Hamming Codes
EECS 869: Error Control Coding
Fall 2017

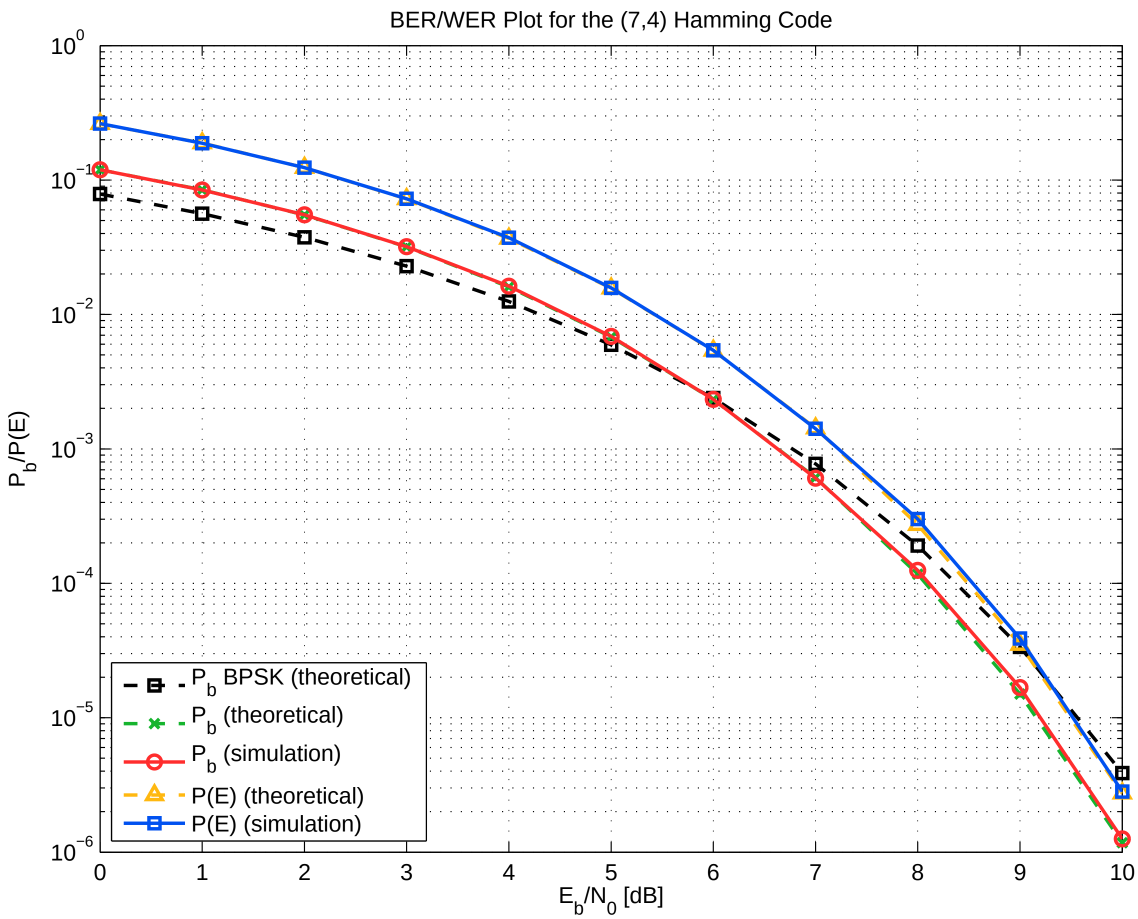
Complete the following tasks. You should submit an e-mail with two .m file attachments and a separate PDF document with your simulation code and the requested plot; e-mail these to `esp@eecs.ku.edu`.

1. **Specify the Generator Matrix and the Parity Check Matrix.** For a linear block code, the generator matrix G has size $k \times n$, where k is the *dimension* of the code and n is the *length* of the code. Hamming codes are defined on p. 97 of the textbook. Your task is to specify the generator matrix G and the parity check matrix H given the integer m. These should be specified in *systematic* form, as defined in the textbook. You should implement this specification as a MATLAB function called `HammingEncodeXXX.m`, where you replace `XXX` with your first name. This MATLAB function should have the following syntax:
   ```
   [G,H] = HammingEncodeXXX(m);
   ```
   In other words, the input argument is some integer $m \geq 2$ and the two output arguments are G and H. You should submit your function via e-mail.

2. **Use the Generator Matrix and the Parity Check Matrix.** You should test your generator matrix and parity check matrix. To generate a codeword $\mathbf{c}$, you take a $1 \times k$ binary message vector $\mathbf{m}$ and perform the operation $\mathbf{c} = \mathbf{m}G$, where all operations are performed modulo-2. You can verify that G is systematic by finding $\mathbf{m}$ in the codeword $\mathbf{c}$. You can also verify that the code subspace is orthogonal to the parity check subspace by checking the fact that $\mathbf{c}H^T = \mathbf{0}$, for any $\mathbf{c}$. You can also verify that $(\mathbf{c} + \mathbf{e})H^T = \mathbf{e}H^T$ as long as the Hamming weight of $\mathbf{e}$ is no more than one, i.e. $w(\mathbf{e}) \leq 1$. You do not have to submit anything for this task.

3. **Decode Hamming Codes with the Hard-Decision Syndrome Decoder.** Implement the syndrome decoding algorithm, which is described on p. 94 of the textbook. You should implement this as a MATLAB function called `HammingDecodeXXX.m`, where you replace `XXX` with your first name. This MATLAB function should have the following syntax:
   ```
   c_hat = HammingDecodeXXX(r,H);
   ```
   where $\mathbf{r} = \mathbf{c} + \mathbf{e}$ is the received codeword, H is the parity check matrix, and $\hat{\mathbf{c}}$ is the output of the decoder. You should submit your function via e-mail.

4. **Generate Bit- and Word-Error-Rate Curves for the (7,4) Hamming Code with Hard-Decision Decoding.** Figure 1.19 in the textbook gives a curve which which can serve as a reference for this task. Modulate the codewords with BPSK and transmit them over the discrete AWGN channel. Make hard decisions on the received symbols before decoding (i.e., convert the received symbols back to ones and zeros, then feed them to the Hamming decoder). Run the simulation for E_b/N_0 in the range $0, 1, \cdots, 8$ dB. Keep track of the bit error rate (BER) for the information bits and also the word error rate (WER) for the codewords. Run the simulation until at least 100 information bit errors are observed and at least 1000000 information bits have been transmitted for each value of E_b/N_0. (To speed things up, you might want to create a more streamlined syndrome decoder that consists of only the syndrome table itself.) Plot your simulation curves together with the theoretical curves for P_b and $P(E)$, along with the theoretical P_b curve for uncoded BPSK. Submit a hardcopy of your simulation code and the plot. You may also submit these in softcopy form, but please provide them in a PDF file, and not as .m files, image files, etc.


BER/WER Plot for the (7,4) Hamming Code
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
$P_b/P(E)$
0
1
2
3
4
5
6
7
8
9
10
E_b/N_0 [dB]
P_b BPSK (theoretical)
P_b (theoretical)
P_b (simulation)
P(E) (theoretical)
P(E) (simulation)