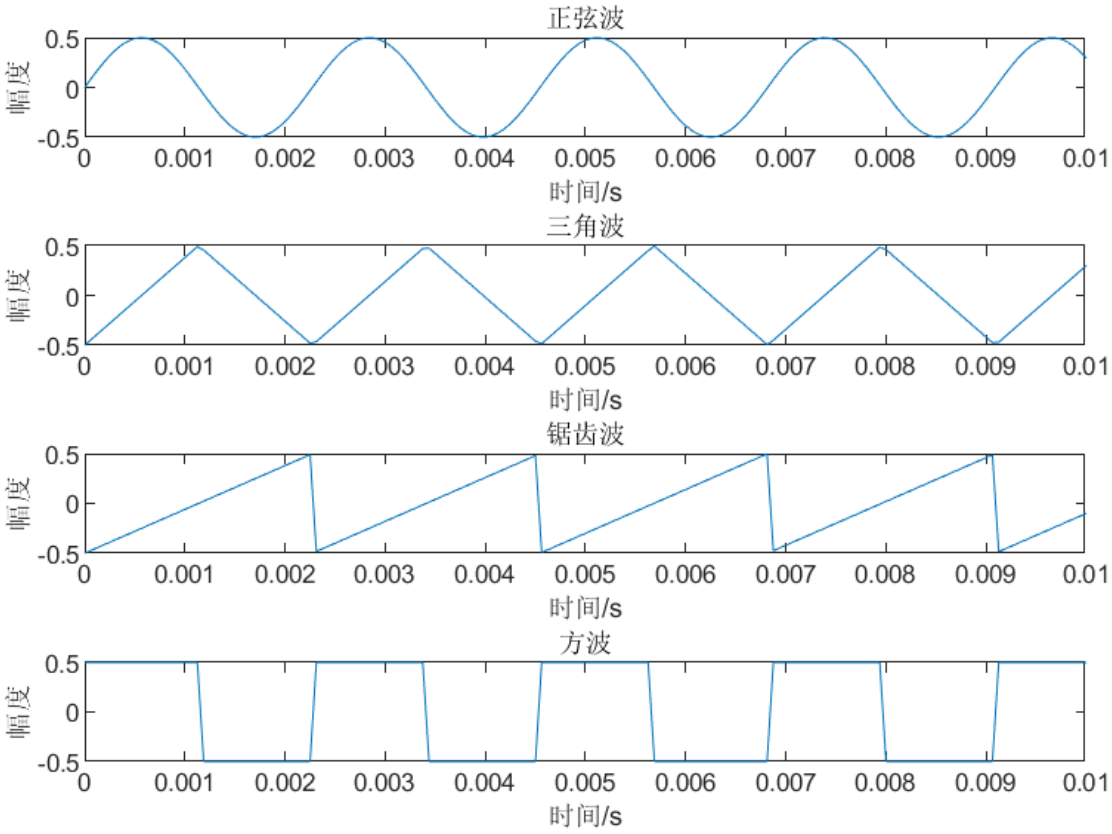
DSP 大作业——电子音乐

谢国强 无 02 2020010674

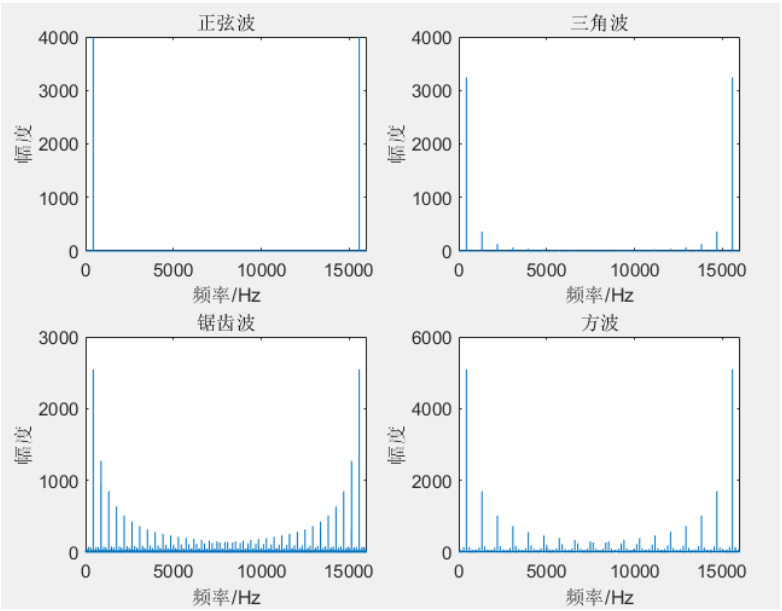
一、电子合成器

（1）通过 matlab 自带的 sin、sawtooth、square 函数生成指定参数的信号波形

因为频率问题绘制 1s 的图像不好观察，因此只绘制了 0.01s 的图像，便于观看



（2）通过 FFT 获取并绘制四种波形的图谱



锯齿波是一种带有快速上升和下降的波形，其频谱包含了基波频率和其奇次谐波频率。具体而言，三角波的频谱中包含了频率为基波频率、3 倍、5 倍、7 倍..的奇次谐波，其幅度按照 1/fˆ2 的规律递减。正弦波是一种平滑的波形，其频谱只包含了基波频率。具体而言，正弦波的频谱中只有频率为基波频率的分量，其幅度为原始信号的振幅。因此，正弦波的频谱相对于锯齿波的频谱更为简单。从四种图像的对比也可以看出，越是具有陡升陡降特性的波形，其频谱也越复杂。

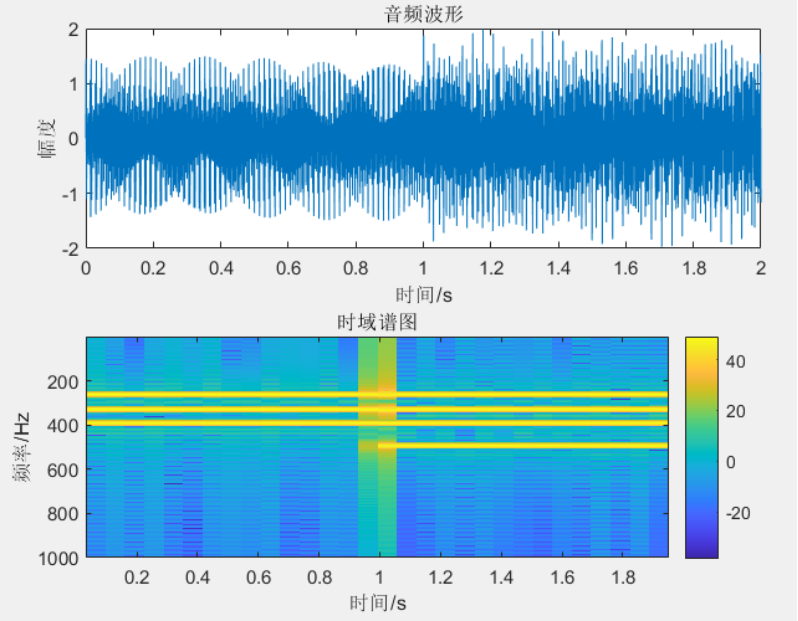
二、音乐生成

（1）通过修改频率和正弦信号叠加生成波形，通过 sound 函数产生音频

（2）通过 STFT 绘制音频的时频谱图：

选择 overlap 为设置的窗长一半，FFT 点数与窗长一致，避免引入补零等情况，，物理分辨率与观测分辨率一致，即为$f_0 = \frac{F_s}{NFFT}$，在 fs=16000Hz 的情况下，考虑要使音频容易区分，即分辨率要达到 15Hz 左右，则考虑窗长和 NFFT 至少要大于等于 1024（选择 2 的 n 次幂）。

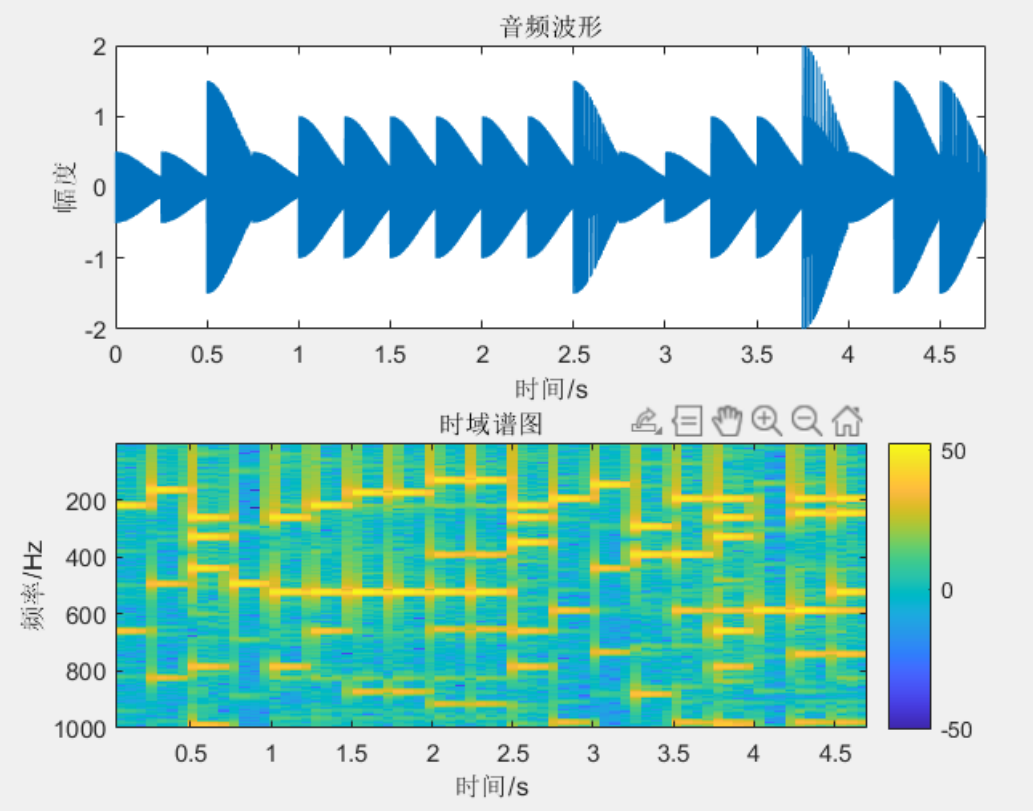
长窗具有较高的频率分辨率，较低的时间分辨率。长窗起到了时间上的平均作用。窗宽的选择需折中考虑。短窗具有较好的时间分辨率，能够提取出语音信号中的短时变化，损失了频率分辨率。考虑此次所分析的音频其时间变化并不大，不需要太高的时间分辨率，则选择长窗，设置为 2048。利用 matlab 自带的 spectrogram 函数，最终图像如下



时频谱图中的黄色线条即为音频含有的音调的频率，对照指导书中的部分音高与频率的关系即可得到音调。例如，在此图中可以看出第 2 秒比第 1 秒多出了一段约 492Hz 频率的声音，对照表格，这正是多出的 B4 调。

三、自动读谱

（1）通过 load 函数读取文件，可以如任务二中的流程获取音频和视频谱图，如下：



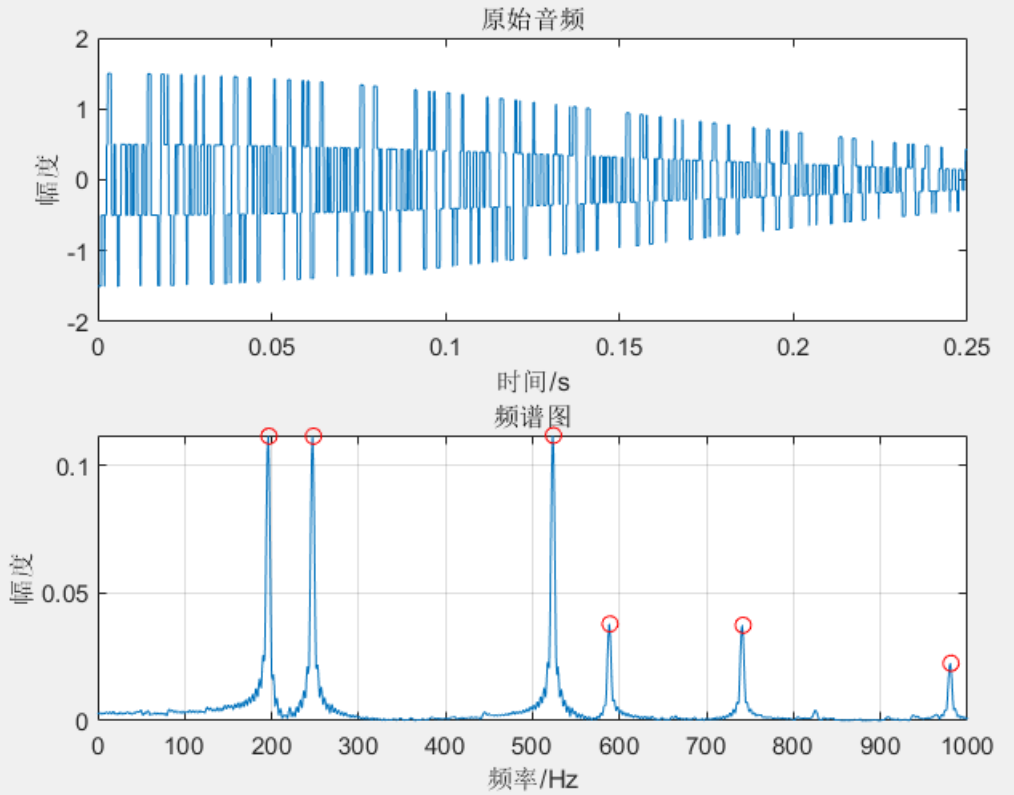
（2）获取的 STFT 频谱已经可以通过人工读取获取乐谱。但是考虑到要自动获取，因此采用 FFT 频谱分析来获取

首先，每段音频是 0.25s，一段内波形不变，所以考虑将文件切割成 19 份，进行逐个处理。在 FFT 变换后，通过 abs(fft)*2/N 获取正确的频率幅度值，然后通过 findpeaks 函数来获取局部最大值。因为本身 FFT 函数波形是波动的，因此 findpeaks 函数需要添加最小高度差参数来获取真实最大值。代码如下：

```
[pks,locs]=findpeaks(result,'MinpeakProminence',0.015,'Annotate','extents');
```

即寻找最小高度差大于等于 0.015 的局部最大值，人工核验后确认寻找无误。效

果图如下：



（3）显然获取的频率不可能与标准音高频率完全符合，因此采用寻找最接近值进行匹配。不同音高对应的频率存在相邻音倍乘系数$K = 2^{\frac{1}{12}}$的规律，因此利用 A3=220，A4=440，A5=880 可以直接获取音调-频率矩阵，不必手动输入。

```
for i = 1 : 12
    if (i ==1)
        tunes = zeros([12 3]);
        tunes(10, 1) = 220;
        tunes(10, 2) = 440;
        tunes(10, 3) = 880;
        frequency_diff = 2^(1/12).^[-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2]';
    end
    tunes(i, 1:end) = tunes(10, 1:end) .* frequency_diff(i);
end
```

将前面获取的 locs 向量和 tunes 矩阵进行相减求最小绝对值，即可获得最接近的音调频率的位置。通过自写函数 tostring 将位置转化成对应的音调。

```
for j=1:length(locs)
    if locs(j)>1000
        break;
    end
    cor=abs(tunes-locs(j));
    cor_min=min(cor,[],"all");
    [row,col]=find(cor==cor_min);
    fre(i,j)=tunes(row,col);
    str=['第',int2str(i) ,'段'];
    answer(i,1)=cellstr(join(str));
    answer(i,j+1)=cellstr(join(tostring(row,col)));
end
```

最终可以获得两张表格，如下：

fre

19x10 double

	1	2	3	4	5	6	7	8	9	10
1	220	659.2551	0	0	0	0	0	0	0	0
2	164.8138	493.8833	830.6094	0	0	0	0	0	0	0
3	261.6256	329.6276	440	783.9909	987.7666	0	0	0	0	0
4	493.8833	0	0	0	0	0	0	0	0	0
5	261.6256	523.2511	783.9909	0	0	0	0	0	0	0
6	220	523.2511	659.2551	0	0	0	0	0	0	0
7	174.6141	523.2511	880	0	0	0	0	0	0	0
8	174.6141	523.2511	880	0	0	0	0	0	0	0
9	130.8128	391.9954	523.2511	659.2551	932.3275	0	0	0	0	0
10	130.8128	391.9954	523.2511	659.2551	932.3275	0	0	0	0	0
11	220	261.6256	349.2282	659.2551	783.9909	0	0	0	0	0
12	195.9977	587.3295	987.7666	0	0	0	0	0	0	0
13	146.8324	440	739.9888	0	0	0	0	0	0	0
14	293.6648	391.9954	880	0	0	0	0	0	0	0
15	195.9977	391.9954	587.3295	987.7666	0	0	0	0	0	0
16	195.9977	261.6256	329.6276	587.3295	659.2551	783.9909	987.7666	987.7666	0	0
17	587.3295	0	0	0	0	0	0	0	0	0
18	195.9977	246.9417	587.3295	739.9888	987.7666	0	0	0	0	0
19	195.9977	246.9417	523.2511	587.3295	739.9888	987.7666	0	0	0	0
20										

上表 fre 每行存放了每段包含的音调频率。

fre answer

19x9 cell

	1	2	3	4	5	6	7	8
1	'第1段'	'A3'	'E5'	[]	[]	[]	[]	[]
2	'第2段'	'E3'	'B4'	'G5#'	[]	[]	[]	[]
3	'第3段'	'C4'	'E4'	'A4'	'G5'	'B5'	[]	[]
4	'第4段'	'B4'	[]	[]	[]	[]	[]	[]
5	'第5段'	'C4'	'C5'	'G5'	[]	[]	[]	[]
6	'第6段'	'A3'	'C5'	'E5'	[]	[]	[]	[]
7	'第7段'	'F3'	'C5'	'A5'	[]	[]	[]	[]
8	'第8段'	'F3'	'C5'	'A5'	[]	[]	[]	[]
9	'第9段'	'C3'	'G4'	'C5'	'E5'	'A5#'	[]	[]
10	'第10段'	'C3'	'G4'	'C5'	'E5'	'A5#'	[]	[]
11	'第11段'	'A3'	'C4'	'F4'	'E5'	'G5'	[]	[]
12	'第12段'	'G3'	'D5'	'B5'	[]	[]	[]	[]
13	'第13段'	'D3'	'A4'	'F5#'	[]	[]	[]	[]
14	'第14段'	'D4'	'G4'	'A5'	[]	[]	[]	[]
15	'第15段'	'G3'	'G4'	'D5'	'B5'	[]	[]	[]
16	'第16段'	'G3'	'C4'	'E4'	'D5'	'E5'	'G5'	'B5'
17	'第17段'	'D5'	[]	[]	[]	[]	[]	[]
18	'第18段'	'G3'	'B3'	'D5'	'F5#'	'B5'	[]	[]
19	'第19段'	'G3'	'B3'	'C5'	'D5'	'F5#'	'B5'	[]

上表 answer 存放了每段包含的音调名。