

MATLAB 音乐合成大作业

2019010562 无 97 潘修睿

1.2.1 简单的合成音乐

(1) music_01.m

$$1 = F \frac{2}{4} \quad | \quad 5 \quad \widehat{5 \ 6} \quad | \quad 2 \quad - \quad | \quad 1 \quad \widehat{1 \ 6} \quad | \quad 2 \quad - \quad |$$

歌曲为 F 大调, 所以 $do = 349.23\text{Hz}$ 。通过十二平均律分别计算出每个音符对应的频率, 设定好每个音符的延时和时长, 使用 `sound` 函数播放即可。

单频的声音听起来有浓浓的电子音乐味道, 而且音符之间会有“啪”的声响。

(2) music_02.m

根据实验指导书, 当四分音符时长 0.5s 时, 计算出冲激、衰减、消失的包络函数为 $e^{9.1629x} - 1, 1.5e^{-4.05465(x-0.1)}, 1.5e^{-7.3241(x-0.4)} - 0.5$ 。其中, 冲激时间为 0~0.1s, 衰减时间为 0.1~0.2s, 保持时间为 0.2~0.4s, 消失时间为 0.4~0.55s。八分音符的包络函数将 x 的系数加倍、时间减半即可, 二分音符将 x 的系数减半、时间加倍即可。

这样消除了相位不连续, 去除了“啪”的声响。且声音听起来有管乐的音色。

(3) music_03.m

因为音乐播放的时间可以变化, 因此最简单的方法是直接修改 `sound` 函数中音乐的采样频率。2 倍频率即将音阶升高八度 (速度变快), 频率减半即降低八度 (速度减慢)。使用 `resample` 函数升高半个音阶, 即采样率升高 $2^{(1/12)}$, 然后使用 `resample` 重采样即可。

(4) music_04.m

选择基波幅度为 1, 二次谐波幅度为 0.2, 三次谐波幅度为 0.3。即将原来的 $\sin(wx)$ 修改为 $\sin(wx) + 0.2\sin(2wx) + 0.3\sin(3wx)$ 。

为每个音符添加谐波后, 声音更加饱满厚重, 有了风琴的音色, 听起来已经和真实的音乐有些相像了。

(5) music_05.m



选择小星星的第一节进行合成。仍然使用上一题中计算出的包络函数和谐波函数。其中, do 为 440Hz, 其他音符按十二平均律计算即可。

最后成功合成了“风琴”音色的小星星。

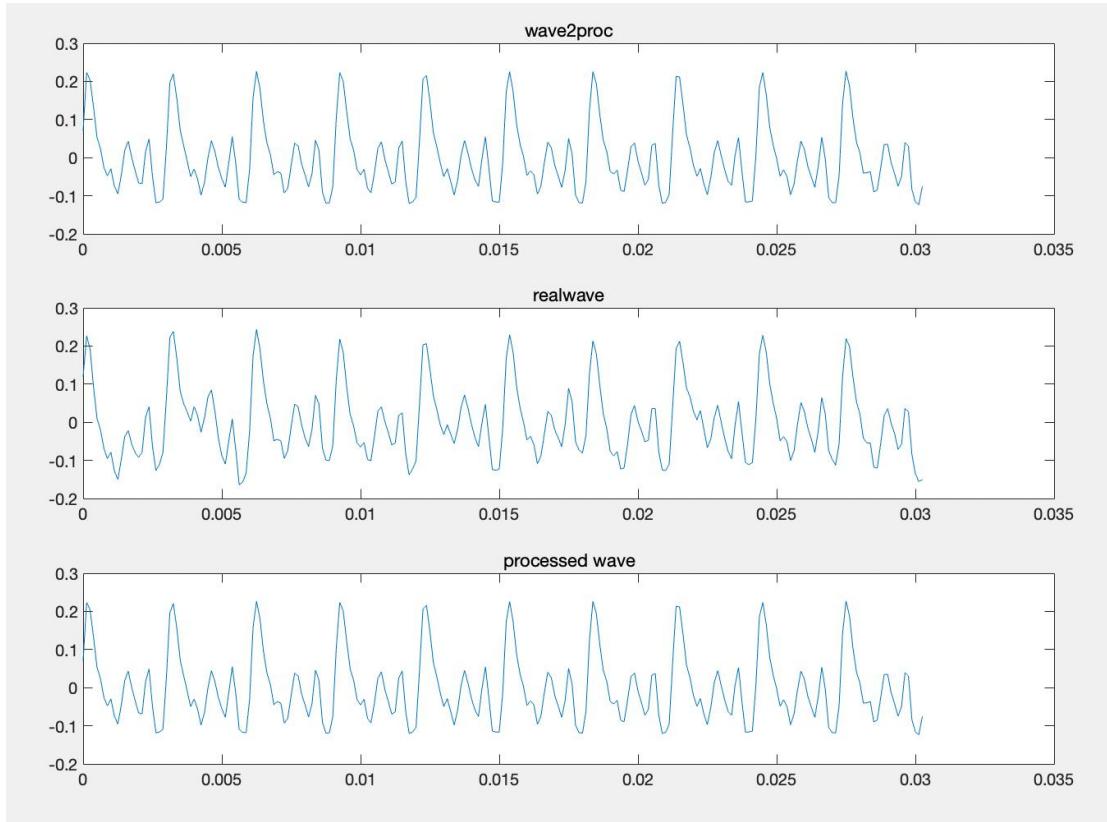
1.2.2 用傅里叶级数分析音乐

(6) music_06.m

新版的 matlab 已经禁用了 `wavread` 函数，改用 `audioread` 函数读取。

这是真实录制的吉他音乐，声音十分真实，和刚刚合成的音乐能听出明显区别。

(7) music_07.m



观察 `realwave` 发现，这个波形近似一个周期信号，约有 10 个周期。考虑到噪声基本为随机底噪，因此考虑叠加十个周期的信号取平均来消除底噪。

声明有非原创内容，消除噪声的方法与同学进行了讨论。此外细节和具体实现均为独立完成。

进行此步预处理后，绘制出 `processed wave` 波形，发现与 `wave2proc` 波形几乎完全一致，这说明预处理的操作是正确的。

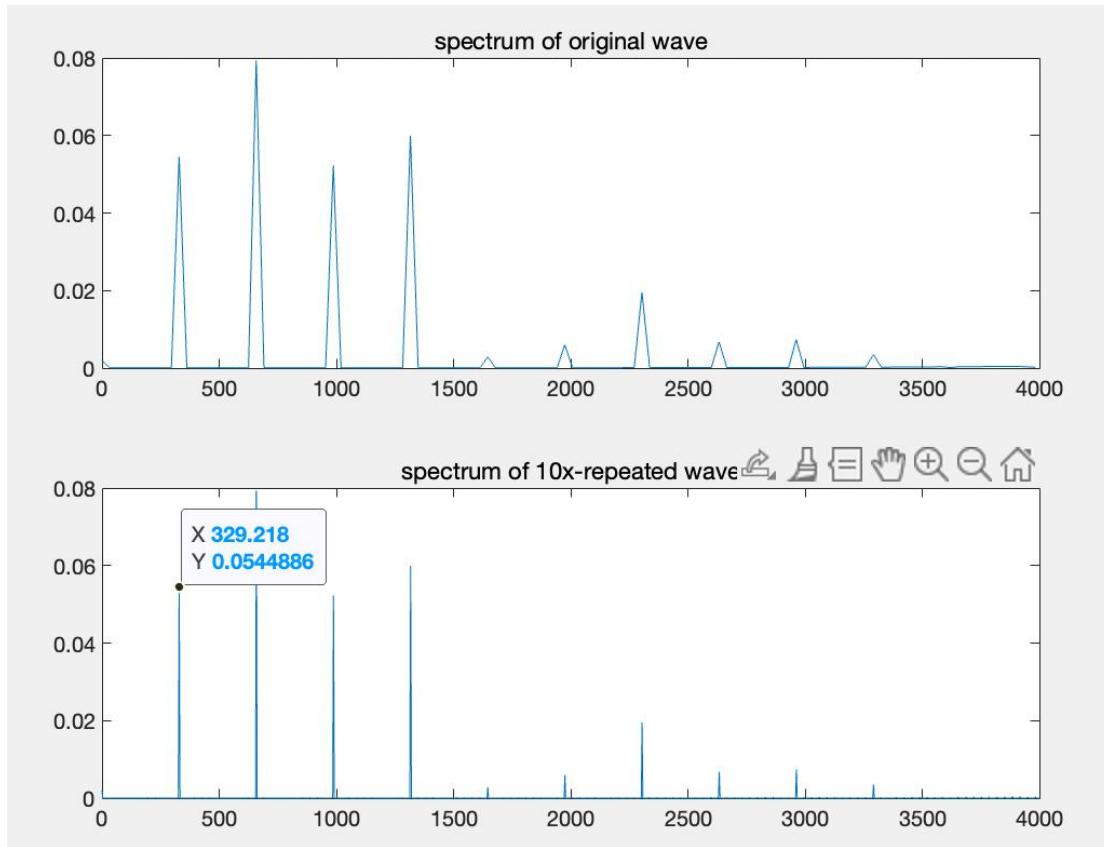
(8) music_08.m

先运行 `music_07.m`，因为使用了处理后的 `wave` 变量。

声明有非原创内容，FFT 的使用方法查找了网络上的资料。

首先尝试直接对整个波形使用 FFT 来求傅里叶变换，这相当于近似 10 个周期。变换出的频谱可以看出，每个频点的波形并非冲激函数，说明周期重复还不够，频点附近还带有一些邻近的分量。

然后再对原波形重复 10 倍，这相当于 100 个周期。重复后变换出的频谱已经很“尖”了，效果很好，近似冲激函数了。



解释：在未经过重复的时候，原信号只有近似 10 个周期，严格来说并不是周期信号，所以得到的频谱仍然是连续的。在重复了 10 倍后，信号变为 100 个周期，已经和周期信号较为接近，因此得到的频谱近似离散的冲激函数。

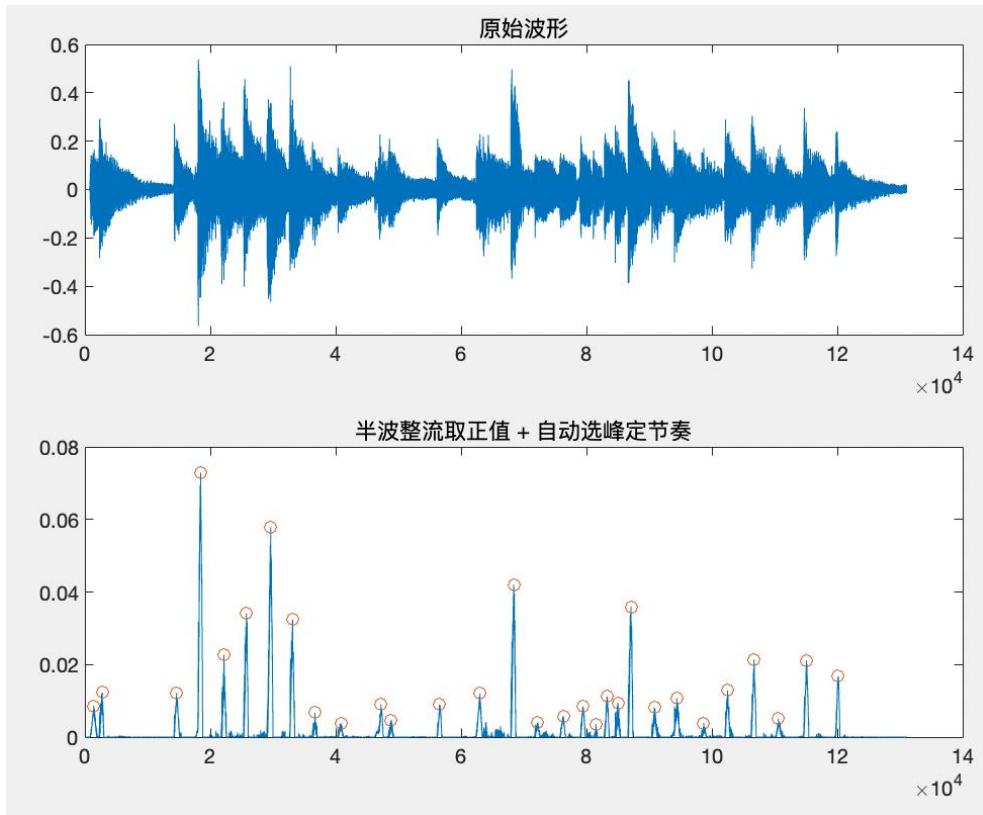
由于快速傅里叶变换得到的频点关于奈奎斯特频率对称，所以可以只分析奈奎斯特频率以下的频点。分析 10 倍重复后的频谱，得到基频为 329.218Hz，为 E 大调。

(9) music_09.m、freq.txt

本题大概是这个大作业中最复杂的一项，花费了不少时间。

首先是自动定节拍。**声明有非原创内容**，自动定节拍的算法参考了信号与系统大作业给出的算法，节奏点的选定在网络上查找相关论文后选择了非极大值抑制算法。

经过幅度平方求能量、加窗平滑求包络（选取宽度为 1000 的汉明窗）、差分提取变化点、半波整流取正值后，得到节奏谱。分析如下节奏谱，可以看出节奏谱的峰值很好地反映了原始波形中的突变位置，而我们可以认为原始波形中突变的位置即为需要判定的节奏点。然后使用 NMS 算法进行选定，先对所有峰值进行排序后进行迭代。经过多次实验验证后选取宽度为 1000 的窗进行了 $\text{length}/4300$ 次迭代，此参数下效果最好，得到了如下的节奏谱图，用红圈圈出。可以看出，效果还是很理想的。可以很容易地通过时间间隔观察出每个音符的节拍。



然后进行音频的切分。因为节奏点位于冲激达到最高点处，要想完整截取音符还需要包括最高点前上升的一小段波形。经过测量后我选取了一个固定的偏移量 $\text{EPS}=550$ ，以每个节奏点前 550 个时间单位为分割来切分音符。

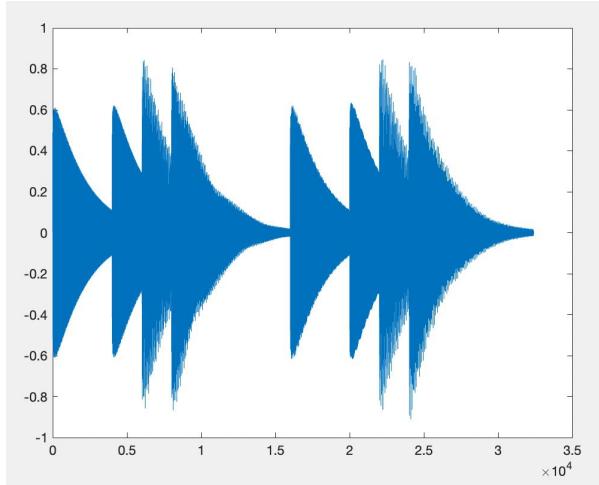
最后进行频率的分析。采用与上一题相同的方法对每个音符重复 100 次、进行 FFT，绘制频谱图。最后得到共 30 个音符，每个的频率和音调已经保存在 freq.txt 文件中。此外，每个音符的节拍也保存在该文件中了。这样，本题就可以算作完成了。

另外，在本题实现的过程中遇到了不少困难。其中一点是，有些装饰音速度很快，例如开头的几个音符，通过分析幅度突变情况根本无法有效自动定节奏。另外，有些音是多个音符同时弹奏的和弦，由于要求只分析单个主旋律的频率，只取其中一个频率进行了记录。但分析出的多个音符很难客观判断哪一个是主旋律，给工作带来了很大不便。

1.2.3 基于傅里叶级数的音乐

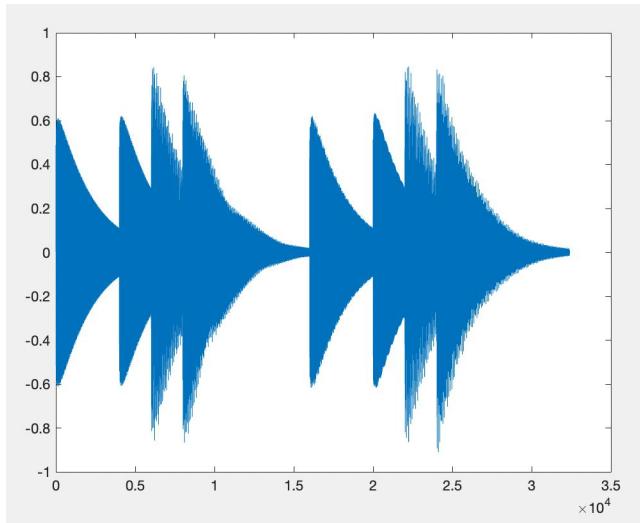
(10) music_10.m

为了让声音更像吉他演奏，我将包络改为了自己设计的函数 $y = \frac{x^{\frac{1}{15}}}{e^{4x}}$ 。使用 (7) 中计算出的频谱，记录其前 10 次谐波后保存到数组 coeff 中，然后合成《东方红》，得到如下波形。声音还蛮不错的，有些接近真实的弹拨乐器了，但感觉仍然有些刺耳和不自然。



(11) music_11.m

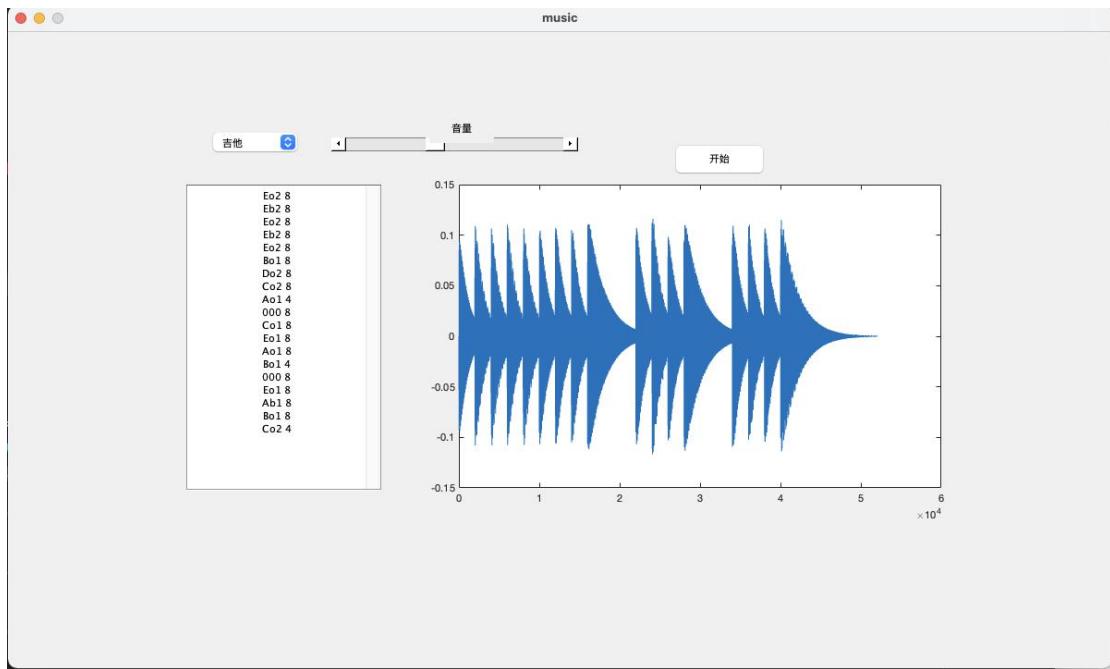
使用 (9) 中提取出的音调和对应谐波，再合成一次“东方红”，得到如下波形。由于东方红音调较高，而 (9) 中旋律较低，很难为每个音符找到对应的频率，因此只能取相近的音符的谐波数据。事实上，利用这些数据合成出的音乐很动听，像八音盒的声音。



(12) music_12.m, music.m, music.fig

本题通过 matlab 的 guide 实现了一个简易的 GUI，将前面完成的内容封装了起来。GUI 提供的功能为，输入一串音符的音调和节拍，输出这段声音，并显示波形；可以选择音色为管乐或吉他，即之前实现的指数包络和自己设计的包络；可以通过滑动滑块来调节音量；点击“开始”开始播放后，会自动将音频文件生成为 music.wav。

输入音符的语法为，音名+空格+拍子，其中音名：白键请输入{X}o{n}，{X}表示音调（如 C、D），{n}表示第几组，空拍则输入 000；拍子请输入一个数字 x，表示 x 分音符。如“Co1 4”表示 4 分音符中央 C，“Ao0 8”表示 8 分音符的 220Hz 的低音 la 等；黑键则将 o 替换为 b。具体例子如下图所示，输入的音乐是《致爱丽丝》的第一句。



成功合成了声音! 十分动听优美。至此, 本次大作业算是完成了, 十分感谢老师提供的详细文档, 让我在本次大作业中收获满满!