Flute Sound Synthesis and Real Time Play

Wen Huang, Zejin Li

University of Rochester Department of Biomedical Engineering

ABSTRACT

Flute sounds can be synthesized with given parameters such as fundamental frequency (pitch), harmonics, amplitudes and phases. In this project, we synthesized three flute octaves with both non-vibrato and vibrato sound effects. The parameters were extracted from the real flute samples notes by using MatLab. With these sound feature parameters, sounds could be synthesized. Comparison between real sound and synthesized sound will be displayed.

C# is being used to create a GUI interface to achieve real time playing. Options like playing demonstration pieces, non-vibrato and vibrato sound effects could be heard by switch between the radius buttons and drag down menu.

1. INTRODUCTION

Note dynamic

Flute can make sounds when the air hits the tube at flute's resonant frequency. There are many effects that composer used to add more emotion in the music. For example, mezzo-forte (*mf)*, forte (*f)* and piano (*p)*. These effects represented the intensity of sound. In music, p refers to 'piano' or soft notes, and f refers to 'forte' or strong notes, mf and mp are mild notes that between loudness p and f. When playing flute, these sound effect could be changed with volume and speed of blow.

Timbre

People tell one instrument from another is based on the characteristic timbre of each instrument. Since timbre is described by the envelope, frequency range, and harmonics of a piece of music, we hypothesized that after these features of a flute sound had been extracted, the parameterized features can be used to develop a synthesizer. Here we made nonvibrato sounds with three loudness levels: p, mf, f, and vibrato sound with mf intensity.

Vibrato Effect

Vibrato could be generated by musician vibrate the abdomen, which causes the air vibrates. This vibrating frequency is much lower than the resonant frequency of flute. Vibratos enhance the audio effect by modulating the spectrum of original signal within a small range, thus, a shift was caused, but was not significant enough for people to detect.

2. METHOD

1) Sound effect on real instrument

Flute can generate the harmonic sounds. The mechanism of sound generation is to letting the air hit the tube and cause tube vibrates. Flute will sound when the frequency reach to its resonant frequency. There are more than one type of flute. The octave ranges of each type varies with its size. As show in figure 1, there are five common type flutes and our synthesis is based on the pitch scale of soprano, the fourth one from top. As you can tell from their name, the smaller flute have higher pitch range. The higher frequency is generated because the smaller item have higher resonant frequency. Besides the piccolo flute, the rests all share three parts, head, body and tail. When playing the flute, lip position and air changes with the pitch. In figure 2, the low, medium and high refers to the low octave, medium octave and high octave. In summary, at low octave, sound is generated when larger amount air and slower speed air is blew into the flute. High octave sound requires faster air speed and less air volume.

Mezzo-forte (*mf*), forte (*f*) and piano (*p*) is the intensity of sound. These sounds effect could be achieved when control the volume. *f* requires stronger vibration, i.e., more air volume flows into the flute. But the shape of your lip should keep the same for the same octave. Vibrato is another effect the musician usually used when performing. Even though the air is breathed by lung, musician should use their abs to control, just like a signers 'stores' and 'use' the air. The abdomen vibrate the air 'stored' in abdominal is the major mechanism of generating a vibrato effect when playing the flute. These are very common effects and are synthesized in our project.

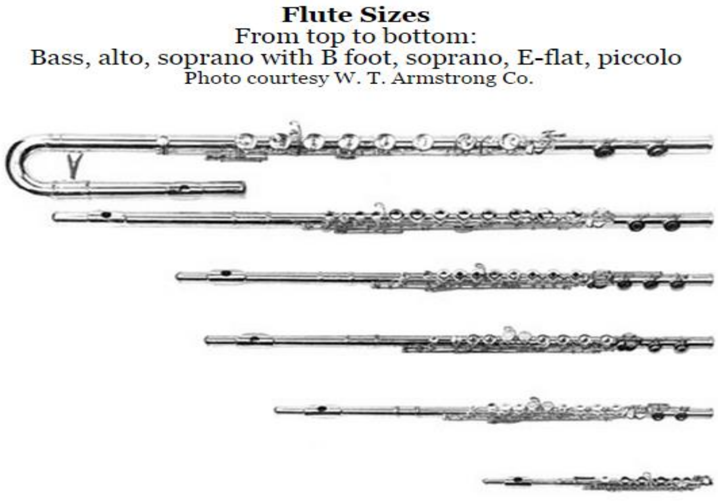
Flute Sizes
From top to bottom:
Bass, alto, soprano with B foot, soprano, E-flat, piccolo
Photo courtesy W. T. Armstrong Co.

Figure 1. Family of Flute. The one we synthesized is soprano.

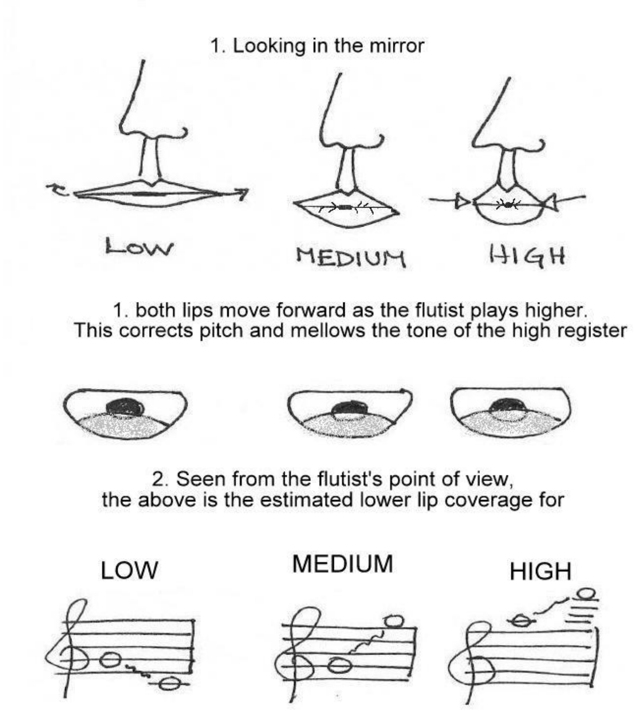
1. Looking in the mirror
LOW
MEDIUM
HIGH
1. both lips move forward as the flutist plays higher.
This corrects pitch and mellows the tone of the high register
2. Seen from the flutist's point of view,
the above is the estimated lower lip coverage for
LOW
MEDIUM
HIGH

Figure 2. Lip coverage for playing different pitches

2) Sound effect analysis and synthesize theory

We used Matlab to do the feature extraction of sample flute sounds, which were obtained from University of Iowa Electronic Music Studio [1]. By looking at the spectrum of each note, the information necessary for synthesis was obtained. Then we generated simulated flute sounds by pooling all frequency components together based on equation2.

$$\mathrm{y(t)} = \sum_{k=1}^{N} \sin(2\pi f_k t + \emptyset_k) + airblow \quad (1)$$

Equation 1 shows how we derived different nonvibrato notes. K refers to the index of harmonics, N is the total number of harmonics that we collected. Here we made N = 5. f_k, $\emptyset_k$ are the frequency and phase of the K_{th} harmonic. To make the sound more natural, like a real flute sound, we also added noise that simulated airblow vibration to the signal.

In our project, we produced vibrato sounds based on Frequency Modulation (FM). Specifically, FM modulated the spectrum by pooling the change of frequency components and phase information together. Mathematical model is described in Eqn.2,

$$\mathrm{y(t)} = A_1(t)\cos[(\omega_1 + A_2(t)\cos(\omega_2 t))t] \quad (2)$$

where $A_1$and ω_1 refer to the amplitude and frequency of original signal, and A_2 and ω_2 refer to that of the modulation signal.

3) MatLab analysis processes

To analyze the vibrato sounds, we compared the synthesized vibrato sounds with the nonvibrato sounds with the same fundamental frequency by looking at their spectrogram respectively. The signal was windowed by 1024-point frame with 50% overlap. To make the spectrogram more clear, we also applied four-time zero-padding.

4) C#

C# was used in this project. The GUI interface was created and able to play note while a key pressed. Four radio bottoms were used to change the sound intensity and sound effects, i.e., mf, pp, f or vibrato. By checking the 'vibrato' bottom, users should allowed to play the flute sound with vibrato effect. Parameter for this fancy sound effect will be added shortly. The demonstration on a whole piece of music is being written in Matlab. The drag down menu has two options underneath, play and stop. The demonstration piece was synthesized in MatLab and User should able to hear four sounds effects sequentially.

There were three octaves and each had a label on the left. Each note was signed to a key on keyboard and were shown in figure 3. When running the program, you will see that name of the note you are playing.

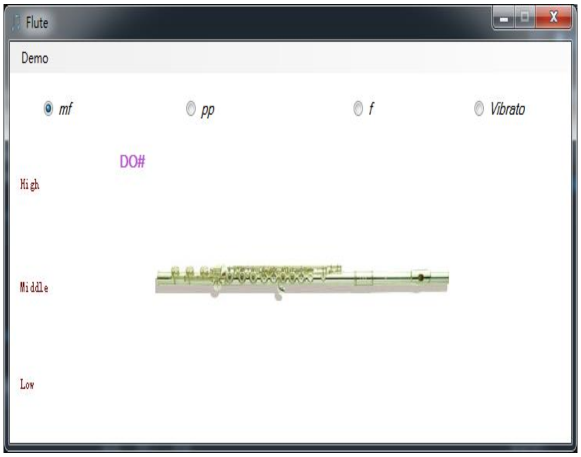

Figure 3. Demo menu allows play and stop the demo piece. The name of the note being played will only be displayed when pressing the key that controls it. Four radius buttons on the top allow user to change the form of music that they playing.

3. RESULTS

Extracted parameters are shown in Table1.

	Pitch (Hz)	Harmonics (Hz)	Amplitude (no unit)	Phase (degree)
C	526.205	1051.5, 1579.7 2103.2, 2633.1	1522.4, 611.3, 412.6 195.7, 66.1	-110.194, 122.091,-142, -160.32, -17.5336
C#	558.039	1115.9, 1674.5 2232.3, 2786.4	1086.9, 438.5, 243.4, 69.7, 54.7	-101.24, -124.59, -39.76 175.142, 149.439
D	589.977	1176.9, 1765.6 2360.1, 2942.7	809.6,646.39, 177.55 104.92, 56.93	-86.14, 60.59, 49.29 50.79, -65.445
D#	624.512	1248.8,1873.9 2494.2	1108.1, 467.8, 219.7 85.6	155.295, -141.48, 169.528 23.577
E	662.061	1324.3, 1986.6	1358.2, 404.9, 345.9	-40.598, 60.433, -5.727
F	700.045	1399.5, 2099	2137.4, 504.1 392.4	16.463, 144.473,54.86
F#	747.028	1493.4, 2239	1876.7, 482, 216	134.864, 19.184, 75.995
G	786.977	1573.4, 2360.2	1649.2, 228.8, 184.2	111.556, -561.7, -70.358
G#	833.219	1666.3, 2489.6	1328.3, 501.3, 38.6	131.78, 91.116, 120.36
A	886.658	1326.6, 1772.5	2150.2, 171.1, 35.6	-80.525, -163.703, 36.81
A#	943.029	1886.2, 2829.1	2893.2, 230.8, 90	170.928, -66.58, -123
B	998.325	1995.5, 2993.7	1716.6, 225.9, 120.1	-20.607, 29.063, -41.767

Table 1. Extracted parameters of the middle octave. Here only shows the information of mf intensity notes as an example. These extracted parameters were then used for generating synthesized sounds in both Matlab and C#.

We produced the simulated flute sounds with the statistics provided above, the synthesized sounds were generated by adding different frequency components together directly. The waveforms and spectrums of eight notes in middle octave are shown below:

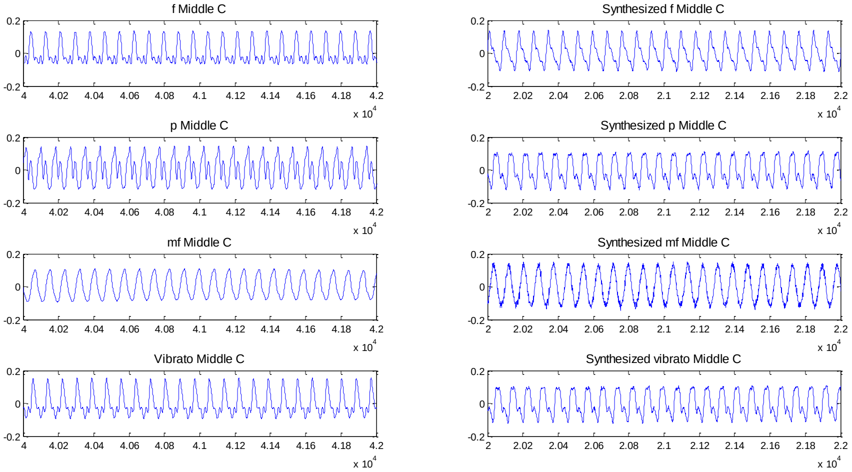

Figure 4.Waveform of notes in the middle octave. Figures on the left show the waveform of flute sample note, right traces represent that of what were synthesized.

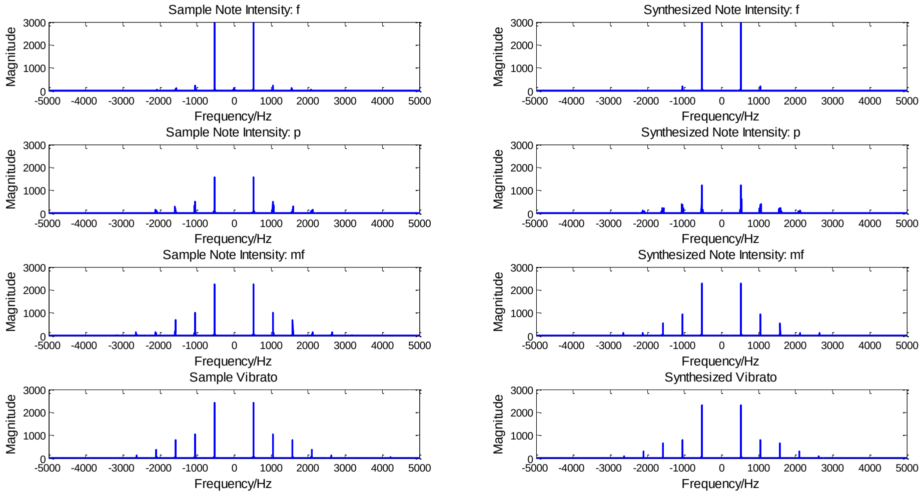

Figure5. Spectrum of the notes in middle octave. Figures on the left show the spectrum of flute sample note, right traces represent that of what were synthesized.

The spectrum shows that the stronger notes (i.e. f notes) are not only louder than the relative weaker notes (i.e. p notes), but also more convergent in frequency domain. This means that more energy is focused on fundamental frequency for strong sounds.

Then we compared the vibrato sound with nonvibrato sound by looking at the spectrogram in figure.6 and 7. Figure.6 and 7 show that vibrato sounds have the same fundamental frequency and harmonics of the corresponding nonvibrato sound, but there's a small fluctuation among the harmonic frequencies, which caused the signal sounds like vibrating. However, basically, people can still perceive which note it refers to.

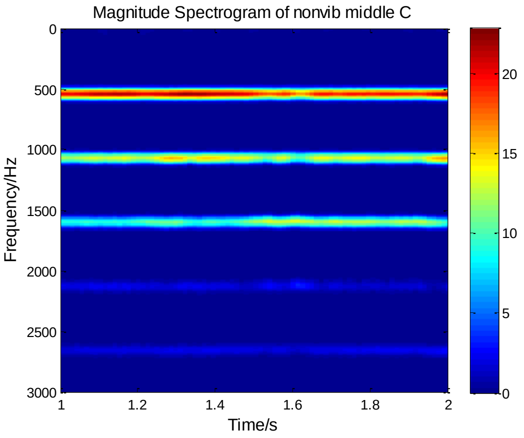

Figure 6 Spectrogram of nonvibrato middle C

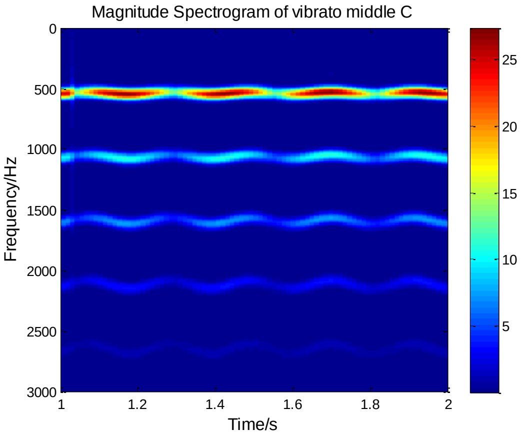

Figure7. Spectrogram of vibrato middle C

4. IMPLEMENTATION

Matlab GUI Keyboard

After we derived all needed single notes, we continued to extend the project by generating an electronic keyboard using GUI interface. Each key corresponds to a particular note. Besides, the GUI interface also allows people to play a piece of demo music and a scale.

The keyboard is shown in Figure8

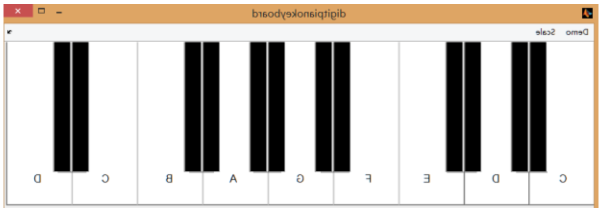

Figure 8. GUI interface in Matlab. Single note, demo and scale will be played by cliking the keyboard or the Demo and Scale button. However, our Matlab simulated keyboard can only play the sounds with fixed time length. It cannot detect the hit and release time point of the key and achieve real-time playing. Luckily, the C# keyboard makes this possible.

C# interface

Each notes were signed to different keys on computer keyboard. The following figure will give you a brief idea on how they were signed. The light pink represent the highest octave, violet represent middle octave and red represent the low octave.

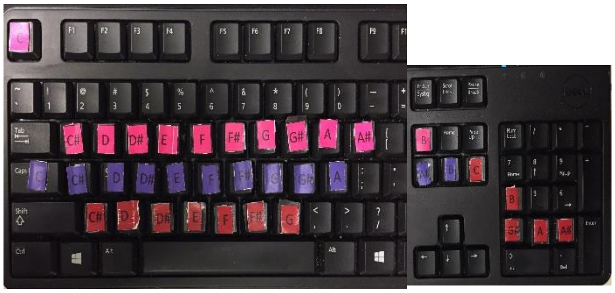

Figure 9 Keyboard match with flute notes as shown above.

5. DISCUSSION

The synthesized sounds effects were very close to the real instrument played sounds. However, the sounds synthesized in our project were constructed simply by adding different frequency components together.

By looking at the spectrum, we can see that the synthesized signals have the same harmonics and amplitude, but by analyzing the waveforms, we noticed that the shape and envelop between each pair of sample and synthesized note still has minor difference. Such difference may be the main cause of the timbre distortion, i.e. the generated audio signals sound like some instruments other than flute.

To make the sounds more closed to real flute sounds, many other factors are needed to take into consideration, including specific airflow vibration, and the extracted parameters need to contain more detailed information. Additionally, the sample we used in this project itself may have harmonic distortion. Since probably a proportion of the harmonic information was lost when transferring the real analog flute sound into the digital sound that we used.

And in future workload, we are planning to add more instruments sounds synthesis. It is also very interesting to create some artificial sounds. As for the C# part, Multi-sound can be combined and played at same time. And the multi key control goal can be pressed to hear two or more tones.

6. CONCLUSION

The project was inspired by MIDI controlled flute synthesis using OMAP board. Later we found the real time implement with the board could be a lot of difficulties to accomplish. Then we switched our thoughts to synthesizing with Matlab and playing the sounds with the C# GUI interface. Thus, the flute sound can be played as the key pressed at same time. The synthesized sound was not exactly the same as the real flute sound. There were many ways to improve the performances. The flute sound recording may causes sound information loss and the parameters we extracted were not accurate at all. More harmonic information can be added in future.

7. REFERENCES

[1] University of Iowa Electronic Music Studio http://theremin.music.uiowa.edu/MISviolin.html

[2] S.Rossignol, P.Depalle, J.Soumagne et al, vibrato_detection estimation extraction modificatio