Electrical & Computer Engineering 434

Computer Exercise 2

This assignment may be completed with the assistance of other students (remember to properly document any assistance received). However, each student must turn in an individual item that consists of their own work. You are not allowed to tell other students the filter orders you achieved, the filter types you chose (IIR or FIR; Chebyshev, Elliptic, Equiripple, etc; LPF, HPF, BPF, BRF), your filter specs (cutoff frequencies, pass or stopband levels, pole-zero locations, etc) and the contents of any messages you've discovered. You are allowed to teach other students how to use the tools, like the pole-zero editor and remez function, but cannot give them solutions.

You are also allowed to consult online sources, like Mathworks, Google, and generative AI (e.g., ChatGPT), but the work you turn in must be your own work. Be sure to document how you used the online resources, and for generative AI, document the "prompts" you used.

This assignment is due ~~1630~~ 2359 on lesson M31.

Important: Turn in your report via e-mail or Teams to your professor as an MS Word document or pdf (use file name **`"cpx2_lastname.doc"`**) and turn-in your matlab file "**cpx2_lastname.m**" and any support files, such as your filters (i.e, filt8, filt3, etc). You can use the save command to save them to a file, and the load command in your matlab script to load the filters.

1 Introduction

1.1 Objective

Learn how to design and use digital filters.

1.2 Description

For this exercise, use your judgment and the knowledge you've gained so far in ECE434 to analyze and filter the given signals using MATLAB. Employ any basic DSP techniques with which you are familiar, including the FIR and IIR filter design tools available in MATLAB. You have *no* way to obtain any more data points for each of the given signals.

Given: You are given a file named cpx2.bin with several signals, and are in a binary format MATLAB data file—Your instructor will email you a file with your unique signals.

Bring the data into the MATLAB workspace by using the command `load cpx2.bin -mat`. The signals you need are:

Name	Description
Fs	The sampling frequency in hertz for *x1*
Fs_x8	The sampling frequency in hertz for *x8*
Fs_x3	The sampling frequency in hertz for *x3*
Fs_x4	The sampling frequency in hertz for *x4*
Fs_x7	The sampling frequency in hertz for *x7*
x1	Data vector for signal x1
x8	Data vector for signal x8
x3	Data vector for ECG signal x3
x4	Data vector for noisy voice signal x4
x7	Data vector for 10 test tones in signal x7

If you have trouble loading these data files, send an e-mail to your professor ASAP.

2 CompEx Requirements

You must successfully process the 5 required signals to this computer exercise, along with a brief write-up described in Section 3. You will send only the overall report in a single MS Word file or pdf file containing your write-up to your professor. The report should specify your filter designs (with enough information that your instructor can re-create the filter) and rational for selecting the filter you choose (IIR vs FIR; filter choice; LPF, HPF, BPF, etc; Cheby, Elliptic, etc). The results will be evaluated in terms of correctness of approach, achievement of the design objectives, and design efficiency (minimizing the number of multiplies, adds, and memory). Be sure to include filter designe (from *filterDesigner* tool) and the achieved the filter order for each of your filter designs. Hint: Before designing the filters, it will be useful to determine the spectrum of the signals.

1. **Data vector x1** is the same as signal x2 from CPX1, but with many more samples. Implement a digital filter of your choosing to attenuate the higher magnitude of the two sinusoids such that it is reduced to ≥ 30 dB lower in magnitude than the other sinusoid. Assume preserving linear phase is **not** important.

   For this signal:
   What is the period of the dominate sinusoid **before** and **after** filtering? Can you measure this in the time domain and frequency domain? Does this make sense?
   Use the *sound()* command the hear both the original and final signal, and explain their differences. Is this what you expected from the time and frequency plots?

2. **Data vector x8** is an audio signal you can listen to with the *sound()* command. The signal starts off clean but is then jammed with one or more tones. Your job is to remove (or minimize) the jamming and restore the original signal with a minimum order filter. You can use sptool's (or filterDesigner) to design the filter (maybe pole-zero editor?) to create the filter to remove the one or more tones, similar to one of the filter's your instructor demonstrated in class. It is okay if your final solution retains two "pings" in the recording, but if you can also remove these you'll get some bonus points. Assume preserving linear phase is **not** important. You cannot use *iircomb* to design the filter.

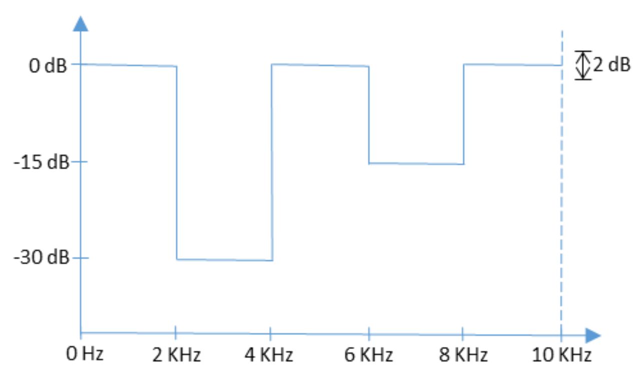
   For this signal:
   (1) explain what you initially hear (how much of the message you can transcribe before the jamming?)
   (2) plot and explain what you see in the initial time domain and in the frequency domain
   (3) explain your approach to remove the jamming signal [justifying IIR vs FIR, Filter Design, Choice of filter]
   (4) give the filter design specs [minimum order gets most points]
   (5) what is the complete message in the signal?
   (6) plot and explain the final output time and frequency domain plots

3. **Data vector x3** contains an actual sampled electrocardiogram (ECG) signal from a human subject, obtained from a clinic in Europe. [Your I&J textbook discusses ECG on pages 30-33, 628-31, 673, 878-81] No aliasing has occurred, but the data has been contaminated with the local power line artifact. Your objective is to mitigate this power

line contamination while minimizing any detrimental effects on the ECG signal itself. *Can you find the power line noise in the frequency plot, and can you also measure/find it a zoomed time domain plot?* Assume that a cardiologist will make decisions regarding this patient's prognosis based on this filtered signal. Important diagnostic information exists in nearly all of the bandwidth from 0 Hz to Fs/2, so your filtering efforts should minimize any effects on signal energy outside the power line artifact. Note that **you wish to preserve the phase relationships** of the many frequencies that make up the ECG signal. Implement a digital filter of your choice to reduce the power line artifact such that you restore the ECG signal to as close to what it would have been had there been no power line contamination. This means you don't want to have too much attenuation in the stopband, nor too little. Hint: observe the spectrum of the ECG signal and estimate what the signal "should" probably be at the frequency band you are filtering. Observe the spectrum of the signal after you've filtered it and see how well you did. This may take a few iterations. How does the signal look in the time domain before and after filtering (zoomed in to one pulse)? **One more thing**: include in the write-up your estimate of the patient's **average heart rate** when this ECG signal was obtained. That information is definitely in the signal. Estimate the heart rate using the **time domain plot**, then verify this frequency occurs in the **frequency domain plot**. What frequency is the heart rate?

4. **Data vector x4:** Using an appropriate filtering scheme, determine what the intelligible audio signal is in x4. Since it is an audio signal, **preserving linear phase is important**. Hint: Looking at both the log and linear frequency plot may be helpful.

5. **Data vector x7:** Design a multiband filter with the ideal specs shown below with a goal of minimum order to meet these specs, where Fs = 20 KHz. (*lesson25.m* might be helpful). You must use the matlab *remez* (or *firpm*) function and create one filter (one h[n]) to implement this multi-band filter (i.e., you can't use two separate band reject filters)



Given the h[n] from this design, filter x7[n] with this multiband filter creating the output y[n], and plot the output spectrum of Y[f] in a log plot (dB). x7 has 10 sinusoids to test each band of the multiband filter. Measure your error E[f] in dB for each of these 10 frequencies to an accuracy of at least 4 significant figures:

$$\mathrm{E[f]} = |Y[f] - Y_d[f]|$$

Where Y_d[f] is the desired dB of each tone (e.g. 0dB, -15dB, -30dB, etc). Then calculate the overall *average E[f]* as well as the overall *max(E[f])*. Your goal is to not have any of the 10 E[f] exceed 1 dB. A spreadsheet might be useful with a nice table in the report. For example, here is an example:

Table 8. Error for N=100 Remez Filter

Frequency (kHz)	Y[f] (dB)	Y_d[f] (dB)	E[f] (dB)
.5	-0.000269	0	0.000269
1.5	-0.0006964	0	0.0006964
2.5	-30	-30	0
3.5	-30	-30	0
4.5	-0.0006963	0	0.0006963
5.5	0	0	0
6.5	-15	-15	0
7.5	-15	-15	0
8.5	-0.0006964	0	0.0006964
9.5	-0.0003309	0	0.0003309

Average Error: 0.0002689 dB

Max Error: 0.0006964 dB

Write-Up Requirements

The write-up is *not* to be a dissertation—be concise. Please don't misinterpret that to mean you can just throw together a sloppy, incomplete write-up. Your write-up grade will be based on solution completeness, succinctness, organization, evidence of insight into DSP topics, *and* use of the English language. Also remember that your instructor likes to see plots of the signal in both the time and frequency domain, before and after processing, zoom in and zoomed out, linear and log scale, etc, if helpful. Plots should be versus time and frequency, not plotted versus n and k, respectively. Frequency plots should be normalized such that the max frequency is "1" in a linear plot and 0 dB for the log plots.

1. For *each* of the 5 required signals of the Computer Exercise that are specified in Section 2, address the following topics:

   (a) What is the goal you are tasked to achieve for this signal?
   (b) To meet the given design objectives what filter choice (IIR vs FIR; filter choice, like Cheby, Elliptic, etc; LPF, HPF, BPF, etc) and filter specifications did you choose (your professor should be able to duplicate your design based only on the specifications you provide);
   (c) What techniques did you try (what worked and what didn't—be specific);
   (d) How well did you meet the design objectives;
   (e) What significant thing did you learn or observe?

2. Your paper should have an introduction as well as a conclusion. Summarize in just a few sentences what you learned from the entire process.

Grading criteria is summarized in the file "CPX2 Grading Criteria for students.pdf" on the course blackboard assignment page. Also see "CPX2_FAQ.pdf"

Don't wait until the last minute to do this!
Have fun...