SOFT-INPUT SOFT-OUTPUT LINEAR PROGRAMMING DECODING

Erica L. Daly

The University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering
1406 W. Green St. Urbana, IL 61801
edaly@illinois.edu

ABSTRACT

This paper presents the soft-input soft-output (SISO) linear programming (LP) decoder. It is shown that the soft information gleaned from a pseudo-codeword solution to the LP optimization is not only useful, but that it can be superior to the soft information output from a SISO belief propagation (BP) decoder in certain situations.

Index Terms— LP decoding, turbo-equalization, LDPC, linear programming, short block length codes

1. INTRODUCTION

Efficiently and accurately communicating short packets of data through a frequency selective channel is a challenge. In the large block length regime, capacity may be approached by protecting the data with a LDPC code and receiving it with a turbo equalizer that employs a MMSE equalizer and a belief propagation (BP) decoder [1]. However, in the short block length regime, the performance of the BP decoder is difficult to ensure analytically due to its sensitivity to short cycles [2].

Linear programming (LP) decoders have been proposed as an alternative to BP decoders. These decoders relax the decoding problem into that of a linear program. LP decoders have a performance that is competitive with BP decoders [3] and a similar complexity [4]. They have the advantage of proven performance bounds as well as a guarantee that any integer-valued output is the ML codeword [5]. Recently, applying LP decoding to an ISI channel has been explored. It was shown that joint equalization and decoding may be accomplished within the LP framework by formulating the equalization task as a linear program [6, 7]. A joint LP equalization and decoding scheme similar to a turbo equalizer using a BCJR equalizer with a LP decoder was presented, which demonstrates that joint channel equalization and LP decoding can be achieved with a complexity similar to a turbo equalizer [8].

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant number OCI-1053575.

This paper introduces the SISO LP decoder. It is demonstrated that valuable soft information may be gleaned from the codeword or pseudo-codeword output from the LP decoder. The SISO LP decoder enables joint equalization and decoding using an arbitrary equalizer within a turbo equalizer setting. The ability to separate the equalization from the LP decoding task is beneficial, for example, when a short data packet must be transmitted over a long ISI channel. In this situation, a MMSE-based turbo equalizer may be preferred due to complexity issues, and a LP decoder may be preferred due to short block length issues. The soft output from the SISO LP decoder is also suitable for use by the channel estimator and in other applications where *a priori* information about the transmitted symbols is useful.

2. SYSTEM DESCRIPTION

This section describes the structure of the communication system that is depicted in Figure 1.

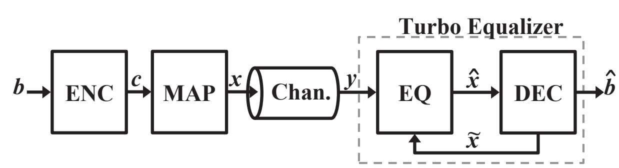

Fig. 1. Communication system model.

A block of bits $\boldsymbol{b}$ are protected with a LDPC code to create $\boldsymbol{c}$ and then mapped to N BPSK symbols $\boldsymbol{x}$. These symbols are transmitted through a LTI, frequency selective channel with channel response $\boldsymbol{h}$. The received signal $\boldsymbol{y}$ is assumed to be described by Equation 1. The parameter L is the length of the discrete channel response. The noise $\boldsymbol{\omega}$ is AWGN with variance σ^2.

$$y_n = \sum_{i=0}^{L-1} h_i x_{n-i} + \omega_n \tag{1}$$

The signal is received by a turbo equalizer that is composed of a SISO MMSE equalizer and a SISO LP decoder.

The decoder is discussed in detail in Section 4. At turbo iteration k, the equalizer takes as inputs the signal $\boldsymbol{y}$ and soft symbols $\tilde{\boldsymbol{x}}^{(k)}$. The soft symbols are distributions representing the knowledge the receiver has about the transmitted symbols. As is typical [1], they are modeled by the Gaussian distribution described by Equation 2. The parameter $\boldsymbol{\mu}^{(k)}$ is the mean of the soft estimate of $\boldsymbol{x}$, and $\mathbf{V}^{(k)}$ is its covariance matrix.

$$\tilde{\boldsymbol{x}}^{(k)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(k)}, \mathbf{V}^{(k)}\right) \tag{2}$$

The MMSE equalizer outputs soft symbols $\hat{\boldsymbol{x}}$ that are calculated in Equation 3. The vector $\boldsymbol{h}_i$ is $\boldsymbol{h}$ shifted by i time steps and truncated to the first N taps. The matrix $\mathbf{H}$ is constructed by setting its ith row equal to $\boldsymbol{h}_i$.

$$\hat{x}_i^{(k)} = \boldsymbol{h}_i^T \left(\sigma^2 \mathbf{I} + \mathbf{H}\mathbf{V}_i^{(k)}\mathbf{H}^T\right)^{-1} \left(\boldsymbol{y} - \mathbf{H}\boldsymbol{\mu}_i^{(k)}\right) \tag{3}$$

In order to ensure that the equalizer only outputs extrinsic information, $\tilde{x}_i^{(k-1)}$ is not involved in calculating $\hat{x}_i^{(k)}$ [9]. To ensure an extrinsic output, the vector $\boldsymbol{\mu}_i^{(k)}$ is equal to $\boldsymbol{\mu}^{(k)}$ except for the ith entry, which equals 0. Also, $\mathbf{V}_i^{(k)}$ is equal to $\mathbf{V}^{(k)}$ with its ith row and ith column replaced by zeros except entry (i, i), which is replaced by 1. Section 4 describes a method to reduce the complexity of the calculation of $\hat{\boldsymbol{x}}^{(k)}$.

It is convenient to transform the output into a log likelihood ratio (LLR), denoted by $\gamma_i^{(k)}$. Equation 4 describes the LLR of a BPSK symbol output from the equalizer.

$$\gamma_i^{(k)} = \frac{2\hat{x}_i^{(k)}}{1 - \boldsymbol{h}_i^T \left(\sigma^2 \mathbf{I} + \mathbf{H}\mathbf{V}_i^{(k)}\mathbf{H}^T\right)^{-1} \boldsymbol{h}_i} \tag{4}$$

3. LP DECODING

LP decoding transforms the problem of finding the maximum likelihood (ML) estimate of $\boldsymbol{x}$ given the code structure into a constrained linear optimization problem. In order to describe the ith symbol estimate $\check{x}_i$ with only positive variables so that the problem may be formulated as a linear program, define a variable representing the estimate of the ith code bit $\check{c}_i$ and a second variable $\check{c}_i^- = 1 - \check{c}_i$. Then $\check{x}_i$ equals $\check{c}_i - \check{c}_i^-$.

The solution to the LP optimization is constrained to lie within a polytope defined by a relaxation of the parity check constraints that define the structure of the code. The mth parity constraint may require that the binary sum of the M bits in the set S_m be equal to zero, which is equivalent to requiring that the sum of the M bits in the set S_m be even. In LP decoding, the parity check constraint is relaxed to Equation 5.

$$\begin{array}{lll} 0 & \leq \sum_{i \in S_m} \check{c}_i & \leq M - mod_2\{M\} \\ mod_2\{M\} & \leq \sum_{i \in S_m} \check{c}_i^- & \leq M \end{array} \tag{5}$$

The requirement that the bits are either 0 or 1 is relaxed to the constraint described in Equation 6, and the construction of x_i is relaxed to Equation 7.

$$\begin{array}{lll} 0 \leq & \check{c}_i & \leq 1 \\ 0 \leq & \check{c}_i^- & \leq 1 \end{array} \tag{6}$$

$$-1 \leq \check{c}_i - \check{c}_i^- \leq 1 \tag{7}$$

The objective function is equal to the sum of the symbol estimates weighted by their log likelihood ratios $\boldsymbol{\gamma}$. For BPSK, the objective function is given by Equation 8.

$$\boldsymbol{f} = -\boldsymbol{\gamma}^T \left(\check{\boldsymbol{c}} - \check{\boldsymbol{c}}^-\right) \tag{8}$$

The LP optimization converges to constraint vertex $\check{\boldsymbol{x}}$. This vertex may be the ML codeword or a pseudo-codeword, which is a fractional solution to the linear programming problem. It will be shown in Section 6 that pseudo-codewords contain valuable information that may be used to improve the performance of the receiver. A novel method for extracting soft information from $\check{\boldsymbol{x}}$ is presented in the next section.

4. SISO LP DECODING

The SISO LP decoder extracts soft information $\tilde{\boldsymbol{x}}^{(k)}$ from the solution $\check{\boldsymbol{x}}^{(k)}$ to the LP problem. It is imagined that $\check{\boldsymbol{x}}^{(k)}$ is an observation of $\bar{\boldsymbol{x}}$, the mean of the "true" soft output given the input, that is quantized to the nearest constraint polytope vertex. The additional uncertainty that this quantization introduces is accounted for in the covariance of $\tilde{\boldsymbol{x}}^{(k)}$.

Since $\bar{\boldsymbol{x}}^{(k)}$ is not directly observed, it may be modeled as a uniform distribution over a region determined by the code polytope. This region is a pyramid oriented along the line between the origin and $\check{\boldsymbol{x}}^{(k)}$ and bounded by the constraint $-1 \leq \boldsymbol{x} \leq 1$. It may be bounded further by observing that the turbo equalizer is expected to decrease the entropy $\hbar$ of its output after each turbo iteration as it improves its estimate of $\boldsymbol{x}$. A scalar $\alpha^{(k)}$ is found so that the entropy associated with the soft symbol $\alpha^{(k)}\check{\boldsymbol{x}}^{(k)}$ is equal to the entropy associated with $\tilde{\boldsymbol{x}}^{(k-1)}$. The $\bar{\boldsymbol{x}}^{(k)}$ region is the section of the pyramid that is outside a radius equal to α times the magnitude of $\check{\boldsymbol{x}}^{(k)}$.

An example of the $\bar{\boldsymbol{x}}^{(k)}$ region for a trivial code that encodes one bit into two code bits is depicted in Figure 2. The transmitted symbol $\boldsymbol{x}$ corresponds to the zero codeword. The points $(-1, -1)$ and (1,1) are valid codewords, and there are six additional pseudo-codewords in the polytope. The extrinsic decision $\check{\boldsymbol{x}}^{(k)}$ is constructed from the ith element of $\check{\boldsymbol{x}}_i^{(k)}$ and the jth element of $\check{\boldsymbol{x}}_j^{(k)}$. The $\bar{\boldsymbol{x}}^{(k)}$ region is bounded by the dashed purple circle, lines connecting the origin to the points midway between $\check{\boldsymbol{x}}^{(k)}$ and its neighbors, and $x_j = -1$.

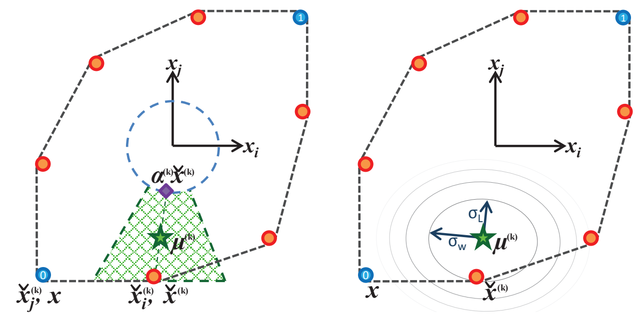

Fig. 2. An example of a SISO LP decoding scenario

For convenience, a Gaussian distribution capturing the aggregate properties of the code polytope is used to approximate the uniform distribution over the $\bar{\boldsymbol{x}}^{(k)}$ region. The mean $\boldsymbol{\mu}^{(k)}$ of the distribution is given by Equation 9. As depicted in Figure 2(b), the variance ${\sigma_L^{(k)}}^2$ of $\bar{\boldsymbol{x}}^{(k)}$ along the $\check{\boldsymbol{x}}^{(k)}$ direction is a function of $\alpha^{(k)}$, and the variance σ_W^2 of $\bar{\boldsymbol{x}}^{(k)}$ along every direction orthogonal to $\check{\boldsymbol{x}}^{(k)}$ is a constant parameter that is related to the distance between neighboring vertices on the constraint polytope.

As is typical, the soft output $\tilde{\boldsymbol{x}}^{(k)}$ is modeled as a Gaussian distribution [1]. Its mean is equal to $\boldsymbol{\mu}^{(k)}$ and its covariance is given by Equation 10. Define $\boldsymbol{\delta}_i$ to be a vector of all zeros except for the ith element which is a one. Define $\mathbf{A}^{(k)}$ to be a rotation matrix that rotates $\boldsymbol{\delta}_0$ into the direction of $\check{\boldsymbol{x}}^{(k)}$. Define $\boldsymbol{\Sigma}^{(k)}$ to be the diagonal matrix with its jth element equal to $1 - |\mu_j^{(k)}|^2$, which would describe the covariance matrix of $\tilde{\boldsymbol{x}}^{(k)}$ if this were a MAP decoder [1]. A second term is added to $\mathbf{V}^{(k)}$ to represent the quantization effect of the polytope of the code.

$$\boldsymbol{\mu}^{(k)} = \frac{1+\alpha^{(k)}}{2}\check{\boldsymbol{x}}^{(k)} \tag{9}$$

$$\mathbf{V}^{(k)} = \boldsymbol{\Sigma}^{(k)} + \mathbf{A}^{(k)} \begin{bmatrix} {\sigma_L^{(k)}}^2 & 0 & & 0 \\ 0 & \sigma_W^2 & & 0 \\ & & \ddots & \\ 0 & 0 & & \sigma_W^2 \end{bmatrix} {\mathbf{A}^{(k)}}^T \tag{10}$$

As with the SISO equalizer in Section 2, the SISO LP decoder must be carefully constructed to output only extrinsic information. This may be accomplished by solving the optimization problem for each code bit i with the ith element of the LLR vector set to 0. Define the vector $\check{\boldsymbol{x}}_i^{(k)}$ to be the solution to the optimization problem at turbo iteration k with $\gamma_i = 0$. The extrinsic solution $\check{\boldsymbol{x}}^{(k)}$ is constructed elementwise by taking the ith element of $\check{\boldsymbol{x}}_i^{(k)}$ to be the ith element of $\check{\boldsymbol{x}}^{(k)}$. Since the optimization may be implemented with linear complexity in N, and the optimization is performed N times per turbo iteration, the SISO LP decoder presented here has quadratic complexity.

If σ_W^2 may be approximated as zero, then $\mathbf{V}_i^{(k)}$ may be described by Equation 11, which allows the inverse of $\left(\sigma^2\mathbf{I} + \mathbf{H}\mathbf{V}_i^{(k)}\mathbf{H}^T\right)$ in the MMSE solution to be computed efficiently. The superscripts indicating the turbo iteration are left off the following equations.

$$\begin{aligned}\mathbf{V}_i =& \boldsymbol{\Sigma} + \sigma_L^2\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T + \left(|\mu_i|^2 + \sigma_L^2\boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\boldsymbol{\delta}_i\right)\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T \\ &-\sigma_L^2\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T - \sigma_L^2\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T \end{aligned} \tag{11}$$

Define the following matrices:

$$\begin{aligned}\mathbf{X} &:= \sigma^2\mathbf{I} + \mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^T + \sigma_L^2\mathbf{H}\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\mathbf{H}^T \\ \mathbf{B}_i &:= \mathbf{X} + \left(|\mu_i|^2 + \sigma_L^2\boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\boldsymbol{\delta}_i\right)\boldsymbol{h}_i\boldsymbol{h}_i^T \\ \mathbf{C}_i &:= \mathbf{B}_i - \sigma_L^2\boldsymbol{h}_i\boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\mathbf{H}^T \\ \mathbf{S}_i &:= \sigma^2\mathbf{I} + \mathbf{H}\mathbf{V}_i^{(k)}\mathbf{H}^T = \mathbf{C}_i - \sigma_L^2\mathbf{H}\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\boldsymbol{\delta}_i\boldsymbol{h}_i^T\end{aligned}$$

Then the inverse of $\mathbf{S}_i^{(k)}$ may be computed efficiently by applying the Woodbury identity three times. The advantage of doing this is that the MMSE equalizer is only required to calculate one matrix inverse when computing $\hat{\boldsymbol{x}}^{(k)}$.

$$\begin{aligned}\mathbf{B}_i^{-1} &= \mathbf{X}^{-1} - \frac{\mathbf{X}^{-1}\boldsymbol{h}_i\boldsymbol{h}_i^T\mathbf{X}^{-1}}{\boldsymbol{h}_i^T\mathbf{X}^{-1}\boldsymbol{h}_i + \left(|\mu_i|^2 + \sigma_L^2\boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\boldsymbol{\delta}_i\right)^{-1}} \\ \mathbf{C}_i^{-1} &= \mathbf{B}_i^{-1} - \frac{\mathbf{B}_i^{-1}\boldsymbol{h}_i\boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\mathbf{H}^T\mathbf{B}_i^{-1}}{\boldsymbol{\delta}_i^T\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\mathbf{H}^T\mathbf{B}_i^{-1}\boldsymbol{h}_i - \sigma_L^{-2}} \\ \mathbf{S}_i^{-1} &= \mathbf{C}_i^{-1} - \frac{\mathbf{C}_i^{-1}\mathbf{H}\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\boldsymbol{\delta}_i\boldsymbol{h}_i^T\mathbf{C}_i^{-1}}{\boldsymbol{h}_i^T\mathbf{C}_i^{-1}\mathbf{H}\mathbf{A}\boldsymbol{\delta}_0\boldsymbol{\delta}_0^T\mathbf{A}^T\boldsymbol{\delta}_i - \sigma_L^{-2}}\end{aligned} \tag{12}$$

Input: LLRs $\boldsymbol{\gamma}^{(k)}$ describing the bit probabilities.
Solve the LP problem for each $\check{\boldsymbol{x}}_i^{(k)}$ with the ith element of $\boldsymbol{\gamma}^{(k)}$ set to zero.
Construct $\check{\boldsymbol{x}}^{(k)}$ by setting its ith element equal to the ith element of $\check{\boldsymbol{x}}_i^{(k)}$.
Find $\alpha^{(k)}$ such that the entropy $\hbar\{\alpha^{(k)}\check{\boldsymbol{x}}^{(k)}\} = \hbar\{\tilde{\boldsymbol{x}}^{(k-1)}\}$
Calculate $\boldsymbol{\mu}^{(k)} = \frac{1+\alpha^{(k)}}{2}\check{\boldsymbol{x}}^{(k)}$.
Calculate $\mathbf{V}^{(k)}$ using Equation 10.
Output: the soft symbols $\tilde{\boldsymbol{x}}^{(k)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(k)}, \mathbf{V}^{(k)}\right)$.

Table 1. The SISO LP decoding algorithm.

5. SIMULATION SETUP

A turbo equalizer with a SISO LP decoder and a turbo equalizer with a SISO BP decoder were simulated. The channel was the Proakis Channel B given by $h = [0.407, 0.815, 0.407]$ with real-valued AWGN. The transmitted symbols were BPSK. The turbo equalizer performed up to ten turbo iterations, as going beyond ten iterations did not significantly affect the performance of the system. The simulations were run until 100 block errors were encountered.

A rate $\frac{1}{3}$ LDPC code with a block length of 96 code bits was used. This code was taken from [10], which is a repository of codes designed to have good minimum distance properties. The BP decoder ran a maximum of 100 message passing iterations per turbo iteration. The constraint polytope used by the LP decoder was tightened by adding redundant constraints, which were created by taking a linear combination of two constraints that were selected at random. The polytope was tightened until the density of the polytope vertices was sufficient to calculate Equation 3 more efficiently using Equation 12. The parameters describing the $\bar{\boldsymbol{x}}^{(k)}$ distribution were determined experimentally. The parameter $\sigma_L^{(k)} = \frac{1}{12}\left(1 - \alpha^{(k)}\right)$ and the parameter $\sigma_W = 0$.

6. SIMULATION RESULTS

The BER versus SNR curves for the turbo equalizer using the SISO LP decoder are shown in Figure 3. Each curve represents the performance of the system after the specified number of turbo iterations. Iteration 0 represents the initial pass of the equalizer and decoder. Without turbo equalization, this system requires 37 dB of SNR to achieve a BER of 10^{-4}. After ten iterations of the turbo equalizer, the required SNR to achieve a BER of 10^{-4} is reduced by 25 dB. Figure 4 compares the BER performance after ten iterations of a turbo equalizer using a SISO LP decoder with one using a SISO BP decoder. It can be seen that the system with the LP decoder outperforms the system with the BP decoder when the BER is required to be less than 10^{-6}.

7. CONCLUSION

The SISO linear programming decoder was introduced. It was demonstrated that a turbo equalizer using a SISO LP decoder can outperform one using a SISO BP decoder given a short block length LDPC code. The SISO LP decoder is superior in this situation, because it does not suffer from convergence issues the BP decoder encounters that arise from short cycles inherent in short block length codes. An efficient SISO MMSE equalizer that can handle the correlated soft symbols output from a SISO LP decoder was also discussed.

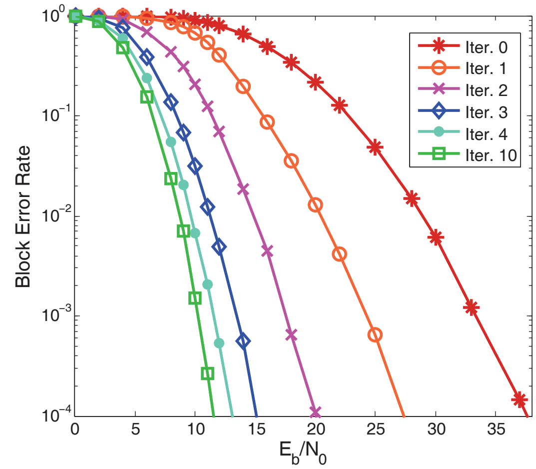

Fig. 3. The performance of a turbo equalizer using a SISO LP decoder to decode a rate $\frac{1}{3}$ LDPC code after a specified number of turbo iterations.

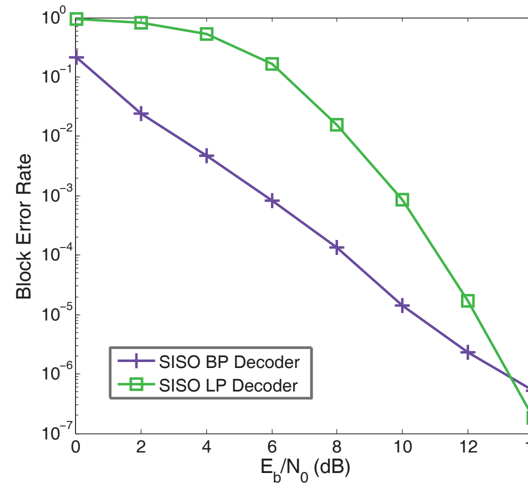

Fig. 4. The performance after ten iterations of a turbo equalizer using a SISO LP decoder compared to the performance of a turbo equalizer using a SISO BP decoder. Both turbo equalizers use the same MMSE equalizer and LDPC code.

The receiver presented in this paper is a proof-of-concept showing that a SISO LP decoder is a promising potential solution to the problem of approaching the capacity of an ISI channel with a short data transmission. Its performance may be improved by more accurately modeling the quantization depicted in Figure 2. It is left to future work to explore the performance of a SISO LP decoder given a long code block. Given prior work on LP decoders, it is suspected that the SISO LP decoder has an application in this regime when the bit error rate must be extremely low. To make it practical, the SISO LP decoding problem must be reformulated to leverage the structure of the problem in order to achieve a low complexity.

8. REFERENCES

[1] M. Tuchler, R. Koetter, and A.C. Singer, "Turbo equalization: principles and new results," *IEEE Transactions on Communications*, vol. 50, no. 5, pp. 754–767, 2002.

[2] C. Di, D. Proietti, I.E. Telatar, T.J. Richardson, and R.L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, 2002.

[3] J. Feldman, "Decoding error-correcting codes via linear programming," *PhD thesis, Massachusetts Institute of Technology*, 2003.

[4] P.O. Vontobel and R. Koetter, "Towards low-complexity linear-programming decoding," *Proceedings of the 4th International Symposium on Turbo Codes and Related Topics*, pp. 1–9, 2006.

[5] J. Feldman, M.J. Wainwright, and D.R. Karger, "Using linear programming to decode binary linear codes," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 954–972, 2005.

[6] M.H. Taghavi and P.H. Siegel, "Graph-based decoding in the presence of ISI," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2188–2202, 2011.

[7] Byung-Hak Kim and Henry D Pfister, "Joint decoding of LDPC codes and finite-state channels via linear-programming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 8, pp. 1563–1576, 2011.

[8] Byung-Hak Kim and Henry D Pfister, "An iterative joint linear-programming decoding of LDPC codes and finite-state channels," *IEEE International Conference on Communications*, pp. 1–6, 2011.

[9] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[10] D. MacKay, "Encyclopedia of sparse graph codes," *http://www.inference.phy.cam.ac.uk/mackay/codes/data.html*, 2012.