Decoding and Turbo Equalization For LDPC Codes Based on Nonlinear Programming

Ronald A. Iltis
Department of Electrical and Computer Engineering
University of California,
Santa Barbara, CA 93106-9560
Tel: (805) 893-4166, E-mail: iltis@ece.ucsb.edu

ABSTRACT

Decoding and Turbo Equalization (TEQ) algorithms based on the Sum-Product Algorithm (SPA) are well established for LDPC codes. However there is increasing interest in linear and nonlinear programming (NLP)-based decoders which may offer computational and performance advantages over the SPA. We present NLP decoders and Turbo equalizers based on an Augmented Lagrangian formulation of the decoding problem. The decoders update estimates of both the Lagrange multipliers and transmitted codeword while solving an approximate quadratic programming problem. Simulation results show that the NLP decoder performance is intermediate between the SPA and bit-flipping algorithms. The NLP may thus be attractive in some applications as it eliminates the tanh/atanh computations in the SPA.

KEYWORDS

Forward error correction, belief propagation, LDPC codes, nonlinear programming.

INTRODUCTION

A wide range of algorithms have been developed for decoding LDPC codes based on MAP and ML approximations [1, 2, 3, 4, 5, 6]. The sum-product algorithm [6] is viewed as having the

*This work was supported in part by the International Foundation for Telemetering
best performance but requires computation of differential probabilities using the \texttt{tanh()} and \texttt{atanh()} functions. Various approximations to the SPA computations have been developed reducing the need for nonlinear function evaluations [5] with bit-flipping being the simplest [7]. Alternative approaches based on ML decoding using linear and quadratic programming have been shown to yield lower word-error rates than the SPA in some cases [3]. However, due to the large number of vertices in the fundamental polytope defining the codewords in LP [1], it is not clear that LP is feasible for long LDPC codeword lengths.

Recently, gradient projection decoders based on nonlinear codeword constraints have been proposed [8, 7]. The problem in such algorithms is selecting the search direction and effective Lagrange multipliers. Here the gradient projection method is developed as an instance of nonlinear programming using the Augmented Lagrangian method [9], which jointly estimates the Lagrange multipliers while decoding. The problem of Turbo Equalization using the Augmented Lagrangian method is also considered, and a simplified TEQ algorithm is simulated.

\textbf{SIGNAL MODEL AND NLP FORMULATION OF ML DECODING}

Consider a scalar Gaussian binary coded channel. The vectorized received signal is \( r = c + n \) where \( r = [r_1 r_2 \ldots r_n]^T \) for a length-\( n \) code. The code symbols \( c_k \in \{-1, +1\} \) satisfy the binary-to-antipodal mapping \( \{0, 1\} \to \{1, -1\} \) and \( c = [c_1 \ldots c_n]^T \) is an LDPC codeword. The codewords correspond to the sparse parity check matrix \( H \in \mathbb{B}^{m \times n} \), with elements \( H_{q,l} \in \{0, 1\} \). The i.i.d. Gaussian noise samples satisfy \( n \sim \mathcal{N}(0, (N_0/(2E_bR_c)))I \), where \( R_c = k/n \) is the code rate. The parity checks for the code are then represented in product form [8, 7] as

\[
p_q = \prod_{l=1}^{n} c_l^{H_{q,l}},
\]

for \( q = 1, 2, \ldots, m \). The parities satisfy \( p_q = 1 \ \forall q \) if \( c \) is a codeword. The vector of parities is denoted \( p = [p_1 \ldots p_m]^T \).

The ML decoding problem is posed as the following optimization.

\[
\text{Minimize} \quad ||r - c||^2,
\]

\[
\text{Subject to} \quad \left( 1 - \prod_{l=1}^{n} c_l^{H_{q,l}} \right) = (1 - p_q) = 0 \quad \forall q,
\]

and \(-1 \leq c_k \leq 1, \quad k = 1, \ldots, n.\)

The parity checks are set as equality constraints, while the antipodal code symbols satisfy the inequalities \( |c_k| \leq 1 \) corresponding to a relaxation of the ML decoding problem onto the unit hypercube \([-1, +1]^n\).

The feasible points for the ML decoder optimization (2) are characterized by the following Theorem.
Theorem 1 The set of feasible points of the decoding optimization (2) is equivalent to the set of codewords \( \{ c \} \in \{-1, +1\}^n \) determined by the parity check matrix \( H \).

Proof 1 Sufficiency – assume \( c \) is an antipodal codeword, then all parity constraints are satisfied with equality \( (p_q = 1) \forall q \) and all relaxation inequality constraints are active at the boundaries and thus satisfied. Necessity – Assume \( c \) is not an antipodal codeword. Define the vector sign(\( c \)) as the elementwise sign operator. Even if \( c \) is not an antipodal codeword, the \( m \) parity checks (equality constraints) can still be satisfied if sign(\( c \)) has correct parity, and \( \prod^n_{l=1} |c_l H_{n,l}| = 1 \) \( \forall q \). But if \( c \) is not antipodal then at least one \( |c_l| > 1 \) and at least one \( |c_{l'}| < 1 \). The relaxation constraints on \([-1, +1]^n\) are then violated and \( c \) cannot be a feasible point (contradiction).

AUGMENTED LAGRANGIAN DECODER (ALD)

The following Augmented Lagrangian using a quadratic penalty [9] is proposed corresponding to the optimization (2).

\[
L(c, \lambda, \lambda^+, \lambda^-) = \frac{1}{2} ||r-c||^2 + \lambda^T (1-p) + \frac{1}{2\mu} ||1-p||^2 + \sum^n_{k=1} \lambda^+_k (c_k - 1) + \sum^n_{k=1} \lambda^-_k (-c_k - 1). \tag{3}
\]

The multipliers \( \lambda^+_k, \lambda^-_k \) enforce the relaxation on \([-1, 1]^n\). The multiplier vector \( \lambda \) corresponds to the \( m \) parity constraints, and \( \mu \) is the penalty parameter which is gradually decreased in the algorithm developed below.

The following derivatives of the parity checks are required for the ALD.

\[
\hat{c}_{k,q} = \frac{\partial p_q}{\partial c_k} = H_{q,k} \prod_{l \neq k} c_l^{H_{n,l}}. \tag{4}
\]

The derivative \( \hat{c}_{k,q} \) can be interpreted as a message from the \( q \)-th check to \( k \)-th bit. The check matrix \( \hat{C} \in \mathbb{R}^{n,m} \) is also defined with elements \( \hat{c}_{k,q} \).

Sequential quadratic programming (SQP) solutions for minimizing (3) are infeasible for long codes, as they require computation or approximation of an \( n \times n \) Hessian matrix. A steepest-descent based updating is used here, which is shown to lead to a message passing algorithm with similar structure to SPA. The steps are as follows, beginning with the current solution \( c^n, \lambda^n \).

(A1) Update \( c^{n+1} = c^n - (\partial L(c, \lambda^n, \lambda^+, \lambda^-)/\partial c)|_{c^n} \)

(A2) Update \( (\lambda^+)^n, (\lambda^-)^n \rightarrow (\lambda^+)^{n+1}, (\lambda^-)^{n+1} \) such that \( |c^{n+1}_k| \leq 1 \) \( \forall k \).

(A3) Recompute \( c^{n+1} \) using updated \( (\lambda^+)^{n+1}, (\lambda^-)^{n+1} \)
(A4) Lagrange multiplier update: \( \lambda^{n+1} = \lambda^n + \frac{1}{\mu^n}(1 - p^n) \).

(A5) Decrease penalty parameter \( \mu^{n+1} = \beta \mu^n, 0 < \beta < 1 \).

(A6) Test for convergence: \( \hat{c} = (1 - \text{sign}(c^{n+1})) \mod 2 \). If \( H\hat{c} \mod 2 = 0 \) break.

(A7) Go to (A1).

Step (A1) above becomes the following using the result for the derivative of the parity vector \( p \) in (4).

\[
c^{n+1} = r + \hat{C}^n (\lambda^n + (1/\mu^n)(1 - p^n)) - \lambda^+ + \lambda^-.
\] (5)

Now consider updating the relaxation or box constraints in step (A2). If \( |c_k^{n+1}| < 1 \) with \( \lambda^+_k = \lambda^-_k = 0 \) then both constraints are inactive and satisfied, and (A2) becomes \( (\lambda^+_k)^{n+1} = (\lambda^-_k)^{n+1} = 0 \). If \( c_k^{n+1} > 1 \) when \( \lambda^+_k = \lambda^-_k = 0 \), the box constraint is satisfied with equality by selecting \( \lambda^- \) sufficiently positive, and similarly for \( c_k^{n+1} < 1 \) with \( \lambda^+ \) also sufficiently positive. Thus (A3) becomes the projection

\[
c^{n+1} = \left[ r + \hat{C}^n (\lambda^n + (1/\mu^n)(1 - p^n)) \right]_{-1}^{+1},
\] (6)

where the Euclidean projection is \( [x_k]_{-1}^{+1} = x_k, |x_k| < 1, +1, x_k > 1, -1, x_k < -1 \), and the vector projection operates element-wise.

The overall ALD in (A1)-(A7) is equivalent to the projection decoder of [8], but with an added updating of Lagrange multiplier estimates. The relative advantages of decoding using linear parity constraints (fundamental polytope) in [1, 2, 3] versus the nonlinear parity constraints in (2) used in the ALD appear to be as follows. The polytope constraint [1] leads to a decoder implementable via the Simplex algorithm that is guaranteed to converge to an optimal solution, albeit possibly a fractional codeword. In contrast, SQP algorithms for nonlinear constraints yield quadratic or superlinear convergence only if initialized sufficiently close to the optimum [9]. Thus, the steepest descent method above will be less likely than SQP to yield a global minimum of the Augmented Lagrangian at each step. However, the polytope method requires \( m2^{w_r-1} \) parity constraints [2] where \( w_r \) is the row weight of \( H \), compared with \( m \) parity equations in (2). Furthermore the ALD here leads to simple message-passing algorithms strongly resembling the sum-product structure. Thus, the augmented Lagrangian decoder with nonlinear parity constraints is worthy of investigation due to its close connection to the sum-product algorithm but with simpler message-passing rules and fewer constraints than the LP/polytope formulation.

**TURBO EQUALIZATION USING THE AUGMENTED LAGRANGIAN**

Consider an intersymbol interference channel arising from multipath, bandlimiting, a MIMO system, or a combination thereof. Assume QPSK modulation with symbols \( s_k = (c_k + j c_{n/2+k})/\sqrt{2}, k = 1, \ldots, n/2 \), where the LDPC codeword length \( n \) is even. Then \( r = \)
\( \sqrt{2E_s} \mathbf{F} \mathbf{s} + \mathbf{n} \), where \( E_s = 2E_b R_c \) is the energy per symbol and \( \mathbf{n} \sim \mathcal{CN}(0,2N_0) \). The received signal model for QPSK can be written in terms of purely real quantities as

\[
\mathbf{r}_r = \mathbf{F}_r \mathbf{c} + \mathbf{n}_r
\]

and \( \mathbf{n}_r \sim \mathcal{N}(0, (N_0/(2E_b R_c)) \mathbf{I}) \). The real-valued received vector is \( \mathbf{r}_r = [\text{Re}\{\mathbf{F}\} - \text{Im}\{\mathbf{F}\}; \text{Im}\{\mathbf{F}\} \text{Re}\{\mathbf{F}\}]^T \).

The channel matrix \( \mathbf{F} \) may be highly structured, e.g. circulant [10, 11] as in the single-carrier system in the simulation here.

A direct formulation of the optimization (2) to incorporate the channel \( \mathbf{F}_r \) unfortunately does not lead to the message-passing projection decoder structure in algorithm (A1)-(A7).

To see this, consider the obvious modification of the Lagrangian in (3)

\[
L(\mathbf{c}, \lambda, \lambda^+, \lambda^-) = \frac{1}{2} ||\mathbf{r}_r - \mathbf{F}_r \mathbf{c}||^2 + \lambda^T (1 - \mathbf{p}) + \frac{1}{2\mu} ||1 - \mathbf{p}||^2 + \sum_{k=1}^{n} \lambda^+_k (c_k - 1) + \sum_{k=1}^{n} \lambda^-_k (-c_k - 1).
\]

Note that the objective function can be rewritten as

\[
||\mathbf{r}_r - \mathbf{F}_r \mathbf{c}||^2 = [\mathbf{c} - \hat{\mathbf{c}}_{LS}]^T \mathbf{F}_r^T \mathbf{F}_r [\mathbf{c} - \hat{\mathbf{c}}_{LS}]
\]

\[
\hat{\mathbf{c}}_{LS} = [\mathbf{F}_r^T \mathbf{F}_r]^{-1} \mathbf{F}_r^T \mathbf{r}_r.
\]

Step (A1) now becomes

\[
\mathbf{c}^{n+1} = \hat{\mathbf{c}}_{LS} + (\mathbf{F}_r^T \mathbf{F}_r)^{-1} \left( \hat{\mathbf{C}}_n (\lambda^+ + (1/\mu^n)(1 - \mathbf{p})^n) - \lambda^+ + \lambda^-ight).
\]

The updating of the Lagrange multipliers \( \lambda^+, \lambda^- \) no longer corresponds to a Euclidean projection of \( \mathbf{c} \) on \([-1,1]^n\) due to premultiplication of the multipliers by \( (\mathbf{F}_r^T \mathbf{F}_r)^{-1}\).

Simulations suggest that the best approach to TEQ based on the Augmented Lagrangian uses the following formulation with the minimum mean-square error detector as the starting point for updating \( \mathbf{c} \).

\[
L(\mathbf{c}, \lambda, \lambda^+, \lambda^-) =
\]

\[
\frac{1}{2} ||\mathbf{c} - \hat{\mathbf{c}}_{MMSE}||^2 + \lambda^T (1 - \mathbf{p}) + \frac{1}{2\mu} ||1 - \mathbf{p}||^2 + \sum_{k=1}^{n} \lambda^+_k (c_k - 1) + \sum_{k=1}^{n} \lambda^-_k (-c_k - 1)
\]

\[
\hat{\mathbf{c}}_{MMSE} = [\mathbf{F}_r^T \mathbf{F}_r + (N_0/(2E_b R_c)) \mathbf{I}]^{-1} \mathbf{F}_r^T \mathbf{r}_r.
\]

The TEQ then is identical to (A1)-(A7) for the scalar Gaussian channel, except that the linear MMSE detected symbols \( \hat{\mathbf{c}}_{MMSE} \) replace \( \mathbf{r} \).

**SIMULATION RESULTS AND CONCLUSIONS**

The Augmented Lagrangian, SPA and bit-flipping decoders were simulated for the scalar Gaussian channel in Fig 1. All decoders were simulated to a maximum of 32 iterations for
each received vector $r$. A (1008, 504) Gallager code from [12] was used. As expected, the performance of the ALD is intermediate between the SPA and bit-flipping decoder.

The ALD Turbo-equalizer was simulated and compared with a SPA-based TEQ and bit-flipping equalizer in Fig. 2, again with 32 iterations maximum for each decoder. A (204, 102) Gallager code again from [12] was employed. The SPA algorithm follows the method of [13]. The bit-flipping method treats $\hat{c}_{MMSE}$ as an equivalent received signal vector and uses the inversion algorithm of [7]. Again, the ALD TEQ performs midway between the SPA and bit-flipping.

The results here indicate that the ALD may be a better performing somewhat more complex alternative to bit-flipping if a simpler decoder than SPA without the need to compute tanh functions is desired. However, further work to optimize the updating of the penalty parameter $\mu$ is required to reduce the BER. Finally, alternative minimizations of the Lagrangian at each penalty step based on line-search methods, and if possible approximate Newton methods should be considered.

Figure 1: BER for decoders, (1008,504) Gallager code scalar Gaussian channel.
Figure 2: BER for Turbo equalizers, (204,102) Gallager code length 8 Rayleigh fading channel.
References


