Implementation of Soft Decision Low-Density Parity-Check Decoder for Bahl-Cocke-Jelinek-Raviv Algorithm

Manthena Varma[1], J. S. S. Ramaraju[2], Sappa Srinivas[3]

[1]*Student, M-tech, ECE, Bhimavaram Institute of Engineering &Technology, Bhimavaram, AP India.*
[2]*Assistant professor, ECE, Bhimavaram Institute of Engineering &Technology, Bhimavaram, AP India.*
[3]*Student, M-tech, ECE, Bhimavaram Institute of Engineering &Technology, Bhimavaram, AP, India.*

ABSTRACT

In this paper we focus on the BCJR algorithm, and its prior estimate of the channel state information (CSI), In case of uncertainties during the estimation, overconfident posterior probability tends to mislead the performance of soft decoders. Our approach takes into consideration not only the uncertainty due to the noise in the channel, but also the uncertainty in the CSI estimate. Thus, we resort to a Bayesian approach for the computation of the APP. Hence, we also put forward an approximation for each symbol's posterior, using the expectation propagation algorithm, which is optimal from the Kullback- Leibler divergence. View pointed yields an equalization with a complexity identical to the BCJR algorithm. This algorithm has the same complexity as the BCJR, exhibiting lower bit error rate at the out of the channel decoder than the standard BCJR that considers maximum likelihood (ML) to estimate the CSI. We also use a graphical model representation of the full posteriori which the proposed approximation can be readily understood. This proposed method exhibits a much better performance compared to the ML-BCJR when a LDPC decoder, which needs the exact posterior for each symbol to detect the incoming word and it is sensitive to a mismatch in those posterior estimates, for example, for QPSK modulation and a channel with three taps, we can expect gains over 0.5db with same computational complexity as the ML receiver.

Keywords**:** *IR filters, expectation propagation, LDPC coding, ML-BCJR equalization, QPSK modulator, fading Channels.*

I. INTRODUCTION

Communication channels can be characterized by a linear finite impulsive response that either represents the dispersive nature of a physical media or the multiple paths of wireless communications [1]. This representation causes inter-symbol interference (ISI) at the receiver end can impair the digital communication. The channel state information(CSI) is typically using pilots (a preamble) and a maximum likelihood (Ml) estimator. These preamble are typically start to reduce the transmission of non informative symbols, yielding in accurate CSI estimates. In the following we will refer to the BCJR equalizer with ML estimation of the channel as ML-BCJR. The ML-BCJR is the approximation to the app for each symbol because it does not include the uncertainty in the estimate. In accuracies in the APP estimates degrade the performance of modern channel LDPC decoders. Such as turo or low density parity checks codes[2][3].The channel decoder may fail to the correct transmitted codeword or may even fail to converge at all.

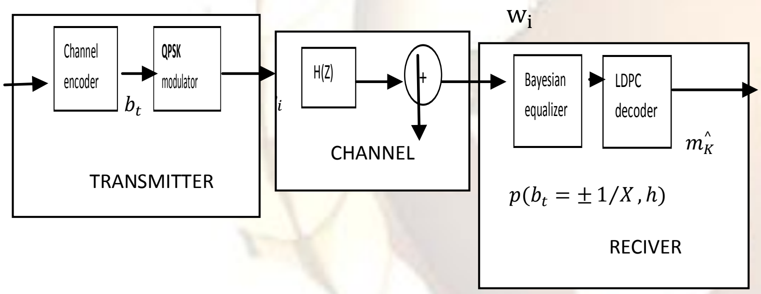
In a previous work, we have show that accurate posterior probability estimates increase the performance of LDPC decoder[4],although that work focuses on nonlinear channel estimation. In the framework of turbo-receivers[5], some approaches can be found in the literature that incorporate there uncertainties in the iterative process of equalization and decoding. We find a proposal to estimate some parameters in a OFDM system to later include them in the decoding. For nonlinear channels we have shown in[4] that accurate APP estimates increase the performance of LDPC decoders. We propose a simple yet accurate approximation to the Bayesian solution that allows to recovers the Markova property. In this novels solution, the Bayesian framework is embedded into the BCJR algorithm. The difference between the bit error rate (BER) of the ML-BCJR and the Bayesian BCJR equalizer is not the significant, although it slightly favors the BE. The advent of turbo processing, some Bayesian approaches have been proposed to embed and consider the uncertainties in the whole iterative process of equalization and decoding. We propose a Bayesian equalizer(BE),which integrates the uncertainty in the CSI to produce more accurate a posterior probability estimates. Hence, our approach given a direct estimation without iterating between the APP estimation and LDPC decoder, as we do not only provide a point estimate but a probability estimate.

We experimentally show that at the output of the LDPC decoder the Bayesian BCJR equalizer considerably improves the performance of the ML-BCJR equalizer, When we measure the probability of error. These gains are more significant for high signal to noise rations channel with long impulsive

responses and/or short training sequences, we consider the performance of both LDPC codes [3][6]and BER-optimal BCJR algorithm as equalizer[7]. Other techniques ensure
near Shannon limit results. Furthermore, modern machine learning techniques and inference in graphs afford the lack of hug computational complexity of these algorithms yielding to efficient and near-optimal algorithms to equalize [11] and decode [12], [13]. The LDPC decoder very much benefits from this approach, exhibiting gains of 1dB with respect to the ML-BCJR solution. The proposes equalizer does not have an analytical description and cannot be computes in linear-time in the number of symbols as the BCJR algorithm, thus we also propose an approximation to the Bayesian solution. , but its able to retain most of the gain of the full Bayesian approach.

The paper is organized as fallows. In second term we describe the structure of the general communication system proposed in third term. The LDPC decoder presented in the fourth term. Thus, experimental results of fifth term shows the performance of our method. Finally, in sixth term is the results obtained are summarized and future work about our proposal is presented.

II. BCJR ALGORITHM



Block Diagram

2.1. FUNCTION MODEL.

We consider the discrete-time channel communication system in fig 1. The channel H(z) is completely specified bye the channel state information(CSI). i.e $h = [h_0 h_1, \dots, h_L]^T$, Where L is the length of the channel. The 'h' is the model value of the channel. As independent, unit-variance Gaussians(UVG) and zero mean (Rayleigh fading).

The transmitted block of K message bits, $m = [m_0 m_1, \dots, m_k]^T$ is encoded with a rate R= K/N and to obtain the codeword b =[b1, b2,….., bN] that is transmitted over the channel by using the modulator block. In that we are using the QPSk modulator.

$$x_i = b_i^T h + w_i, \qquad (1)$$

Where $b_i^T = [b_{i,} b_{i-1}, \dots, b_{i-L+1}]$, $h = [h_0, h_1, \dots, h_L]$

and w_i is the additive white Gaussian noise(AWGN) with variance σ_n^2. Thus, the received sequence is $X = [x_1, x_2, \dots, x_N]$.

Startlingly We transmit a preamble with η is the bits, $D = \{x_i^* b_i^*\}_{i=1}^{n}$ that are be used to estimate the unknown channel at the receiver, then we transmit the codeword b. The maximum likelihood is considerable for the task os estimation.

$$h_{ML}^{\wedge} = \arg\max p\left(\frac{X^*}{b^*}, h\right) \qquad (2)$$

The channel coefficients are estimation with the preamble. And We apply the BCJR algorithm for to obtain the a posterior probability (APP) estimation for the each transmitted symbols.

$$P(b_i = b/x, h_{ML}^{\wedge}) \quad i = 1\dots..N \qquad (3)$$

Finally We decode the received word by using the LDPC decoder to a maximum a posterior (MPA) estimates for M.

2.2.ML-BCJR Algorithm

The ML BCJR algorithm $P\left(b_i = \frac{b}{X}, h_{ML}^{\wedge}\right)$. It is the bais of the praction implementation of CSI.Hence, the CSI, the BCJR consider the a posterior probability (APP) estimates. These probability are computed as[8]

$$P(b_i = b/x, h_{ML}^{\wedge}) = \sum_{(a,b)s_b} \frac{P(s_i = a, s_{i+1} = b, X/h_{ML}^{\wedge})}{P(X/h_{ML}^{\wedge})} \qquad (4)$$

Where s_i and s_{i+1} is the states in the equivalent trellis at time I and i+1, and s_b is the set of all possible transition from $s_i = a$ to $s_{i+1} = b$ by the input $b_i = b$. The numerator in[9] can be expressed as:

$$P(s_i = a, s_{i+1} = b, X_1^{i-1}, x_i\, X_{i+1}^N / h) = P(s_i = a, X_1^{i-1}/h)$$
$$P(s_{i+1} = b, x_i / s_i = a, h_{ML}^{\wedge})$$
$$P(X_{i+1}^N / s_i = b, h_{ML}^{\wedge})$$

Where the vector X is divided in three different sets:
Before instant of received samples are i, X_1^{i+1}
After instant of received samples are i, X_{1+1}^N; and the received sample at instant i, x_i.
In the second term, X_1^{i+1} does not provide any information gives $s_i = a$. And third term all the information about the receiver samples and the evolution in the trellis is contained in the last state $s_{i+1} = b$.

The BCJR algorithm are the probabilities of the three terms that the computes its forward and backward recursions.

I term $P(s_i = a, X_1^{i-1}/h_{Ml}^{\wedge}) = \alpha_i$(a) (6)
II term $P(s_{i+1} = b, x_i / s_i = a, h_{ML}^{\wedge}) = \gamma_i$(a,b) (7)
(7)
III term $P(X_{i+1}^N / s_{i+1} = b, h_{ML}^{\wedge}) = \beta_{i+1}$(b) (8)

Where by Markovity some variables are extracted from both probabilities. In the first term, the transition from $s_i = a$ *to* $s_{i+1} = b$ can be rewritten in terms of the set b_i of transmitted bits, and assuming a channel with additive white Gaussian noise (AWGN) it follows that:

$P(x_i/s_{i+1} = b, x_i/s_i = a, \hat{h}_{ML}) = P(x_i/b_i, \hat{h}_{ML}) \sim \tilde{N}(b_i^T h, \sigma_w^2 I)$ (9)

The probabilities [6] and [8] are computed at each stage through the forward recursion

$\alpha_{i+1}(b) = \sum_{a=0}^{Q-1} \gamma_i(a,b)\alpha_i(a)$ (10)

$= P(s_{i+1} = b, X_1^i/\hat{h}_{ML})$

And backward recursion

$\beta_i(a) = \sum_{b=0}^{Q-1} \gamma_i(a,b)\beta_{i+1}(b)$ (11)

$= P(X_i^N/s_i = a, \hat{h}_{ML})$

Where Q are the states a(or b) at time i(or i+1) that, for any value of b_i, time i+1 (or i) in state b(or a) We assume both recursion start from know states.

III. BAYESIAN EQUALIZATION

3.1 ML-BCJR EQUALIZATION

We consider the BCJR, assuming a ML estimation, The half of the time gives overconfident predictions. This criterion does not assume the uncertainties in the CSI. If the training sequence is long enough this might be the case, but it does not need to be in most case of interest, where we need to keep this training sequence as short as possible.

We compute the posterior possibility as: $P(b_i = b/X, D) = \int p(b_i = b/X, h) p(h/D) dh$ (12)

Where $p(b_i = b/X, h)$ is the APP computed by the BCJR algorithm for a given h and p(h/D)

Is the CSI posterior.

$$P\left(\frac{h}{D}\right) = \frac{p(h)\pi_{i=1}^{n} p(\frac{x_i^*}{b_i^*})}{p(x_n^*, \ldots, x_n^*/b_1^* b_2^*, \ldots, b_n^*)} \quad (13)$$

Is the posterior probability for CSI, given the Gaussian noise and Rayleigh fading.

The result of the BCJR algorithm assuming a ML estimation is quit similar to the

$P(b_i = b/X, D) = \int p(b_i = b/X, h) p(h/D) dh$. In that uncertainty in the CSI, the performance of the ML-BCJR is misled due to in accuracies in the estimation. The BE in [10] considers the information of both variance and mean of the posterior of h, and the uncertainty in the estimation of the CSI, for the improving much accurate APP.

A. COMPUTATION OF THE SOLUTION

To compute $P(b_i = b/X, D)$ we consider the following steps

- Measure the posterior of the channel:

$$P\left(\frac{h}{D}\right) = \frac{p(h)p(\frac{x_i^*}{b_i^*}, h)}{p(x^*/b^*)} \quad (14)$$

The numerator is the product of the likelihood $p(\frac{x_i^*}{b_i^*}, h)$ and the prior of h. In the proposed system, both are Gaussians distributed as

$P(X^*/b^*, h) \sim \tilde{N}((b^*)^T h, \sigma_4^2, I)$ (15)

$P(h) \sim \tilde{N}(0, C_h)$

The numerator in[5] is the product of complex valued Gaussian that leads to a Gaussian posterior:

Whose mean and covariance matrix are

$h_h/D = (C_h^{-1} + b^*(b^*)^H \sigma_w^{-2})\, b^* X^* \sigma_w^{-2}$

$C_{h/D} = (C_h^{-1} + b^*(b^*)^H \sigma_w^{-2})$

But we can interchange the integral by the marginalization respect to b/b_i in the BCJR algorithm. As follows

$P(b/X, D) = \frac{1}{2}\sum_{b/b_i} \int p(X/b, h)\; p(b)\, p(h/D) dh$ (16)

The posterior consider the produced random samples:

The vector of mean and covariance matrix, We exactly sample to obtain G random samples.

B. CALCULATE THE BCJR ALGORITHM

The transmitted of the each bit from the posterior probability for the G different samples of P(h/D)[7].

The G is the different values of $p(b_i = b/x_1, \ldots, x_N, h)$ average the posterior probability of each transmitted bit over all possible cases of h;

$P(b_i/X, D) = \frac{1}{G}\sum_{j=1}^{G} p(b_i/X, h_J)$ (17)

This solution is time demanding, because we have to measure the G time BCJR algorithm. The resulting algorithm The ABE presents the same complexity of the ML-BCJR and it is able to incorporate the uncertainties in the CSI estimation

3.2 APPROXIMATE BAYESIAN EQUALIZER:

The ML-BCJR algorithm as an approximation to the BE, in which the Gaussian density in [7] is replaced by

$P(b/X, D) \sim \tilde{N}(b^T h_{ML}, \sigma_w^2 I)$ (18)

and since the covariance matrix is diagonal the forward-backward recursion can be used to compute $p(b_i/X, D)$ in linear time.

$P(b/X, D) \sim \tilde{N}(b^T h_{ML}, \Sigma)$

The ABE does not consider statistical dependence and assumes for each local computation of the forward-backward algorithm that p(h|D) is independent for each received symbol.

The corresponding graphical model is included in Fig II.

It exhibits hidden Markov model structure equivalent to the BCJR equalizer. In this graphical model, we find an inter pretention for [19], in which we assume that the channel changes in each transmission according to the CSI posterior, instead of using the same realization for all the symbols.

In Fig.[7] , we have plotted the inverse of the covariance matrix in[19] for 1000

BPSK symbols transmitted through a channel with L=6, N=15 training symbols, and an Eb/N0 6 dB. We plot the inverse covariance, because its zero off-diagonal terms represent conditional independent components in a Markov random field.

In this figure, we notice that the main diagonal dominates the inverse cross-covariance terms. Those off-diagonal terms present similar values and decay equally fast towards zero as the training sequence increases.

The proposed ABE is a better approximation to the BE than the ML-BCJR equalizer in two ways. First, it uses the posterior mean instead of the ML estimate. Second, the variance for each sample has two components: one due to the noise (the only one considered by the ML-BCJR equalizer) and the other due to the CSI estimation error.

The approximation loses the correlation between the symbols, but these correlations are not so significant in the SNR ranges of interest and they disappear as the training sequence increases. If we were to improve the ABE, there are two natural extensions. Although none of them should provide a significant gain and we have-not explored them further, because the inverse covariance matrix (see Fig. 3) is almost diagonal. We can either use a low-rank approximation for the inverse covariance matrix or a tree-structure for the EP approximation.

The low-rank approximation will concentrate most of its energy in the main diagonal, as the ABE does, and the remaining Eigen values will add little to the approximation unless a significant proportion are added increasing the complexity substantially. The Tree-Enwall also suffer from the same limitation and the additional complexity will not significantly improve the performance.

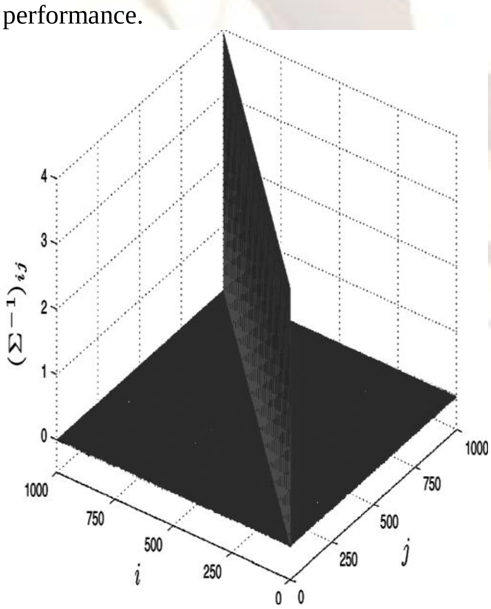


Fig. We plot in the inverse covariance matrix in (14) for 1000 BPSK symbols

Transmitted through a channel with 6 taps. We have used 15 training symbols and Eb/N0= 6 dB.

IV. LDPC DECODER

An LDPC code is a liner block code and through has a parity-check matrix. What distinguishes an LDPC code from conventional linear codes is that parity check matrix which is sparse, i.e the number of non zero entries is much smaller than the total number of entries can be found for it. LDPC codes can be extended to GF(q) by considering a set of non zero weights $w_{i,j}\epsilon$ GF(q) for the edges of G. the parity-check matrix in this case is formed by the set of weights. In other words $h_{i,j} = w_{i,j}$. In the remainder of this thesis, we assume that the codes are binary unless otherwise stated.

LDPC codes, decreases on their structure, can be classified as being regular codes have variable nodes of a fixed degree and check nodes of a fixed degree. Promoting the variable node degree of a regular code by d_v *and the check node degree by* d_c.

It follows that

$$E=r.d_c = n.d_v.$$

Therefore the code rate R can be computed as

$$R: =\frac{x}{n} \geq \frac{n-r}{n} = 1 - \frac{d_v}{d_c}$$

If the rows of H are linearly independent, R=1 - $\frac{d_v}{d_c}$

The quantity $\frac{n-r}{n}$ is referred to as the design rate(), but usually possible linear dependencies among the rows of H are ignored and the design rate and the actual rate are assumed to be equal. Now consider the ensemble of regular LDPC codes with variable degree d_v, check degree d_c and length η. If n is large enough, the average behavior of almost all instances of this ensemble concentrates around the expected behaviors().

Given the degree distribution of an LDPC code its number of edges E, its is easy to see that the number of variables nodes η is,

$$\eta = E \sum_i \frac{d_i}{i} = E \int_0^1 \lambda(x)dx$$

The number of check nodes r is,

$$r = E \sum_i \frac{p_i}{i} = E \int_0^1 \lambda(x)dx$$

Therefore the design rate of the code will be

$$R = 1 - \frac{\sum_i \frac{p_i}{i}}{\sum_i \frac{d_i}{i}}$$

Or equivalently

$$R = 1 - \frac{\int_0^1 p(x)dx}{\int_0^1 \lambda(x)dx}$$

We try to formulate the performance of the code family in terms of its degree distribution in an easy from to allow for maximum flexibility in the design stage, and at the same time we avoid too much simplification to keep our predicted results does to the actual results.

V. SIMULATION

5.1 SIMULATION RESULTS:

We have use the Monte carlo to obtained the results of the [7], Thus we proposed Bayesian ML-BCJR equalization. To improve the performance of the APP technique, We compare of the bit error rate curve with the ones of the ML-BCJR, before and after the LDPC decoder.

In all the experiments we consider the following scenario.

- The data sequence of 500 random bits are encoded with a general LDPC code (3,6) of rate ½.
- We set a halt condition of 100 wrong decode bits to avoid unnecessary computations.
- Up to 10^6 frames of 1000 bits are transmitted over the channel.
- Every frame of test bits, and its associated training sequence, is sent over the same channel, whose taps are Rayleigh distributed. We consider that the channel coherence time is greater than the duration of the frame, i.e the channel does not charge during this time. Further more, in our experiments we take the same value for the taps of the channel during all transmitted frames.

5.2 Performance After Equalization and Decoding:

The BER curves before and after the LDPC decoder for the BE, the ABE and ML-BCJR equalizer shown in fig [7]. The codeword is BPSK modulated and the symbols are transmitted through a^ taps channel. n=15 and n=40 is the length of the training sequence.

The difference between BER curves before the LDPC decoder is negligible it will be observed in fig , since at this point we can only measure the app estimate of the 50%. We obtain a significant gain, when we measure the BER after the LDPC decoder, because the LDPC decoder benefits from accurate APP estimates to correctly decode the transmitted codeword, i.e the BP uses the APP for each individual bit.

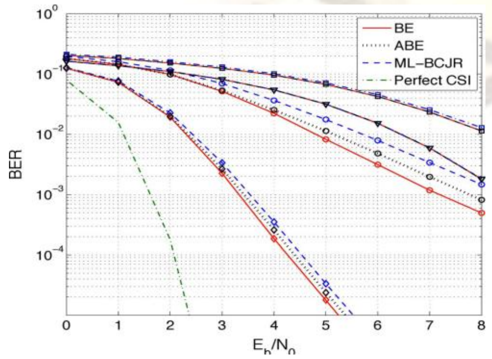


Fig. BER performance for BE (solid lines), the ABE (dotted lines) animal-BCJR equalizer (dashed lines), for a channel with 6 taps before the decoder with n= 15, before the decoder with n=40, after the decoder within=15, and after the decoder with =40.

We consider in section-2 monte carlo sampling to obtain the APP estimates of the Bayesian equalizer (4). In section-3 the approximation Bayesian equalizer (ABE) it is an approximation to the exact result of ML-BCJR are the same complexity.

5.3 Results for Different Modulations and Lengths of the Channel:

In fig. , we compare the performance in terms of WER for the ABE versus the ML-BCJR equalizer, for two different QPSK modulations, different lengths of the training sequence and a channel with L=3.The WER curve with a perfect knowledge of the CSI is also included to set a lower bound of performance for the system. For shorter training sequences, we need higher SNR to compensate for poorer channel estimations. In this scenario the Bayesian equalizer exhibits an improvement compared to the ML-BCJR. In Fig. (a)[7], we have a gain close to 0.3 dB at anEb/N0= 7 dB for n= 10 training samples. This gain tends to cancel as we increase the number of training samples, i.e., by reducing the net throughput. Compared to Fig. with L= 6 taps, we observe that, for L= 3 taps, we have a better performance and less room for improvement, given a training sequence length. A higher modulation order translates into more states in the forward and backward recursions. Therefore, in case of uncertainties in the estimated CSI, the number of inaccurate operations grows and we can expect higher degradation of the equalizer performance, which finally yields into more inaccurate APP estimations. Thus, if we increase the order of the modulation, we can expect a greater gain for the proposed Bayesian equalizer. To illustrate this point, in Fig. 5(b), we include the WER curves for the ABE and the ML-BCJR equalizer after the LDPC decoder, assuming a 16-QAM modulation, L=3 and different lengths of the training sequence. We can observe in Fig. 5(b)[7] a gain over 0.5dB for an Eb/N0= 9 dB and n=10. In all the curves, the gain of the Bayesian equalizer increases compared to the previous results for a QPSK modulation.

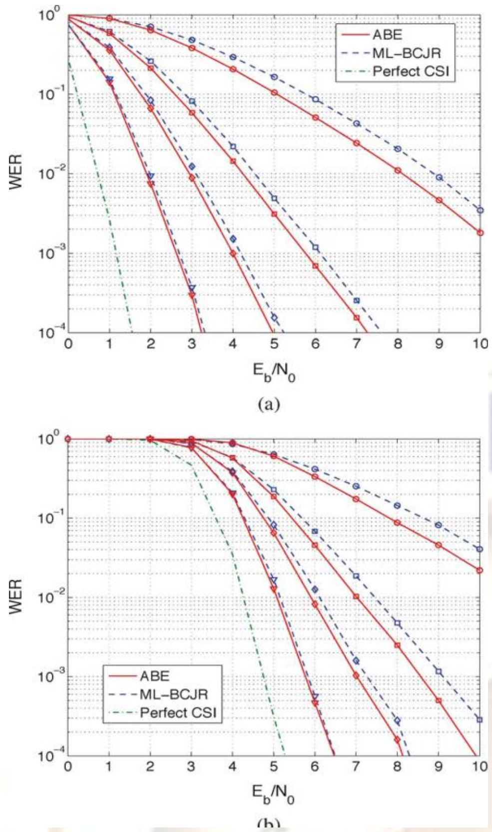


Fig. WER performance for the ABE (solid lines) and ML-BCJR equalizer(dashed lines) after the LDPC decoder, for a channel with 3 taps, QPSK in (a)and 16-QAM in (b) modulation and different lengths of the training sequence= 5,n=10,n=15 and n= 25. In dashed–dotted line, the BCJR with perfect CSI.

VI. CONCLUSIONS AND FUTURE WORK

The ML-BCJR equalizer, that use the ML estimates of the channel state information(CSI), provide the same number of over and under confident posterior probability of each transmitted symbol. specifically when the channel is hard to estimate, these overconfident prediction can assign a value near to one to a wrong estimated bit, which degrades the performance of the decoder due to these bits are harder to flip. The generative model introduced in this paper, where the posterior probability density function of the estimated CSI is included, is a more principled solution. If we are to just estimate the encoded transmitted symbols, the discriminative model is a good choice. However, if the estimation of the APP is needed, i.e., the decoder very much benefits from this information, the discriminative solution exhibits poor results whenever the CSI is badly estimated. On the contrary, the Bayesian approach exploits the full statistical model to provide better APP estimates. We show in the experimental section that these estimations are useful if a LDPC encoding is used. Other soft-decoders may take advantage of this solution as well. We measure the quality of the APP estimates using an LDPC decoder, the standard channel codes in today's communications systems, since the LDPC decoder needs the exact APP to perform.

optimally. We also propose an approximate Bayesian equalizer that can keep most of the gain of the Bayesian equalizer at the same computational cost as the ML-BCJR equalizer. This gain is remarkable in scenarios with short training sequences, long channels and multilevel constellations. We have illustrated these results for LDPC codes and they can be carried over to other coding schemes that need accurate APP ,such as turbo codes

REFERENCES

[1] J. G. Proakis, Digital Communications, 5th ed. New York, NY: McGraw-Hill, 2008.

[2] T. Richardson and R. Urbanke, Modern Coding Theory.Cambridge University Press, March 2008.

[3] S. Chung, D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," IEEE Commun.Lett., vol. 5, no. 2, pp. 58–60, Feb. 2001.

[4] P. M. Olmos, J. Murillo-Fuentes, and F. Perez-Cruz, "Joint nonlinear channel equalization and soft LDPC decoding with Gaussian processes," IEEE Trans. Signal Process., vol. 58, no. 3, pp. 1183–1192, Mar 2010.

[5] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glevieux, "Iterative correction of intersymbol interference: Turboequalization," European Trans. on Telecomm., vol. 6, no. 5, pp. 507–511, Sept.-Oct. 1995.

[6] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Efficient erasure correcting codes," IEEE Trans. on Inform. Theory, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[7] Luis Salamanca,Tuan jose murillo-fuentes,and Fernando perez-cruz"Bayesian Equalization for LDPC Channel Decoding," IEEE Trans. Of signal processing, vol. 60, no. 5, may. 2012.

[8] T. K. Moon, Error Correction Coding: mathematical methods and algorithms, ser. Information Theory. John Wiley & Sons, 2005.

[9] T. M. Cover and J. A. Thomas, Elements of Information Theory. New Jersey, USA: John Wiley & Sons, 2006.

[10] T. Richardson and R. Urbanke, Modern Coding Theory. Cambridge University Press, March 2008.

[11] F. R. Kschischang, B. I. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," IEEE Trans. on Inform. Theory, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[12] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Improved low-density parity-check codes using irregular graphs and belief propagation," IEEE Trans. on Inform. Theory, vol. 47, no. 2 pp. 585–598, Feb. 2001.
[13] T. Richardson and R. Urbanke, "Efficient encoding of low-density paritycheck codes," IEEE Trans. on Inform. Theory, vol. 47, no. 2, pp. 638– 656, Feb. 2001.
[14] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[15] X. Wang and R. Chen, "Blind Turbo equalization in Gaussian and impulsive noise," IEEE Trans. on Vehicular Technology, vol. 50, no. 4, pp. 1092–1105, Jul. 2001.