EECS 121
Homework 1

Due: Tue Sep 17 2013

1 Problems

Note: Use a computer wherever you feel the need to.

1. *Review of basic probability:*

   (a) State and prove a weak law of large numbers for binary random variables that are i.i.d.

   (b) What is Bayes' rule?

   (c) What does the central limit theorem assert?

2. *Review of sampling and linear algebra*

   (a) State the traditional Shannon/Nyquist sampling theorem for sampling a continuous-time signal.

   (b) To better understand the linear algebra behind the sampling theorem, let's pretend we live in a toy finite-dimensional world. Consider the vector space C^n as the space of all signals so a signal is an n-dimensional complex vector. Time-domain will be the vectors viewed in the traditional basis $\vec{e}_k = [0, \ldots, 0, 1 \text{ in the } k \text{ position}, 0, \ldots, 0]^T$. Show that the "frequency domain" basis of $\vec{f_k} = \frac{1}{\sqrt{n}}[1, e^{j2\pi\frac{1k}{n}}, e^{j2\pi\frac{2k}{n}}, \ldots, e^{j2\pi\frac{(n-1)k}{n}}]^T$ is an orthonormal basis.

   (c) Continuing the previous part, give an explicit formula for the matrix that maps signals (vectors) in time-domain to frequency domain. Does this have any connection to projections?

(d) Continuing, the counterpart to a "bandlimited" signal is a vector that, when viewed in the frequency-domain basis, is known to have zeros in "high" frequency coordinates. What, intuitively, is the counterpart to the Shannon/Nyquist sampling theorem in such cases? Why is it intuitively plausible?

(e) Continuing, assume $\alpha \in (0, 1)$ and αn is an integer. Given αn regularly-spaced samples (coordinates) of a vector that is known to be bandlimited so that the highest $n - \alpha n$ frequency components are all zero, give a system of linear equations that would enable you to take the samples and solve for the frequency components that are not guaranteed to be zero. Why is this system of linear equations solveable?

3. *Uniform 2D Noise:* Consider the following communication channel,

$$Y_i = X_i + U_i$$

where $X_i \in \mathbb{C}$ represent channel input symbols from the set of complex numbers. U_i is a complex noise variable. The real and imaginary parts of U_i are $Unif[-\sigma, \sigma]$, independent of each other and independent between symbols. Y_i is the received complex symbol. X_i has a peak voltage constraint in each dimension given by $\Re(X_i) \leq \sqrt{E}$ and $\Im(X_i) \leq \sqrt{E}$.

(a) Let n represent the number of bits per symbol we can communicate using this channel with zero probability of error. Find n as a function of E and σ. (You want to put down 2^n points in the complex plane so that noise can't confuse them with each other — in other words you want to put down non-overlapping little squares.)

(b) Calculate n for $\sqrt{E} = 10$ and $\sigma = 0.5$. On the complex plane, plot the 2^n possible positions for X_i (symbols) you have selected to communicate all the possible values for these n bits. This plot is called a symbol constellation.

(c) In a practical communication system the transmitter usually has a peak **energy** constraint instead of a peak voltage constraint. This can be written as $|X_i| \leq \sqrt{E}$. For the values given in Que. 3b recalculate n under the peak energy constraint and plot the corresponding constellation.

(d) In real systems, we often also have constraints on the **average** (as opposed to peak) energy used by the transmitter. Let the bits entering our channel encoder be Bernoulli with parameter 0.4. (they're not fair coin tosses) In channel coding we take incoming bit sequences of length n (as calculated in Que. 3c) and map **all** possible sequences to unique symbols on the constellation of Que. 3c. How would you perform this mapping so as to minimize the *average* energy $E[|X_i|^2]$?

(e) Would your answer to Que. 3d change if they were fair coin tosses? Why?

4. *Calculating Error Probabilities*

Suppose that we have a communication channel with two distinct sources of random variation. There is both additive and multiplicative noise. So $Y = AX + N$ where N is a Gaussian random variable with mean 0 and unit variance. And A is a uniform random variable from $(0, 1)$. The decoder gets access to Y.

Assume that we are using BPSK signaling with energy P. So X is either $+\sqrt{P}$ or $-\sqrt{P}$. Also assume that the bits we are encoding are fair coin tosses.

(a) First, assume that we knew what A was. So we're given that $A = a$. What is your decision rule? What is the probability of error? (feel free to use the Gaussian Q function) Does the rule depend on a? Does the probability of error depend on a?

(b) Write out an integral for the probability of error.

(c) Plot the probability of error numerically as a function of P. Use the style of plot that is most appropriate to show how the probability of error decreases with P.

(d) Comment on this behavior. How would it have changed if N had been Uniform[-1,1] ?

5. *Start hacking* Get started on your project, in toy format.

(a) Write a computer program on your laptop that encodes a binary string into an sequence of tones using OOK. A 1 should correspond

to a 440Hz tone and a 0 should be silence. The duration of the tone should be a quarter of a second.

(b) Write a computer program that takes samples from your microphone, divides them into quarter-second duration chunks, and then decodes each chunk separately into 0s and 1s.

(c) For testing purposes, first use a "virtual noiseless channel" that is perfectly synchronized. (Don't even go through the speaker and microphone, just simulate what would happen if you did it perfectly.) Verify that there are no errors.

(d) Test this in a virtual channel that is perfectly synchronized, but you add iid noise. How does the probability of error change with the noise. Plot it.

(e) Test this using your speaker and microphone, but use whatever aid you'd like to help with initial synchronization.

(f) In your test, plot the measured probability of error as you adjust the volume (keep the distance between the laptops fixed).

(g) Comment on what your group learned from this little test as far as the main project goes.

6. *Write your own* Write your own problem and solution to share with the class.