ECE-497/BME-491: Applied Biomedical Signal Processing
Laptop Day #2
Due at the end of class, December 8, 2006

Today we have the following goals

- use a simple amplitude threshold to detect beats
- make a template to use for signal averaging using the program *select_template.m*
- signal average multiple beats using the program *signal_average.m*

At the end of class you should turn in four plots and answers to some questions.

1) Go to the class website and download the programs *laptop2.m*, *select_template.m*, *signal_average.m* and the files *pt8 vt.pa* and *pt9 vt.bi*. We will be slightly modifying (laptop2.m) today. You will need to modify this more for your homework. Some things to note:

- there are five **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
  - the threshold level (at this fraction of the maximum amplitude a beat will be detected)
  - the dead time, no other beats within *dead* sample points can be detected
- there is one **output**, the locations (indices into the data array) of the beats. 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:

```
beat_index = laptop2('pt8 vt.pa',1,511,0.8,10);
```

If you have done everything correctly so far, it should plot the original signal.

3) The program will have located the indices to the beats, and they are stored in the array **beat_index**. You need to modify the plotting so the places the beats have been detected will be marked by a '+' sign, as they have in Figure 1. Note that the first beat is not detected. Don't worry for now how some of this program works, you'll get to figure it out on your homework.

Now we want to try some *signal averaging*. There are generally three steps to this:

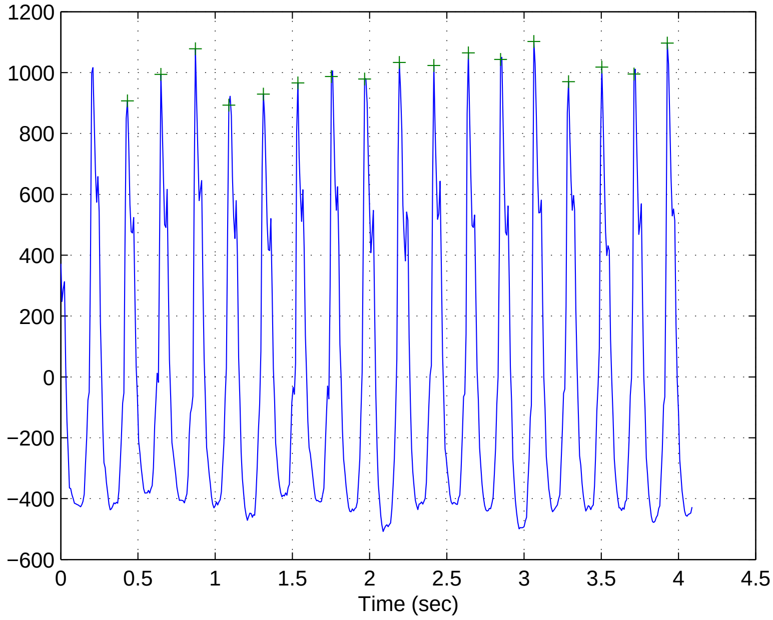

Figure 1: The signal and the detected beats. The '+' indicates the location the beat first exceeded the threshold.

- determining where the beats are (*triggering points*, similar to an oscilloscope triggering points). We just did this in step 3.
- selecting a *template*. This is what we consider to be a typical beat. In selecting a template we need to determine how far to the left and right of our triggering point the beat extends.
- averaging the template with other similar beats. What constitutes a similar beat we will get to later.

I have written programs for you to use to do the remaining parts.

4) *Selecting a Template* We need to determine what we think is a typical beat. This involves determining how much of the signal we want to call a beat. There are three input arguments to *select_template.m* and the program has three outputs.

The **input** arguments are:

- trigger, the indices into the data indicating the beat locations. This is the data in beat_index, the output of laptop2.m
- beat, the beat to use to create the template. This can be beat 2, 3, 4, etc.
- name, the name of the data file containing the data, in quotes

The **outputs** of this file are:

- template, the data points within the template
- lefti, the beat starts at lefti sample points from where we identified the beat
- righti, the beat ends at righti sample points from where we identified the beat

You need to start this program using the command

```
[template,lefti,righti] = select_template(beat_index,2,'pt8 vt.pa');
```

The program should present a picture with your chosen beat in the middle of the window, as shown in Figure 2.

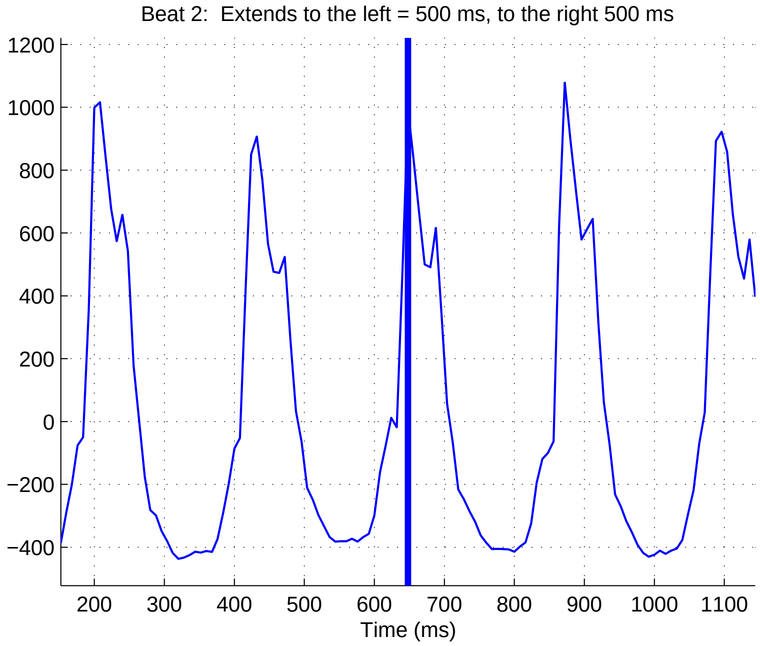

Figure 2: Initial window in select_template.m. The solid line is the location of the beat indicator. From this indicator the user determines the template to use.

Determining the correct size for a template usually involves some experience (it's easier using body surface electrograms), but for this assignment let's assume we want the template to extend for 75 ms in each direction from the mid-point. You should end up with the graph shown in Figure 3.

5) *Signal Averaging* We will use the code *signal_average* to do our signal averaging. There are 6 input arguments to this program and one output. The **input** arguments are

- trigger, the indices into the data indicating the beat locations. This is the data in beat_index, the output of laptop2.m
- lefti (defined above)

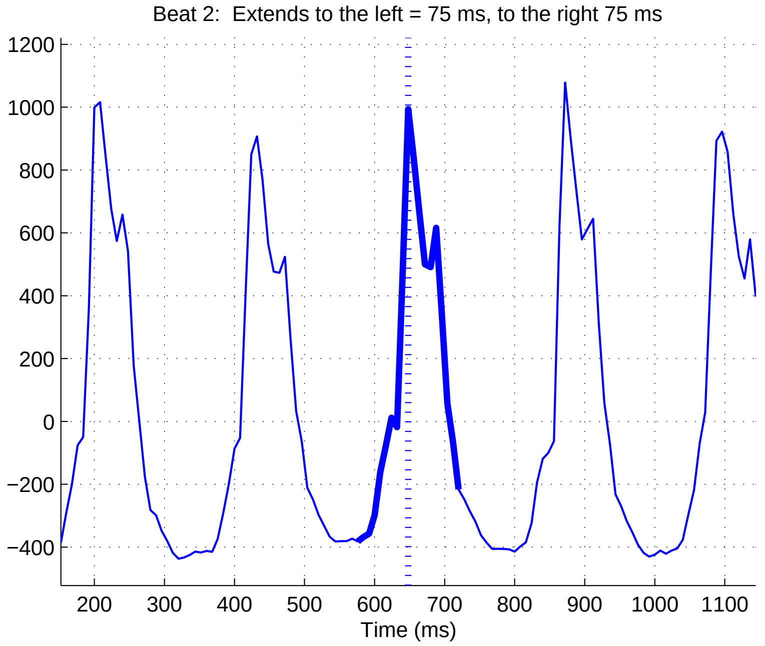

Figure 3: The final window in select_template.m. The thick solid line is the location of the template.

- righti (defined above)
- template (defined above)
- threshold. This is used to determine if the template matches the beat well enough that we want to call this beat an average beat and use it in our signal averaging. When we align the template with each beat we compute the correlation coefficient between the template and the beat. The correlation coefficient is independent of amplitude and is bounded between -1 and 1. The closer to 1 the correlation coefficient, the more closer the template fits the beat. To start with, assume we want this threshold to be 0.9
- name, the name of the data file containing the data, in quotes

The **output** of the file is the signal averaged beat.

As the program runs it first indicates how many beats meet the threshold requirement (how many beats can be used in the signal averaging). Usually as the threshold is reduced the number of beats that can be used in the signal averaging will increase, but they become less like the original beat. At some point we just don't want to include bad beats anymore. Finally, the program also plots the original template and the final signal averaged beat.

No run this program using the following command

```
average_template = signal_average(beat_index,lefti,righti,template,0.9,'pt8 vt.pa');
```

There should be 16 good beats. Run the program using 2 beats in the signal averaged beat, then 5 beats, then 10 beats, then 15 beats. How do the template and signal averaged beats compare? Is signal averaging more like a low-pass filter or a high-pass filter?

6) Repeat steps 1-5 using file *pt9 vt.bi.* Specifically,

- Be sure to use a small enough threshold you can locate all of the beats (except maybe the first and last beats). *Turn in your plot with the '+' locating the beats (similar to Figure 1)*
- Use the same sized template (± 75 ms from the center point). *Turn in your plot showing the template selected (similar to Figure 3)*
- When signal averaging, use an initial threshold of 0.9 and average all of the beats that meet this criteria. *Turn in your plot showing the template and signal averaged beat.* How does the signal averaging affect the template?
- Lower the threshold so you can include at least 12 beats in your signal averaging. *Turn in your plot showing the template and the signal average beat.* How does the signal averaging with these beats affect the template? What do you think is causing this result?