

ECE-497/BME-491: Applied Biomedical Signal Processing

Laptop Day #3

Due at the end of class, December 15, 2006

Today we have the following goals

- plot the frequency response of a discrete-time filter
- filter a discrete-time signal
- implement a filter to remove baseline wander
- implement a moving average (MA) filter

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

1) Go to the class website and download the program *laptop3.m* and the data files *ecg_bn.dat* and *ecg5.dat*.

There are five **input** arguments to *laptop3.m*:

- 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
- the location of the pole in the filter used to remove the baseline wander
- the order of the MA filter

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

```
laptop3(0.8,21,'ecg_bn.dat',1,3000);
```

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

3) We next want to implement a simple filter that will remove the baseline wander without modifying the signal too much. In class we showed one such filter had the transfer function

$$H(z) = \frac{1 - z^{-1}}{1 - \alpha z^{-1}} \frac{1 + \alpha}{2}$$

Here α is the location of the pole we will use so the filter will (hopefully) remove the low frequency components of the signal while leaving the remaining frequency components alone.

Matlab generally expects a filter to be of the form

$$H(z) = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2} + \dots + b_{N+1} z^{-N}}{a_1 + a_2 z^{-1} + a_3 z^{-2} + \dots + a_{N+1} z^{-N}}$$

and to implement the filter in Matlab we enter the b and a coefficients as vectors,

```
b = [b1 b2 b3 ... bN];  
a = [a1 a2 a3 ... aN];
```

Note that the lengths of the a and b vectors must be the same, which usually means you need to pad the a vector with zeros.

Determine the correct a and b vectors for our filter to remove baseline wander, and type it into *laptop3.m*. Your coefficients should be a function of the variable 'alpha' which is passed to the function.

4) In order to filter the input signal using the filter coefficients we have determined, we will use the Matlab function *filter*, which is fairly easy to do. We just need to pass it the b and a vectors as well as the signal to be filtered. Let's assume the signal to be filtered is an array called x and we want the filtered signal to be called p . Then we just type

```
p = filter(b,a,x);
```

Modify your Matlab code so the input signal can be filtered using the filter you implemented in part 3. Call your filtered signal p .

5) We also want to be able to plot the frequency response of our filter. There are many ways to do this, we will use the *freqz* function. For this function we pass it the a and b vectors, and it returns the frequency response (magnitude and phase) $H(e^{i\Omega})$ at selected frequencies Ω . Note that $0 \leq \Omega \leq \pi$. To use this we type

```
[H1,Omega1] = freqz(b,a);
```

Implement this in Matlab for your filter. Note that in order to convert this frequency to a real frequency we need to scale using $f_{real} = \frac{\Omega_1 f_s}{\pi}$ where f_s is the sampling frequency and f_{real} corresponds to the real frequency.

6) Uncomment the plotting parts of the code at the bottom of *laptop3.m*, and run *laptop3.m* using $\alpha = 0.8$ you should get plots like those shown in Figure 1. (There are more plots here that you have so far).

7) We next need to implement the moving average filter. The input signal to this filter will be p , which was the output of our first filter, and the output of the filter will be y . The easiest way to implement this is with the Matlab code

```
a = zeros(1,order);  
a(1) = 1;  
b = ones(1,order)/order;
```

Implement this in *laptop3.m* and put in the code to filter p to get y . Next put in code to determine the frequency response of the MA filter. Finally modify the code to plot all of the results. If you have done everything correctly you should get the same graph as in Figure 1 (if you use $\alpha = 0.8$ and $order = 21$).

8) The choice of parameters for this exercise was really quite poor. Modify your choice of α and the order of the MA filter so the output looks as much like the original signal as possible (without the original signals baseline offset and noise). Include (uncomment) the code to limit all of the axes for x , p , and y so the range is from -2.5 to 3.5, so they are all plotted on the same scale. *Turn in your final plot.*

9) Try process the data in the file *ecg5.dat*. This data file is sampled at 200 Hz, so your code will have to be modified. In particular, remove the axis scaling and modify the sampling frequency. Process this file using the first 3000 samples. Modify your parameters as necessary to produce a resulting plot with little baseline wander and with the signal looking as much like the original signals as possible. *Turn in your plot.*

Turn in your code.

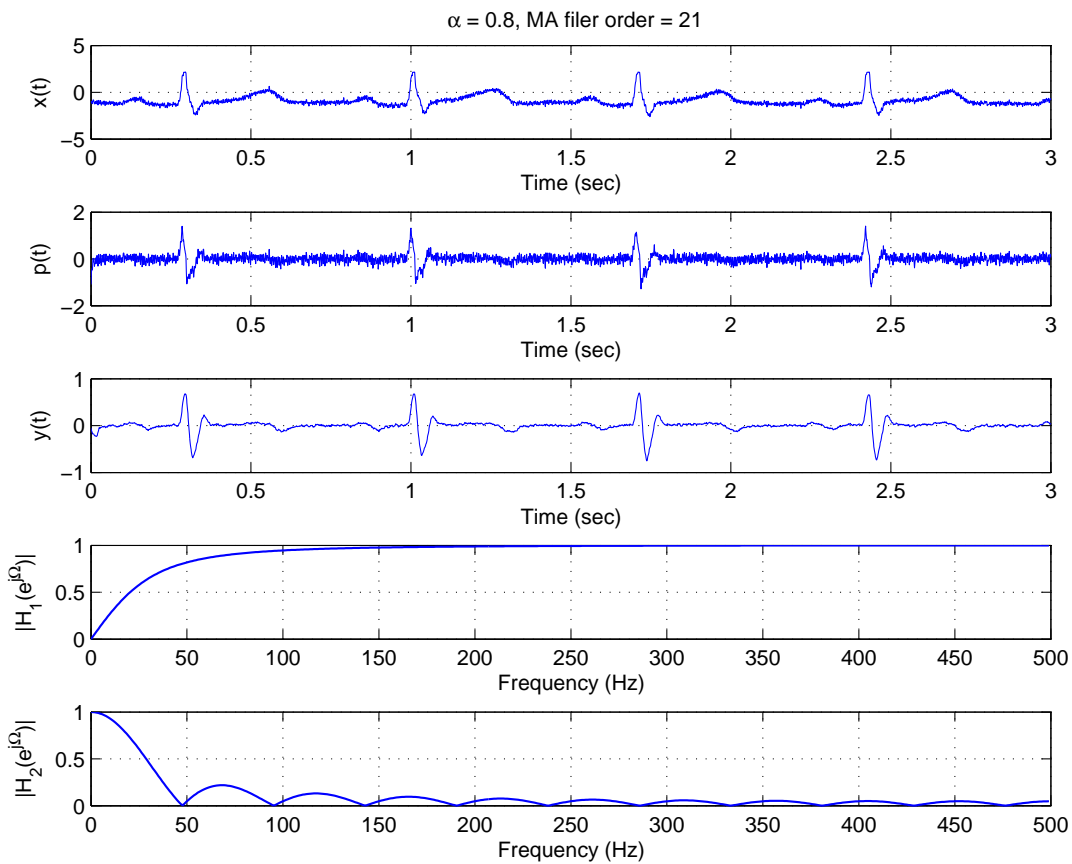


Figure 1: Results from processing data file *ecg_bn.dat* using $\alpha = 0.8$ and *order* = 21