BLOCK EXPECTATION PROPAGATION EQUALIZATION FOR ISI CHANNELS

Irene Santos[*], *Juan José Murillo-Fuentes*[*], *Pablo M. Olmos*[†]

[*] ETSI, Universidad de Sevilla
Dept. Teoría de la Señal y Comunicaciones
Camino de los Descubrimientos s/n,
41092 Sevilla, Spain.
E-mail: {irenesantos,murillo}@us.es

[†] Universidad Carlos III de Madrid
Dept. Teoría de la Señal y Comunicaciones
Avda. de la Universidad 30,
28911, Leganés (Madrid), Spain.
E-mail: olmos@tsc.uc3m.es

ABSTRACT

Actual communications systems use high-order modulations and channels with memory. However, as the memory of the channels and the order of the constellations grow, optimal equalization such as BCJR algorithm is computationally intractable, as their complexity increases exponentially with the number of taps and size of modulation. In this paper, we propose a novel low-complexity hard and soft output equalizer based on the Expectation Propagation (EP) algorithm that provides high-accuracy posterior probability estimations at the input of the channel decoder with similar computational complexity than the linear MMSE. We experimentally show that this quasi-optimal solution outperforms classical solutions reducing the bit error probability with low complexity when LDPC channel decoding is used, avoiding the curse of dimensionality with channel memory and constellation size.

***Index Terms*—** Expectation propagation, BCJR algorithm, low complexity, channel equalization, ISI.

1. INTRODUCTION

Single input single output (SISO) communication channels are corrupted by additive white Gaussian noise (AWGN) and introduce inter-symbol interference (ISI) between transmitted symbols, due to its dispersive nature and the multiple paths of wireless communications [1]. Channel equalization is a solution to this problem, which provides estimations of the transmitted symbols and exploit diversity. Furthermore, rather than hard decision on the received symbols, nowadays channel decoders highly benefit from probabilistic estimates for each transmitted symbol given the received sequence [2].

Consider a discrete-time dispersive digital communication system, where the channel is completely defined by the channel state information (CSI) which is known at the receiver. Assuming perfect CSI and a channel with finite memory, linear equalization, such as the linear minimum-mean-squared-error (LMMSE) [1], is a low-cost alternative based on the minimization of the signal error. However, its results are far from the optimal solution provided by symbol maximum a posteriori (MAP) BCJR algorithm. The BCJR algorithm [3] computes the a posteriori probabilities (APP) for each transmitted symbol providing optimal decisions

$$p(u_k = \mathcal{A}|\mathbf{h}, \mathbf{y}) \quad \forall\, k = 1, ..., N \tag{1}$$

where $\mathbf{u}$ is the block frame transmitted taken from an N-dimensional alphabet $\mathcal{A}^N$ (of a M-ary constellation, i.e., of order $|\mathcal{A}| = M$), $\mathbf{h}$ is the CSI of a channel with L taps and $\mathbf{y}$ is the received sequence.

The BCJR algorithm works on a trellis representation and its complexity is proportional to the number of states. This number increases with the number of taps of the channel and the size of the constellation used. Specifically, for each symbol we have M^{L-1} possible states whose transition to the next state depends on each M possible received bit, so the final complexity of each step of the BCJR algorithm is $\mathcal{O}(M^L)$, which becomes intractable for the actual communications systems. The memory needed by this algorithm also grows exponentially, because it stores M^{L-1} variables. For all these reasons, in this paper we focus on an approximated solution whose complexity and memory are computationally realizable for the actual communications systems.

In this paper, we propose the EP algorithm as a low-complexity and high-accuracy solution for equalization in SISO systems and channels with memory. This approach has been successfully already applied to MIMO detection [4] and channel decoding [5]. The EP algorithm [6–8] can naturally and efficiently work with continuous distributions by moment matching and it powerfully deals with complex and versatile approximating functions. This novel solution exhibits a performance close to the optimal, as illustrated in the experiments included, with linear complexity similar to the one of the LMMSE. Using EP, we construct a Gaussian approximation to the posterior distribution of the transmitted symbol vector, i.e., $q_{EP}(\mathbf{u}) \approx p(\mathbf{u}|\mathbf{y})$. Iteratively, EP finds $q_{EP}(\mathbf{u})$

This work was partially funded by Spanish government (Ministerio de Economía y Competitividad TEC2012-38800-C03-01/02 and FEDER) and by Comunidad de Madrid in Spain (project 'CASI-CAM-CM', id. S2013/ICE-2845).

that aims to match the first two moments for each dimension (in parallel), whose direct computation from $p(\mathbf{u}|\mathbf{y})$ becomes computationally prohibitive for large N. The computational complexity of the algorithm per iteration is dominated by the inversion of a N-dimensional matrix, i.e., $\mathcal{O}(N^3)$. In addition, EP is a soft-output algorithm that provides a posterior probability estimation for each received symbol, which can be naturally fed to modern channel decoders.

The following notation is used throughout the paper. If $\mathbf{u}$ is a vector, u_i denotes the entry i of the vector $\mathbf{u}$ and $\mathbf{u_{i:j}}$ is a vector with the entries of $\mathbf{u}$ in the range i to j. The operator $\mathrm{diag}(\cdot)$ when applied to a vector, e.g. $\mathrm{diag}(\mathbf{u})$, returns a diagonal matrix with diagonal given by $\mathbf{u}$. To denote a normal distribution of a random variable u with mean μ and variance σ^2 we use the notation $\mathcal{N}\left(u : \mu, \sigma^2\right)$. In case of a random vector $\mathbf{u}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ we use $\mathcal{N}\left(\mathbf{u} : \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$.

2. SYSTEM MODEL AND SOLUTIONS

We consider the discrete-time dispersive communication system depicted in Figure 1. A block of I message bits, $\mathbf{m} = [m_1, ..., m_I]^\top$, is encoded with a rate $R = I/T$ code into $\mathbf{b} = [b_1, ..., b_T]^\top$. An M-ary modulation is considered to obtain $N = \lceil T/\log_2 M \rceil$ symbols, $\mathbf{u}$. Then, the block frame $\mathbf{u} = [u_1, ..., u_N]^\top = \mathcal{R}(\mathbf{u}) + j\mathcal{I}(\mathbf{u})$ is transmitted over the channel, where each component $u_k = \mathcal{R}(u_k) + j\mathcal{I}(u_k) \in \mathcal{A}$. Here $\mathcal{A}$ denotes the set of symbols of the constellation of order $|\mathcal{A}| = M$, hence the alphabet of $\mathbf{u}$ symbols has size $|\mathcal{A}|^N$. The mean symbol energy transmitted is denoted by E_s. The channel is completely specified by the CSI, i.e., $\mathbf{h} = [h_1, ..., h_L]^\top$, where L is the length of the channel impulsive response. The received signal $\mathbf{y} = [y_1, ..., y_{N+L-1}]^\top \in \mathbb{C}$ is given by

$$\begin{bmatrix} y_1 \\ \\ \vdots \\ \\ y_{N+L-1} \end{bmatrix} = \begin{bmatrix} h_1 & & \mathbf{0} \\ \vdots & \ddots & \\ h_L & \ddots & h_1 \\ & \ddots & \vdots \\ \mathbf{0} & & h_L \end{bmatrix} \begin{bmatrix} u_1 \\ \\ \vdots \\ \\ u_N \end{bmatrix} + \begin{bmatrix} w_1 \\ \\ \vdots \\ \\ w_{N+L-1} \end{bmatrix} \tag{2}$$

or more compactly

$$\mathbf{y} = \mathbf{H}\mathbf{u} + \mathbf{w} \tag{3}$$

where $\mathbf{H}$ is a $(N+L-1) \times N$ matrix, the k-th received entry is given by

$$y_k = \sum_{i=1}^{L} h_i u_{k-i+1} + w_k = \mathbf{h}^\top \mathbf{u_{k:k-L+1}} + w_k \tag{4}$$

and $\mathbf{w} \sim \mathcal{N}\left(\mathbf{w} : \mathbf{0}, \sigma_w^2 \mathbf{I}\right)$ is a AWGN vector. In (2) we consider a transmission of N symbols where $u_i = 0\ \forall i \leq 0$ and $\forall i > N$.

Inference is typically presented using real-valued random variables, instead of complex-valued variables used in signal processing for communications. The system model in (3) can be translated into an equivalent double-sized real-valued representation that is obtained by considering the real and imaginary parts separately. Therefore, without loss of generality, in the following we adopt the real-valued channel model.

Given the model above, the posterior probability of the transmitted symbol vector $\mathbf{u}$ has the following expression:

$$p(\mathbf{u}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})} \propto \mathcal{N}\left(\mathbf{y} : \mathbf{H}\mathbf{u}, \sigma_w^2 \mathbf{I}\right) \prod_{k=1}^{N} \mathbb{I}_{u_k \in \mathcal{A}} \tag{5}$$

where $\mathbb{I}_{u_k \in \mathcal{A}}$ is the indicator function that takes value one if $u_k \in \mathcal{A}$ and zero otherwise.

Note that we are using simple equalization (see Figure 1). However, to improve the results, we could iteratively feed the soft detector with the output probabilities of the decoder, as in turbo equalization [9].

2.1. LMMSE algorithm

Given the CSI, the LMMSE equalizer [1] first proceeds by computing

$$\boldsymbol{\mu}_{MMSE} = \left(\mathbf{H}^\top \mathbf{H} + \frac{\sigma_w^2}{E_s} \mathbf{I}\right)^{-1} \mathbf{H}^\top \mathbf{y} \tag{6}$$

and then, it performs a component-wise hard decision by projecting each component of $\boldsymbol{\mu}_{MMSE}$ into the corresponding constellation

$$\hat{u}_{k\ MMSE} = \arg \min_{u_k \in \mathcal{A}} |u_k - \mu_{k\ MMSE}|^2. \tag{7}$$

The complexity of this solution is dominated by the matrix inversion in (6). The posterior approximate provided by the LMMSE algorithm is a Gaussian distribution with mean $\boldsymbol{\mu}_{MMSE}$ and covariance $\boldsymbol{\Sigma}_{MMSE}$

$$q_{MMSE}(\mathbf{u}) = \mathcal{N}\left(\mathbf{u} : \boldsymbol{\mu}_{MMSE}, \boldsymbol{\Sigma}_{MMSE}\right) \tag{8}$$

where

$$\boldsymbol{\Sigma}_{MMSE} = \sigma_w^2 \left(\mathbf{H}^\top \mathbf{H} + \frac{\sigma_w^2}{E_s} \mathbf{I}\right)^{-1}. \tag{9}$$

The symbol probability of each entry is computed by independently deciding on each component

$$q_{MMSE}(u_k = \mathcal{A}_i) \propto \mathcal{N}\left(\mathcal{A}_i : \mu_{k\ MMSE}, \Sigma_{k,k\ MMSE}\right). \tag{10}$$

3. EXPECTATION PROPAGATION

Expectation propagation or EP [6–8, 10] is a technique in Bayesian machine learning for approximating the true posterior distribution with exponential family distributions. It

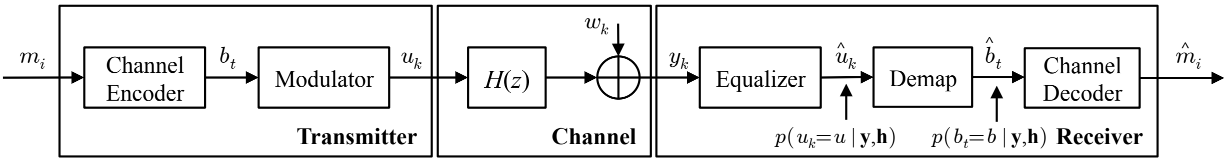


Fig. 1: System model.

is based on the minimization of the Kullback-Leibler divergence.

Suppose we are given some statistical distribution with hidden $\mathbf{x}$ and observables $\mathcal{D}$[1] that factors as follows

$$p(\mathbf{x}|\mathcal{D}) \propto f(\mathbf{x}) \prod_{i}^{\mathcal{I}} t_i(\mathbf{x}), \tag{11}$$

where $f(\mathbf{x})$ belongs to an exponential family $\mathcal{F}$ with sufficient statistics $\mathbf{\Phi}(\mathbf{x})$ and $t_i(\mathbf{x})$ are nonnegative factors that do not belong to the exponential family $\mathcal{F}$. When the true posterior $p(\mathbf{x})$ in (11) is analytically intractable or prohibitively complex, EP provides a feasible approximation to $p(\mathbf{x})$ by an exponential distribution $q(\mathbf{x})$ from $\mathcal{F}$ which factorizes as

$$q(\mathbf{x}) \propto f(\mathbf{x}) \prod_{i} \tilde{t}_i(\mathbf{x}) \tag{12}$$

where each factor $\tilde{t}_i(\mathbf{x}) \in \mathcal{F}$ is an approximation of the factor $t_i(\mathbf{x})$ in the true posterior (11). The approximation $q(\mathbf{x})$ is obtained by minimizing the Kullback-Leibler divergence with respect to $p(\mathbf{x})$, i.e. $q(\mathbf{x}) = \arg\min_{q'(\mathbf{x})\in\mathcal{F}} D_{KL}(p(\mathbf{x})\|q'(\mathbf{x}))$. This solution is equivalent to matching the expected sufficient statistics

$$\mathbb{E}_{q(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})] = \mathbb{E}_{p(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})] \tag{13}$$

where $\mathbb{E}_{q(\mathbf{x})}[\cdot]$ denotes expectation with respect to the distribution $q(\mathbf{x})$. Equation (13) is called *moment matching* condition. If $q(\mathbf{x})$ is a Gaussian distribution $\mathcal{N}(\mathbf{x}:\boldsymbol{\mu},\mathbf{\Sigma})$ then we minimize the Kullback-Leibler divergence by setting the mean $\boldsymbol{\mu}$ of $q(\mathbf{x})$ equal to the mean of $p(\mathbf{x})$ and the covariance $\mathbf{\Sigma}$ equal to the covariance of $p(\mathbf{x})$. However, the computation of the moments $\mathbb{E}_{p(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})]$ to construct $q(\mathbf{x})$ according to them is intractable because we can not infer over $p(\mathbf{x})$. To solve this problem, Minka proposed a sequential EP algorithm to iteratively obtain the solution in (11). The main idea behind the sequential EP algorithm is to do inference over a distribution of the form

$$\tilde{p}_i(\mathbf{x}) \propto \frac{q(\mathbf{x})}{\tilde{t}_i(\mathbf{x})} t_i(\mathbf{x}) \tag{14}$$

and optimize each factor $\tilde{t}_i(\mathbf{x})$ in turn independently in the context of all of the remaining factors. A detailed description of the EP algorithm is given in Algorithm 1 where $q^{(\ell)}(\mathbf{x})$ is the approximation to $q(\mathbf{x})$ in (12) at iteration ℓ.

[1]For simplicity, we omit the dependence on the observed data $\mathcal{D}$ to keep the notation uncluttered in the rest of the paper.

Algorithm 1 The EP algorithm

Initialiaze all the approximating factors $\tilde{t}_i(\mathbf{x})$ and then the approximation $q(\mathbf{x})$ in (12) by setting these factors $\tilde{t}_i(\mathbf{x})$.
repeat
  for $i = 1, ..., \mathcal{I}$ **do**
    1) Compute the *cavity* distribution by removing $\tilde{t}_i(\mathbf{x})$ from the approximated distribution $q(\mathbf{x})$ by division, i.e., $q^{(\ell)\setminus i}(\mathbf{x}) = q^{(\ell)}(\mathbf{x})/\tilde{t}_i^{(\ell)}(\mathbf{x})$.
    2) Compute the distribution $\tilde{p}_i(\mathbf{x}) \propto t_i(\mathbf{x})q^{(\ell)\setminus i}(\mathbf{x})$ and its moments

$$\mathbb{E}_{\tilde{p}_i(\mathbf{x})}[\mathbf{\Phi}(\mathbf{x})] \tag{15}$$

    3) Compute the new refined factor $\tilde{t}_i^{(\ell+1)}(\mathbf{x})$ by setting the moments of the distribution $\tilde{t}_i^{(\ell+1)}(\mathbf{x})q^{(\ell)\setminus i}(\mathbf{x})$, denoted as $\mathbb{E}_{\tilde{t}_i^{(\ell+1)}(\mathbf{x})q^{(\ell)\setminus i}(\mathbf{x})}(\mathbf{x})$, equal to (15).
  end for
until convergence (or stopped criterion)

4. BLOCK-EP EQUALIZER

In this section, we propose as a novel approach using the EP for channel equalization in a SISO system with ISI, naming it block-EP equalizer. We approximate the optimal solution in (5) by replacing each one of the non-Gaussian factors by an unnormalized Gaussian [4]

$$q(\mathbf{u}) \propto \mathcal{N}\left(\mathbf{y}:\mathbf{H}\mathbf{u},\sigma_w^2\mathbf{I}\right) \prod_{k=1}^{N} \exp\left(\gamma_k u_k - \frac{1}{2}\Lambda_k u_k^2\right) \tag{16}$$

where γ_k and $\Lambda_k > 0$ are real constants. For any value $\boldsymbol{\gamma} \in \mathbb{R}^{\mathrm{N}}$ and $\mathbf{\Lambda} \in \mathbb{R}_{+}^{\mathrm{N}}$, $q(\mathbf{u})$ is a Gaussian $\mathcal{N}(\mathbf{u}:\boldsymbol{\mu},\mathbf{\Sigma})$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$

$$\mathbf{\Sigma} = (\sigma_w^{-2}\mathbf{H}^\top\mathbf{H} + \mathrm{diag}(\mathbf{\Lambda}))^{-1} \tag{17}$$

$$\boldsymbol{\mu} = \mathbf{\Sigma}(\sigma_w^{-2}\mathbf{H}^\top\mathbf{y} + \boldsymbol{\gamma}). \tag{18}$$

A detailed implementation of the *block-EP equalizer* (BEP equalizer) is included in Algorithm 2. At this point it is important to remark that in the approximation proposed we retain all the knowledge on the systems by including the first factor in (16) while approximating with EP the unknowns.

Algorithm 2 Block-EP equalizer

Initialize $\gamma_k^{(0)} = 0$ and $\Lambda_k^{(0)} = E_s^{-1}$ for $k = 1, ..., N$.
The pair $(\gamma_k^{(\ell+1)}, \Lambda_k^{(\ell+1)})$ is computed as follows:
for $\ell = 0, ..., P-1$ **do**
for $k = 1, ..., N$ **do**
1) Compute the k-th marginal of the distribution $q^{(\ell)}(\mathbf{u})$ in (16), namely $q_k^{(\ell)}(u_k) = \mathcal{N}\left(u_k : \mu_k^{(\ell)}, \sigma_k^{2(\ell)}\right)$.
2) Compute the $cavity$ marginal

$$q^{(\ell)\backslash k}(u_k) = \frac{q_k^{(\ell)}(u_k)}{\exp\left(\gamma_k^{(\ell)} u_k - \frac{1}{2}\Lambda_k^{(\ell)} u_k^2\right)} \sim \sim \mathcal{N}\left(u_k : t_k^{(\ell)}, h_k^{2(\ell)}\right) \quad (19)$$

where

$$h_k^{2(\ell)} = \frac{\sigma_k^{2(\ell)}}{1 - \sigma_k^{2(\ell)}\Lambda_k^{(\ell)}}, \quad t_k^{(\ell)} = h_k^{2(\ell)}\left(\frac{\mu_k^{(\ell)}}{\sigma_k^{2(\ell)}} - \gamma_k^{(\ell)}\right)$$

3) Compute the mean $\mu_{p_k}^{(\ell)}$ and variance $\sigma_{p_k}^{2(\ell)}$ of the distribution $\hat{p}^{(\ell)}(u_k) \propto q^{(\ell)\backslash k}(u_k)\mathbb{I}_{u_k \in \mathcal{A}}$.
4) Finally, the pair $(\gamma_k^{(\ell+1)}, \Lambda_k^{(\ell+1)})$ is updated so that the following unnormalized Gaussian distribution

$$q^{(\ell)\backslash k}(u_k) \exp\left(\gamma_k^{(\ell+1)} u_k - \frac{1}{2}\Lambda_k^{(\ell+1)} u_k^2\right) \quad (20)$$

has mean and variance equal to $\mu_{p_k}^{(\ell)}$ and $\sigma_{p_k}^{2(\ell)}$. The solution is given by

$$\Lambda_k^{(\ell+1)} = \beta\left(\frac{1}{\sigma_{p_k}^{2(\ell)}} - \frac{1}{h_k^{2(\ell)}}\right) + (1-\beta)\Lambda_k^{(\ell)} \quad (21)$$

$$\gamma_k^{(\ell+1)} = \beta\left(\frac{\mu_{p_k}^{(\ell)}}{\sigma_{p_k}^{2(\ell)}} - \frac{t_k^{(\ell)}}{h_k^{2(\ell)}}\right) + (1-\beta)\gamma_k^{(\ell)} \quad (22)$$

end for
end for
Obtain the Gaussian approximation after EP algorithm, $q(\mathbf{u}) \propto \mathcal{N}(\mathbf{u} : \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given by (18) and (17), respectively.
Compute the hard output and its symbol probability as

$$\hat{u}_k = \arg\min_{u_k \in \mathcal{A}} |u_k - \mu_k|^2 \quad (23)$$

$$q(u_k = \mathcal{A}_i) \propto \mathcal{N}(\mathcal{A}_i : \mu_k, \Sigma_{k,k}) \quad (24)$$

Eqn. (21) and (22) are proposed following the guidelines in [4, Eq. 35-36]. The parameter update in (21) may return a negative value $\Lambda_k^{(\ell+1)}$ for some k's which means that there is no pair $(\Lambda_k^{(\ell+1)}, \gamma_k^{(\ell+1)})$ that sets the variance of the Gaussian in (20) at $\sigma_{p_k}^{2(\ell)}$. For that k's, we keep the previous values for these parameters. Note that all $(\gamma_k^{(\ell+1)}, \Lambda_k^{(\ell+1)})$ pairs for $k = 1, ..., N$ can be updated in parallel and we only require the computation of a N-dimensional inverse matrix in (17) for each ℓ-iteration (typically around 10 [4]), so complexity of EP is dominated by the size of that inverse, i.e., $\mathcal{O}(N^3)$. We introduce a smoothing parameter $\beta \in [0, 1]$ and a small constant ϵ that sets a minimum variance $\sigma_{p_k}^{2(\ell)} = \max(\epsilon, \sigma_{\hat{p}_k}^{2(\ell)})$ allowed per component to avoid numerical instabilities.

5. SIMULATION RESULTS

In this section, we illustrate the good performance of the BEP equalizer for channels with memory. We have set $\beta = 0.3$, $\epsilon = 1e^{-4}$ (for hard decisions), $\epsilon = 0.5$ (for soft decisions) and $P = 10$ iterations in the EP algorithm. In all the experiments presented in this section we consider block frames of 500 random bits encoded with a regular LDPC of rate 1/2 and we average the BER over 1000 different frames and 100 realizations of channels. Each tap is Gaussian distributed, and the whole channel response is normalized.

We first consider channels of 5 taps and 4-PAM modulation. In Figure 2, we depict the BER curves before (solid lines) and after (dashed lines) the LDPC decoder for BEP, LMMSE and BCJR equalization. Compared to the BCJR solution before the decoder, we are far about 3 dB for BER=10^{-3} and compared with the LMMSE method, BEP is able to improve the performance in 5 dB. After the decoder, we are less than 3 dB far from optimal solution for BER=10^{-3} and EP outperforms LMMSE in 3 dB for the same BER. A similar study is presented in Figure 3(a) for channels of 6 taps and 16-PAM modulation, excluding the BCJR solution, which we do not simulate due to its unaffordable computational complexity. For BER=10^{-3} BEP equalization outperforms LMMSE in 5 dB before the decoder and 4 dB after the decoder. In Figure 3(b) we illustrate the same constellation than in (a), but now increasing the number of taps to 15. Even with this high memory, EP exhibits an excellent performance. Specifically, we obtain a gain of 4 dB before the decoder for BER=10^{-3} and 2 dB after the decoder, with respect to LMMSE.

Finally, a computational complexity[2] analysis between BEP and BCJR is given in Table 1. LMMSE algorithm is not included because it only differs in a factor P compared with the BEP algorithm. When both L and M are not large, as in Figure 2, the complexity of the BCJR algorithm is not high and it can be computed. However, when L and M are increased, as in Figure 3, its complexity grows exponentially and becomes intractable while BEP remains unchanged with L or M.

[2]The complexity of the BCJR algorithm is $\mathcal{O}(M^L N)$ while BEP is $\mathcal{O}(PN^3)$.

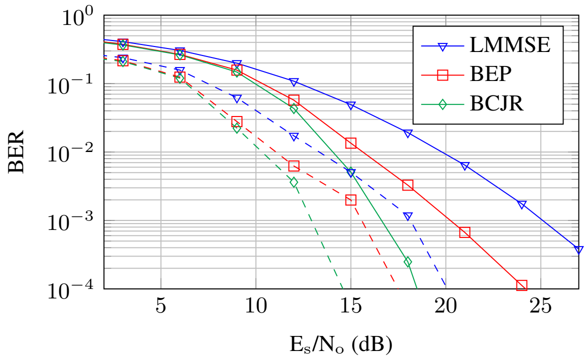
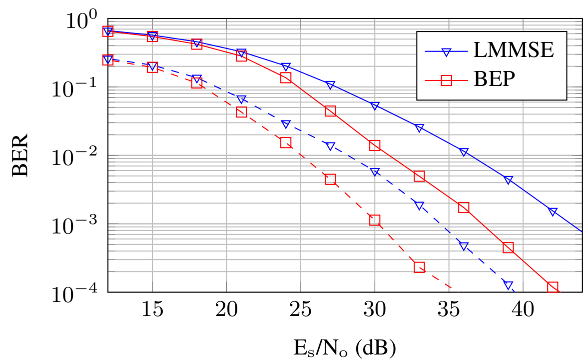
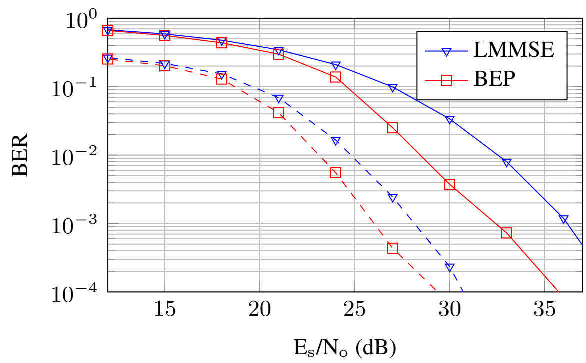


Fig. 2: BER for LMMSE, BEP and BCJR equalizers for channels with 5 taps and 4-PAM modulation.



(a) Channels with 6 taps



(b) Channels with 15 taps

Fig. 3: BER for LMMSE and BEP equalizers for 16-PAM.

Figure	M	L	N	Complexity BCJR	Complexity BEP	Reduct. factor
Fig. 2	4	5	500	$512e^3$	$125e^7$	$4e^{-4}$
Fig. 3(a)	16	6	250	$419e^7$	$156e^6$	27
Fig. 3(b)	16	15	250	$288e^{18}$	$156e^6$	$184e^{10}$

Table 1: Complexity comparison between algorithms.

6. CONCLUSION AND FUTURE WORK

The design of efficient equalizers is a challenging open problem. In this paper, we focus not only on symbol estimation but also the posterior probability estimation for each received symbol since the LDPC decoder needs a high quality APP to perform optimally. The optimal solution is intractable for the actual communications systems whenever we have large channel memory and/or large constellations. Classical methods such as linear MMSE can be used at the cost of a poorer performance. The novel BEP equalizer proposed in this paper is a soft-output algorithm that solves this problem, constructing tractable approximation to a given probability distribution. We have shown through simulations that the BEP equalizer quite outperforms the LMMSE, even with a high number of taps. Since it exhibits a similar structure, its computational burden and memory needs are similar to those of the LMMSE. Further improvements on the reduction of its computational complexity, i.e. of the covariance matrix inversion, remains as a future line of research.

REFERENCES

[1] John G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, 5 edition, 2008.

[2] Luis Salamanca, Juan José Murillo-Fuentes, and Fernando Pérez-Cruz, "Bayesian equalization for LDPC channel decoding," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2672–2676, May 2012.

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

[4] J. Céspedes, P.M. Olmos, M. Sánchez-Fernández, and F. Perez-Cruz, "Expectation propagation detection for high-order high-dimensional mimo systems," *IEEE Transactions on Communications*, vol. 62, no. 8, pp. 2840–2849, Aug 2014.

[5] P.M. Olmos, J.J. Murillo-Fuentes, and F. Perez-Cruz, "Tree-structure expectation propagation for LDPC decoding over the BEC," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3354–3377, 2013.

[6] Thomas P. Minka, *A family of algorithms for approximate Bayesian Inference*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[7] T. Minka, "Expectation propagation for approximate bayesian inference," in *UAI*, 2001, pp. 362–369.

[8] M.W. Seeger, "Expectation propagation for exponential families," *Univ. Calif., Berkeley, CA, USA, Tech. Rep.*, 2005.

[9] M. Tuchler and A.C. Singer, "Turbo equalization: An overview," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 920–952, Feb 2011.

[10] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, New York, Secaucus, NJ, USA, 2006.