CPE 367 – Experiment 2 v2a
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 chords 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.

Use Eq 1 to determine the frequencies (Table 1c, in Hz) of each note, using the note numbers in Table 1b. Referring to the sheet music, complete Table 1d, specifying the note duration (in beats).

“Jesu, The Joy of Man’s Desire” by J. S. Bach.

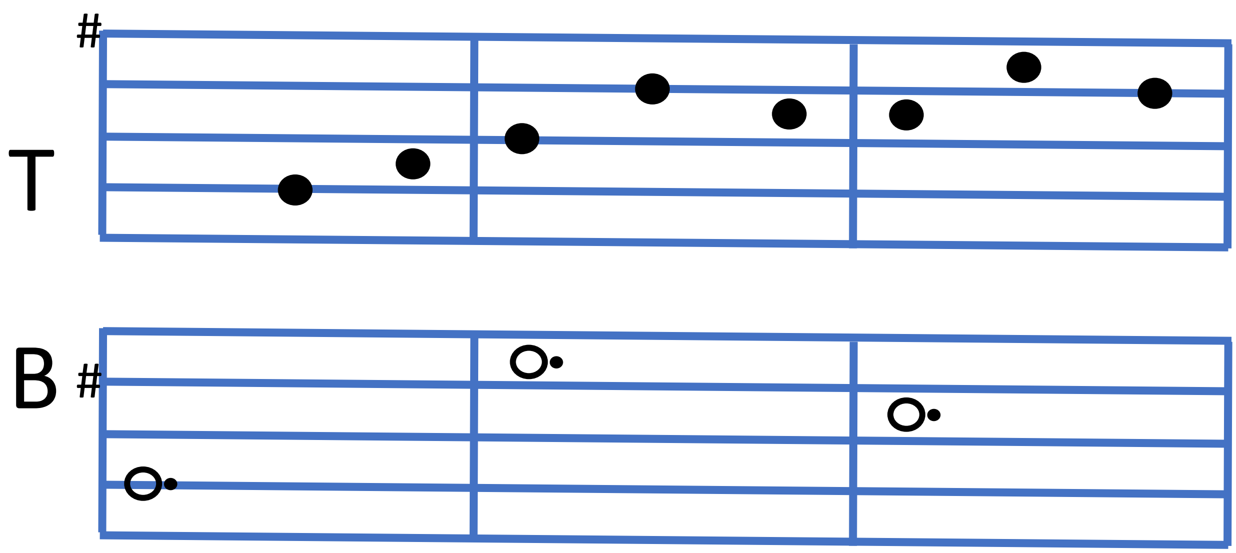


Table 1a – Notes (A-G is the note of the musical scale and the numbers below designate an octave)

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									
B									

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

T									
B									

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). When you generate the music, 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

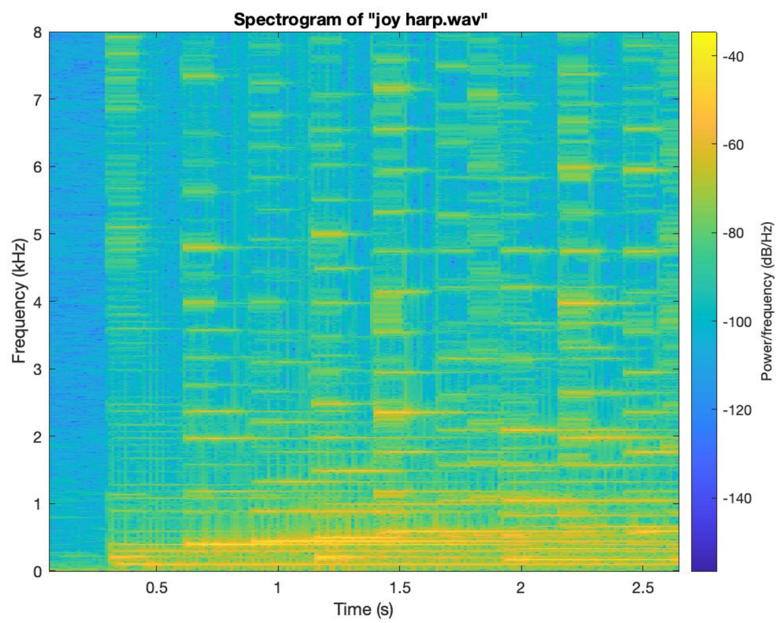
$$y(t) = A\ e^{-t/\tau} \sin\left(2\,\pi\, f_0\, t + \theta\right)$$

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.

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

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 has 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)

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

3) Generate Music in Python

Write a Python program to generate music in WAV file. The sample rate should be 16kHz. Set your amplitudes such that the superposition of the fundamental frequencies and harmonics add to 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. You’ll likely want to create a data structure to store the note frequency, duration and other info. You’ll need to make decisions regarding harmonics, envelope parameters, amplitudes and phase shift (if any). You may want to vary these parameters for notes in the treble clef (T) versus the bass clef (B).

See the example in cpe367_music_gen.py which includes the cpe367_wav.py class.
3a) Deliverable: Upload a WAV file of your synthetic music to Canvas (6)

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)

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

You are strongly encouraged to listen to your synthetic music during the development process. For example, you might first implement a tone burst version of the notes (without the decaying envelope). Listen to that to verify correctness and then incrementally improve your program by making more sophisticated waveshapes (including envelopes and harmonics…) An incremental implementation may help with debugging. It may also help you appreciate the impact on the sound associated with modifications to your mathematical functions ☺