

ECE 300
Signals and Systems
 Homework 2

Due Date: Thursday September 14 at 2:30 PM (Next one Tuesday)

Reading: K & H, pp. 21-37 (skip the discrete-time stuff).

Problems

1. Consider the following system representations:

- a)** $y(t) = e^{x(t)}$ **b)** $y(t) = [\sin(t)]x(t)$ **c)** $\frac{dy(t)}{dt} = y(t)x(t)$ **d)** $\frac{dy(t)}{dt} = -ay(t) + bx(t)$
- e)** $y(t) = \sin(x(t))$ **f)** $y(t) = x(t-1)$ **g)** $y(t) = x\left(\frac{t}{3}\right) + 2$
- h)** $y(t) = e^t \int_{-\infty}^t e^{-\lambda} x(\lambda - c) d\lambda, c > 0$ **i)** $y(t) = \int_0^t \lambda x(\lambda) d\lambda \quad t > 0$

For these system models, fill in the following table (or a similar table) to summarize your results (put a Y or N for each question).. You need to justify your answers, For **a** and **e**, look at the case when $x(t)$ is large, and then when $x(t)$ is assumed to be sufficiently small that you can use a Taylor series approximation up to (and including) the first term in $x(t)$.

Part	Causal?	Memoryless?	Linear?	Time Invariant?
a- $x(t)$ large				
a- $x(t)$ small				
b				
c				
d				
e- $x(t)$ large				
e- $x(t)$ small				
f				
g				
h				
i				

For part **c**, you should show $y(t) = y(t_0)e^{\int_{t_0}^t x(\lambda)d\lambda}$ in order to determine whether the system is or is not causal and has memory or is memoryless.

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

determine the system is or is not causal and has memory or is memoryless. If you have trouble with these, read the handout on integrating factors.

2 For each of the following signals, determine E_∞ and P_∞ into 1Ω . Classify each of the following signals as energy or power signals (or neither).

- a. $v(t) = 4$
- b. $v(t) = 3\cos(2\pi 10t + 15^\circ)$
- c. $i(t) = 4\exp(-2|t|)$
- d. $x(t) = 4\text{rect}\left(\frac{t-2}{3}\right)$

3 For each of the following signals, determine if the signal is periodic and, if so, the fundamental period.

- a. $x(t) = \sin(2t) + \cos(3t + 30^\circ)$
- b. $x(t) = \cos(2t) + \cos(\pi t)$
- c. $x(t) = e^{-t} \cos(t)$
- d. $x(t) = 2e^{j2t} + 3e^{j(3t+2)}$

4. K & H, Problem 1.15. Hint: Draw pictures and think about what you are doing. The answer to **(d)** is $1/2$

5. An LTI system responds to the following inputs with the corresponding outputs:

$$\text{If (input) } x(t) = u(t) \text{ then (output) } y(t) = (1 - e^{-2t})u(t)$$

$$\text{If (input) } x(t) = \cos(2t) \text{ then (output) } y(t) = 0.707 \cos\left(2t - \frac{\pi}{4}\right)$$

Find (output) $y(t)$ for the following inputs:

- a) $x(t) = 2u(t) - 2u(t-1)$
- b) $x(t) = 4\cos(2(t-2))$
- c) $x(t) = 5u(t) + 10\cos(2t - 6)$
- d) $x(t) = \delta(t)$
- e) $x(t) = tu(t)$

*Hint: Use the results from problem K&H 1.23 for parts **d** and **e**.*

6. (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**, 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

$$\begin{aligned} x(t) &= \cos(t) & 0 < t < \pi \\ x(t) &= \cos(t) & 0 < t < 2\pi \\ x(t) &= |t| & -1 < t < 1 \\ x(t) &= t \cos(t) & -2 < t < 4 \end{aligned}$$

Hint: You will probably find the **sqrt** function useful.

For this problem, you can just copy down the answers from the Matlab screen. I will assume you are doing this in Matlab because if you are not it will soon become obvious....

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 Matlab function quadl tries to approximate the integral of a given scalar valued function using higher order recursive adaptive quadrature (aren't you glad you asked?) 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 2π

Note that an optional fourth argument is the tolerance, which defaults to 10^{-6} . When the function is very small or the integration time is very small you may need 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)
```

Note that we need to make up the function z, we cannot use something like

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

This will not work since quadl expects a single function.

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);
```

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);
```