{x}^l_{j-L:j}\right) P\left(\mathbf{x}_{i:i+G-1} = \mathbf{u}^l | \boldsymbol{\lambda}^e_2\right) \\
&= C \cdot \exp\left\{ \sum_{j=i}^{i+G-1+L} \left(-\frac{1}{2\sigma^2} \left| y_j - \sum_{m=0}^{L} h_m x^l_{j-m} \right|^2 \right) \right. \\
&\qquad \left. + \sum_{m=0}^{M_bG-1} (-1)^{b^l_m} \lambda^e_{2,m}/2 \right\}
\end{aligned} \tag{10}$$

where $\{b^l_0, \cdots, b^l_{M_bG-1}\}$ denotes the bit vector corresponding to $\mathbf{u}^l$, C is a scaling constant such that $\sum_{l=1}^r \gamma_l = 1$. Here, we apply the independence assumption again such that $P(\mathbf{x}_{i:i+G-1} = \mathbf{u}^l | \boldsymbol{\lambda}^e_2) = \prod_{j=i}^{i+G-1} P(x_j = u^l_{j-i} | \boldsymbol{\lambda}^e_2) = \prod_{m=0}^{M_bG-1} P(b^l_m | \boldsymbol{\lambda}^e_2)$.

When the size of the full sample space $\mathcal{S}_f$ is large, we can use a QRD-M algorithm [14] described in Section III-C to reduce the complexity of the g-MCMC. The resulting g-MCMC is summarized in Algorithm 2, where $a\%b$ denotes the remainder of a divided by b, and $\lfloor a \rfloor$ denotes the maximum integer less than a. Note that line 5 of Algorithm 2 allows us to group different adjacent symbols over iterations. As shown in Section IV, this is necessary to speed up the mixing rate of Gibbs sampler. After the sample set $\mathcal{I}$ is generated, the g-MCMC follows the same procedure as b-MCMC to compute the extrinsic LLR value of each bit.

Algorithm 2: Group-wise MCMC Equalizer

```
Input: Prior LLR for transmitted bits (from channel decoder)
Output Extrinsic LLR for channel decoder
Use group-wise Gibbs sampler to generate sample set
  repeat
    Generate initial sequence $\mathbf{x}^{(0)}$;
    for $n \leftarrow 1$ to $I$ do
      $m_0 \leftarrow n\%G + 1$, $J \leftarrow \lfloor (N - m_0)/G \rfloor$;
      Generate $\mathbf{x}_{0:m_0-1}^{(n)}$ from the distribution
        $P(\mathbf{x}_{0:m_0-1} | x_{m_0}^{(n-1)}, \cdots, x_{N-1}^{(n-1)}, \mathbf{y}, \boldsymbol{\lambda}_2^e)$
      Generate $\mathbf{x}_{m_0:m_0+G-1}^{(n)}$ from the distribution
        $P(\mathbf{x}_{m_0:m_0+G-1} | x_0^{(n)}, \cdots, x_{m_0-1}^{(n)}, x_{m_0+G}^{(n-1)}, \cdots,$
        $x_{N-1}^{(n-1)}, \mathbf{y}, \boldsymbol{\lambda}_2^e)$
          ⋮
       Generate $\mathbf{x}_{m_0+JG:N-1}^{(n)}$ from the distribution
         $P(\mathbf{x}_{m_0+JG:N-1} | x_0^{(n)}, \cdots, x_{m_0+JG-1}^{(n)}, \mathbf{y}, \boldsymbol{\lambda}_2^e)$
       Save $\mathbf{x}^{(n)}$ into the important set $\mathcal{I}$
  until $D$ times;

SubFunction: Generate $\mathbf{x}_{i:i+G-1}$ from the distribution $P$
  if $r$ is small (e.g., $r < 64$) then
    for each $\boldsymbol{x}_{i:i+G-1} = u^l$ do
      Compute $\gamma_l$ using (10).
    Generate one sample vector randomly according to
    PMF $\gamma = \{\gamma_1, \cdots, \gamma_r\}$.
  else
    Perform QRD-M to generate reduced sample space $S_i$
    containing $r_0$ samples
    Generate one sample vector from $S_i$ according to PMF
    (10).

Compute the LLR
  for $k \leftarrow 0$ to $M_bN - 1$ do
    Construct $\mathcal{I}_{i_1:i_2}^{k,0}$ and $\mathcal{I}_{i_1:i_2}^{k,1}$.
    Compute extrinsic LLR for $b_k$ using (8)
```

C. Complexity Reduction of g-MCMC Using QRD-M

The complexity for computing $\{\gamma_1, \cdots, \gamma_r\}$ using (10) becomes high when $r = 2^{M_bG}$ is large. For small values of r, we can still consider the full sample space S_f. When r is large, instead of S_f, we introduce a reduced sample space S_i for $\mathbf{x}_{i:i+G-1}$ that contains r_0 sample vectors with large *a posteriori* probabilities (APPs). Typically r_0 is much smaller than r. We propose to adopt the QR decomposition and M-algorithm (QRD-M)[1] to find S_i. The QRD-M algorithm selects only M branches with the smallest accumulated metric at each level of the tree search. Hence, it has a constant complexity that is controlled by the M parameter. Larger values of M improve the algorithm performance at the cost of increased complexity. The QRD-M algorithm has been widely applied in MIMO detection [14]. Next, we show that the QRD-M algorithm is also applicable to channels with ISI.

The set of (1) can be represented in a matrix form as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \tag{11}$$

where $\mathbf{z} = [z_0, \cdots, z_{N+L-1}]^{\mathrm{T}}$, and

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & \cdots & & 0 \\ h_1 & h_0 & 0 & \cdots & & & 0 \\ \vdots & \ddots & & \ddots & & & \vdots \\ 0 & h_L & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \cdots & 0 & h_L & h_{L-1} & \cdots & h_0 \\ 0 & \cdots & & 0 & \ddots & & \vdots \\ 0 & \cdots & & & 0 & h_L & h_{L-1} \\ 0 & \cdots & & \cdots & & 0 & h_L \end{bmatrix} \tag{12}$$

is a $N + L$ by N channel matrix. We first simplify (10) as follows. Let $P = G + L - 1$. Since $\mathbf{x}_{i-L:i-1}^l$ and $\mathbf{x}_{i+G:i+P}^l$ are independent of l, we can drop the superscript l in these vectors and subtract their contribution from the received signals by letting

$$\tilde{\mathbf{y}}_{i:i+P} = \mathbf{y}_{i:i+P} - \mathbf{H}_{i:i+P}^{i-L:i-1}\mathbf{x}_{i-L:i-1} - \mathbf{H}_{i:i+P}^{i+G:i+P}\mathbf{x}_{i+G:i+P} \tag{13}$$

where $\mathbf{H}_{i_1:i_2}^{j_1:j_2}$ denotes the submatrix of $\mathbf{H}$ with rows i_1 to i_2 and columns j_1 to j_2. For notational convenience, when $j_1 < 0$ or $j_2 > N + L - 1$, the corresponding column of $\mathbf{H}$ is assumed to be zero. The received signal $\tilde{\mathbf{y}}_{i:i+P}$ in (13) can be thought of received signals of a MIMO channel with G transmit antenna and $P + 1$ receive antenna, where the transmitted signal vector is $\mathbf{x}_{i:i+G-1}$, and the channel matrix is given by $\mathbf{H}_{i:i+P}^{i:i+G-1}$. Hence, we can directly apply the QRD-M algorithm to $\tilde{\mathbf{y}}_{i:i+P}$ to find r_0 samples with large APPs. We then scale the APPs of these samples to get a desired probability distribution $\{\gamma_l, l = 1, \cdots, r_0\}$, based on which the Gibbs sampler will generate a random sample from $\mathcal{S}_i$ for $\mathbf{x}_{i:i+G-1}$.

D. Parallel Implementation of MCMC Equalizer

In Sections III-A and III-B, we presented the serial implementation of the b-MCMC and g-MCMC, where we run each Gibbs sampler over the entire transmitted sequence $\{x_0, \cdots, x_{N-1}\}$ during which each bit or each group of symbols is updated sequentially. This sequential processing is repeated over I iterations for each Gibbs sampler. From an implementation point of view, sequential processing requires large number of clock cycles that is proportional to the sequence length N. This in turn means a long processing time or, equivalently, long delay in the detection path. This, of course, is undesirable and should be avoided if possible.

[1] One may also use b-MCMC or s-MCMC in place of QRD-M to generate the reduced sample space S_i. For the test cases studied in this paper (see Section IV), we empirically found that QRD-M is a superior choice.

TABLE I
SAMPLES UPDATING IN PARALLEL GIBBS SAMPLER

		Gibbs sampler 1		Gibbs sampler 2	
Iteration n	Time t	Update sample	Conditioned upon samples	Update sample	Conditioned upon samples
0	0	Initialized to $x_0^{(0)}, x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}, x_5^{(0)}, x_6^{(0)}, x_7^{(0)}$			
1	1	$x_0^{(1)}$	$x_1^{(0)}, x_2^{(0)}$	$x_4^{(1)}$	$x_2^{(0)}, x_3^{(0)}, x_5^{(0)}, x_6^{(0)}$
	2	$x_1^{(1)}$	$x_0^{(1)}, x_2^{(0)}, x_3^{(0)}$	$x_5^{(1)}$	$x_3^{(0)}, x_4^{(1)}, x_6^{(0)}, x_7^{(0)}$
	3	$x_2^{(1)}$	$x_0^{(1)}, x_1^{(1)}, x_3^{(0)}, x_4^{(1)}$	$x_6^{(1)}$	$x_4^{(1)}, x_5^{(1)}, x_7^{(0)}$
	4	$x_3^{(1)}$	$x_1^{(1)}, x_2^{(1)}, x_4^{(1)}, x_5^{(1)}$	$x_7^{(1)}$	$x_5^{(1)}, x_6^{(1)}$

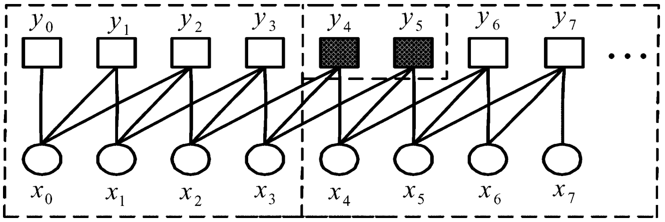


Fig. 2. Graphical representation of an ISI channel with $L = 2$. A transmitted sequence of length 8 is divided into two subblocks each with a length of $n_0 = 4$. Symbol x_3 from Gibbs sampler 1 affects y_4 and y_5, which depends on the values of x_4 and x_5 (belonging to Gibbs sampler 2).

We solve the problem discussed above by introducing a parallel implementation of the Gibbs sampler. For ease of disposition, let us assume BPSK modulation and consider the parallel implementation of a b-MCMC. Here each transmitted symbol represents one bit. We first divide the transmitted symbol vector $\mathbf{x}$ into subblocks each containing n_0 symbols. Instead of running a single Gibbs sampler over the entire sequence of $\mathbf{x}$, we run parallel Gibbs samplers, one for each subblock. Within each subblock, the Gibbs sampler updates the n_0 symbols sequentially from left to right. Due to the channel memory, the APP of a given symbol in a subblock may depend on the values of some symbols belonging to other subblocks. To speed up the convergence rate of the Gibbs sampler, it is helpful to synchronize the parallel Gibbs samplers and have the updated symbol values available to the neighboring subblocks. To show how the parallel implementation works, we examine the example shown in Fig. 2 where $L = 2$, $N = 8$, and $n_0 = 4$. Two Gibbs samplers are run in parallel. The symbol updating process of these two Gibbs samplers is shown in Table I. At time 0, the samples are initialized randomly to obtain $x_k^{(0)}$. Subsequently, at each time instance, each Gibbs sampler updates a sample conditioned upon previously updated samples in parallel. For instance, at time $t = 3$ of the first iteration $n = 1$, Gibbs sampler 1 updates bit $x_2^{(1)}$, where the superscript is the iteration index, and the subscript is the index for the bit that is being updated. Bit $x_2^{(1)}$ is generated conditioned upon the values of four bits, where $x_0^{(1)}, x_1^{(1)}$ were updated at $t = 1$ and $t = 2$, respectively, during iteration 1 by Gibbs sampler 1; $x_3^{(0)}$ was updated at $t = 0$ during iteration 0; $x_4^{(1)}$, which belongs to the second Gibbs sampler, was updated at time $t = 1$ during iteration $n = 1$. This procedure is then repeated for a total of I iterations to find the important sample set $\mathcal{I}$.

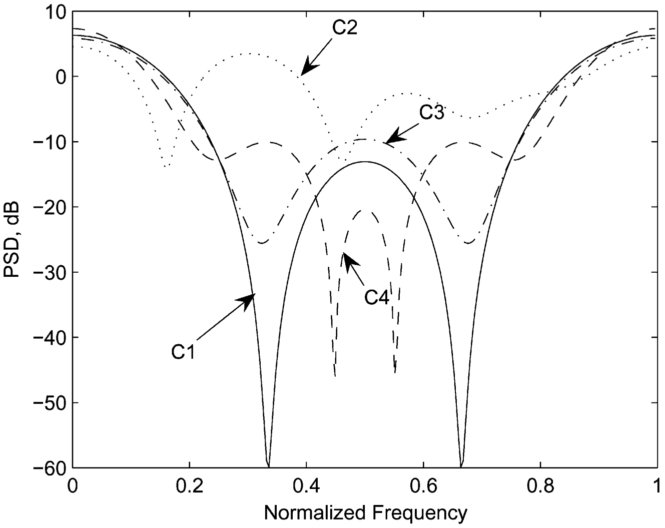


Fig. 3. Power spectral density function at the output of four ISI channels C1, C2, C3 and C4. Note that C1 has a severe amplitude distortion, C2 has a mild amplitude distortion, C3 and C4 have moderate amplitude distortion.

IV. SIMULATION RESULTS

In this section, we present simulation results of the proposed MCMC equalizers. We consider four ISI channels with channel impulse responses:

$$\begin{aligned}h_1[n] =& 0.227\delta[n] + 0.46\delta[n-1] + 0.688\delta[n-2]\\&+ 0.46\delta[n-3] + 0.227\delta[n-4]\\h_2[n] =& (2+0.4i)\delta[n] + (1.5+1.8i)\delta[n-1] + \delta[n-2]\\&+ (1.2-1.3i)\delta[n-3] + (0.8+1.6)\delta[n-4]\\h_3[n] =& 0.227\delta[n] + 0.406\delta[n-1] + 0.688\delta[n-2]\\&+ 0.406\delta[n-3] + 0.227\delta[n-4]\\h_4[n] =& 0.16\delta[n] + 0.227\delta[n-1] + 0.460\delta[n-2]\\&+ 0.688\delta[n-3] + 0.460\delta[n-4] + 0.227\delta[n-5]\\&+ 0.16\delta[n-6].\end{aligned}$$

The first two channels are taken from [27] and [5]. Assuming that the transmit symbols x_i are i.i.d. random variables and ignoring the channel noise, the power spectral density at output of these channels is presented in Fig. 3. As seen, Channel 1 (C1) has deep spectral nulls, which corresponds to a high channel condition number (defined in Section I) and significant amplitude distortion. Channel 2 (C2) has a mild amplitude distortion.

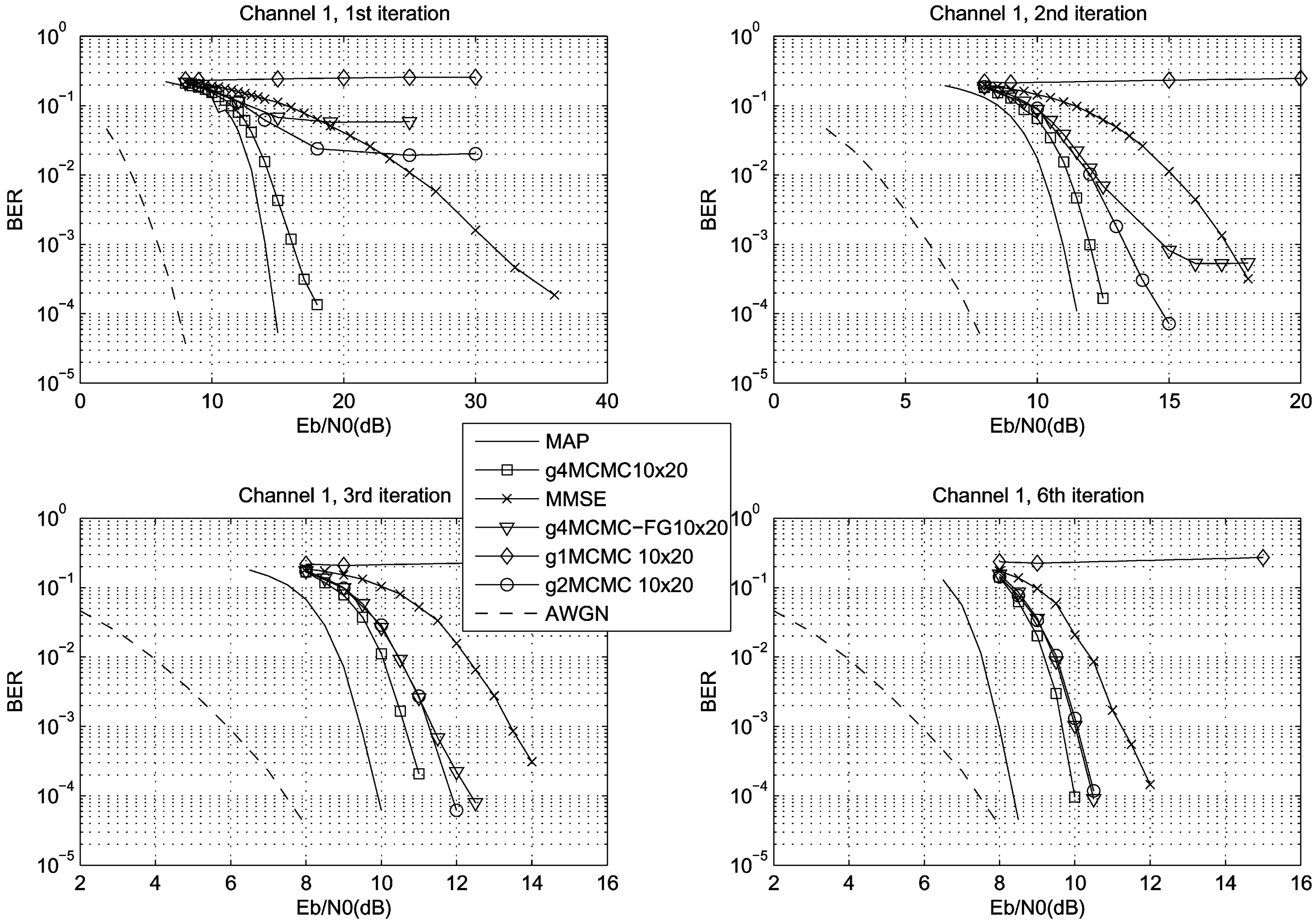


Fig. 4. Performance comparisons of various equalizers over ISI channel C1 which has a severe amplitude distortion.

The amount of amplitude distortion for Channel 3 (C3) lies in between that of C1 and C2. Channel 4 (C4) has a memory of $L = 6$ and has moderate amplitude distortion. The channel response is assumed to be perfectly known to the receiver. We will compare the performance of various proposed MCMC detectors with the exact soft MMSE [7] equalizer and the MAP equalizer over these four ISI channels with different levels of amplitude distortion. The simulation set up is the same as that of [7]. A rate $R = 1/2$ convolutional code with generator polynomials $(1 + D^2, 1 + D + D^2)$ is used. The coded bit length is 4098. The bit sequence is mapped to a sequence of 8-PSK symbols of length $N = 4098/3 = 1366$ using gray mapping. The channel interleaver is an S-random interleaver [28] with $S = 0.5\sqrt{0.5K_c}$, where K_c is the number of coded bits. At the receiver we perform six iterations of channel equalization and MAP decoding.

In Fig. 4, we examine the equalization performance over the severely distorted ISI channel C1. We plot the bit-error-rate (BER) curves (shown as solid lines) of the optimal MAP detector, the MMSE detector [7], and 10 × 20 (10 parallel Gibbs samplers with 20 iterations each) g-MCMC detectors (G = 1, 2, 4), after one, two, three, six iterations of channel equalization and decoding. The g1-MCMC $(G = 1)$ operates symbol-wise inside the Gibbs sampler. The QRD-M algorithm with $M = 8$ is applied to the g4-MCMC $(G = 4)$ and g2-MCMC $(G = 2)$ to generate the reduced sample space $\mathcal{S}_i$, consisting of $r_0 = 25$ samples with the largest APPs at the end of the QRD-M algorithm. For the case of g4-MCMC, the size of the full sample space is $r = 2^{M_b G} = 2^{12} = 4096$ and, thus, the number of samples in $\mathcal{S}_i$ is only about 0.6% of that of $\mathcal{S}_f$. Assuming that the transmitted symbols x_i has unit energy, the average energy per bit to noise ratio is defined as

$$\frac{E_b}{N_0} \triangleq \frac{E_s}{N_0 R M_b} = \frac{\sum_{l=0}^{L-1} |h_l|^2}{2\sigma^2 R M_b}.$$

The main observations from Fig. 4 are summarized as follows.

1) For a severely distorted ISI channel, neither b-MCMC (not shown in the figure) nor g1-MCMC (s-MCMC) works. The g4-MCMC performs the best among the MCMC detectors considered. Performance gap between g4-MCMC and g2-MCMC reduces after six iterations.
2) The g4-MCMC detector significantly outperforms the MMSE detector of [7]. After six iterations, the g4-MCMC is about 2 dB better than the MMSE at $\mathrm{BER} = 10^{-4}$. More iterations does not reduce this performance gap (a performance curve of the MMSE detector after ten iterations is shown in the last subfigure of Fig. 4).
3) The g4-MCMC detectors performs closely to the optimal MAP detector. The performance gap at $\mathrm{BER} = 10^{-4}$ is

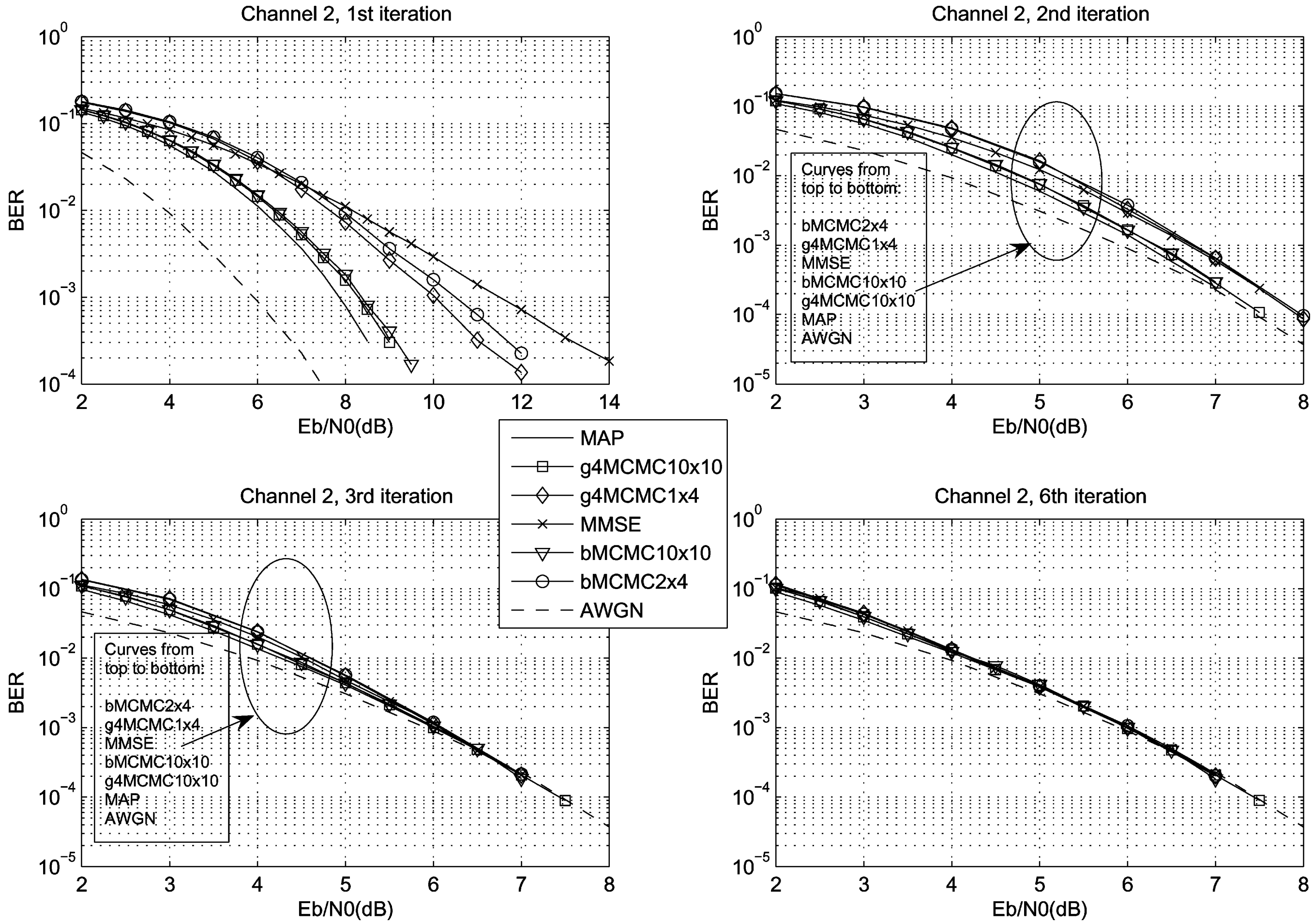


Fig. 5. Performance comparisons of various equalizers over ISI channel C2 which has a mild amplitude distortion.

about 1.7 dB after six iterations. With only one iteration, the g4-MCMC performs within 3 dB of the MAP detector, while the MMSE performs about 18 dB worse.

4) To optimize performance of g4-MCMC, it is necessary to group different adjacent symbols over iterations (see Line 5 of Algorithm 2). The performance of g4-MCMC with fixed grouping (i.e., grouping the same set of symbols over iterations), denoted by g4-MCMC-FG, is much worse than the g4-MCMC with varying grouping, especially for small number of iterations.

In Fig. 4, we also present the BER curve (shown as the dashed line) for an AWGN channel with no ISI, i.e., $h(n) = \delta(n)$. This is also called the matched filter bound (MFB) which provides a performance lower bound for the ISI channel. We observe from Fig. 4 that the MAP detector approaches the MFB after six iterations of joint detection and channel decoding.

Fig. 5 shows the BER performance over channel C2. Since C2 has a mild amplitude distortion, all the detectors considered (g4-MCMC, b-MCMC, and MMSE) can approach the performance of MAP equalizer after six iterations and they all approach the MFB. We also note that MCMC detectors with small parameters (1 × 4 for g4-MCMC and 2 × 4 for b-MCMC) are sufficient to obtain good performance after six iterations. This leads to very low detection complexity. After only one iteration, both the 10 × 10 g4-MCMC and 10 × 10 b-MCMC perform closely to the MAP detector, while the MMSE is more than 2 dB worse.

In Fig. 6, we examine the performance of various equalizers over channel C3 which has a moderate amplitude distortion. Here, all MCMC equalizers have parameters 10 × 10. It is shown that the b-MCMC performs the worst and has an error floor at $\mathrm{BER} = 10^{-3}$. The g1-MCMC performs nearly 4 dB worse than the g2-MCMC and g4-MCMC. The MMSE detector is about 1.5 dB worse than the g4-MCMC at $\mathrm{BER} = 4 \times 10^{-5}$. The MAP detector also approaches the MFB after six iterations.

In Fig. 7, we consider an ISI channel C4 with a longer channel memory of $L = 6$. The amount of amplitude distortion for C4 is shown to be moderate (see Fig. 3). Since the optimal MAP detector needs to operate on a trellis with $8^6 = 2^{18}$ states, it becomes computational prohibitive to operate this detector over C4. Fig. 7 shows that g-MCMC with small parameters still work well for this channel. In particular, after six iterations, the g4-MCMC 10 × 20 with $M = 8$ outperforms the MMSE (with a filter length of 21) by about 1.5 dB at $\mathrm{BER} = 10^{-4}$. The g6-MCMC 10 × 20 with $M = 16$ improves performance slightly at the cost of additional complexity. Note that neither g1-MCMC (s-MCMC) nor b-MCMC (not shown here) works for this channel. We note that the gap between the MFB and that of the g4-MCMC/g6-MCMC reduces at higher SNRs.

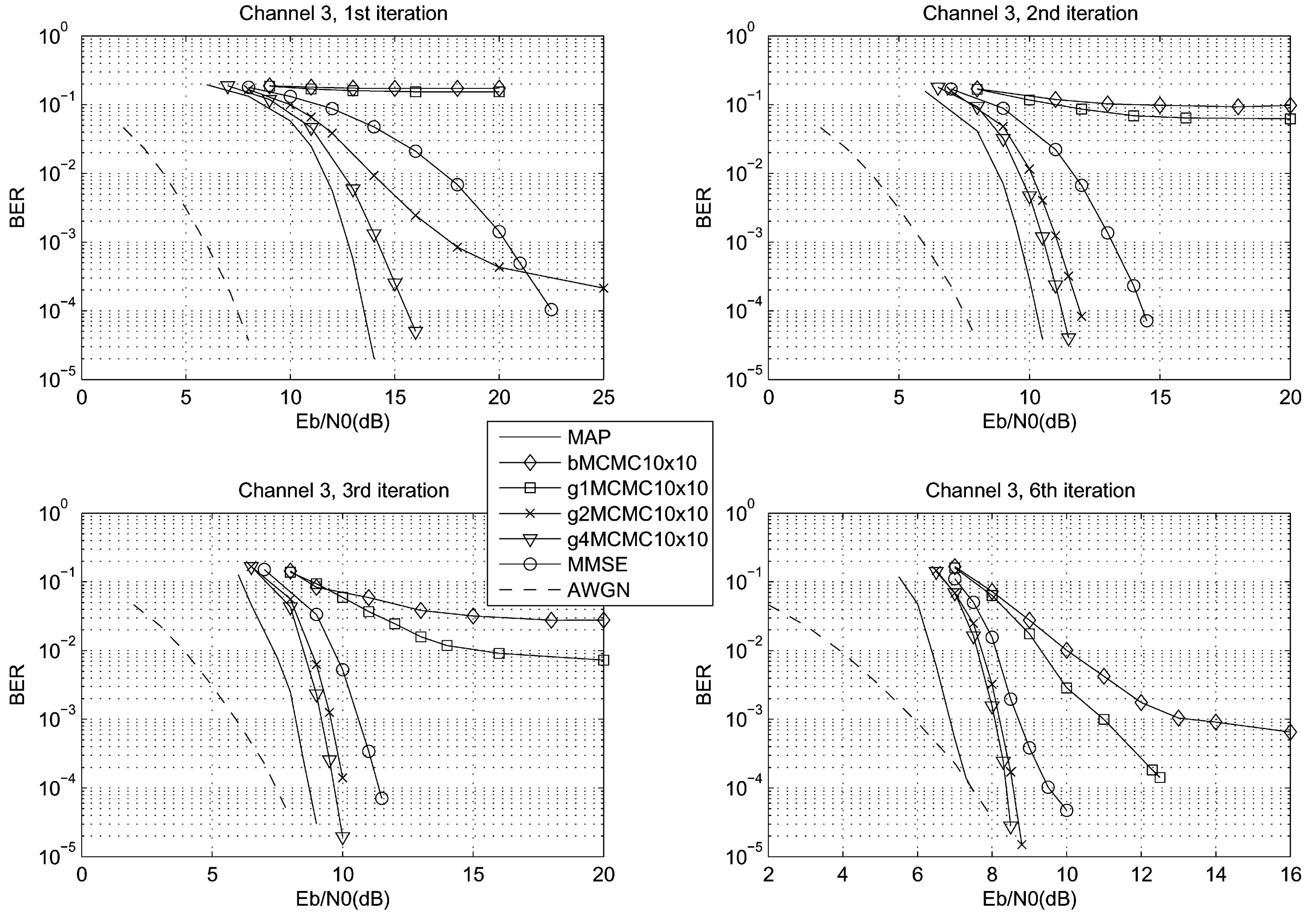


Fig. 6. Performance comparisons of the different equalizers over ISI channel C3 which has a moderate amplitude distortion.

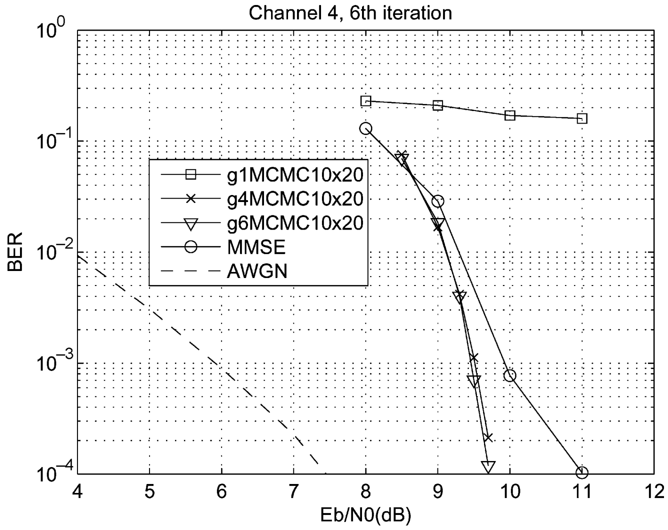


Fig. 7. Performance comparisons of the different equalizers over ISI channel C4 with a longer memory of $L = 6$ and moderate amplitude distortion.

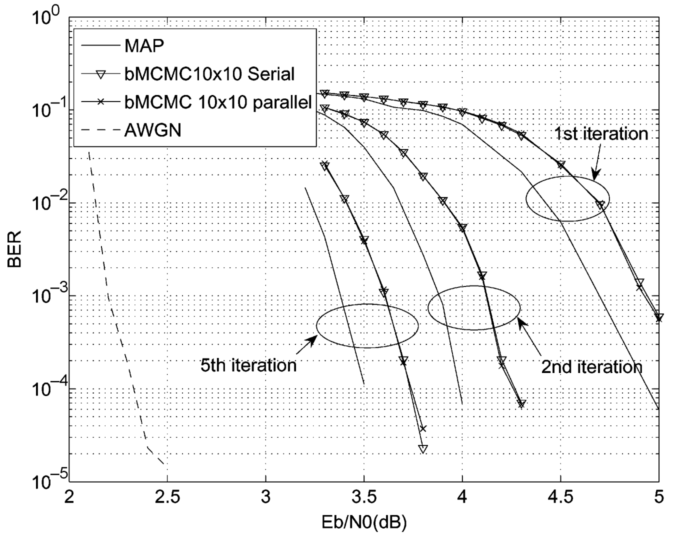


Fig. 8. Performance comparisons of b-MCMC with parallel/serial implementation over ISI channel C2.

Finally, we compare performance of serial and parallel implementation of the MCMC detector. Fig. 8 shows performance curves of a 10×10 b-MCMC detector over C2. Since a rate 1/2 turbo code is employed here, the operating SNR is lower than that of Fig. 5 employing a convolutional code. The length of the coded bit sequence is 7800 bits. Assuming 8-PSK modulation, the length of the symbol sequence is $7800/3 = 2600$ symbols. For the parallel implementation, the sequence of $2600 = 520 \times 5$

received signal samples is divided into 520 subblocks, each containing five samples. The b-MCMC equalizer is applied to these 520 subblocks in parallel. We perform five outer iterations of joint equalization and channel decoding. For each outer iteration, channel equalization is performed once followed by eight inner iterations of turbo decoding. From Fig. 8, we find that the parallel implementation achieves almost identical performance as the serial implementation while significantly reducing processing delay. At $\mathrm{BER}=10^{-4}$, the 10×10 b-MCMC equalizer is only 0.23 dB worse than the MAP detector. We observe a 1 dB gap between the MFB and that of the MAP detector, which is much larger than the gap shown in Fig. 5 for the convolutional coded system. We believe that this is because the turbo coded system operates at a much lower SNR, in which case a precoder may be needed to further improve performance of the turbo equalization in order to better approach the MFB [6], [29]. We note that a 2 dB gap between the MFB and that of the MAP detector is also observed in [30] for a turbo coded system.

V. Conclusion

In this paper, we demonstrated that MCMC techniques are highly effective in the design of low-complexity equalizers for frequency selective channels. This work extends previous work on b-MCMC MIMO detectors for frequency flat channels to channels with ISI. We showed that the b-MCMC equalizer works well for channels with mild amplitude distortion, however, fails for channels with moderate or severe amplitude distortion. The g-MCMC, on the other hand, is robust to the different levels of channel amplitude distortion. For several ISI channels examined in this work, the MCMC detectors demonstrate significant performance gain over the widely used turbo MMSE equalizer. The proposed MCMC detector enables parallel processing, which makes it amenable for practical implementation. Extensions of this work to channels with longer memory, and to channels with unknown impulse response are the subject of future research.

Acknowledgment

The authors would like to thank Dr. M. Tüchler for providing them with the Matlab program for the MMSE detector. The authors would also like to thank the anonymous reviewers for valuable comments that lead to a greatly improved manuscript.

References

[1] R. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, pp. 547–588, Apr. 1965.

[2] M. E. Austin, "Decision feedback equalization for digital communication over dispersive channels," MIT Lincoln Lab., Lexington, MA, Tech. Rep., Aug. 1967.

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

[4] C. Douillard *et al.*, "Iterative correction of intersymbol interference: Turbo equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sept.–Oct. 1995.

[5] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *Proc. Int. Symp. Turbo Codes, Related Topics*, Brest, France, Sep. 1997, pp. 96–102.

[6] M. Tuchler, R. Koetter, and A. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.

[7] M. Tuchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using *a priori* information," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

[8] V. Franz and J. B. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.

[9] J. Hagenauer and C. Kuhn, "Turbo equalization for channels with high memory using list-sequential (LISS) equalizer," in *Proc. 3rd Int. Symp. Turbo Codes Related Topics (ISTC)*, Brest, France, Sep. 2003, pp. 9–13.

[10] C. Kuhn and J. Hagenauer, "8-PSK turbo equalization with the list-sequential (LISS) algorithm," in *Proc. Int. Symp. Information Theory (ISIT)*, Jun. 27–Jul. 2, 2004, p. 555.

[11] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 152–170, Nov. 2005C. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[12] R. Peng, R.-R. Chen, and B. Farhang-Boroujeny, "Low complexity Markov chain monte carlo detector for channels with intersymbol interference," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dresden, Germany, Jun. 14–18, 2009.

[13] F. M. Kashif, H. Wymeersch, and M. Z. Win, "Monte carlo equalization for nonlinear dispersive satellite channels," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 245–255, Feb. 2008.

[14] K. J. Kim and J. Yue, "Joint channel estimation and data detection algorithms for MIMO-OFDM systems," in *Proc. 36th Asilomar Conf. Signals, Systems, Computers*, 2002, pp. 1857–1861.

[15] Z. Shi, H. Zhu, and B. Farhang-Boroujeny, "Markov chain Monte Carlo techniques in iterative detectors," in *Proc. IEEE GLOBECOM*, Dallas, TX, Nov./Dec. 2004, pp. 325–329.

[16] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," *IEEE Trans. Signal. Process.*, vol. 54, no. 5, pp. 1896–1909, May 2006.

[17] H. Zhu, B. Farhang-Beroujeny, and R. R. Chen, "On performance of sphere decoding and Markov chain Monte Carlo methods," *IEEE Signal Process. Lett.*, vol. 12, no. 10, pp. 669–672, Oct. 2005.

[18] S. A. Laraway and B. Farhang-Boroujeny, "Implementation of a Markov chain Monte Carlo based multiuser/MIMO detector," in *Proc. IEEE Int. Conf. Communications (ICC)*, Istanbul, Turkey, Jun. 2006, vol. 7, pp. 3088–3093.

[19] R.-R. Chen, R. Peng, A. Ashikhmin, and B. Farhang-Beroujeny, "Approaching MIMO capacity using bitwise Markov chain Monte Carlo detection," *IEEE Trans. Commun.*, vol. 58, no. 2, pp. 423–428, Feb. 2010.

[20] T. Hashimoto, "A list-type reduced-constraint generalization of the Viterbi algorithm," *IEEE Trans. Inf. Theory*, vol. 33, no. 6, pp. 866–876, Nov. 1987.

[21] R. E. Kamel and Y. Bar-Ness, "Reduced-complexity sequence estimation using state partitioning," *IEEE Trans. Commun.*, vol. 44, no. 9, pp. 1057–1063, Sep. 1996.

[22] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 354–364, Feb./Mar./Apr. 1995.

[23] T. M. Aulin, "Breadth-first maximum likelihood sequence detection: Basics," *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 208–216, Feb. 1999.

[24] M. Ruan, M. C. Reed, and Z. Shi, "Turbo equalization using particle filter with grouping," in *Proc. Int. Symp. Communications Information Technologies (ISCIT)*, Oct. 2007, pp. 1197–1200.

[25] J. S. Liu, W. H. Wong, and A. Kong, "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes," *Biometrika*, vol. 81, pp. 27–40, 1994.

[26] G. O. Roberts and S. K. Sahu, "Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler," *J. Roy. Stat. Soc. B*, vol. 59, pp. 291–317, 1997.

[27] J. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[28] C. Heegard and S. Wicker, *Turbo Coding*. Boston, MA: Kluwer, 1999.

[29] I. Lee, "The effect of a precoder on serially concatenated coding systems with an ISI channel," *IEEE Trans. Commun.*, vol. 49, no. 7, pp. 1168–1175, Jul. 2001.

[30] D. Raphaeli and Y. Zarai, "Combined turbo equalization and turbo decoding," *IEEE Commun. Lett.*, vol. 2, no. 4, pp. 107–109, Apr. 1998.

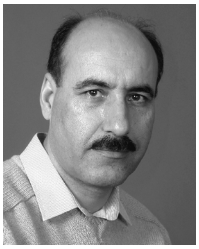
Rong-Hui Peng received the B.S. degree in electrical engineering from Chongqing University, Chongqing, China, in 1998 and the M.S. degree from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2002. He is currently working towards the Ph.D. degree at the University of Utah, Salt Lake City.

From 2002 to 2004, he was a Research Engineer with LG Electronic (China) R&D center, Beijing, China. During summer 2004, he was with T3G Technology Company, Ltd., Beijing, China. From 2004 to 2009, he was a Research Assistant with the University of Utah. During fall 2007, He worked in the Digital Communications group at Mitsubishi Electric Research Laboratories, Cambridge, MA. Since 2009, he has been a System Engineer in SpiralGen, Inc. His research interests include code design and analysis of LDPC codes, iterative decoding, low-complexity MIMO detection, software defined radio, and algorithm mapping to multicore processor.

Rong-Rong Chen (S'03–M'03) received the Bachelor's degree in applied mathematics from TsingHua University, China, in 1994 and the Master's degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign (UIUC), in 1996 and 2003, respectively.

Since August 2003, she has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Utah, Salt Lake City. Her current research interests include wireless communication, coding, and information theory, and queueing networks. She was the recipient of the University Fellowship for graduate study in the Mathematics Department at UIUC and the M. E. Van Valkenburg Graduate Research Award for excellence in doctoral research in the Department of Electrical and Computer Engineering at UIUC. Her main research interest is in the general area of wireless communication and signal processing.

Dr. Chen is a recipient of the prestigious National Science Foundation (NSF) Faculty Early Career Development (CAREER) award in 2006.



Behrouz Farhang-Boroujeny (M'84–SM'90) received the B.Sc. degree in electrical engineering from Teheran University, Iran, in 1976, the M.Eng. degree from University of Wales Institute of Science and Technology, U.K., in 1977, and the Ph.D. degree from Imperial College, University of London, U.K., in 1981.

From 1981 to 1989, he was with the Isfahan University of Technology, Isfahan, Iran. From 1989 to 2000, he was with the National University of Singapore. Since August 2000, he has been with the University of Utah, where he is now a Professor and Associate Chair of the department. He is an expert in the general area of signal processing. His current scientific interests are adaptive filters, multicarrier communications, detection techniques for space-time coded systems, cognitive radio, and signal processing applications to optical devices. In the past, he has worked and has made significant contribution to areas of adaptive filters theory, acoustic echo cancellation, magnetic/optical recoding, and digital subscriber line technologies. He is the author of the books *Adaptive Filters: Theory and Applications* (Wiley, 1998) and *Signal Processing Techniques for Software Radios* (self-published at Lulu publishing house, 2009).

Dr. Farhang-Boroujeny received the UNESCO Regional Office of Science and Technology for South and Central Asia Young Scientists Award in 1987. He served as Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from July 2002 to July 2005. He is now serving as Associate Editor of the IEEE SIGNAL PROCESSING LETTERS. He has also been involved in various IEEE activities, including the chairmanship of the Signal Processing/Communications chapter of IEEE of Utah in 2004 and 2005.