Lab P-5: Synthesis of Sinusoidal Signals-Music Synthesis

2018140088 생명과학부 이도성

In music, the concept of an envelope is very essential for the unique sound. The envelope is the pattern of a soundwave that changes over time. The most frequently used model is the ADSR envelope, which has four parts of attack, decay, sustain, and release. Every time a pulse signal comes in, the pattern starts with an attack and ends with a release.

In this project, the function key2note(X,keynum,dur) was first defined. Using the frequency of note 49, which is 440, as the base, the frequency of the note was defined as f = 440\*2^((keynum-49)/12), making the frequency difference as 2^(1/12) per 1 key. Sample number was given as 11025 and the time variable as tt = 0 : (1/fs) : dur to represent the note duration. The most important real part of the signal function was declared as real(X.\*exp(j\*2\*pi\*f\*tt)), which is the return value of the key2note function.

Notes in the music has a unique parameter of their durations, which is called “BPM”. BPM is an abbreviation of beats per minute, therefor we need to change it into the beats per second since the data is given as pulses, a quarter of beat. In the code, the bpm was given as 120, which means that seconds per pulse = (1/(120/60))/4 = 1/8.

The song “bach\_fugue.mat” is consisted of 3 voices of different length of arrays. Two for loops, which are the adjusted loops of the playscale.m, were used in the code. The first for loop was introduced into the code to save each voice in the array named fugue and the second loop was used to allocate each ADSR envelope to the notes. Since, the zeros(1,N) function needs a positive integral N, while (the sum of the durations)\*(seconds per pulse) is a double type, data type conversion with ceil() function was used to define xx and in order to put the arrays in an array fugue was defined with cell(), which uses all type of data. The two loops allowed each note to have an unique envelope of different length, implementing the bpm into the song.

차트이(가) 표시된 사진

자동 생성된 설명

% ADSR envelope array length setting

Envelope\_length = length(note);

A = round(0.15\*Envelope\_length);

D = round(0.1\*Envelope\_length);

R = round(0.2\*Envelope\_length);

S = Envelope\_length - A - R - D;

% linspace to plot the figure 5

Attack = linspace(0,1,A);

Decay = linspace(1,0.8,D);

Sustain = linspace(0.8,0.7,S);

Release = linspace(0.7,0,R);

% Add the arrays into the envelope

ADSR\_Envelope = [Attack, Decay, Sustain, Release];

ADSR\_Envelope = ADSR\_Envelope(1:length(note));

Figure 1. The plot and the code of the envelope

The Envelope used in the code consists of four parts, Attack, Decay, Sustain, and Release. Linespace() function allowed each part to increase and decrease in a linear manner, as shown in figure 1. Length of each four parts was tweaked to make the song sound more natural referring the figure 5 graph in the project instruction. The length of the envelope has to match the note length, in order to fit into the array. The function round() was used, turning the length of each part into an integral, in order match the indices of the array.

차트이(가) 표시된 사진

자동 생성된 설명

Figure 2. Spectrogram of “bach\_fugue\_short.mat”.

Spectrogram is a visual representation of a soundwave over time, showing the energy of the various waves over time. The 2D spectrogram with the third axis of color allow us to see the overlapping of the notes in the music. In the project, we can use the spectrogram, when the music sounds somewhat discrete, to check the concatenation of the notes and make the music flow continuously.

The tweaks of envelope changed the overlapping soundwaves to fade out and fade in, making the transition of each note more natural. During the tweaking process, the envelope was tweaked to be a discontinuous linear plot by accident, which returned a beeping sound. Changing the bpm, envelope plot, signal strength could change the song, enabling the various instructions of the sheet music to be applied.

Reference

von Coler, Henrik & Lerch, Alexander. (2014). CMMSD: A Data Set for Note-Level Segmentation of Monophonic Music. Proceedings of the AES International Conference.

Himanshu Chaurasiya, Time-Frequency Representations: Spectrogram, Cochleogram and Correlogram, Procedia Computer Science, Volume 167, 2020, Pages 1901-1910, ISSN 1877-0509