

6.02 Introduction to EECS II
Spring 2011

Quiz 1

<i>Name</i> SOLUTIONS	<i>Score</i>
---------------------------------	--------------

Please write your answers legibly in the spaces provided. You can use the backs of the pages if you need extra room for your answer or scratch work. Make sure we can find your answer!

You can use a calculator and one 8.5" x 11" cribsheet.

Partial credit will only be given in cases where you show your work and (very briefly) explain your approach.

Problem	Score
#1 (30 points)	
#2 (20 points)	
#3 (50 points)	

Problem 1. Information, Entropy and Huffman Codes (30 points)

There's a weekly surprise party at a local independent living group with an equal probability that the event will happen on any of the seven days.

- (A) (3 points) You learn that party won't be on the weekend, i.e., not Saturday or Sunday. Give an expression for the number of bits of information you have received.

Expression for number of bits of information received: $\log_2(7/5)$

We've gone from $N=7$ equally-probable outcomes down to $M=5$ equally-probable outcomes, so bits of information is $\log_2(N/M)$.

- (B) (4 points) Give an expression for the expected length in bits of a Huffman encoding of a message that lists the day of the party for each week of the 52-week year, i.e., a message consisting of 52 variable-length symbols, where each day is encoded separately using the Huffman code. The choice for each week is independent of the choices for other weeks.

Expression for expected length of message in bits: $52*((1/7)*2 + (6/7)*3)=52*(20/7)$

The Huffman algorithm for 7 equally-probable symbols will build a tree with a depth of 3 for 6 of the symbols and depth of 2 for the seventh symbol.

Examining the historical record, you discover that the probabilities for party days aren't in fact equal – weekends are very popular and the party is never held on Wednesday when 6.02 psets are due. You prepare the following table showing the updated probabilities, which should be used when answering the following questions.

<i>day</i>	<i>Mon</i>	<i>Tue</i>	<i>Wed</i>	<i>Thu</i>	<i>Fri</i>	<i>Sat</i>	<i>Sun</i>
<i>p(day)</i>	0.125	0.125	0	0.125	0.125	0.25	0.25
$\log_2(1/p)$	3	3	--	3	3	2	2
$p*\log_2(1/p)$	0.375	0.375	--	0.375	0.375	0.5	0.5
<i>Encoding from part (C)</i>	101	100	--	001	000	11	01

- (C) (6 points) Using the updated probabilities, create a variable-length Huffman code for sending messages listing party days. Note that no code is required for Wednesday. Please enter the encoding for each day in the last row of the table above.

Fill in last table row

The Huffman algorithm will build a tree where M,Tu,Th,F have a depth of 3 and Sa, Su have a depth of 2. Any code consistent with these constraints is okay as long as none of the encoding is the prefix of another.

- (D) (4 points) Compute the expected length in bits to encode message containing one day using your code from part (C). Please give a numeric answer.

Expected length in bits: 2.5

$$\begin{aligned} \text{Expected length} &= \text{Sum of } p(\text{sym}) * \text{len}(\text{encode}(\text{sym})) \\ &= 0.125 * (3+3+3+3) + 0.25 * (2+2) = 0.125 * 12 + 0.25 * 4 \end{aligned}$$

- (E) (4 points) Using the updated probabilities, compute the entropy of the underlying probability distribution. Please give a numeric answer. Hint: much of the computation has already been performed for you!

Entropy: 2.5

$$\text{entropy} = \text{Sum of } p(\text{sym}) * \log_2(1/p(\text{sym})) = 0.375 * 4 + 0.5 * 2$$

- (F) (4 points) By changing the encoding scheme (say, by encoding pairs of days), would it be possible to improve the expected length of messages? Briefly explain why or why not.

Brief explanation

It's not possible to improve on the expected length of messages by changing the encoding since the expected length of the encoding of part (C) already equals the entropy, which we know is a lower bound on the expected length of messages that deliver the required information.

- (G) (5 points) A phone call from a friend causes you to revise the probabilities for the coming week as follows:

<i>day</i>	<i>Mon</i>	<i>Tue</i>	<i>Wed</i>	<i>Thu</i>	<i>Fri</i>	<i>Sat</i>	<i>Sun</i>
$p(\text{day})$	0.1	0.1	0	0.1	0.1	0.6	0
$\log_2(1/p)$	3.322	3.322	--	3.322	3.322	0.737	--
$p * \log_2(1/p)$	0.332	0.332	--	0.332	0.332	0.442	--

How many bits of information did the phone call deliver? Please give a numeric answer.

Bits of information from phone call: 0.73

Entropy before phone call, from part (E) = 2.5 bits

Entropy after phone call = $4 * .332 + .442 = 1.77$ bits

Information in phone call is given by change in entropy = $2.5 - 1.77$

Problem 2. LZW compression (20 points)

An 8-character message was encoded using the LZW encoder whose pseudo-code is shown below:

```
STRING = get input symbol
WHILE there are still input symbols DO
    SYMBOL = get input symbol
    IF STRING + SYMBOL is in the string table THEN
        STRING = STRING + SYMBOL
    ELSE
        output the code for STRING
        add STRING + SYMBOL to the string table
        STRING = SYMBOL
    END
END
output the code for STRING
```

When the encoding process was complete the following additions had been made to the string table:

```
table[256] = ho
table[257] = oh
table[258] = hoh
table[259] = hoho
```

(A) (10 points) What was the original 8-character message?

Original message: hohohoho

Observe from the pseudo-code that additions to the string table are `STRING + SYMBOL` where the index for `STRING` is what's sent. So simply by stripping the last character from the table entries we can read off all but the last part of the message: h, o, ho, hoh. From the last entry we know that the last symbol group starts with `SYMBOL = o`. Since there are no further entries, that means the message ends with either 'o' or 'oh'. We're told that the message is 8 characters, so the message must have been hohohoho.

(B) (10 points) Recall that the encoder only sends indices into the string table. What indices did the encoder send? Hint: everything can be figured out from the string entries and their order. The index of 'h' is 104 and of 'o' is 111.

Indices sent by encoder: 104, 111, 256, 258, 111

This is what gets transmitted encoding the message from part (A) – the transmitter sends the codes for 'h', 'o', 'ho', 'hoh', 'o'

Problem 3. LTI Models for Communication Channels (50 points)

Consider a communications channel $C1$ that is accurately modeled as a noise-free linear time invariant system with the following causal unit sample response:

$h_{C1}[0]$	$h_{C1}[1]$	$h_{C1}[2]$	$h_{C1}[3]$	$h_{C1}[4]$	$h_{C1}[\geq 5]$
0.0	0.0	1.8	0.5	0.7	0.0

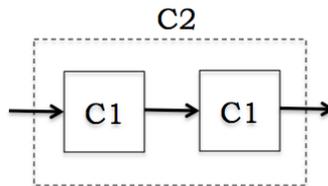
- (A) (4 points) The unit step response for this channel, $s_{C1}[n]$, eventually reaches a steady state value v . What is v and what is the smallest k such that $s_{C1}[k] = v$?

Steady state value v : 3.0

Smallest k : 4

$$s[n] = u[n]*h[n] = [0, 0, 1.8, 2.3, 3.0, 3.0, 3.0, \dots]$$

- (B) (10 points) Suppose we built a communications channel $C2$ composed of two $C1$ channels connected in series:



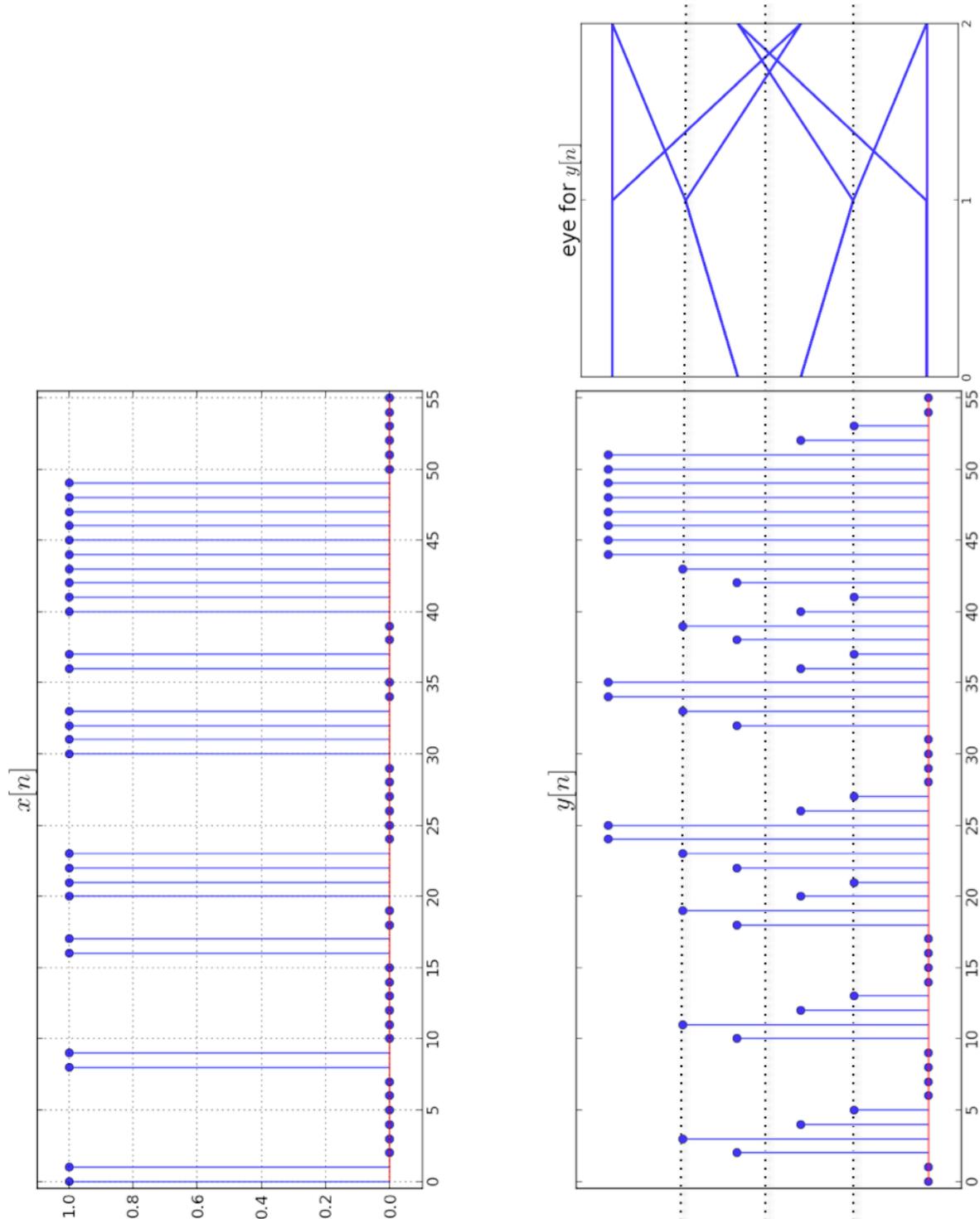
Please fill in the following table, giving the first 10 values of the unit sample response for the $C2$ channel.

Fill in table

$h_{C2}[0]$	$h_{C2}[1]$	$h_{C2}[2]$	$h_{C2}[3]$	$h_{C2}[4]$	$h_{C2}[5]$	$h_{C2}[6]$	$h_{C2}[7]$	$h_{C2}[8]$	$h_{C2}[9]$
0.0	0.0	0.0	0.0	3.24	1.8	2.77	0.7	0.49	0.0

$$\begin{aligned}
 h_{C2}[n] &= h_{C1}[n]*h_{C1}[n] \\
 h_{C2}[4] &= 1.8*1.8 \\
 h_{C2}[5] &= 1.8*.5 + .5*1.8 \\
 h_{C2}[6] &= 1.8*0.7 + .5*.5 + 0.7*1.8 \\
 h_{C2}[7] &= .5*.7 + .5*.7 \\
 h_{C2}[8] &= .7.7
 \end{aligned}$$

Consider digital transmissions over the original channel C1 where we use 2 samples/bit . The following figure shows a test sequence $x[n]$, the channel's response $y[n]$ and an eye diagram constructed from $y[n]$. Assume $x[i] = 0$ for $i < 0$. Note that there are no vertical scales on the plots for $y[n]$ and the eye diagram, but both plots use the same vertical scale (which is *not* the same vertical scale used to plot $x[n]$ – you can't get the answers by measuring!). The receiver will periodically sample $y[n]$ at the widest part of the eye and compare those voltages against a digitization threshold V_{th} to determine the message bits.



- (C) (10 points) What are the possible voltages the receiver will see when it periodically samples $y[n]$ at the widest part of the eye? Since the diagrams have no scale, you will need to compute the voltage values. To receive credit for this part you must *show your work*.

Possible voltage values at sample point: 0.0, 0.7, 2.3, 3.0

Use convolution sum to compute $y[k]$ where $y[k]$ = voltage in eye diagram (avoid $y[0]$ and $y[1]$ since they are due to 2-sample delay in channel)

lowest voltage ($k=6$): $y[6] = 0*x[6] + 0*x[5] + 1.8*x[4] + .5*x[3] + .7*x[2] = 0.0$

next voltage ($k=5$): $y[5] = 0*x[5] + 0*x[4] + 1.8*x[3] + .5*x[2] + .7*x[1] = 0.7$

next voltage ($k=11$): $y[11] = 0*x[11] + 0*x[10] + 1.8*x[9] + .5*x[8] + .7*x[7] = 2.3$

highest voltage ($k=24$): $y[24] = 0*x[24] + 0*x[23] + 1.8*x[22] + .5*x[21] + .7*x[20] = 3.0$

- (D) (6 points) Referring to the figure for $y[n]$, give the first three indices for $y[n]$ where the receiver will sample to determine the first 3 bits of the message.

First index: 3 Second index: 5 Third index: 7

Sample at the widest part of the eye, taking into account 2-sample delay.

- (E) (3 points) Assuming there is an equal probability of sending 0's and 1's, what value of V_{th} will maximize the noise margins at the receiver?

Value of V_{th} : 1.5

Maximize noise margin by choosing voltage a mid-point of eye.

- (F) (3 points) What is the noise margin in volts using your threshold of part (E)?

Noise margin: $2.3 - 1.5 = 0.8$

- (G) (9 points) Since the C1 channel is noise-free (obviously this a work of fiction), it is possible to reliably use deconvolution to construct a perfect estimate, $w[n]$, of the input waveform given $y[n]$ and $h_{C1}[n]$. Give an equation for $w[n]$ where the only variables are from the response ($y[n]$, $y[n-1]$, $y[n+1]$, ...) and earlier values of w ($w[n-1]$, $w[n-2]$, ...), everything else must be numeric. In other words, use numeric values for any h_{C1} elements appearing in the equation.

Give equation for $w[n]$

$$w[n] = (1/1.8) * (y[n+2] - .5*w[n-1] - .7*w[n-2])$$

To eliminate channel delay and ensure a non-zero $h[0]$, we need to shift $h[n]$ and $y[n]$ by 2 to the left, which we can accomplish by adding 2 to their indices in the standard deconvolution equation.

- (H) (5 points) The lecture slides and notes discuss some criteria under which the deconvolution equation will be stable in the presence of noise, i.e., where the estimate $w[n]$ will not grow without bound if some of the $y[n]$ have been affected by noise. Does $h_{C1}[n]$ meet this criteria? Briefly explain.

Brief explanation

The notes say the deconvolution will be stable if $\sum \text{abs}(h[m])/\text{abs}(h[0]) < 1$.

$.5/1.8 + .7/1.8 = 1.2/1.8 < 1$. So $h_{C1}[n]$ meets this criterion.

END OF QUIZ 1!