Mechtron 3TB4: Embedded Systems Design II
Tutorial 3

Name:____________________ Name:____________________

Building a digital filter using Matlab:

In this tutorial we are going to decrypt secret information using a software filter. Each group will be given a wave file that contains a secret code for that group only. A deliberate noise at some frequency is added to the file in order to disguise the information. Your task is to build a software filter using Matlab to filter out the noise so that the secret information can be revealed (hearable by headphone).

The wave file for your group (secret_code_groupX.wav, where X is your group ID) can be found in your group's subversion repository at

`https://websvn.mcmaster.ca/mt3tb4/groupX/Lab3/secret_code_group#.wav`

where # is your group ID. The type of filter we are going to build is the FIR filter that was discussed in the class. Matlab has a set of DSP functions that enables us to do this easily. Please follow the instructions and fill in the blanks:

Start Matlab. The following commands perform the filtering; before you execute them, read the help files of these functions by using "help FunctionName" to understand their use.

```
% read the .wav file - replace # by your group number!
% Variable x stores the wave and fs stores the sampling rate

[x,fs] = wavread('AbsolutePath\secret_code_group#.wav');

% perform FFT on the original signal to determine the frequency
% of the ‘noise’
L=length(x);
NFFT = 2^nextpow2(L);
X=fft(x,NFFT)/fs;

% Show the sampling rate.
fs

% We know the sampling rate is __________

%

% We need now to plot our FFT to find the source of the noise.
% Plot single-sided amplitude spectrum.
f = fs/2*linspace(0,1,NFFT/2+1);
plot(f,2*abs(X(1:NFFT/2+1)));
```

```
% reading the FFT we realize that the frequency we want to
% remove is _________ (hint: our noise is a pure sign wave)
% Now specify the frequency you want to eliminate by setting

fkill=_______________;

% Hint: fkill is always in the range 0 - 1, and is
%        normalized to frequency fs/2


% Determine coefficients of the FIR filter that will remove
% that frequency.
% Start off the following blank with the value 4.
% Note the following filter only works with even numbers.

coeff=firgr(_____,[0,fkill-.1,fkill,fkill+.1,1],[1,1,0,1,1],
                                    {'n','n','s','n','n'});

%Plot the filter

% Plot the frequency response of the designed filter to
% verify that it satisfies the requirements

freqz(coeff,1);

% You should try different filter lengths in the firgr command
% and find out which one is the best. Filter length of 4 is terrible
% ideally your filter should only filter out the noise while passing
% all other signals. Try increasing your filter length until you
% can achieve an adequate result.
% Be sure to plot (with freqz) each time you create a new filter.
% If you pick a filter length too big the filter will "blow up".
% If you are unsure whether your filter is blown up or not,
% seek help from a TA.

coeff * 32768

% coeff * 32768=______________________________________________________

%Please record these numbers. You will need them when doing lab3.

% filter the input signal x(t) using the designed FIR filter to get y(t)

y = filtfilt(coeff, 1, x);
```

```
% perform FFT on the filtered signal to observe the
% absence of frequency of the ‘noise’
Y=fft(y,NFFT)/L;

% play the original (x) and filtered (y) file to hear the difference

wavplay(x,fs);

wavplay(y,fs);

% The secret code for your group is ______________

% create the vector for horizontal axis

% create two plots

subplot(2,1,1);

%first one shows FFT of the original signal

plot(f,2*abs(X(1:NFFT/2+1)));
xlabel('Frequency (Hz)')
ylabel('|X(f)|')

subplot(2,1,2);

% second one shows FFT of the filtered signal

plot(f,2*abs(Y(1:NFFT/2+1)))
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')

% write the file to disk replacing # by your group number!

wavwrite(y, fs, 16,'AbsolutePath\secret_code_broken_group#.wav');
```