EE102A Winter 2012-13 Pauly
Signal Processing and Linear Systems I

Problem Set #2

Due: Wednesday, Jan 23, 2013 at 5 PM.

1. *Impulse Functions*

   Simplify these expressions:

   (a) $\cos(\pi t)(\delta(t) + \delta(t-1))$

   (b) $\int_{-\infty}^{\infty} \cos(\pi t)(\delta(t) + \delta(t-1))dt$

   (c) $\int_{-\infty}^{\infty} f(t+1)\delta(t-2)\ dt$

   (d) $\int_{-\infty}^{\infty} e^{j\omega T}\delta(t)\ dt$

   (e) $\int_{0}^{\infty} f(t)\left(\delta(t-1) + \delta(t+1)\right)\ dt$

   (g) $\int_{-\infty}^{\infty} f(\tau)\delta(t-1)\delta(t-\tau)d\tau.$

2. *Linearity and Time Invariance*

   State whether the following systems are linear or nonlinear; time invariant or time variant; and why.

   (a) $y(t) = x(t)\sin(\omega t + \phi)$

   (b) $y(t) = tx'(t)$

   (c) $y(t) = 1 + x(t)\cos(\omega t)$

   (d) $y(t) = \cos(\omega t + x(t))$

   (e) $y(t) = \int_{-t}^{t} x(\tau)d\tau$

   (f) $y(t) = x(\sin(t))$

3. *Periodic Signals*

   A periodic signal $x(t)$, with a period T, is applied to a linear, time-invariant system H. Show that the output $y(t)$

   $$y(t) = H(x(t))$$

   is also periodic, with period T.

4. *Sample and hold system.*

   A sample and hold system is a very simple system for reconstructing a signal from its samples. A sample and hold (S/H) system, with sample time h, is described by $y(t) = x(h\lfloor t/h \rfloor)$, where $\lfloor a \rfloor$ denotes the largest integer that is less than or equal to a.

   Sketch an input and corresponding output signal for a S/H, to illustrate that you understand what it does.

   Is a S/H system linear?

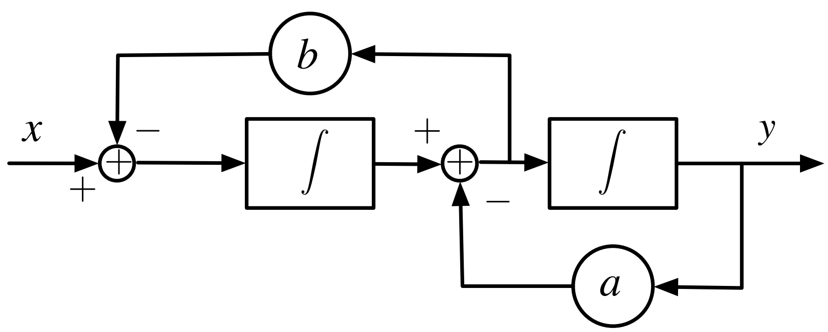
5. Consider a system that takes a signal $x(t)$ and returns the even part of $x(t)$ as it's output

$$x_e(t) = H(x(t))$$

   where $x_e(t)$ is the even part of $x(t)$. Is this system linear? Is it time invariant?

6. *System Equations from Block Diagrams*

   Find the differential equation that correspond to this block diagram. *Hint:* Label all of the intermediate signals, and write equations for each.



7. *Invertible Systems*

   Determine whether the following systems are invertible. If the system is not invertible, find two input signals that produce the same output.

   (a) $y(t) = x(t-1)$
   (b) $y(t) = \displaystyle\int_{-\infty}^{t} x(\tau)d\tau$
   (c) $y(t) = \frac{d}{dt}x(t)$
   (d) $y(t) = x(|t|)$

Laboratory 2

This lab has two components. The first is an introduction to using complex numbers with matlab, and some of the subtle points you need to be aware of, particularly for plotting. The second part gives you several systems for you to characterize, based on their responses to inputs that you choose.

Complex Numbers in Matlab

Matlab deals with complex numbers quite naturally. If you type in

```
>> sqrt(-1)
```

you get the response 0 + 1.0000i. Matlab automatically returns complex values when it is appropriate, even though this may not be what you are expecting.

You enter complex numbers in matlab as the real part plus i times the imaginary part. For example, to enter $1 + 2i$, you would type

```
>> a = 1 + 2*i ;
```

Note that matlab predefines `i` as `sqrt(-1)`. However, `i` is just another matlab variable that you can change. Hence it is generally a bad idea to use `i` for other purposes, such as the loop variable in a `for` loop (which we will introduce later). If you change the value of `i` you can generate bugs that are very hard to find! Matlab also predefines `j` to be `sqrt(-1)`, which you can also change if you aren't careful. The matlab variable `pi` is also predefined, and is changeable.

Matlab provides quite a few different functions for manipulating complex numbers. If `x` is a complex number, `real(x)` returns the real part and `imag(x)` returns the imaginary part. The functions `abs(x)` returns the magnitude of `x`, and `angle(x)` returns the angle in radians. In addition, most functions will take complex arguments, and do the proper thing with them. In particular `exp()` will compute a complex exponential given a complex argument.

Task 1 First compute a vector representing time from 0 to 10 seconds in about 500 steps. Use this to compute a complex exponential with a period of 2 seconds, and a decay rate that reduces the signal level at 10 seconds to half level to half of its original value. What σ and f did you use? You can assume the phase angle θ is zero.

If your complex exponential is `y`, plot

```
>> plot(y);
```

What is matlab doing here? Add a comment to your diary file.

Task 2 Use the `real()` and `imag()` matlab functions to extract the real and imaginary part of the complex exponential, and plot them as a function of time, as we did in the first lab. This should look more reasonable. Label your axes, and check that your signal has the required period and decay rate.

System #	Linear	Time-Invariant	Causal	Memory	Stable	Invertible
1		X	X		X	
2						
3						
4						
5						
6						
7						

Table 1: Test systems, and the properties that hold for them. The first row has been filled in. You have to fill in the rest. Assume the properties are true unless you can find a counterexample.

Task 3 Use the `abs()` and `angle()` functions to plot the magnitude and phase angle of the complex exponential. Scale the `angle()` plot by dividing it by `2*pi` so that it fits well on the same plot as the `abs()` plot (*i.e.* plot the angle in cycles, instead of radians).

Task 4: Black Box Characterization Download the `blackbox.p` function from the course website. It is a content-obscured Matlab file. Make sure you have a copy of the function in your current directory (use command `pwd` to see your current directory and `cd` to change directory).

You are a signal processing engineer at a local Silicon Valley tech company and have been asked to study a software module written by a previous intern. Unfortunately, the company no longer has the documentation or source code for the module, so it operates as a "black box." Your boss recalls that there are 7 separate systems within the module. The module allows you to input a 1 second real-valued, signal sampled every 0.001 seconds into a system, and it returns the output signal over that same time period. Each system assumes that the input is zero both before and after the 1 second input signal, but the output may be non-zero before or after the 1 second (i.e., you only see what happens during that 1 second).

Instead of having you reverse engineer the module, your boss asks you to simply characterize the various system properties by inputting test signals and observing what the output signals are. Assume that all properties are true unless you can find a counterexample.

For the properties that you rule out, provide a plot (or plots) of the counterexample, along with an explanation for why your input(s) and the resulting output(s) conclusively rule out the properties for that system. Finally, describe what you think each system does.

Begin by generating a time sequence.

```
>> dt = .001;
>> t = 0:dt:1;
```

You may want to create a set of standard input functions. For example,

```
>> x0 = zeros(size(t));          % all zero input
```

```
>> xi = [1 zeros(1,length(t)-1)]; % impulse at t = 0
>> xid = [zeros(1,100) xi(1:901)];% delayed impulse at t = 0.1
>> xcos4 = cos(2*pi*4*t);         % cosine with 4 periods
>> xc = .5*ones(size(t));         % constant input  for 1 s
```

Now we characterize the systems found in `blackbox.p`. The inputs to the module are the signal input and system number, and the output is the signal output. Below, we input signal `xcos4` into system 1, and get the output in variable `y`.

```
>> y = blackbox(xcos4, 1);   % output = blackbox(input, systemNumber);
```

Compare the output to the case when we multiply the input by -1.

```
>> yn = blackbox(-xcos4, 1);
>> plot(t, y, t, yn);
```

They are the same. Therefore, linearity does not hold (the output is not multiplied by -1), and the system is not invertible (two different inputs lead to the same output). If we put `xi` or `xid` in we get impulses at the same times, it looks like it doesn't have memory. Delaying the input produces the same delay in the output, so it is time-invarient, and it is later, so it is causal. Since the amplitude of the output is the same as the magnitude of the input, it is bounded-input bounded-output stable.

You don't have enough data to answer all the questions definitively since you only have one second of data, and some of the properties may be difficult to apply in some cases. If you get most of them right, you'll get full credit. The most important characteristics are linearity and time invariance, so focus your attention on these if time is short.

Hints

1. Other inputs may be useful, or combinations of inputs. However, most of the questions can be answered with the test inputs given above, pretty much in the order given.

2. To check for time-invariance, you can delay the input signal by adding zeros at the beginning. See the definitions of the impulse at the origin `xi`, and a delayed impulse `xid` given above for an example.

3. For system #7, what happens if the magnitude of your input stays below 1? What happens if it exceeds 1?