Communications Lab 1 Report B05901092 歐瀚墨

1. Preface

2. Experiment Results

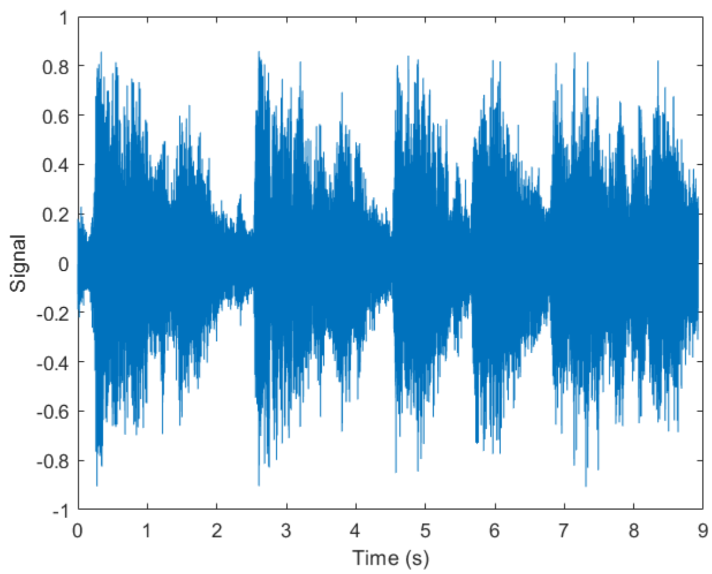
I. Manipulating audio files:

The signal read in is first transposed to make it a column vector.

a. When choosing $f = 2f_s$, the audio sounds like being played faster and with a higher tone. When choosing $f = 0.5f_s$, the audio is slower and only a low humming voice can be heard, the main tune is barely recognizable.

b. // TODO

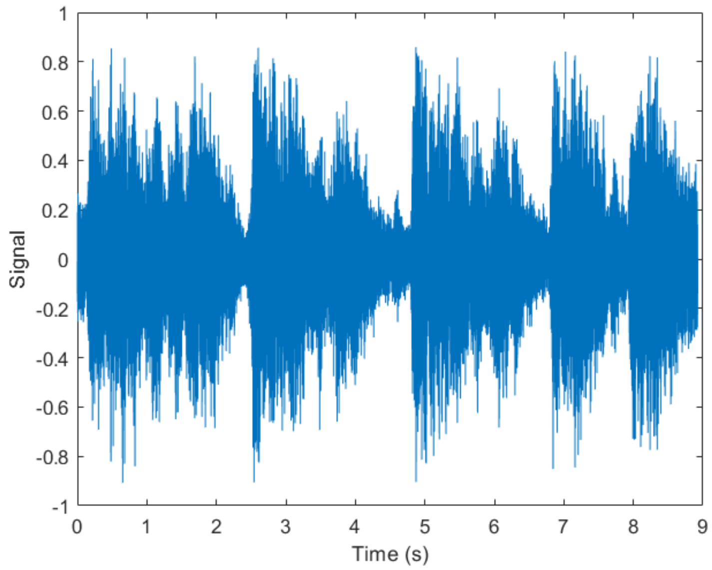
c. The plot is below with real time as X-axis



II. Redistributing the time index:

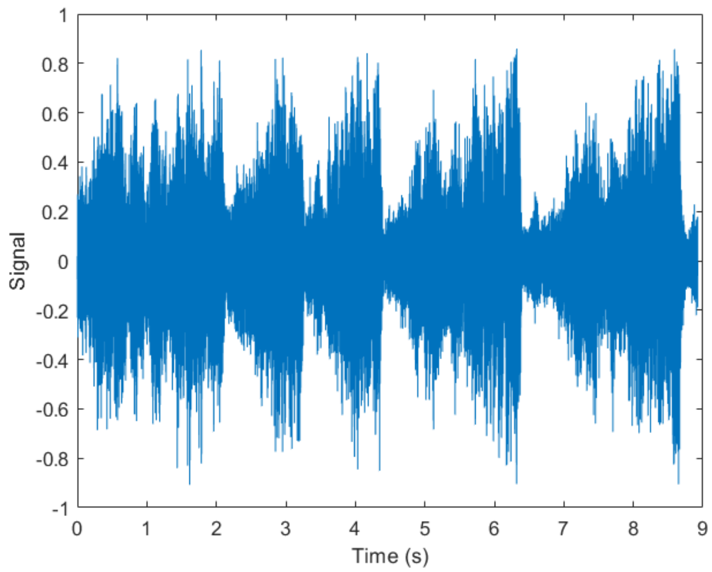
a. Circular Shift

The sound basically is the original one played from the middle and replays from the start to the starting point.



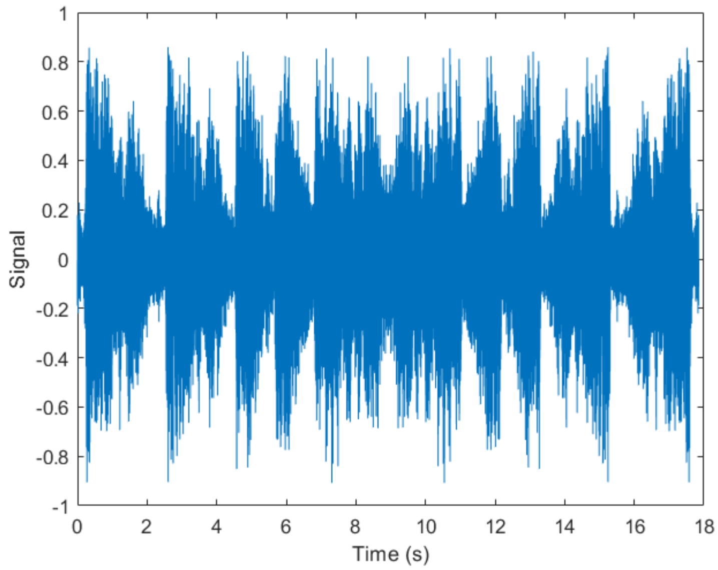
b. Reverse

The signal sounds like a totally different normal song, with the tune still very harmonic and peaceful; however, the lyrics played backwards sounded pretty weird. The figure below is the signal in reverse.



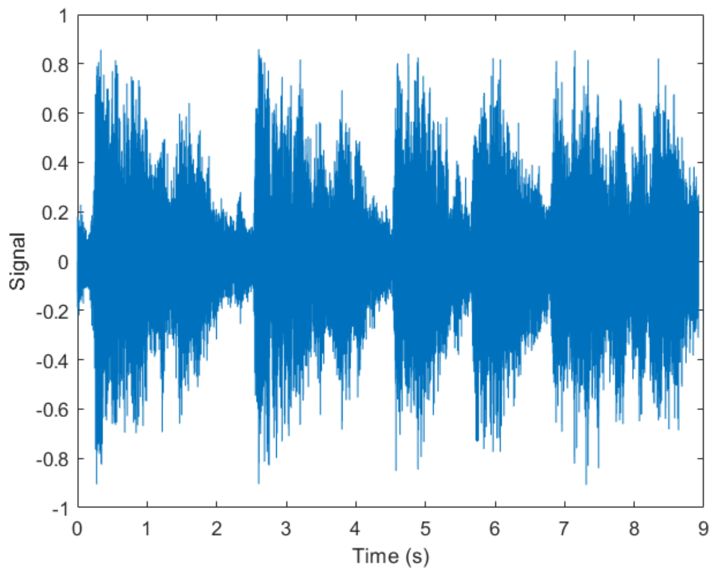
c. Forward and Backwards

I did this task by just merging the signal in (b) and the original one with the code “x_forNback = [x x_rev];” The signal sounds just like the song played forwards and backwards.

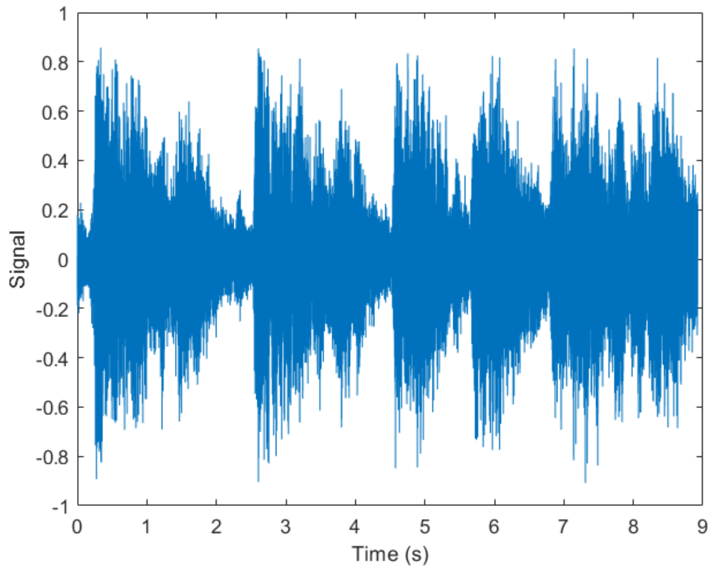


d. Upsampling, Downsampling

When upsampling, I first created a zero vector with the appropriate size, and then filled some indices according to the rate. For example, the figure below is the signal after upsampling with rate 2, the signal is filled with a zero between each sample. The signal resembles the original x, with minor distortions.

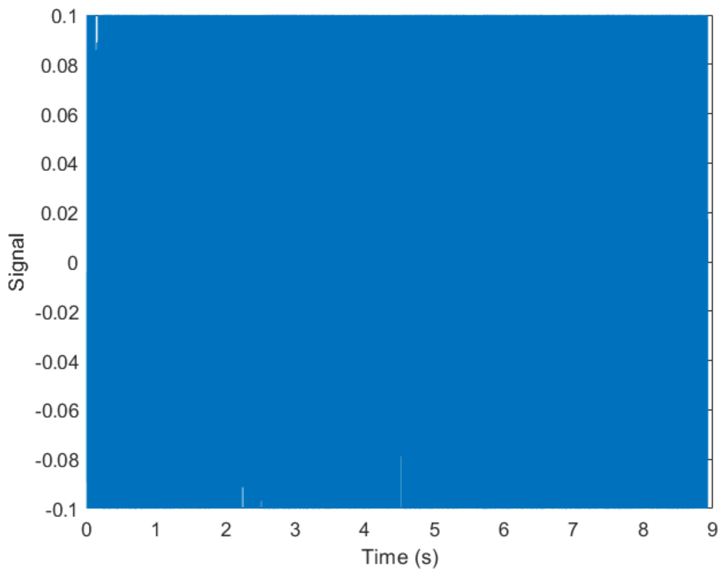


Downsampling is rather easier since I only have to skip every other sample. The figure is below. The signal sounds just like the original one, I personally cannot distinguish between the two.



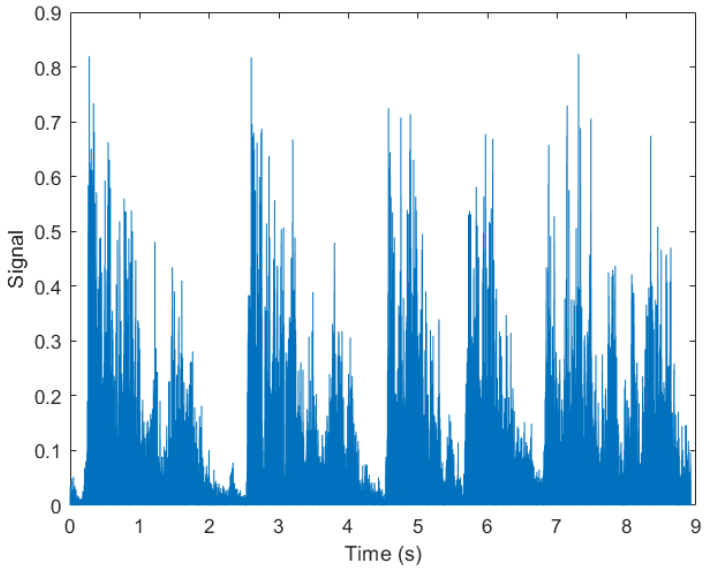
III. Amplitude distortion:

a. The first plot is the signal after applying a hard limit with $\mathrm{T} = 0.1$:



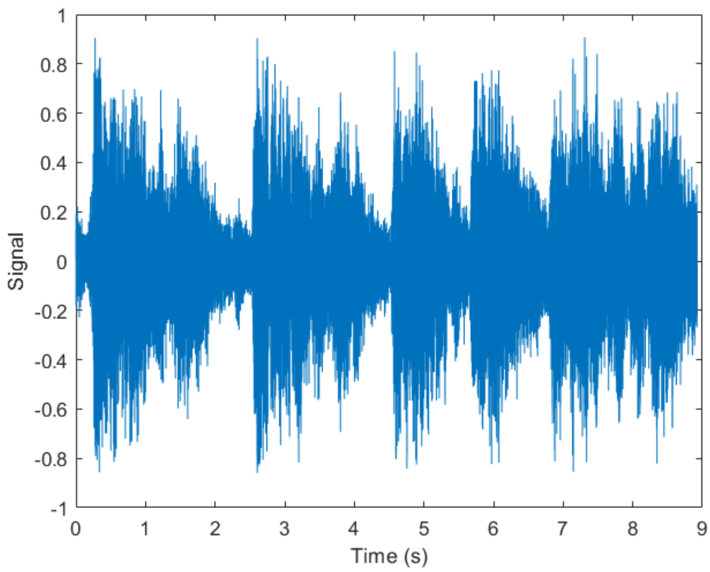
Since most signals are either bounded by T or -T, the signal on the figure looks very compact. The tune can still be heard, but with a certain degree of noise interference. The result here explains to me why some speakers produce noise when operating to provide a high volume, since it those circumstances the voltage limit of the circuits can be seen as the limit distortion.

Below is the signal squared:



The signal becomes positive since the square of all real numbers are positive. The tune becomes so distorted that the lyrics are incomprehensible. Significant amount of noise can be observed. The rhythm of the song is still there, and I believe that most people are able to name the song when listening to the squared signal.

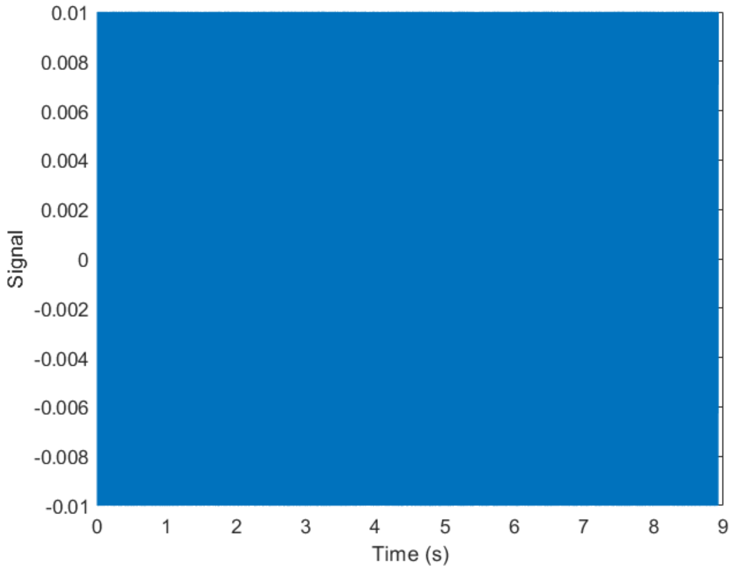
Below is the negated signal, it looks the same as the original one on the figure and sounds the same as well.



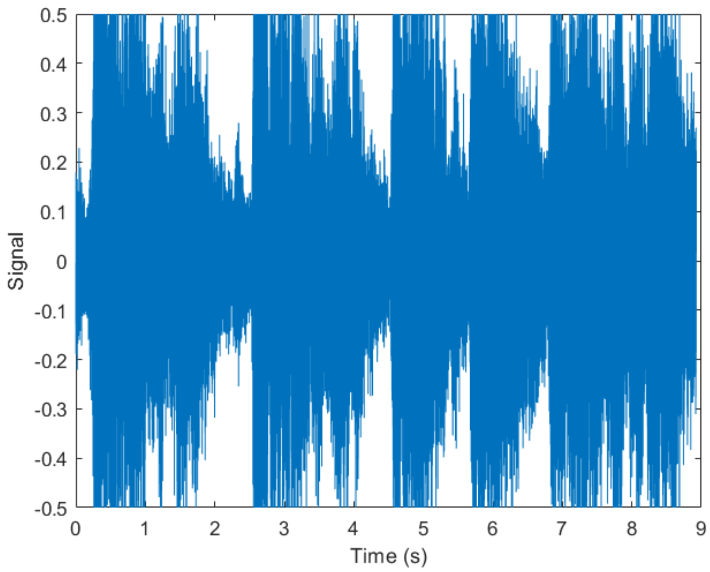
b. Different Ts

I did some experiments with $\mathrm{T} = 0.01, 0.05, 0.5$. As T becomes smaller, the signal become more distorted and the volume of the song decreases.

T = 0.01 and T = 0.05's figures resembles each other except of the y-axis marks, so I only paste the figure when T = 0.01.



When T = 0.5, most of the characteristics of the signal is preserved, and it sounds the same although the signal is cut. The figure is as follows.



IV. Quantization:

a. The expression of function:

$$\mathrm{d(x)} = \left(\left\lfloor\frac{x + x_{max}}{\Delta}\right\rfloor + \frac{1}{2}\right)\Delta$$

While: $\Delta = \frac{2\mathrm{x_{max}}}{\mathrm{L}}$

This formula assumes that x is strictly bounded by x_{max}.

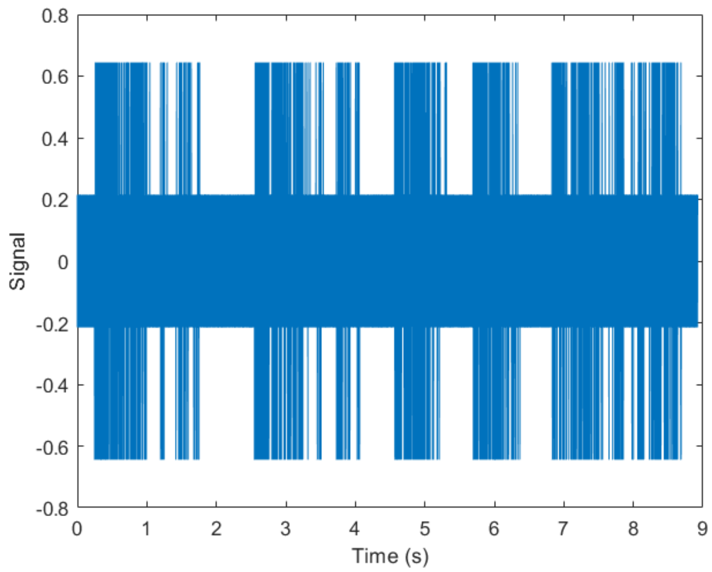
b. The code of the function is as follows

```
lab1.m  ×  quantizer_L_level.m  ×  +
1      function y = quantizer_L_level(x, x_max, L);
2 —    delta = 2*x_max/L;
3 —    y = floor((x+x_max)/delta)*delta -x_max + delta/2;
4 —    for ind = 1: length(y)
5 —        if y(ind) > (L/2-0.5)*delta;
6 —            y(ind) = (L/2-0.5)*delta;
7 —        end
8 —        if y(ind) < -(L/2-0.5)*delta;
9 —            y(ind) = -(L/2-0.5)*delta;
10 —       end
11 —   end
```

The code from line 4 adjusts the output in case that the value of x equals x_max.

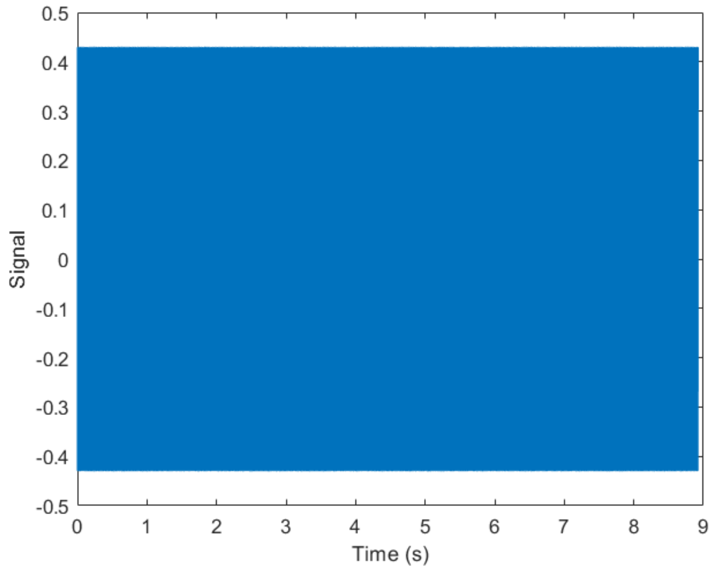
c. $L = 4$

Figure as follows, the x_max in the simulation is the maximum value of abs(x) , around 0.87, hence the need of the checks mentioned in (c). The sound is very similar to the original song, which is quite a surprising fact. This result tells us that we can compress each data sample to two bits, while still preserving a relatively good quality of listening experience.

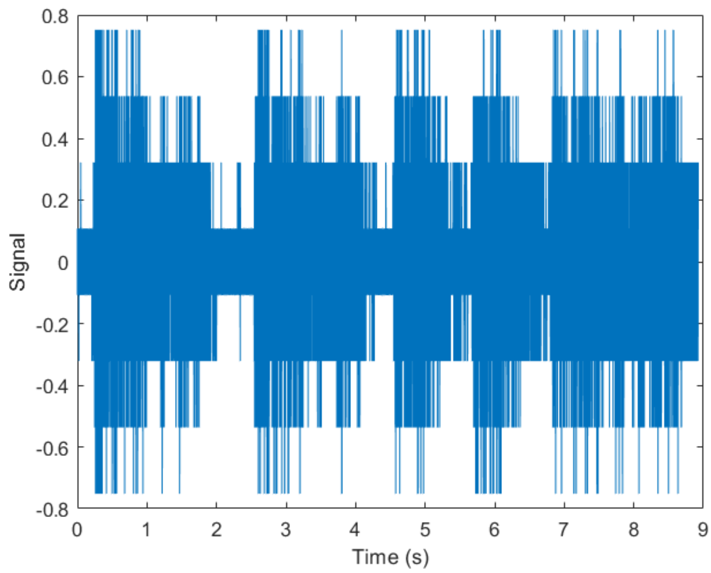


d. Experiments when $L = 2, 8, 15$

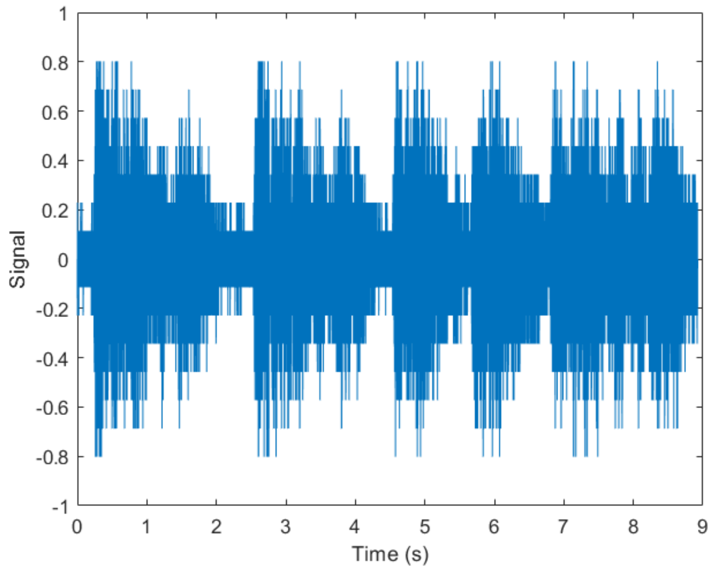
The following figures are the results with L = 2, 8, 15, respectively. Choosing 15 is because I felt the importance of choosing an odd number.



When L = 2, the only possible values are 0.43 and -0.43. However, the actual tune of the song is still recognizable. Some distortion can be observed when listening, but the effect isn't significant.



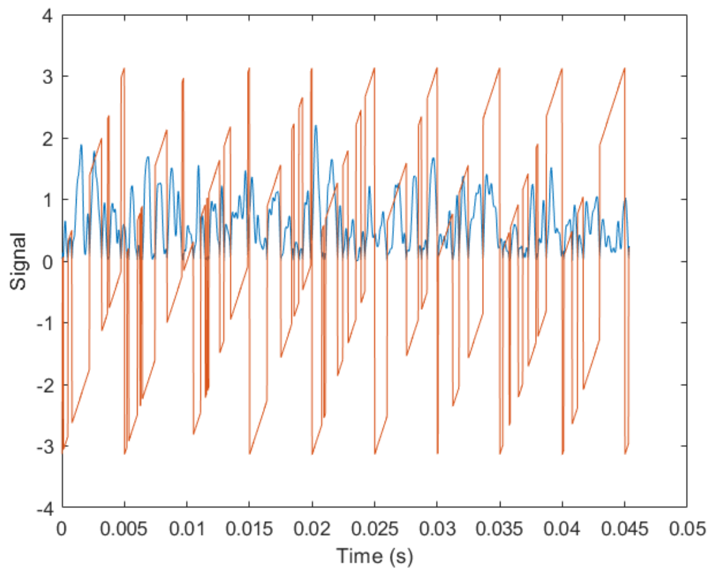
As L increases, the signal becomes more similar to the original file. The quality of the sound heard is good enough for most people.



In the case L = 15, the output of the quantizer can be zero, which is one property of quantizers mentioned in the slides.

V. Modulation:

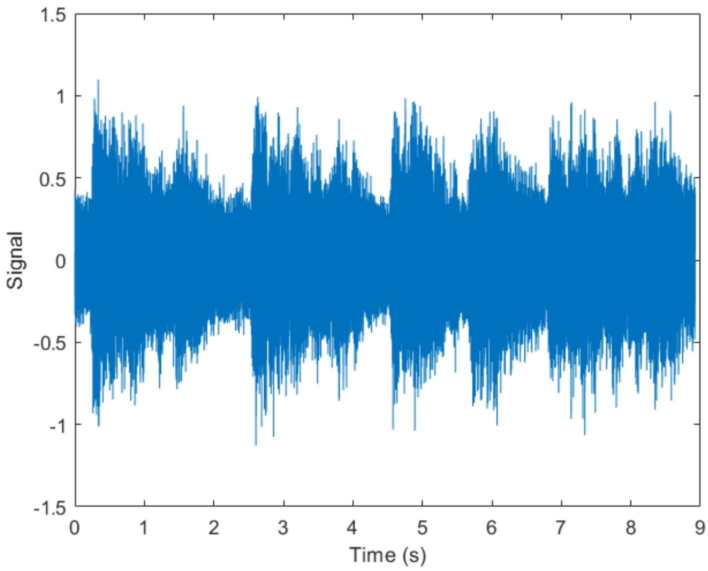
Modulation can be done by using the element-wise multiplication operator. The resulting plot of the signal is as follows. To get a better observation of the signal, I only plotted the first 0.05 seconds.



The resulting signal sounds completely wrong in terms of tones and chords. My explanation is that chords reflect on the ratio of frequencies rather than the difference[1]. Therefore, after a linear shift in their frequencies, the tones of the chorus no longer match.

VI. Noise:

The figure below shows the signal with Gaussian noise added on. The difference between this signal and the distorted signal above is that the noise sounds like a second channel, as the main tune remains in a very good shape.



VII. Filtering:

3. Remarks

4. References

[1] https://pages.mtu.edu/~suits/notefreqs.html