EECE72425 DSP Assignment - Lab 4 – Phase, FIR Filters and Convolution

Due: Monday 30 March 2020 at 11:59 pm. Submit required document to eConestoga dropbox.

This assignment consists of two parts:

A - Phase characteristic investigation with MATLAB

B - FIR Filter Design.

Part A

Use MATLAB to compare the following 4 waveform sums (use 8 kHz, 512 samples)

Frequency	**Amplitude**	Set A: phase angle	Set B: phase angle	Set C: phase angle	Set D: phase angle
80	1	0	-30	-30	-30
240	0.3333	0	-30	-90	-60
400	0.2	0	-30	-150	-120
560	0.1429	0	-30	-210	-150

1. [12] Hand in 4 plots – 1 for each set of waveform sums
2. [9] Calculate the delay for each waveform
3. [4] Question to answer: Which set exhibits linear phase?

Submit a word document to eConestoga incorporating these items

Part B

We have studied the mathematical operation of convolution in class and performed convolution by hand. We have also learned that an FIR filter is performing a convolution of the filter kernel with the input sequence to generate the output sequence. This part of the assignment has three sub sections:

1. [10] Design a pair of FIR bandpass filters using FilterDesigner tool in MATLAB to pass the 2 component frequencies of your DTMF tone. Submit filter design specifications, frequency domain performance plot.
   (**your tone is the 4th digit of your student ID number**).

2. [40] Test the performance of these filters in MATLAB by convolving the filter kernel with signal vectors you create. See the "Hints" section on the next page for guidance. Submit time domain plots from MATLAB for ***EACH*** filter:
   - Filter passing its design frequency when the input is only that frequency
   - Filter blocking the other frequency from your DTMF tone when the input is only that other frequency
   - Filter passing its design frequency when the input is the DTMF tone containing both frequencies

   Also include plots of the 3 input signals used in the above test.

Filter specifications:
- FIR bandpass filters
- 8 kHz sample rate
- High-group frequency
  i. +/- 25 Hz
  ii. Use Kaiser design technique
  iii. 30 dB attenuation to adjacent high-group frequency
- Low-group frequency
  i. +/- 10 Hz
  ii. Use Equiripple (Parks-McClellan) design technique
  iii. 30 dB attenuation to adjacent low-group frequency

Hints: Testing Your Filter Design in MATLAB

Here are some instructions for testing your filter.

1. **Design the filter** using filterDesigner and export the filter coefficients to the workspace using **File|Export -> to workspace,** as coefficients.

2. **Load the DTMF tone** that you want to process using the command:
   load –ascii 'filename'
   This will create a variable with the name of the file (less any extension). For example, if you wanted dtmf 9, you'd load –ascii 'dtmf_9.samples' and you'd end up with a variable called dtmf_9 containing the sample data points

   Alternatively, you can use the MATLAB functions you wrote for assignment 2 to generate the signals for the DTMF tone.

3. **Convolve** the input signal sample with the filter coefficients using the command:
   out = conv(Num, filename)
   In the above example, it'd be out = conv(Num, dtmf_9)

4. **Plot the output** using plot(out). Edit the Y axis so that it always goes from -2 to +2 (the amplitude of the input signal).

5. Also **plot the source signal** sample plot(dtmf_9) so we can see what we're starting with.

NOTE: One of the important things to notice is that the filter doesn't produce useful output data until it's filled with real sample data. Therefore, the first and last <filter kernel size> samples of the output sequence are garbage, since the actual sample data isn't filling the kernel – it's padded out with 0's.