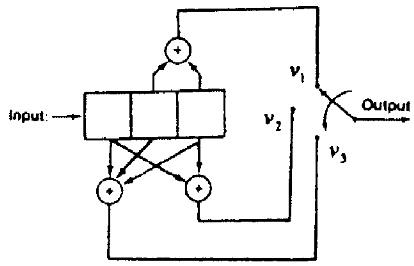
Aalto University
School of Electrical Engineering
Department of Communications and Networking

OPEN-BOOK EXAM
7.3.2012

S-72.3410 Coding Methods



1. Consider the convolutional encoder shown above.

   (a) (3p.) Find the output sequence if the input sequence is (1100101).

   (b) (2p.) Draw the state diagram of the encoder.

   (c) (1p.) Find the minimum free distance d_{free}.

2. (6p.) One generator matrix of a linear block code is

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

   (a) (1p.) Find a systematic generator matrix of the form $\mathbf{G}_{\text{syst}} = [\mathbf{P} \mid \mathbf{I}]$ for this code.

   (b) (2p.) Find a parity-check matrix $\mathbf{H}$ for this code and draw the Tanner graph corresponding to the matrix that you obtained.

   (c) (2p.) Construct a syndrome table for this code.

   (d) (1p.) What is the minimum distance $d_{\min}$ of this code?

3. (a) (3p.) What are the possible dimensions of a binary cyclic code of length 37? What about the possible dimensions of a 16-ary cyclic code of length 37? Justify your answers.

   (b) (3p.) Find the generator polynomial of a two-error-correcting 128-ary cyclic code of length 127. Express the coefficients of the generator polynomial as powers of a primitive element of GF(128). *Hint:* vector space representations of the elements of a certain Galois field may be useful here.

4. **Introduction.** Half-rate invertible block codes can be constructed as follows. First, we need an (n, k) cyclic code with $n - k < k$. If systematic encoding is used, the data digits are at the end of the codeword. Hence, we can shorten the code by first selecting those codewords for which the last $2k - n$ bits are zeros and then deleting these $2k - n$ zeros from each selected codeword. The resulting set of codewords have the length $n - (2k - n) = 2(n - k)$, and the dimension of the new code is $k - (2k - n) = n - k$. It can be shown that this kind of shortened half-rate codes have the desired invertible property: when systematic encoding is used, no two codewords have the same parity-check digits.

   The encoding with the shortened $[2(n - k), n - k]$ code works in the same way as with the original (n, k) code, except that the incoming data block is of length $n - k$ instead of k digits. The generator polynomial $g(x)$ of the original (n, k) code is used. If the data polynomial is $u(x) = u_0 + u_1 x + \cdots + u_{n-k-1} x^{n-k-1}$, then the systematic codeword is, as we know,

$$w(x) = b(x) + x^{n-k} u(x),$$

   where $b(x)$ is the parity-check portion of the codeword and obtained in the usual way. It can be shown that the inversion process is very similar to computing the parity-check digits in the systematic encoding. Namely, if we know $b(x)$, then the corresponding data polynomial is obtained as the remainder when $b(x)x^k$ is divided by $g(x)$.

   Problem. As an application of the theory presented above, let us consider a type-II hybrid ARQ protocol of the kind that was discussed at the end of the last lecture. We assume that the messages are 8 bits long. First, the 8-bit data block is encoded systematically into a 12-bit word P_1 by using as the code $\mathbf{C}_1$ the CRC-4 code whose generator polynomial is $g_1(x) = 1 + x + x^4$. The second code $\mathbf{C}_2$ is taken to be the invertible $(24, 12)$ code obtained by shortening the $(63, 51)$ primitive BCH code, whose generator polynomial $g_2(x)$ has the octal representation 12471 (see e.g. Appendix E of [Wic]). The parity block P_2 is then computed in the usual way of systematic encoding, with P_1 as the input data and $g_2(x)$ as the generator polynomial.

   At the receiving end, for any received 12-bit block $\tilde{P}_2$, the inversion procedure can be carried out producing another 12-bit block $\tilde{P}_1$. Whether the latter block is a codeword in $\mathbf{C}_1$, must then be checked.

   (a) (3p.) If the message polynomial is $m(x) = 1 + x^4 + x^7$ (i.e. the message block is 10001001), find $P_1(x)$ and $P_2(x)$ (i.e., the blocks P_1 and P_2 in the polynomial form).

   (b) (3p.) Try to solve $m(x)$ if $P_2(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + x^9 + x^{11}$.