LAB 7

TRANSFER FUNCTION AND FILTER DESIGN

OBJECTIVES

- Understand the effect of the system's poles and zeros on its response to inputs of different frequencies.
- Design a filter with desired frequency response using zeros and poles placement technique.

INTRODUCTION

Placement of Poles and Zeros of *H*(*s*)

The Frequency response of a system provides information about the filtering capability of the system. The frequency response of a system is determined by the locations of the poles and zeros of its transfer function *H*(*s*) in the s-plane. Poles and zeros locations in the s-plane is a simple intuitive procedure to any filter design. To amplify a frequency component then place a pole next to this frequency. To attenuate a frequency component then place a zero next to this frequency.

The numerator and denominator of the transfer function *H*(*s*) of any system can be expressed as the product of vectors in the s-plane. The vectors originate from the locations of the poles and zeros for *H*(*s*) to any point in the s-plane that represents a specific frequency. For example, the magnitude |*H*(j*ω*)| of the transfer function shown below has the highest value when the frequency is equal 10 rad/sec. This system resembles a first order bandpass filter with central frequency of 10rad/sec.

$$H_1(s) = \frac{(s-1)}{(s+1-j10)(s+1+j10)} = \frac{r_1 e^{\theta_1}}{d_1 e^{\emptyset_1} d_2 e^{\emptyset_2}}$$

Frequency Components using Spectrogram

A spectrogram is a plot that shows the strength of different frequency contents present in a signal at different time. The *x*-axis is time and the *y*-axis is frequency. The strength of the different frequency components is coded by color. The spectrogram is found by the short time Fourier transform of the signal. The signal is divided to small time intervals and then the Fourier transform for each interval is calculated. Fourier transform is a special case of the Laplace transform when *σ* = 0 and *s* = j*ω*. The Fourier transform for each interval will reveal the frequency components present in this interval. The following example for the signal *s*(*t*) demonstrates the use of the spectrogram function in MATLAB to show the spectrum of the signal.

$$s(t) = 3sin(2\pi 500t) + sin(2\pi 2000t)$$

As shown in **Fig.** 1, the sinusoidal wave with 500 Hz exists in the interval 0 to 3 second, and the sinusoidal wave with 2000Hz exists in the interval 1 to 2 second.

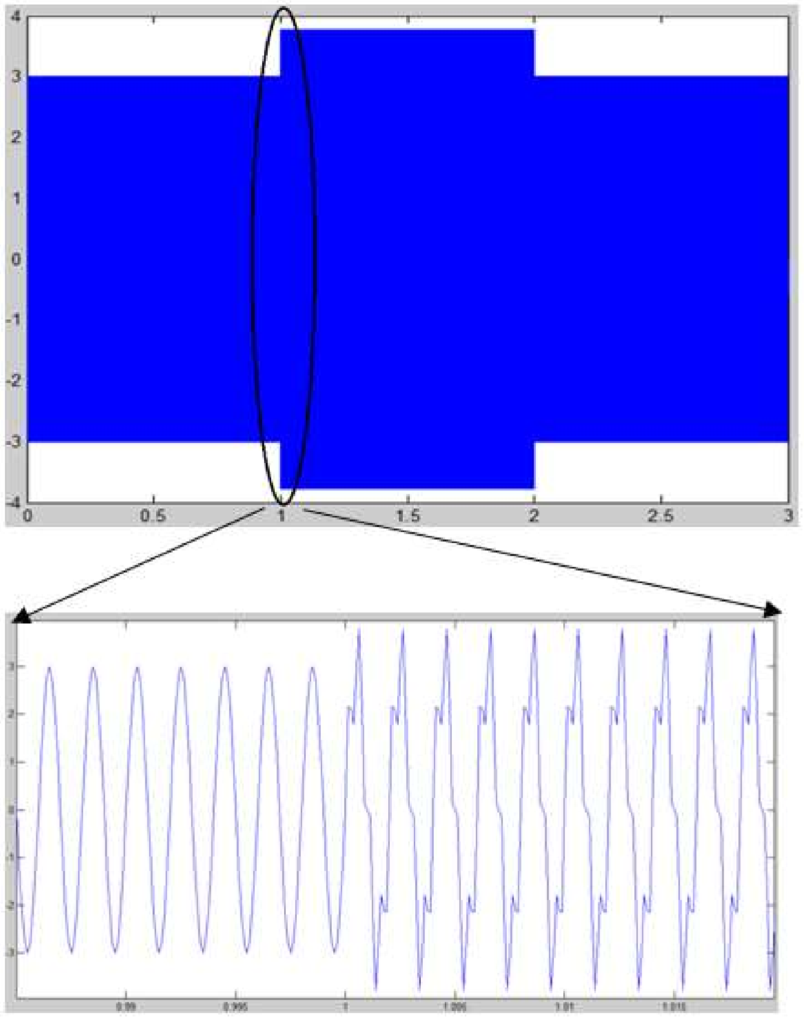

Fig. 1: Signal S(t) in the Time Domain

MATLAB code generating the signal *s*(*t*) and its spectrogram is below

```
clear all
Fs=8000;                                   % sampling frequency
Wind = 0.1;                         % time interval for calculating the Fourier transform (FT) is 0.1 second
NumSampWind = Fs*Wind;              % number of samples in the window for calculating the Fourier Transform
t1=0:1/Fs:3;                               % time scale from zero to 3 seconds
t=0:1/Fs:1;                                % time scale from zero to 1 seconds
ZeroTime = zeros(1, length(t)-1);
t2=[ZeroTime t ZeroTime];                  % first and third seconds are zeros
Freq = [0:10:4000];                        % frequency at which the Fourier transform will be calculated
Sig = 3*sin(2*pi*500*t1) + sin(2*pi*2000*t2);  % sin(2π500t) for 0<t<3 and sin(2π2000t) for 1<t<2
spectrogram(Sig, NumSampWind, 0, Freq, Fs, 'yaxis')  % calculate FT and plot spectrum where time in x-axis and
                                           % frequency in y-axis
```

The plot of the spectrogram is shown in **Fig.** 2. The two red lines indicate the presence of the 500 Hz and the 2000 Hz frequency components. The 500 Hz exists in the time interval from 0 to 3 second and the 2000 Hz exists in the time interval from 1 to 2 second. The red color indicates higher magnitude (stronger) while blue indicates lower magnitude (weaker) frequency components in the signal *s*(*t*).

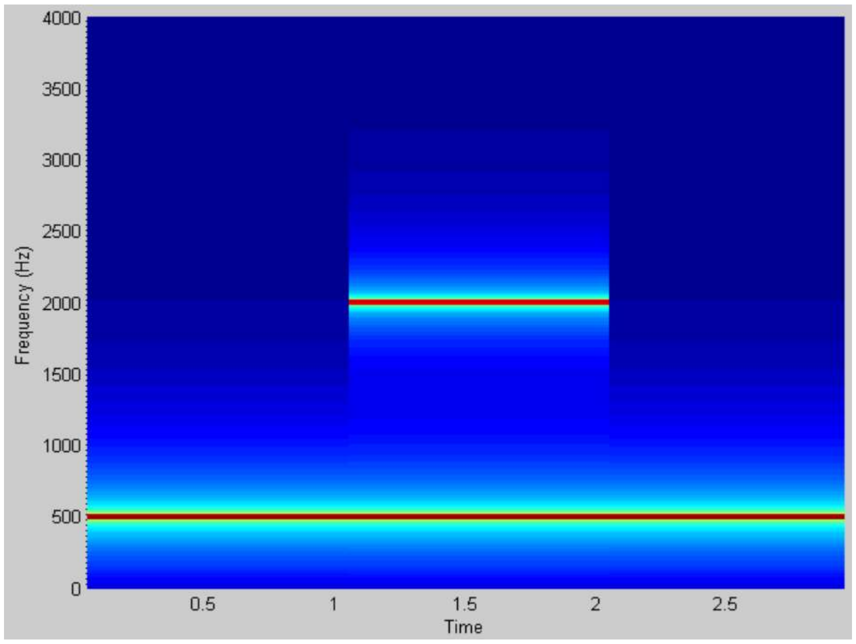

Fig. 2: Spectrogram of Signal *s*(*t*)

Notch Filter Design using *H*(*s*) of RLC Circuit

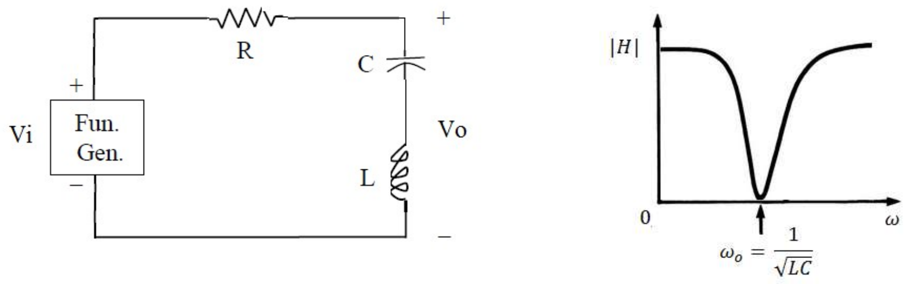

Fig. 3: 2nd Order Notch Filter

The circuit above can be used as a 2nd order notch filter if the voltage output is taken across the inductor and capacitor. The transfer function *H*(*s*) of Vo/Vi is obtained from the voltage divider rule.

$$Vo = Vi\left(\frac{\frac{1}{sc} + sL}{\frac{1}{sc} + sL + R}\right)$$

$$H(s) = \frac{Y(s)}{X(s} = \frac{Vo}{Vi} = \frac{\frac{1}{sc} + sL}{\frac{1}{sc} + sL + R}$$

Multiply the numerator and denominator by *sc*

$$H(s) = \frac{s^2LC + 1}{s^2LC + sCR + 1} = \frac{s^2 + \frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}$$

The frequency response *H*(j*ω*) can be found by setting *s* =j*ω*,

$$H(j\omega) = \frac{-\omega^2 + \frac{1}{LC}}{-\omega^2 + j\omega\frac{R}{L} + \frac{1}{LC}}$$

$$For\ |H(j\omega)| = 0, \omega = \omega_o = \frac{1}{\sqrt{LC}}$$

A 2nd order notch filter can be specified in terms of its poles and zeros. For example, a 2nd Order notch filter that suppresses 60Hz hum in a radio receiver will have two zeros at $s = \pm j\omega_o$ and two poles at $-\omega_o cos\theta \pm j\omega_o sin\theta$. The filter transfer function for this notch filter with $\omega_o = 2\pi f = 2\pi 60 = 120\pi$ is

$$H(s) = \frac{(s - j\omega_o)(s + j\omega_o)}{(s + \omega_o cos\theta + j\omega_o sin\theta)(s + \omega_o cos\theta - j\omega_o sin\theta)} = \frac{s^2 + \omega_o^2}{s^2 + (2\omega_o cos\theta)s + \omega_o^2}$$

$$H(s) = \frac{s^2 + 142122.3}{s^2 + (753.98cos\theta)s + 142122.3}$$

$$and\quad |H(j\omega)| = \frac{-\omega^2 + 142122.3}{\sqrt{(-\omega^2 + 142122.3)^2 + (753.98\omega cos\theta)^2}}$$

If you plot the poles and zeros in the s-plane for different values of *θ*, the poles will be closer to the zeros as *θ* get closer to π/2. The closer the poles are to the zeros (the closer θ is to π/2), the faster the gain recovery for the 60 Hz bandstop or notch filter.

You can simulate in Matlab the system whose transfer function is defined above as follow:

```
Clear all
[InputSig sr]=audioread('RainFireNoise.wav');          % loading the signal
t=0:1/sr:(length(InputSig)-1)/sr;              % creating time vector that determine the samples locations
thet=80*pi/180;
NumCof = [1 0 142122.3];                  % the numerator's coefficients of the transfer function H(s)
```

```
DenCof = [1 753.98*cos(thet) 142122.3];   % the denominator's coefficients of the transfer function H(s)
Sys = tf(NumCof, DenCof);                 % the Matlab function tf build the system  H(s)
OutputSig = lsim(Sys, InputSig, t);       % lsim returns the output "OutputSig" of the system "Sys" for InputSig
```

The code below plots the zeros and poles for a notch filter with the parameters' values $\omega_o = 120\pi$ and $\theta = 85^o$. It also calculates the coefficients of the numerator and denominator of the transfer function $H(s)$ and plots the magnitude and the phase of $H(j\omega)$.

```
Clear all
k= 1;                            % gain
f=0:0.01:100;
w=2*pi*f;
wo=2*pi*60;                      % frequency
theta = 85*pi/180;
z=[wo*i; -wo*i];                 % Two zeros
p=[-wo*cos(theta) + wo*sin(theta)*i; -wo*cos(theta) - wo*sin(theta)*i];        % Two poles
zplane (z, p);                   % plot the zeros and poles on the s-plane/z-plane
figure;
[NumCof,DenCof] = zp2tf(z,p,k);      % return the numerator and denominator coefficients for H(s)
H=freqs(NumCof,DenCof,w);            % return the system, specified by Coff, responses to the frequency vector w
magH = abs(H);
phaseH = angle(H);
plot(w, magH)
figure
plot(w, phaseH)
```

PRELAB EXERCISES

Design a notch filter that will eliminate a tone noise with 1000 Hz frequency. You have in the lab 10 mH inductor and a variety of capacitors and resistors. Choose the value of $\theta = 85^o$.

Call the instructor to verify the prelab. Make sure to box the transfer function *H*(*s*) and the values of *R* and *C*.

Attendant Signature: ____

LAB EXERCISES

Filter out pure tones with minimum impact on the frequencies of a musical signal.

1) First, download from the course website the music file "RainFireNoise" to an active MATLAB folder and play the *.wav file:

```
>> [SigN sr]=audioread('Type the directory of the folder/MusicTone.wav');
>> sound(SigN, sr)
```

Can you hear the pure-tone noises? How many tones due hear?

2) Plot the corrupted musical waveform in the time domain.

3) Use the Matlab ***spectrogram*** function to find the frequencies of the pure-tone noises. You can find in the introduction section of this lab an example using the ***spectrogram*** function. **Verify your finding with the lab instructor.**

4) Use the pole-zero placements in the s-plane technique to design a filter to eliminate as much as possible of the pure-tone noises without deteriorating the music quality.

5) Plot the frequency magnitude $|H(\omega)|$ and the phase of the designed filter.

6) Test your design by simulating the filter in Matlab using the ***tf*** and ***lsim*** functions to filter out the noise. Play and plot the cleaned-up music signal.

Call your instructor to verify the quality of the cleaned music.
Attendant Signature:

____

7) Use the ***spectrogram*** function in MATLAB to plot the spectrogram of the cleaned-up music signal. Can you tell from the spectrogram if the strength of the noise is eliminated or reduced?

8) Did your designed filter eliminate all the pure noisy tones? If not, identify the problem and provide a solution for a better design.

9) Build the physical circuit of the filter you designed to remove the two frequencies of the noise. Remember if your system is 4th order then you can redesign it to be two subsystems connected in series (cascaded) where each subsystem is a 2nd order. The first subsystem eliminates the lower frequency pure-tone noise and the second subsystem eliminates the higher frequency pure-tone noise. Test your circuit by applying inputs of sinusoidal signals to your filter from the function generators. Use the frequencies shown in the table below for the sinusoidal inputs. Display the sinusoidal inputs and outputs in the oscilloscope. Fill the table below.

Freq (Hz)	Input Voltage (Vi)	Output Voltage (Vo)	|H(jω)| = Vo/Vi
50			
100			
200			
500			
1000			
2000			
3000			
4000			
5000			
6000			
7000			
8000			

Plot $H(j\omega)$ versus frequency. Does the shape agree with the simulation? If the output decreases significantly for sinusoidal with frequencies equal to the frequencies of the noise while sinusoidal signals of other frequencies are not attenuated by the same factor then your design is working.

10) Apply the original signal with pure-tone noises from the computer audio output to your filter. The output of your filter should be the input to a speaker. You can play the corrupted music by clicking on the wave file without Matlab. Do you hear the pure-tone noises?

Call your instructor to verify the quality of the cleaned music.
Attendant Signature:

____

Lab Report Format

Submit the solutions to the questions and tasks in the order given above, and indicate the corresponding number of each question or task. Attach a title page to the front of your report.

Your report must include the following sections: (1) Title Page (2) Introduction (3) Results (Include answers to all questions and tasks) (4) Conclusion