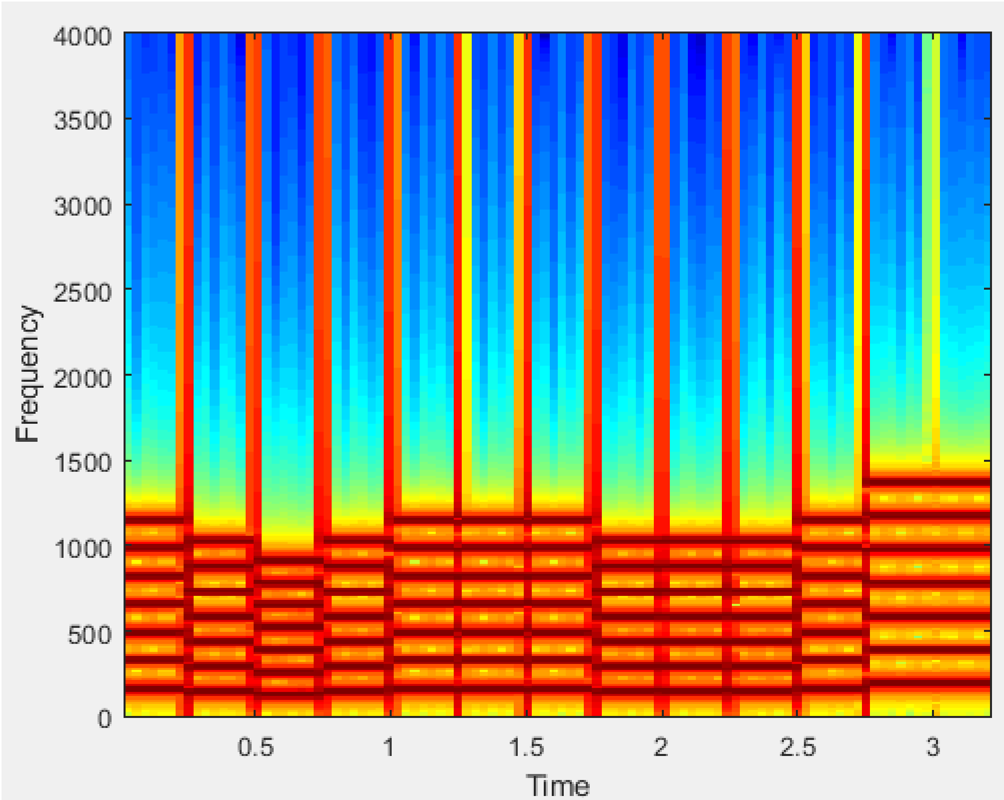
```
diary on
format compact
%Johnny Li
%EEL3135 Fall 2018
%Lab 7 Part 1

%1.1

%Create a one-second-long glottal source signal, containing 7 nonzero
%harmonics, with fundamental frequency 150Hz.
%Given Frequency
f=150;
%Sampling rate
fs=7*f*2;
%Time Interval, 1 second long
tt=0:1/fs:1
%Storage/Initial value
glottal=0;
%Loop 7 harmonics
for i=1:7
    %Function
    glottal=cos(2*pi*f*tt*i)+glottal;
end

%Quesiton: What is the minimum sampling frequency necessary to avoid
%aliasing?
%The minimum sampling frequency necessary to avoid aliasing is 2 times the
%highest frequency of the system which is the seventh harmonic, given by
%7*150Hz, therfore the minimum sampling frequency =2*7*150=2100 Hz.

%1.2

%Write a function, name it glottal_key_to_note, and use a sampling
%frequency of 8000 Hz.
type glottal_key_to_note

function xx = glottal_key_to_note(keynum, dur)
%GLOTTAL_KEY_TO_NOTE Function created for lab7.1.
%Takes in a key number anda duration, to produce a glottal source signal of
%given duration with fundamental frequency corresponding to the desired %
%note.

%{
    KEY_TO_NOTE: Produce a sinusoidal waveform corresponding to a given
    piano key number.
    Input Args:
        X: amplitude (default = 1)
        keynum: number of the note on piano keyboard
        dur: duration of the note (in seconds)
    Output:
        xx: sinusoidal waveform of the note
%}

%Code Taken from key to note
%Smaple frequency
```

```
fs = 8000;
%Time interval
tt = 0:(1/fs):dur-1/fs;
%Given frequency function
freq = 220*2^((keynum-49)/12);

%Storage/Initial value
xx=zeros(size(tt));
%Loop 7 harmonics
for i=1:7
    %Sinusoidal function
    xx = real(exp(i*j*2*pi*freq*tt))+xx;
end

end

%1.3

%Use glottal_key_to_note to write a script that plays mary.
type play_maryg

%Plays a series of notes from mary.
%Code taken from lab 3.
%Script based on given instruction 1.3.

%Code given
% --------------play_mary.m-------------- %
mary.keys = [44 42 40 42 44 44 44 42 42 42 44 47 47];

%Notes: C D E F G
%Key #40 is middle-C

mary.durations = 0.25 * ones(1,length(mary.keys));
fs = 8000;  % 11025 Hz also works
xx= zeros(1, sum(mary.durations)*fs);

n1 = 1;
for kk = 1:length(mary.keys)
    keynum = mary.keys(kk);

    %Tone function
    tone = glottal_key_to_note(keynum,mary.durations(kk));

    n2 = n1 + length(tone) - 1;
    xx(n1:n2) = xx(n1:n2) + tone;
    n1 = n2 + 1;
end

%Create autofile
audiowrite('play_maryg.wav',xx,fs);

%Plot the frequency-time spectrogram of Mary.
specgram(xx,512,fs)
```




```
%1.4

%Now, use your glottal_key_to_note to synthesize Bach Fugue from Lab 3.
type play_bachg

%Plays a series of notes from bach fugue.
%Code taken from lab 3.
%Script based on given instruction 1.4.

%Load bach fugue
%load bach_fugue.mat;

%Code take from play__song

%Frequency
fs = 8000;
%Beat per minute->beats per second->second per beats->second per pulse
%Given Code
beats_per_minute  = 120;
beats_per_second  = beats_per_minute / 60;
seconds_per_beat  = 1 / beats_per_second;
%spp = seconds_per_beat / 4;
%seconds per pulse, theVoices is measured in pulses with 4 pulses per beat

%Set spp to 0.15 for better fugue
spp=0.15;

%Length of voices
```

```
numV=length(theVoices);
%Length of notes
numN=length(theVoices(numV).noteNumbers);
%Final start pulse
fsp=theVoices(numV).startPulses(numN);
%Final durations
fd=theVoices(numV).durations(numN);

%Get Max value in theVoices
M=0;
for a=1:numV
    for b=1:length(theVoices(a).durations)
        d=theVoices(a).durations(a);
        st=theVoices(a).startPulses(b);
        if M<(d+st)
            M=d+st+1;
        end
    end
end

%Longest value in better
song = zeros(1,ceil(M*spp*fs));
%Create a vector of zeros with length equal to the total number of samples
%in the entire song

%Then add in the notes
for i = 1:length(theVoices)
    for j = 1:length(theVoices(i).noteNumbers)
         note =
glottal_key_to_note(theVoices(i).noteNumbers(j),theVoices(i).durations(j)*spp
);
         %Create sinusoid of correct length to represent a single note
         locstart = theVoices(i).startPulses(j)*spp*fs;
         %Index of where note starts
         locend = locstart+length(note)-1;
         %Index of where note ends
         song(locstart:locend) = song(locstart:locend) + note;
    end
%For clipping
song=song/(max(abs(song)));
end
%Create autofile
audiowrite('bach_fugueg.wav',song,fs);

%1.5

%Question: Where in the complex plane can zeros and poles be placed to have
%the strongest influence on the magnitude response of the filter?
%When the zeros are placed closer to the unit circle and the poles are
%place closer to the origin, this has the strongest influence on the
%magnitude response of the filter.


%1.6
```

Johnny Li
ECE 3135
P.3

Lab 7 Part 1

1.6) $y[n]=x[n]+2y[n-1]$

$Y(z)=X(z)+2z^{-1}Y(z) \Rightarrow Y(z)-2z^{-1}Y(z)=X(z)$

$Y(z)(1-2z^{-1})=X(z)$

$H(z)=\frac{Y(z)}{X(z)}=\frac{1}{1-2z^{-1}}\cdot\frac{z}{z}=\frac{z}{z-2}$

$\boxed{\text{poles: } z=2 \quad \text{zeros: } z=0}$

$a^n u[n] = \frac{1}{1-az^{-1}} \quad \therefore \frac{1}{1-2z^{-1}} \Leftrightarrow \underline{2^n u[n]}$ impulse response

%Question: What are the poles and zeros of this filter?
%The pole is z=2 while the zero is z=0 of this filter.
%Work done by hand.

%Question: What is this filter’s impulse response?
%The filter's impulse response is equal to (2^n)u[n].
%Work done by hand.

%1.7.1

%Table of the zeros location in normalized radiant frequency.
%Done by hand.

1.7.1) zeros: $f=1000,\ 2100,\ 3500$

$\hat{\omega}=\frac{2\pi f}{f_s}=\frac{2\pi f}{8000}=\frac{\pi f}{4000}$

zeros	positive conjugate	negative conjugate
1000	$\frac{\pi}{4}$	$-\frac{\pi}{4}$
2100	$\frac{21\pi}{40}$	$-\frac{21\pi}{40}$
3500	$\frac{7\pi}{8}$	$-\frac{7\pi}{8}$

$\boxed{\therefore\ e^{j\pi/4},\ e^{-j\pi/4},\ e^{j\frac{21\pi}{40}},\ e^{-j\frac{21\pi}{40}},\ e^{j\frac{7\pi}{8}},\ e^{-j\frac{7\pi}{8}}}$

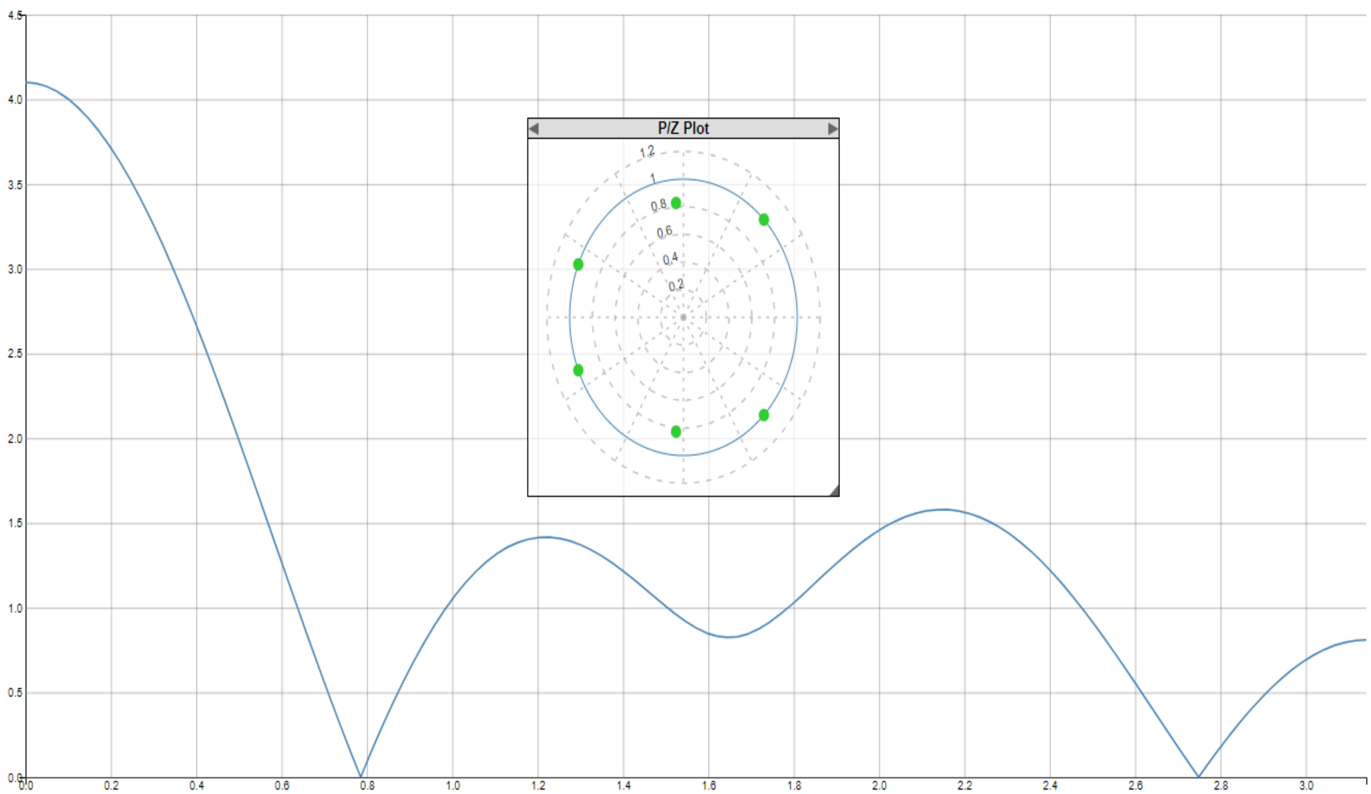
%1.7.2

%Create an FIR filter with six nontrivial zeros that matches the following
%magnitude response.
%H(z)=(z-0.25pi)(z-0.525pi)(z-0.875pi)

%Question: What are the filter coefficients b and a?
%The filter coefficients are:
B = [1, 0.61536, 0.20421, 0.64225, 0.62603, 0.50200, 0.73851]
A = [1]

%1.7.3

%Submit a screenshot of your GUI.



%1.7.4

%Filter a glottal source signal, with fundamental frequency 150Hz and
%sampling frequency 8000Hz, through the eh filter.
%Code Taken from 1.7.1, generate source signal.
%Given Frequency
f=150;
%Sampling rate
fs=7*f*2;
%Time Interval, 1 second long
tt=0:1/fs:1;

```
%Storage/Initial value
glottal=0;
%Loop 7 harmonics
for i=1:7
    %Function
    glottal=cos(2*pi*f*tt*i)+glottal;
end

%Filter coefficients from GUI
B = [1, 0.61536, 0.20421, 0.64225, 0.62603, 0.50200, 0.73851];
A = (1);
%Given Frequency
fs=8000;

%Filter
eh= filter(B,A,glottal);

%For clipping
eh=eh/(max(abs(eh)));

%Create autofile
audiowrite('eh.wav',glottal,8000);
```