Lab 5 - IIR Filter Design and Analysis

Section	L01	L02	L03	L04
Lab Date	Nov. 27	Dec. 4	Nov. 28	Dec. 5
Due Date for Report	Dec. 6	Dec. 6	Dec. 6	Dec. 6

Assessment: 5% of the total course mark.

Objectives:

- To gain experience in IIR filter design and analysis in `MATLAB` and TMS 320 DSP processor

Assessment:

- Your grade for this lab will be based on your ability to use filter design tools from the `MATLAB` Signal Processing Toolbox and to analysis and implement the resultant filters within `MATLAB` and the TMS 320 DSP processor, and on your reporting of the results.
- Clearly label all plots and their axes (points for style will be deducted otherwise).
- Please attend the lab section to which you have been assigned.
- You can complete this lab with one lab partner.
- By the end of the lab session you must demonstrate to your TA the MATLAB code and the filter output on `MATLAB` and the TMS 320 DSP processor.
- Each pair of students should complete one lab report together. The source code and the report have to be submitted by **11:59 pm on Dec. 6, 2023**. One of the group members can submit the source code and the report.

Pre-lab:

- Carefully read through this lab description so that you know what is required.
- Read through the lecture notes so that you know how to answer the questions.
- Familiarize yourself with the `MATLAB` commands that may be required for this lab – see the list at the end of this lab description for some hints.

Experiments:

1. **Chebyshev Type II lowpass IIR filter**

   (a) Design linear-phase lowpass **FIR** filters with the cutoff frequency $\omega_c = \frac{\pi}{3}$ using the window method with a Dolph-Chebyshev window for filter orders $M = 20, 50$ and 150. The `MATLAB` Signal Processing Toolbox function `fir1()` can be used to implement the window method and the `MATLAB` Signal Processing Toolbox function `chebwin()` can be used to obtain the desired Dolph-Chebyshev windows. Note that you will need to specify the amount of sidelobe attenuation for your Dolph-Chebyshev windows. Try some different values, and demonstrate what effect this has on:

      i. the passband ripple and stopband attenuation.

      ii. the slope of the transition region.

   for <u>one</u> of the filters ($M = 20, 50$ or 150).

(b) Design Chebyshev Type II lowpass **IIR** filters with the cutoff frequency $\omega_c = \frac{\pi}{3}$ using the `MATLAB` Signal Processing Toolbox function `cheby2()` for filter orders $M = N = 20, 50$ and 150. Note that you will need to specify the stopband attenuation – make it the same as one of the values that you used for the Dolph-Chebyshev window sidelobe attenuation in part (a).

(c) Determine the impulse responses of these filters. Do they appear to be infinite?

(d) Compare and contrast the IIR filter frequency responses (both the magnitude and the phase) with the FIR frequency responses from part (a). Explain the differences in light of filter theory as it relates to causal FIR and IIR filters.

(e) Create a signal 1000 samples long of white Gaussian noise. Using the `filter()` function, calculate the output of the IIR filter for this signal. Now estimate the output by convolving the input signal with a very long impulse response that you obtained from part (c). How good is the estimate?

2. **Experiment on the TMS 320 DSP processor: IIR filter design and group delay equalization**

(a) A sample IIR DSP program is placed on the Y-drive for your reference. This program consists of a **lowpass** Butterworth filter implementation with cutoff frequency $\omega_c = \frac{\pi}{2}$ and filter order $M = 10$. The sampling frequency of the AIC23 codec on the TMS320C6713 DSK board is set to 8 kHz by default. The filter coefficients are generated with the following `MATLAB` code.

```
[b, a] = butter(10,0.5,'low');
```

Run this IIR DSP program and test it with the input signal obtained with the following `MATLAB` code.

```
fs = 8000; f1 = 1500; f2 = 750;
L = 8000*5; n = (0:L-1);
A = 1/5; % adjust playback volume
x1 = A * cos(2 * pi * n * f1 / fs);
x2 = A * cos(2 * pi * n * f2 / fs);
x = x1 + x2;
sound(x, fs);
```

Use the `grpdelay()` function to plot the group delay response of the IIR filter, then:

i. Find the group delay (in samples) for both f_1 and f_2, and calculate the difference between them.

ii. Compare the input signal with the filter output on the oscilloscope, describe what you observed and explain the possible reasons.

(b) To improve the output response of the Butterworth filter designed in part (a), you need to design a group delay equalizer section (an allpass filter), so that when it is cascaded with the Butterworth filter section, the overall group delay response is as flat as possible. The `iirgrpdelay()` function can help you design the allpass filter with desired group delay response. Test your allpass filter with following steps:

i. Plot the magnitude response and group delay response of the overall structure (a Butterworth filter cascaded with an allpass filter, the `MATLAB` function `conv()` can be used for cascading). Plot the input signal, filter output of the unequalized filter alone and of the overall structure on the same figure using `MATLAB`. Comment on the results.

ii. Save your filter coefficients in ascii format. The `MATLAB` command `dlmwrite()` is very useful for this purpose, it saves the coefficients in a .txt file as a row vector, separated by commas. In the following C program, replace the value of `M2` and elements of arrays `a2` and `b2` with your allpass filter order and coefficients, respectively.

```
// Delay equalization coefficients, no delay for default settings
#define M2 1 // allpass filter order
double a2[M2+1] = {  1.0, 0 };
double b2[M2+1] = {  1.0, 0 };
double d2[M2] = {  0  }; // delay samples
```

Run the IIR DSP program and test it with the input signal used in part (a). Observe the input signal and the filter output on the oscilloscope. Does the actual group delay of the output match the group delay you plotted in part (b) i.? Assess the flatness of the group delay characteristic of the overall structure.

Note: Check out the demo video to learn how to use `iirgrpdelay()` and `dlmwrite()`.

REPORT: The report should contain

- Any mathematical calculations or derivations carried out
- `MATLAB` plots of results with brief descriptions
- Answers to questions

You **do not** need to include the `MATLAB` code in the report. However, you have to submit the `MATLAB` code separately.

POTENTIALLY USEFUL `MATLAB` COMMANDS:

Note that this is not an exhaustive list! You are not required to incorporate all of these in your scripts.

help topic	helpwin	figure	plot	stem
histogram	subplot	hold on	xlabel	ylabel
legend	title	function	clear	close
clc	zeros	ones	cos	exp
abs	round	max	min	find
if	for	end	real	imag
angle	unwrap	phase	audioinfo	audioread
audiowrite	soundsc			