

DSP 大作业-芯片音乐

班 级： 无 57 班

学 号： 2015011175

姓 名： 王 复 英

时 间： 2017.11.29

任务 1 基本波形

请用 **Matlab** 实现上述基本波形发生器函数，前三个波形的时域可调节，其中矩形脉冲波的占空比为 $1/8$ ， $3/8$ ， $4/8$ 。请从时频域分析音色不同的原因。

【实现思路】

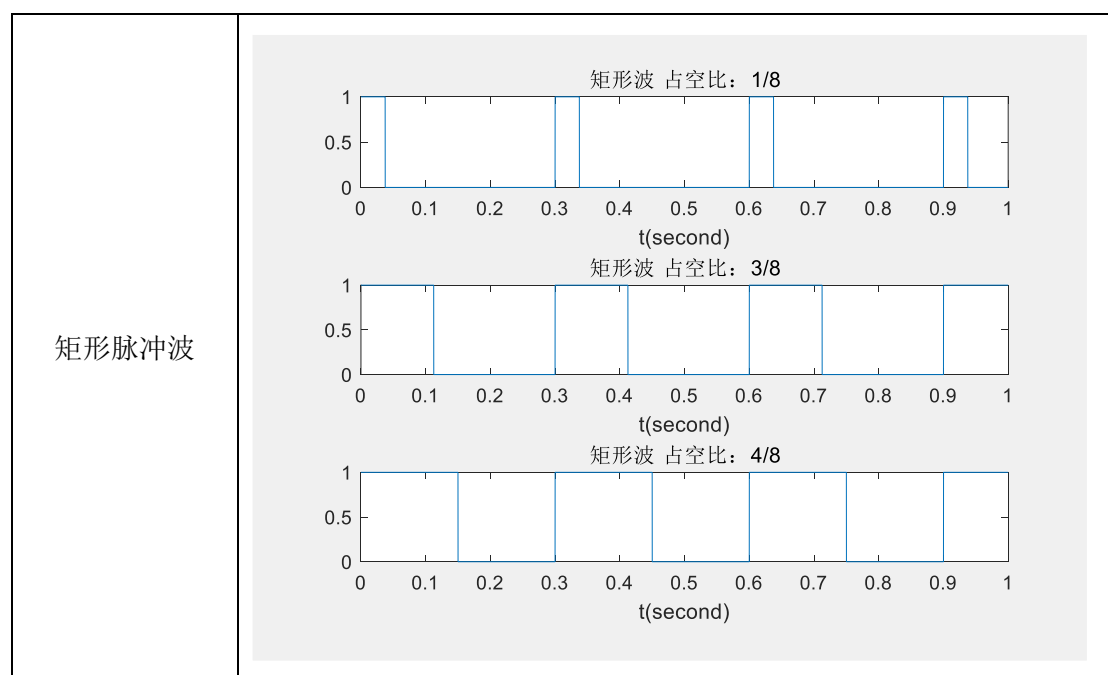
- 1) 矩形波使用 **square** 函数实现；
- 2) 锯齿波和三角波使用 **sawtooth** 函数实现；
- 3) 伪随机噪声实现时首先产生固定序列长度（这里是 **1s** 即 **8000** 个采样点）的随机数，然后将大于 **0.5** 的随机数赋值为 **1**，小于 **0.5** 的随机数赋值为 **0**。但是考虑到这样产生的信号过于密集，将随机信号长度设置为信号长度的 **1/20** 倍，再重采样得到伪随机噪声信号的波形。

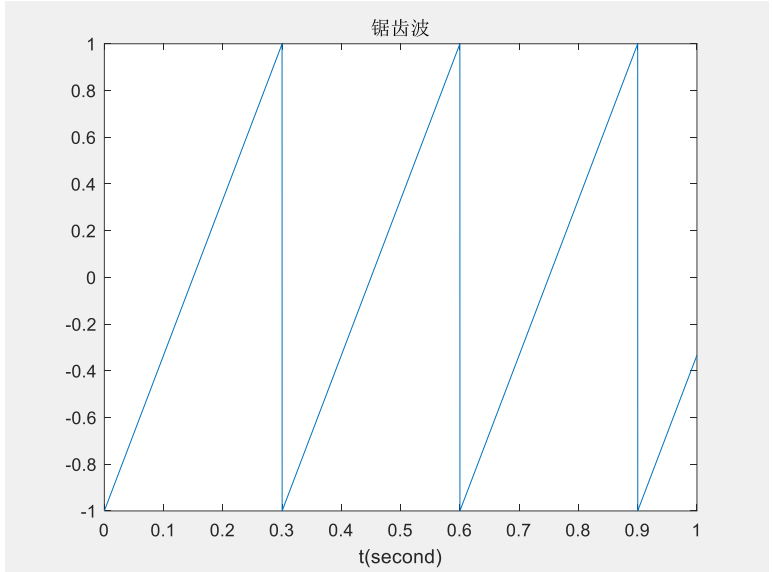
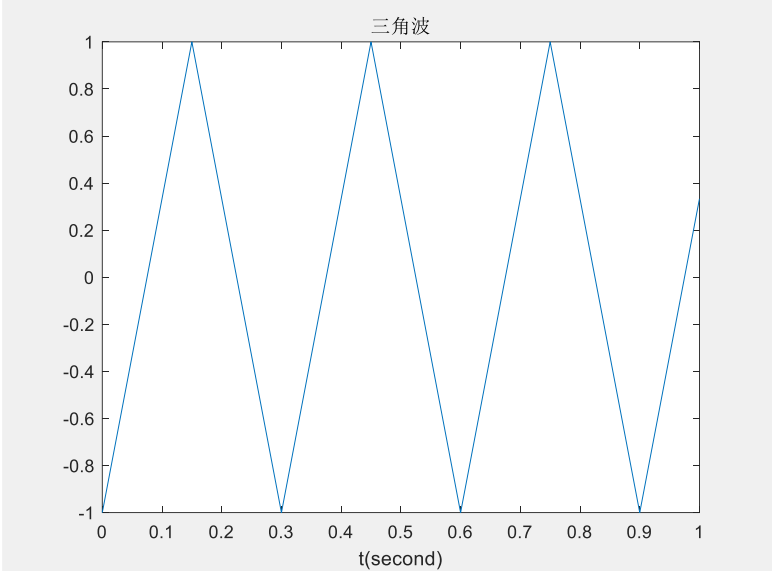
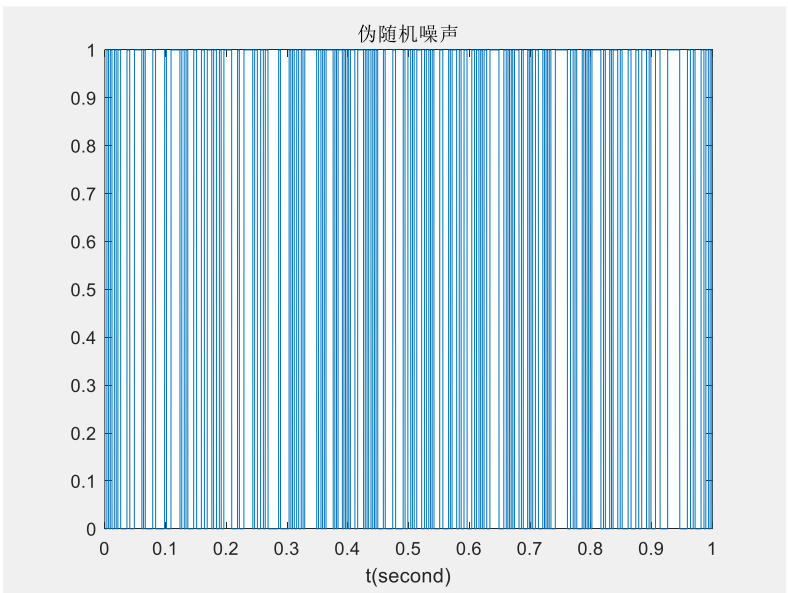
【关键代码与实现】

```
y1=double(square(2*pi*t/T,D(1)*100)>0);  
y2=double(square(2*pi*t/T,D(2)*100)>0);  
y3=double(square(2*pi*t/T,D(3)*100)>0);  
y4=sawtooth(2*pi*t/T,1);  
y5=sawtooth(2*pi*t/T,0.5);
```

【实验结果】

采样率为 **8000Hz**，下表中画出的为 **1s** 的时域波形。



锯齿波	
三角波	
伪随机噪声	

【结果分析】

1) 时域分析

音色受到以下时域波形参数的影响：**Attack**（声音从无到峰值的时间）、**IOI**（每个频段相邻两个点之间的距离）、**Decay**（由峰值降回平稳状态的时间）、**Sustain**（平稳状态的时间）、**Release**（由平稳开始衰减到无的时间）和暂态（变量改变但是还没有达到稳定）。这几种基本波形的以上参数存在很大的差异，这也决定了它们音色的不同。

2) 频域分析

音色主要是由泛音决定的。根据 **Fourier** 变换的知识，这几种基本波形的频谱分布存在很大的差异。除了基音，每种波形都存在不同的谐波组成，其泛音成分存在很大的差别，这也导致了它们的音色的差异。

任务 2 音乐合成

请选用两种基本波形，根据 **sheetCz.mat** 中给出的两声部音乐的频率序列，用 **Matlab** 生成这段音乐，存储为音频文件 **Cz.wav**，描述实现方法和效果。

【实现思路】

- 1) 首先假定采样频率为 **8000Hz**，一拍为 **1s**，四分之一音符为一拍。由此计算得每个音符的持续时间为 **1/4s**，即 **2000** 个采样点。根据给定频率依次产生长度为 **2000** 采样点的波形，将两个声部的波形叠加即得到要求的音乐。
- 2) 但是将音乐叠加后发现波形幅度较大，音色不够柔和。将波形的幅值映射到 **[0,1]** 的区间内，音乐的音色能取很好的效果。

【关键代码与实现】

```
x=double(square(2*pi*sheetCz(1,1)*t,D*100)>0)+...
    sawtooth(2*pi*sheetCz(2,1)*t);
for n=2:size(sheetCz,2)
    temp=double(square(2*pi*sheetCz(1,n)*t,D*100)>0)+...
        sawtooth(2*pi*sheetCz(2,n)*t,0.5);
    x=cat(2,x,temp);
end
```

【实验效果】

这里选用占空比为 **3/8** 的矩形脉冲波产生低声部音乐，用三角波产生高声部音乐，乐音较为悦耳。（笔者也做了其他波形组合的尝试，没有发现比该组合明显优越的波形。）

任务 3 乐谱检测

现有一段音频 `audioTest.wav`，包含一个多声部的乐段，其被噪声污染，音乐的起始时刻不明。音频工程师想分析出其伴奏的低声部的曲谱是什么。他得到了六个曲谱的频率序列 `base1 ~ base6`，以及每个音的时长 `tNote = 0.1575 s`，存储在 `whichSheet6.mat`。请用 Matlab 分析哪个曲谱是该段音乐用到的低声部曲谱？为什么？是否能分析出这段音乐的起始时刻？请思考如何提高效率？

考虑到分析该段音乐的曲谱需要在知道音乐的具体波形的基础上分析，这里先分析音乐的起始时间，再给出音乐的曲谱分析。

A. 音乐起始时刻分析

1. 时域分析

【理论依据】

相关函数：相关指的是信号本身在不同时间点的互相关。直观意义上理解，相关是信号之间的相似程度和它们之间的时间差的函数。

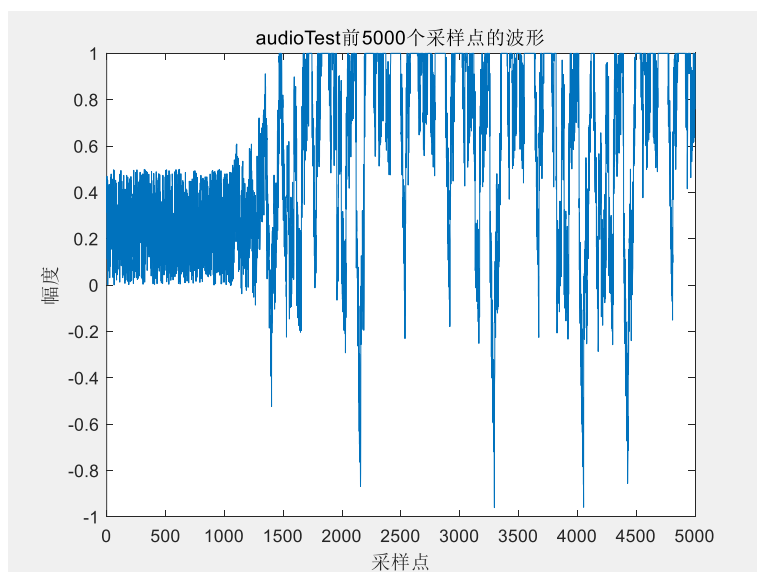
$$R_{xy}(m) = \begin{cases} \sum_{n=0}^{N-m-1} x_{n+m} y_n^* & m \geq 0 \\ R_{yx}(-m) & m < 0 \end{cases}$$

【思路分析】

在时域上看，音乐的开始时间一定在信号幅值变化较大的区域。做出信号的波形图发现音乐开始前的一段近似于高斯白噪声。因此截取前一段信号（必须同时包含噪声和音乐）做自相关，如果序列和沿时间轴平移的序列在突变点位置相似程度一定较低，只有当两段信号的时间偏移恰好为音乐的起始时间时，信号的相似程度相对较高。因此找到相关函数的次大峰值对应标号，除以采样频率即得到音乐起始时间。

【关键代码与实现】

1) 选取合适长度（这里选取前 5000 个采样点）的信号序列，画出其波形图。



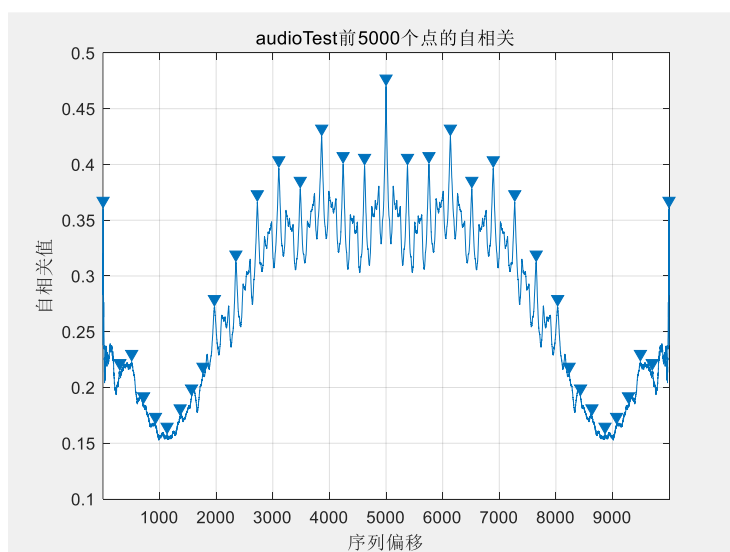
2) 做序列的自相关，这里采用无偏误差估计，防止序列的长度对相似程度造成影响。

```
x=mapminmax(x,0,1);
[a,b]=xcorr(x,'unbiased');
```

3) 寻找自相关序列的次大峰值，音乐起始时间即为次大峰值与最大峰值的距离除以采样率。

```
[peaks,locs]=findpeaks(a,'minpeakdistance',200,'minpeakheight',0.1);
[c,d]=sort(peaks,'descend');
idx=locs(d);
start_time=abs(idx(2)-idx(1))/FS    %start_time=0.0258
```

【实验结果】



由上图计算得最大峰值和次大峰值间隔为 1136 个采样点，因此音乐起始时间为 $1136/FS=0.0258s$ 。

2. 频域分析

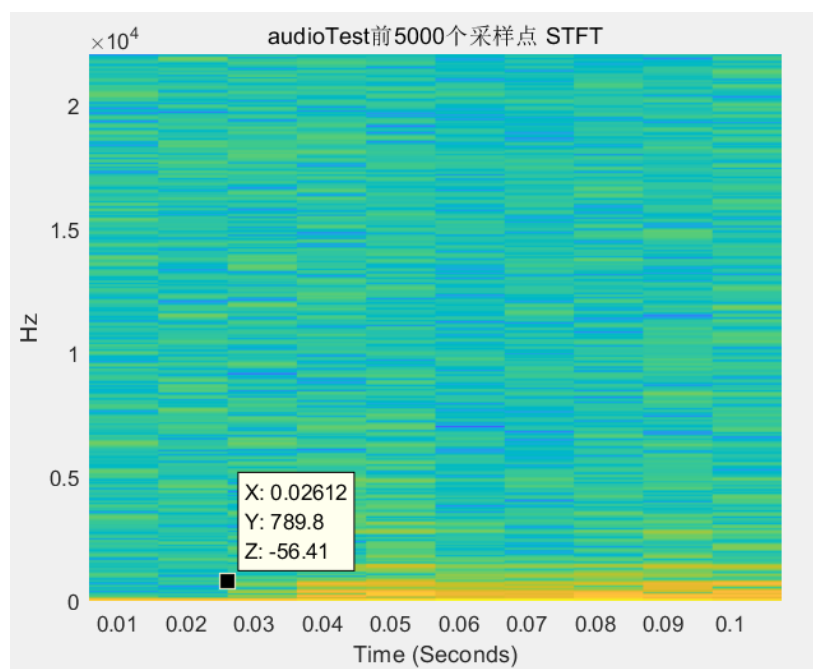
【理论依据】

STFT（短时傅里叶变换）：为了研究不平稳信号的频谱，我们需要把信号加窗处理，计算出信号在一段时间内的 **Fourier** 变换，由此就可以分析出信号的频率分量随时间的变化。

【思路分析】

截取信号的前 **5000** 个点，做短时傅里叶变换。

【实验结果】



3. 结果分析与评价

时域计算出的音乐起始时间为 **0.0258s**，频域计算出的音乐起始时间为 **0.0261s**，不过由于短时傅里叶变换存在一定误差，其计算结果没有时域根据相关性分析的结果准确，只能作为大致范围的参考。这里就取音乐开始时间为 **0.0258s**，即第 **1136** 个采样点。

B. 音乐曲谱分析

【思路分析】

根据 A 部分计算出的音乐起始时间，将音乐划分成 **96** 个片段（已知可能频率向量的长度均为 **96**）。为了确定每个片段中的频率分量和哪个频率最接近，可以采用计算相关系数的方法，分别计算 **96** 个音乐片段和 **6** 个频率序列中对应频率波形的相关系数矩阵，其中相关性最强的片段最多的序列即为可能的频率序列。

另外需要考虑高低声部的问题，如果两个频率序列占比相当，所求的为整体频率较低的序列。

【关键代码与实现】

1) 对信号做小波去噪处理；

```
yd=wden(y,'heursure','s','mln',lev,'haar');
```

2) 获得 96 个片段的起始编号；

```
idx=round(start_time*FS:tNote*FS:start_time*FS+tNote*FS*96);
```

3) 计算相关系数矩阵；

```
for k=1:6
    base=eval(['base1',num2str(k)]);
    for m=1:96
        y1=yd(idx(m):idx(m+1));
        t=0:1/FS:length(y1)/FS-1/FS;
        temp=sin(2*pi*base(m)*t); %采用 sin 函数波形
        s=corrcoef(y1,temp);
        cor(k,m)=abs(s(1,2));
    end
end
```

4) 对相关系数矩阵排序，求最大相关系数最多的频率序列；

```
[a,b]=sort(cor,1,'descend'); %column dimension
tabulate(b(1,:))
```

【结果分析与评价】

计算结果如下表所示，其中序列 1 和序列 4 所占比重大致相当，可以观察频率分量得 1 整体上频率大于 4. 因此序列 4 为该段音乐低声部的频率，序列 1 为该段音乐高声部的音乐。

Value	Count	Percent
1	43	44.79%
2	0	0
3	0	0
4	53	55.21%

C. 关于提高效率的思考

1. 算法层面

1) 音乐起始时刻分析

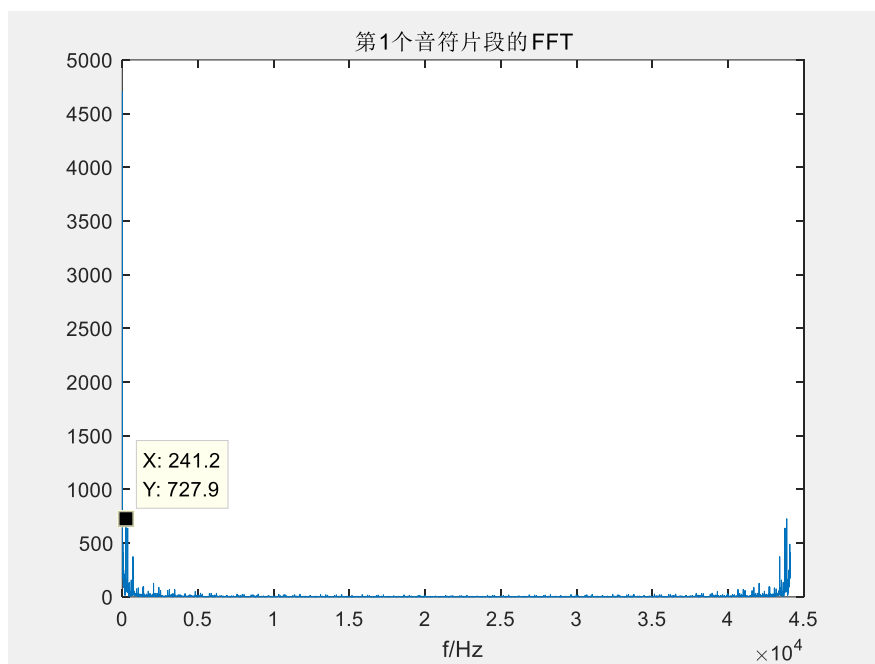
在音乐起始时间的分析中，相对于时域分析做自相关的方法，做短时傅里叶变换效率更高。

2) 音乐低声部曲谱分析

本题中备选项只有 6 个，因此可以直接分析某些特定片段包含的频域分量来确定音乐的低声部曲谱。比如对第一个音符对应的片段做 FFT。

```
y1=yd(idx(1):idx(2));  
y1_fft=abs(fft(yd(idx(1):idx(2))));  
f=[1:length(y1_fft)].*FS/length(y1_fft);  
plot(f,y1_fft);
```

做出频谱如下：



由上图分析可知该段音乐的频域分量最大值位于 241.2Hz 附近，考察 6 个频率序列发现第四个序列最为接近，则可以确定该音乐的曲谱为 base14。

2. 代码层面

常见的提升代码效率的方法大致有以下几种：

- 1) 用矩阵代替循环；
- 2) 预分配内存；

3) 使用函数句柄;

其中 1) 在本次实验中难以实现, 2) 在已经加以使用, 3) 可以使用但是对代码效率提升不大。

综上可发现, 往往更精确的解法会带来代码执行效率下降, 一般情况下我们要在这两者中取一个折中, 但在本题中效率影响很小, 我们可以采用精确求解的方法。同时利用已知的信息的特殊性, 比如只有 6 组备选频率序列等已知信息, 可以做出快速的选择。另外利用一些经典的函数和方法 (比如 STFT) 也可以达到效率和结果都比较理想的效果。

附加题 音符切分

给出一段音频 `audio.wav`, 请你用 `Matlab` 分析出每个音的时长 `tNote`, 描述实现思路。

【思路分析】

- 1) 根据之前问的经验, 音乐的音符变化的位置应该是波形幅度突增的位置, 因此本问的关键就是寻找信号的突增沿;
- 2) 本信号的突增沿表现得并不明显, 需要进行带通滤波、锐化、提取包络、中值滤波等处理, 才能更好地寻找信号峰值的位置;
- 3) 本信号为双声道信号, 因此为了更准确的划分, 将对信号的两列分别寻找划分位置, 然后将音符的时长做平均即可计算出每个音符的时长。

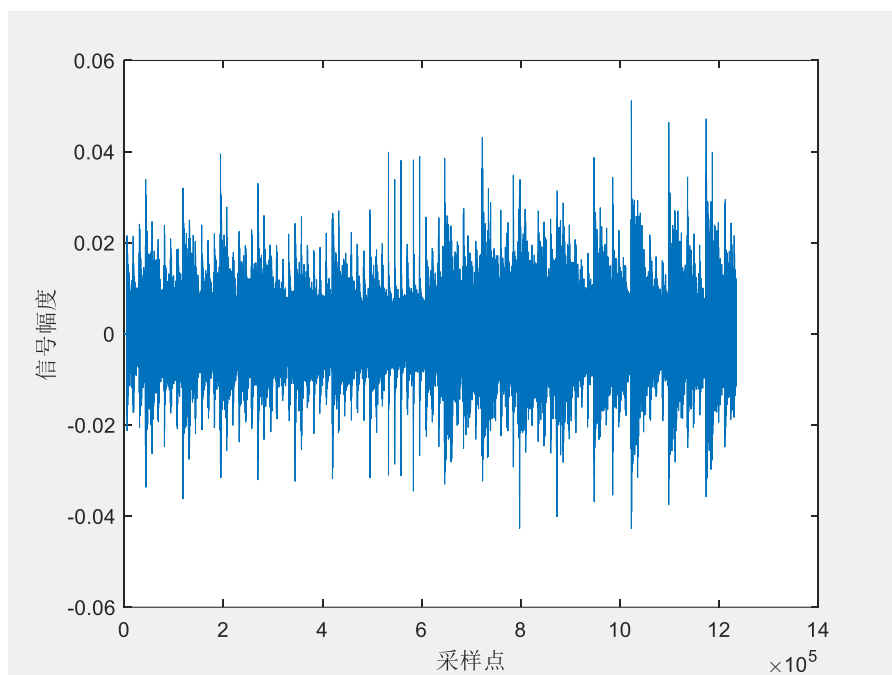
【关键代码与实现】

- 1) 采用巴特沃兹滤波器对信号做带通滤波, 滤除 100Hz 以下明显不属于音乐的低频成分和 2500Hz 以上的高频成分;

```
wp=[150 2000]/4000;  
ws=[100 2500]/4000;  
Rp=3; Rs=40;  
[n,wn]=buttord(wp,ws,Rp,Rs);  
[B,A]=butter(n,wn);  
y=filter(B,A,y);
```

- 2) 对信号做锐化, 其实现就是做两次差分, 这时便可以明显看出信号的突增沿;

```
y1=diff(diff(y));
```

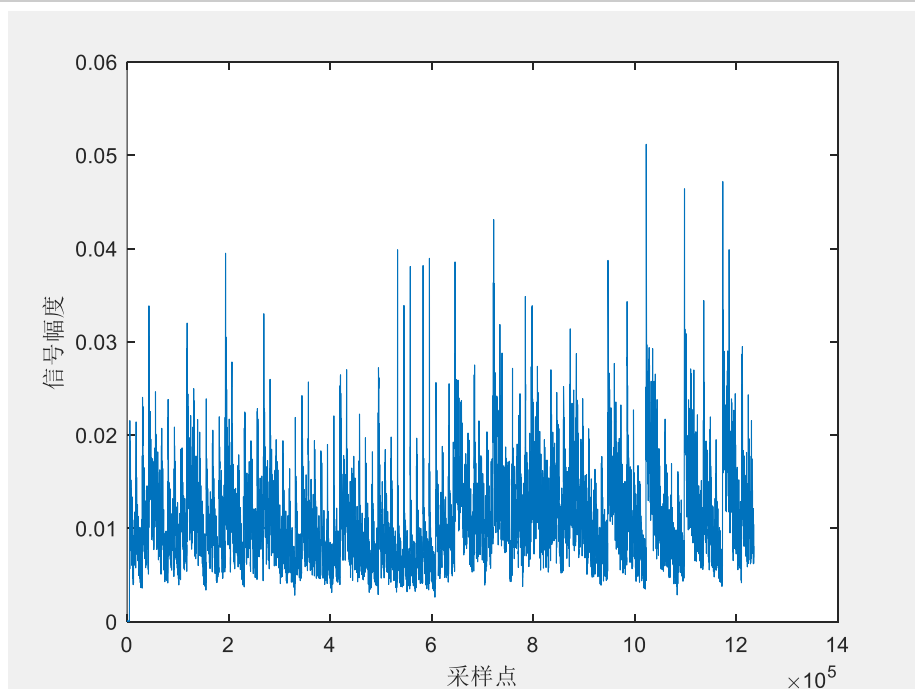


- 3) 提取信号包络，采用一个长度为 **40** 的窗在信号上滑动，以窗内最大值为包络上该点的取值；

```
for m=1:len
    z(m)=max(y1(max(1,m-step):min(len,m+step)));
end
```

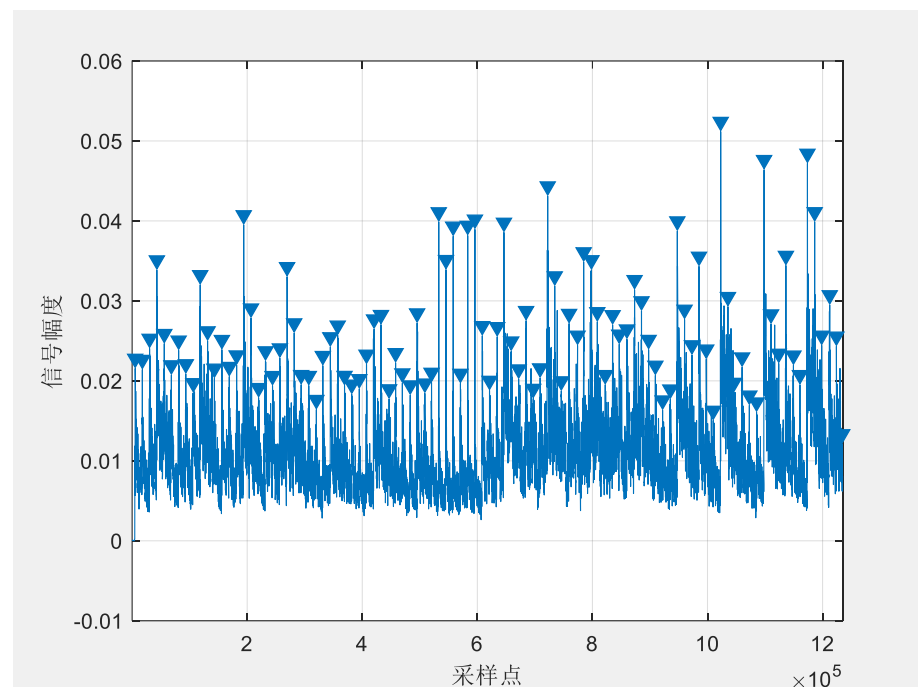
- 4) 对包络进行中值滤波处理，即可明显区分出音符的划分位置；

```
z=medfilt1(z,40);
```



5) 寻找包络的峰值, 采用第 15-75 个间隔求平均即可得到每个音符的时长。

```
[peaks,locs]=findpeaks(z,'minpeakdistance',10000);  
note_time=diff(locs);  
mean(note_time(15:75))/FS
```



【结果分析与评价】

通过观察提取出的包络峰值点的分布, 发现中间的间隔较为平均, 因此取第 15-75 个间隔的平均时长作为每个音符的时长。

	每个音的采样点数	每个音的时长
声道 1	12556	0.2847
声道 2	12564	0.2849

思考与体会

完成本次大作业我收获很大。本次大作业涉及到了基本的时域、频域 MATLAB 信号处理的常用知识, 通过完成本次大作业, 我对理论课上学到的数字信号处理基本理论的理解也更加深入。完成本次大作业时, 我在部分问题中也涉及到了自相关等常用的信号处理的方法, 体会到了其在信号分析中的强大之处。另外, 通过对代码效率的思考和探索, 我对各种方法的应用场景有了更深入的理解。最后感谢老师和助教的付出!

附录 代码文件清单

文件名称	对应功能
exp1.m	基本波形的生成
exp2.m	音乐合成
exp3_1.m	时域（自相关）分析音乐起始时间
exp3_2.m	频域（STFT）分析音乐起始时间
exp3_3.m	音乐低声部曲谱分析
exp4.m	音符划分
start_time.mat	音乐的起始时间