

# DSP 实验-MATLAB

电子信息与通信学院 鄢舒  
email: yan0shu@gmail.com

# 实验任务

## 语音信号的数字滤波

- 对语音信号进行频谱分析，找出干扰分量的频谱。
- 设计数字滤波器滤除文件中语音信号中的干扰分量，使音频文件听起来自然。
- 原始文件：SunshineSquare.wav

# 需要提交的内容

1. 滤波后的wav文件

2. matlab代码（m文件，要求有注释）

3. 实验报告

- 统一的实验报告封面（另有word文档）。
- 实验项目名称、内容（时域和频域的滤波效果）。
- 简短的实验结果分析。
- 一两句话写出结论。

# 提交形式

只交电子版

文档命名方式：“班号-学号-姓名”

提交时间期限：实验结束后一周内

提交途径：

- 个人发给→各班学习委员
- 学习委员再压缩打包发给我→yan0shu@gmail.com

# 实验方法

## ■ 分两个步骤去除干扰分量

1. 噪声是什么样的??

- 确定干扰分量的频率（频谱分析）



2. 滤掉这些频率

- 设计数字滤波器

# 确定噪声频率

计算幅度谱

- FFT

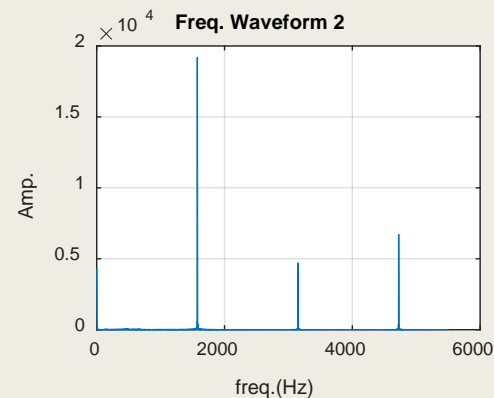
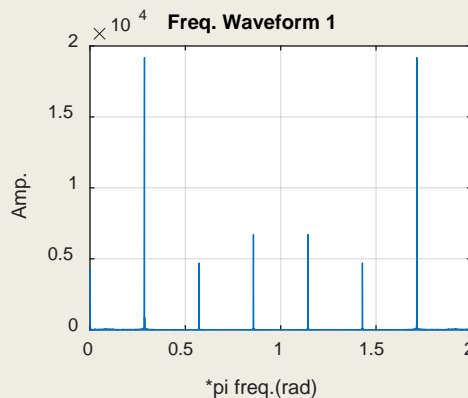
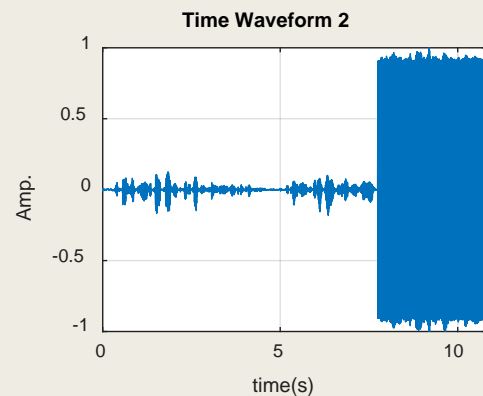
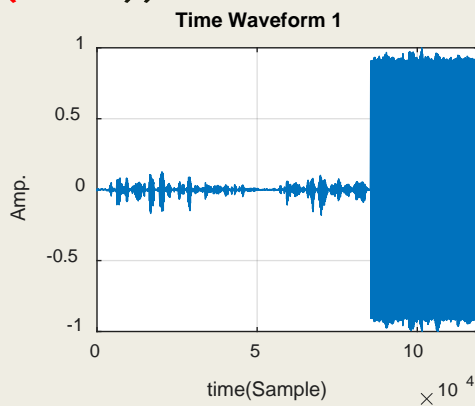


从幅度谱中分析  
出噪声频率

- 幅度谱最大的位置

# 读文件，计算幅度谱

- `[xx, fs] = audioread('SunshineSquare.wav');`
- `N = length(xx);`
- `figure(1);`
- `xxSpectrum=abs(fft(xx,N));`
- `plot(xxSpectrum);`



# 注意

- N是FFT的点数，应当是2的幂次，而且应该足够大
- N需要你自已确定是多少，可尝试不同的N，看看幅度响应的不同。
- 因噪声加在音频文件的最后，所以幅度响应只需要查看后面时段的即可
- 时间轴
  - $n = 0:(N-1);$
  - $t = n/fs;$
- 频率轴
  - $w = n*2*pi/N;$
  - $f = n*Fs/N;$



# 确定噪声频率

- 通过画出幅度响应来确定有几个幅度响应较大的位置。
- 用`max()`函数找到幅度响应最大的位置，即噪声所在的频率，注意可能有多个噪声频率。
- 还需要把频率从Hz转换为数字频率，后面要用。

## 方法二：利用声谱图确定噪声频率

- `[xx, fs] = audioread('SunshineSquare');`

- `xx = xx';`

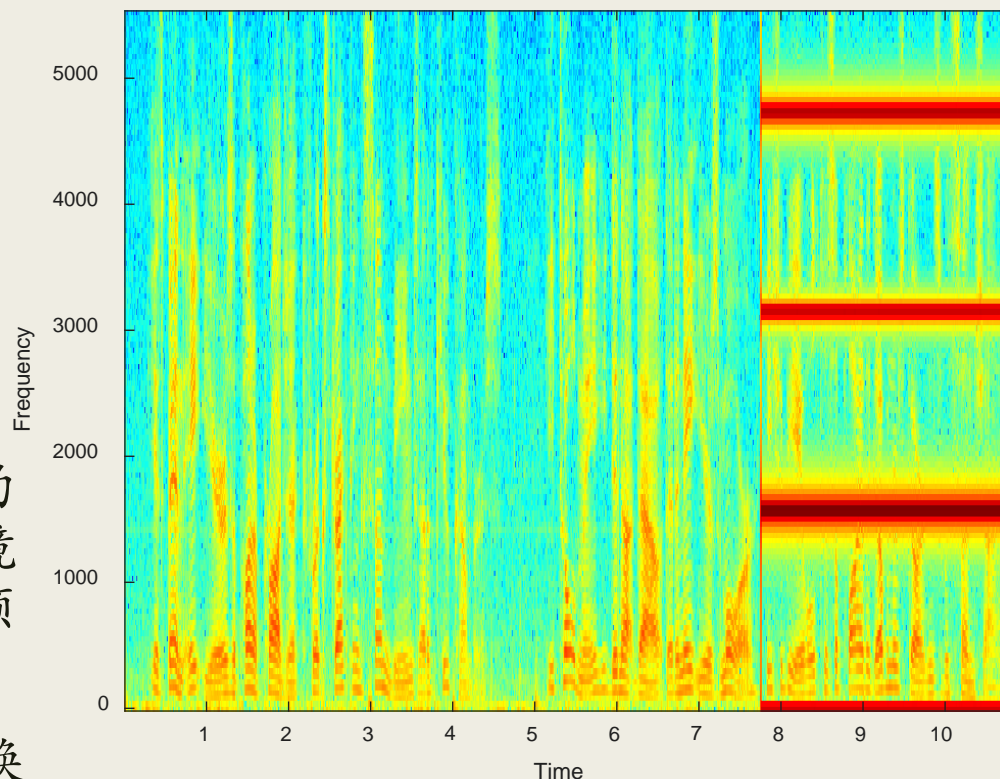
- `figure(1)`

- `specgram(xx, [], fs)`

- 注意：fs是wav文件的采样率。

- 根据声谱图估计声调的频率，可以借助放大镜工具找到更为准确的频率。

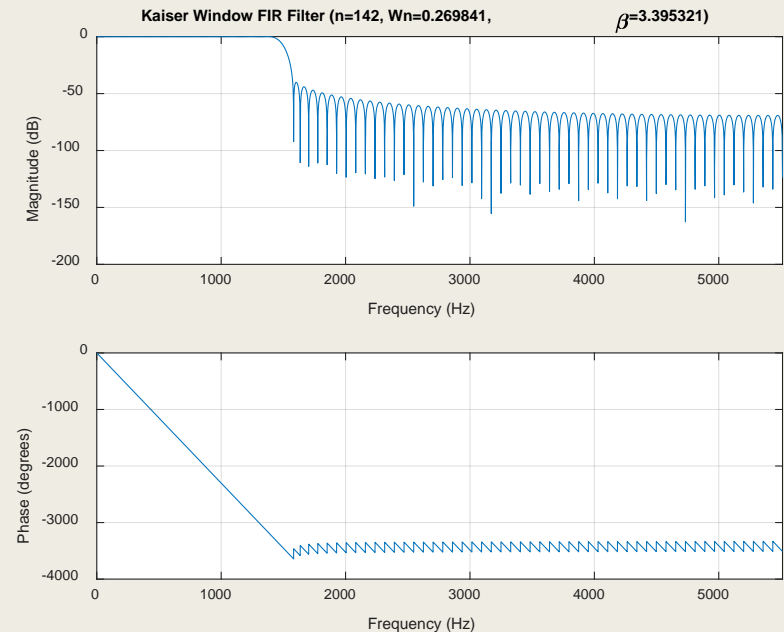
- 也需要把频率从Hz转换为数字频率。



```
spectrogram(xx,hann(256),250,256,fs,'yaxis');
```

# 低通滤波器设计举例

- 利用kaiser窗设计一个FIR低通滤波器
- `[filter_n,filter_Wn,bta,filtype] = kaiserord( [1400 1575], [1 0], [0.01 0.1], Fs);`
- `filter_b = fir1(filter_n, filter_Wn, filtype, kaiser(filter_n+1,bta), 'noscale');`
- `filter_a = 1;`
- `figure;`
- `freqz(filter_b,filter_a,N,Fs);`



# 另一种滤波器设计

- 一个加权三点平均器（FIR滤波器）足以一次滤掉一个频率，给定如下的冲激响应：
- $h[n] = \{1, B, 1\}$
- 用B表示频率响应。找到B的值以去除不要的频率。只要找到了正确的值，就可用以下的代码一次去除一个频率：
- `hh = [1, B, 1];`
- `yy = filter(hh, 1, xx);`

# 原理

- $h[n] = \{1, B, 1\},$

$$\begin{aligned} |H(e^{j\omega})| &= |1 + Be^{-j\omega} + e^{-j2\omega}| \\ &= |e^{-j\omega}(e^{j\omega} + e^{-j\omega}) + Be^{-j\omega}| \\ &= |(2\cos \omega + B)e^{-j\omega}| = 0 \end{aligned}$$

- 所以  $B = -2\cos \omega$

- 你需要自己填入B的值。

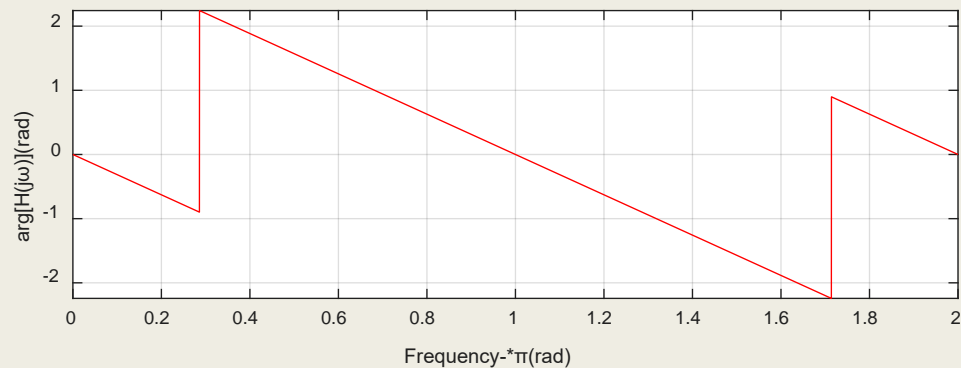
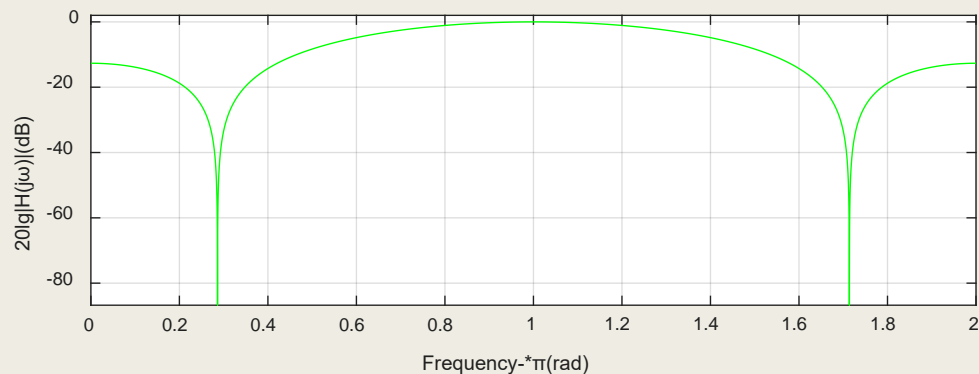
# 查看滤波器响应

- 可以用freqz查看你的滤波器的频率响应：
- `ww = 0:pi/100:2*pi;`
- `HH = freqz(hh, 1, ww);`
- `plot(ww,abs(HH));`
- `plot(ww,angle(HH));`

提示：

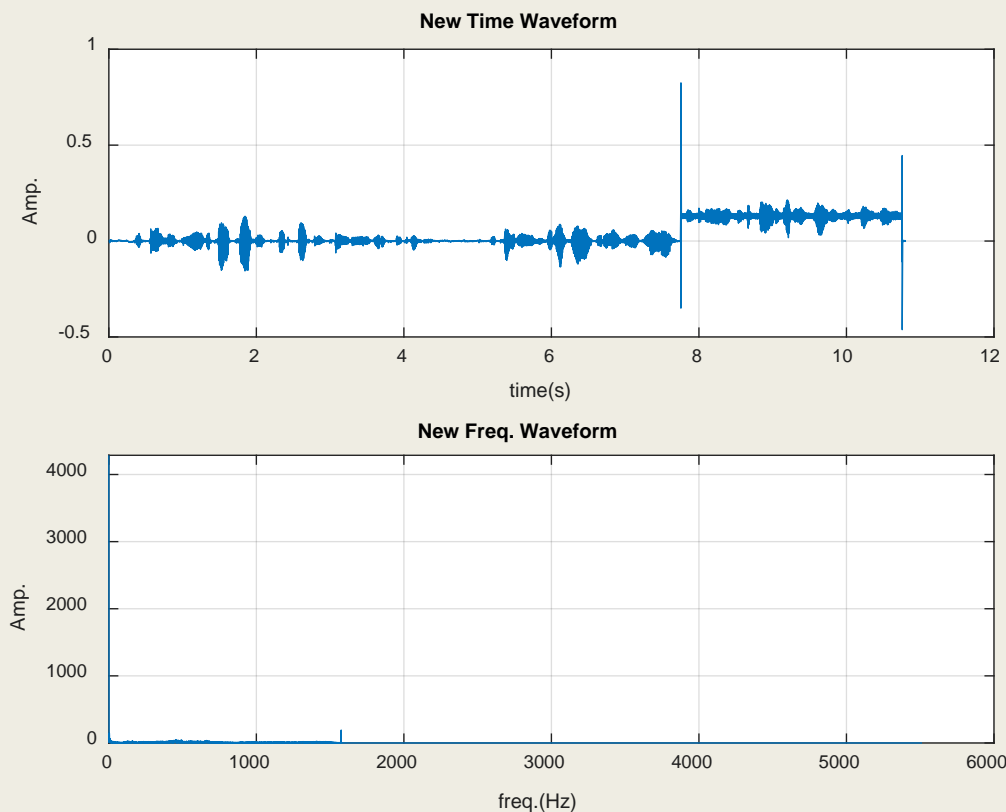
需要多个滤波器。

可以选择依次滤波，也可以用conv()把多个滤波器合成一个滤波器。



# 其他额外的处理（选做）

- 这些处理的结果并不完美，可以经过你自己的观察和思考，加入额外的处理，使效果更完美。
- 这部分如果做了，请在报告中高亮指出，并且要说明是如何做的，这么做的依据是什么，理论依据和实践依据都可以。



# 其他绘图函数

- (1) `plot(xn,'g');`     `stem(xn,'r')`
- (2) `grid;`     `axis([0,100,-1,1])`
- (3) `xlabel('x(n));`     `ylabel('X(k)')`
- (4) `title('XXX figure');`
- (5) `figure(2); subplot(3,1,1);`
- (6) `clear;`     `close all;`