CPE 367 – Experiment 2 v2
Music Generation – Dr F DePiero
30 points, including report and WAV upload

Overview and Motivation

This experiment provides an opportunity to work with signals that have time-varying frequency content and to learn a little about music. You will write a Python program to generate a short musical piece, and will learn about signal envelopes and harmonics. MatLab functions are provided that generate time-frequency plots (spectrograms). Synthetic music that sounds like natural music is difficult to produce and we will use spectrograms to compare the two.

A comment for those with some background in music – when the notes of a chord are played simultaneously, some of the harmonics of the various notes occur at identical frequencies. For example, suppose the 1 note of a 1-3-5 major chord has a fundamental at f0. In a major chord, the 5 note has a fundamental at 1.5 f0. Thus, the 3rd harmonic of the 1 note and the 2nd harmonic of the 5 note each have a frequency of 3 f0. This is one example of the repeated harmonics of a chord and are a reason why they sound good.

Learning Objectives

- Interpret spectrogram plots that depict frequency versus time
- Define specifications for common waveform shapes (envelope and harmonics) and generate them in Python

Prerequisite Learning Objectives

- Introductory knowledge of MatLab as well as prior programming experience in Python
- Generate discrete-time signals via samples of CT functions (using t = n/fs)
- Find a digital frequency, given an analog frequency and sample rate $\omega_0 = 2\,\pi\,\frac{f_0}{f_s}$
- No knowledge of reading music or playing an instrument is necessary.

Procedures, Questions and Deliverables

1) Identify specific requirements for note frequencies and duration

The truncated piece of music below is translated into notes (e.g. “C5”, “E5”) in Table 1a.

To determine the frequencies and note duration, translate the sheet music below into note numbers and frequencies (Hz). This truncated piece is presented in both standard tablature and as a list of notes (e.g. “C5”, “E5”). Complete Table 1b, with note numbers. Use equation (1) to compute note frequencies for Table 1c.

"Jesu, The Joy of Man's Desire" by J. S. Bach.

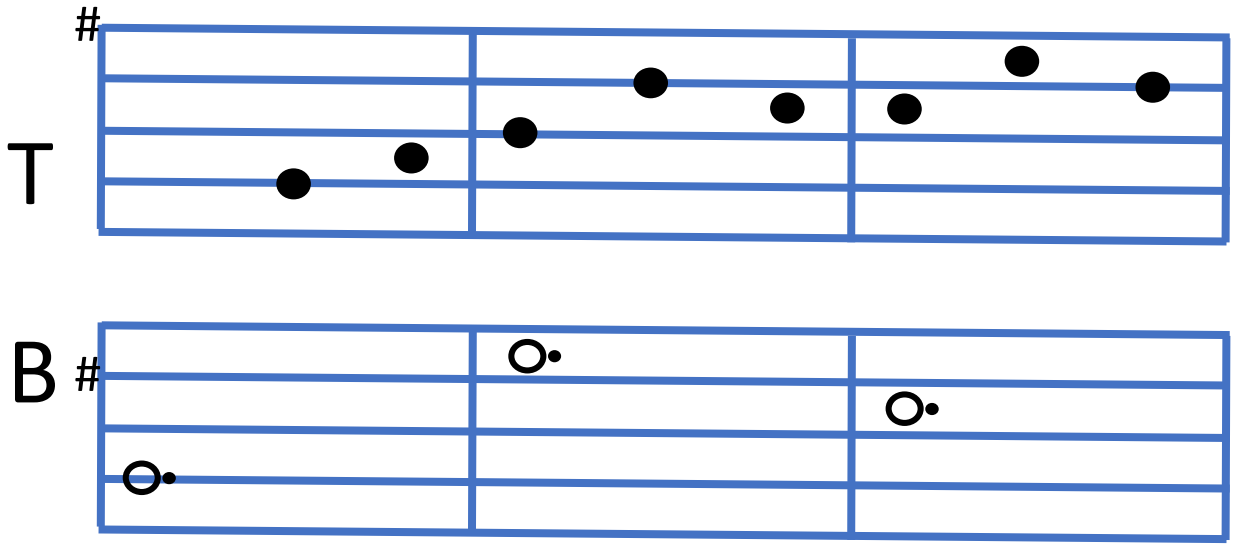


Table 1a - Notes

T		G4	A4	B4	D5	C5	C5	E5	D5
B	G2			G3			E3		

Table 1b – Note Numbers

T		**47**	**49**	**51**	**54**	**52**	**52**	**56**	**54**
B	**23**			**35**			**32**		

$f_n = 440 * 2^{(n-49)/12}$ Eq 1, From DSP 1st text, "n" is a note number and f_n is in Hz

Table 1c – Note Frequency (Hz) **1a) Compute the frequency of each note (2)**

T	**194.6**	**391.99**	**440**	**493.88**	**587.32**	**523.25**	**523.25**	**659.25**	**587.3**
B	**97.99**			**195.99**			**164.32**		

Table 1d – Note Duration (in beats) **1b) Specify the duration of each note (2)**

T	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
B	**3**			**3**			**3**		

This version of "Joy" is in ¾ time, meaning that the solid notes have a duration of 1 beat. The open dotted notes have a duration of 3 beats. Thus, there are 3 beats between the vertical blue lines (measures). Each beat should have a duration of **0.285** Seconds. The musical notes have been transcribed from the musical staff into note numbers in Table 1b. The "T and "B" designations on the musical staff refer to the treble and bass clef. This designation appears on Tables 1a - 1d also.

2) Identify specific requirements for envelope and harmonics

Generating a signal that only contains simple tone bursts does not yield a pleasant sound, as you will hear during the development process. To make a more pleasant sound, add both harmonics and an exponentially decaying envelope. The mathematical description of a mechanical oscillation provides insight regarding a more accurate waveshape (or envelope) for a plucked note. In general, the motion of a mechanical oscillation is described by a 2nd order underdamped differential equation. The solution of this type of differential equation has the general form of

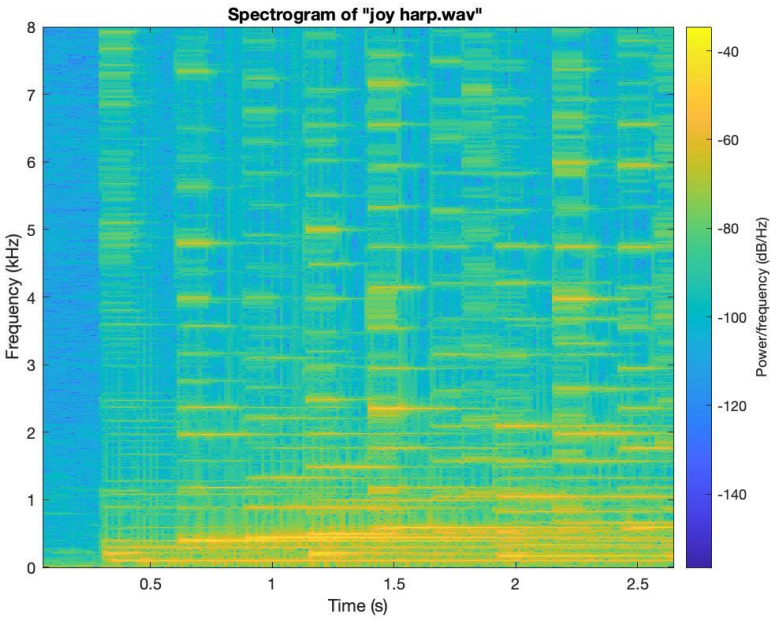
$$y(t) = A\ e^{-t/\tau} \sin(2\,\pi\, f_0\, t + \theta)$$

To generate the signal for a musical note, we will implement the discrete-time version of this. Substitute $t = n/f_s$ where n is a discrete-time sample index and f_s is the sample rate.

$$y[n] = A\ e^{-n/\sigma} \sin(2\,\pi\, \frac{f_0}{f_s}\, n + \theta)$$

To use this expression, you need the sample rate (16kHz), the note frequency (f_0, computed previously) and a value for sigma. The sigma parameter is the time constant of the envelope. It determines how quickly the note fades in intensity, as time increases.

The spectrogram below depicts the time varying frequency content of the music; time is plotted horizontally and frequency vertical. This musical piece is an excerpt of "The Joy of Man's Desiring" (J.S Bach) played by Amy Turk, https://www.youtube.com/watch?v=BrrzmAlGsIg



Spectrogram of Amy Turk's harp music.

The spectrogram shows both harmonics and decaying envelopes associated with each note. Observe that a quarter note essentially fades out in the time interval of one beat. Use this criterion to select a value for the time constant, sigma, which units of samples.

2a) Describe your method of determining the value of sigma for the envelope. What value did you choose (units are in samples). (4)

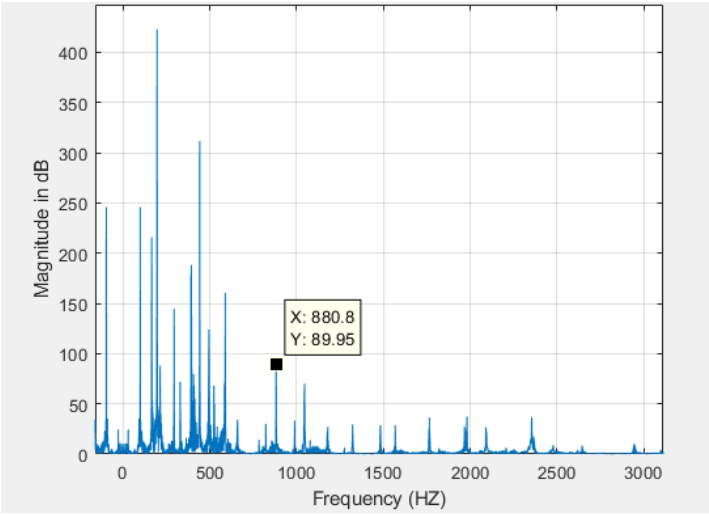
As you can hear throughout the production phase, creating a signal that only includes basic tone bursts does not produce a good sound. Adding harmonics and an exponentially decaying envelope creates a more pleasant tone. Following equation is used for the envelope.

$$y = A \times e^{\frac{-n}{\sigma}} \times \sin\left(\frac{2\pi f_o}{fs} n\right)$$

To find the value of sigma, hit and trial method is used. A signal of one of the beats using above equation is generated and played. Upon hearing the sound, we estimated the value of sigma. If the sound is played for very short time, sigma will be increased. In this way, value of sigma is selected to be 4000 which plays the sound for the desired time.

2b) Describe your method of generating harmonics. Include any relevant parameters. (4)

To generate harmonics, first the Fourier transform of the signal is determined using matlab built in function. Following graph is obtained.



As it can been from the FFT graph of the music signal, there are harmonics at frequency 821 Hz, 880 Hz, 1045 Hz, 1179 Hz, 2354 Hz, and so on. In order to incorporate theses harmonics damped sine wave are generated at these frequencies and added in the generated signal of each note. For instance

$$y = Ae^{\frac{-n}{\sigma}} \sin\left(\frac{2\pi a f_o}{fs} n\right)$$

Where **a** is the corresponding factor of the harmonics.

3) Generate Music in Python

Write a Python program to generate music in WAV file. The sample rate should be 16kHz. Set the amplitude to be approximately 75% of the maximum, for a 16 bit, signed number.

A main program is provided that first allocates an empty list to store the signal. It also defines a function, “add_note()” that adds a contribution to the signal (e.g., one note). Lastly it outputs

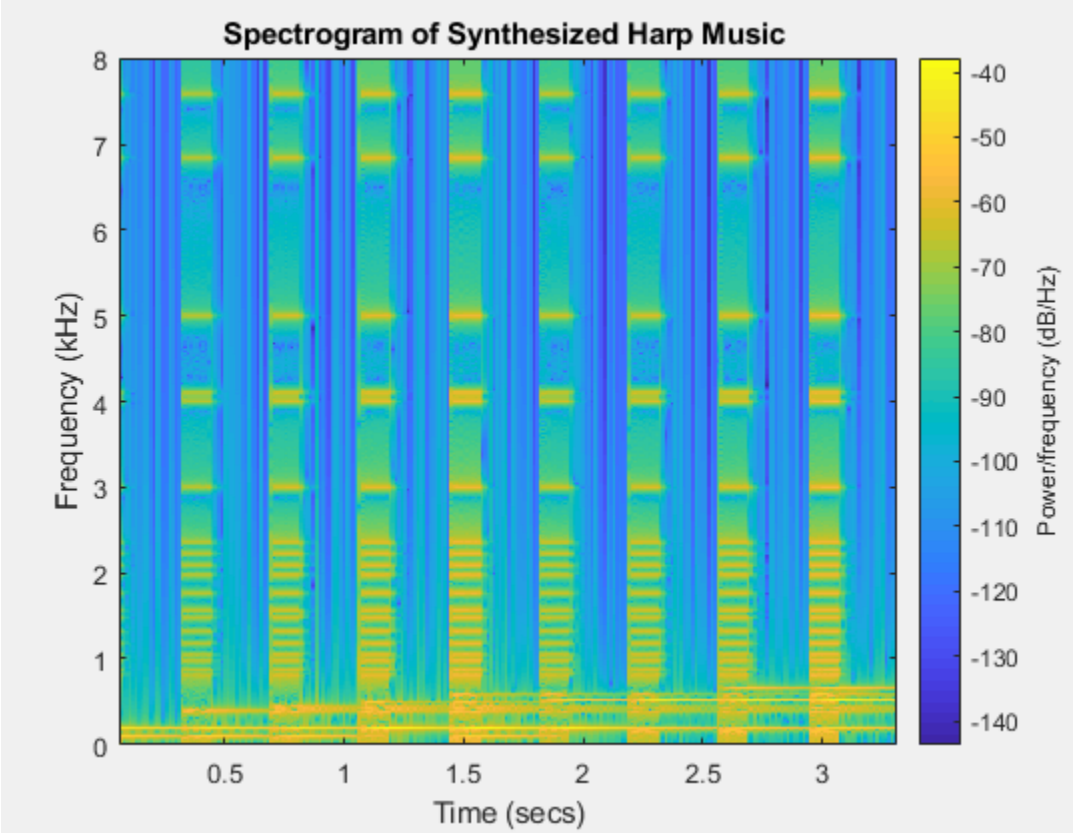
the final contents of the list to a WAV file. You are invited to modify this program, specifically the definition and usage of the add_note() function.

See the example in cpe367_music_gen.py which includes the cpe367_wav.py class.

4) Compare Spectrograms of Synthetic and Natural versions of Harp Music

Modify and run the MatLab script **spectrogram_signal.m** and generate a spectrogram of your synthetic harp music.

4a) Include a spectrogram of your synthesized harp music (2)



4b) Describe differences in the synthetic and harp spectrograms. Compare similarities or differences in envelopes or harmonics. (4)

In spectrogram of synthetic harp music, the note durations are slightly less. However, the total time is above 3 seconds which is more than what was encountered in spectrogram of original music. Also, the number of beats are less in the spectrogram.

Some of the harmonics in both of the spectrograms are same. However, duration of harmonics as well as the numbers of harmonics are less in the synthetic spectrogram. The envelope detected is relatively same, but the signal fades faster in case of synthetic spectrogram.

4c) Include sections of your add_note() function here, with comments, as well as sections of your program that call add_note() (6)

The following code will generate a loop with notes 60,62 and 64 playing in sequence. The length and distance between notes will be 0.285 a beat. The velocities will loop as follows: 100, 90, 100, 90, etc.

```
from midiutil.MidiGenerator import MidiGenerator
from midiutil.TrackGen import LoopingArray

midiGenerator = MidiGenerator(tempo=105)

# Create a looping array of notes
notes =
LoopingArray([[194.6],[392],[440],[493.88],[587.32],[523.23],[523.23],[659.25],[587.3]])

# Create a looping array of beats (note duration, note length)
beats = LoopingArray([(0.285,0.2855)])

# Create a looping array of velocities
velocities = LoopingArray([100,90])

midiGenerator.add_track(0, 0,
                                                    beat=beats,
                                                    notes=notes,
                                                    velocities=velocities,
                                                    length=9)

midiGenerator.write()
add_note()
```

4d) Deliverable: Upload a WAV file of your synthetic music to Canvas (6)