Chen Ji
SUNetID:jjjjjc

Homework 6a

Music424
May 29, 2019

Problem 1

(a) Referring to the table of Sabine coefficients in chapter 9 as below:

Absorption Coefficients, S						
material	*frequency*					
	125	250	500	1000	2000	4000
marble	0.01	0.01	0.01	0.01	0.02	0.02
brick	0.03	0.03	0.03	0.04	0.05	0.07
concrete block	0.36	0.44	0.31	0.24	0.39	0.25
plywood	0.28	0.22	0.17	0.22	0.10	0.11
cork	0.14	0.25	0.40	0.25	0.34	0.21
glass window	0.35	0.25	0.18	0.12	0.07	0.04
drapery	0.10	0.25	0.46	0.60	0.56	0.52
carpet	0.02	0.06	0.14	0.37	0.66	0.65
hardwood	0.15	0.11	0.10	0.07	0.06	0.07
grass	0.11	0.26	0.60	0.69	0.92	0.94

Figure 1: Derivation of all-pass group delay

There are two parts need to be condidered in the reverberation time.
For solid surfaces, we use following formula:

$$T_{60}(\omega) = -2ln(0.001)\frac{V}{gc\sum_i a_i S_i}$$

Also, considering the air absorption, we need:

$$T_{60}(\omega) = [\frac{1}{T_{60surface}(\omega)} + \frac{1}{T_{60air}(\omega)}]^{-1}$$

Approximate the church as a totally enclosed shoebox, 25m wide, 40m long, 10m high. Use plywood for the ceiling and half the floor, use glass for about a tenth of the walls and plywood for another tenth, and assume marble for everything else. The following MATLAB script can be written to calculate T60.

```
% Constants:
c = 340.27; % speed of sound m/s
g = 1/4; % geometric constant
k = -2*log(0.001)*(1/(g*c));
% Sabine coefficients:
freqs = [250 1000 4000];
Swood = [0.22 0.22 0.11];
Sglass = [0.25 0.12 0.04];
```

```
Smarb = [0.01 0.01 0.02];
% Room dimensions:
length = 25; % meters
width = 40; % meters
height = 10; % meters
volume = length*width*height;
% Calculate surface areas:
ceilArea = length*width;
floorArea = ceilArea;
wallArea = length*height*2 + width*height*2;
% Calculate material areas:
woodArea = ceilArea + 0.5*floorArea + 0.1*wallArea;
glassArea = 0.1*wallArea;
marbleArea = 0.5*floorArea + 0.8*wallArea;
% Pre-allocate:
t60Solids = [];
% Calculate t60 for surfaces:
for i = 1 : 3
t60Solids(i) = k / ((woodArea*Swood(i) +glassArea*Sglass(i) + ...
    marbleArea*Smarb(i)) / volume);
end
% Print output:
% freqs
% t60Solids
% Calculate total t60:
t60Air = [130 30 6.8];
totalT60 = ( (1./t60Solids) + (1./t60Air) ) .^-1
```

Then we can get the estimated values as below:

250Hz	1kHz	4kHz
3.8761 seconds	3.6600 seconds	3.5762 seconds

(b) If the church were half the size, we see the values change as below:

250Hz	1kHz	4kHz
1.9674seconds	1.9489 seconds	2.4260 seconds

(c) If the church floor were instead carpet, we see the values change as below:

250Hz	1kHz	4kHz
2.2699 seconds	1.2090 seconds	0.8829 seconds

Problem 2

(a) Write a matlab function [sweep] = chrpLin(duration, fs), the MATLAB function can be written as:

```
function [sweep] = chrpLin(duration,fs)
% Ramps from 0 to fs/2 with constant frequency
% trajectory (linear). Returns the signal.
L = ceil(fs*duration);
n = 0:L-1;
t = n/fs;
sweep = chirp(t,0,duration,fs/2);
end
```

(b) Write a matlab function [sweep] = chrpLog(f0, f1, duration, fs), that ramps from f0 to f1 with a frequency trajectory that increases so it spends equal time on each octave.

```
function [ sweep ] = chrpLog( f0, f1, duration )
% Ramps from 0 to fs/2 with logarithmic frequency
% trajectory (equal time per octave).
L = ceil(fs*duration);
n = 0:L-1;
t = n/fs;
sweep = chirp(t,f0,duration,f1, 'logarithmic');
end
```

(c) Write a matlab function [ir] = chrp2ir(ss, rs) that takes a sine sweep and the response to that sweep and performs the deconvolution necessary to convert a sine sweep into an impulse response.

```
function [ ir ] = chirp2ir( ss, rs )
% Takes a sine sweep and the response
% to the sweet and deconvolves them
% to retrieve the impulse response.
ir = ifft(fft(rs)./fft(ss));
end
```

(d) Write a matlab function [env] = envergyEnvelope(sig, fs, eta) that takes an input signal, sample rate, and smoothing time (in ms) and returns an running-RMS smoothed amplitude envelope of the signal.

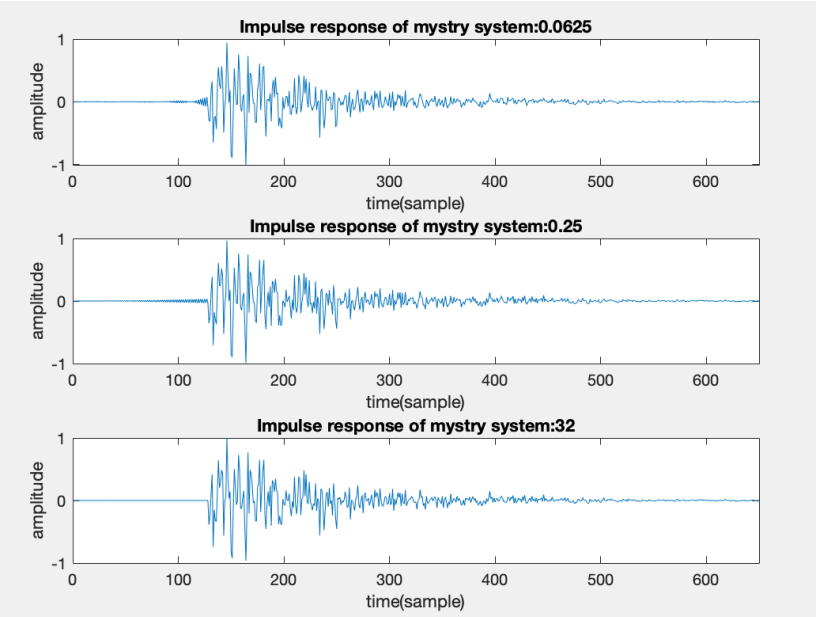
```
function [ env ] = energyEnvelope( sig, fs, eta )
% Takes an input signal, sample rate, and smoothing
```

```
% time, and returns a running RMS smoothed
% envelope of the signal. Eta is in milliseconds.
[up,low] = envelope(sig,eta*fs/1000,'rms');
env = up;
end
```

Problem 3

(a) The MATLAB code to generate the impulse response is:

```
duration = [1/16 1/4 32];
fs = 44100;
for i = 1:3
sweep = chrpLin(duration(i),fs);
response = hmeasure(sweep)';
ir = chirp2ir(sweep,response);
subplot(3,1,i);
plot(ir);
xlim([0,650]);
xlabel('time(sample)');
ylabel('amplitude');
title(strcat('Impulse response of mystry ...
    system:',num2str(duration(i))));
hold on;
end
```



Homework 6a

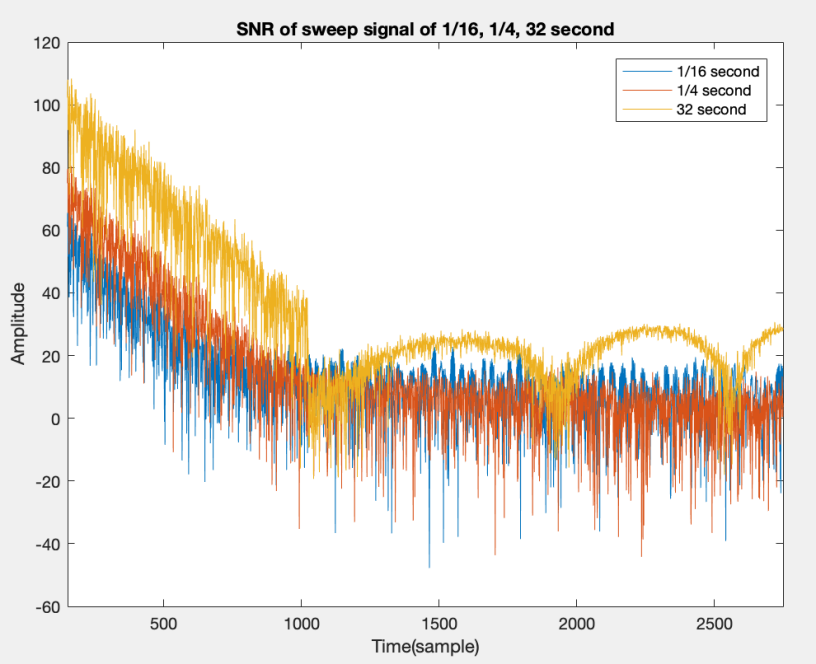
(b) Using the fact that the impulse response h(t) begins with a series of zeros, we can estimate the noise level by using the very first second of the sweep signal. The noise level of sweep signal in 1/16, 1/4 and 32 second are shown as below:

1/16	1/4	32
4.7491e-04	-9.4191e-05	3.6971e-06

The SNR can be calculated by using the noise level and peak value of sweep:

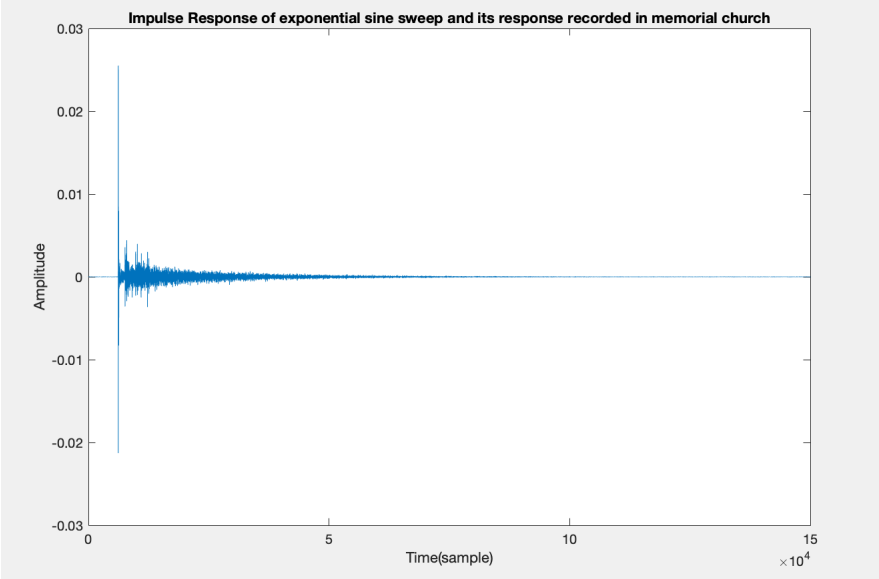
1/16	1/4	32
66.4961	80.4374	108.6423

Then the plot can be shown as:

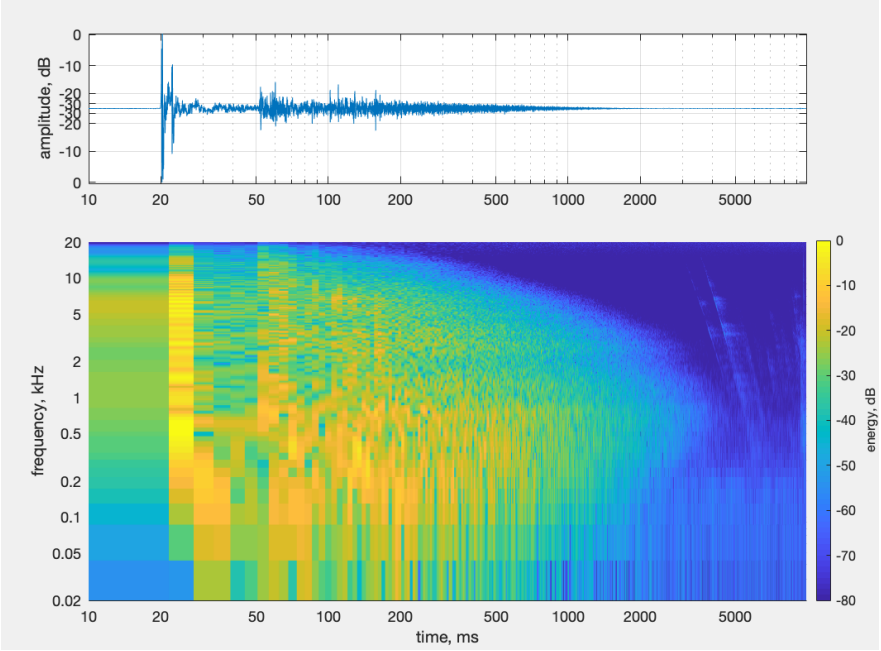


Problem 4

(a) The impulse response of an exponential sine sweep and its response recorded in memorial church can be shown as below:



Then we can get the spectrogram of the computed impulse response can be shown as below:



(b) Convolve a balloon pop recorded in Memorial Church and the sine sweep-converted impulse responses with a guitar track or the provided recording of Tom's Diner. I used the conv() function in MATLAB to do the convolution. Compared with sweep one, the balloon pop one is more noisy than the sweep one. Personally, i like the sweep one better.

The convoluted audio clips are included in the submitted file.