# ECE2311 Homework Assignment Number 1
# Due by 10:50am on 23-Mar-2010

Try to do some of these problems each day; don't wait until the last minute. Be sure your name and ECE box appear legibly on each page. Show all work, and keep your work neat and organized. Whenever appropriate, use *words* to explain what you are doing, not just mathematical symbols. Be sure to justify your answers so the grader can be confident that you aren't just guessing.

**An Introduction to Matlab:** One of the goals of ECE 2311 is for you to get acquainted with signal-processing computer applications including Matlab, Simulink, and Maple. This assignment will provide an introduction to some of the capabilities and syntax of Matlab.

Matlab can be run on any WPI computer simply by selecting the "Matlab" icon from the desktop or the Windows start menu. There is also a free open-source alternative to Matlab called Octave that you can download from `http://www.gnu.org/software/octave/` and run on your own computer. Octave syntax is 99% identical to Matlab and can be used to complete all of the exercises in this assignment.

After starting Matlab (or Octave), you should see a command prompt that looks like this: $\gg$. Most of what you will want to do in Matlab is accomplished from the creation of a scripted program file, called an "m-file". For now, though, we can work strictly from the Matlab command prompt, so you won't have to worry about things like saving and running files until later.

As with most computer applications, getting started in Matlab requires a little help. One good place to start would be with the Matlab "Getting Started" tutorial, which can be found at The MathWorks website. You can also get it from the **Matlab Help Desk**, which is accessed from the Matlab prompt using the `helpdesk` command (a web browser, such as Netscape, should begin to run). You could also start with the tutorial on page 53 of your textbook. Note that Lathi has Matlab sections sprinkled throughout the book; there is one "Session" at the end of each chapter.

Although getting better at Matlab is not difficult, it *does* require the constant effort of learning the proper use and syntax of new commands. Luckily, you can get help on any of Matlab's hundreds of commands by typing `help`, followed by a space and the name of the command, at the prompt. Also, you may search for command help in either the Matlab Help Window (type `helpwin`), or at the Help Desk (which usually provides greater detail, and visual examples of usage).

With all that out of the way, let's begin.

1. The ability to shift, scale, and otherwise manipulate functions is invaluable to the study of continuous-time signals. Consider a function $x(t)$, defined by

$$x(t) = \begin{cases} 4 - t^2, & \text{if } -3 < t < 2 \\ 0, & \text{otherwise.} \end{cases}$$

First, sketch this function by hand over the range $-5 < t < 5$. Now, let's plot it using Matlab. Since Matlab is a computer application, it can't "draw" a graph in the intuitive way that we do—it can only calculate values for $x(t)$, given a finite set of values for $t$. Thus, we have to define an *array* to store the range of time values at which $x(t)$ will be evaluated. One way to do this is to type

```
t = -5:0.01:5;
```

where t is the array name. This command will create an array of values starting with –5, incremented by 0.01 until it reaches 5. Next, to create the function $x(t)$, we can use the following commands:

```
x1 = 4-t.^2;
x2 = (t>-3)&(t<2);
x = x1.*x2;
```

Although it might look a little messy, this code is actually quite straightforward. Let's walk through it, step-by-step. In the first line, we create an array x1 with elements calculated according to the expression for $x(t)$. Note that MATLAB evaluates all of the elements in the array in one line of code. Unlike other languages like C, a for loop is not necessary here. Also note the period before the ^ operator. This tells MATLAB to perform the squaring operation *element-by-element* on the array t. Try this command without the period, i.e. try x1 = 4-t^2;. In this case, MATLAB thinks you want to perform a matrix/vector squaring of the array t, which doesn't make any sense and will result in an error.

The difference between element-by-element and vector/matrix operations in MATLAB is very important. In ECE2311, you will almost always be using element-by-element operations.

The next line computes another array with elements equal to one when $-3 < t < 2$ and equal to zero otherwise. This is an important part of the function $x(t)$, since $x(t)$ is nonzero only between –3 and 2 and is equal to zero otherwise. MATLAB automatically computes all of the elements in this array without any for loops.

The third line performs an element-by-element multiplication of x1 and x2 to generate the desired function $x(t)$. Again, note the period before the * operator. Try this command without the period and MATLAB will generate an error. Note also that you can use either "*" or ".*" for the scalar multiplication of a constant times an array, but we must use the element-by-element operator to multiply the respective elements of two arrays. In other words, the logical construction in this statement merely serves to set our piecewise function $2 - |t|$ to zero outside the range $-3 < t < 2$.

To graph the function $x(t)$, we simply have MATLAB plot the values in array f against the values in array t. This can be done by typing

```
plot(t,x);
xlabel('time (seconds)');
ylabel('x(t)');
```

You should always apply axis labels to your figures. You may also want to apply a title, legend, or other features to your figures. You can learn all about the extensive plotting features MATLAB by typing doc plot on the MATLAB command line.

Print your MATLAB figure by selecting "File" and then "Print..." from the figure window that appears (or try the print command from the MATLAB command line). It's that easy!

2. Each of the functions $y_n(t)$ below is some transformation of the function $x(t)$ from Problem 1. **First sketch each function by hand**, showing your work. Then plot the function using MATLAB, over the range [–5, 5]. For these problems, keep in mind how MATLAB's arrays work. In part (a), for instance, if we were to do something like "t = t+1;", and then re-store and re-plot x, we would get the right function, but the *range* will have shifted (that is, we will have plotted $y_1(t)$ over the range [–4, 6], *not* [–5, 5]). Change the function, but not the range.

Just use MATLAB to check your answer; be sure you know how to create each new signal by hand. Also, state what transformations you use (shift, time scale, etc.), and in what order. Be sure to print out your graphs, and attach them to your homework.

(a) $y_1(t) = x(t+1)$

(b) $y_2(t) = x(\frac{t}{2})$

(c) $y_3(t) = x(1-t)$

(d) $y_4(t) = x(3t+3) + 2x(t^2)$

3. Sketch

$$\begin{aligned} x_1(t) &= \cos(2\pi t + \pi/6) \\ x_2(t) &= \sin(2\pi t) \end{aligned}$$

over the range $-2 \leq t \leq 2$ and describe the relationship between $x_1(t)$ and $x_2(t)$ in terms of one signal leading or lagging the other.

4. Given the periodic signal $x(t)$ shown in Figure 1, determine the fundamental period, the fundamental frequency in Hertz, and the fundamental frequency in radians/sec.

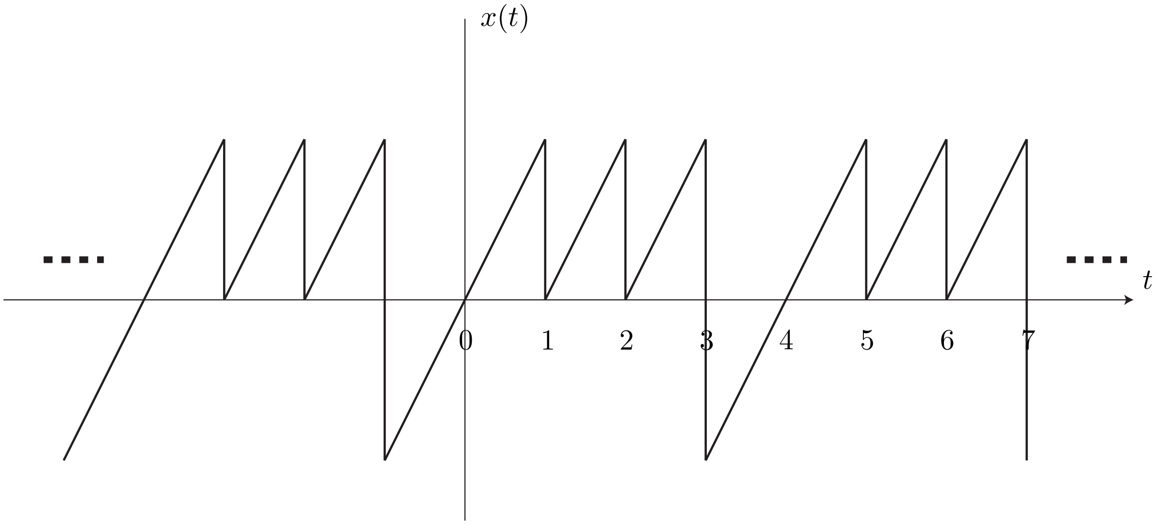

Figure 1: A periodic signal $x(t)$.

5. Lathi, problem 1.7-9.

6. Lathi, problem 1.7-11 (b), (c), (d), (e).