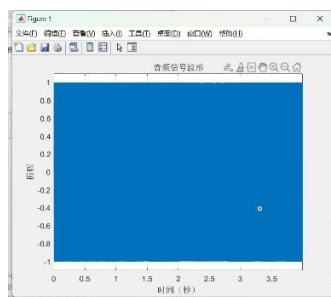


# 信号与系统 音乐合成

Bowen Tan

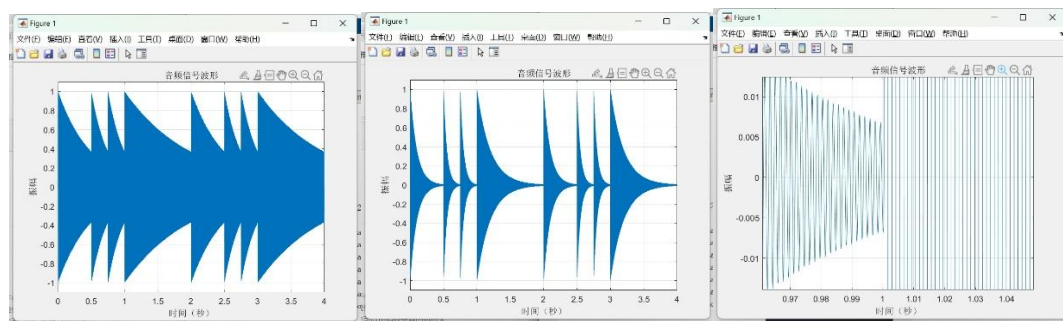
## Part 1——6.2.1 简单的音乐合成

(1) 对于每个频率为  $f$  的音符，使用  $g(t) = \sin(2\pi ft)$  的正弦波生成频率为  $f$  的声波，对声音信号不做额外处理直接播放，绘制出振幅-时间函数图像如下，由于相位不连续产生高频分量的干扰，相邻乐音之间产生“啪”的杂音，丧失真实感；



(2) 采用  $f(t) = e^{-t}$  的指数衰减包络修正乐音可以得到下方左图所示的振幅-时间图像（即通过  $g'(t) = \sin(2\pi ft)e^{-t}$  生成声波信号），杂音几乎消失，乐音具有混响的音乐效果；

若改用  $f(t) = e^{-5t}$  的指数衰减包络修正乐音可以得到下方中图所示的振幅-时间图像（即通过  $g'(t) = \sin(2\pi ft)e^{-5t}$  生成声波信号），杂音消失，乐音跳跃感更强；下方右图所示为中图的放大视角的波形；



(3) 升降八度时可以采取两种方案实现：

第一种方法为直接调整 `sound()` 函数中的采样频率，将频率调为 2 倍时相当于每个乐音频率乘 2，即升高八度，此时播放时间缩短一半；将频率调为原来的 1/2 时相当于每个乐音频率除以 2，即降低八度，此时播放时间翻倍；

第二种方法为借助 `resample()` 函数，将采样率减半得到升高八度、播放速度加倍的音乐；将采样率翻倍得到降低八度、播放速度减半的音乐。

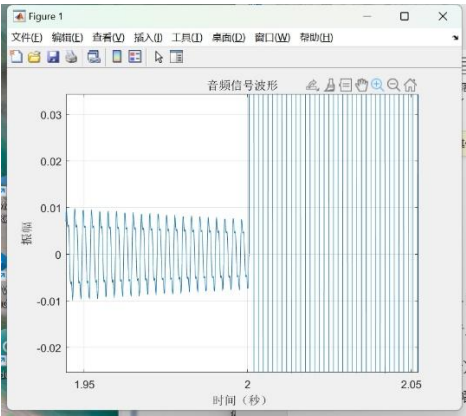
升高半个音阶时，采用估值的方式，借助 `resample()` 函数以 17:16 的采样缩放比例进行采样（大致相当于升高半个音节的频率比值），即可得到升高半

个音节的音乐。

(4) 在原声波的基础上加入二次、三次谐波，并使对应幅度分别为 0.2 和 0.3，得到新的声波函数：

$$g'(t) = [\sin(2\pi ft) + 0.2 \sin(4\pi ft) + 0.3 \sin(6\pi ft)]e^{-5t}$$

采用这个声波生成函数可以得到风琴音色的更有厚度的音乐，下图为放大波形图：



(5) 自选音乐合成：挑选加勒比海盗主题曲前多个小节进行合成，存储于文件 Question6\_2\_1\_5.m 中，考虑到乐曲特性，经过实验拟合采用模拟钢琴音色的包络  $g'(t) = 2\sqrt{t}\sin(2\pi ft)e^{-3t}$  函数制造声波，并在节奏停顿音符出增加处理得到更好的混响效果，下图乐谱及为合成后的幅度-时间函数图：

He's a Pirate

Pirates of the Caribbean

1 = G

$\frac{6}{8}$

5

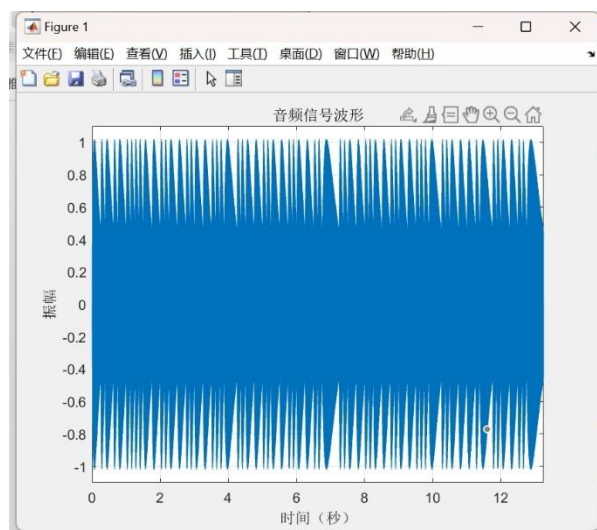
0 0 0 | 0 0 0 | 0 0 0 | 0 0 0 |  $\underline{\underline{6}} \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{6}} | \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{3}} \underline{\underline{5}} | \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{7}} | 1 \ 1 \ \underline{\underline{1}} \underline{\underline{2}} |$

10

$\underline{\underline{7}} \underline{\underline{7}} \underline{\underline{6}} \underline{\underline{5}} | \underline{\underline{5}} \underline{\underline{6}} \underline{\underline{0}} \underline{\underline{0}} \underline{\underline{3}} \underline{\underline{5}} | \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{7}} | 1 \ 1 \ \underline{\underline{1}} \underline{\underline{2}} | \underline{\underline{7}} \underline{\underline{7}} \underline{\underline{6}} \underline{\underline{5}} | \underline{\underline{6}} \underline{\underline{0}} \underline{\underline{0}} \underline{\underline{3}} \underline{\underline{5}} | \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{1}} | 2 \ 2 \ \underline{\underline{2}} \underline{\underline{3}} |$

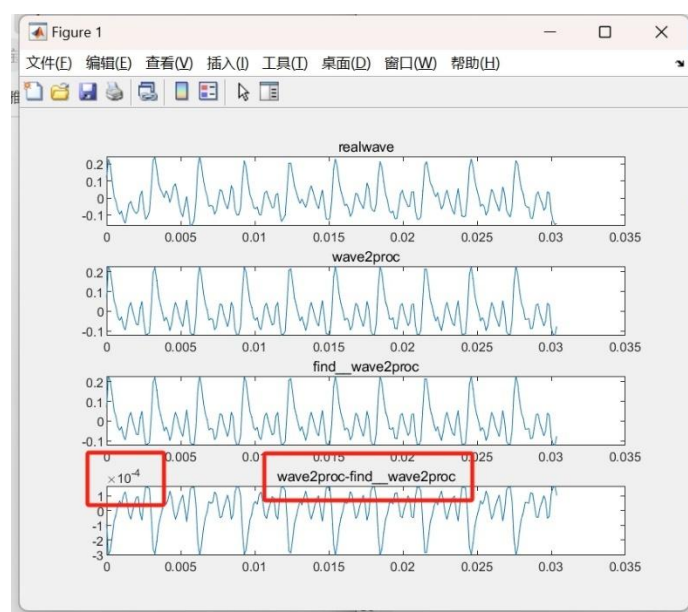
20

$4 \ 4 \ \underline{\underline{3}} \underline{\underline{2}} | \underline{\underline{3}} \underline{\underline{6}} \underline{\underline{0}} \underline{\underline{0}} \underline{\underline{6}} \underline{\underline{7}} | 1 \ 1 \ 2 | \underline{\underline{3}} \underline{\underline{6}} \underline{\underline{0}} \underline{\underline{0}} \underline{\underline{6}} \underline{\underline{1}} | \underline{\underline{7}} \underline{\underline{7}} \underline{\underline{1}} \underline{\underline{6}} | \underline{\underline{7}} \underline{\underline{0}} \underline{\underline{0}} \underline{\underline{3}} \underline{\underline{5}} | \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{6}} \underline{\underline{7}} | i \ i \ \underline{\underline{1}} \underline{\underline{2}} |$



## Part 2——6.2.2 用傅里叶级数分析音乐

(2) 由于噪声具有非周期性的特点且 `realwave` 数据集中出现了 10 个周期，想对 `realwave` 中的数据降噪得到接近于 `wave2proc` 的数据，先对 `realwave` 的数据用 `resample` 函数进行 10 倍插值，获得 10 倍采样率，再对十个周期的波形取均值，重复还原为 10 个周期，即可完成降噪得到 `find_wave2proc` 数据集。从下图最后一张子图中可以看出，`find_wave2proc` 与 `wave2proc` 作差结果的误差仅为  $10^{-4}$  数量级，可见结果与题目给定的 `wave2proc` 非常相似，成功实现降噪。

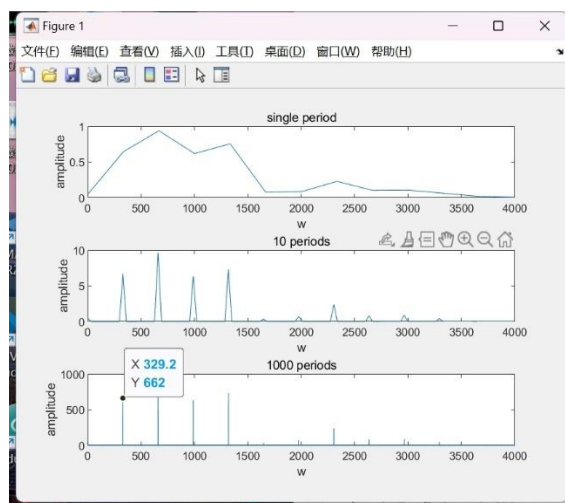


(3) 为了减小频谱带宽，将 (2) 中得到的 `find_wave2proc` 函数重复 1000 次，再作傅里叶变换得到下方右图所示的频谱图，即可得到近似脉冲信号的频谱信息，由图像可知，基频  $\omega_1 = 329.2$ ，其对应音调 E ( $e^1$ )。统计右侧下图每个谐波频谱的纵坐标，求出其相对于基波幅度（由下方右图可知为 662）的比值，即可得到相应谐波分量：记  $A_1 = 1$ ，则：

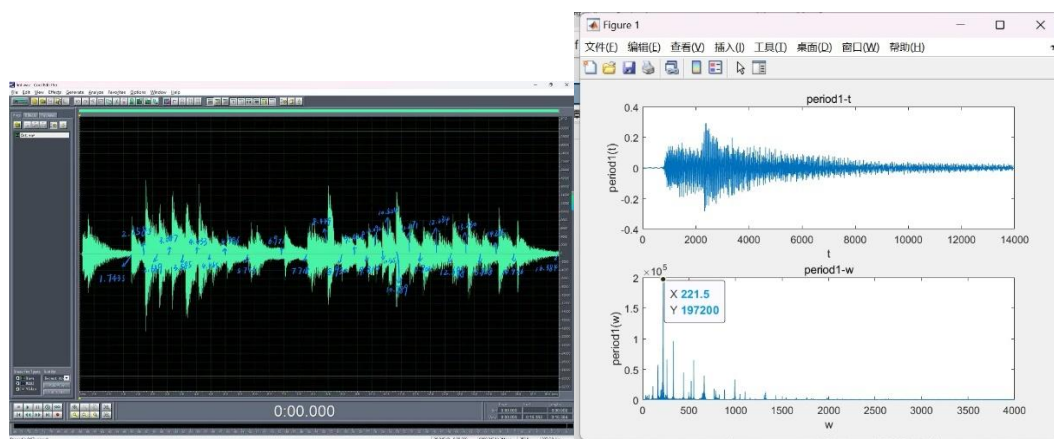
$$A_2 = 1.46, A_3 = 0.96, A_4 = 1.10, A_5 = 0.05,$$

$$A_6 = 0.11, A_7 = 0.36, A_8 = 0.12, A_9 = 0.14,$$

$$A_{10} = 0.06, A_{11} = 0.01$$



(4) 先利用 Cool Edit 软件手工标定每个音调的起止时间，如下图（左），例如第一个音符 0 秒开始，1.743 秒结束，第二个音符 1.743 秒开始，2.158 秒结束等等，再利用程序单独提取出每一个音符，为使频带宽度更小，将每一个音符重复 1000 次后进行傅里叶变换，得到该音符的频谱信息，即可得知该音符的基频和音调。以第一个音符为例，截取第一个音符片段，进行傅里叶变换后，用函数提取出其频谱图中第一个幅度最大的频率分量，即为对应基频，从下图（右）可知第一个音符的基频为 221，对应音符为 A（a）。



利用程序分析每个音符的基频(由于音符为手动分割可能存在误差)，得到所有音符基频结果如下图（左上角\*1.0e3 表示每个数据乘 1000）：

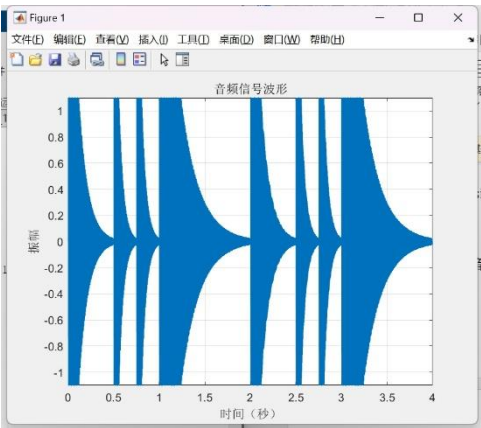
1.0e+03 *								
列 1 至 9								
0.2215	0.2482	0.2225	0.5893	1.3153	0.1964	0.2222	0.1751	0.2945
列 10 至 18								
0.2476	0.3293	0.2216	0.6570	0.4418	0.2199	0.3936	0.3519	0.6583
列 19 至 27								
0.5887	0.6577	0.2483	0.2932	0.2608	0.1748	0.2209	0.2212	0.2219
列 28								
0.2097								

Part 3——6.2.3 基于傅里叶技术的合成音乐

(1)根据在 Part2 中计算出来的傅里叶技术各谐波分量，将 Part1 中的声波生成函数改写为：

$$g'(t) = [\sin(2\pi ft) + 1.46 \sin(4\pi ft) + 0.96 \sin(6\pi ft) + 1.10 \sin(8\pi ft)]e^{-5t}$$

即可获得吉他音色的演奏效果。



(2) 根据 Part2-（4）中的结果，将分离出的每个频率的音符分别进行傅里叶变换后，找出对应单倍、双倍、三倍、四倍谐波的傅里叶级数系数 $A_{1i}$ 、 $A_{2i}$ 、 $A_{3i}$ 、 $A_{4i}$ ，对于每一个对应的音符，按照相应提取出的系数，采用公式：

$$g'_i(t) = \left[ \sin(2\pi ft) + \frac{A_{2i}}{A_{1i}} \sin(4\pi ft) + \frac{A_{3i}}{A_{1i}} \sin(6\pi ft) + \frac{A_{4i}}{A_{1i}} \sin(8\pi ft) \right] e^{-5t}$$

对每个音符分别生成对应的音乐效果，即可得到吉他音色特征更强的音乐效果演奏的《东方红》，包络形状与上图类似。

(3) 将上述实现的部分功能进行设计加工和封装，得到图形交互界面，此功能在程序 Question6\_2\_3\_3.mlapp 中，点击文件即可直接运行得到如下图形交互界面：

