ECE 280L: Introduction to Signals and Systems

Lab 5: Sampling and Aliasing

Nathan Lewis

Cemal Yağcioğlu

November 9, 2017

Duke University

Instructor:	Professor L. Collins
TA:	Chance Fleeting
Section:	ECE 280L-04L

I have adhered to the Duke Community Standard in completing this assignment.

____

____

Contents

1 Objectives

In this lab we explore sampling signals, the aliasing that can occur when this happens and how anti-aliasing filters can counteract this. There are few objectives to this lab, first being able to use MATLAB and SIMULINK to visually see the effects of aliasing via graphs and hear it using audio output. A second objective is using the programs available to us to implement an anti-aliasing filter and see the effects of that on screen with the spectrum analyzer and as audio output. The final objective of the lab is that having completed this lab we should have a comprehensive knowledge as to why we need to apply this anti-aliasing pre-filter to continuous digital signals in most cases to achieve best results.

2 Background

Sampling is the process of taking a continuous time analog signal and converting it to a digital format by taking discrete samples of the signal at a specific sampling period. Aliasing occurs when the sampling frequency is less than double the frequency of the input signal (Nyquist sampling rate) and results in distortion of the original signal. In telephone systems these techniques are utilized as the analogue signal from someone's voice is transferred from the handset to the attached phone exchange via analog form and then sampled (typically at 8kHz) and sent across the telephone network as digital signals and reconstruction (digital to analog) occurs at the phone exchange on the other end.

When music is recorded the analog sound signal in the recording room is picked up by microphones and that analog signal is then sampled and quantized (at a much higher sampling rate than telephones) to convert it to a digital signal that can be shared over the Internet for example.

3 Results and Discussion

Explain the purpose of each block in the Sampling w/ Aliasing model in Exercise 1

Audio Device Reader reads the sound input that is generated by signal generator. Down Sample block down samples the input with 48kHz frequency to 8kHz (1/6) frequency as its parameter is 6. In1 Out1 block increases frequency back 48kHz for audio device writer to work as expected. Spectrum Analyzer shows the frequency vs amplitude graph of the resulting signal.

Using the plots generated by your MATLAB simulation, explain how the system (without anti-aliasing) works and why it behaves the way it does.

Our MATLAB simulation creates sin waves of differing frequencies and samples them at a sampling rate of 8kHz and plots the sampled graph. In other words this system essentially retrieves the amplitude of a sin wave every 1/8000th of a second and adds this value to a vector which we can then plot against time - essentially taking discrete values of the sin wave. In our MATLAB plots we can see that as the fundamental frequency of the sin waves we produce goes beyond 800 Hz to 7200 Hz and above the plot of the sampled signal compared to the original signal (which we plotted with a much higher sampling rate -

50kHz) appears to have a much larger period than the original signal does. This is because the fundamental frequency is way above half of the sampling frequency meaning these discrete values being sampled from the original signal, when combined and joined together appear to produce a signal with a much lower period, which does not represent the original signal.

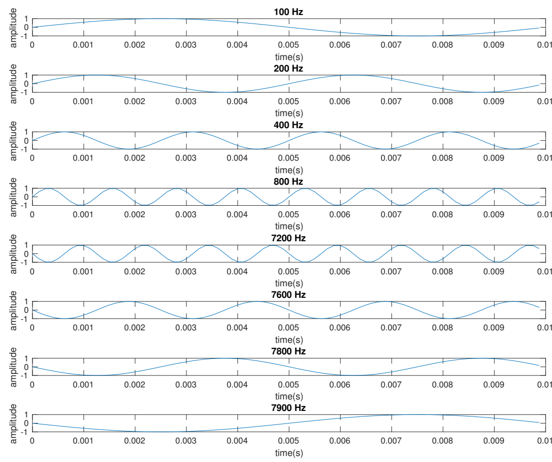

Figure 1: Plots showing the sampled signal of different sin waves with specified fundamental frequencies

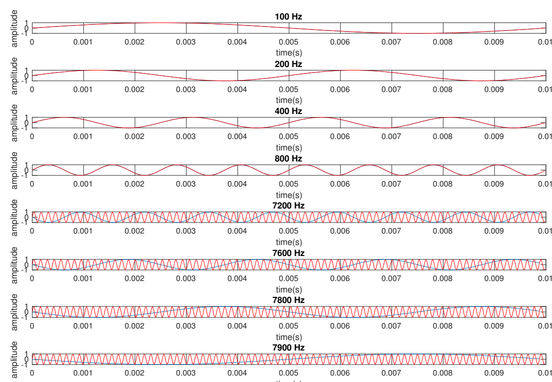

Figure 2: Plots showing the sampled signal of different sin waves with specified fundamental frequencies with the original signal in red overlayed

Function to produce sample sin wave

```
function output = SampleWave(frequency, amplitude,fs)
    t = 0:(1/fs):(0.01-0.01/fs);
    y = amplitude*sin(2*pi*frequency*t);
    output = y;
end
```

Script to plot sampled sin waves and 'original' sin waves

```
fs = 8000;
frequencies = [100, 200, 400, 800,7200,7600,7800 7900];
freqStrings = {'100','200','400','800','7200','7600','7800','7900'};
for i=1:size(frequencies,2)
    fs = 8000;
    f = frequencies(i);
    y = SampleWave(f,1,fs);
    t = 0:(1/fs):(0.01-0.01/fs);
    fs2=50000;
    y2 = SampleWave(f,1,fs2);
    t2 = 0:(1/fs2):(0.01-0.01/fs2);
    subplot(8,1,i)
    plot(t,y)
    hold on;
    plot(t2,y2,'-r')
    title(strcat(freqStrings(i),' Hz'));
    xlabel('time(s)');
    ylabel('amplitude');
end
```

Describe, in your own words, how the Sampling Anti-Aliasing model (from Exercise 2) differs from the Sampling w/ Aliasing model in Exercise 1. (You may ignore the Subsystem block.) Why is the anti-aliasing filter necessary?

In the Sampling Anti-Aliasing model there is an addition of a Digital Filter Design block that appears before the down-sampling occurs. This block represents a low-pass filter and applies the effect to the input signal. This essentially reduces the frequency components of the signal that are above 3.5kHz before sampling them which is necessary for removing the distortion in the sampled signal caused by aliasing.

Assume that the input to your system is music (with a highest frequency of 22.05 kHz). Sketch the spectrum of the signal at the following points in your models:

`Sampling w/ Aliasing Model`

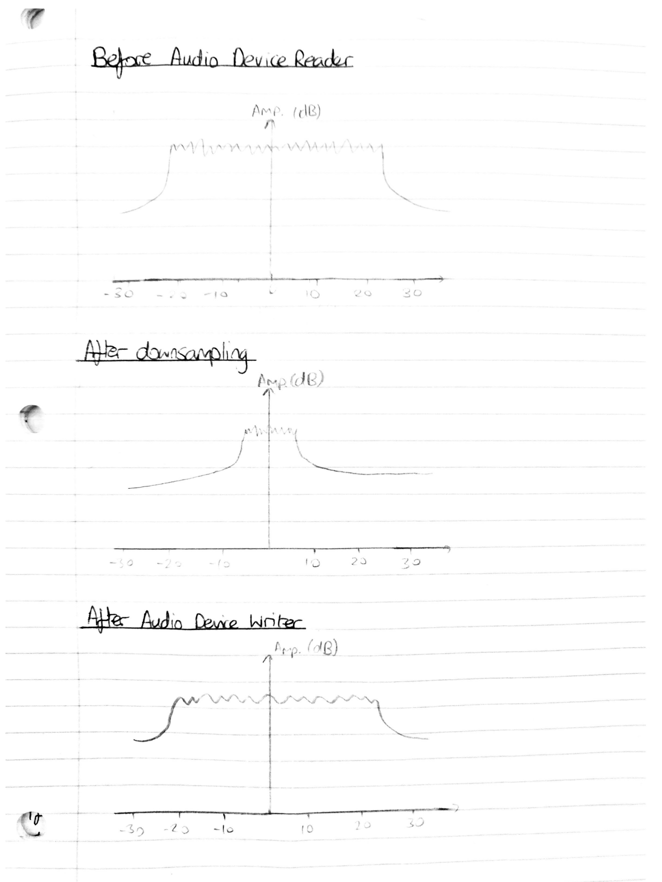

Figure 3: Sketches showing Frequency vs Amplitude spectrum of a music signal before sampling, after sampling and after reconstruction

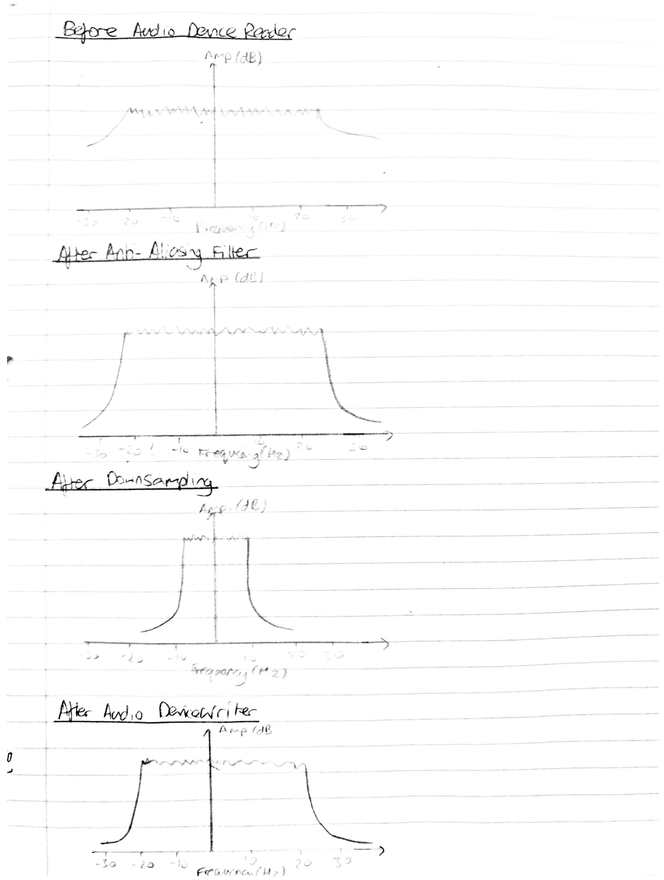

Figure 4: Sketches showing Frequency vs Amplitude spectrum of a music signal before sampling, after anti-aliasing filter, after sampling and after reconstruction

Questions included in the instructions

a. **Run Double-Click on the Spectrum Analyzer (to see the frequency domain signal). You should now be able to both hear and see the output signal. What do you expect to hear/see? Is this actually the case?**

- It is expected to see peak at frequency at 100 Hz as spectrum analyzer is in frequency domain. It is also expected to see two peaks, one to the right of 0, and one to the left of 0Hz. The observed output matches to the expected output, two peaks symmetric about x=0 and are at x=100Hz and x=-100Hz. As expected, a comparatively higher frequency sound is heard.

b. **Now, vary the input signal so that the frequency of the sinusoid is 200 Hz. What do you hear/see at the output? Make similar observations when the frequency of the input signal is 400 Hz and 800 Hz. Describe what you hear and explain why this is what you would expect.**

- As expected, the output seen as two maximum's at the given frequency. Thus when it is increased from 200Hz to 400Hz and to 800Hz, these two maximum nodes starts getting away from the center x=0 line. In addition to that the sound heard goes into higher pitch with each increase. As values 200,400 and 800 is much lower than half of the base frequency (8000/2=4000Hz), no aliasing caused by downsampling is expected to be observed. Thus, it is expected for the signals of these frequencies to act in a regular manner.

c. **Keeping the model running on the Audio System Toolbox, change the input frequency of your sinusoid to 7900 Hz. What do you see/hear? Record your observations as you change the frequency to 7800, 7600, and 7200 Hz. Describe what you hear and explain why this is what you would expect.**

- The frequency was gradually increased from 800Hz to 7900Hz first. During this, frequency of the sound heard increased at first, and decreased afterwards. The maximal points in the graph slowly diverged from each other, until they came to the —x—=4000Hz where any increase in frequency led disappearance of those nodes. This is expected as anything above 4000Hz would be too undersampled for a correct extrapolation meaning alias occurs. Moving the frequency around 7900 Hz to 7800,7600 and 7200 caused little changes to the output. For all those values, sound heard was very unclear. Lowering input frequency from 7900Hz to 7200Hz resulted in higher pitch sound heard as output. This is also expected as 7900Hz has bigger portions of overlap on frequency domain when replicated, and therefore has a higher aliasing occurrence. The undersampled sinwave cannot follow the frequency of the input sign wave and this effect is also reflected with the output figures of the matlab code.

d. **What is the maximum frequency that will pass through the system without aliasing? Verify your answer by changing the frequency of your sinusoidal input signal and listening to the output of the system and viewing the spectral analyzer**

- Maximum frequency that will pass through the system without analiasing is 8000Hz/2 = 4000Hz. This is the point where maximal nodes reach at the end of the high amplitude region in Spectrum Analyzer.

e. **Change the audio device reader block to a From Multimedia File block and set the Samples per Audio Channel to 60. Set the File Name to an audio file (Audio File is available on Sakai) What do you observe about the quality of the output signal? Why is this the case?**

- Quality of the output signal is very low. The output is too distorted to understand the conversation in the audio file ("may the force be with you") or clearly hear the sweeping sound especially in originally higher frequencies. This is the case because aliasing causes replicas that poorly correlates to the real input.

f. **ANTI-ALISING PRE-FILTERING:Double click on the blocks of this model and be sure you are able to explain how this model works. You will need to include this explanation in your final report. Pay particular attention to the Anti-Aliasing Filter block. What is the cutoff frequency of the filter? Why is this frequency appropriate for the telephone system application? Generally, how should the cutoff frequency of an anti- aliasing filter be chosen?**

- The model works in exactly same manner except the Low-Pass Filter added after the input signal. Antialiasing added to the system modifies the input before it is down sampled. With low pass filter, it lowers the amplitude of any sounds with frequency higher than the cut-off frequency, which is 4kHz in this case. Thus for frequencies higher than 4kHz, the amplitude is lowered before it actually goes into the system. This prevents an output with aliased signals by simply ignoring the inputs with higher frequency than the threshold.
- The cutoff frequency of the filter is 4kHz, after 4kHz the amplitudes are gradually decreased to prevent aliasing. The cut-off frequency is not a point of drastical decrease of amplitude to zero as its not an ideal low pass filter. 4kHz is an appropriate threshold for telephone systems as human voice ranges up to 3.5kHz and additional 0.5kHz is left to make sure system works as expected even in rare occasions. Generally cut-off frequencies as chosen to be a little higher than the maximum frequency expected to be transferred to ensure all expected signals are transferred.

g. **Using the function generator, create sinusoidal signals with frequencies of 100, 200, 400, and 800 Hz and process them with the system. Describe what you hear/see and explain why this is what you would expect.**

- A similar output to the exercise one is observed. As all of these values are under 4kHz cut-off frequency, no anomalies are expected and no anomalies are observed. The maximal nodes in frequency domain graph in Spectrum Analyzer diverges as frequency increases.

h. **Now, create signals with frequencies of 7900, 7800, 7600, and 7200 Hz. You will notice that instead of being completely cut-off, the amplitude just dropped significantly. Describe what you hear/see and explain why this is what you would expect.**

- Unlike the first exercises where outputs were still audible, the outputs for this model for the given range was hard to hear. In a similar manner, the maximal nodes in frequency domain went off from the 4kHz threshold. However, this time, instead of causing audible sounds of other pitches, the output's magnitude is lowered as frequency increases. Even though the threshold is 4kHz, the output's amplitude is not zeroed as the Low-Pass Filter used is not an ideal filter and rather smoothly lowers the amplitudes of frequencies higher than the threshold. Thus, this is expected.

i. **What is the maximum frequency that will pass through the system undistorted? Verify your answer by changing the frequency of your sinusoidal input signal and listening to the output of the system. How does this compare to your answer for the system in Exercise 1?**

- The maximum frequency that will pass through the system undistorted is still 4kHz. The difference is that anything higher than 4kHz will now be absorbed by the LPF before it results in aliased output.

j. **Change the audio device reader block to a From Multimedia File block and set the Samples per Audio Channel to 60. Set the File Name to an audio file (Audio File is available on Sakai). What do you observe about the quality of the output signal? Why is this the case? How does it differ from what you observed in Exercise 1?**

- Quality of the output noticeably increased. The quality lowering elements in the exercise 1, high frequency sounds, are eliminated before they go through the system. Thus, the output has either undistorted correct signals or incorrected signals with greatly reduced amplitude. The 'the force' audio is more audible. 'The Sweep' audio sounds much better, and the amplitude lowering effect can easily be heard as frequency gradually increases in the input sound, and the amplitude gradually decreases in the output sound.

k. **Based on what you have observed, discuss why the performance of the telephone system would degrade significantly if anti-aliasing pre-filtering were not used. Explain how this filtering prevents these negative effects.**

  - The main function of a phone is to transmit the signals in range of human voice in a high quality manner to the receiver. A telephone system without anti-alising pre-filtering would still transmit the human sound, however it would also transmit all the frequencies including the ones a human ear cannot hear. However, due to aliasing, those sounds that are inaudible to human ear would result in frequencies hearable by human ear due to down sampling of the phone's microphone. This would result in insertion of a lot of unwanted distortions to the transmitted sound, causing a phone conversation where parties cannot understand each other even though their voices are also transmitted. In addition to preventing that, anti-aliasing pre-filtering can also prevent some of the sounds coming from surroundings by limiting the frequency of transmission to only include the frequency range of human voice.

4 Conclusion

Having completed this laboratory there are a few key takeaways, firstly being how much we have improved our understanding in how sampling a continuous time signal works and how we can implement it in MATLAB and SIMULINK using both sinusoidal waves we produce or using live audio input. We've also learned how aliasing can occur when the sampling rate isn't appropriate and how this can distort the output signal, witnessing the effects both visually via the output spectrum and aurally. Additionally we learned how to implement an anti-aliasing filter and its importance in removing distortion caused by aliasing when higher frequencies are part of the input. Furthermore, we got a sense for what scenarios these techniques are used e.g. in music recording and telephone communication systems.

5 Extension

- The frequency of a sound is only a simple glance on effects of aliasing in our lives. Aliasing is happening anywhere where there is an important down sampling of any inputs. A human brain is programmed to connect the dots (similar to matlab), and create continous-like experience. A simple example for this can easily be given by how human's vision that is originally consists of too many discrete photographs added on top of each other every second to create this 'illusion' of continuous image. Akin to the sound experiment, one can test the possible consequences of such discrete input by observing a case where observed object's movement (similar to frequency in this experiment) is faster than a retina's frame per second rate(similar to base frequency in this experiment). A man observing wheels that are turning the half round(similar to fs/2) faster than the unit time of retina's discrete signals causes the man to see the wheels as if they are rotating backwards. This is because the man's brain connects the dots of discrete visual inputs. While trying to connect the point in r=1 and angle theta to x axis=30 degrees to the same point in one unit retina's observing time later to in r=1 and angle theta to x axis=40, rather than concluding that the point went 350 degrees clockwise, the brain concludes point went 10 degrees counter clockwise.

References

Duke University ECE Department, Lab 5: Sampling and Aliasing. Retrieved from Sakai, 2017.

Wikipedia, Anti-Aliasing. Retrieved from `en.wikipedia.org/wiki/Anti-aliasing`, 2017.

University/School Laboratory Report Template. (2016, March 25). Retrieved from Chance Fleeting September 2017.