

Mini-project: MATLAB-based Piano Sound Synthesis

1st Qiu Kunyuan

EEE, Southern University of Science and Technology

Shenzhen, PRC

11913019@mail.sustech.edu.cn

Abstract—In this paper, the advantages and disadvantages of various sound simulation techniques have been analyzed, and then a method for synthesizing piano sounds using ADSR model and string vibration model has been implemented based on the previous work [1] [2].

This method simulates the physical properties of piano string vibration [3] [4] and resonance box by MATLAB, and then generates the tones with overtones of various orders by using the string vibration law, and then modifies the string vibration tones according to the amplitude envelope and spectral envelope of the piano sound collected, and generates the syllables with specific duration and timbre.

In this work, the proposed synthesis system is experimentally validated, and it is shown that the synthesized notes are harmonious, and the timbre performance is excellent, and the generated timbre characteristics can be flexibly changed, and the synthesized sound is close to various real instruments.

Index Terms—sound simulation, ADSR model, string vibration, MATLAB

I. INTRODUCTION: MUSIC THEORY EXPRESSED IN PHYSICS

Music is an art made up of sound, and sound is a sound wave that can be perceived by the human ear. Sound signals with a specific frequency spectrum and amplitude envelope can stimulate the human auditory organs and bring aesthetic experiences to humans, and music is formed when humans generate such sound signals on their subjective initiative.

A. Octaves and Frequency Response of Human Ears

The primary acoustic characteristics that determine the human auditory experience are pitch and intensity. Pitch is the central frequency of the acoustic signal, and intensity is the amplitude of the acoustic signal.

Such rule of human acoustic response is firstly concluded and expressed by the A. Bach and other classic musicians like L. Beethoven and Hayton. It can be said that all the European classic music are based on the mathematical conclusions and empirical formulas with no exaggeration.

Since the linear distance from the top of the cochlear basilar membrane to the maximum displacement is proportional to the logarithm of the input frequency, and the output signal of the human auditory nerve is proportional to the logarithm of the sound pressure, the frequency response of the human auditory system follows a double logarithmic coordinate:

$$p = \log\left(\frac{f}{f_0}\right), B = 10 \log\left(\frac{W}{W_0}\right) \quad (1)$$

For every doubling or doubling of frequency, the pitch rises by one unit, and this unit is called an octave. The average person can hear sounds from 20 to 20,000 Hz, and using 440 Hz, the most sensitive frequency in the human ear, as a reference, the human ear can hear about

$$\left\lceil \frac{\log\left(\frac{f_{min}}{440}\right)}{\log(2)} \right\rceil = -4, \left\lfloor \frac{\log\left(\frac{f_{max}}{440}\right)}{\log(2)} \right\rfloor = 5 \quad (2)$$

octaves below and above.

B. Temperament and Key Scale

Within an octave, the more sensitive sound frequencies of the human ear are concentrated in eight frequency points, and the ratio of two of these frequency points is a specific value of one or two squares.

Let q is the minimal frequency ratio between one note and the other, then an octave can be divided into 8 frequency points including 2 minor gaps and 5 major gaps that the frequency ratio of which is q^2 . Therefore, the ratio satisfies

$$q^2 \cdot (q^2)^5 = 2 \Rightarrow q = 2^{1/12} \quad (3)$$

Using this ratio and the gap relation, it is convenient to determine the frequency of arbitrary note inside an octave with the base frequency f_0 given.

$$f[k] = f_0 q^k = f_0 2^{\frac{k}{12}} \quad (4)$$

In actual musical composing, it is a convention that the C1 note at 523Hz is used to determine the base frequencies,

$$f_0[k] = f_{C1} 2^{\frac{k}{12}} \quad (5)$$

and the gap sequence

$$k = [0, 2, 4, 5, 7, 9, 11] \quad (6)$$

is conventionally referred to as the major scale. Therefore, an arbitrary major scale can be expressed as the formula below by substituting (1) and (5) into (4):

$$f[k] = f_{C1} 2^{(k_0+k)/12+p} \quad (7)$$

Where

$$p \in [-4, 5], \{k_0, k\} \subset \{0, 2, 4, 5, 7, 9, 11\} \quad (8)$$

C. Tone-color and Sounding Model of Piano String

The subjective features of a musical tone other than its pitch, intensity and length is concluded as the tone-color, which is mostly determined by the frequency spectrum and the amplitude envelope of the sounding instrument.

Most mechanical oscillating bodies in the real world produces fading non-harmonic sound wave after a single excitation. For the sounding strings in the piano, the oscillation models under idealistic conditions can be theoretically derived by solving the 1-dimensional Helmholtz equation.

$$\begin{cases} u_{tt} + \mu u_t = v^2 u_{xx}, \\ u|_{x=0} = u|_{x=l} = 0, \\ u|_{t=0} = 0, \\ u_t|_{t=0} = \text{rect}\left(\frac{x-x_0}{2d}\right) v_0 \cos\left(\pi \frac{x-x_0}{2d}\right) \end{cases} \quad (9)$$

Solving this equation by variable separation method gives that

$$\begin{cases} u(x, t) = A \exp(-\mu/2t) \sum_{n=1}^{\infty} \frac{1}{n} \frac{1}{1-4d^2n^2/l^2} B(n, x, t), \\ B(n, x, t) = C_n \sin(n\pi x_0/l) \sin(n\omega_0 t) \\ \quad + D_n \cos(n\pi x_0/l) \cos(n\omega_0 t) \end{cases} \quad (10)$$

The coefficients C_n, D_n in this solution is very hard to express analytically and requires experimental testing or manually adjusting. These coefficients are amplitudes of the harmonic components, and the combination of the coefficients partially controls the tone-color of the piano.

D. ADSR Envelope Model of the Resonance Chamber

The abundant tone-color and the time domain features of piano sound is determined not only by the excite signal from the oscillating strings, but also by the filter formed by the resonating chamber:

$$y(t) = h_{\text{chamber}}(t) * \sum_{n=1}^N e_n(t) \quad (11)$$

The effect of the resonating chamber normally results in echoing of the syllables that are simultaneously stroke and the dragging after the key is released. And the large volume of the typically wooden chamber attenuates the non-resonant frequency components produced by the string, while enabling the resonant frequency pass through. Thus, the resonating chamber can be modeled as a FIR filter with non-zero delay and more than 1 pass bands.

Taking into account the damping of the piano string and the impulse response of the resonant cavity, the piano tone also has a changing amplitude envelope in the time domain. Such envelope is characterized by a large initial value, a slow decay of the amplitude, and then a rapid decay to zero.

In the field of electronic musical instruments, the ADSR model, an empirical polyline model, can be used to provide a good approximation of the amplitude envelope of various musical tones.

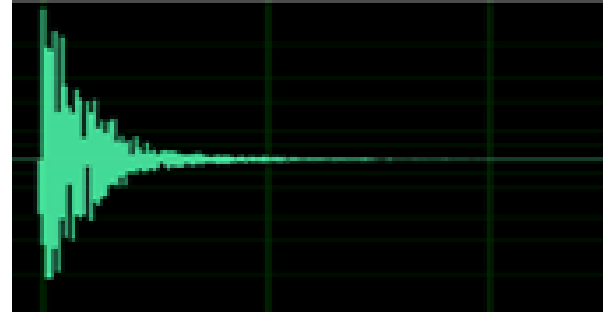


Fig. 1: Real envelope of Piano Sound

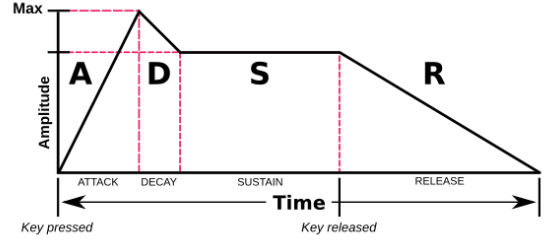


Fig. 2: The Main Conception of ADSR Envelope

II. SYNTHESIZER DESIGN

With the analysis above on the sounding progress of piano, there is no theoretical difficulty in forming the architecture of the synthesizer program. The simulator is divided into the tone generator, envelope modulator and echoing filter, since the sounding process in a real piano is composed by striking the string and echoing in the resonance chamber.

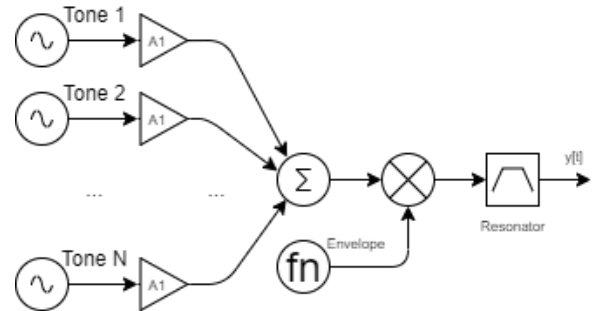


Fig. 3: Architecture of the Synthesizer

A. Tone Generation and Synthesis

In the string oscillation equation (10), the base frequency ω_0 are injected from the note parameters in the simplified stave as the equation (7) and (8).

Let the

$$l = [0, 2, 4, 5, 7, 9, 11] \quad (12)$$

denotes the relative pitch sequence of arbitrary major scale, this list injects the pitch name

$$k = [1 : 7] \quad (13)$$

to the relative interval. For a scale consisting a tonic keynote⁴
 $l[k_0]$, a conventional pitch name sequence $l[k]$ and an octave⁵
number p , the frequency of each note is obtained by deduction
from the (7) and (8):

$$f[k] = f_{C1} 2^{(l[k_0] + l[k] + \Delta)/12 + p} \quad (14)$$

Where

$$p \in [-4, 5], \Delta = \{-1, 0, 1\} \quad (15)$$

and the

$$f_{C1} = 523.2511\text{Hz}, f_{A1} = 440\text{Hz} \quad (16)$$

is the absolute baseline that determines the absolute pitch or frequency.

Therefore, the function `tone2freq(tone, noctave, rising)` and `gen_wave(tone, scale, noctave, rising, rhythm, fs)` that evaluates the fundamental tone of a given note can be programmed easily by directly translating the formulas above to MATLAB code:

a) `tone2freq.m`:

```
1 function freq=tone2freq(tone,noctave,rising,base,  
   minority)  
2 % 计算任意八度音程的频率,基音f0.  
3 % 语法: freq=tone2freq(tone,noctave,rising,  
   base,minority)  
4 % tone: 输入数字音符,数值范围1到7  
5 % noctave: 高或低八度的数量,数值范围整数. 0表示  
   中音,正数表示高 noctave个八度,负数为低  
   noctave个八度  
6 % rising: 升或降调. 1为升,-1为降,0无升降调  
7 % base: 基音音调,默认为C大调音阶  
8 % minority: 大小调切换,0为大调,非0为小调  
9 % freq为输出的频率  
10 if tone<1||tone>7  
11     error('输入的音符 %d 不在表示范围 [1,7] 内',tone);  
12 end  
13 tbl1=ToScale(base,minority);  
14 freq=440*2.^((tbl1(tone)+rising)/12+noctave);  
15 end
```

b) `ToScale.m`:

```
1 function table=ToScale(Base,Minority)  
2 % 将调号转为音阶序,大调音阶起始音符为Base,小调  
   音阶起始音符为Base降低四度  
3 % 语法: table=ToScale(Base,Minority)  
4 % Base: 起始音符  
5 % Minority: 大小调切换,0为大调,非0为小调  
6 scale1=cumsum([2,1,2,2,1,2,2]);  
7 scale2=cumsum([2,2,1,2,2,2,1]);  
8 if all(isa(Base,'char'))  
9     Base=Base-'A';  
10 end  
11 minor=[4,3,3,4,4,3,4];  
12 if Base==0  
13     table=[0,scale2]-(Minority~=0)*4;
```

```
else  
    table=scale1(Base)+[0,scale2]-(Minority  
~=0)*minor(Base);  
end  
end
```

B. Harmonic Overtone Synthesizing

With the fundamental tone(frequency) deduced from pitch name in the section (A), the sound wave produced by the oscillation of the piano string can be determined in the frequency domain. The solution of 1-dimension Helmholtz equation states that the vibration waveform of a string is a periodic function expressed in Fourier series as (10):

$$u(x, t)[k] = A \exp(-\mu t/2) \sum_{n=1}^{\infty} P_n (B_n \exp(jn2\pi f[k]) + C_n \exp(-jn2\pi f[k])) \quad (17)$$

Where the $f[k]$ is the fundamental frequency evaluated above from the pitch name, and the A, μ, B_n, C_n are the empirical amplitude coefficients and damping coefficients measured by experiment. The common coefficient

$$P_n = \frac{1}{n} \frac{1}{1 - 4d^2 n^2 / l^2} \quad (18)$$

is the common part of the sine and cosine terms in the solution, which indicates the amplitude decreasing along with the increase of mode order.

The far-field radiation pattern of the string is approximately cylindrical symmetric, similar to that of a dipole. This effect is neither dispersive nor time-varying, thus the effect of far-field radiation can be aggregated with the effect of the resonance chamber and treated as a BPF after.

The amplitude coefficients B_n, C_n has already been measured experimentally in the existing research [2] and [1]. With the help of these two existing research, the highest harmonic order is chosen empirically as the formula below:

$$N_{max} = \left\lceil \frac{f_{A1} \times N_{A1}}{f[k]} \right\rceil \quad (19)$$

where the N_{A1} is the highest order that has significant effect(>30dB) to the synthesized tone color:

$$\frac{B^2[N_{A1}] + C^2[N_{A1}]}{\sum (B_n^2 + C_n^2)} \Big|_{440\text{Hz}} > 30\text{dB} \quad (20)$$

All the deductions above leads to the waveform generation code below:

```
1 fc1=tone2freq(note(1),note(2),note(3),key,0);  
2 fc1_list=fc1*(1:19);  
3 amp=[987.8,368.6,620.2,483.9,156.7,...  
4       83.62,120.1,70.73,5.348,24.41,...  
5       27.35,21.3,10.31,6.477,15.91,...  
6       3.495,2.546,0.4751,0.8858]/1000;  
7 wfm1=cell2mat(arrayfun(@(a,x) a*sin(2*pi*x*Tv)  
   ,amp,fc1_list,'UniformOutput',false).'));  
8
```

C. Resonance Chamber and ADSR Enveloping

The frequency generator mentioned above generates a continuous signal that is almost constant in amplitude. Considering the damped vibration of the strings, the quadratic inverse attenuation of the sound waves in the air and the filtering of the resonance box together, the musical signal from the piano should have an ADSR envelope in the time domain and several concentrated resonance bands in the frequency domain.

The research [2] [1] both give out the experimentally measured envelopes in terms of piecewise function, and [1] also measured the resonance bands of the resonant box of a piano. The enveloping and filtering functions can be programmed with referencing to the two achievements above therefore.

```
1 % Enveloping and Filtering
2 wfm1=sum(wfm1,1).*GenADSRenv(wfm1
3 , [0.008,0.09,0.11,0.25,0.67],...
4 [0.88,0.88,0.56,0.50,0]);
5 [b_b1,a_b1]=butter(6,2*[1300,1900]/Fs);
6 [b_b2,a_b2]=butter(6,2*[3600,5000]/Fs);
7 [b_b3,a_b3]=butter(6,2*[7800,8200]/Fs);
8 wfm2=1.1*wfm1+0.4*filter(b_b1,a_b1,wfm1)+0.35*
9 filter(b_b2,a_b2,wfm1)+0.25*filter(b_b3,a_b3,
10 wfm1);
11 wfm2=wfm2/(max(wfm2));
```

The expression of piecewise linear ADSR envelopes can be easily obtained by concatenating line sections in terms of two-point equation.

$$y_{ADSR}[n] = \sum_0^{N_C} \left(y_C[k] + nF_S \frac{y_C[k+1] - y_C[k]}{x_C[k+1] - x_C[k]} \right) \times ((u(n - x[k+1]F_S) - u(n - x[k]F_S))) \quad (21)$$

where the vector

$$\begin{pmatrix} x_C[n] \\ y_C[n] \end{pmatrix} = \begin{pmatrix} y_C[0], \dots, y_C[N_C] \\ x_C[0], \dots, x_C[N_C] \end{pmatrix} \quad (22)$$

are the time and amplitude of the control points $x_C[n] = (x_1, x_2, \dots, x_n)$ and $y_C[n] = (y_1, y_2, \dots, y_n)$, and F_S is the sampling frequency corresponding to the input soundwave.

D. Score File Reading

In order to decouple the score data from the player and enhance the user-friendliness of the program, a separate text file is used to record the key signature, the beat, the sample rate and the score. By separating the score from the player, it is also convenient for users to compose their own scores for playing custom tracks, as well as enabling future extensions such as multitracking.

The capability of reading the score is implemented through `fscanf()` function, which is a MATLAB built-in function.

```
1 function PlayMusicFile(str)
2     if ~exist(str,'file')
3         error("Inexist File Path:%s",str);
4     end
```

```
script=fopen(str,"rt");
key_mat=fscanf(script,"%c %d Fs=%d\n",[1,3]);
scr=fscanf(script,"%d %d %d %f\n",[4,inf]);
fclose(script);

Fs=key_mat(3);
T0=60/key_mat(2); %节拍
key=char(key_mat(1));

scr_note=scr(1,:);
scr_noct=scr(2,:);
scr_risg=scr(3,:);
scr_dur=scr(4,:);
len=1:length(scr_note);
Y=[];

for i=len
    note=[scr_note(i),scr_noct(i),scr_risg(i),
scr_dur(i)*T0];
    wfm2=SoundNote(key,note,Fs);
    Y=[Y,wfm2];
end
sound(Y,Fs);
end
```

The format of an example score file is shown below. The first row of the score contains the base key, rhythm speed and sampling frequency in Hertz, and the succeeding rows consist of the 4 elementary data of a note, respectively the tone, number of octave, quarter rising and sustaining time.

```
1 D 76 Fs=48000
2 6 0 0 0.5
3 7 0 0 0.5
4 1 1 0 1.5
5 7 0 0 0.5
```

III. EXTENSION ON SYMPHONY

A symphony is composed by a couple of different soundtracks playing simultaneously, and the harmonic mixing of such individual waveforms in a resonating chamber creates grand music experience with sacred feeling. For MATLAB based synthesizer, such symphony can be easily realized by using multiple synthesizers configured to emulate different musical instruments, and then use a mixer with the frequency response of the Golden Hall or any other music halls.

With this technique, the synthesizer mentioned above can be used to play the classic Germany and Austrian symphonies composed in the 19th to early 20th centuries.

REFERENCES

- [1] 曹莎莎, “一种钢琴乐音仿真模型的研究,” 2017.
- [2] 程美芳, “钢琴音色识别与电子合成系统的设计与实现,” 2014.
- [3] 刘金寿; 刘远, “钢琴弦列的声学特性与理想弦振动模型,” 演艺设备与科技, no. 04, pp. 62–63, 2007.
- [4] 杨拯, “钢琴弦的非理想性与钢琴调律,” 乐器, no. 10, pp. 22–25, 2016.