

**ECE 300**  
**Signals and Systems**  
 Homework 3

**Due Date:** Thursday September 24, 2009 *at the beginning of class*

**Reading** Roberts pages 54-58, 115-140

**Problems**

1) Consider the following mathematical models of systems:

a)  $y(t) = e^{-(t+1)}x(t)$     b)  $y(t) = \int_{-\infty}^t e^{-(t-\lambda)}x(\lambda)d\lambda$     c)  $y(t) = \int_0^t e^{-(t-\lambda)}x(\lambda)d\lambda$

d)  $y(t) = x\left(1 - \frac{t}{2}\right)$     e)  $y(t) = \int_{-\infty}^{t/2} x(\lambda)d\lambda$     f)  $\frac{dy(t)}{dt} = 3y(t) + 2x(t)$

g)  $\frac{dy(t)}{dt} = -2ty(t) + 2x(t)$     h)  $y(t) = \int_{-\infty}^t (t-\lambda)x(\lambda+1)d\lambda$

Fill in the following table (Y or N for each question) for each system. You must justify your answers to receive credit. Assume  $t$  can be any possible value.

Part	Causal?	Memoryless?	Linear?	Time Invariant?
a				
b				
c				
d				
e				
f				
g				
h				

For part f you should show  $y(t) = y(t_0)e^{3(t-t_0)} + \int_{t_0}^t 2e^{3(t-\lambda)}x(\lambda)d\lambda$  in order to

determine the system is or is not causal and has memory or is memoryless.

For part g you should solve the DE first (see handout about integrating factors) and then determine whether the system is or is not causal and has memory or is memoryless.

Remember that we always assume the initial conditions are zero and the initial time is  $-\infty$  when determining time-invariance and linearity.

2) (**Matlab Problem**) The **average value** of a function  $x(t)$  is defined as

$$\bar{x} = \frac{1}{b-a} \int_a^b x(t) dt$$

and the **root-mean-square (rms)** value of a function is defined as

$$x_{rms} = \sqrt{\frac{1}{b-a} \int_a^b x^2(t) dt}$$

Read the **Appendix** (located at the end of this homework), then

a) use Matlab to find the average and rms values of the function  $x(t) = t^2$  for  $-1 < t < 1$

b) use Matlab to find the average and rms values of the following functions

$$x(t) = \cos(t) \quad 0 < t < \pi$$

$$x(t) = \cos(t) \quad 0 < t < 2\pi$$

$$x(t) = |t| \quad -1 < t < 1$$

$$x(t) = t \cos(t) \quad -2 < t < 4$$

Hint: You will probably find the **sqrt** function useful. You should write a Matlab m-file for this problem, and turn it in with your homework, as well as the answers.

3) **Pre-Lab Exercises (to be done by all students. Turn this in with your homework and bring a copy of this with you to lab!)**

a) Calculate the impulse response of the RC lowpass filter shown in Figure 1, in terms of unspecified components R and C. Determine the time constant for the circuit.

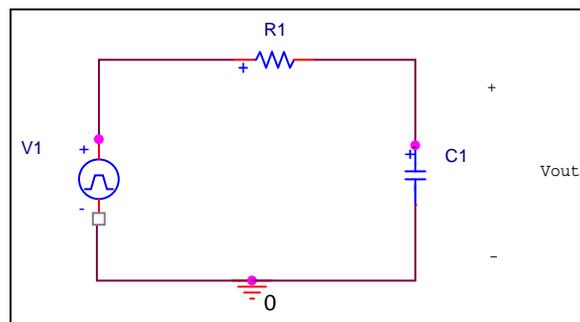


Figure 1. Simple RC lowpass filter circuit.

**b)** Show that the **step response** of the circuit (the response of the system when the input is a unit step) is  $y_s(t) = (1 - e^{-t/\tau})u(t)$ , and determine the 10-90% rise time,  $t_r$ , as shown below in Figure 2. The rise time is simply the amount of time necessary for the output to rise from 10% to 90% of its final value. Specifically, show that the rise time is given by  $t_r = \tau \ln(9)$

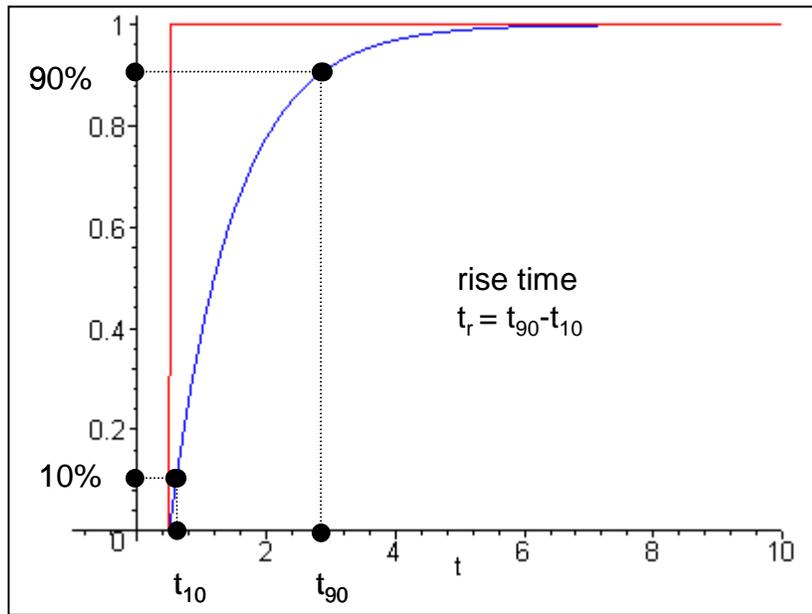


Figure 2. Step response of the RC lowpass filter circuit of Figure 1, showing the definition of the 10-90% risetime.

**c)** Specify values  $R$  and  $C$  which will produce a time constant of approximately 1 msec. Be sure to consider the fact that the capacitor will be asked to charge and discharge quickly in these measurements.

**d) Using linearity and time-invariance**, show that the response of the circuit to a pulse of length  $T$  and amplitude  $A$  (, i.e. a pulse of amplitude  $A$  starting at 0 and ending at  $T$ ) is given by

$$y_{pulse}(t) = A(1 - e^{-t/\tau})u(t) - A(1 - e^{-(t-T)/\tau})u(t - T)$$

**e)** Plot the response to a unit ( $A=1$ ) pulse (in Matlab) for  $\tau = 0.001$  and  $T = 0.003$ , 0.001, and 0.0001 from 0 to 0.008 seconds. Note on the plots the times the capacitor is charging and discharging. Use the **subplot** command to make three separate plots, one on top of another (i.e., use subplot(3,1,1), subplot(3,1,2), subplot(3,1,3)). *Turn in your plots.*

# Appendix

Maple is often used for symbolically integrating a function. Sometimes, though, what we really care about is the numerical value of the integral. Rather than integrating symbolically, we might want to just use numerical integration to evaluate the integral. Since we are going to be using Matlab a great deal in this course, in this appendix we will learn to use one of Matlab's built-in functions for numerical integration. In order to efficiently use this function, we will learn how to construct what are called *anonymous* functions. We will then use this information to determine the average and rms value of a function. Some of this is going to seem a bit strange at first, so just try and learn from the examples.

**Numerical Integration in Matlab** Let's assume we want to numerically integrate the following:

$$I = \int_0^{2\pi} (t^2 + 2)dt$$

In order to do numerical integration in Matlab, we will use the built-in command **quadl**. The **arguments** to quadl, e.g., the information passed to quadl, are

- A function which represents the integrand (the function which is being integrated). Let's call the integrand  $x(t)$ . This function must be written in such a way that it returns the value of  $x(t)$  at each time  $t$ . Clearly here  $x(t) = t^2 + 2$
- The lower limit of integration, here that would be 0
- The upper limit of integration, here that would be  $2\pi$

Note that an optional fourth argument is the tolerance, which defaults to  $10^{-6}$ . When the function value is very small, or the integration time is very small, you will have to change this.

**Anonymous Functions** Let's assume we wanted to use Matlab to construct the function  $x(t) = t^2 + 2$ . We can do this by creating what Matlab calls an **anonymous function**. To do this, we type into Matlab

```
x = @(t) t.*t+2;
```

If we want the value of  $x(t)$  at  $t = 2$ , we just type `x(2)`

Hence, to evaluate the integral  $I = \int_0^{2\pi} (t^2 + 2)dt$  in Matlab we would type

```
x = @(t) t.*t+2;  
I = quadl(x,0,2*pi)
```

Note that it is important to define x **before** it is used by (passed to) `quadl`

**Example 1** To numerically evaluate  $I = \int_{-1}^1 e^{-t} \cos(2t) dt$  we could type

```
x = @(t) exp(-t).*cos(2*t);  
I = quadl(x,-1,1);
```

**Example 2** To numerically evaluate  $I = \int_{-2}^1 |t| e^{-|t|} dt$  we could type

```
y = @(t) abs(t).*exp(-abs(t));  
I = quadl(y,-2,1);
```

**Integrating Products of Functions** Sometimes we are going to want to integrate the product of functions. While we could just multiply the functions together, it is usually easier to let Matlab do it for us.

Let's assume we want to evaluate the integral  $I = \int_0^1 x(t)y(t)dt$ , and let's assume that we already have anonymous functions `x` and `y`. The function **`quadl`** needs to be passed a function which is the product of `x` and `y`. To do this, we make a new anonymous function `z`, using the following:

```
z = @(t) x(t).*y(t);
```

and then perform the integration

```
I = quadl(z,0,1)
```

An alternative is to write

```
I = quadl(@(t) x(t).*y(t),0,1);
```

**Example 3** To numerically evaluate  $I = \int_{-1}^1 e^{-t} \cos(2t) dt$  we could type

```
x = @(t) exp(-t)  
y = @(t) cos(2*t);  
z = @(t) x(t).*y(t);  
I = quadl(z,-1,1);
```

or

```
I = quadl(@(t) x(t).*y(t),-1,1);
```

**Example 4** To numerically evaluate  $I = \int_{-2}^1 |t| e^{-|t|} dt$  we could type

```
x = @(t) abs(t);  
y = @(t) exp(-abs(t));  
z = @(t) x(t).*y(t);  
I = quadl(z,-2,1);
```

or

```
I = quadl(@(t) x(t).*y(t),-2,1);
```