

Problem Set 5

Your answers will be graded by actual human beings (at least that's what we believe!), so don't limit your answers to machine-gradable responses. Some of the questions specifically ask for explanations; regardless, it's **always a good idea to provide a short explanation for your answer**.

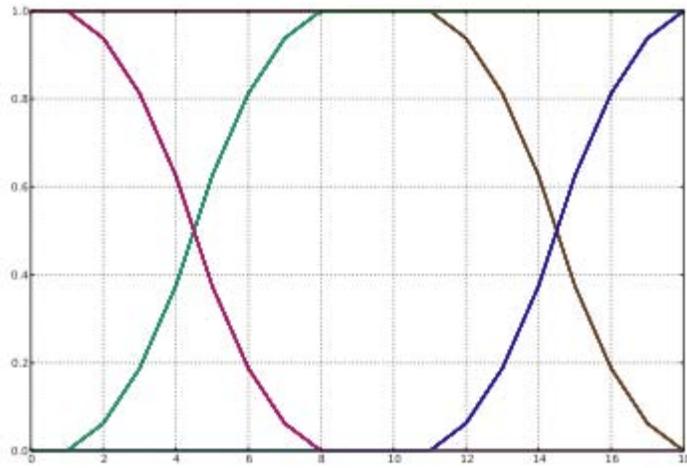
Before doing this PSet, please read Chapters 11 and 12 of the readings. Also attempt the Convolution and Frequency Response practice problems on this material.

Due dates are as mentioned above. Checkoff interviews for PS4 and PS5 will happen together between October 24 and 28, 2012.

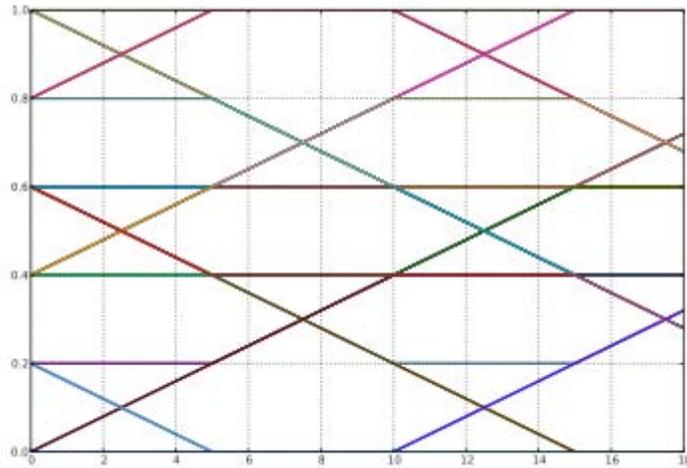
Problem 1. Eye Diagrams

Consider the following three eye diagrams generated by applying a random sequence of 200 bits, with 10 samples/bit, to three different causal LTI systems:

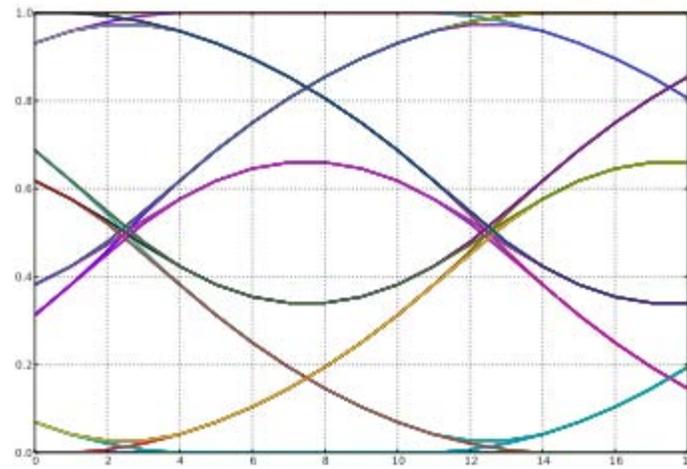
Eye Diagram A



Eye Diagram B

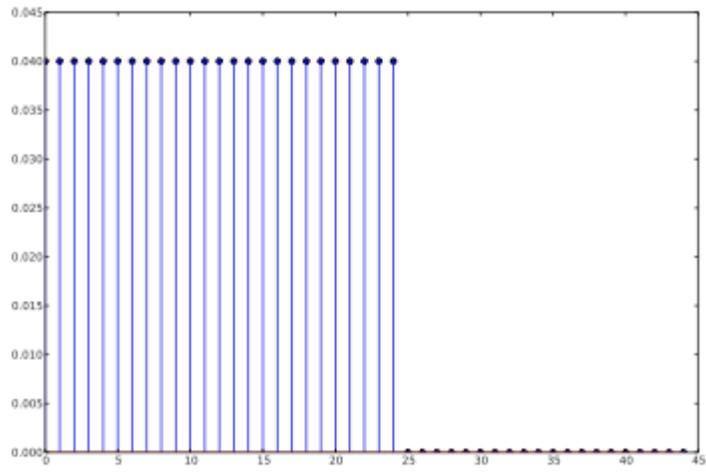


Eye Diagram C

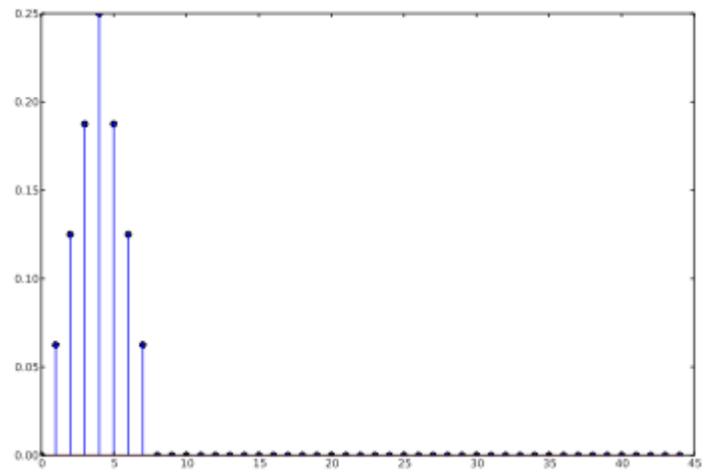


The unit sample responses for each of the three causal LTI systems are given below, in some order:

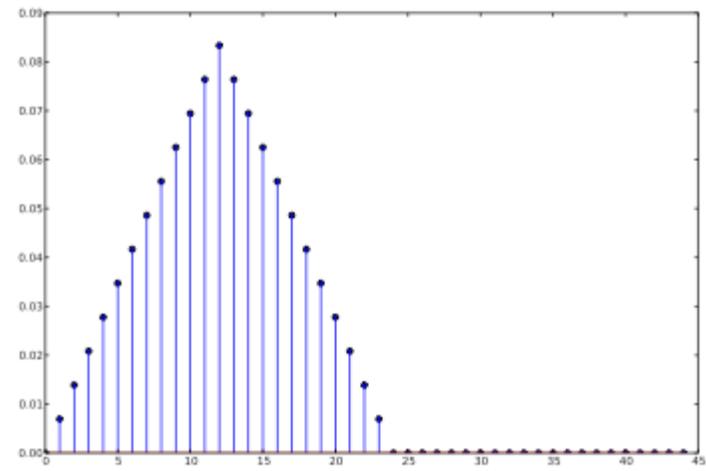
Unit-sample Response 1



Unit-sample Response 2



Unit-sample Response 3



S. For each of the channels 1, 2, and 3, determine the value of the step response at time $n = 4$

(points: 1)

A. Which unit sample response goes with eye diagram A?

(points: 1)

B. Which unit sample response goes with eye diagram B?

(points: 1)

C. Which unit-sample response goes with eye diagram C?

(points: 1)

Problem 2.

Suppose the only nonzero values of the input $x[n]$ to a causal LTI system are:

$$\begin{aligned}x[0] &= 3 \\x[1] &= 2 \\x[2] &= 1 \\x[3] &= -2 \\x[4] &= -1\end{aligned}$$

If the unit sample response of the system is $h[n] = (1/3)^n$ for $n \geq 0$, what is the maximum value of the output $y[n]$, and for what value of n does $y[n]$ achieve its maximum? (Hint for how to do this easily: Sketch $x[\cdot]$ and $h[\cdot]$ and think about the "flip-slide-dot product" implementation of convolution.)

(points: 2)

Problem 3.

Suppose the unit sample response of a causal LTI channel is given by $h[k] = (1/2)^k$ for $k \geq$

0 . The input to the channel is $x[n]$, and the output is $y[n]$.

- A. Find the unit sample response $g[\cdot]$ of a causal LTI filter that the receiver could use, such that applying the channel output $y[n]$ to the input of this filter results in a filter output $z[n]$ that equals the channel input $x[n]$.

(points: 2)

- B. Write down the frequency responses $H(\Omega)$ and $G(\Omega)$.

(points: 2)

Problem 4.

Consider a causal LTI channel with the unit sample response $h[k] = (1/3)^k$ for $k \geq 0$.

Assume the input is $x[n] = \cos(\Omega_0 n + \pi/4)$ for all n .

- A. Determine the frequency response $H(\Omega)$ of the channel.

(points: 1)

- B. Find an expression for $y[n]$ when $\Omega_0 = 0$, and write down the value of $y[3]$.

(points: 1)

- C. Find an expression for $y[n]$ when $\Omega_0 = \pi$, and write down the value of $y[3]$.

(points: 1)

D. Find an expression for $y[n]$ when $\Omega_0 = \pi/2$, and write down the value of $y[3]$.

(points: 2)

Programming Tasks

As in PSet 4, we will use the Audiocom communication system, which uses your computer's speakers to transmit signals, and your computer's microphone to receive signals. In this lab, we will explore the unit-step and unit-sample response of the "baseband-equivalent" channel.

Reminder: The documentation for audiocom is available at <http://audiocom602.blogspot.com>.

Task 1: Bypass Channel

It will help us debug issues over the audio channel if we implement a bypass channel that simulates an LTI channel with delay, lag, and (additive Gaussian) noise.

Go to `bypass_channel.py`, which provides a template class, `BypassChannel`. Notice that its `__init__` method takes three arguments. You already dealt with the first argument, `noise`, in PSet 4. Here, your task is, in addition, to use the specified `lag` and `h` to simulate their effects.

Write `xmit_and_recv`: given the channel's unit-sample response, `self.h`, the number of samples of delay between transmitter and receiver, `self.lag`, and Gaussian noise, `self.noise`, return a numpy array in which each the input numpy array, `tx_samples`, is convolved with `h`, followed by the prepending by `self.lag` samples of value 0 volts, and finally a Gaussian random variable with zero-mean and variance `self.noise` added. Return the resulting numpy array.

Upload your `bypass_channel.py`:

(points: 2)

Task 2: Eye diagrams

In this task, we will be looking at the effects of noise and channel `h` on eye diagrams, as well as their effect on our ability to successfully decode messages sent across a channel.

Before actually making any plots, we will make some changes to make sure we always see an eye diagram, regardless of whether we successfully decode the message. In `receiver.py`, add the following lines before the call to `plot_hist`:

```
p.figure(1)
```

```
plot_eye(demod_samples,spb,'eye diagram')
p.figure(2)
```

Also uncomment lines 80 and 81 in `graphs.py`.

All parts of this problem except 2.5 should be completed using the bypass channel.

Task 2.1: Noise and Eyes

With that out of the way, let's consider the effect of noise on our eye diagrams. Try sending 500 random bits across your bypass channel with a noise variance of 0.1:

```
python sendrecv.py -b -g -n 200 -z 0.1
```

Now repeat the experiment with all parameters the same except noise variance; adjust it to be 0.3, and then 0.5. What effect does the noise variance have on the "openness" of the eye diagrams in this experiment? Why?

(points: 1)

Task 2.2: Noise and SPB

You may have noticed in the previous lab that higher values of samples per bit generally allowed us to more accurately decode our received messages. Let's try keeping noise variance constant at 0.3, and adjusting samples per bit (specified by parameter `-s`). Try with `spb` values 16, 64, 256, and 1024. What effect does the number of samples per bit have on the "openness" of the eye diagrams in this experiment? Why?

(points: 2)

Task 2.3: h and Eyes

In the previous two questions, we have looked at the effect of noise on our eye diagrams. Now we will consider the effect of h , the unit sample response of the channel. In this experiment, we will hold constant both the noise variance (at 0.1) and the samples per bit (at 64). First, let's try sending 200 random bits with the following channel `h`: `-u "0.25 0.25 0.25 0.25"`

How long does it take the unit step response of a system with unit sample response `[0.25,0.25,0.25,0.25]` to converge to 1?

(points: 1)

Now let's repeat the experiment with the following channel USR's (command line arguments have been reproduced to save you some typing):

- -u "0.5 0.5"
- -u "0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1"
- -u "0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05"

What effect does the channel's USR have on the "openness" of the eye diagrams in this experiment? Why?

(points: 1)

Task 2.4: h and SPB

In this section, we will hold noise variance and h constant, and change the number of samples per bit. Try running with the last USR from the previous step (0.05 0.05...), with samples per bit values of 8, 16, 32, 64, and 256.

What effect does the number of samples per bit have on the "openness" of the eye diagrams in this experiment? Why?

(points: 2)

Task 2.5: The Real World

Now try sending 200 random bits across the actual audio channel. Try this with a number of different values of samples per bit.

What effect does the number of samples per bit have on the "openness" of the eye diagrams in this experiment? Why?

(points: 2)

MIT OpenCourseWare
<http://ocw.mit.edu>

6.02 Introduction to EECS II: Digital Communication Systems
Fall 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.