Lab 3.2

Digital Filtering Prelab

Authors:

Jonas Lussi, Alexandre Mesot, Naveen Shamsudhin and Prof. Bradley J. Nelson

Institute of Robotics and Intelligent Systems

Date: 2020

Version: 1.0

Summary: This Prelab will help you understand and apply the concepts of the Digital Filtering Lecture.

2.1 Background

2.1.1 Digital Filtering

The background information for this Prelab is given in the lecture entitled **Digital Filtering**. There will be no Postlab in this Lab.

2.2 Prelab Procedure

Note: This Prelab report must be uploaded to Moodle by next Tuesday. The deadline is available on Moodle. All the prelab tasks are marked with **PreLab Qx**

1. Design a first order FIR filter using a Blackman windowed sinc with a cut-off frequency of 100Hz and a transition band of 20Hz. Assume a sampling frequency of 1000 Hz. You don't have to write any code, but just note on your prelab report the M and f_c you found along with your analytical calculations. **(PreLab Q1)**

2. Use the following Matlab code to analyze the frequency response of the filters.

```
%%
clear;
 close all;
 newplot;
 % How long is the sample kernel?
 M = 100;
 % What is the cut-off frequency?
 fc = .2;
 % This function builds a Blackman filter kernel
 % with cut-off frequency fc.
```

```
blk = fir1(M,fc, blackman(M+1));
% We also want a moving average filter
% It can be of different lenghts than the Blackman
N = 20;
% This will not be as long as the Blackman filter
% So we have to pad the rest of the vector with 0s
avg = [zeros(M/2-N+1,1) ; ones(N,1)./N ; zeros(M/2,1)];
% We'd like to plot multiple times on the same graph to show filter
% kernels:
hold on;
plot(blk,'-b');
xlabel('sample number')
ylabel('Amplitude')
plot(avg,'.-r');
legend('Blackman','Moving Average');
% This is the filter visualization tool: go on
% Analysis->Analysisparemters and change Magnitude Display from
% "Magnitude(dB)" to "Magnitude" to have a linear scale.
fvtool(blk,1,avg,1);
legend('Blackman','Moving Average');
```

With the Filter Visualization tool, plot the Magnitude Response at different frequencies for your previously designed Blackman window (from Prelab question 1). Then double the cut-off frequency and plot again the response. Do the same with the kernel length M. Finally, also plot the smoothing filter with N = 10 and 100. On your report, briefly describe the influence of these parameters (cut-off frequency, M, N) on the magnitude response. When you look at the magnitude response (linear scale), where do you see the two biggest disadvantages of the moving average filter (for your Prelab Report you don't have to hand in any plots)? **(PreLab Q2)**

3. You have to design a filter to eliminate 50 Hz noise in an ECG signal. What is the sampling frequency you should use? Explain why you chose this frequency. **(PreLab Q3)**

4. You are given a signal: A sinusoid with noise (`noisy_sinusoid.txt`). Write the C function smoothing_filter in the file `digital_filter.c` (provided as skeleton) that implements a moving average filter on the given signal. The provided code will then save the filtered data in a file. The function is called from your main file `lab03_2.c`, where you can change the length of your filter length N. Plot for filtered and unfiltered signal in a single plot in Matlab and comment on the differences between them. (Note: For big N you might have to account for the delay in the filters to be able to easily compare the output to the raw data in Matlab). You only have to hand in one plot with a suitable filter length N. Comment why you choose this filter length. **(Prelab Q4)**

Explanation of given Code: To be able to read a file into your C program you must define a variable of type `FILE` which the code uses to access the file. For example, your file is called `myfile.txt` and you want the variable that access it to be named `mydata`. What you need to do is `FILE *mydata` in the variables declaration, `mydata=fopen("myfile.txt","r");` to open the file for reading and `fscanf(mydata,"%f%*c%f",&variable1,&variable2);` to read its contents. Do not forget to close the file by `fclose(mydata);`!

```
FILE *mydata;
double variable1[100], variable2[100];
mydata = fopen("myfile.txt","r");
for (j=0;j<length_array;j++)
    {
        fscanf(mydata,"%f%*c%f\n",&variable1[j], &variable2[j]);
    }
fclose(mydata);
```

2.3 Literature Reference

[1] Smith, Steven W., "The Scientist and Engineers Guide to Digital Signal Processing." California Technical Publishing, Sand Diego, CA 1997-1999.