ECE-497/BME-491: Applied Biomedical Signal Processing
Laptop Day #1
Due at the end of class, December 1, 2006

Today we have the following goals

- pass arguments to a program
- load data into a program
- determine the normalized autocorrelation of a signal using the direct method
- plotting data nicely, reviewing subplot

In this worksheet I will give you alot more help than usual, but it has probably been awhile since you have had to use Matlab. I expect you to learn by example, so you will not be shown most of this again.

At the end of class you should turn in your two plots and a copy of your program.

1) Go the the class website and download the program shell laptop1.m and the files *pt1 vf.pa* and *pt2 vt.pa*. We will be filling in this program shell (laptop1.m) today. You will need to modify this more for your homework. Some things to note:

- there are three **input** arguments to this file:
  - the name of the data file, in single quotes
  - the starting point (index) in the file, usually the first point in the file so this is usually a 1
  - the number of sample points in the file, usually this will be 511
- there is one **output**, the location of the peak of the autocorrelation. This is included in brackets so you can add other items to return (if you should need to later)
- the data file is read in for you, and modified so you can use it
- the sampling rate it 125 Hz (fs = 125)

2) Start Matlab, set the directly to where your program/data files are. To be sure everything is ok, invoke the program as follows:

```
laptop1('pt1 vf.pa',1,511);
```

If you have done everything correctly so far, nothing should happen (at least it didn't crash!)

3) Now we want to plot the data we have, so we need to create a time vector. We know there are N data points, and we want the data to start at zero. We also know the sampling frequency is fs. Hence the following command can be used to determine an appropriate time vector:

```
time = [0:N-1]/fs;
```

We are going to want to plot in landscape mode, so before any plotting, add the command:

```
orient landscape
```

You will see no difference on your screen, but it should print out in landscape mode.

We are going to plot two signals on one page, so you will need to review the use of the **subplot** command. Now plot the original signal (amplitude versus time) on the top panel. Be sure to label the time axis in seconds. Uncomment the appropriate graph title. See Figure 1.

4) Now we need to compute the normalized autocorrelation using the direct method. This method is usually given in a formula as

$$\rho_k = \frac{\sum_{i=k}^{i=N-1} x(i)x(i-k)}{\sqrt{\sum_{j=k}^{j=N-1} x(j)^2}\sqrt{\sum_{j=0}^{j=N-k-1} x(j)^2}}$$

Here ρ_k is the normalized autocorrelation at lag k. We are going to want to compute the lags from $k = 16$ to $k = 64$, so we will need a **for** loop, like

```
for k=16:64
%
%  stuff happens
%
end;
```

Now we need to fill in the **for** loop. We are going to have to be a little bit careful here, since Matlab starts its arrays with 1, not zero. Let's assume we wanted to look at the $k = 3$ lag, and there were $N = 10$ points in the x array. Then we would want the numerator to be

$$x(1)x(4) + x(2)x(5) + x(3)x(6) + x(4)x(7) + x(5)x(8) + x(6)x(9) + x(7)x(10)$$

We can easily implement this in Matlab using the commands

$$sum(x(1 : N - k). * x(k + 1 : N))$$

From this we can easily determine how to determine both the numerator and the denominator (using the **sqrt** command), so we need to do one last thing. We are going to be plotting this, so we want to save the normalized autocorrelation at each lag, so let's assign this to an array variable

```
rho_direct(k) = ...
```

At this point our **for** loop can end.

5) We are going to need to determine the location of the peak of the autocorrelation. This tells us the best possible autocorrelation of our signal with itself. To do this in Matlab we would use

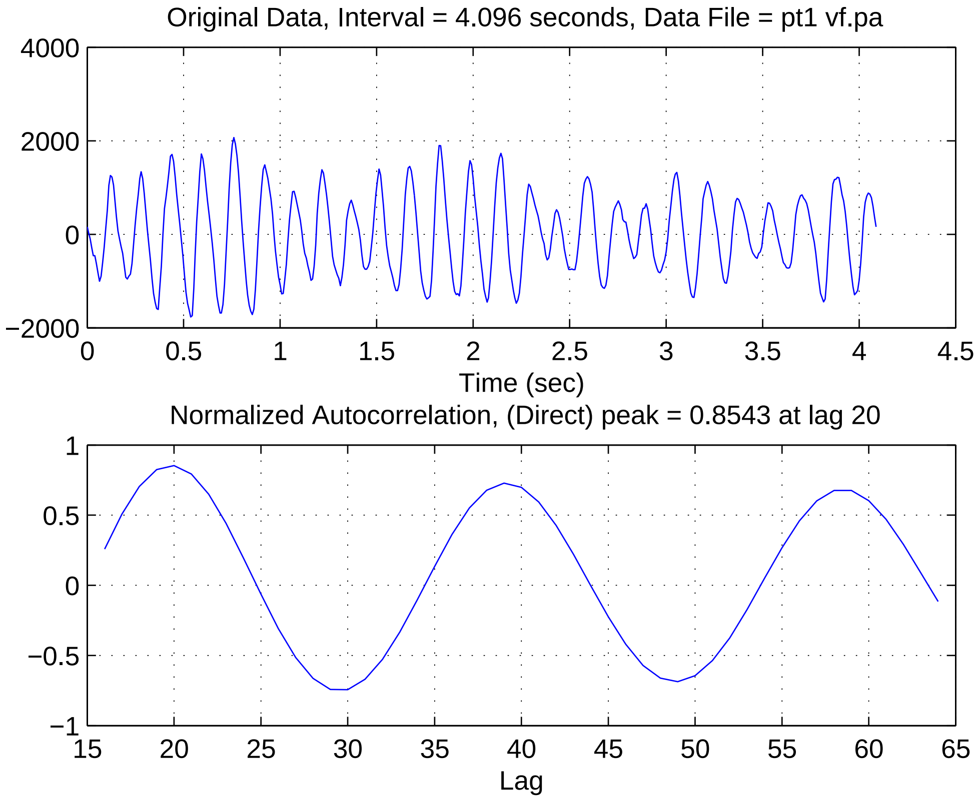

Figure 1: Results for *pt1 vf.pa*

```
[peak_direct, index_direct] = max(rho_direct)
```

This will allow us to determine both the value of the peak and the normalized autocorrelation lag at which this occurred.

6) Now plot the normalized autocorrelation versus lag on the bottom panel (the top should be the original signal). You should be sure to label the x-axis as **lag** and only plot the normalized autocorrelations with lags from 16 to 64. Uncomment the appropriate graph title. Your final graph should look like Figure 1.

7) Run your program using file *pt2 vt.pa*, and compare your results with those in Figure 2.

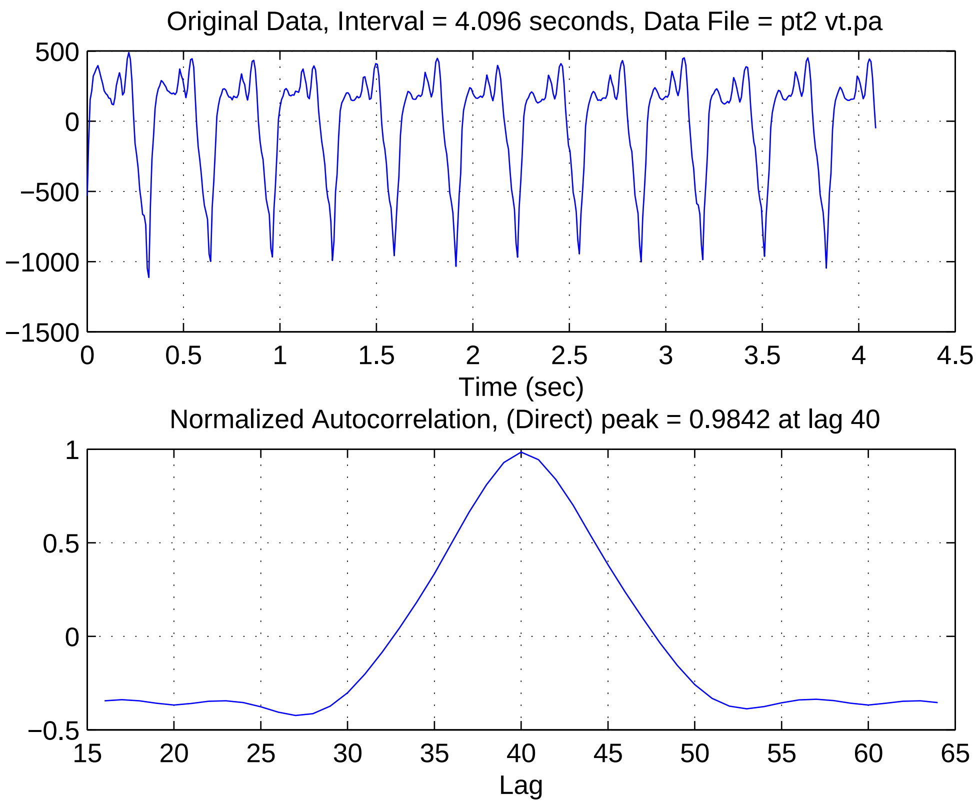

Figure 2: Results for *pt2 vt.pa*