ECE-205 : Dynamical Systems

Homework #5

Due : Thursday January 14 at the beginning of class

1) Simplify the following expressions as much as possible

a) $g(t) = e^{-t}\delta(t-1)$ **b)** $g(t) = \int_{-2}^{t-3} e^{-(t-\lambda)}\delta(\lambda-1)d\lambda$ **c)** $g(t) = \int_{-2}^{t+2} e^{-(t-\lambda)}\delta(\lambda+1)d\lambda$

d) $g(t) = \int_{t-2}^{2} e^{-(t-\lambda)}\delta(\lambda+1)d\lambda$ **e)** $g(t) = \int_{0}^{t} e^{-(t-\lambda)}\delta(\lambda+1)d\lambda$ **f)** $g(t) = \int_{0}^{t} e^{-(t-\lambda)}\delta(\lambda-2)d\lambda$

g) $g(t) = \int_{-\infty}^{\infty} e^{-3\lambda}\delta(t-\lambda)d\lambda$ **h)** $g(t) = \int_{-\infty}^{\infty} e^{4\lambda}u(\lambda)\delta(t-\lambda+2)d\lambda$ **i)** $g(t) = \int_{-\infty}^{\infty} e^{\lambda}u(\lambda-1)\delta(t-\lambda-3)d\lambda$

Scrambled Answers:

$e^{-(t-1)}u(t-4), 0, e^{-3t}, e^{t-3}u(t-4), e^{4(t+2)}u(t+2), e^{-(t-2)}u(t-2), e^{-(t+1)}u(t+3), e^{-(t+1)}u(1-t), e^{1}\delta(t-1)$

2) Determine the impulse response for each of the following systems

a) $y(t) = 2x(t) + x(t-1)$ **b)** $y(t) = \int_{-\infty}^{t} e^{-2(t-\lambda)}x(\lambda)d\lambda$ **c)** $y(t) = \int_{-\infty}^{\infty} e^{-\lambda}x(t-\lambda)d\lambda$

d) $\dot{y}(t) - \frac{1}{2}y(t) = 2\,x(t)$ **e)** $\dot{y}(t) = 2y(t) + e^{t-1}x(t)$ **f)** $\dot{y}(t) = e^{-t}x(t-1)$

Scrambled Answers: $e^{-1}u(t-1), e^{-t}, e^{-2t}u(t), 2e^{t/2}u(t), e^{2t-1}u(t), 2\delta(t) + \delta(t-1)$

3) For the following two systems,

a) $y(t) = \int_{-\infty}^{t} e^{-(t-\lambda)}x(\lambda+1)d\lambda$ **b)** $y(t) = \int_{-\infty}^{\infty} e^{2\lambda}u(\lambda)x(t-\lambda)d\lambda$

i) compute the impulse response directly

ii) compute the step response directly

ii) show that the derivative of the step response is the impulse response for these two systems. You should get two terms for the derivative, but one of the terms is zero (you should be able to show why this is true)!

Note that this is a general property of LTI systems, and is important since it is much easier to determine the step respones of a system than it is to determine the impulse response of an actual system.

4) For the following inpulse responses and inputs, determine the system output using analytical convolution. Be sure to include any required unit step functions in your answers.

a) $h(t) = e^{-t}u(t)$, $x(t) = u(t-1)$ **b)** $h(t) = e^{-t}u(t) + \delta(t)$, $x(t) = \mathrm{e}^{-(t-1)}u(t-1)$

c) $h(t) = \delta(t) + \delta(t-1)$, $x(t) = \mathrm{e}^{-t}u(t)$ **d)** $h(t) = \delta(t-1)$, $x(t) = \delta(t+2)$

Scrambled Answers $y(t) = \delta(t+1)$, $y(t) = [1 - e^{-(t-1)}]u(t-1)$, $y(t) = te^{-(t-1)}u(t-1)$, $y(t) = e^{-t}u(t) + e^{-(t-1)}u(t-1)$ *:*

5) Matlab From the class website download the files **homework5.m** and **convolve.m**. **homework5.m** is a script file that sets up the time arrays and functions, and the invokes the function **convolve.m** to compute the convolution of the two functions. **homework5.m** then plots the two functions to be convolved, and then the resulting convolution of the two functions. Note that we are passing arguments to the function convolve and the function is returning values to the calling program. In **part c** of this problem you will need to complete this code, *don't try to use it until then.*

a) Show(by performing the convolution) that if $h(t) = e^{-t}u(t)$, $x(t) = e^{-(t-T)}u(t-T)$, then
$y(t) = h(t) \star x(t) = e^{-t}e^{T}(t-T)u(t-T)$

b) Show (by performing the convolution) that if $h(t) = e^{-t}u(t)$, $x(t) = u(t) - u(t-T)$, then
$y(t) = h(t) \star x(t) = [1 - e^{-t}]u(t) - [1 - e^{-(t-T)}]u(t-T)$

c) Read the **Appendix** at the end of this homework and then complete the code for the function **convolve.m.**

d) Use the script **homework5.m** to compute and plot the convolution for the functions of part **a** with $T = 1.5$ (it should default to this). If you have done everything correctly the analytical (true) convolution should be indistinguishable from the estimated (numerical) convolution.

e) Comment out your functions from part **d**, and then modify the code in **homework5.m** to compute the convolution for the functions in part b, with $T = 1$. Be sure to also plot the correct (true) solution to check for any errors.

f) The functions in parts **b** and **e** illustrate a capacitor charging and then discharging, something we have seen before. Change the width of the pulse to the pulse is five time constants long, and run the simulation.

Turn in work for parts a and b, and your plots for parts d and e, and your code.

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. Another 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. Finally, we need fine enough resolution (the sampling interval must be sufficiently small) so that our sampled signals are a good approximation to the continuous signal.

First, we need to have two functions to convolve. Let's assume we want to convolve the functions $h(t) = e^{-t}u(t)$ and $x(t) = e^{-(t-T)}u(t-T)$

Let's denote the time vector that goes with ***h*** as ***th***, and the time vector that goes with ***x*** as ***tx***. Then we create the functions with something like

```
%
tx = [-1:0.001:8]; % time from -1 to 8 with an increment of 0.001 seconds
th = [-1:0.001:8];
%
h = @(t) 0*(t<0)+exp(-t).*(t>=0);
T = 1.5;
x = @(t) 0*(t<T)+exp(-(t-T)).*(t>=T)
```

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

dt = tx(2)-tx(1);

It doesn't matter if we use ***tx*** or ***th***, since the sample interval must be the same.

Now we can use Matlabs **conv** function to do the convolution. However, since we are trying to do continuous time convolution we need to do some scaling. To understand why, let's look again at convolution:

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda$$

If we were to try and approximate this integral using discrete-time samples, with sampling interval Δt, we could write

$$y(t) \approx y(k\Delta t)$$
$$x(\lambda) \approx x(n\Delta t)$$
$$h(t-\lambda) \approx h([k-n]\Delta t)$$

We can then approximate the integral as

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda \approx y(k\Delta t) = \sum_{n=-\infty}^{n=\infty} x(n\Delta t)h([k-n]\Delta t)\Delta t = \Delta t \sum_{n=-\infty}^{n=\infty} x(n\Delta t)h([k-n]\Delta t)$$

The Matlab function **conv** computes the sum $\sum_{n=-\infty}^{n=\infty} x(n\Delta t)h([k-n]\Delta t)$, so to approximate the continuous time integral we need to multiply (or scale) by Δt. Hence

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

Finally, we need to determine a time vector that corresponds to y. To do this we need to determine where the starting point should be. Let's consider the convolution

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda$$

Let's assume $x(t)$ is zero until time t_1, so we could write $x(t) = \tilde{x}(t)u(t-t_1)$ for some function $\tilde{x}(t)$. Similarly, let's assume $h(t)$ is zero until time t_2, so we could write $h(t) = \tilde{h}(t)u(t-t_2)$ for some function $\tilde{h}(t)$. The convolution integral is then

$$y(t) = \int_{-\infty}^{\infty} \tilde{x}(\lambda)u(\lambda-t_1)\tilde{h}(t-\lambda)u(t-\lambda-t_2)d\lambda = \int_{t_1}^{t-t_2} \tilde{x}(\lambda)\tilde{h}(t-\lambda)d\lambda$$

This integral will be zero unless $t-t_2 \geq t_1$, or $t \geq t_1+t_2$. Hence we know that $y(t)$ is zero until $t = t_1 + t_2$. This means for our convolution, the initial time of the output is the sum of the initial times:

```
initial_time = tx(1)+th(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 (ty) that starts at the correct initial time and runs the correct length

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