Lab3,EE4C5,DSP

Prof. Naomi Harte, Dr Jean-Baptiste Monteil

October 20, 2023

1 Introduction

In this lab, we study the properties of multiple filters by looking at their magnitude and phase responses. In lectures, we did an introduction covering the functionality of the filter designer in MATLAB. If you need a refresher, start with the help function inside MATLAB or go to:
https://uk.mathworks.com/help/signal/ug/introduction-to-filter-designer.html.
Note about capturing visual output from FilterDesigner: The tool does not have an easy way to quickly capture a Figure. In a Windows environment, you can press the Windows logo key + Shift + S and engage the snipping tool to make your life very easy.
If you are ready to get started, type `filterDesigner` at the MATLAB command prompt.

1.1 IIR filters

1. Design a minimum order, stable, low-pass Butterworth filter with a passband frequency of 1 kHz and a stopband frequency of 2 kHz. Assume a sampling frequency of 10 kHz. Make the attenuation 1 dB at the passband frequency and 80 dB at the stopband frequency. Plot the magnitude and phase responses and record them in your write-up. What is the order of the lowpass Butterworth filter you designed? Comment also on whether the phase is as you would expect for an IIR filter.
2. Change (both narrow and then widen) the transition region. Do you observe any changes in the magnitude response? How does it affect the filter order? Comment specifically on the trade-off you observe between the transition width and filter order.
3. Consider a Butterworth filter with the following specifications for its discrete-time frequency re-

sponse:

$$0.89 \leq |H(e^{j\omega})| \leq 1, \quad 0 \leq \omega \leq 0.2\pi \tag{1}$$

$$|H(e^{j\omega})| \leq 0.18, \quad .6\pi \leq \omega \leq \pi \tag{2}$$

What is the minimum order filter? Create this filter in `filterDesigner` to verify your theoretical result. Show the magnitude response in your write-up.

4. Can a low pass Butterworth filter of order 4 meet the following specifications: 3 dB cutoff frequency of 1.5 kHz and attenuation of 40 dB at 3 kHz? If not, what order would be sufficient?

5. Include in your report the pole-zero plot of the Butterworth filter of question 4. Does it correspond to the poles that you can find with the impulse invariance method? *Hint:* use solution from Tutorial 3 question 1.

6. Design a minimum order, stable low-pass Chebyshev Type I filter with the same specifications as in question 1. Compare the magnitude and phase responses to the Butterworth filter in question 1, including the plots in your write-up. What is the order of the Chebyshev filter?

7. What are the pros and cons of the Chebyshev filter in comparison with the Butterworth filter? Consider filter order, passband response, stopband response, transition region, phase response and any other factor you think relevant. Use your designed filter example to give specific examples to support your points.

1.2 FIR filters

1. Design an FIR filter with the windowing method to meet the specifications of question 1 in the previous section. Use a suitable Hamming window. What filter order is needed for this filter? Include the resulting magnitude and phase responses in your report. Comment on how the phase differs from the IIR filter. Overall, are the characteristics of the FIR filter as good? Better? Worse? Examine the filter impulse response. Is this Type I, II III or IV? Explain.

2. Design an FIR filter with the windowing method to have 40 dB attenuation before 15 kHz. Assume a sampling rate of 48 kHz. Give your filter specifications in your write-up. Start with a Bartlett window. Increase the filter order until you are satisfied with the trade-off between response and filter order. What filter order is necessary? Include the observed magnitude response in your report and comment on how close it is to the desired response. Now try a Kaiser window with the same order. How does the magnitude response compare? Can you tune the Beta parameter to improve the response? In-

clude this response in your write-up to support your comments. Show the impulse response and phase response. Comment on their characteristics.

3. Select any type of FIR filter that we have covered in lectures. By using the following formula:

$$N \approx \frac{A_{dB} F_s}{22 \Delta f}, \tag{3}$$

where A_{dB} is the stopband attenuation in dB, F_s is the sampling frequency, Δf is the transition bandwidth, specify the filter order to have the following characteristics:

- the input signals may contain frequencies from $0 - 20$ kHz;
- the filter only passes frequencies less than 10 kHz;
- the filter achieves a minimum of 40 dB attenuation at 15 kHz;
- $F_s = 50$ kHz.

Verify that the approximation of N is valid by quickly designing your filter in MATLAB. Comment on your findings in your write-up.

1.3 Restore a speech file with a suitable filter

In this part, we provide a speech file called *'OSR_us_000_0010_8k.wav'* in Blackboard Lab 3 folder. We superimpose a tone with the help of the MATLAB script `create_corrupted_speech.m`, which ultimately corrupts the file. The goal is to recover the original speech file by applying a filter that you design. It can be any type of filter from those covered in this module. You will need to think about the high-level characteristics of the filter (IIR, FIR, lowpass/highpass etc) first. Then think about the response.

You can generate MATLAB code that constructs the filter you designed in the filter designer. Then you can either use that function from the command line or call it in another .m file. Select **File** → **Generate MATLAB Code** → **Filter Design Function** and specify the filename in the Generate MATLAB code dialog box. If you aim to filter a given signal with the filter you just created, use the following code:

```
filtered = filter(my_butterworth,signalToBeFiltered) %you apply
    the filter you created (named my_butterworth for instance) to
     the signal to be filtered.
```

1. Corrupt the speech file with a note number of 105, by calling the provided MATLAB function `createNote.m`. The code of the script `create_corrupted_speech.m` is incomplete and thus you must complete it to corrupt the speech file. Compute the MSE between your corrupted speech file and the original file as a basic* metric of similarity. **Warning: Be careful if you listen to the corrupted speech file, the note number 105 is very high and is uncomfortable to listen to. So put a low volume and choose a low amplitude for the tone.**

2. Recover the original speech file with a filter of your choice. Have you reduced the MSE with this filtered speech file? Document your filter parameters and resulting MSE as you adjust your filter design to try and improve the output. Try up to 10 different filters. Additionally, listen to the output to assess the performance of your filter. Fully detail the specifications of your final choice of filter and justify why you selected these specifications in the report.

```
function Hd = gen_butterworth
%This is an example of typical code the FilterDesigner
%can auto-generate
%gen_butterworth Returns a discrete-time filter object.

% Butterworth Lowpass filter designed using FDESIGN.LOWPASS.

% All frequency values are in Hz.
Fs = 10000;  % Sampling Frequency

Fpass = 1000;              % First Passband Frequency
Fstop = 3000;              % Second Stopband Frequency
Apass  = 1;  % Passband Ripple (dB)
Astop = 80;   % Second Stopband Attenuation (dB)
match = 'stopband' %Band to match exactly


%Construct an FDESIGN object and call its BUTTER method.

h = fdesign.lowpass(Fpass, Fstop, Apass, Astop, Fs);
Hd = design(h, 'butter', 'MatchExactly', match);
```

```
% [EOF]
```

Optional: (stretch goal - you can set up your code to sweep through various parameter settings and automatically report back on the resulting MSE if you are feeling ambitious)
3. Do the same steps with a deeper super-imposed tone (note number lower than 40). Can you perfectly recover the original signal in this case?
4. Do the same steps with two super-imposed tones (one deep with a note number lower than 40, one high-pitched with a note number higher than 60).
* the MSE is not a great metric of similarity for speech in reality. It is overly sensitive to time misalignments. Metrics for speech quality assessment and similarity of speech to some clean "reference" is an entire body of research in speech processing, and beyond the scope of 4C5.

2 Pre-clinic submission

Before your assigned clinic, you must upload a record of your "work in progress" on the lab material. This is rough-work and can include e.g. initial code, cut'n'paste from the MATLAB commandline, initial figures or draft code. This submission is required whether or not you attend your clinic. It is not marked, but is a record of your progress before the clinic.

3 Write-up for (post-clinic) submission

The purpose of this lab is that you engage with the assigned tasks, get a better appreciation of the real-world analysis of signals, and make connections between what you are learning in lectures and practical use of those concepts. Therefore, you should not be spending a huge amount of time on the write-up. The purpose of the submission is to show you have done all the assigned tasks, that you have thought about the questions asked, and that you are achieving the required level of proficiency in MATLAB.
With this in mind, you should submit:

- A single pdf with all required figures and commentary on the tasks in the lab. Bulleted comments are allowed.
- The pdf above must use a sans serif font, and adequate font size and spacing.

- You must include the declaration on plagiarism as per your course handbook.
- You can use snippets of code in the pdf to illustrate a point you wish to make, but all code must be submitted in .m files.
- Ultimately, we should be able to run your code and recreate your plots etc from what you submit.
- Include all relevant wav files, giving them informative names
- All code should be commented.

Note: All deadlines are detailed on Blackboard, and students must make themselves aware of when items are due.