**华中科技大学《数字信号处理》Matlab实验报告一二**

**姓名:\_\_\_\_吴叶赛\_\_\_\_\_**

**学号:\_\_U201813405\_\_\_**

**班级:\_\_电磁1802班\_\_**

**联系电话:18727095060**

**《数字信号处理》实验1(4 hours)**

**2020年11月-12月**

**一、实验目的（实验一）**

**掌握matlab的基本运算对象单位、特有的赋值方式、掌握matlab的程序化编写（脚本、函数）、掌握数据图形综合分析**

**二、方框思考题（及代码）**

问题1：

现在写一条语句,xx用(b)中定义的方法，把xx的偶数索引的元素(即xx(2),xx(4) 等)的值替换为常数。使用向量替换，不要用循环。

解：

% xx = [ zeros(1,3), linspace(0,1,5), ones(1,4) ] 这是输入数组

xx(2:2:end) = pi^pi

问题2：

解释为什么需要写yy(k+6)。如果用yy(k)会发生什么？

解：

保证数组下标索引是合理的正值。

使用yy(k)会报错，提示 “下标索引必须为正整数类型或逻辑类型”

问题3：

解释为什么real(zz)的图形是一个正弦曲线。它的相位和幅度是多少？由所绘图形的时移量计算相位。

解：

zz = 1.4 \* exp(j\*pi/2)\*exp(j\*5\*pi\*tt) = 1.4\*j \* ( cos(5\*pi\*t) + j\*sin(5\*pi\*t) ) = 1.4\*j\* cos(5\*pi\*t) - 1.4\*sin(5\*pi\*t)

Real(zz) = -1.4\*sin(5\*pi\*t) = 1.4\*cos(5\*pi\*t + pi/2)

则幅值A = 1.4 相位φ = pi/2

问题4：

绘制以上两个信号在-T≤t≤T的图形。为了输出下面（d）的要求，使用subplot(3,1,1)和subplot(3,1,2)来使图在一个图形窗口内，详见helpwin subplot

一般用法是：

subplot(2,1,2);

plot(xx);

（c）创建第3个正弦信号为 x3(t) = x1(t) + x2(t). 在Matlab中，这表示把两个正弦向量中的值对应相加。绘制 x3(t) 在-T≤t≤T的图形，使用subplot(3,1,3)绘图。

（d）对每个图都添加一个图名，图名都要包含你的姓名，用title函数

解：

clear all;close all;

A1 = 21; %年龄

A2 = 1.2 \* A1;

M = 12;%月

D = 3;%日

T = 1/4000;

tm1 = (37.2/M)\*T; %时延1

tm2 = -(41.3/D)\*T;%时延2

%步骤（a）

tt = -T : 0.000001 : T;

%步骤（b）

x1 = A1\*cos(2\*pi\*4000\*(tt-tm1));

x2 = A2\*cos(2\*pi\*4000\*(tt-tm2));

subplot(3,1,1);plot(tt,x1);grid on

title('吴叶赛1号')

xlabel( 'TIME (sec)')

subplot(3,1,2);plot(tt,x2);grid on

title('吴叶赛2号')

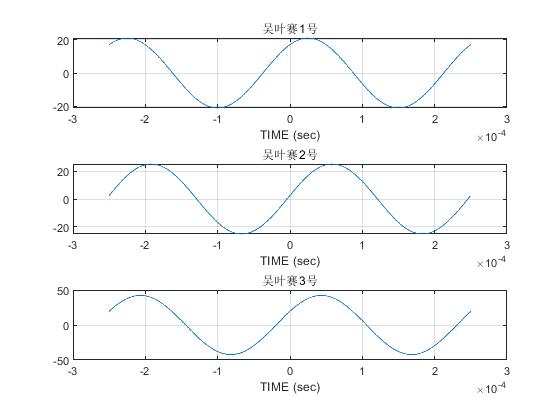
xlabel( 'TIME (sec)')

%步骤（c）

x3 = x1 + x2;subplot(3,1,3);plot(tt,x3);grid on

title('吴叶赛3号')

xlabel( 'TIME (sec)')



问题5：

用这种向量化的方法编写2－3行代码完成以下Matlab代码，不使用循环。（注：当xx是向量时，xx\*xx和xx.\*xx是不同的）

解：

N = 20;

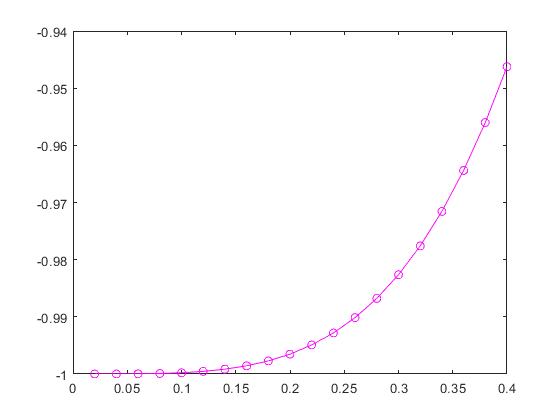
k = 1:N;

xk = k/50;

rk(k) = sqrt( xk(k).\*xk(k) + 2.25 );

sig(k) = exp( j\*2\*pi\*rk(k) );

plot( xk, real(sig), 'mo-')



问题6：

写一个可以生成单一正弦信号x(t) = Acos(ωt+φ)的函数，使用4个输入参量：幅度A，频率，相位和时长dur。函数应当返回两个输出参量：正弦信号的值x和对应的时间t。确保函数生成的正弦信号在每个周期有20个值，函数名为one\_cos()。提示：可借鉴上面的goodcos()函数。

绘制你们的one\_cos()函数，参数选为：A=95，=200弧度／秒，=/5弧度，时长为0.025秒。推导所绘图形的周期和相位是否正确。如果周期以毫秒为单位是多少？

解：

function [xx,tt] = one\_cos(A,w,angle,dur)

tt = 0:1/(20\*(w/(2\*pi))):dur;

xx = A\*cos(w\*tt + angle);

subplot(2,1,1);

plot(tt,xx);

grid on

title('默认单位')

xlabel('TIME(sec)')

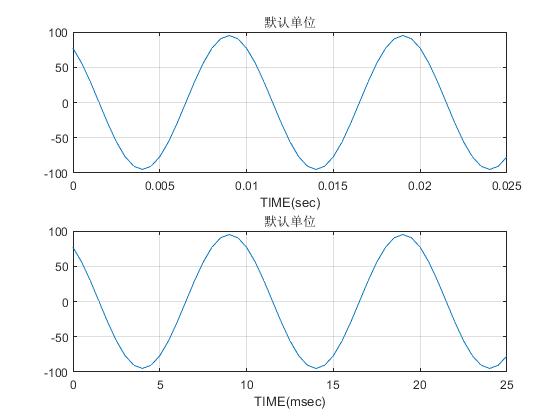
subplot(2,1,2);

plot(tt\*1000,xx);

grid on

title('默认单位')

xlabel('TIME(msec)')



命令行输入：one\_cos(95,200\*pi,pi/5,0.025)

观察图像，发现相邻相同值之间距离相差0.01s，满足T = 2\*pi/w;初始 相位为pi/5

问题7：

再合成一个chirp信号，使用如下参数：

1.总时长为3秒，采样率为fs=11025Hz

2.频率起始于3000Hz，终止于-2000Hz（负频率）

听一听信号。频率是怎么变化的？

显示这个chirp信号的声谱图。

使用频谱理论（正频率成分和负频率成分）解释你听到的声音和看到的声谱图。

解：

function [ xx,tt ] = mychirp( f1, f2, dur, fsamp )

tt = 0:1/fsamp:dur;

beta = (f2-f1)/dur;%f2是终止频率，f1是起始频率

xx = cos( 2\*pi \* ( beta\*(tt.^2)/2 + f1.\*tt ) );

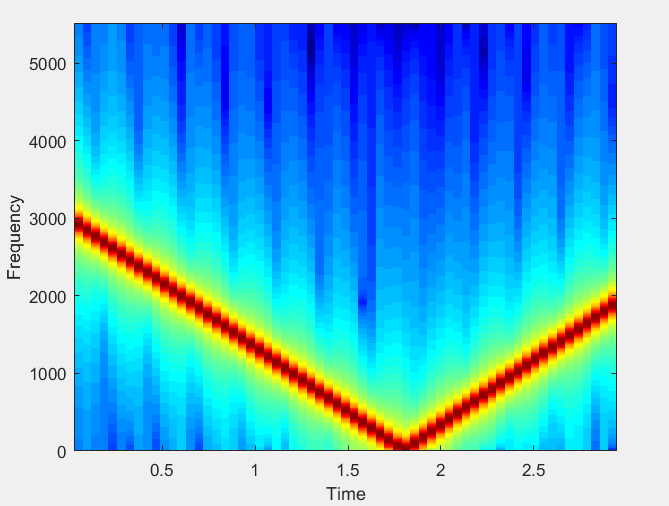
soundsc(xx,fsamp)

specgram(xx,1024,fsamp);

end

命令窗口：

mychirp(3000,-2000,3,11025)

通过听信号，发现声音起始很大，然后减小，小到一定值后后开始增大，因此判断频率是先减小后增大

f(t) = 3000-5000\*t/3 ----> w(t) = 3000t - 2500\*t^2/3

短时傅里叶变换 X（t，w）=∫w(t-τ)x(τ)e^−jωτdτ

频谱*Px*​(*t*,*f*)=∣*X*(*t*,*f*)∣^2=∣​∫−∞∞​*w*(*t*−*τ*)*x*(*τ*)*e^*−*j*2*πfτdτ*∣2

因为是绝对值，所以声谱会把负频率翻转，形成上述图形，同时函数为下凹函数，取样起始点在对称轴左侧，所以听到的声音会先变小在变大

**三、结论与收获（实验一）**

收获：

基本的收获是知道了matlab的各种语法和运算规则，掌握了matlab形式的程序编写规则，基本掌握了matlab的使用；

实验中遇到的难题：

最主要的可能是如何掌握新遇到的函数的使用方法。我的方法是看函数方法的源代码，虽然麻烦点，但是看懂了就掌握了

**四、实验结果验收**

由主讲教师在实验课当场验收，并记录实验成绩。

**《数字信号处理》实验2(4 hours)**

**一、实验项目：**语音信号的数字滤波

**二、实验目的：**

1、掌握使用FFT进行信号谱分析的方法

2、设计数字滤波器对指定的语音信号进行滤波处理

**三、实验内容**

（一）、设计数字滤波器滤除语音信号中的干扰（4 学时）

1、使用Matlab的fft函数对语音信号进行频谱分析，找出干扰信号的频谱；

2、用两种方法设计数字滤波器滤除语音信号中的干扰分量，并进行播放对比滤波器性能及进行结果分析。

实验平台：MATLAB2020b

实验方法一：极零点向量法

实验代码：

clear ;close all;

[y,Fs]=audioread('SunshineSquare.wav');

sigLength=length(y);

Y = fft(y);

t=(0:sigLength-1)/Fs;

figure(1);plot(t,y);xlabel('Time(s)');%做出时域分析图

l=0:sigLength-1;

w=(pi/floor(sigLength/2))\*l;

figure(2);plot(w/pi,abs(Y));xlabel('frequency')%做出离散傅里叶变换图

%滤除0.2857和0.8571

f1=0.2857\*pi;f2=0.8571\*pi;f=0.99;a=[1 f f^2 f^3 f^4];

b1=[1 -2\*(cos(f1)+cos(f2)) 2+4\*cos(f1)\*cos(f2) -2\*(cos(f1)+cos(f2)) 1];

a1=b1.\*a;

[H,W0]=freqz(b1,a1,w);

figure(4);plot(w/pi,abs(H));

%滤除0和0.5714

f3=0;f4=0.5714\*pi;

b2=[1 -2\*(cos(f3)+cos(f4)) 2+4\*cos(f3)\*cos(f4) -2\*(cos(f3)+cos(f4)) 1];

a2=b2.\*a;

[H2,W1]=freqz(b2,a2,w);

figure(5);plot(w/pi,abs(H2));

Ydtft=Y.'.\*H.\*H.\*H2.\*H2;%滤两遍减轻噪声干扰

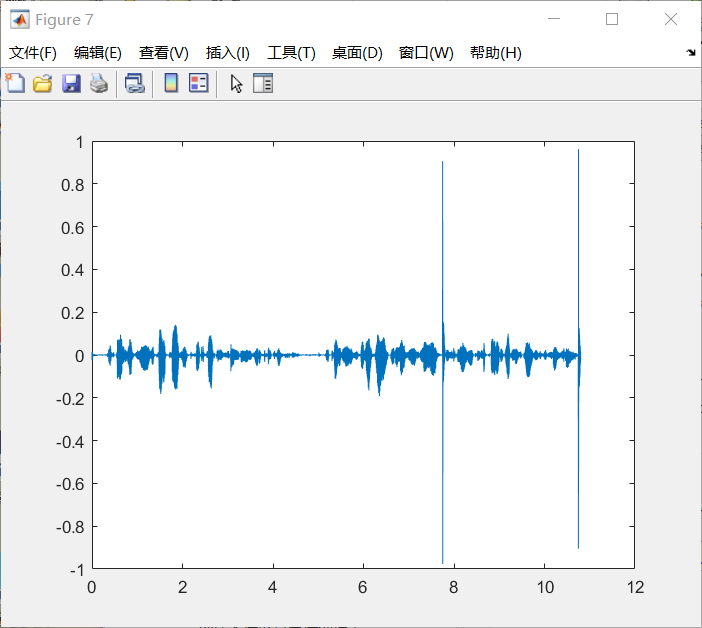
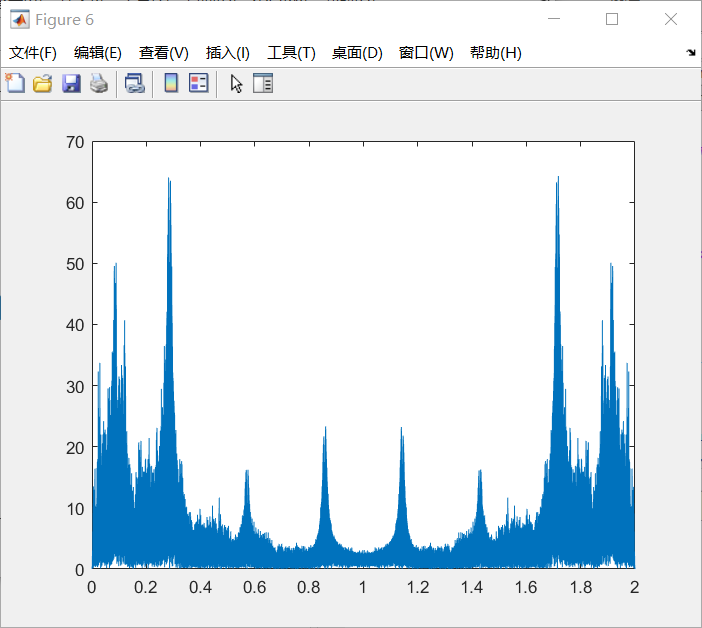
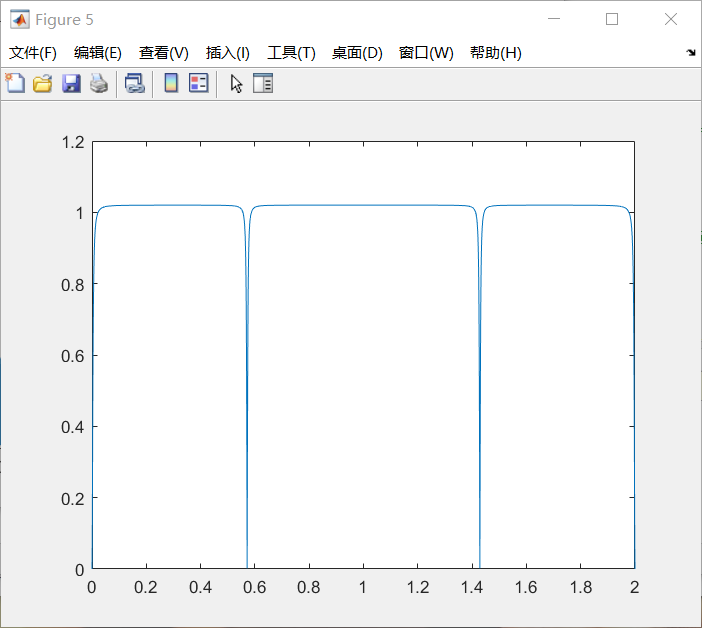
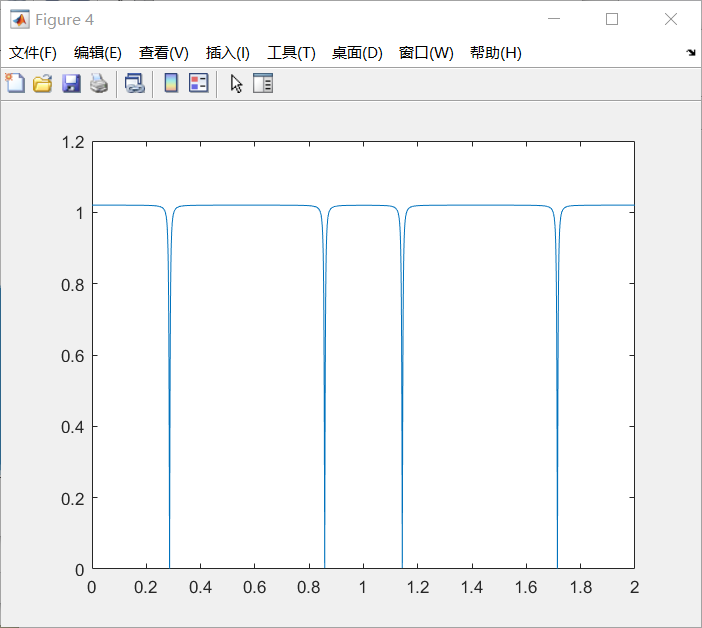
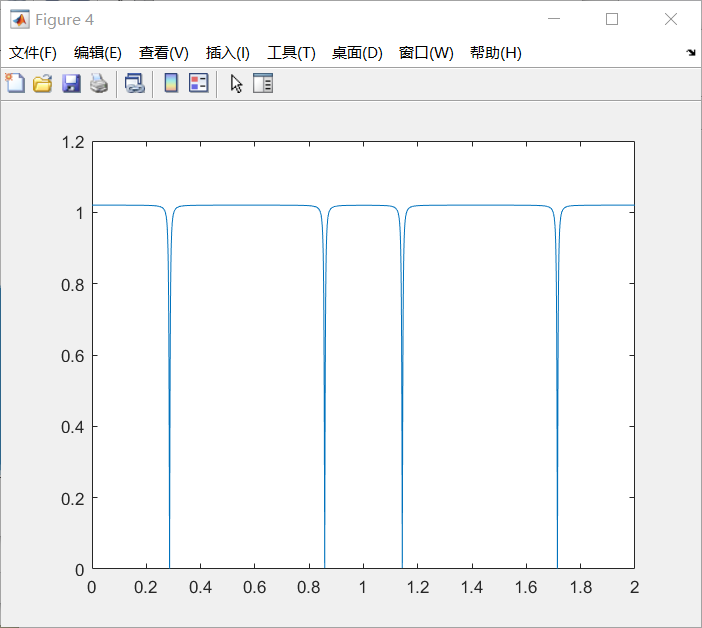
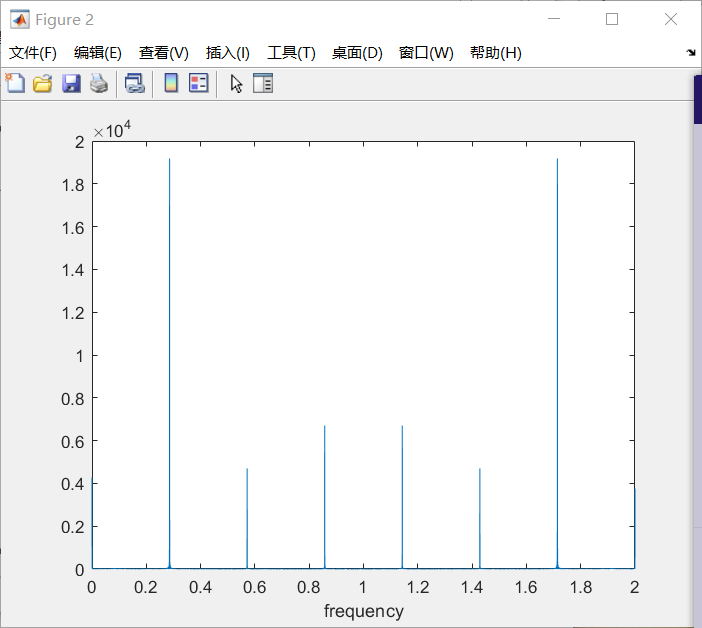
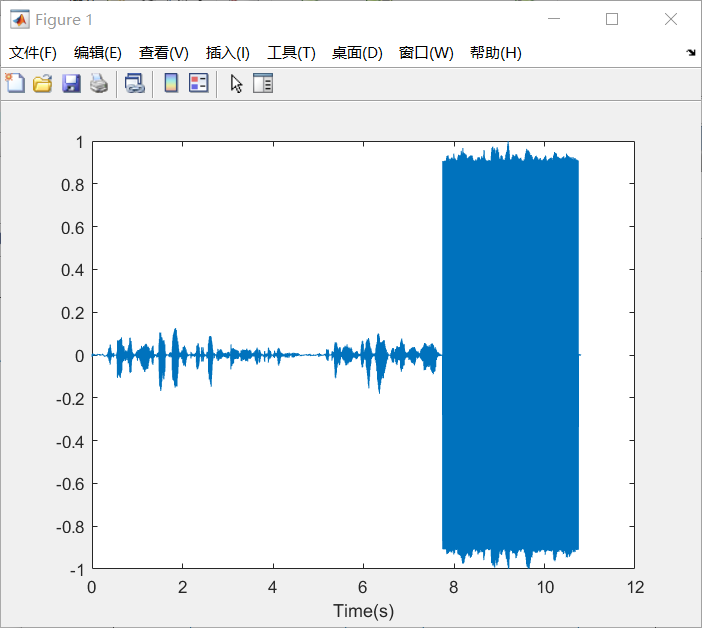
figure(6);plot(w/pi,abs(Ydtft));

y2=ifft(Ydtft);

figure(7);plot(t,real(y2));

soundsc(real(y2),Fs);

audiowrite('SunshineSquare\_New.wav',real(y2),Fs);



实验方法二：零点向量法

实验代码：

clear ;close all;

%读取音频文件

[xn,fs]=audioread('SunshineSquare.wav');

%原始数据的频谱

figure(1);

subplot(1,2,1);

N=round(length(xn)/2);

Y=fft(xn);

Y=Y(1:N);

plot((0:N-1).\*fs/length(xn),abs(Y));

title('原始频谱');

xlabel('频率f/Hz');ylabel('|Xk|');

%原始数据的时域图

subplot(1,2,2);

plot((0:length(xn)-1),xn);

title('初始时域图');

xlabel('n');ylabel('xn');

%第一次滤波

[y\_max1,n\_max1]=max(abs(fft(xn)));

A1=-2\*cos(2\*pi\*(n\_max1-1)/length(xn));

hn1=[1,A1,1];

yn1=conv(xn,hn1);

%第一次滤波的频谱

figure(2);

subplot(1,2,1);

N=round(length(yn1)/2);

Y=fft(yn1);

Y=Y(1:N);

plot((0:N-1).\*fs/length(yn1),abs(Y));

title('第一次滤波的频谱');

xlabel('频率f/Hz');ylabel('|Yk1|');

%第一次滤波的时域图

subplot(1,2,2);

plot((0:length(yn1)-1),yn1);

title('第一次滤波的时域图');

xlabel('n');ylabel('yn1');

%第二次滤波

[y\_max2,n\_max2]=max(abs(fft(yn1)));

A2=-2\*cos(2\*pi\*(n\_max2-1)/length(yn1));

hn2=[1,A2,1];

yn2=conv(yn1,hn2);

%第二次滤波的频谱

figure(3);

subplot(1,2,1);

N=round(length(yn2)/2);

Y=fft(yn2);

Y=Y(1:N);

plot((0:N-1).\*fs/length(yn2),abs(Y));

title('第二次滤波后的频谱');

xlabel('频率f/Hz');ylabel('|Yk2|');

%第二次滤波的时域图

subplot(1,2,2);

plot((0:length(yn2)-1),yn2);

title('第二次滤波的时域图');

xlabel('n');ylabel('yn2');

%第三次滤波

[y\_max3,n\_max3]=max(abs(fft(yn2)));

A3=-2\*cos(2\*pi\*(n\_max3-1)/length(yn2));

hn3=[1,A3,1];

yn3=conv(yn2,hn3);

%第三次滤波的频谱

figure(4);

subplot(1,2,1);

N=round(length(yn3)/2);

Y=fft(yn3);

Y=Y(1:N);

plot((0:N-1).\*fs/length(yn3),abs(Y));

title('第三次滤波后的频谱');

xlabel('频率f/Hz');ylabel('|Yk3|');

%第三次滤波的时域图

subplot(1,2,2);

plot((0:length(yn3)-1),yn3);

title('第三次滤波的时域图');

xlabel('n');ylabel('yn3');

%第四次滤波

[y\_max4,n\_max4]=max(abs(fft(yn3)));

A4=-2\*cos(2\*pi\*(n\_max4-1)/length(yn3));

hn4=[1,A4,1];

yn4=conv(yn3,hn4);

%第四次滤波的频谱

figure(5);

subplot(1,2,1);

N=round(length(yn4)/2);

Y=fft(yn4);

Y=Y(1:N);

plot((0:N-1).\*fs/length(yn4),abs(Y));

title('第四次滤波后的频谱');

xlabel('频率f/Hz');ylabel('|Yk4|');

%第四次滤波的时域图

subplot(1,2,2);

plot((0:length(yn4)-1),yn4);

title('第四次滤波后的时域图');

xlabel('n');ylabel('yn4');

%时域滤波

yn5=yn4;

for n=1:length(yn4)

if abs(yn4(n))>0.2

yn5(n)=0;

end

end

%时域滤波后的频谱

figure(6);

subplot(1,2,1);

N=round(length(yn5)/2);

Y=fft(yn5);

Y=Y(1:N);

plot((0:N-1).\*fs/length(yn5),abs(Y));

title('时域滤波后的频谱');

xlabel('频率f/Hz');ylabel('|Yk5|');

%时域滤波后的时域图

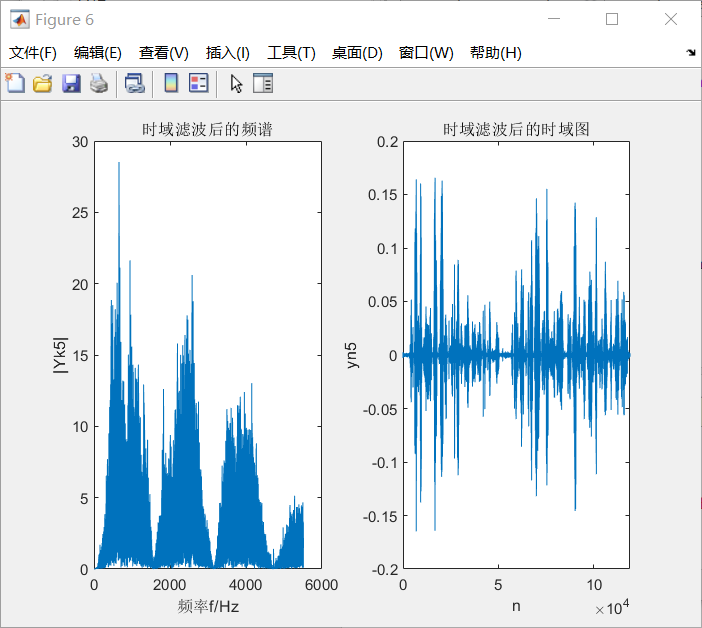
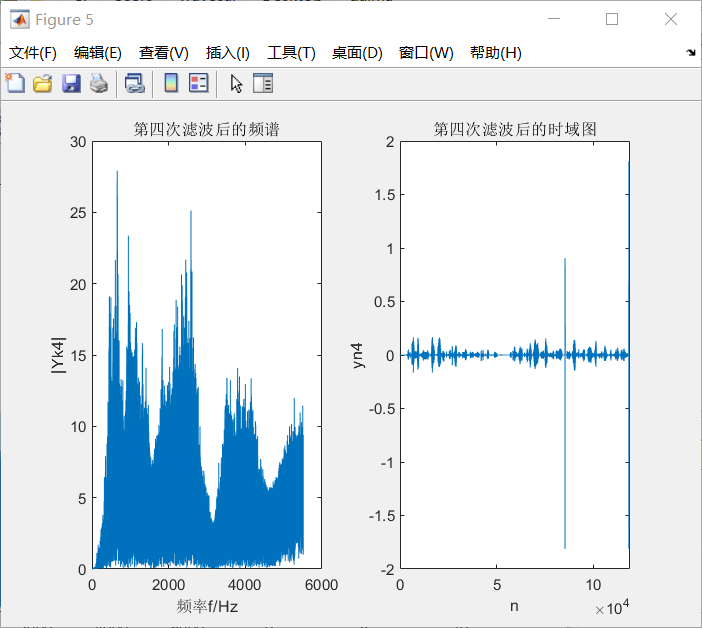
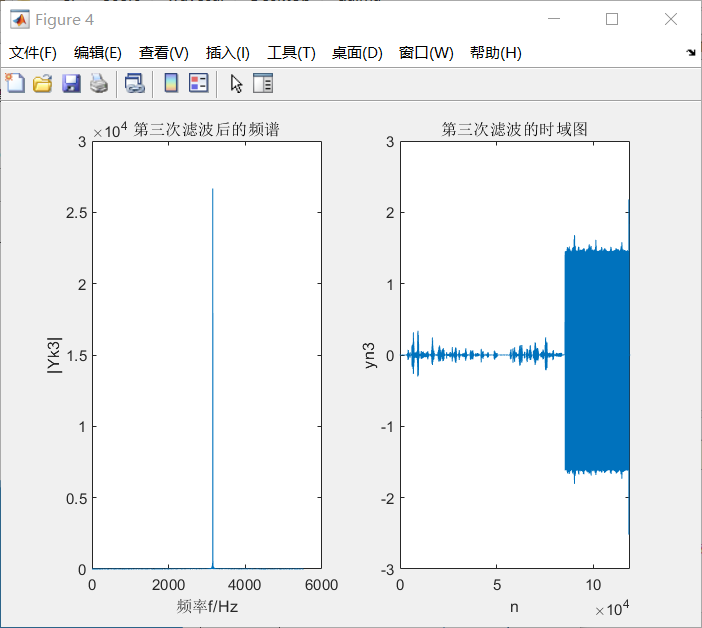
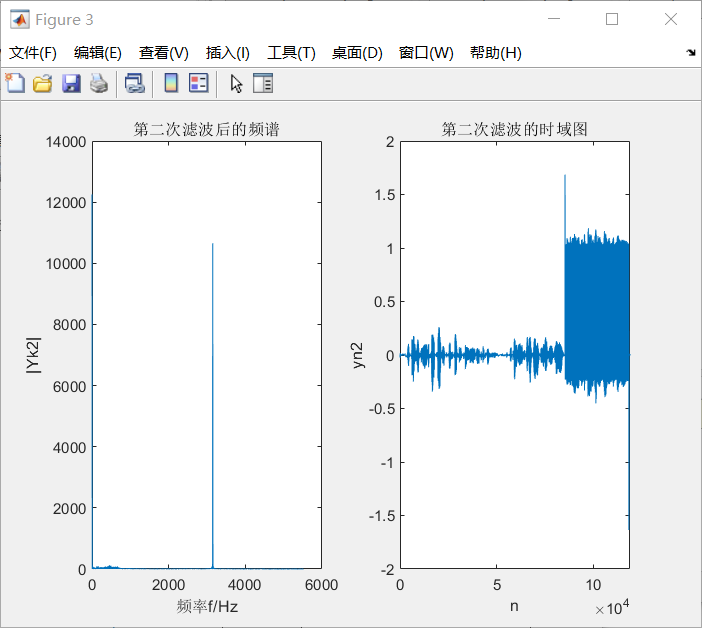
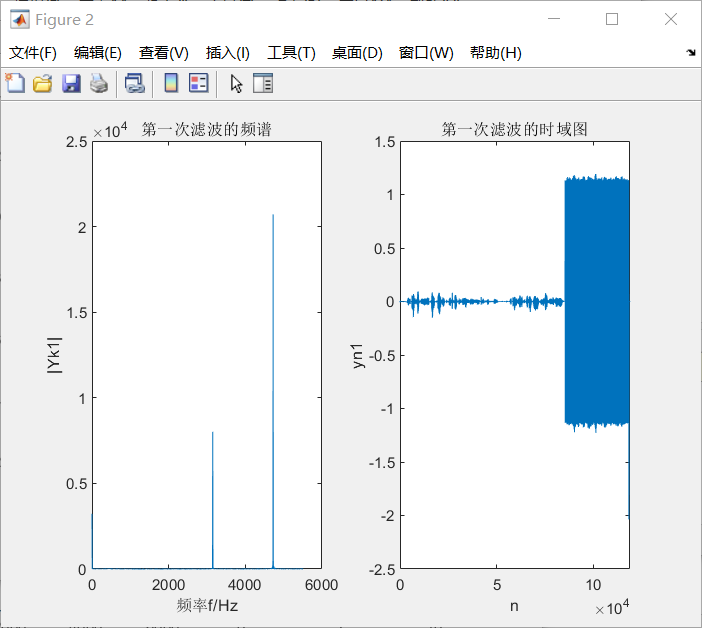
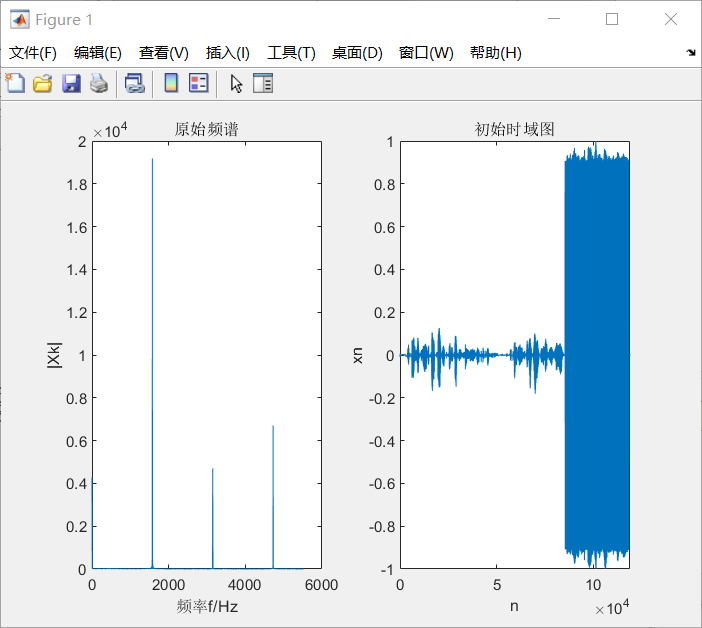
subplot(1,2,2);

plot((0:length(yn5)-1),yn5);

title('时域滤波后的时域图');

xlabel('n');ylabel('yn5');

soundsc(yn5);udiowrite('New\_SunshineSquare.wav',yn5,fs);



**四、实验平台**

1、Matlab 8.0以上；

2、主要相关函数介绍：

（1）fft(xn)

（2）[H,W]=freqz(b,a,ww) ； filter(b,a,xn)

（3）plot(xn,’g’) ； stem(xn,’r’)

（4）grid； axis([0,100,-1,1])； xlabel('x(n)')； ylabel('X(k)')

（5）title('XXX figure')；

（6）figure(2)； subplot(3,1,1)；

（7）clear； close all；

（8）abs(Xk)； ange(Xk)； max(xn)； min(xn)

（10）dB=20\*log10((mag+eps)/max(mag))；

（11）[audio\_data,fs]=wavread('SunshineSquare.wav')；

（12）wavwrite(audio\_data\_new,fs,'SunshineSquare\_New.wav')；

**六、实验结果验收（可选）**

由主讲教师在实验课当场验收，并记录实验成绩。