ECE 300
Signals and Systems
Homework 8

Due Date: Tuesday October 24 at 2:30 PM

You are free the use the Fourier transform tables handed out in class or in the book for any of the following problems.

Problems

1. At the end of this assignment are two sets of plots: one is a group of time-domain functions, $x(t)$, labeled **a-h**, and a group of frequency domain functions, $|X(\omega)|$, the magnitude of the Fourier transform. These are labeled **1-8**. You are to determine which time domain function goes with which frequency domain function. Specifically, you need to fill in the following table (and show some justification for your answers)

$x(t)$	a	b	c	d	e	f	g	h
$\|X(\omega)\|$								

Hint: $e^{-at}\cos(bt)u(t) \leftrightarrow \dfrac{1}{(s+a)^2+b^2}$

2. K & H, Problem 3.20 (parts b and c only). Do all integrations by hand, but do not plot. Since the functions are real and even, your answers should be real.
3. K & H, Problem 3.22. Note that $p_4(\omega) = rect\left(\dfrac{\omega}{4}\right)$.
4. K & H, Problem 3.27
5. In this problem we'll look at a real world situation when we have to truncate a data set. This actually happens more with digital signal processing, but we can get the basic idea using our continuous time abilities.

    a) Find an expression for the Fourier transform of $f(t) = \cos(4t) + \cos(5t)$.

    b) Now assume we look at $f(t)$ for a finite time, say T seconds. What we see is actually $y(t) = f(t)rect(t/T)$. Determine an expression for the Fourier transform of $y(t)$, and write your answers in terms of sinc functions.

c) Plot, using **Matlab**, $Y(\omega)$ for ω between 0 and 10 when T=1, T=6, T=10, T=20, and T=40. Can you clearly tell there are two cosines present when you are looking at $Y(\omega)$ for all values of T? What happens as T gets larger (you are looking at more and more data)? Think in terms of the width of the sinc function (the distance between the first nulls). Note: The **sinc** function exists in **Matlab**.

In the following problem you are going to have to define a lot of anonymous functions, particularly functions that are the product of two other functions. You may want to review the appendix to homework 2.

6. Read the **Appendix** and then do the following:

a) Show that the functions $v_k(t) = e^{jk\omega_o t}$ are orthogonal, that is, show that

$$< v_j(t), v_i(t) > = \begin{cases} 0 & i \neq j \\ T & i = j \end{cases}$$

In addition, show that for these functions $c_k = \frac{1}{T}\int_T x(t)e^{-jk\omega_o t}dt$

<u>Write an m-file that will do all of the following (parts b-h), and turn it in with your homework.</u>

For parts **b-f** use the following functions:

$$v_1(t) = 0.7071$$
$$v_2(t) = 1.0754e^t - 1.2638$$
$$v_3(t) = 4.9632t + 4.9632 - 4.2232e^t$$

b) Using Matlab, show that these functions are (approximately) orthogonal over the interval [-1,1].

c) Using Matlab, show that these functions have unit length.

d) Assume we want to approximate $x(t) = t^4 - t$ in this interval using only $v_1(t)$, so that $x(t) \approx c_1 v_1(t)$. Determine c_1.

e) Assume we want to approximate $x(t)$ using the first two functions, so that $x(t) \approx c_1 v_1(t) + c_2 v_2(t)$. Determine c_1 and c_2.

f) Assume we want to approximate $x(t)$ using all three functions. Determine c_1, c_2, and c_3.

g) Plot the original function $x(t) = t^4 - t$ and the three approximations (from parts **d**, **e**, and **f**) on the same graph. Be sure to use different line types and a legend.

h) Using the *principle of orthogonality*, with the vectors

$$
\begin{aligned}
w_1(t) &= 1 \\
w_2(t) &= t \\
w_3(t) &= e^t
\end{aligned}
$$

determine an approximation to $x(t)$ using all three functions. Plot the approximation and the real function on the same graph. Compare your results to those in part **g. It may be useful to plot the error between the approximation made using the principle of orthogonality and the estimate from part f.**

Hint: To solve the system of equations Ac =b in Matlab, just type c = A\b. To enter the matrix

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

into Matlab, type

A = [a b c; d e f; g h i];

Appendix

The Fourier series is just one possible way of representing one function in terms of other functions. In order to determine a Fourier series representation for a function the function must be periodic. However, it is not necessary for a function to be periodic in order to represent it in terms of other functions. In fact, the Fourier series is really just a very special case of a general idea. In order to understand the general procedure a bit, we need some definitions first.

Inner Products An inner product can be thought of as a projection of one vector onto another vector. The “dot product” of two ordinary (Euclidean) vectors is an example of an inner product. We need to expand this type of definition to include functions. We will denote the inner product between the two functions $x(t)$ and $v(t)$ as $< x(t), v(t) >$.There are mathematical rules for what constitutes a valid inner product, but we will use the following

$$< x(t), v(t) > = \int_T x(t) v^*(t) dt$$

Here the * means the complex conjugate. Note that the inner product of a function with itself gives us the length squared of the function, just as the dot product of a vector with itself gives us the length squared of the vector.

Orthogonal Functions Two functions $x(t)$ and $v(t)$ are orthogonal if their inner product is zero, i.e., if $< x(t), v(t) > = 0$. This is similar to saying to vectors are orthogonal if their dot product is zero.

Orthonormal Functions Two functions $x(t)$ and $v(t)$ are orthonormal if they are both orthogonal and have unit length, i.e. if

$$
\begin{aligned}
&< x(t), v(t) >= 0 \\
&< x(t), x(t) >= 1 \\
&< v(t), v(t) >= 1
\end{aligned}
$$

Orthogonal Function Representation If functions $v_1(t), v_2(t), ..., v_n(t)$ are orthogonal, then the estimate of a function $x(t)$ that minimizes the squared error is

$$x(t) \approx c_1 v_1(t) + c_2 v_2(t) + ... + c_n v_n(t)$$

where

$$c_k = \frac{< x(t), v_k(t) >}{< v_k(t), v_k(t) >}$$

If the $v_i(t)$ are orthonormal, then $c_k = < x(t), v_k(t) >$ since the denominator is clearly equal to 1. Here the squared error is defined as $< e(t), e(t) >$ where the error signal $e(t)$ is defined as

$$e(t) = x(t) - [c_1 v_1(t) + c_2 v_2(t) + ... + c_n v_n(t)]$$

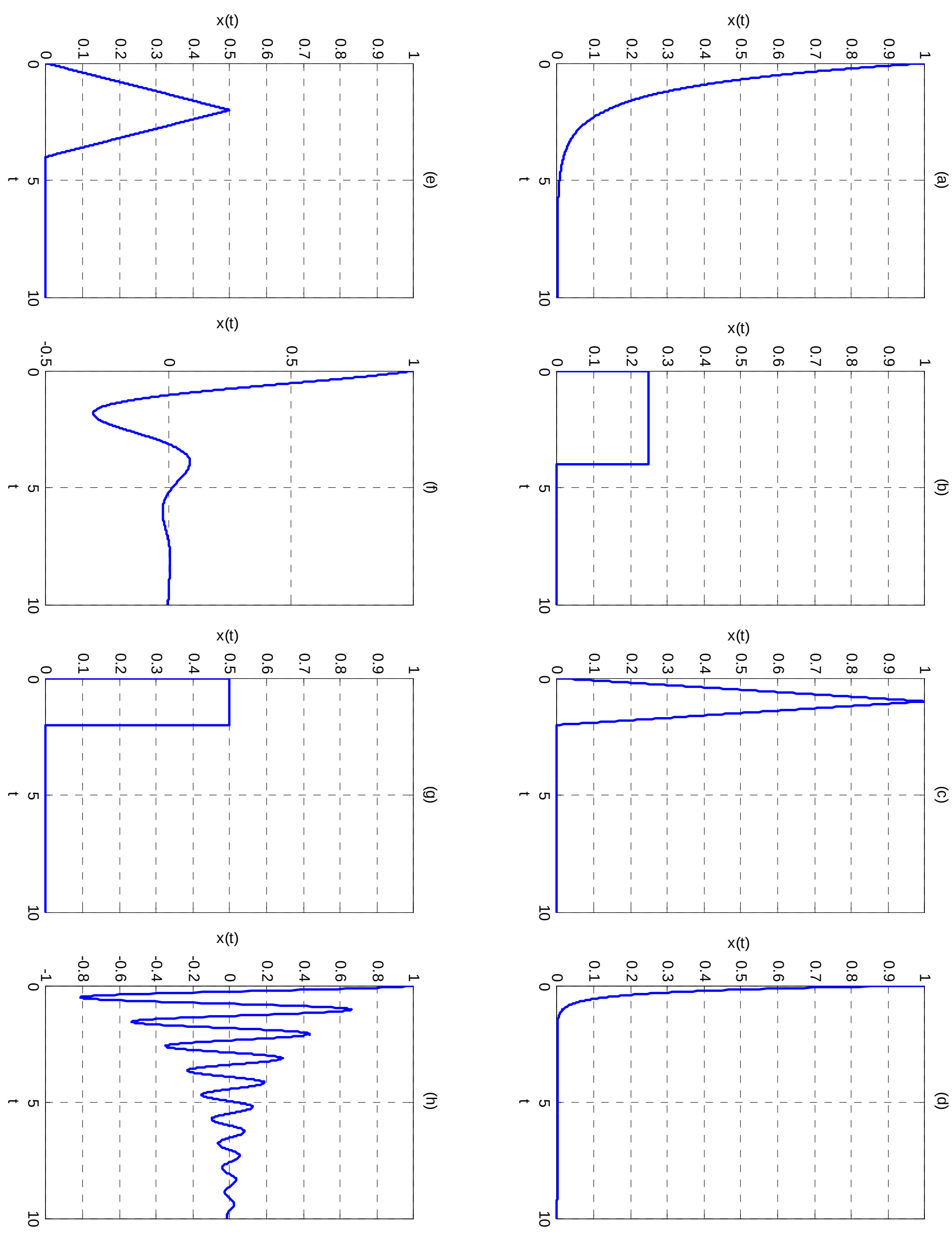
Principle of Orthogonality This principle states that to produce the estimate that minimizes the squared error, the error signal should be orthogonal to each of the functions used in creating the estimate. Mathematically, this means that

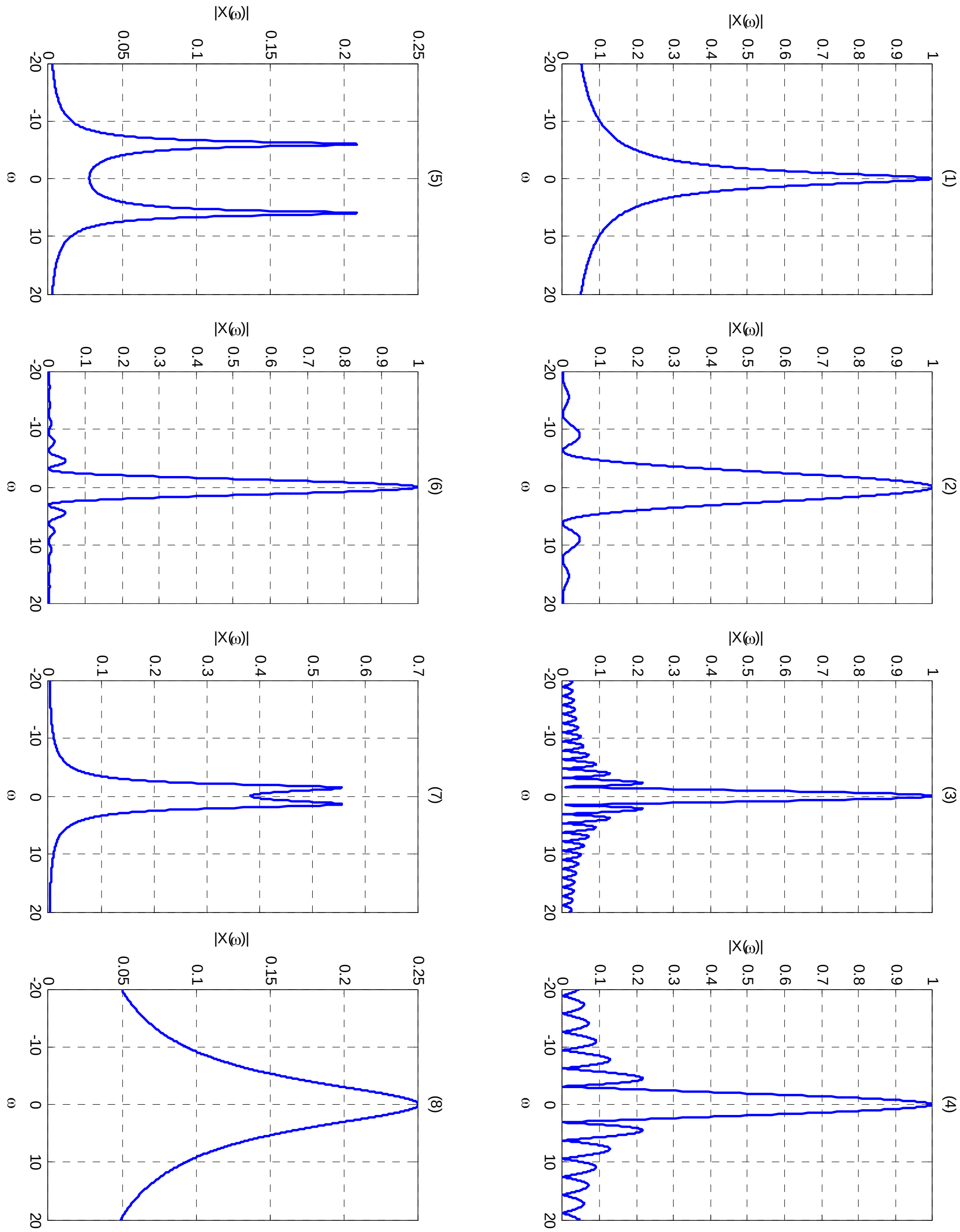
$$< e(t), v_i(t) > = 0 \text{ for } i = 1...N$$

Note that this does not require the $v_i(t)$ to be orthogonal or orthonormal. However, there is a set of simultaneous equations that must be solved.

Note: If the function is a constant, say 0.7071, and you want to square it, you may need to use the following function definition:

*v1 = @(t) 0.7071*ones(1,length(t))*


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
x(t)
t


(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
|X(ω)|
ω