ECS 452: Digital Communication Systems **2019/2**

HW 6 — Due: April 7, 4 PM

Lecturer: Prapun Suksompong, Ph.D.

Instructions

(a) This assignment has 6 pages.

(b) (1 pt) Work and write your answers **directly on the provided file** (not on other blank sheet(s) of paper).

(c) (1 pt) Write your first name and the last three digits of your student ID in the spaces provided on the upper-right corner of this page. Furthermore, your file name should start with your 10-digit student ID, followed by a space, the course code, a space, and the assignment number: "5565242231 452 HW6.pdf"

(d) (8 pt) Try to solve all non-optional problems.

(e) Late submission will be heavily penalized.

(f) Write down all the steps that you have done to obtain your answers. You may not get full credit even when your answer is correct without showing how you get your answer.

Problem 1. Consider a single-parity-check linear code. For each of the data block below, find the corresponding codeword.

b	x
010	0101
111	1111
001	0011

We simply add one bit to the end to make even number of 1s in the codeword. Note that we use even parity because the code is assumed to be linear.

Problem 2. For each of the codes below, check whether it is a linear code.

(a) $\mathcal{C} = \{\overset{\underline{c}^{(1)}}{000}, \overset{\underline{c}^{(2)}}{001}, \overset{\underline{c}^{(3)}}{100}, \overset{\underline{c}^{(4)}}{101}\}$

$\oplus$	$\underline{c}^{(3)}$	$\underline{c}^{(4)}$
$\underline{c}^{(2)}$	$\underline{c}^{(4)}$	$\underline{c}^{(3)}$
$\underline{c}^{(3)}$		$\underline{c}^{(2)}$

Yes. The sum of any two codewords in the collection is still inside the collection.

We have shown in class that this checking can be performed in two steps:
1) check that the zero codeword is in the collection
2) check that the sum of any distict non-zero codewords in the collection is still inside the collection

(b) $\mathcal{C} = \{000, 100, 110, 111\}$

No. $100 + 110 = 010 \in \mathcal{C}$

(c) $\mathcal{C} = \{001, 100, 101\}$

No. 000 is not a member. Any linear code must contains the zero codeword.

Problem 3. Consider a block code whose generator matrix is

$$\mathbf{G} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}}_{n=6} \Bigg\} k = 3$$

(a) Find the dimension k of this code.

$k = 3$

(b) Find its code rate.

$= \frac{k}{n} = \frac{3}{6}$

(c) Suppose the message is $\underline{\mathbf{b}} = [1\,0\,1]$. Find the corresponding codeword $\underline{\mathbf{x}}$.

There are several equivalent ways to approach this problem.

1) We can simply use

$$\underline{x} = \underline{b}\,G = [1\,0\,1]\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} = [1\ 0\ 1\ 0\ 1\ 1].$$

2) Recall that $\underline{b}\,G = \sum_{j=1}^{n} b_j \underbrace{\underline{g}^{(j)}}_{\text{the } j^{th} \text{ row of } G} = (1 \times \underline{g}^{(1)}) + (0 \times \underline{g}^{(2)}) + (1 \times \underline{g}^{(3)}) = \underline{g}^{(1)} \oplus \underline{g}^{(3)} = [1\ 0\ 1\ 0\ 1\ 1].$

(d) For each of the following vectors, indicate whether it is a valid codeword for this code. If yes, find the message $\underline{\mathbf{b}}$ that produces it. If no, state your reason.

$\underline{x} = \underline{b}\,G = [b_1\ b_2\ b_3]\,G = [b_1, b_2, b_3, b_1 \oplus b_3, b_2 \oplus b_3, b_1 \oplus b_2]$ ← All codewords must satisfy this structure.

In particular, we know that if it is a valid codeword, the first three bits must be $\underline{b}$.

(i) $[0\,1\,1\,1\,0\,1]$

$b_1 = 0$, $b_2 = 1$, $b_3 = 1$

$= b_1 \oplus b_2 = 0 \oplus 1 = 1$

$= b_2 \oplus b_3 = 1 \oplus 1 = 0$

$= b_1 \oplus b_3 = 0 \oplus 1 = 1$

All of these agree with the bits in the given vector. Therefore, yes, the vector is a valid codeword and $\underline{b} = [0\ 1\ 1]$

(ii) $[1\,1\,0\,1\,1\,1]$

$b_1 = 1$, $b_2 = 1$, $b_3 = 0$

$\neq b_1 \oplus b_2 = 1 \oplus 1 = 0$ → This value is not the same as the value in the given vector. Therefore, no, the given vector is not a valid codeword.

$= 1 \oplus 0 = 1$

$= b_1 \oplus b_3 = 1 \oplus 0 = 1$

Problem 4. Consider a block code whose codewords are generated by $\underline{\mathbf{x}} = \underline{\mathbf{b}}\mathbf{G}$ where $\underline{\mathbf{b}}$ is the data block and

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

Let the row vector $\underline{\mathbf{g}}^{(i)}$ represents the ith row of $\mathbf{G}$. Observe that $\underline{\mathbf{g}}^{(3)} = \underline{\mathbf{g}}^{(1)} \oplus \underline{\mathbf{g}}^{(2)}$. Why is this bad?

Both give $\underline{c} = \underline{g}^{(3)}$.

The codewords for 110 and 001 are the same. So, even without bit corruption in the observed vector, the receiver can't distinguish these two cases.

Remark: There are three rows in the generator matrix; hence, k = 3 and each message block has 3 bits. For no ambiguity at the receiver, we should have $2^3 = 8$ distinct codewords. The observation above shows that some different message blocks map to the same codeword. In fact, this code has only 4 distinct codewords.

$\underline{g}^{(1)}, \underline{g}^{(2)}, \underline{g}^{(3)}, \underline{0}$

$\underline{b}$			$\underline{c}$				
0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0
0	1	0	0	1	0	1	1
0	1	1	1	0	1	0	1
1	0	0	1	0	1	0	1
1	0	1	0	1	0	1	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	0	0

Problem 5. Consider each of the block codes whose codebooks are provided below. For each code, is the code a linear code that is generated by a generator matrix? If yes, find the corresponding generator matrix. If no, provide a counter-example to support your conclusion.

(a)

	b_1	b_2	b_3	c_1	c_2	c_3	c_4	c_5	
	0	0	0	0	0	0	0	0	
b_3	0	0	1	0	0	1	1	0	$g^{(3)}$
b_2	0	1	0	1	0	1	0	1	$g^{(2)}$
	0	1	1	1	0	0	1	1	
b_1	1	0	0	1	1	1	0	0	$g^{(1)}$
	1	0	1	1	1	0	1	0	
	1	1	0	0	1	0	0	1	
	1	1	1	0	1	1	1	1	

We read the structure of the bits in the codewords:

$c_1 = b_1 \oplus b_2$ $\quad c_3 = b_1 \oplus b_2 \oplus b_3$

$c_2 = b_1$ $\quad c_4 = b_3$

$c_5 = b_2$

We then check the rest of each column whether all the bits satisfy the above structure or not.

Here, all the bits satisfy the above structure.

Yes, it is a linear code.

$$G = \begin{bmatrix} g^{(1)} \\ g^{(2)} \\ g^{(3)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

(b)

	b_1	b_2	b_3	c_1	c_2	c_3	c_4	c_5	
	0	0	0	0	0	0	0	0	
b_3	0	0	1	0	0	1	1	0	$g^{(3)}$
b_2	0	1	0	1	0	1	0	1	$g^{(2)}$
	0	1	1	1	0	0	1	1	
b_1	1	0	0	1	1	1	0	0	$g^{(1)}$
	1	0	1	1	1	$\boxed{1}$	1	0	
	1	1	0	0	1	0	0	1	
	1	1	1	0	1	1	1	1	

We read the structure of the bits in the codewords:

$c_1 = b_1 \oplus b_2$ $\quad c_3 = b_1 \oplus b_2 \oplus b_3$

$c_2 = b_1$ $\quad c_4 = b_3$

$c_5 = b_2$

We then check the rest of each column whether all the bits satisfy the above structure or not.

This bit does not satisfy $c_3 = b_1 \oplus b_2 \oplus b_3$.

The corresponding message is 101.

So, try $g^{(1)} \oplus g^{(3)}$ which gives $\underbrace{11010}_{\text{not a codeword.}}$

$11100 \oplus 00110$

$\Downarrow$

No, the code is not linear.

This part is not needed for the second conclusion below. We have already shown that the code is not linear. So, it can not be generated by any generator matrix.

Furthermore, if the code is produced by a generator matrix G, then, when

$$\underline{b}^{(1)} = [1\,0\,0] \text{ and } \underline{b}^{(2)} = [0\,1\,0].$$

The corresponding codeword for $\underline{b}^{(1)} \oplus \underline{b}^{(2)}$ must be

$$\underbrace{(\underline{b}^{(1)} \oplus \underline{b}^{(2)})}_{[1\,0\,1]} G = \underbrace{\underline{b}^{(1)} G}_{\text{code word for } \underline{b}^{(1)}} \oplus \underbrace{\underline{b}^{(2)} G}_{\text{code word for } \underline{b}^{(2)}} = 11100 \oplus 00110 = 11010 \neq \underbrace{11110}_{\text{provided in the codebook.}}$$

So, there is no generator matrix that can generate this code.

Problem 6. Consider the following encoding for a systematic linear block code:

- The bit positions that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits.
- The rest (3, 5, 6, 7, 9, etc.) are filled up with the k data bits.
- Each check bit forces the parity of some collection of bits, including itself, to be even.
  - To see which check bits the data bit in position i contributes to, rewrite i as a sum of powers of 2. A bit is checked by just those check bits occurring in its expansion.

This is a general statement about systematic linear block code.

We will consider the case when the codeword's length $n = 7$.

(a) How many bits are check bits?
Hint: How many bit positions are powers of 2?

There are $n=7$ bits in each codeword.

The check bits are defined to be the bits whose positions are powers of 2.

Among the possible positions $(1, 2, 3, \dots, 7)$, three positions $2^0=1$, $2^1=2$, $2^2=4$ are powers of 2.

position of p_1 — position of p_2 — position of p_3

So, there are three check bits. (Note that $k = 7-3 = 4$ bits)

(b) Find the generator matrix **G** for this code.

Let p_1, p_2, p_3 be the check bits and
d_1, d_2, d_3, d_4 be the data bits

Then each codeword is of the form $\underline{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7]$
$= [p_1\ p_2\ d_1\ p_3\ d_2\ d_3\ d_4]$

positions that are powers of 2

For example, d_1 is in the $i = 3^{rd}$ position.
$= 2^0 + 2^1 + 0.$

So, it is used in the eqn. of p_1 and p_2.

Following the encoding instructions, we express the position values in binary

1 0 1 0 1 0 1 $\rightarrow p_1 \oplus d_1 \oplus d_2 \oplus d_4 = 0 \quad \dots (1)$

0 1 1 0 0 1 1 $\rightarrow p_2 \oplus d_1 \oplus d_3 \oplus d_4 = 0 \quad \dots (2)$

0 0 0 1 1 1 1 $\rightarrow p_3 \oplus d_2 \oplus d_3 \oplus d_4 = 0 \quad \dots (3)$

forcing even parity

$p_1 = d_1 \oplus d_2 \oplus d_4$
$p_2 = d_1 \oplus d_3 \oplus d_4$
$p_3 = d_2 \oplus d_3 \oplus d_4$

$$G = \begin{array}{c} x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7 \\ p_1\ p_2\ d_1\ p_3\ d_2\ d_3\ d_4 \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

(c) Find the corresponding parity check matrix **H**.

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

Extra Questions

Here are some optional questions for those who want more practice.

Problem 7 (Carlson and Crilly, 2009, P13.2-1)**.** In mathematical analysis, a function $d(\underline{\mathbf{x}}, \underline{\mathbf{y}})$ is a "true" distance if it satisfies all of the following properties:

(i) positivity: $d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) \geq 0$ with equality if and only if $\underline{\mathbf{x}} = \underline{\mathbf{y}}$

(ii) symmetry: $d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = d(\underline{\mathbf{y}}, \underline{\mathbf{x}})$

(iii) triangle inequality: $d(\underline{\mathbf{x}}, \underline{\mathbf{z}}) \leq d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) + d(\underline{\mathbf{y}}, \underline{\mathbf{z}})$

Is the Hamming distance a "true" distance? (Prove or disprove)

Hint: For the triangle inequality, first consider the number of 1s in $\underline{\mathbf{u}}$, $\underline{\mathbf{v}}$, and $\underline{\mathbf{u}} \bigoplus \underline{\mathbf{v}}$ and confirm that $d(\underline{\mathbf{u}}, \underline{\mathbf{v}}) \leq w(\underline{\mathbf{u}}) + w(\underline{\mathbf{v}})$. Then, from this inequality, replace $\underline{\mathbf{u}}$ by $\underline{\mathbf{x}} \bigoplus \underline{\mathbf{y}}$ and $\underline{\mathbf{v}}$ by $\underline{\mathbf{y}} \bigoplus \underline{\mathbf{z}}$.

First, by definition, we can write $d(\underline{x}, \underline{y}) = w(\underline{x} \oplus \underline{y})$.

(i) Because $d(\underline{x}, \underline{y})$ is the weight of the vector $\underline{x} \oplus \underline{y}$, which is simply counting #1s in $\underline{x} \oplus \underline{y}$, it is always ≥ 0.

Next, suppose $\underline{x} = \underline{y}$. Then, $\underline{x} \oplus \underline{y} = \underline{0}$ and $d(\underline{x}, \underline{y}) = w(\underline{x} \oplus \underline{y}) = w(\underline{0}) = 0$.

suppose $\underline{x} \neq \underline{y}$. Then, there must be at least one position whose corresponding values are different in $\underline{x}$ and $\underline{y}$. This implies there must be at least a 1 in $\underline{x} \oplus \underline{y}$ and hence

$d(\underline{x}, \underline{y}) = w(\underline{x} \oplus \underline{y}) \geq 1 > 0$
$\neq 0$

Therefore, $d(\underline{x}, \underline{y}) = 0$ if and only if $\underline{x} = \underline{y}$.

(ii) Because $\underline{x} \oplus \underline{y} = \underline{y} \oplus \underline{x}$,

$d(\underline{x}, \underline{y}) = w(\underline{x} \oplus \underline{y}) = w(\underline{y} \oplus \underline{x}) = d(\underline{y}, \underline{x})$

(iii) Triangle inequality

First we show that for any pair of vector $\underline{u}$ and $\underline{v}$,

$d(\underline{u}, \underline{v}) \leq w(\underline{u}) + w(\underline{v})$

Recall that $d(\underline{u}, \underline{v}) \equiv w(\underline{u} \oplus \underline{v})$.

and that the XOR operation will give a 1 iff we have $1 \oplus 0$ or $0 \oplus 1$.

Let A be the set of the positions of 1s in $\underline{u}$

$|A| = w(\underline{u})$

Let B be the set of the positions of 1s in $\underline{v}$.

$|B| = w(\underline{v})$

Observe that these areas give the positions of 1s in $\underline{u} \oplus \underline{v}$

Therefore, $w(\underline{u} \oplus \underline{v}) = |\ \ | \leq w(\underline{u}) + w(\underline{v})$ (by comparing the area in the Venn's diagram)

Now, let $\underline{u} = \underline{x} \oplus \underline{y}$ and $\underline{v} = \underline{y} \oplus \underline{z}$.

From the above inequality, we have

$w(\underline{x} \oplus \underbrace{\underline{y} \oplus \underline{y}}_{\underline{0}} \oplus \underline{z}) \leq w(\underline{x} \oplus \underline{y}) + w(\underline{y} \oplus \underline{z})$

(In GF(2), $\underline{v} \oplus \underline{v} = \underline{0}$)

Hence, $d(\underline{x}, \underline{z}) \leq d(\underline{x}, \underline{y}) + d(\underline{y}, \underline{z})$

Problem 8 (Carlson and Crilly, 2009, P13.2-2 and P13.2-3). Consider a block code. Suppose $\underline{\mathbf{x}}$ is the transmitted codeword and that $\underline{\mathbf{y}}$ is the vector that results when $\underline{\mathbf{x}}$ is received with $i > 0$ bit errors. Use the triangle inequality for the Hamming distance to show that

(a) if the code has $d_{\min} \geq \ell + 1$ and if $i \leq \ell$, then the errors are detectable.

Recall that, to detect error(s), we simply check whether the received vector $\underline{y}$ is a valid codeword.

The errors in $\underline{y}$ are detectable iff $\underline{y}$ is not a valid codeword.

Consider any codeword $\underline{c} \in \mathcal{C}$ that is not $\underline{x}$.

From

$$\ell+1 \le d_{min} \le d(\underline{x},\underline{c}) \le d(\underline{x},\underline{y}) + d(\underline{y},\underline{c}) = w(\underline{e}) + d(\underline{y},\underline{c}) \le \ell + d(\underline{y},\underline{c}),$$

given

by definition of being d_{min}

triangle inequality

$\underline{x} \oplus \underline{y} = \underline{e}$

given ($w(\underline{e}) \le \ell$)

we have $d(\underline{y},\underline{c}) \geq 1 > 0$. So, $\underline{y}$ cannot be the same as any codeword in $\mathcal{C}$.

(unless $\underline{y} = \underline{x}$, in which case, there is no error to detect.)

Hence, the errors in $\underline{y}$ are detectable.

(b) if the code has $d_{\min} \geq 2t + 1$ and if $i \leq t$, then the errors are correctable by the minimum distance decoder.

Consider any codeword $\underline{c} \in \mathcal{C}$ that is not $\underline{x}$.

From

$$2t+1 \le d_{min} \le d(\underline{x},\underline{c}) \le d(\underline{x},\underline{y}) + d(\underline{y},\underline{c}) = w(\underline{e}) + d(\underline{y},\underline{c}) \le t + d(\underline{y},\underline{c}),$$

given

by definition of being d_{min}

triangle inequality

$\underline{x} \oplus \underline{y} = \underline{e}$

given ($w(\underline{e}) \le t$)

we have $d(\underline{y},\underline{c}) \geq t+1 > t$

However, $d(\underline{y},\underline{x}) = w(\underline{e}) \le t$. So, $\underline{y}$ is closer to $\underline{x}$ than any other valid codeword.

Hence, when $\underline{y}$ is observed, the min distance decoder will output $\underline{x}$ correcting all the errors in $\underline{y}$.