Exercise 2.4 (a and b) and 2.6a

Table of Contents

2.4

A binary block code has code vectors in systematic form as given in Table P.2.1.

```
clear; clc;
```

Table P.2.1 A block code table

0	0	0	0	0	0
0	1	1	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	0	0	0	1
1	0	1	1	0	1
0	1	1	0	1	1
0	0	0	1	1	1

a) What is the rate of the code?

```
table = [0 0 0 0 0 0; 0 1 1 1 0 0; 1 0 1 0 1 0; 1 1 0 1 1 0; 1 1 0 0 0 1;
    1 0 1 1 0 1; 0 1 1 0 1 1; 0 0 0 1 1 1];

% Since the number of code words is 8 (2^3), the dimension of the code
% subspace is 3,k = 3. n = 6.

codeRate = 3/6
```

```
codeRate =

    0.5000
```

b) Write down the generator and parity check matrices of this code in systematic form.

```
% Since three vectors include the 3x3 identify matrix as the last 3
% elements in the vector, these vectors are used to comprise the generator
% matric G.

G = [0 1 1 1 0 0; 1 0 1 0 1 0; 1 1 0 0 0 1]

% The H matrix is comprised of a kxk identity matrix and the 'P' part of
% the G matrix. The 'P' part of the generator matrix is the left hand side
% of the matrix when considering the right hand side as the identify
% matrix.
```

```
G =

     0     1     1     1     0     0
     1     0     1     0     1     0
     1     1     0     0     0     1
```

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

```
I3 = [1 0 0; 0 1 0; 0 0 1]

P = [0 1 1; 1 0 1; 1 1 0]
```

```
I3 =

     1     0     0
     0     1     0
     0     0     1


P =

     0     1     1
     1     0     1
     1     1     0
```

$$\mathbf{H} = \begin{bmatrix} I_{n-k} & P^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

```
HT = [I3; P.']
```

```
HT =

     1     0     0
     0     1     0
     0     0     1
     0     1     1
     1     0     1
     1     1     0
```

c) What is the minimum Hamming distance of the code?

```
% Since the minimum distance of the codewords is 3:

Dmin = 3
```

```
Dmin =

     3
```

d) How many errors can it correct, and how many can it detect?

```
Ecorrect = floor((Dmin-1)/2)

Edetect = Dmin - 1
```

```
Ecorrect =

     1


Edetect =

     2
```

e) Compute the syndrome vector for the received vector r = (101011) and hence find the location of any error.

```
r = [1 0 1 0 1 1];
```

```
% s = innerProduct(r,HT)
s = mod(r*HT,2) % if this fails try typecast to double
```

```
s =

     1     1     0
```

Location of the error.

```
%    Error patterns  Syndromes
%    0 0 0 0 0 0     0 0 0
%    1 0 0 0 0 0     1 0 0
%    0 1 0 0 0 0     0 1 0
%    0 0 1 0 0 0     0 0 1
%    0 0 0 1 0 0     0 1 1
%    0 0 0 0 1 0     1 0 1
%    0 0 0 0 0 1     1 1 0
```

Given the above correspondance between error patterns and syndrome vectores, the error must be in the sixth position.

2.5

a) Construct a linear block code Cb(5, 2), maximizing its minimum Hamming distance.

```
% Cb(5,2) means 5 columns and 2 rows. Maximum hamming distance is achieved
% by having maximum weight in the parity bits.
P = [1 1 1; 1 0 1]

% Generator matrix
G = [P eye(2)]

% Parity check matrix
H = [eye(3) P.']
```

```
P =

     1     1     1
     1     0     1


G =

     1     1     1     1     0
     1     0     1     0     1


H =

     1     0     0     1     1
```

```
0    1    0    1    0
0    0    1    1    1
```

2.6

A binary linear block code has the following generator matrix in systematic form:

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

a) Find the parity check matrix H and hence write down the parity check equations.

```
% As described above the 'P' matrix is used along with the 3x3 identify
% matrix since the vectors have messages of length 10.

G = [1 1 0 1 1 0 0 1 1 0 1 0 0;
     1 0 1 1 0 1 0 1 0 1 0 1 0;
     1 1 1 0 0 0 1 1 1 1 0 0 1];

P = [1 1 0 1 1 0 0 1 1 0;
1 0 1 1 0 1 0 1 0 1;
1 1 1 0 0 0 1 1 1 1];

I10 = eye(10); % identity matrix size 10x10

HT = [ I10 P.'];
```

Since the resulting matrix cannot fit in the page width, is has been split into a left half and a right half. The left half shows columns 1-7 and the right half shows colums 8-13.

```
HTlefthalf = HT(:,1 : 1 : 7)
HTrighthalf = HT(:,8 : 1 : end)
```

```
HTlefthalf =

     1    0    0    0    0    0    0
     0    1    0    0    0    0    0
     0    0    1    0    0    0    0
     0    0    0    1    0    0    0
     0    0    0    0    1    0    0
     0    0    0    0    0    1    0
     0    0    0    0    0    0    1
     0    0    0    0    0    0    0
     0    0    0    0    0    0    0
     0    0    0    0    0    0    0
```

```
HTrighthalf =

     0     0     0     1     1     1
     0     0     0     1     0     1
     0     0     0     0     1     1
     0     0     0     1     1     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
     1     0     0     1     1     1
     0     1     0     1     0     1
     0     0     1     0     1     1
```

Parity check equations. The codeword bits will be denoted as c0-c9 and message bits m0-m2. The '+' operator used below denotes modulo2 addition.

```
% c0  = m0  +   m1  +   m2
% c1  = m0          +   m2
% c2  =         m1  +   m2
% c3  = m0  +   m1
% c4  = m0
% c5  =         m1
% c6  =         m2
% c7  = m0  +   m1  +   m2
% c8  = m0          +   m2
% c9  =         m1  +   m2
```

b) Find the minimum Hamming distance of the code.

```
% The minimum distance of the block code is equal to the minimum weight of
% the "lightest" codeword. However Matlab has a function that can calculate
% the weight from the generator matrix:

HamDis = gfweight(G)

HamDis =

     7
```

2.7

The generator matrix of a binary linear block code is given below:

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

a) Write down the parity check equations of the code.

```
k = 2;
n = 8;

P = [1 1 0 0 1 1; 0 0 1 1 1 1];

G = [P eye(k)]

H = [eye(n-k) P']
```

```
        G =

             1     1     0     0     1     1     1     0
             0     0     1     1     1     1     0     1


        H =

             1     0     0     0     0     0     1     0
             0     1     0     0     0     0     1     0
             0     0     1     0     0     0     0     1
             0     0     0     1     0     0     0     1
             0     0     0     0     1     0     1     1
             0     0     0     0     0     1     1     1
```

Parity check equations. The codeword bits will be denoted as c0-c5 and message bits m0-m1. The '+' operator used below denotes modulo2 addition.

```
% c0  = m0
% c1  = m0
% c2  =         m1
% c3  =         m1
% c4  = m0  +   m1
% c5  = m0  +   m1
```

b) Determine the code rate and minimum Hamming distance.

```
codeRate = k/n

HamDis = gfweight(G)
```

```
        codeRate =

            0.2500


        HamDis =

             5
```

c) If the error rate at the input of the decoder is 10?3, estimate the error rate at the output of the decoder.

```
p = 10^-3;

t = floor((HamDis-1)/2);

BER = nchoosek(n-1,t)*p^(t+1)
```

```
        BER =

           2.1000e-08
```

2.8

The Hamming block code Cb(15, 11) has the following parity check submatrix:

$$P = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

a) Construct the parity check matrix of the code.

```
P = [0 0 1 1;
    0 1 0 1;
    1 0 0 1;
    0 1 1 0;
    1 0 1 0;
    1 1 0 0;
    0 1 1 1;
    1 1 1 0;
    1 1 0 1;
    1 0 1 1;
    1 1 1 1];

n = 15;
k = 11;

H = [ eye(n-k) P']
```

```
        H =
```

```
  Columns 1 through 13

    1     0     0     0     0     0     1     0     1     1     0     1
    0     1     0     0     0     1     0     1     0     1     1     1
    0     0     1     0     1     0     0     1     1     0     1     1
    0     0     0     1     1     1     1     0     0     0     1     0

  Columns 14 through 15

    1     1
    0     1
    1     1
    1     1
```

b) Construct the error pattern syndrome table.

```
G = [P eye(k)];

HamDis = gfweight(G);

t = floor((HamDis-1)/2)
```

```
t =

     1
```

Since t = 1 the error pattern syndrome table looks like:

```
EPST = [eye(n) H']
```

```
EPST =

  Columns 1 through 13

    1     0     0     0     0     0     0     0     0     0     0     0
    0     1     0     0     0     0     0     0     0     0     0     0
    0     0     1     0     0     0     0     0     0     0     0     0
    0     0     0     1     0     0     0     0     0     0     0     0
    0     0     0     0     1     0     0     0     0     0     0     0
    0     0     0     0     0     1     0     0     0     0     0     0
    0     0     0     0     0     0     1     0     0     0     0     0
    0     0     0     0     0     0     0     1     0     0     0     0
    0     0     0     0     0     0     0     0     1     0     0     0
    0     0     0     0     0     0     0     0     0     1     0     0
    0     0     0     0     0     0     0     0     0     0     1     0
    0     0     0     0     0     0     0     0     0     0     0     1
    0     0     0     0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0
```

```
  Columns 14 through 19

     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
     0     0     0     0     1     1
     0     0     0     1     0     1
     0     0     1     0     0     1
     0     0     0     1     1     0
     0     0     1     0     1     0
     0     0     1     1     0     0
     0     0     0     1     1     1
     0     0     1     1     1     0
     0     0     1     1     0     1
     1     0     1     0     1     1
     0     1     1     1     1     1
```

c) Apply syndrome decoding to the received vector r = (011111001011011).

```
r = [0 1 1 1 1 1 0 0 1 0 1 1 0 1 1];

s = mod(r*H',2) % if this fails try typecast to double
```

```
s =

     0     1     1     0
```

Given a syndrome vector of 0 1 1 0, the error is located at the eigth bit.

Published with MATLAB® R2013b