

# 《信号与系统》期末大作业报告

工学院 梅庆宇 2200011191

## 6.2.1 简单的合成音乐

(1) 简谱中 1=F, 题目指出 F=349.23Hz, 根据十二平均律, 可以算出简谱中其他数字对应的频率, 一个八度内, 1-7 简谱对应键盘上的 0, 2, 4, 5, 7, 9, 11, 12 级音级, 核心公式是

$f = f_0 \cdot 2^{\frac{c}{12}}$ ,  $f_0$  是基频 F,  $c$  是对应的音级。对于下加点的音符需要频率除 2 降低一个八度。

东方红乐曲的频率序列为

523.2538 523.2538 587.3325 391.9974 349.2300 349.2300 587.3325 391.9974。

将简谱的数字表示为一个序列, 每个音调的持续时间也表示成一个序列。每一个音符的时间内, 使用一个 `gate` 函数与对应的正弦波点乘, 即可截取一个规定时值的音符。将所有音符按照时间连接, 得到音乐。代码块:

```
nums=length(tone)
for i=1:nums
gate=(t>=left_t&t<left_t+one_step*last_time(i)); %一个音符的起始时间和结束时间
song=song+gate.*sin(2*pi*f(i)*(t-left_t)); %连接音符
left_t=left_t+one_step*last_time(i); %进行下一个音符合成
```

播放音乐能听出调子, 但每一个音连接处有杂音。

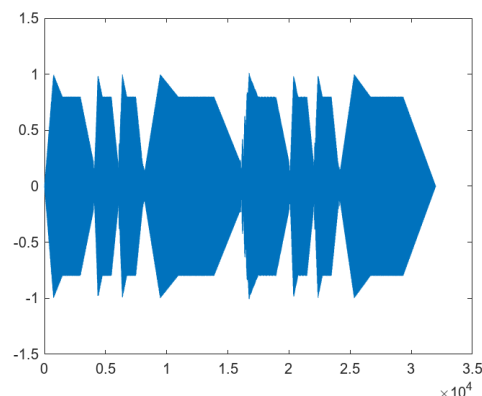
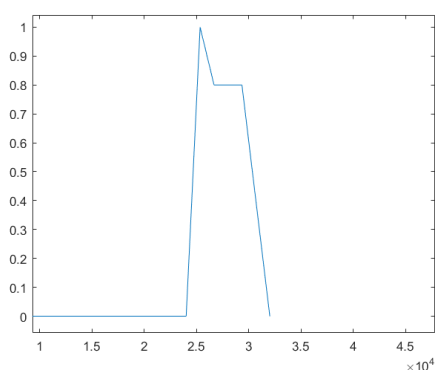
(2) 我们使用 6.5 给出的分段函数包络修正乐音, 使得音符连接处平滑, 该音符最末的衰减段与下一个音符的冲激段叠加, 不产生明显的杂音。首先把每一个音符的正弦波时长延长  $scale=1.1$  倍, 使得两个相邻音符间有重叠部分。然后使用自定义的 `adj_music` 函数, 给每一个音符加上包络, 其余部分与问题(1)类似, 由此就明显减少了杂音的产生。

采用的包络线的解析函数式是:

$$f(t) = \begin{cases} \frac{6}{time} * (t - t_0), & t_0 \leq t < t + \frac{time}{6} \\ \frac{-1.2}{time} * (t - t_0 - \frac{time}{6}) + 1, & t_0 + \frac{time}{6} \leq t < t_0 + \frac{time}{3} \\ 0.8, & t_0 + \frac{time}{3} \leq t < t_0 + \frac{2}{3} time \\ \frac{-2.4}{time} * (t - t_0 - time), & t_0 + \frac{2}{3} time \leq t < t_0 + time \end{cases}$$

其中  $time$  指本段音乐时长,  $t_0$  是起始时间。

波形如图:



(3) 最简单的升高和降低一个八度的方法是把频率序列中计算出的各音符的频率分别乘以 2

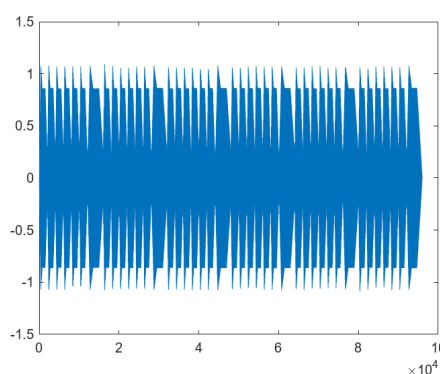
和除以 2。如果采用 `resample` 重新采样，让 `resample` 以  $8000 \times 1.0595$ （即  $\sqrt[12]{2}$ ）采样率重新采样，即可让音乐上升一个音阶。同理以 16000 采样率重新采样，即可上升一个八度。

(4) 在音符的连接段，在基频正弦波之后加上二次、三次的谐波分量，可以使得音符有更高次的谐波，听起来更真实。通过给谐波赋予不同的权重值，来调整音符的音色。本题我们给基波、2 次谐波、3 次谐波分别赋予 1, 0.21, 0.2 的权值。此外，要进一步增强听感的真实度，还可以在谐波权重里加入一个随机值  $0.005 \times \text{rand}(1)$  (0~0.005 的随机数)，使得听起来更真实。在上述谐波的加入下，听起来更像钢琴的声音。谐波部分代码块：

```
weight = [1,0.21,0.2]; %各谐波分量权重
song_of_single_key= gate.* ( (weight(1)+0.005*rand(1))*sin( 2*pi*f(i)*(t-left_t)) +...
    (weight(2)+0.005*rand(1))*sin( 4*pi*f(i)*(t-left_t)) + ...
    (weight(3)+0.005*rand(1))*sin( 6*pi*f(i)*(t-left_t)));
```

(5) 我们另来合成《小星星》，主要音符序列是：

[1,1,5,5,6,6,5,4,4,3,3,2,2,1,5,5,4,4,3,3,2,5,5,4,4,3,3,2,1],时长序列是 1,1,1,1,1,2。合成后音乐的波形图如下：



## 6.2.2 用傅里叶级数分析音乐

(1) 使用 `audioread` 指令读取 `fmt.wav` 音频文件，使用 `sound` 播放。`fmt.wav` 文件播放听起来很像真实吉他演奏出来的音乐。明显优于之前的合成音乐。

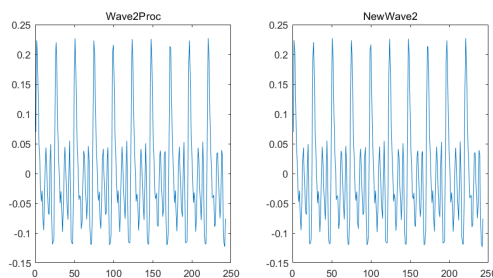
(2) 绘制两个信号的图形进行比较，`wave2proc` 波形少了很多噪声和不和谐的波形。明显看出信号大致是由 10 个周期的波形组成，因此可以考虑使用**平均值滤波**。

原来信号采样点不一定是 10 的整数倍，因此可以先使用 `resample` 函数增加 10 倍的采样率。扩大十倍采样后的信号的总采样点除以 10 就是一个周期的采样点。我们对十个周期的对应位置的点（即音乐的振幅）做平均值，所得的平均值作为目标周期对应位置的值，对 10 个周期内对应位置的每一个点都进行此操作，我们就得到一个周期内的平均波形。

然后让该波形在全周期上复制延展十次，以 1/10 的采样率再重新采样一次，就实现了滤波的目的。平均值滤波核心代码块：

```
for i=1:len
    for k=0:9
        new_song(i)=new_song(i)+song(i+k*len); %新周期的第 i 点值是十个周期 i i+len i+2len ... i+9len 的值的均值
    end
    new_song(i)=new_song(i)/10;
end
```

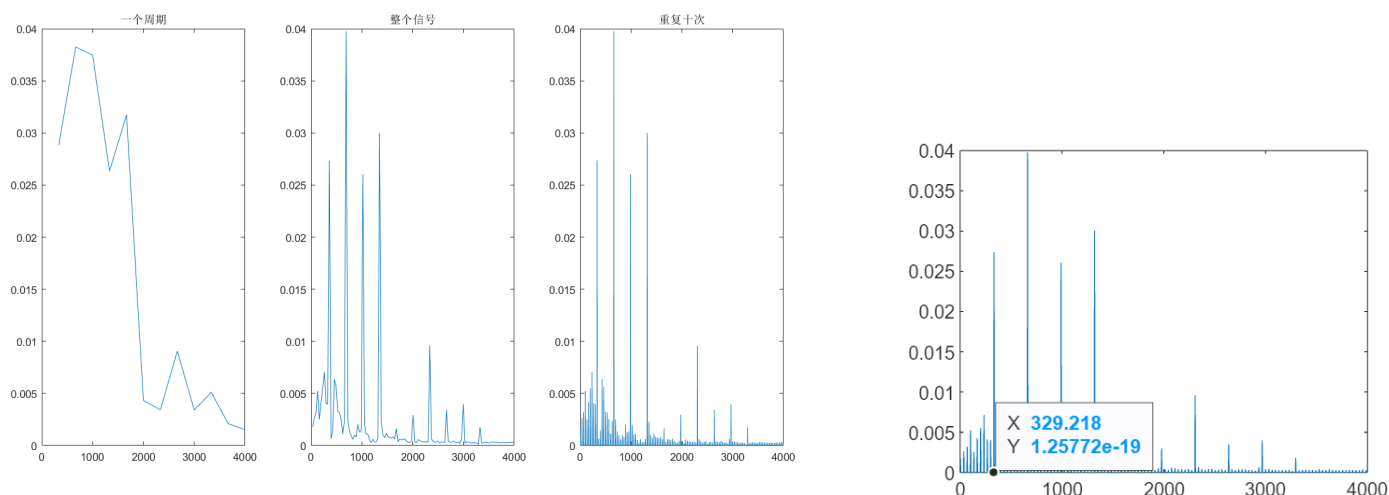
前后波形对比:



(3) 先取信号的一个大概周期进行傅里叶变换,发现极其难以辨识频域特征。我们取一个整 `realwave` 进行傅里叶变换,效果就明显较好了,可以大概看出基频和各次谐波。将 `realwave` 信号重复十次,再进行傅里叶变换,频谱更加接近冲激信号组合的形式。

根据信号的傅里叶变换频谱图可以得到,信号基频是 329Hz, 查找音调表可知是 E 调。

增加时域信号的数据量,即将信号重复若干次,信号更接近周期信号,傅里叶变换后就更接近若干并立的冲激  $\delta$  信号,更加明显分辨出频域特征。

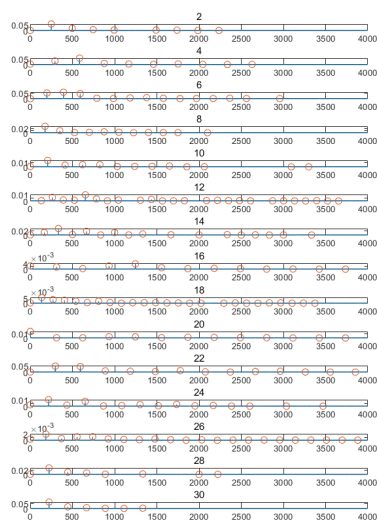
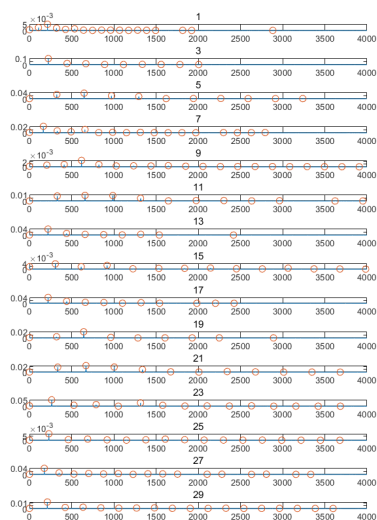
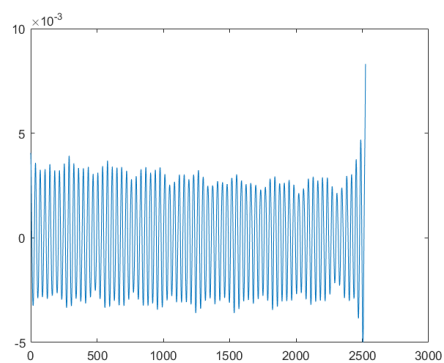
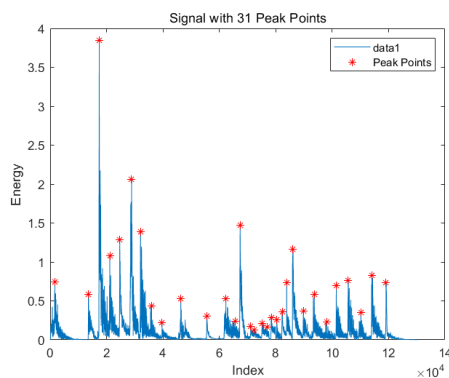


(4) 对于音乐信号,我们把幅度-时间信号转化为能量-时间信号。计算信号中连续 `step` 长度的子段的能量(即子段中所有元素的平方和),并将这些能量值存储在一个名为 `energy` 的向量中。这里 `step` 的选择需要进行尝试。之后,我们找到 `energy` 的尖峰点。这样音乐的高能尖峰点对应的就是每一个音符的起始时间位置,求取差分就知道了每一个音符的时长,由此可分析音乐的节拍。如图 31 个尖峰点对应了 30 个音符的时间起始位置,分析求出一拍时长 0.4688s。具体每个音符的情况存储在文件 `mqy_msg.txt` 中。

除了求取节拍外,还要求取音调。上面我们利用尖峰点把音乐按照音符进行了划分。考虑对于音乐信号的每一个音符进行平均值滤波,减少噪音的干扰。然后利用与问题(3)类似的重复傅里叶变换法找到每一个音符的基频和谐波信息,对于基频信息与我们已经求到的频率-音级对照,即可获得音乐的音调信息。

具体来看,对于一个音符,要想进行平均值滤波,首先需要找到每一个音符内振动周期。我们使用自相关系数法,利用 `xcorr` 函数求取音符内一段声音的自相关系数,看到有明显的周期性。再找到尖峰点,即可找到周期。在进行平均值滤波时,我们要把一个周期内的振幅归一化,获得同幅度的振动信息,方便后续的傅里叶变换。

对于每一个音符,重复上面的过程,然后对得到的周期信号进行傅里叶变换,获得了每一个音符的基频和谐波信息,读取之后储存在 `mqy_msg.txt` 文件中。



### 6.2.3 用傅里叶级数合成音乐

(1) 我们在上面的问题中获得了每一个音符的基波和谐波的频率和幅度信息，储存在矩阵中。在本题中可以利用此傅里叶信息，对音乐合成的谐波进行重新设计。

对于要合成的每一个音符，我们先根据音符的频率在矩阵中找到对应的谐波信息，然后利用循环，读取谐波的幅度，得到正弦波并不断累加在一起，就获得了一个音符的基波和谐波，之后重复问题 6.2.1 的音乐制作过程即可。

关键的基波和谐波累加的代码：

```
[r,c] = size(weight);
y=0*t;
for j= 1:c
    if(weight(n,j)~=0)
        y = y + weight(n,j)*sin(2*pi*j*f(i)*(t-left_t));
    end
end
%上面完成各次谐波的加和
song_of_single_key= gate.*y;
```

(2) 事先把东方红的乐谱和节奏分别存储在两个数组 `dfh_song_tone` 和 `dfh_last_time` 里。然后只需要在(1)的基础上，稍加修改，即可演奏出一首完整的《东方红》乐曲。