6.451 Principles of Digital Communication II
MIT, Spring 2005

Wednesday, April 27, 2005
Handout #21
Due: Wednesday, May 4, 2005

Problem Set 9

Problem 8.3 (revised) (BCJR (sum-product) decoding of SPC codes)

As shown in Problem 6.4 or Figure 1 of Chapter 10, any $(n, n-1, 2)$ single-parity-check code has a two-state trellis diagram. Consider the $(8, 7, 2)$ code, and let the received sequence from a discrete-time AWGN channel with input alphabet $\{\pm 1\}$ and noise variance $\sigma^2 = 1$ be given by $\mathbf{r} = (0.1, -1.0, -0.7, 0.8, 1.1, 0.3, -0.9, 0.5)$. Perform BCJR (sum-product) decoding of this sequence to determine the APP vector for each symbol Y_k. [Hint: the special algorithm given in Section 13.5.2 (see Problem 9.5) to implement the sum-product update rule for zero-sum nodes may be helpful here.]

Compare the complexity of BCJR decoding to Viterbi algorithm decoding (Problem 6.6).

Problem 9.1 (Iterative decoding on the BEC)

(a) Using a graph of the $(8, 4, 4)$ code like that of Figure 1 of Chapter 13 for iterative decoding, decode the received sequence $(1, 0, 0, ?, 0, ?, ?, ?)$. Then try to decode the received sequence $(1, 1, 1, 1, ?, ?, ?, ?)$. Why does decoding fail in the latter case? Give both a local answer (based on the graph) and a global answer (based on the code).

(b) For the received sequence $(1, 1, 1, 1, ?, ?, ?, 0)$, show that iterative decoding fails but that global (*i.e.,* ML) decoding succeeds.

Problem 9.2 (Simulation of LDPC decoding on a BEC)

(a) Perform a simulation of iterative decoding of a regular $(d_\lambda = 3, d_\rho = 6)$ LDPC code on a BEC with $p = 0.45$ (*i.e.,* on Figure 9 of Chapter 13), and show how decoding gets stuck at the first fixed point ($q_{\ell \to r} \approx 0.35, q_{r \to \ell} \approx 0.89$). About how many iterations does it take to get stuck?

(b) By simulation of iterative decoding, compute the coordinates of the fixed point to six significant digits.

Problem 9.3 (Iterative decoding threshold)

By analysis or simulation, show that the smallest p such that the equation $x = 1-(1-px^2)^5$ has a solution in the interval $0 < x < 1$ is $p^* = 0.429....$ Explain the significance of this calculation for iterative decoding of LDPC codes on a BEC.

Problem 9.4 (Stability condition)

(a) Show that if the minimum left degree of an irregular LDPC code is 3, then the stability condition necessarily holds.

(b) Argue that such a left degree distribution $\lambda(x)$ cannot be capacity-approaching, in view of Theorem 13.2.

Problem 9.5 (Sum-product update rule for zero-sum nodes)

(a) Prove that the algorithm of Section 13.5.2 implements the sum-product update rule for a zero-sum node, up to scale. [Hint: observe that in the product $\prod_j (w_0^{\text{in},j} - w_1^{\text{in},j})$, the terms with positive signs sum to w_0^{out}, whereas the terms with negative signs sum to w_1^{out}.]

(b) Show that if we interchange $w_0^{\text{in},j}$ and $w_1^{\text{in},j}$ in an even number of incoming APP vectors, then the outgoing APP vector $\{w_0^{\text{out}}, w_1^{\text{out}}\}$ is unchanged. On the other hand, show that if we interchange $w_0^{\text{in},j}$ and $w_1^{\text{in},j}$ in an odd number of incoming APP vectors, then the components w_0^{out} and w_1^{out} of the outgoing APP vector are interchanged.

(c) Show that if we replace APP weight vectors $\{w_0, w_1\}$ by log likelihood ratios $\Lambda = \ln w_0/w_1$, then the zero-sum sum-product update rule reduces to the "tanh rule"

$$\Lambda^{\text{out}} = \ln \left(\frac{1 + \prod_j \tanh \Lambda^{\text{in},j}/2}{1 - \prod_j \tanh \Lambda^{\text{in},j}/2} \right),$$

where the hyperbolic tangent is defined by $\tanh x = (e^x - e^{-x})/(e^x + e^{-x})$.

(d) Show that the "tanh rule" may alternatively be written as

$$\tanh \Lambda^{\text{out}}/2 = \prod_j \tanh \Lambda^{\text{in},j}/2.$$