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

Problem Set #2 Solutions

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

1. *Impulse Functions*

Simplify these expressions:

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

Solution: This can be simplified by first multiplying the terms out, and then using the fact that $\phi(t)\delta(t) = \phi(0)\delta(t)$, so

$$\cos(\pi t)\delta(t) + \cos(\pi t)\delta(t-1) = \cos(0)\delta(t) + \cos(\pi)\delta(t-1) = \delta(t) - \delta(t-1)$$

since the first impulses is at 0, and $\cos(0) = 1$, and the second impulse is at π, and $\cos(\pi) = -1$.

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

Solution: From the previous problem, this is

$$\int_{-\infty}^{\infty} \cos(\pi t)(\delta(t) + \delta(t-1))dt = \int_{-\infty}^{\infty} (\delta(t) - \delta(t-1))\, dt = 1 - 1 = 0$$

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

Solution: The impulse is located at $t = 2$, so by the sifting property, the answer is the value of $f(t+1)$ evaluated at $t = 2$

$$\int_{-\infty}^{\infty} f(t+1)\delta(t-2)\, dt = \left. f(t+1)\right|_{t=2} = f(3).$$

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

Solution Since the complex exponential doesn't depend on t, we can factor it out, and

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

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

Solution: The integral is over two impulses, one at $t = 1$ and another at $t = -1$. The limits of integration are from 0 to ∞, so the impulse at $t = -1$ is not included. Then

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

by the sifting property.

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

Solution:

The first impulse is not a function of the integration variable, so it just factors out of the integral,

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

again by the sifting property, and the fact that a function multiplied by an impulse is a scaled impulse.

2. *Linearity and Time Invariance*

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

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

Solultion: Let $x(t) = ax_1(t) + bx_2(t)$, and check

$$\begin{aligned} y(t) &= x(t)\sin(\omega t + \phi) \\ &= (ax_1(t) + bx_2(t))\sin(\omega t + \phi) \\ &= ax_1(t)\sin(\omega t + \phi) + bx_2(t)\sin(\omega t + \phi) \\ &= ay_1(t) + by_2(t) \end{aligned}$$

So, this is a linear system.

If we delay the input $x(t)$ by τ we get

$$x(t-\tau)\sin(\omega t + \phi)$$

However, the delayed output would be

$$x(t-\tau)\sin(\omega(t-\tau) + \phi)$$

which is not the same, so the system is **time variant**.

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

Let $x(t) = ax_1(t) + bx_2(t)$, and check

$$y(t) = tx'(t) = t\frac{d}{dt}(ax_1(t) + bx_2(t)) = atx_1'(t) + btx_2'(t) = ay_1(t) + by_2(t)$$

so this is **linear**.

If we delay the input by τ the output is

$$tx'(t-\tau)$$

which is not the same as delaying the output waveform by τ

$$y(t-\tau) = (t-\tau)x'(t-\tau)$$

so this is **time variant**.

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

Solultion:

We first check homogeneity,

$$1 + (ax(t))\cos(\omega t) = 1 + ax(t)\cos(\omega t) \neq a(1 + x(t)\cos(\omega t))$$

so this is not linear, since scaling the output by a does not result in a scaled output.

The system is **time variant** since

$$1 + x(t-\tau)\cos(\omega t) \neq 1 + x(t-\tau)\cos(\omega(t-\tau)) = y(t-\tau)$$

Delaying the input by τ produces something different than delaying the output by τ.

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

Solution:

We first check homogeneity

$$\cos(\omega t + ax(t)) \neq a\cos(\omega t + x(t))$$

so scaling the input by a does not result in a scaled output. This system is **non-linear**.

Next, if we delay the input by τ

$$\cos(\omega t + x(t-\tau)) \neq \cos(\omega(t-\tau) + x(t-\tau)) = y(t-\tau)$$

so this system is **time variant**.

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

Solultion: This time we'll check superposition and homogeneity together,

$$\int_{-t}^{t}(ax_1(\tau) + bx_2(t))\,d\tau = a\int_{-t}^{t} x_1(\tau)d\tau + b\int_{-t}^{t} x_2(t)d\tau = ay_1(t) + by_2(t)$$

so the system is **linear**.

Next, if we delay the input by T (since τ is already used)

$$\int_{-t}^{t} x(\tau - T)d\tau = \int_{-t-T}^{t-T} x(\tau')d\tau' \neq \int_{-(t-T)}^{(t-T)} x(\tau')d\tau' = y(t-T)$$

where we have made the change of variables $\tau' = \tau - T$. So this system is **time variant**.

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

Solution:

First, this system is linear. With the input $ax_1(t) + bx_2(t)$ we get,

$$ax_1(\sin(t)) + bx_2(\sin(t)) = y_1(t) + y_2(t)$$

It is also time variant. The output is is only over the range of $x(t)$ for $-1 \leq t \leq 1$. If we delay the input by 2, we get a completely different part of $x(t)$

$$x(\sin(t) - 2) \neq y(t-2).$$

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.

   Solution:

   Assume that $y(t) = Hx(t)$. If a system is time invariant, then
   $$y(t-T) = H(x(t-T))$$
   If the input is periodic with period T, then $x(t) = x(t-T)$. If we combine these
   $$y(t-T) = H(x(t-T)) = H(x(t)) = y(t).$$
   So $y(t)$ is periodic with period T.

4. *Sample and hold system.*

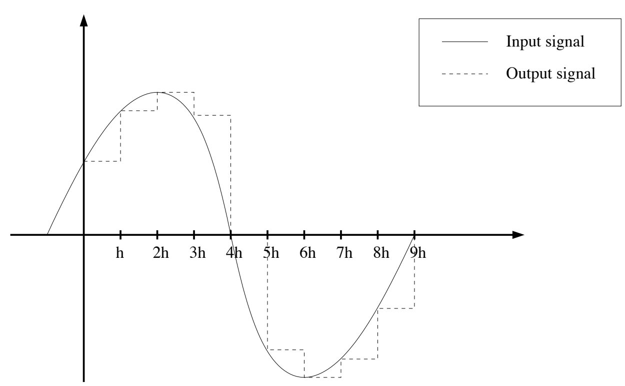
   A sample and hold system is a very simple system for reconstructing a signal from its samples. A sample and hold (S/H) system, with sample time h, is described by $y(t) = x(h\lfloor t/h \rfloor)$, where $\lfloor a \rfloor$ denotes the largest integer that is less than or equal to a.

   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?

   Solution:

   An example of input and corresponding output signal for a S/H is shown below:



   A S/H system obeys superposition because adding two signals and sampling them at fixed times is equivalent to sampling them independently at these same times and adding the sampled values afterwards. Similarly, it obeys homogeneity because scaling a signal before sampling gives the same result as scaling the sampled value. The system is therefore **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?

Solution:

The system is

$$y(t) = \frac{1}{2}\left(x(t) + x(-t)\right)$$

With an input $ax_1(t) + bx_2(t)$,

$$\frac{1}{2}\left((ax_1(t) + bx_2(t)) + (ax_1(-t) + bx_2(-t))\right) = a\frac{1}{2}\left(x_1(t) + x_1(-t)\right) + b\frac{1}{2}\left(x_2(t) + x_2(-t)\right) = y_1(t) + y_2(t)$$

so this is a **linear** system.

If we input $x(t-\tau)$, then the reversed input is $x(-t-\tau)$, and the even part is

$$\frac{1}{2}(x(t-\tau) + x(-t-\tau)) = \frac{1}{2}(x(t-\tau) + x(-(t+\tau))$$

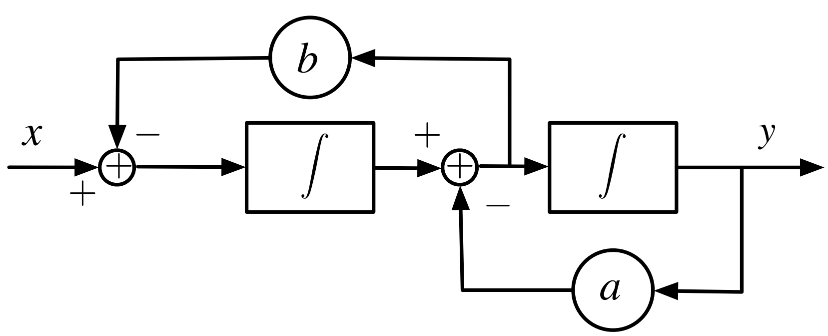
On the other hand, $y(t-\tau)$, the delayed output signal would be

$$y(t-\tau) = \frac{1}{2}(x(t-\tau) + x(-(t-\tau))$$

which is not the same. Delaying the input does not just delay the output, and this system is **time variant**.

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.



Solution

If we label the output of the first integrator to be z, we can write an expression for the input of the second integrator to be

$$y' = z - ay$$

where we have used the fact that the input to an integrator is the derivative of the output (check the class notes if this is not clear). Then, at the input to the first integrator we can write the expression

$$x - by' = z'$$

We need to eliminate z. We solve for z as $y' + ay = z$ in the first equation, differentiate to get $z' = y'' + ay'$, and substitute in the second equation to get

$$x - by' = y'' + ay'$$

Collecting input terms on the right, and outputs on the left, we get

$$y'' + (a + b)y' = x.$$

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) $y(t) = x(t - 1)$

Solution

This is just a delay, so the inverse system is $y(t) = x(t+1)$, which advances the signal.

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

Solution

This is the running integral of a signal. The inverse system is just the derivative

$$y(t) = \frac{d}{dt}x(t).$$

(c) $y(t) = \frac{d}{dt}x(t)$

Solution

This is not invertible, since both $x(t)$ and $x(t) + 1$ have the same output signal.

(d) $y(t) = x(|t|)$

Solution

This system only depends on the input signal for positive time. Any two signals with the same values for positive time will produce the same output, regardless what the signal does for negative time. For example, both e^{-t} and $e^{-t}u(t)$ will produce the same output signal. Hence, this system is not invertable.

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.
Solution: For the decay rate, we want

$$0.5 = e^{-10\sigma}$$

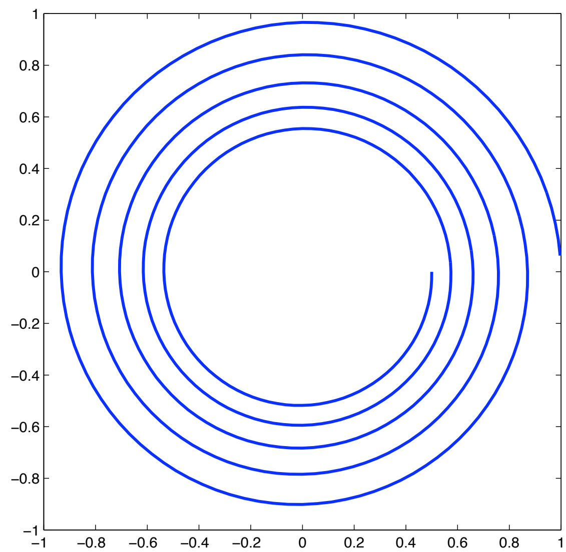
or

$$\sigma = -\frac{\log(0.5)}{10} = 0.06931$$

We want a frequency f of 0.5 Hz to have a 2 second period,

```
>> t = [0:500]/50;
>> y = exp(i*((-0.0693+i*2*pi*0.5)*t);
>> plot(y);
```

The result is plotted below:

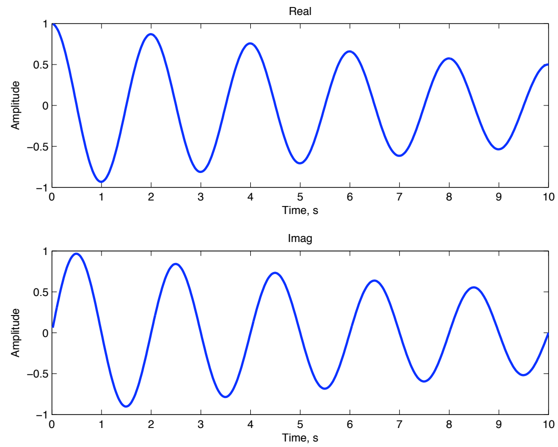


When matlab is given a single complex vector to plot, it plots the real vs imaginary component. This may not be what you expect. This is particularly a problem because matlab automatically converts to complex numbers when it thinks it needs to. Sometime the imaginary component is due to numerical precision errors. You plot what you expect to be a real vector, and you get a wild 2D plot.

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.
Solution: To save paper, we'll put two plots on one page,

```
>> subplot(211)
>> plot(t,real(y))
>> xlabel('Time, s'); ylabel('Amplitude'); title('Real');
>> subplot(212)
>> plot(t,imag(y))
>> xlabel('Time, s'); ylabel('Amplitude'); title('Imag');
```

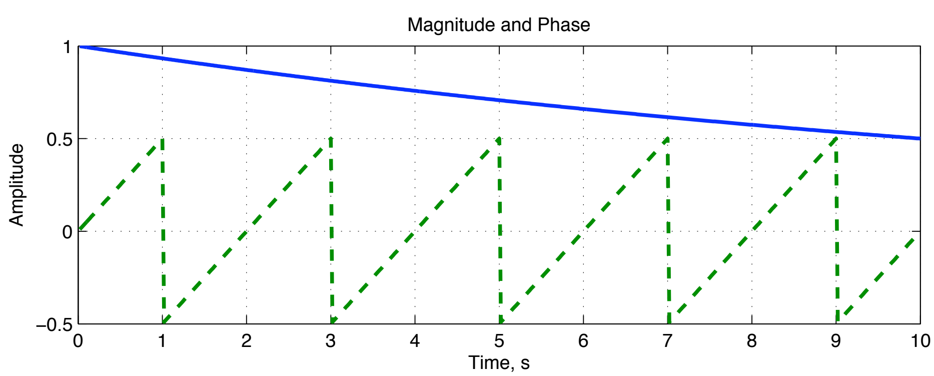
Which results in something more like we expected:



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).
Solution

```
>> plot(t,abs(y),t,angle(y)/(2*pi))
>> ylabel('Amplitude')
>> xlabel('Time, s')
>> title('Magnitude and Phase')
>> grid on
```

The result is



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.

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.

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?

Solution: We will assume the 1 second input time is from $t = 0$ to $t = 1$. For each system we will start with two or three of the inputs described int eh problem, $x_0(t), x_1(t)$, and (if necessary) $x_2(t)$, with corresponding outputs $y_0(t), y_1(t)$, and $y_2(t)$ from `blackbox.p`. While for each system we provide examples that conclusively rule out the properties in the chart, there are many approaches that can be taken for each.

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

Table 2: Here are the properties we identified.

System 2: $y(t) = x(t/3)$. The input is stretched by a factor of 3.

We try three inputs, $x_0(t) = 0, x_1(t) = \delta(t)$, and $x_2(t) = \delta(t - 0.1)$.

```
>> x0 = zeros(size(t)); % all zero input
>> x1 = xi; % impulse at t = 0
>> x2 = [zeros(1,100) xi(1:901)]; % delayed impulse at t = 0.1
```

Input $x_2(t)$ is equivalent to $x_1(t)$ delayed by 0.1 sec, but the output $y_2(t)$ is equivalent to $y_1(t)$ delayed by 0.3 sec, so system 1 is not time invariant. For $t > 0.2$, inputs $x_0(t)$ and $x_2(t)$ are identical, but their outputs are not identical at $t = 0.3$ because of the impulse at $y_2(0.3)$. Therefore, something from before $t = 0.2$ caused the different outputs, so the system has memory.

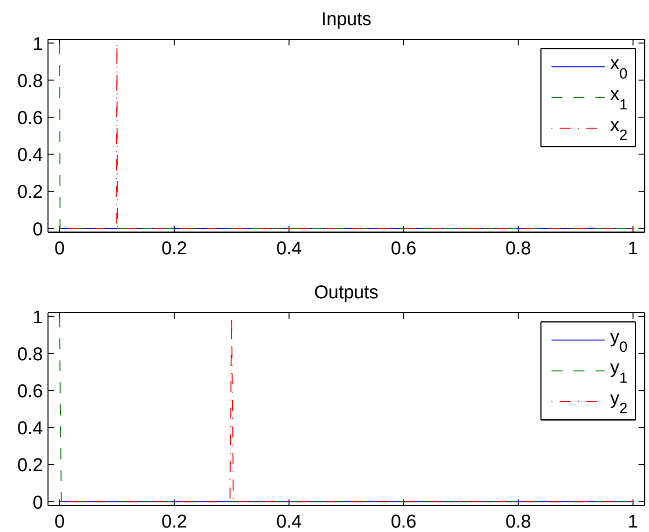


Figure 1: System 2 Inputs and Outputs

System 3: $y(t) = x(t) \cdot \mathrm{sgn}(\sin(10\pi t))$. The input is multiplied by ± 1, alternating every 0.1s.

We try two inputs, the ramp function $x_0(t) = tu(t)$ and the ramp function delayed by 0.25s $x_1(t) = x_0(t - 0.25)$.

```
>> x0 = xr; % ramp
>> x1 = [zeros(1,250) xr(1:751)]; % ramp delayed by 0.25s
```

While input $x_1(t)$ is the delayed version of $x_0(t)$ by 0.25s, it is not the case that $y_1(t)$ is the delayed version of $y_0(t)$ by 0.25s, so the system is not time-invariant.

System 4: $y(t) = \int_{-\infty}^{t} x(\tau) d\tau$. The system integrates the input.

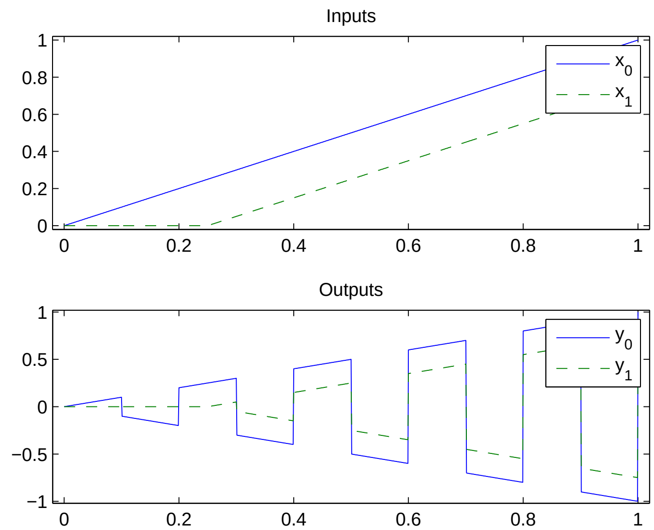


Figure 2: System 3 Inputs and Outputs

We try two inputs, $x_0(t) = 0$ and $x_1(t) = x_s(t)u(0.5 - t)$, a square wave for the first half of the input and zero afterwards.

```
>> x0 = zeros(size(t)); % all zero input
>> x1 = [xs(1:500) zeros(1,501)]; % square wave 1st half, zeros 2nd half
```

The difference in output after t = 0.5s is caused by the difference in input before t = 0.5s. So the output is affected by the past, meaning it has memory. Although an integrator is unstable in the long run, the system cannot be shown to be unstable over this 1 second interval.

System 5: $y(t) = sin(40t^2)$. The output (known as a chirp function) is always the same and independent of the input.

It is clear that the output is always the same. We still need to determine which properties this rules out. We use the cosine function $x_0(t) = \cos(8\pi t)$, half the cosine $x_1(t) = 0.5x_0(t)$, and the cosine delayed by 0.1s $x_2(t) = x_0(t - 0.1)$.

```
>> x0 = xc; % cosine waveform
>> x1 = .5*xc; % half amplitude
>> x2 = [zeros(1,100) xc(1:901)]; % delay by .1 sec
```

From the outputs, which are all the same, we see that halving the input does not halve the output (rules out linearity), that delaying the input does not delay the output (rules out time-invariance), and that all three inputs return the same output so we cannot recover what the input was from the output (rules out invertibility).

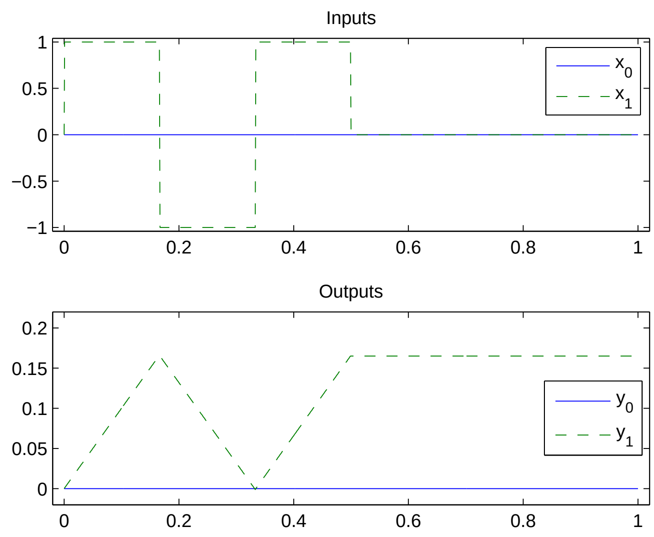


Figure 3: System 4 Inputs and Outputs

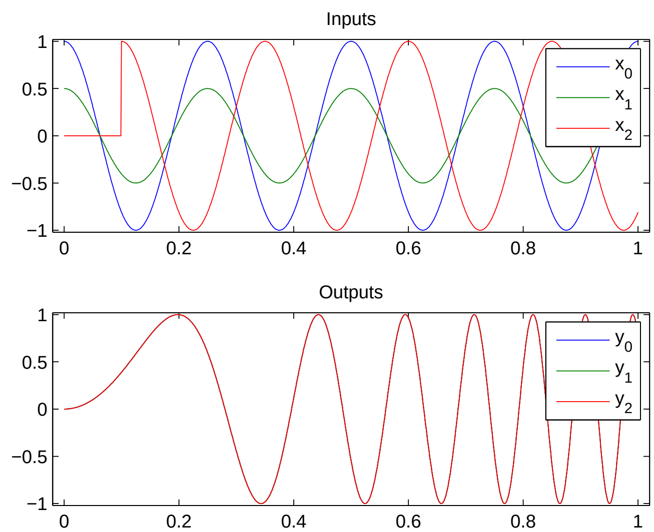


Figure 4: System 5 Inputs and Outputs

System 6: $y(t) = x(1-t)$. The output time-reverses the input.

We use two step functions, one delayed by 0.6s and the other delayed by 0.8s: $x_0(t) = u(t - 0.6), x_1(t) = u(t-0.8)$.

```
>> x0 = [zeros(1,600) xo(1:401)]; % step function delayed by 0.6s
>> x1 = [zeros(1,800) xo(1:201)]; % step function delayed by 0.8s
```

The outputs are the inputs flipped horizontally. Although $x_1(t)$ is $x_0(t)$ delayed by 0.2s, the output $y_1(t)$ is $y_0(t)$ *advanced* by 0.2s (rules out time-invariance). Also, the difference in the outputs from time $0.2 < t < 0.4$ is due to the difference in the inputs from time $0.6 < t < 0.8$. Therefore, the outputs at time $0.2 < t < 0.4$ are somehow affected by the future inputs, ruling out causality. Finally, if a system is not causal, then it is also has memory.

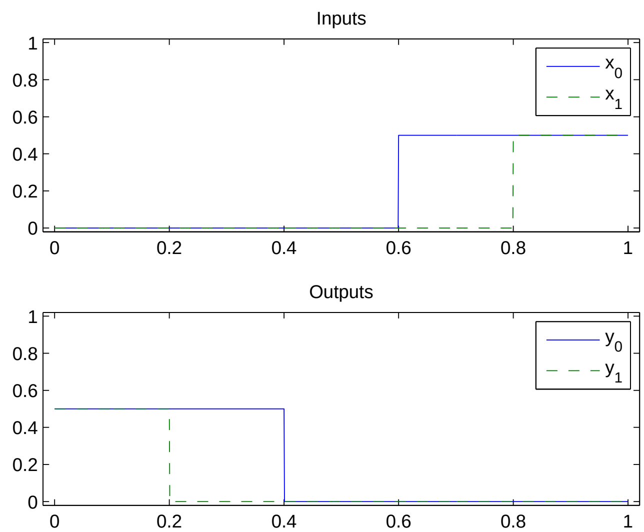


Figure 5: System 6 Inputs and Outputs

System 7: $y(t) = x(t)$ when $t < \tau$, and $y(t) = x(\tau)\exp(2000(t-\tau))$ when $t \geq \tau$, where τ is the smallest time that $|x(t)| > 1$. The output is the same as the input until the input's magnitude exceeds 1, at which point the output blows up. This can occur if a large input triggers an unstable positive feedback loop.

We use two box functions centered at t = 0.5 and width 0.4s but with different heights, 2/3 and 4/3, and a third function that is a step of height 4/3 delayed by 0.4s.

```
>> x0 = 2/3*[zeros(1,300) ones(1,401) zeros(1,300)]; % box fn height 2/3
>> x1 = 2*x0; % box fn from 0.3 to 0.7s, height 4/3
>> x2 = 4/3*[zeros(1,300) ones(1,701)] % step fn height 4/3 delayed 0.3s
```

The first box function emerges unscathed because its maximum amplitude is less than 1. If we double the height to 4/3, we do not get double the output since it blows up to infinity (rules out linearity and stability). The system is not memoryless because in this example, even though the inputs of $x_0(t)$ and $x_1(t)$ for $t > 0.7$ are the same, the outputs for $t > 0.7$ are radically different because of different inputs prior to $t = 0.7$. Once the system blows up, any input does not affect the output, so even though inputs $x_2(t)$ and $x_3(t)$ are different for $t > .7$, they have identical outputs (rules out invertibility).

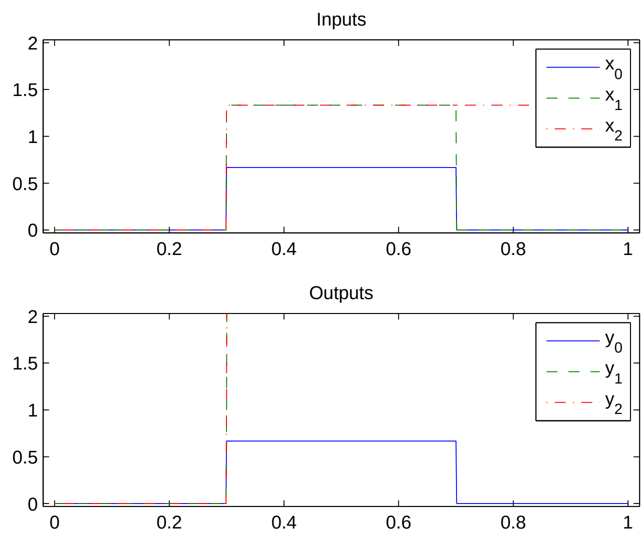


Figure 6: System 7 Inputs and Outputs