

MATLAB 编程实现电子乐合成器

仇琨元 11913019

November 17, 2022

TOC

MATLAB 编程实现
电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

1 Introduction

2 合成器总体架构

3 音阶发生器

- 人耳频率响应
- 十二平均律
- 代码实现

4 连续波合成器

- 钢琴弦振动方程
- 代码实现

5 包络发生器

- ADSR 包络
- 代码实现

6 乐谱文件读取

- 代码实现

电子乐简介

MATLAB 编程实现
电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

- 音乐是组织特定时域波形的声音信号的艺术形式。
- DSP 技术可以依据人的主观意愿任意生成声音信号。

电子音乐是一种使用数字信号处理技术合成音乐声音信号的音乐艺术。

```
script.m x script_tkzc.txt PlayScript.m x GenADSRenv.m x tone2freqA.m
clear;close;clc;

script=fopen("script_tkzc.txt","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));

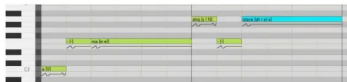
scr_note=scr(1,:);
scr_noct=scr(2,:)-1;
scr_risg=scr(3,:);
scr_dur=scr(4,:);
len=1:length(scr_note);
Y=[];

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
```

(a) 合成器的一部分子程序代码



Further, you can enter a hyphen for a single note to extend the pronunciation of the previous note.



(b) VOCALOID（一种人声合成器）的编辑界面

Figure: 电子乐合成器

项目简介

MATLAB 编程实现
电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

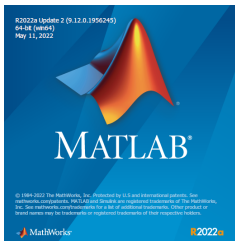
代码实现

乐谱文件读取

代码实现

本项目使用 MATLAB 编程环境实现简单的包络合成器合成电子音乐。该合成器具有以下能力：

- 从文件读取简单的乐谱，包含音符、调号和节拍信息。
- 基于乐器振动方程和实测频谱数据生成连续单音。
- 基于实际乐器共振腔体频率响应的频域滤波处理（音色生成器）。
- 可任意调节参数的 ADSR 包络发生器。



合成器结构框图

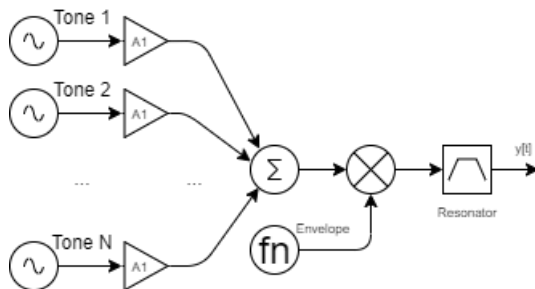


Figure: 合成器总体架构

合成器首先通过傅里叶逆变换生成具有乐器 STFT（短时傅里叶变换）频谱的连续波，然后用音色滤波器进行进一步的频域处理，从而获得具有**一部分音色**的连续波乐音信号。

将连续波信号乘上一定的 ADSR 包络线，即可产生具有完整音色的时域波形。

合成器结构框图

MATLAB 编程实现
电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

为了方便用户使用，合成器只需要读取特定格式的乐谱文件即可演奏音乐，乐谱文件不应固化到前述的数字信号处理模块内部。
由此可见，合成器应当具有如下组成部分：

- 乐谱读取
- 调号、音符到频率数值计算
- 连续波合成器
- 音色滤波器
- 包络发生器

人耳频率响应与音阶划分

MATLAB 编程实现
电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

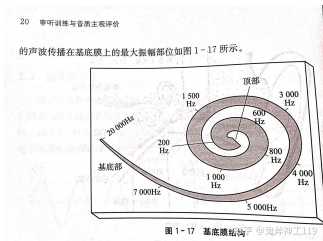
代码实现

乐谱文件读取

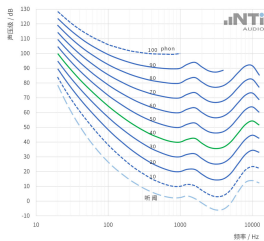
代码实现

人类听觉系统通过耳蜗对声音信号进行频谱分解。耳蜗基底膜受到声音信号刺激后会发生响应，产生振动。

基底膜的宽度和厚度随着耳蜗螺距逐渐变化，研究表明基底膜顶部到谐振点的直线距离与输入频率的对数成正比，因此人耳在频谱上有近似对数标度的响应。



(a) 耳蜗结构简图



(b) 人类听觉等响度曲线

Figure: 人类听觉系统简图

人耳听觉音域

MATLAB 编程实
现电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

长期的归纳总结和前述**对数标度律**均指出，绝对频率为整数倍的纯音（复指数信号 $Ae^{j\omega t}$ ）在听觉感知中表现为线性的整数音高差，而两个音的绝对频率比越接近 1: 1，混频后的和声就越和谐。这种整数音高差的最小单位为一个倍频程 $\Delta f = f_0$ ，称为一个**八度**。以人耳最敏感的中音 $A_0 = 440\text{Hz}$ 作为标定原点，人耳听觉范围大约能覆盖

$$\left\lceil \frac{\log(\frac{f_{min}}{440})}{\log(2)} \right\rceil = -4, \left\lfloor \frac{\log(\frac{f_{max}}{440})}{\log(2)} \right\rfloor = 5 \quad (1)$$

共计 9 个八度。

十二平均律

MATLAB 编程实现
电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

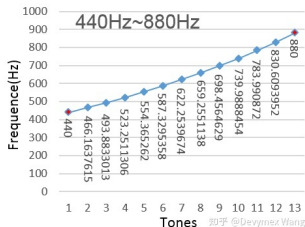
ADSR 包络

代码实现

乐谱文件读取

代码实现

在一个八度之内选取人耳最敏感的 7 个相对音高频点，这些频点的频率均近似处于一个首项等于公比的等比数列上。



一般来说这 7 个谐频点含有 2 个倍频程为 q 的频点和 5 个倍频程为 q^2 的频点。由此可得一个八度内 7 个相对音高频点的分布律

$$q^2 \cdot (q^2)^5 = 2 \Rightarrow q = 2^{\frac{1}{12}} \approx 1.06 \quad (2)$$

十二平均律

MATLAB 编程实
现电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

因此，一个八度之内的 12 个半音频点即可用上述的公比计算：

$$f[k] = f_0 q^k = f_0 2^{\frac{k}{12}} \quad (3)$$

在文艺复兴时期以来的实际音乐创作中，一般以 $C_1 = 523\text{Hz}$ 标定八度音阶起始点，因而包含 2 个半音和 5 个全音的音阶序列

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

被称为大调音阶。将前述的八度音程规约 (1) 代入 (3)，即可得到用十二平均律系统表示的任意音高记号到频率的转换公式：

$$f[k] = f_{C1} 2^{(k_0+k)/12+p}, p \in [-4, 5], \{k_0, k\} \subset \{0, 2, 4, 5, 7, 9, 11\} \quad (5)$$

音符到音阶转换代码

MATLAB 编程实现
电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

```
1 function freq=tone2freq(tone,noctave,rising,base,minority)
2     if tone<1||tone>7
3         error("输入的音符 %d 不在表示范围 [1,7] 内",tone);
4     end
5     tbl1=ToScale(base,minority);
6     freq=440*2.^((tbl1(tone)+rising)/12+noctave);
7 end
8 function table=ToScale(Base,Minority)
9     scale1=cumsum([2,1,2,2,1,2,2]);
10    scale2=cumsum([2,2,1,2,2,2,1]);
11    if all(isa(Base,'char'))
12        Base=Base-'A';
13    end
14    minor=[4,3,3,4,4,3,4];
15    if Base==0
16        table=[0,scale2]-(Minority~=0)*4;
17    else
18        table=scale1(Base)+[0,scale2]-(Minority~=0)*minor(Base)
19        ;
20    end
end
```

钢琴弦振动方程

MATLAB 编程实现
电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

合成器依据钢琴弦振动方程和实测频谱数据产生具有钢琴音色的单音。钢琴弦的振动可以用一维弦振动模型描述：

$$\begin{cases} u_{tt} + \mu u_t &= v^2 u_{xx}, \\ u|_{x=0} = u|_{x=l} &= 0, \\ u|_{t=0} &= 0, \\ u_t|_{t=0} &= \text{rect}\left(\frac{x-x_0}{2d}\right) v_0 \cos\left(\pi \frac{x-x_0}{2d}\right) \end{cases} \quad (6)$$

时域波形遵循该亥姆霍兹方程的解：

$$\begin{cases} u(x, t) &= A \exp(-\mu/2t) \sum_{n=1}^{\infty} \frac{1}{n} \frac{1}{1-4d^2n^2/l^2} B(n, x, t), \\ B(n, x, t) &= C_n \sin(n\pi x_0/l) \sin(n\omega_0 t) \\ &+ D_n \cos(n\pi x_0/l) \cos(n\omega_0 t) \end{cases} \quad (7)$$

钢琴弦振动方程

MATLAB 编程实现
电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

由于解析解的两个傅里叶系数 C_n, D_n 难以求解，因此用实际测量的频谱数据代替根据琴弦物理参数解析求解。根据现有研究，利用以下经验公式确定谐波最高阶数：

$$N_{max} = \left\lceil \frac{f_{A1} \times N_{A1}}{f[k]} \right\rceil \quad (8)$$

其中 N_{A1} 是对最终合成音色有可辨识影响的最高谐波阶数：

$$\frac{B^2[N_{A1}] + C^2[N_{A1}]}{\sum (B_n^2 + C_n^2)} \Big|_{440\text{Hz}} > 30\text{dB} \quad (9)$$

代码实现

MATLAB 编程实现
电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

```
1 function wfm2=SoundNote(key,note,Fs)
2     duration=note(4);
3     Tv=linspace(0,duration,duration*Fs);
4     if note(1)==0
5         wfm2=zeros(1,round(duration*Fs));
6     else
7         fc1=tone2freq(note(1),note(2),note(3),key,0);
8         fc1_list=fc1*(1:19);
9         amp=[987.8,368.6,620.2,483.9,156.7,83.62,120.1, ...
10             70.73,5.348,24.41,27.35,21.3,10.31,6.477, ...
11             15.91,3.495,2.546,0.4751,0.8858]/1000;
12         wfm1=cell2mat(arrayfun(@(a,x) a*sin(2*pi*x*Tv),amp,
13                                fc1_list,'UniformOutput',false).');
14         wfm1=sum(wfm1,1).*GenADSRenv(wfm1
15                                     ,[0.008,0.09,0.11,0.25,0.67],[0.88,0.88,0.56,0.50,0])
16                                     ;
17         [b_b1,a_b1]=butter(6,2*[1300,1900]/Fs);
18         [b_b2,a_b2]=butter(6,2*[3600,5000]/Fs);
19         [b_b3,a_b3]=butter(6,2*[7800,8200]/Fs);
20         wfm2=1.1*wfm1+0.4*filter(b_b1,a_b1,wfm1)+0.35*filter(
21             b_b2,a_b2,wfm1)+0.25*filter(b_b3,a_b3,wfm1);
22         wfm2=wfm2/(max(wfm2));
23     end
24 end
```

ADSR 包络

MATLAB 编程实现
电子乐合成器
仇琨元 11913019

Introduction
合成器总体架构

音阶发生器
人耳频率响应
十二平均律
代码实现

连续波合成器
钢琴弦振动方程
代码实现

包络发生器
ADSR 包络
代码实现

乐谱文件读取
代码实现

声音的时间包络 (Time Envelope) 也是音色的一个重要组成部分。相同的频谱, 在不同的时间包络下, 会让我们听到截然不同的声音。电子乐领域一般用 ADSR 模型 (4) 构造包络。

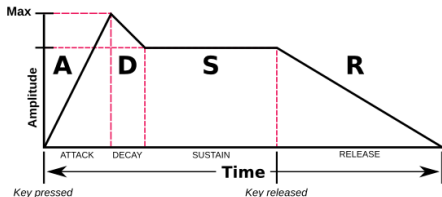


Figure: ADSR 的概念

声音从无到有, 从有到无, 都需要经历一个过渡变化的阶段, 我们将声音从无到有的这段时间, 叫做上升时间 (Attack Time), 而将声音从有到无的时间, 叫做释放时间 (Release Time)。

钢琴的 ADSR 特征

MATLAB 编程实现
电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

像钢琴这样的乐器，除了按下琴键时上升时间很短之外，琴弦在按下琴键后到抬起手指之前还会持续振动，因而钢琴会持续发声。

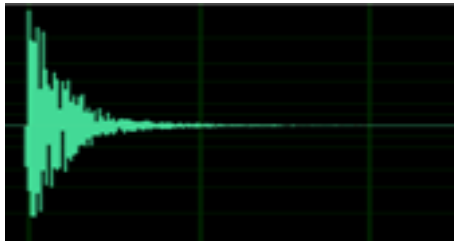


Figure: 实际测量的钢琴音符波形

在这段时间里，琴弦的振动会随着时间缓慢减弱。这段振幅慢慢减小的时间叫做衰减时间 (Decay Time)。

折线 ADSR 包络的代码实现

MATLAB 编程实现
电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

使用两点式方程和阶跃函数实现 (4) 中的折线 ADSR 包络。

```
1 function env=GenADSRenv(x,vx,vy)
2     tv=linspace(0,1,length(x));
3     vxi=[0,vx];
4     vyi=[0,vy];
5     ya=zeros(1,length(tv));
6     for i=1:(length(vxi)-1)
7         ya=ya+Line(tv,[vxi(i),vyi(i)],[vxi(i+1),vyi(i+1)]).*(tv
8             >=vxi(i)&tv<vxi(i+1));
9     end
10    env=ya;
11 function y=Line(x,p1,p2)
12     y=(x-p2(1)).*(p2(2)-p1(2))./(p2(1)-p1(1))+p2(2);
13 end
```

乐谱文件结构

MATLAB 编程实
现电子乐合成器
仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

D 76 Fs=48000

6 0 0 0.5

7 0 0 0.5

1 1 0 1.5

7 0 0 0.5

1 1 0 1

3 1 0 1

7 0 0 3

0 0 0 1

乐谱文件由包含调号、节拍和抽样率的文件头与音符表组成。

文件读取模块

MATLAB 编程实
现电子乐合成器

仇琨元 11913019

Introduction

合成器总体架构

音阶发生器

人耳频率响应

十二平均律

代码实现

连续波合成器

钢琴弦振动方程

代码实现

包络发生器

ADSR 包络

代码实现

乐谱文件读取

代码实现

```
1 script=fopen("script_collective.txt","rt");
2 key_mat=fscanf(script,"%c %d Fs=%d\n",[1,3]);
3 scr=fscanf(script,"%d %d %d %f\n",[4,inf]);
4 fclose(script);
5
6 Fs=key_mat(3);
7 T0=60/key_mat(2); %节拍
8 key=char(key_mat(1));
9
10 scr_note=scr(1,:);
11 scr_noct=scr(2,:)-1;
12 scr_risg=scr(3,:);
13 scr_dur=scr(4,:);
14 len=1:length(scr_note);
15 Y=[];
16
17 for i=len
18     note=[scr_note(i),scr_noct(i),scr_risg(i),scr_dur(i)*T0
19           ];
19     wfm2=SoundNote(key,note,Fs);
20     Y=[Y,wfm2];
21 end
```