

ECE 300
Signals and Systems
Homework 3

Due Date: Friday March 24 at 2:30 PM **Exam #1, Tuesday March 28**

Reading: K & H, pp. 114-128.

Problems

1. For the following system models, determine if the model represents a BIBO stable system. If the system is not BIBO stable, give an input $x(t)$ that demonstrates this.
 - a) $y(t) = \dot{x}(t) + x(t)$
 - b) $y(t) = \sin(x(t))$
 - c) $y(t) = x^2(t)$
 - d) $y(t) + y(t)x(t) = x(t)$

2. For systems modeled by the following equations, determine the systems impulse response, $h(t)$. For simplicity, assume $h(t_0) = 0$.
 - a) $\dot{y}(t) + 2y(t) = x(t)$
 - b) $y(t) = \int_{-\infty}^t e^{-2(t-\lambda)} x(\lambda) d\lambda$

3. The *step response* of a system is the response of the system to a unit step, just as the impulse response of a system is the response to an impulse. Determine the step response of a system with the impulse response $h(t) = e^{-t}u(t-1) + \delta(t-2)$

4. K & H, Problem 3.16 parts **a**, **c**, and **e**. Do these analytically. You only need to plot the results for part **a**.

5. K & H, Problem 3.19. You need to think a little bit here about causality.

6. K & H, Problem 3.31 parts **a** and **b** only. This is pretty easy. You will have a differential equation for $h(t)$ to solve (without Maple, use an integrating factor!)

7. K & H, Problem 3.29. This one is a bit tricky. For part **a**, you need to recognize that the signal just before the integrator is $\frac{dy(t)}{dt}$. Write a differential equation relating $y(t)$ to $x(t)$. Next, let the input be the delta function and the output be the impulse response. You will have a simple differential equation you need to solve (without Maple, use an integrating factor!). You should ultimately get $h(t) = \begin{cases} e^{-t} & 0 \leq t \leq c \\ 0 & t > c \end{cases}$

For part **b**, just use the definition of convolution. You should have one answer for $t \leq c$ and another for $t > c$.

8. (Matlab Problem, read the **Appendix** for help) From the class website download the files **homework3.m** and **convolution.m**. **homework3.m** is a script file that sets up the time arrays and functions, and then invokes the function **convolution.m** to compute the convolution of the two functions. **homework3.m** then plots the two functions to be convolved, and then the resulting convolution of the two functions.

- a) Complete the code for the function **convolution.m**.
- b) Use the script **homework3.m** to compute and plot the convolution of the functions $x_1(t) = e^{-0.5t}u(t)$ and $x_2(t) = 2\Lambda\left(\frac{t-1}{3}\right)$. (These are the default functions in **homework3.m**) If you have done this correctly, your results should like those shown in Figure 2. Turn in your plot.
- c) Use the script **homework3.m** to compute and plot the convolution of the functions $x_1(t) = \text{rect}\left(\frac{t}{2}\right)$ and $x_2(t) = \text{rect}\left(\frac{t}{4}\right)$. Be sure to use reasonable time vectors. Turn in your plot.
- d) For the remainder of this problem, assume we want t_1 to go from -1 to 6 with an increment of 0.01 and t_2 to go from -1 to 8 with an increment of 0.01. Then find the convolution of each of the following:

$$x_1(t) = \text{rect}\left(\frac{t-2.5}{5}\right) \quad x_2(t) = e^{-t}u(t)$$

$$x_1(t) = \text{rect}\left(\frac{t-1}{2}\right) \quad x_2(t) = e^{-t}u(t)$$

$$x_1(t) = \text{rect}\left(\frac{t-0.1}{0.2}\right) \quad x_2(t) = e^{-t}u(t)$$

Turn in your plots and your code. *Note: In this part we can view this as looking at the response of a system with impulse response $h(t) = e^{-t}u(t)$ to inputs which are pulses of decreasing width. This is what we will be doing in Lab 3 when we try to model an impulse, $\delta(t)$, as a narrow pulse and look at how an RC circuit responds.*

Appendix

Although this is a continuous time course, and Matlab works in discrete-time, we can use Matlab to numerically do convolutions, under certain restrictions. The most important restriction is that the spaces between the time samples be the same for both functions. The other restriction is that the functions really need to return to zero (or very close to zero) within the time frame we are examining them.

First, we need to have two functions to convolve. Let's assume we want to convolve the functions $x_1(t) = \text{rect}\left(\frac{t}{2}\right)$ and $x_2(t) = \text{rect}\left(\frac{t-1}{3}\right)$.

Let's denote the time vector that goes with x_1 as t_1 , and the time vector that goes with x_2 as t_2 . Then we create the functions with something like

```
t1 = [-3:0.01:3];
t2 = [-2:0.01:4];
x1 = unit_rect(t1,0,2);
x2 = unit_rect(t2,0,2);
```

Next we will need to determine the time interval between samples. We can determine this as

```
dt = t1(2)-t1(1);
```

It doesn't matter if we use t_2 or t_1 , since the sample interval must be the same.

Now we can use Matlab's **conv** function to do the convolution. However, since we are trying to do continuous time convolution we need to do some scaling:

```
y = dt*conv(x1,x2);
```

Finally, we need to determine a time vector that corresponds to y . To do this we first determine where the starting point should be, which is the sum of the initial times:

```
initial_time = t1(1)+t2(1);
```

Every sample in y is separated by time interval dt . We need to determine how long y is, and create an indexed array of this length

```
n = [0:length(y)-1];
```

Finally we can construct the correct time vector, that starts at the correct initial time and runs the correct length

```
ty = dt*n + initial_time;
```

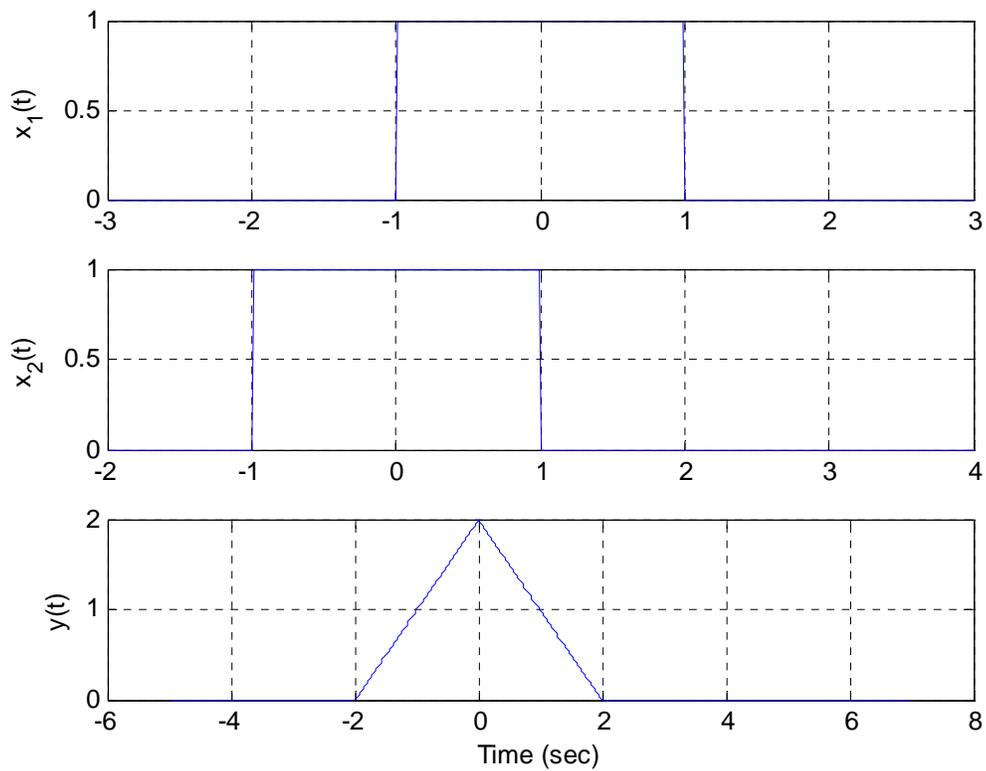


Figure 1: Results for the functions in the **Appendix**

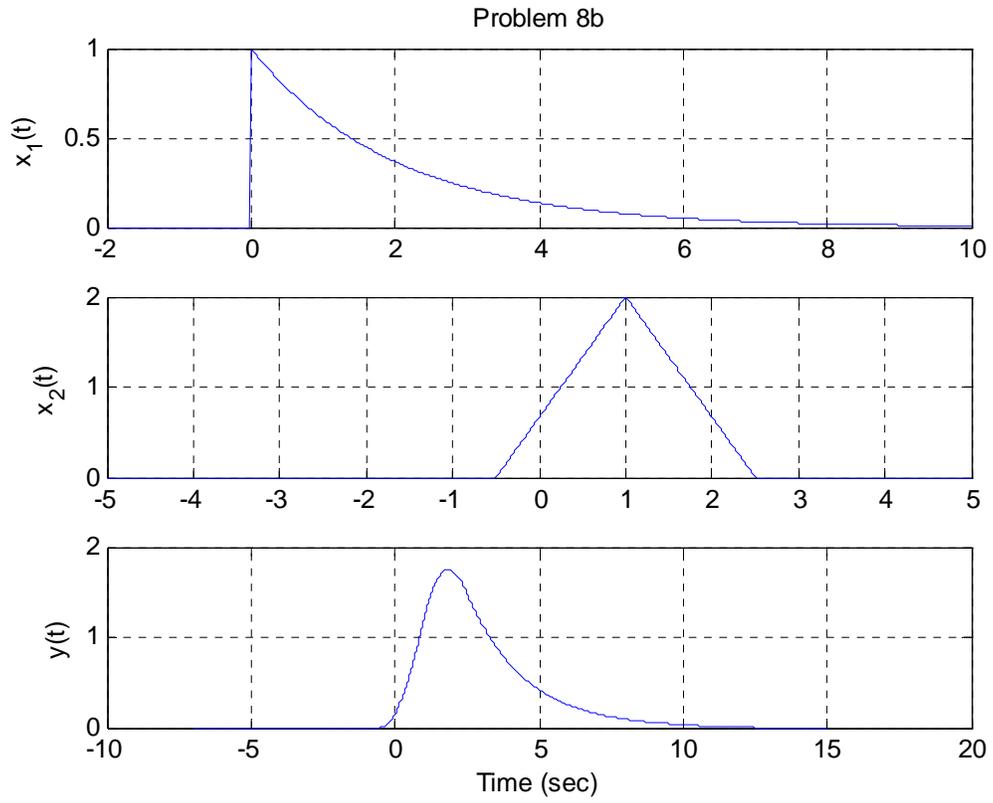


Figure 2: Results for **8-b**.