In this assignment you will be building continuous-time (analog) filters to remove noise from a signal. You will be de-noising an electrocardiogram (ECG) signal that has been corrupted by 60Hz noise, which is a common problem in any recording made from sensitive electronics. You will denoise the signal in two different ways: once using Matlab and once using Multisim.

You are given a noisy ECG signal that has been sampled at $F_s = 1000$ samples per second. Unlike in previous assignments, this signal is given to you as ASCII text, meaning you can open and read it as plain text if you like. The file has two columns. The first column contains time (in seconds) and the second column contains the ECG signal. In Matlab, you can load this file into memory with the following commands:

```
clear
data = load('ecg_data.txt');
t = data(:,1);
x = data(:,2);
fs = 1000;
```

Step 0: Plot the ECG signal with respect to time. You should see a *very* noise corrupted signal.

Step 1: Examine the frequency content of the signal to confirm that the noise source is in fact 60Hz hum.

Step 2: Using the steps outlined in Chapter 4 and in lecture, design an analog (continuous-time) filter in Matlab that will remove as much of the 60Hz noise as possible while keeping as much of the EEG signal as possible. You are limited to a third order filter. You may use commands such as `butter` or `cheby1` to design your filter, and the `lsim` command to filter your ECG signal. Demonstrate that your simulated filter reduces or removes the 60Hz noise. Make sure you give the transfer function equation for your filter in your report.

You may not use the `lowpass` or `highpass` functions because those implement *discrete-time* filters. The purpose of this assignment is to implement a *continuous* filter, such as one built with resistors and capacitors and so on.

Step 3: Design a circuit in Multisim that will filter the ECG signal. You are welcome to try implementing your same filter design from Step 2 as a circuit, or you can make a new, simpler filter if you prefer. Feel free to do a simple RC filter if that's what you're comfortable with, but don't be shy about trying something more ambitious like a Butterworth filter. In either case, you are still limited to a third order filter. Run a 10-second transient simulation to verify that your noisy ECG signal has in fact been filtered. To do so, you will need to use the `PIECEWISE_LINEAR_VOLTAGE` component as your voltage source. Once you have placed that component, edit its properties - under the "Value" tab, select "Use data directly from file" and browse to the `ecg_data.txt` file. This way, when you run your simulation, the voltage source will produce the ECG signal that your circuit will then filter.

Honors Students - Build your circuit from Step 3 in actual hardware. You have several options, but you should use a testbench that has an "Arbitrary Waveform Generator". This will create the

ECG signal as an actual voltage that you can filter. Filter the signal to verify it works.

Questions - What kind of filter did you design in Steps 2 & 3. Why? Justify your filter design choices by discussing the frequency content of the noisy ECG signal and the frequency properties of your filter. How did you know the filter worked? How does the frequency content of the filtered signal compare with that of the unfiltered signal? Would using a higher order filter be helpful? Why or why not?

What to hand in - Using the provided template, you should hand in a report (1-2 pages) written in MS Word with the following sections: Intro, Methods, Results, Discussion. Your report must answer all of the above questions as well as any other interesting observations you make along the way. You should decide for yourself what information is important to present in your report. The purpose of the report is to convince me that you understand the assignment, so think about the most efficient way of getting your point across. This is a technical report, so avoid humor and colloquial jargon!

Teams - You may work by yourself or in groups of two. If you work in groups, put both your names on the paper, but only one of you needs to do the submission in Canvas. In all cases, you and/or your group may work with other individuals or groups, but you/your group's code and paper must be uniquely yours.

You should also turn in your Matlab code and Multisim schematics. Code should use meaningful variable names and be well commented and otherwise easy for me to read. Schematics should be clean and easy for me to read. No need to include the `ecg_data.mat` files - I already have those.

Zip together the MS Word report (not a PDF) and all other files and submit *a single zip file* through Canvas. You may work in teams of two or by yourself. Submissions should arrive by Monday 3/16/2020 at 11pm.