Digital Signal Processing

Computer Experiment #2

Due Date: Jan 6 (Friday), 13:00

Problem 1. Hum Noise Removal

1. Problem: Power line hum is a sound associated with alternating current at the frequency of the mains electricity. The fundamental frequency of this sound is usually 60 Hz. The sound often has heavy harmonic content above 60 Hz. Because of the presence of mains current in mains-powered audio equipment as well as ubiquitous AC electromagnetic fields from nearby appliances and wiring, the 60 Hz electrical noise can get into audio systems, and is heard as mains hum from their speakers. Mains hum may also be heard coming from powerful electric power grid equipment such as utility transformers, caused by mechanical vibrations induced by the powerful AC current in them. This project is concerned with the removal of this annoying 60 Hz hum noise. Fig. 2 shows the frequency response of a 60 Hz notch filter.

Figure 2. Response of 60-Hz notch filter

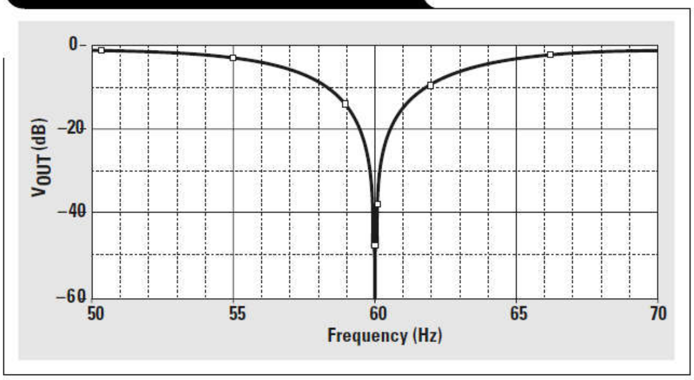
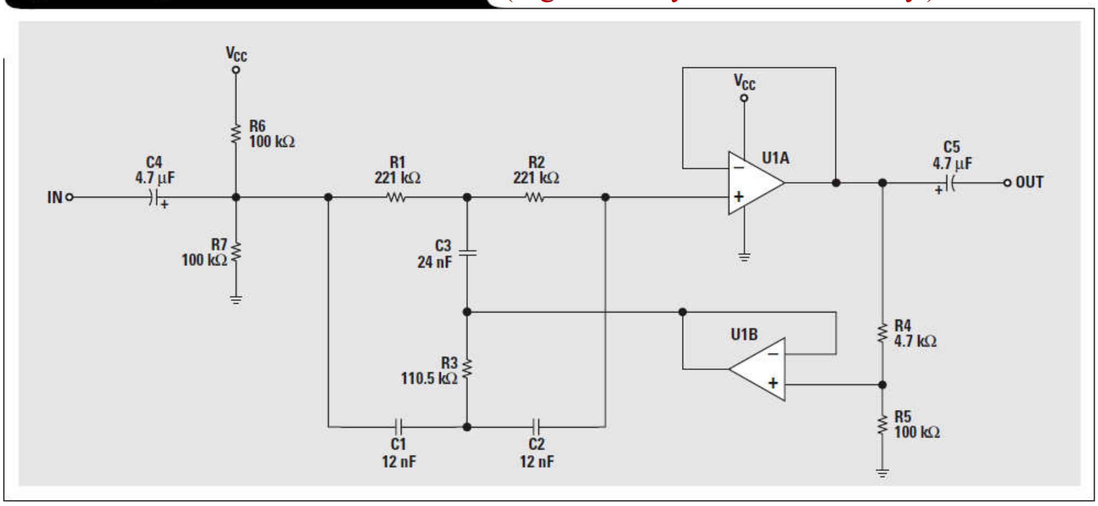


Figure 1. Configuration of a 60-Hz notch filter (Fig. 1 is for your reference only.)



Analog Applications Journal — *February 2001* — *Analog and Mixed-Signal Products* analog filter

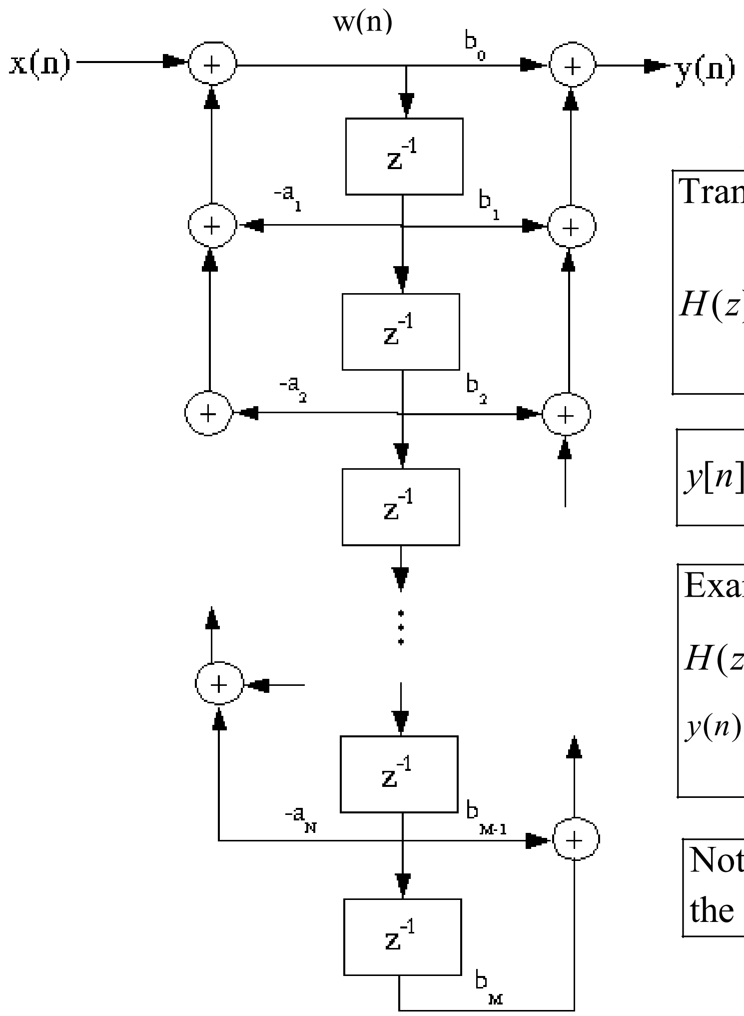
2. Goal: Given a sound file (stored in the .wav format) with hum noise, manage to remove or reduce the 60 Hz hum..

3. Method:

(a) Design a digital IIR notch filter to remove the 60 Hz hum. The notch filter is simply a bandstop filter with a very narrow passband. You may <u>determine your own filter specifications or try the following specs.</u>: (Note that <u>you should modify the specifications if necessary</u>.)

- passband edges (ω_P): 57 Hz and 63 Hz
- stopband edges (ω_S): 59.5 Hz and 60.5 Hz
- passband ripple: 0.5 dB
- stopband attenuation: 40 dB

(b) Implement the IIR filter in the time-domain using the <u>direct form-II structure</u> as shown below.



Transfer Function:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$$

Example:

$$H(z) = \frac{4.5 + 12.5z^{-1} - 9.47z^{-2}}{1 + 3.24z^{-1} - 7.6z^{-2} + 8.92z^{-3}}$$

$$y(n) = -3.24y(n-1) + 7.6y(n-2) - 8.92y(n-3) + 4.5x(n) + 12.5x(n-1) - 9.47x(n-2)$$

Note the **negative sign** of the coefficients $a_k, k = 1, \cdots, N$.

(c) difference equations for the implementation: (with **<u>zero initial states</u>**)

$$w(n) = -a_1 w(n-1) - a_2 w(n-2) - \cdots - a_N w(n-N) + x(n)$$
$$y(n) = b_0 w(n) + b_1 w(n-1) + \cdots + b_M w(n-M)$$

(d) Listen carefully to the sound files to compare the results before and after the hum noise has been removed.

4. Sample Program: A prototype program in Octave is given here for your reference.

```
clear all;
[xin, fs] = audioread('test_filename.wav');   % fs is the sampling rate
x = xin(:,1);

% Design the IIR Notch Filter for Hum Removal
wp = [57 63]/(fs/2);   % 2*pi* (wp/fs) / pi, normalized passband frequencies
ws = [59.5 60.5]/(fs/2);   % 2*pi* (ws/fs) / pi, normalized stopband frequencies
rp = 0.5;   % passband ripple
rs = 40;   % stopband attenuation
...
Your Own Code
...
% Perform the filtering
...
...
% Do NOT use the built-in function filter() to carry out the filtering process.
% Instead, write your own code to fulfill the task. (i.e., write a new myfilter() function)
out = filter(b, a, x); % b is the coefficient vector for the numerator and
                       % a is the coefficient vector for the denominator
out_scaled = out / max(abs(out)); % amplitude scaling
audiowrite('your_filename.wav', out_scaled, fs);
```

5. Key Issues to Be Explored and Discussed:

(1) Do frequency responses of designed filters meet the required specifications?

(2) computational efficiency of the filtering process

(3) BIBO stability of the designed filters

(4) coefficient quantization effects and round-off errors in the filtering process (optional)

6. Report: The following items should be included in your report:

- **tables of all filter coefficients**, including both the numerators and the denominators
- **frequency responses** of the designed filters, with the design specifications verified
- short segments of the filtered waveforms compared to the original waveforms in the time-domain

- Fourier spectra of both the input and the processed signals
- **the filtered signals in the .wav format**

Problem 2. Artificial Reverberation

Delay and reverberation systems. Music generated in an inert studio does not sound natural compared to the music performed inside a room, such as a concert hall. In the latter case, the sound waves propagate in all directions and reach the listener from various directions and at various times, depending on the distance traveled by the sound waves from the source to the listener. The sound wave coming directly to the listener, called the *direct sound*, reaches first and determines the listener's perception of the location, size, and nature of the sound source. This is followed by a few closely spaced echoes, called *early reflections*, generated by reflections of sound waves from all sides of the room and reaching the listener at irregular times. These echoes provide the listener's subconscious cues as to the size of the room. After these early reflections, more and more densely packed echoes reach the listener due to multiple reflections. The latter group of echoes is referred to as the *reverberation*. The amplitude of the echoes decay exponentially with time as a result of attenuation at each reflection. Figure 1.32 illustrates this concept.

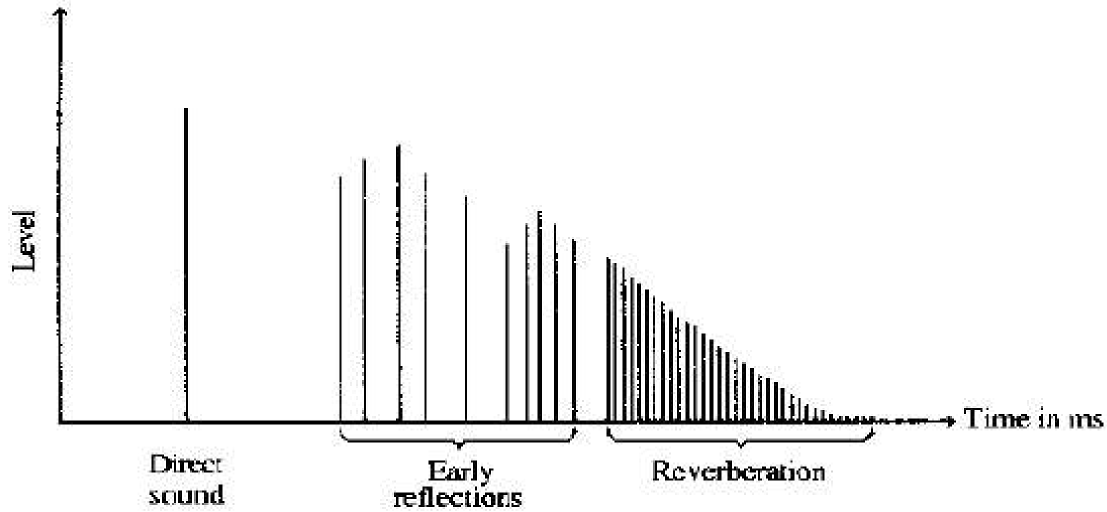


Figure 1.32: Various types of echoes generated by a single sound source in a room.

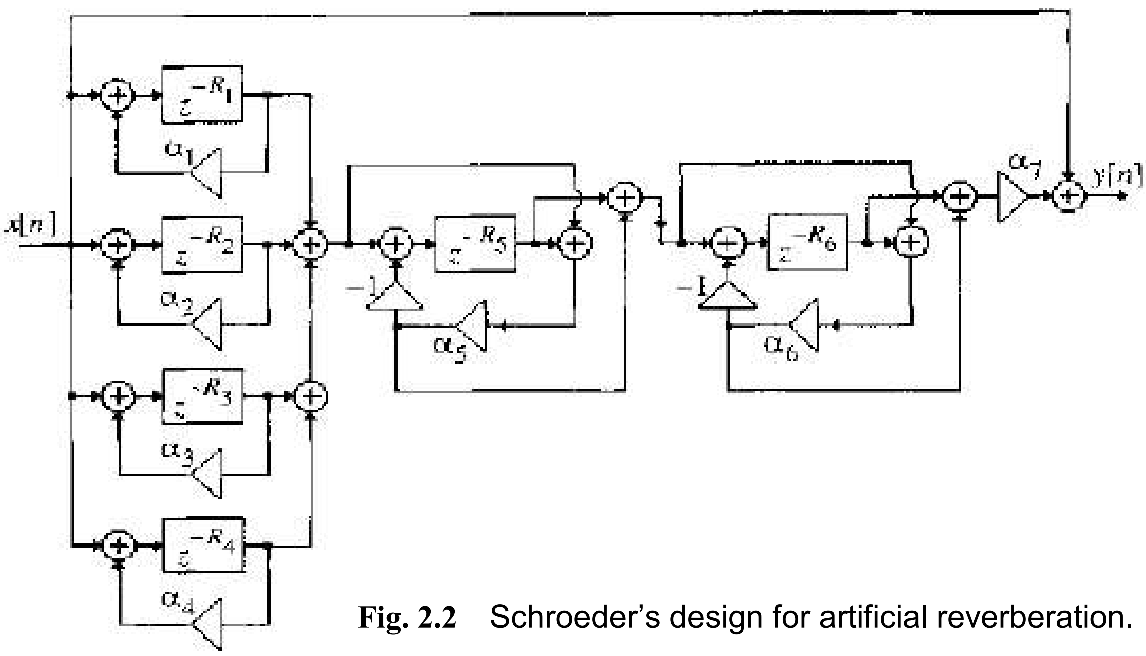


Fig. 2.2 Schroeder's design for artificial reverberation.

1. Problem: Investigate the performance of Schroeder's design for artificial reverberation.

2. Method: See Fig. 2.2 for Schroeder's design for artificial reverberation. In this experiment, you try to apply Schroeder's method with different combinations of parameter settings to examine the effect artificially generated reverberation. For your convenience, you can use the attached programs directly. The design parameters are $\alpha_i, i = 1,\cdots,7, \textit{and } R_i, i = 1,\cdots,6.$

3. Report: The following items should be included in your report:

- at least three processed sound files (using parameters different from suggested)
- list of the parameters you used
- your findings about how to choose the parameters

Note:

The complete report package, including report, programs, and output signals, should be mailed to

dipwork405@gmail.com

prior to the deadline.