Homework #2

*Filter Analysis, Simulation, and Design*

Assigned on Friday, September 29, 2017
Due on Friday, October 6, 2017, by 11:00am sharp in class

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

*Reading:* Johnson, Sethares and Klein, chapters 1, 2, 3 and 7, and Appendices A and F

This assignment is intended to continue our review of key concepts from Linear Systems and Signals, and introduce the simulation and design of discrete-time linear time-invariant 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:

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

** 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.

<table>
<thead>
<tr>
<th>Time Slot</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:00 am</td>
<td>Evans (EER 1.518)</td>
<td>Evans (EER 1.518)</td>
<td>Evans (EER 1.518)</td>
<td>Evans (EER 1.518)</td>
<td>Evans (EER 1.518)</td>
</tr>
<tr>
<td>12:00 pm</td>
<td></td>
<td></td>
<td></td>
<td>Evans (EER cafe)</td>
<td></td>
</tr>
<tr>
<td>1:00 pm</td>
<td></td>
<td></td>
<td></td>
<td>Evans (EER cafe)</td>
<td></td>
</tr>
<tr>
<td>2:00 pm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:00 pm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30 pm</td>
<td></td>
<td>Kanawati (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:00 pm</td>
<td></td>
<td>Kanawati (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:30 pm</td>
<td>Choo (EER 1.810)</td>
<td>Kanawati (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:00 pm</td>
<td>Choo (EER 1.810)</td>
<td>Kanawati (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:30 pm</td>
<td>Choo (EER 1.810)</td>
<td>Kanawati (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:00 pm</td>
<td></td>
<td>Kanawati (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:30 pm</td>
<td></td>
<td>Choo (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:00 pm</td>
<td>Choo (EER 1.810)</td>
<td>Choo (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30 pm</td>
<td>Choo (EER 1.810)</td>
<td>Choo (EER 1.810)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: In your solutions, please put all work for problem 1 together, then all work for problem 2 together, etc.

Please read homework hints at [http://users.ece.utexas.edu/~bevans/courses/realtime/homework](http://users.ece.utexas.edu/~bevans/courses/realtime/homework)

2.1. Frequency Responses. 24 points.

For each LTI system in problem 1.1 on homework assignment #1,

a) plot the pole-zero diagram for the transfer function. 3 points.

b) is the filter bounded-input bounded-output (BIBO) stable? why or why not? 3 points.

c) give a formula for the frequency response. 9 points.

d) plot the magnitude response. 6 points.

e) if the system is BIBO stable, pick the best one of the following choices to describe the frequency selectivity of the filter: lowpass, highpass, bandpass, or bandstop. 3 points.

You may use the solution set for problem 1.1 in your solution for this problem.
2.2. Finite Impulse Response Filter Design for Audio Signals. 30 points.

This problem explores ways to process audio signals.

Please download the audio wave file ‘twosignals.wav’ from the homework Web site:

http://users.ece.utexas.edu/~bevans/courses/realtime/homework/twosignals.wav

This audio file is the sum of two audio signals—a gong sound and a bird chirping. The gong sound and the bird chirping occupy different frequency bands. The gong sound is different from the gong file from the Johnson, Sethares and Klein book.

(a) Plot the spectrum of the ‘twosignals’ audio track using plotspec and spectrogram. Approximately what frequency band does the gong sound occupy? Approximately what frequency band does the bird chirp occupy?

(b) Design an FIR filter using the Parks-McClellan algorithm (a.k.a. Remez Exchange algorithm and Equiripple design algorithm) to extract the gong signal from the ‘twosignals’ audio track. Then, apply the filter to the ‘twosignals’ audio track, play back the filter output to validate that the gong signal has been extracted, and plot the filter output using plotspec.

(c) Design an FIR filter using the Parks-McClellan algorithm (a.k.a. Remez Exchange algorithm and Equiripple design algorithm) to extract the bird chirp from the ‘twosignals’ audio track. Then, apply the filter to the ‘twosignals’ audio track, play back the filter output to validate that the gong signal has been extracted, and plot the filter output using plotspec.

(d) Take the extracted gong signal in part (b) and perform downsampling by 2. Downsampling by 2 keeps every other sample and discards the others. Here’s Matlab code for downsampling a vector vec by 2:

```matlab
vecDownsampledBy2 = vec(1:2:length(vec));
```

- Play the downsampled filtered gong signal at the same playback rate as the filtered gong signal. How does it differ from the gong signal extracted in part (b)?
- Plot the magnitude spectrum of the downsampled filtered gong signal and compare it against the magnitude spectrum of the gong extracted in part (b).

(e) Take the extracted gong signal in part (b) and perform upsampling by 2. Upsampling by 2 inserts zero after every sample. Here’s Matlab code for upampling row vector vec by 2:

```matlab
vec = cumsum( ones(1,10) );
upsampledLength = 2*length(vec);
vecUpsampledBy2 = zeros(1,upsampledLength);
vecUpsampledBy2(1:2:upsampledLength) = vec;
```

- Play the upsampled filtered gong signal at the same playback rate as the filtered gong signal. How does it differ from the gong signal extracted in part (b)?
- Plot the magnitude spectrum of the upsampled filtered gong signal and compare it against the magnitude spectrum of the gong extracted in part (b).

For the sanity of others, you might put in a pair of headphones when working this problem.

The firpm command stands for finite impulse response design using the Parks-McClellan algorithm. The Parks-McClellan algorithm was proposed by James McClellan in his 1973 PhD dissertation, which he completed at Rice University under the research supervision of Thomas Parks. James
McClellan lived in Austin 1981-1987. Since 1987, James McClellan has been a faculty member at Georgia Tech. Thomas Parks retired from the faculty at Cornell University several years ago.

Downsampling is the removal of samples in a regular fashion. The problem uses downsampling by 2. The first sample and every other sample thereafter is kept, and hence, the second sample and every other sample thereafter is removed. Downsampling by would halve the number of samples. Upsampling is the addition of samples in a regular fashion. After every sample in a signal, upsampling by 2 would add a sample of value zero. Upsampling by 2 would double the number of samples. After midterm #1, we'll make heavy use of upsampling in a communications transmitter and downsampling in a communications receiver.

2.3. Finite Impulse Response (FIR) Filter Design for Treatment of Tinnitus Loudness. 46 points.

Tinnitus, a.k.a. “ringing of the ears”, is a symptom due to an underlying condition in the auditory system. People with tinnitus hear a tone, clicking, hiss, roaring or buzzing when no external sound is present [1][2]. The tinnitus sound can be at low, medium or high audible frequencies, and may occur in one ear or both ears. The tinnitus sound might be temporary or chronic. Those suffering from chronic tinnitus would hear the same sound in the same frequency range each time.

This problem asks you to design a discrete-time filter to alleviate the loudness of tinnitus:

"Maladaptive auditory cortex reorganization may contribute to the generation and maintenance of tinnitus. Because cortical organization can be modified by behavioral training, we attempted to reduce tinnitus loudness by exposing chronic tinnitus patients to self-chosen, enjoyable music, which was modified (“notched”) to contain no energy in the frequency range surrounding the individual tinnitus frequency. After 12 months of regular listening, the target patient group (n = 8) showed significantly reduced subjective tinnitus loudness and concomitantly exhibited reduced evoked activity in auditory cortex areas corresponding to the tinnitus frequency compared to patients who had received an analogous placebo notched music treatment (n = 8). These findings indicate that tinnitus loudness can be significantly diminished by an enjoyable, low-cost, custom-tailored notched music treatment, potentially via reversing maladaptive auditory cortex reorganization." [3]

The proposed treatment for tinnitus [3] alters participants’ favorite music to remove an octave of frequencies around the tinnitus frequency \( f_c \). An octave means a range of frequencies from \( f_i \) to \( 2f_i \). Since \( f_c \) would be in the middle of the octave, \( f_i = (2/3)f_c \). After 12 months of listening to the filtered music, patients reported lessening of tinnitus loudness.

A good rule of thumb in filter design is that the transition region is about 10% of the passband width. In this case, the passband width is \((2/3)f_c\).

Here are the bandstop filter specifications for your design:

- For frequencies 0 Hz to 0.6 \( f_c \), the passband ripple should be no greater than 1 dB.
- For frequencies \((2/3)f_c\) to \((4/3)f_c\), the stopband attenuation should be at least 80 dB.
- For frequencies above 1.4 \( f_c \), the passband ripple should be no greater than 1 dB

Please use a tinnitus frequency \( f_c \) of 3000 Hz and a sampling rate \( f_s \) of 44100 Hz.

(a) Design FIR filters with the minimum filter order to meet the specification by using the Equiripple, Least Squares, and Kaiser Window design methods. FIR equiripple design is also known by many other names: Parks-McClellan, Remez Exchange and Chebyshev Design. Please submit a plot of
the magnitude and phase response for each filter design. Validate that each filter design meets the filter specifications. \textit{Please see the hints below. 27 points.}

(b) Plot the impulse response of the FIR filter designed by the Parks-McClellan (Remez) algorithm. What symmetry is in the impulse response? \textit{7 points.}

c) Give the filter lengths required for filters designed for each filter design method. Which method gives the shortest filter length? \textit{6 points.}

d) Analyze the implementation complexity of each FIR filter design:
   
   1) How many multiplication operations are needed? \textit{3 points}
   2) How much memory (in words) would it take to store the FIR coefficients and the circular buffer for the current and past inputs? \textit{3 points}

Feel free to use Matlab, LabVIEW, or any other computer software on this problem. In Matlab, I’d recommend using the filter design and analysis tool, \textit{fdatool}. This tool is particularly useful when exploring different filter structures for implementation. If you type \texttt{help fdatool} in Matlab, then you will see how to retrieve the transfer function for the current filter being designed. LabVIEW has several filter design demonstrations in the filter design toolkit, e.g. \textit{Advanced Remez FIR Design}.

\textbf{Hints for Part (a)}

For FIR filters, the filter length is the filter order plus one. That is, the filter order is the number of zeros, and the filter length is the number of coefficients. In all three FIR filter design algorithms, you will likely have to search for the design with minimum filter order.

By looking at the plots of the magnitude responses, validate that the filters designed meet the specifications. In particular, carefully inspect the magnitude response in the stopband. In making sure that each filter design meets specifications, be sure to check the graphical views of the passband and stopband, i.e. zoom in on the magnitude response near the passband frequency and near the stopband frequency. For example, the FIR Least Squares method commonly misses the stopband specification.

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 \texttt{frezq} function in Matlab. The arguments are the filter transfer function (type \texttt{help fdatool} to find out how to obtain the transfer function of the current design) and the frequency in Hz. Then, you can take the magnitude of the result. Finally, you'll need to convert the magnitude to dB using

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

When the filter design does not meet specification, one can adjust the filter parameters entered in design tool gradually until the filter design meets specification. In particular, one can increase the stopband attenuation value for Parks-McClellan and Kaiser window design methods and decrease the stopband frequency for the FIR Least Squares method.

\textbf{References}

