

Homework #3

Child of Filter Analysis, Simulation, and Design

Assigned on Saturday, February 22, 2014

Due on Friday, February 28, 2014, by 11:00 am sharp in class

Homework submitted after 11:00 am will be subject to a penalty of 2 points per minute late.

Reading: Johnson, Sethares & Klein, chapters 4-6, and Appendix B

This assignment is intended to review frequency-domain analysis of LTI systems and provide an introduction to the simulation and design of infinite impulse response filters.

Here are key sections from Lathi's *Linear Systems and Signals* book (2nd ed) and Oppenheim & Willsky's *Signals and Systems* book (2nd ed) with respect to material in EE 445S:

<i>O&W</i>	<i>Lathi</i>	<i>Topic</i>
1.6	1.7	System properties
1.3 – 1.4	1.4	Basic continuous-time signals
3.2 ##	2.4-4	Fundamental theorem for continuous-time linear systems **
1.3 – 1.4	3.3	Basic discrete-time signals
3.2 ##	3.8-3	Fundamental theorem for discrete-time linear systems **
9.7.2	2.6	Stability of continuous-time filters
10.7.2	3.10	Stability of discrete-time filters
10.1 – 10.3	5.1	Z transforms
10.5	5.2	Properties of the z-transform
10.7.3 – 10.7.4	5.3	Transfer functions
10.8	5.4	Realizations of transfer functions
4.3 – 4.4	7.3	Fourier transform properties
7.1	8.1	Sampling theorem

** Please see Appendix F and slide 5-13 in the course reader for the fundamental theorem.

O&W covers a slightly different version of the fundamental theorem in which a complex exponential is the input to a linear time-invariant system. Lathi also has that version as well.

Other signals and systems textbooks should contain equivalent material.

You may use any computer program to help you solve these problems, check answers, etc. ***Please submit any MATLAB code that you have written for the homework solution.*** In the course reader, Appendix D gives a brief introduction to MATLAB. The MATLAB code in the Johnson, Sethares and Klein book also runs in LabVIEW Mathscript and GNU Octave.

As stated on the course descriptor, “Discussion of homework questions is encouraged. Please be sure to submit your own independent homework solution.”

Office hours for the teaching assistants and Prof. Evans; **bold** indicates a 30-minute timeslot.

<i>Time Slot</i>	<i>Monday</i>	<i>Tuesday</i>	<i>Wednesday</i>	<i>Thursday</i>	<i>Friday</i>
9:30 am					Jia (ENS 137)
10:00 am					Jia (ENS 137)
10:30 am					
11:00 am	Evans (ETC 5.148)		Evans (ETC 5.148)		Evans (ETC 5.148)
12:00 pm	Evans (ETC 5.148)		Evans (ENS 433B)		Evans (cafe)
12:30 pm				Evans (ENS 433B)	Evans (cafe)
1:00 pm				Evans (ENS 433B)	Evans (cafe)
2:00 pm				Evans (ENS 433B)	
2:30 pm					
3:00 pm			Sinno (ENS 137)		
3:30 pm			Sinno (ENS 137)	Jia (ENS 137)	
4:00 pm			Sinno (ENS 137)	Jia (ENS 137)	
4:30 pm				Jia (ENS 137)	
5:00 pm				Jia (ENS 137)	
5:30 pm				Sinno (ENS 137)	
6:00 pm				Sinno (ENS 137)	

The Matlab scripts that accompany the Software Receiver Design book also run in GNU Octave, which is free software. Please see the note on page vii of the SRD book for more information.

3.1. Using Finite Impulse Response Filtering to Improve Signal Quality. 18 points.

Johnson, Sethares & Klein, exercise 4.21, on page 78.

3.2. Quadrature Amplitude Modulation. 32 points.

Johnson, Sethares & Klein, exercise 5.14, parts (a) and (c), on page 92.

3.3. Infinite Impulse Response Filter Design. 50 points.

This problem asks you to design a discrete-time digital IIR filter for an electrocardiogram (ECG) signal. *This problem is a sequel to homework problem 2.3.*

An ECG device “monitors the cardiac status of a patient by recording the heart’s electrical potential versus time. Such devices play a very important role to save life of patients who survive heart attack or suffer from serious heart diseases. The time to respond to a heart attack is very critical for these patients. An early detection of conditions that lead to the onset of cardiac arrest allows doctors to provide proper treatment on time and prevents death or disability from cardiac arrest.” [1]

"There exist three types of noise that contaminate the ECG signal: the baseline wander noise (BW), electromyographic interference (EMG), and the power line interference. The BW is induced by electrodes’ changes due to perspiration, movement and respiration, and is typically below 0.5 Hz. The power line interference either 50 Hz or 60 Hz and its harmonics are a significant source of noise." [1].

Our goal in designing the filter is to attenuate baseline wander noise and powerline interference. It would take more sophisticated processing to track and cancel the EMG noise because EMG noise appears in the same frequencies as the ECG signal.

Here are the bandpass filter specifications for your design:

- For frequencies 0 Hz to 1 Hz, the stopband attenuation should be at least 40 dB.
- For frequencies 6 Hz to 40 Hz, the passband ripple should be no greater than 1 dB.
- For frequencies above 45 Hz, the stopband attenuation should be at least 40 dB.

These specifications would be compatible with the monitor mode in modern ECG monitors.

Please use a sampling rate of 200 Hz. From a quick survey of commercial ECG systems, I found sampling rates that vary from 100 Hz to 1000 Hz. The PTB Diagnostic ECG Database uses a sampling rate of 1000 Hz and the QT ECG Database uses a sampling rate of 250 Hz.

- (a) Design IIR filters using the Butterworth, Chebyshev type I, Chebyshev type II, and Elliptic (Equiripple) design methods. For each design method, find the filter of smallest order to meet the specifications. The filter order is the number of poles. Turn in plots of the magnitude and phase responses for each IIR filter you have designed to meet the specifications. **24 points.**
- (b) List the filter orders required for filters to meet the specification. Which IIR filter family gives the lowest filter order? Plot the pole-zero diagram for the IIR filter with shortest filter order you designed in (a). Because the poles and zeros are close to the unit circle and separated in angle, poles indicate the passband and zeros indicate the stopband. Submit the feedforward and feedback coefficients for the elliptic filter. **6 points.**
- (c) Use the Elliptic filter you designed in (a) for this part. Assuming that the input data samples and the IIR filter coefficients are stored in single precision IEEE floating point format,
 1. How many instruction cycles on the TMS320C6700 DSP family would it take to compute one output value for each input value if the IIR filter routine were handcoded in assembly for optimal performance? **6 points.**
 2. How much storage in bytes would it take to store the IIR coefficients and the circular buffer for the current and past inputs and the circular buffer for past outputs? **6 points.**

(d) How would you advocate using either an FIR filter or IIR filter for this application? In your answer, please include the design method for which you are advocating. **8 points.**

A rule-of-thumb for filter design is that the distance from the (maximum) passband frequency to the (minimum) stopband frequency should be at least 10% of the passband bandwidth.

Feel free to use Matlab, LabVIEW, or another program on this problem. In Matlab, I'd recommend using the filter design and analysis tool, *fdatool*. This tool is particularly useful when exploring different filter structures for implementation. LabVIEW has several IIR filter design demonstrations among the filter design toolkit VIs, e.g. *Classical Filter Design* in the Digital Filter Design toolkit.

Hints

(a) *The automatic order calculations for an IIR filter to meet piecewise magnitude specifications are very accurate. I don't recall seeing an incorrect calculation before.*

By looking at the magnitude response plots, validate that the designed filters meet the specifications. In particular, be sure to check the graphical views of the passband and stopband responses, i.e. zoom in on the magnitude response near the passband frequency and near the stopband frequency. If you are not absolutely sure from visually inspecting the magnitude response, then you can compute the magnitude response at a particular frequency using the `freqz` function in Matlab. The first two arguments are the numerator coefficients of the transfer function and the denominator coefficients of the transfer function. Type `help fdatool` to find out how to export the filter design to Matlab. Then, you can take the magnitude of the result. Finally, you'll need to convert the magnitude to dB using

$$20 \log_{10} \text{magnitude}$$

(c) *Since an IIR filter could be calculated as a tapped delay line applied to current and previous input samples, and a tapped delay line applied to previous output samples, Appendix N in the course reader might be helpful in calculating the number of instruction cycles needed to compute one output value.*

References

- [1] Yong Lian and Jianghong Yu, "A Low Power Linear Phase Digital FIR Filter for Wearable ECG Device", *Proc. IEEE Int. Conf. on Engineering in Medicine and Biology Society*, pp. 7357-7360, 2005.
- [2] "LabVIEW for ECG Signal Processing", <http://zone.ni.com/devzone/cda/tut/p/id/6349>.