Experiment 4 Hamming encoding and decoding

1. Purpose:

1.1 Understanding Hamming codes.
1.2 Understanding the encoding of Hamming codes.
1.3 Understanding the syndrome decoding of Hamming codes.

2. Principle:

The following formulas may be useful during the experiment.

2.1 Hamming code parameters

Number of parity check bits: *m*

Length of codeword: $n = 2^m - 1$

Number of information bits: $k = n - m = 2^m - 1 - m$

2.2 Encoding by using information sequence *u* and generator matrix *G*

$c = u \bullet G$

2.3 Syndrome decoding

Compute syndrome based on the received codeword sequence: $s = rH^T$

Compute estimated codeword based on the error patter: $c = r + e$

3. Procedure:

You need to build **three modules: Hamming encoder, Hamming decoder and BSC** in this experiment to simulate a basic communication system with channel coding. You first build the three modules one by one and check whether they are correct by using the provided examples. After that, you simulate the system by putting the three modules together.

3.1 Establish the Hamming encoder

The input of the Hamming encoder are ***U*** and m, where ***U*** is the information sequence vector with arbitrary length and *m* is the number parity check bits.

1) Make sure that the length of information sequence ***U*** is a multiple of $k = 2^m - 1 - m$, such that it can be coded with blocks of length *k*. If the length of ***U*** is not a multiple of $k = 2^m - 1 - m$, padding 0s after the information sequence.
2) We now determine the parity-check matrix *H* and generator matrix *G* of the Hamming code with parameter *m*. We can use the Matlab function "[H G] = hammgen(m)" to achieve this. Note that by default, the outputs *H* and *G* have the shape $H = \begin{bmatrix} I_r & P_{r\times k}^T \end{bmatrix}$ and $G = \begin{bmatrix} P_{k\times r} & I_k \end{bmatrix}$,

we need to change them into shape $H=\left[P_{r\times k}^{T} \quad I_{r}\right]$ and $G=\left[I_{k} \quad P_{k\times r}\right]$, respectively.

3) Divide the information sequence $\boldsymbol{U}$ into multiple blocks with length k, code each information sequence $\boldsymbol{u}$ according to $c=u\bullet G$. Concatenate the codewords c to form the output C of the encoder.

Before moving to the Hamming decoder, check whether your Hamming encoder is correct or not. If the input $m=3$ and $u=[1,0,1,1,0,1,0,1,1,1]$, the output should be [1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1].

3.2 Establish the Hamming decoder

We now build the Hamming decoder where its input is the received codeword sequence $\boldsymbol{R}$ with arbitrary length, and m which indicates the codeword length n and number of information bits in each codeword k.

1) Decode the codeword sequence $\boldsymbol{R}$ block by block with a block length of n. We use $\boldsymbol{r}$ to denote the codeword in each block of length n, and k information bits shall be decoded for each block.
2) For each block of length n, first compute its syndrome vector $s=rH^{T}$. Since the Hamming code can only correct 1-bit error, find out which column of the parity-check matrix does the syndrome vector equal to and then determine the error pattern vector E. The index can be obtained by the Matlab function "[, index]=ismember(s, H^T,"rows")". Then, if the syndrome vector is the first column of H, the error pattern is 100…0; If the syndrome vector is the second column of H, the error pattern is 010…0; If the syndrome vector is the last column of H, then the error pattern is 00…01.
3) Using Equation 2.3 to obtain the estimated codeword and extract the first k bits for the decoded information sequence. If the input m=3 and R=[0,0,0,1,1,0,0,0,0,0,1,1,1,0,1,1,0,0,1,0,1], check whether the output is [0,0,0,1,0,1,0,1,1,1,0,0].

3.3 Hamming coding based on BSC

The BSC module can be realized by the Matlab function R = bsc(C, p), where C is the input codeword sequence with arbitrary length, R is the received codeword sequence with same length as C, and p is the crossover probability (error probability) of the BSC.

Now you are ready to perform simulations for a communication system using Hamming codes by connecting the encoder and decoder with the BSC.

4. Report Requirements:

4.1 Assuming that the information bits are i.i.d., you can use the matlab function "(rand(1, 1e7)>*0.5*)" to generate 1e7 information bits. When $m = 3$, simulate and plot the error probability of each codeword against the crossover probability p of the BSC.
What is the theoretical expression of the error probability of each codeword based on m and p? Does it agree with the simulation results?
Now assume that the information bits are not i.i.d., you can use the matlab function "(rand(1,

1e7)>0.2)” to generate 1e7 information bits. What is the error probability? Does it agree with the theoretical expression and why?

4.2 For a fixed crossover probability p of the BSC, simulate and plot the error probability of each codeword against $m = 3, 4, 5, 6, 7$. Comments on your observations.