LAB 4

Audio Response for Real-world LTI Systems and Aliasing from Undersampling

ECEn 380 Section 001

Task 1 signoff: [signature] 11/2/15

Task 2 signoff: [signature] 11/9/2015

Extra Credit signoff: [signature] 11/9/2015

10/26/15
Benjamin Bergeson

LAB 4

Objective

In the first task we will solidify some concepts about signals with LTI systems. We will record some audio at 96k samples/sec then convolve them with some impulse responses also sampled at 96k samples/sec.

The second task we will solidify concepts about aliasing. We will generate tones at different frequencies and then sample them below the Nyquist rate

Task 1

a what sampling rate do you wish to use?

- The corner frequency for the given sampling rates would be ½ the sampling rate. For the 96kHz sampling rate we will be using, we want the corner frequency to be 48kHz.

- It is not realistic to assume a perfect, ideal anti-alias filter with a zero-width transition band, because there is ~~not~~ no such thing as a brick wall filter.

b How many bits do you wish to record each sample? ~~16bit~~ 24 bit

- Using 8 bits would use less memory, however the quality wouldn't be as good. Using 24 bits would have better quality but takes up more memory.

c How many ~~ch~~ sound channels will you be recording? 1

myRecording had 480,000 items in the array. This is because there is 96k samples/sec and we recorded for 5 seconds.

We tried playing back what we recorded at 70kHz (below 96kHz) and 120kHz (above 96kHz). Our voices got lower at 70kHz and higher at 120kHz. This is because we recorded the audio at 96k samples/sec, so were playing it back slower than we recorded for 70k and faster than we recorded at 120k.

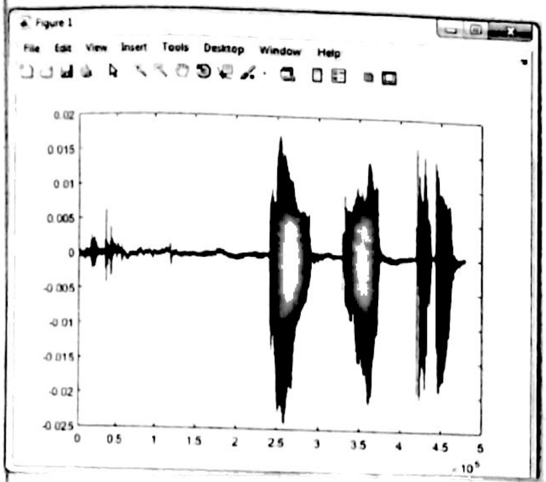
11/2/15
Benjamin Bergman

Code and plot for recording and playing back at difference frequency

Task 1

```
recObj = audiorecorder(96000, 24,1);
disp('Start speaking.')
recordblocking(recObj,5);
disp('End of Recording.');

%play(recObj);

myRecording = getaudiodata(recObj);

plot(myRecording);

player = audioplayer(myRecording, 70000);
playblocking(player);

player = audioplayer(myRecording, 120000);
playblocking(player);
```

11/2/15
BB



4. Code to create impulse response

We heard ~~successive~~ progressively quieter clicks.

```
impulse_echo = zeros(192000,1);
impulse_echo(1) = 1;
impulse_echo(48000) = 0.5;
impulse_echo(96000) = 0.2;
player = audioplayer(impulse_echo, 96000);
playblocking(player);
```

11/2/15
BB

11/2/15

LAB 4

Task 1

5. Using the code below we heard echos to our voices. Because of the convolution, we heard the same recording start to play again at lower volumes after 1 second and ½ second.

```
recObj = audiorecorder(96000, 24,1);
disp('Start speaking.')
recordblocking(recObj,5);
disp('End of Recording.');
%play(recObj);
impulse_echo = zeros(192000,1);
impulse_echo(1) = 1;
impulse_echo(48000) = 0.5;
impulse_echo(96000) = 0.2;
myRecording = getaudiodata(recObj);
myRecording_echo = conv(myRecording, impulse_echo);
player = audioplayer(myRecording_echo, 70000);
playblocking(player);
```

11/2/15 BB

6. We used the code below to hear our recordings convolved with the impulse response of the great hall, octagon, and classroom.

```
recObj = audiorecorder(96000, 24,1);
disp('Start speaking.')
recordblocking(recObj,5);
disp('End of Recording.');
%play(recObj);
impulse_echo = zeros(192000,1);
impulse_echo(1) = 1;
impulse_echo(48000) = 0.5;
impulse_echo(96000) = 0.2;
myRecording = getaudiodata(recObj);
myRecording_echo = conv(myRecording, impulse_echo);

impulse_great_hall = wavread('great_hall.wav');
impulse_octagon = wavread('octagon.wav');
impulse_classroom = wavread('classroom.wav');

myRecording_great_hall = conv(myRecording, impulse_great_hall);
myRecording_octagon = conv(myRecording, impulse_octagon);
myRecording_classroom = conv(myRecording, impulse_classroom);

player1 = audioplayer(myRecording_great_hall, 96000);
player2 = audioplayer(myRecording_octagon, 96000);
player3 = audioplayer(myRecording_classroom, 96000);

playblocking(player1);
playblocking(player2);
playblocking(player3);
```

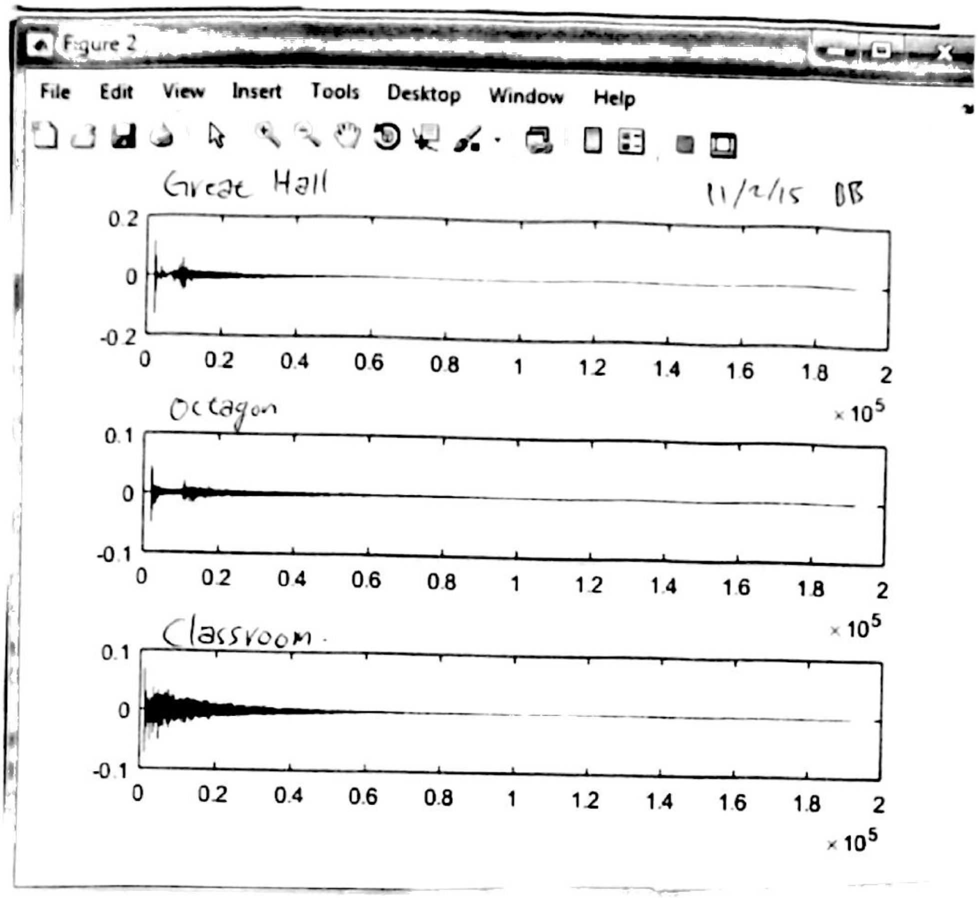
11/2/15 BB

Our recording got more echos as were down the list.

11/2/15
[illegible signature]

Task 1

Here is the plots for the impulse response of the different rooms.



Here's the code for the Task 1 pass off

11/2/15 BB

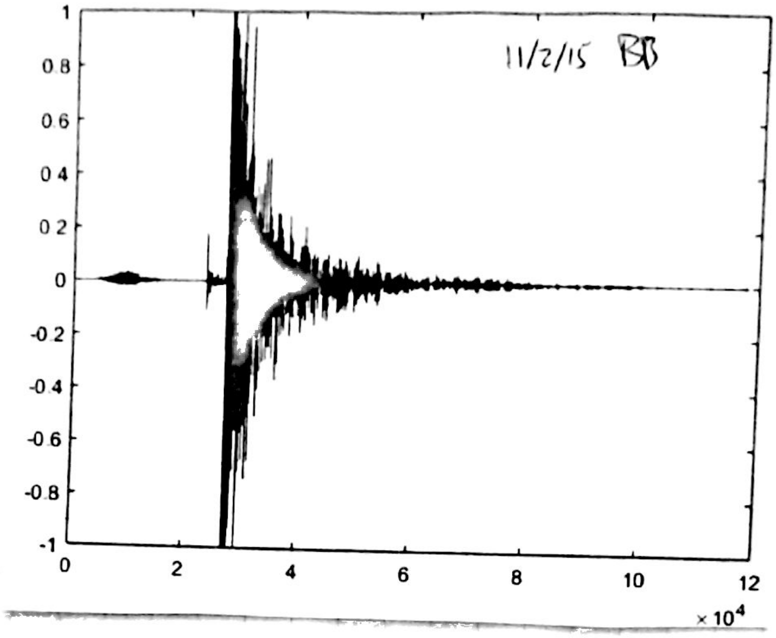
```
recObj = audiorecorder(96000, 24,1);
disp('Start speaking.')
recordblocking(recObj,5);
disp('End of Recording.');
%play(recObj);
impulse_echo = zeros(192000,1);
impulse_echo(1) = 1;
impulse_echo(48000) = 0.5;
impulse_echo(96000) = 0.2;
myRecording = getaudiodata(recObj);
myRecording_echo = conv(myRecording, impulse_echo);

impulse_great_hall = wavread('great_hall.wav');
impulse_octagon = wavread('octagon.wav');
impulse_classroom = wavread('classroom.wav');

myRecording_echo = conv(myRecording, impulse_echo);
myRecording_great_hall = conv(myRecording, impulse_great_hall);
myRecording_octagon = conv(myRecording, impulse_octagon);
myRecording_classroom = conv(myRecording, impulse_classroom);

player0 = audioplayer(myRecording_echo, 96000);
player1 = audioplayer(myRecording_great_hall, 96000);
player2 = audioplayer(myRecording_octagon, 96000);
player3 = audioplayer(myRecording_classroom, 96000);

playblocking(player1);
playblocking(player2);
playblocking(player3);
```

11/2/15
[illegible signature]

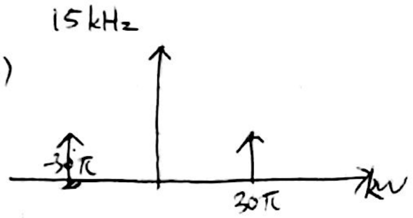
LAB 4

Extra Credit

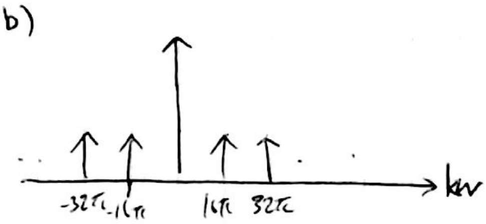
We recorded a balloon pop in the hallway of the 5th floor of the Clyde. When convolved with our audio recording, there weren't too many differences. It just sounded like our recording with a lot more noise. Below is the impulse response.



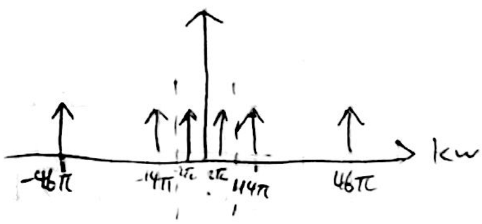
Task 2 1 a)



b)



c)



d) We will hear it at $2\pi k$ rads/sec = 1 kHz

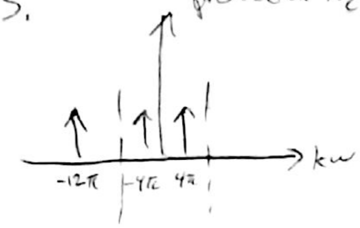
11/9/15

Task 2

2 c) We can hear it, although the volume is not as loud.

2e) The frequency is ~~pt~~ much lower. It matches exactly with our predictions. ~~It~~ It sounds exactly like a 1kHz tone.

3. prediction: 2kHz tone.

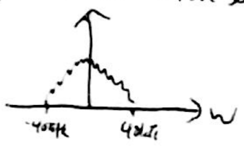


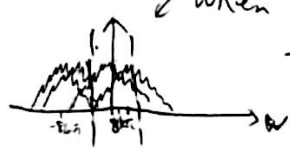
The output tone sounds exactly like a 2kHz tone

4. It oscillates at lower frequencies. This is what we should have expected. The oscillation is because the delta functions within the 8kHz filter range is moving from large numbers to small then large again through the opposite sign.

5. Both recordings were not as good quality as the original. However, the first one had obviously worse quality.

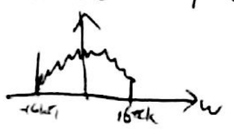
a) music when sampled at 48000 samples/sec.

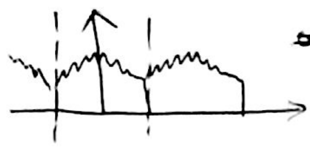




When it's undersampled ~~the signal~~ overlaps a lot

b) music sampled at 8k sample/s.





~~sth~~ no overlap.

code for recording the input.

```
recObj = audiorecorder(48000, 24,1);
recordblocking(recObj,5);
disp('End of Recording.');
my48kHzRecording = getaudiodata(recObj);
my8kHzRecording = my48kHzRecording(1:6:length(my48kHzRecording));
player = audioplayer(my8kHzRecording, 8000);
playblocking(player);
```

11/9/15 BB

11/9/15
Benjamin Bergeson

LAB 4

Task 2

6b) ~~[illegible]~~ 11kHz
Looking at our response plot, it is attenuated by -22 dB.

Conclusion

Conclusion

In this lab we were able to learn more about aliasing and how that affects the quality of the signal. Through testing sampling at frequencies above and below the Nyquist rate. We noticed the obvious loss of quality when signals are sampled below the Nyquist rate.

In the first task, we were able to acquire recording of our voices. We then played them back at different frequencies and noticed that ~~we~~ when we played ~~at~~ the recording back at higher frequency than our initial sampling rate, our voices were higher pitched ~~at~~ and the recording played back much quicker. We then were able to use MATLAB to generate an impulse response with two decreasing volume echoes. When we convolved our recording with the impulse response we could hear the echo of our voices as well. Lastly, we convolved our recording with the impulse response of various venues.

11/15/2015
Benjamin Bergstein

Conclusion

For task 2 we wrote code to record the tone generated by sine waves at different frequencies Then we sampled these recordings at below the nyquist rate. This yielded an output audio ~~in at~~ in a different tune than what was inputed. ~~Then~~ Then we used the Butterworth filter we built for the last lab to filter out signals before they are even sampled.

11/15/2015
Benjamin Bergeron