Homework 3
Due: Thursday, February 10, 2005, at 11:30am

Reading OWN Chapters 2 and 3.

New this week: Please write your section day and time on the upper left of the front page of your homework. This will help us return your homeworks.

You may work in (small) groups to do the homework, but each person must write up their own answers. Note that working together does not mean dividing up the problems and sharing answers later.

For any Matlab problems, submit computer generated plots only. New from this week onwards: **No code is required!**

Niels’ office hours will be rescheduled for the week of Feb 7 only. Check the website for specifics.

Problem 0
Study for Quiz 1! It will be in class on February 16.

Problem 1 (Frequency response.)
Find the outputs when the given inputs are fed into the following systems. Check the newsgroup (ucb.class.ee120) for a hint.

(a) \( H(j\omega) = j\omega \)
   (i) \( x_1(t) = e^{j\pi t} \)
   (ii) \( x_2(t) = 3e^{j2t} - e^{-jt} \)
   (iii) \( x_3(t) = 2\sin(3t + 6) \)

(b) \( h[n] = (\frac{1}{2})^n u[n] \)
   (i) \( x_1[n] = \frac{1}{2}e^{j\frac{\pi}{4} n} \)
   (ii) \( x_2[n] = e^{j\frac{2\pi}{3}(n-2)} - 3e^{-j\frac{2\pi}{3}n} \)
   (iii) \( x_3[n] = 2\cos(\frac{\pi}{2} n + 6\pi) \)

Problem 2 (Discrete time Fourier series.)
Show your work.

(a) OWN 3.28

(b) OWN 3.40
Problem 3 *Multipath in your homemade wireless system.*

You just bought a new laptop which has a wireless card. Unfortunately, you find that you get terrible reception on AirBears, so you decide to use your EE knowledge to create a personal wireless link to your internet connection at home (which is conveniently near Cory on Northside). You’ve read through the FCC’s policies on unlicensed spectrum use and build a legal transmitter and (small) receiver. After taking into account all the issues of sampling, synchronizing the transmitter and receiver, etc., you get the following impulse response for your (LTI) system when you are sitting in the 2nd floor lounge in Cory.

\[ h[n] = \frac{1}{2} \delta[n - 2] + \frac{1}{4} \delta[n - 3] - \frac{1}{8} \delta[n - 4] + \frac{1}{8} \delta[n - 5] \]

Why do you get all these terms when you just sent one pulse? Well, this could be because of multiple paths between your transmitter and your receiver. These various paths also have various attenuations, so you get attenuated and delayed versions of your input. This effect is called multipath, and it is one of the main problems to take into account when designing wireless communications systems. In this problem, you are just thinking of how to use the system and test it. Also, feel free to use Matlab to compute any constants as long as you aren’t having Matlab do the meat of the problem.

(a) Suppose your input is \( x[n] = e^{jw_0n} \), \( w_0 = \frac{\pi}{6} \). What \( y[n] \) do you receive in the 2nd floor lounge in Cory?

(b) Suppose your input is \( x[n] = \cos(w_0n) \), \( w_0 = \frac{\pi}{3} \). What \( y[n] \) do you receive in the 2nd floor lounge in Cory?

(c) Now you are trying to actually use your system. You receive the signal \( y[n] \) below. What was the input \( x[n] \)?

![Diagram of output signal y[n]](attachment:output_signal.png)
Problem 4 (Complex filters.)

Consider the LSI system given by \( y(n) = x(n) * h(n) \) where \( h(n) = (j/4)^n u(n) \). If \( x(n) = e^{j(\pi/6)n} \), draw the block diagram of a system which computes \( y(n) = y_R(n) + jy_I(n) \), using only real signals, adders and multipliers, and delay registers to compute the output pair \( (y_R(n), y_I(n)) \).

Problem 5 (Fourier Series and Gibbs’ Phenomenon - Matlab.)

In this problem we will use Matlab to explore the convergence of Fourier Series and Gibbs’ Phenomenon. One of the main reasons that Fourier series were so controversial when they were discovered by Fourier was that people didn’t believe that an infinite sum of sinusoids could result in discontinuous functions like the square wave. In fact they do converge if the function being approximated satisfies certain conditions (see OWN 3.4), but there is a peculiar behavior that occurs in the convergence at points of discontinuity. This behavior is called Gibbs’ Phenomenon and we will witness it in this problem.

(a) Let \( p(t) \) be periodic with period 1. Then, define it in the unit interval around zero as below.

\[
p(t) = \begin{cases} 
1, & 0 \leq t < \frac{1}{2} \\
0, & -\frac{1}{2} \leq t < 0
\end{cases}
\]

Compute the Fourier coefficients of \( p(t) \), call them \( c_k \).

(b) We want to look at how the partial sum approximation \( p_N(t) = \sum_{k=-N}^{N} c_k e^{j2\pi kt} \) converges as \( N \) goes to infinity. First, let’s see if the Fourier series even starts looking like our function \( p(t) \). Start a new m-file for this problem. Then, create a vector \( k \) with the integers from \(-10\) to \(10\) by typing \( k21 = (-10:10); \) This will create a vector of length \(21\) containing the integers from \(-10\) to \(10\). Now let’s create a time axis. How far apart should our time points be? Well, it depends on the highest frequency component of the signal (this has to do with sampling, you’ll learn about it soon). For this one, we’ll use 201 points in the interval \([-0.5, 0.5]\). You can create the time axis with the command \( t21 = linspace(-0.5,.5,201); \)

Now create a vector of Fourier coefficients. Once you’ve computed them, you can create the vector as follows, with your function being the function you computed.

\[
c21 = \text{yourfunction}(k21)
\]

Take into account that your function probably doesn’t work for \( k = 0 \). So manually enter \( c_0 \) in \( c(11) \) now. Now, you can create \( p_{10}(t) \) with the following code.

\[
y21 = \text{zeros}(1, \text{length}(t21));
\]

for \( i=1: \text{length}(c21) \)
\[
y21 = y21 + c21(i)*\exp(j*2*pi*k21(i).*t21);
\]
end

\[
y21 = \text{real}(y21);
\]

In a perfect world, we wouldn’t need to take the real part of \( y21 \), but because of rounding errors, we have to. Also, create the function \( p(t) \) using the following.

\[
p = (t21 >= 0); \]

Now plot \( p(t) \) and \( p_{10}(t) \) in the same figure.

\[
\text{plot}(t21, y21, 'b-');
\]

hold on;

\[
\text{stairs}(t21, p, 'k-');
\]

hold off;

Of course, you should look at what these functions do and their options using the command \texttt{help}. You should see that your function looks kind of like \( p(t) \), but it overshoots and undershoots a lot.
around \( t = 0 \). Now repeat the above for \( p_{100}(t) \) and \( p_{1000}(t) \). Remember to create new time axes \((t_{201}, t_{2001})\) with smaller spacings so you get a good idea of what’s going on near \( t = 0 \) for these functions.

Plot, in the same figure, \( p(t) \), \( p_{10}(t) \), \( p_{100}(t) \) and \( p_{1000}(t) \). You should now see Gibbs’ phenomenon - the overshoot and undershoot around the discontinuity at \( t = 0 \).

What is the value of your partial sum approximations at \( t = 0 \)? Does it agree with the value of the function at \( t = 0 \)?

(c) What is the approximate value of the maximum overshoot? The maximum overshoot is defined as the maximum value of \( |p(t) - p_N(t)| \) near \( t = 0 \). Does this overshoot keep getting smaller as you increase the number of terms in your sum, or does it stay constant?

(d) Where (at what time) does this maximum overshoot occur for each \( p_N(t) \) that you plotted? What do you notice about the trend of where the maximum overshoot occurs and how many terms are in the sum?