
信号处理实习报告

申报等级： B

MATLAB 版本： R2016a

姓 名： 徐晨辉

学 号： 201718026


班 级： 2017180

专 业： 信息与计算科学

签到次数：

承诺书

本人承诺,实习报告所有内容均由本人独立完成,如有学术不端,
一切后果自行承担。

承诺人: 

日期: 2020.6.12

目录

实验一.....	3
(1) 简述在时域求系统响应的方法.....	1
(2) 简述通过实验判断系统稳定性的方法，分析实验内容 (4) 的 1 的稳定输出的波形.....	1
(3) 对各实验结果进行简单分析和解释.....	1
(4) 简要回答思考题.....	1
(5) 程序清单和信号波形.....	1
实验二.....	6
(1) 运行程序，打印要求显示的图形.....	6
(2) 分析比较实验结果，简述由实验得到的主要结论.....	7
(3) 简要回答思考题.....	7
(4) 程序清单和有关曲线.....	7
实验三.....	10
(1) 完成各个实验任务和要求，附上程序清单和有关曲线.....	10
(2) 简要回答思考题.....	14
实验四.....	15
(1) 简述实验目的及原理.....	15
(2) 画出实验主程序框图，打印程序清单.....	15
(3) 绘制三个分离滤波器的损耗函数曲线.....	18
(4) 绘制经过滤波分离的三路抑制载波调幅小号的时域波形.....	18
(5) 简要回答思考题.....	18
实验五.....	19
(1) 对两种设计 FIR 滤波器的方法（窗函数法和等波纹最佳逼近法） 进行分析比较，简述其优缺点.....	19
(2) 附程序清单，打印实验内容要求绘图显示的曲线图.....	20
(3) 分析总结实验结果.....	23
(4) 简要回答思考题.....	23

实验一

(1) 简述在时域求系统响应的方法

答:

可以通过差分方程递推得到系统的响应,也可以先计算得到系统单位脉冲响应,然后与输入序列进行时域卷积即可。

(2) 简述通过实验判断系统稳定性的方法,分析实验内容 (4) 的 1 的稳定输出的波形

答:

1.可以通过输入 $u(n)$, 观察该序列的输出响应是否收敛,若收敛,则表明系统稳定。

2.分析输入信号为 $u(n)$ 时,其波形最终收敛到 0,由此知,该系统稳定。

(3) 对各实验结果进行简单分析和解释

答:

1.本题要求用三个程序实现题目要求,本文以差分方程 $y(n)=0.05x(n)+0.05x(n-1)+0.9y(n-1)$ 为例, $R(5)$ 作为输入序列进行求解. 图形绘制结果见(5)程序清单和信号波形。

2.第一小问,本文通过 `ones()`函数创建 $x1(n)$ 和 $x2(n)$, 用 `impz()`函数计算系统的单位脉冲响应,再使用 `conv()`函数将输入序列与系统单位进行卷积. 图形绘制结果见(5)程序清单和信号波形。

第二小问,本文直接使用 `impz()`函数计算系统的单位脉冲响应. 图形绘制结果见(5)程序清单和信号波形。

3. 本文使用 `conv()`函数将输入序列与系统单位进行卷积. 图形绘制结果见(5)程序清单和信号波形。

4.第一小问,本文通过输入系统 $u(n)$,观察其输出判断系统的稳定性,用 `impz()`函数计算系统的单位脉冲响应,再使用 `conv()`函数将输入序列与系统单位进行卷积. 图形绘制结果见(5)程序清单和信号波形。

第二小问,用 `impz()`函数计算系统的单位脉冲响应,再使用 `conv()`函数将输入序列与系统单位进行卷积. 图形绘制结果见(5)程序清单和信号波形.发现输出波形类似于 $\sin(0.4n)$,与系统谐振频率吻合。

(4) 简要回答思考题

答:

1.可以采用分段线性卷积的方法,如重叠保留法等

2.将信号高频滤去,则信号在时域变化减缓,在有阶跃的地方会产生过渡现象。

(5) 程序清单和信号波形

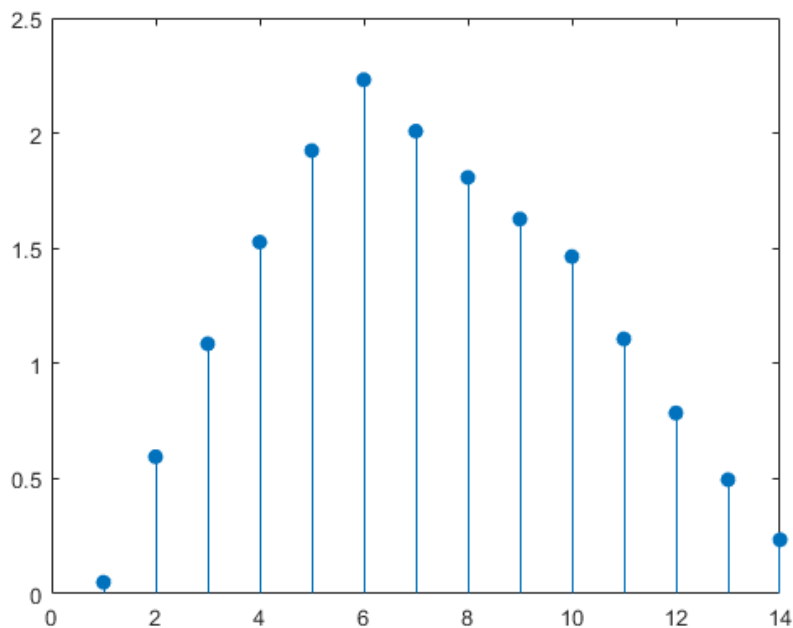
第一题:

```
function main_1()
```

```

h = Unit_pulse_1();% 系统单位脉冲响应序列
x1 = xt();% 输入序列
y1conv=conv(h,x1);% 进行卷积
stem(y1conv,'filled')% 绘制输入序列的波形
end
function xt = xt()
xt = ones(1,5);% 返回输入序列
end
function h = Unit_pulse_1()
% 以系统为差分方程  $y(n)=0.05x(n)+0.05x(n-1)+0.9y(n-1)$  为例
% 用 impz() 函数计算系统的单位脉冲响应
a1=[1,-0.9];% y[n] 系数
b1=[0.05 0.50];% x[n] 系数
[h] =impz(b1,a1,10);% 得到 10 个长度的单位脉冲响应
end
结果:

```



第二题第一小问:

```

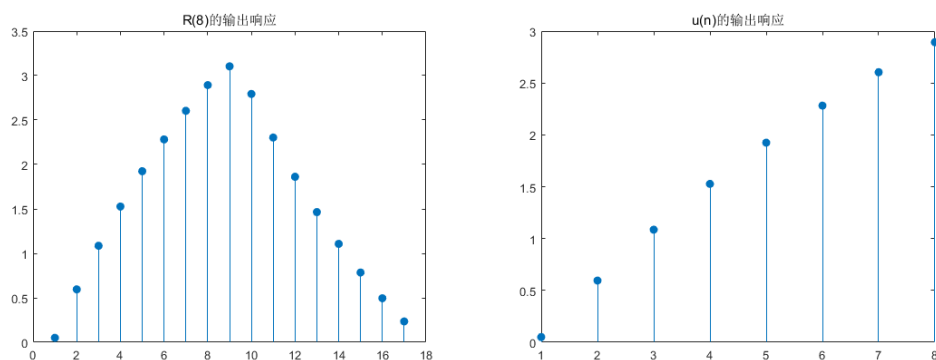
function main_2()
% 系统差分方程  $y(n)=0.05x(n)+0.05x(n-1)+0.9y(n-1)$ 
% 用 impz() 函数计算系统的单位脉冲响应
% 再使用 conv() 函数将输入序列与系统单位进行卷积
a1=[1,-0.9];% y[n] 系数
b1=[0.05 0.50];% x[n] 系数
[h] =impz(b1,a1,10);% 得到 10 个长度的单位脉冲响应
x1 = ones(1,8);% R(8)
x2 = ones(1,8)% 为研究方便, 截取 u(n) 的一部分
y1 = conv(h,x1);% R(8) 的输出响应

```

```

y2 = conv(h,x2);% u(n)的输出响应
y2 = y2(1:8)%但由于 conv 函数自动补 0,导致后边的结果被舍去
figure
stem(y1,'filled');
figure
stem(y2,'filled');
end

```

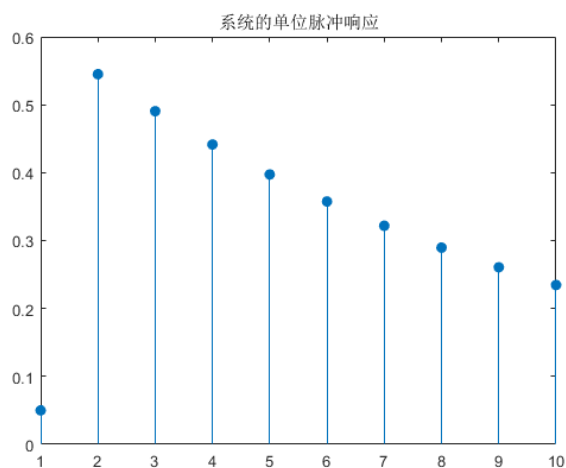


第二题第二小问

```

% 系统差分方程 y(n)=0.05x(n)+0.05x(n-1)+0.9y(n-1)
%用 impz()函数计算系统的单位脉冲响应
a1=[1,-0.9];% y[n]系数
b1=[0.05 0.50];%x[n]系数
[h] =impz(b1,a1,10);%得到 10 个长度的单位脉冲响应
figure
stem(h,'filled');

```



第三题:

```

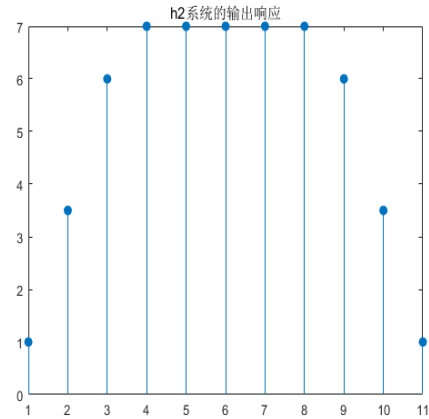
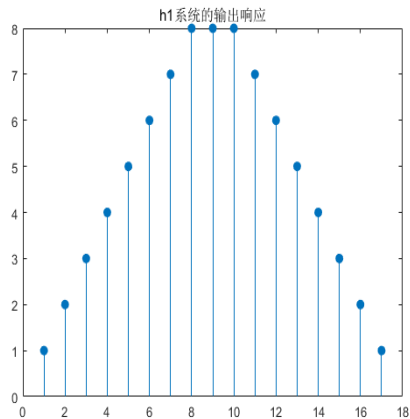
%使用 conv()函数将输入序列与系统单位进行卷积
a1=[1,-0.9];% y[n]系数
b1=[0.05 0.50];%x[n]系数
h1n = ones(1,10);
h2n = [1 2.5 2.5 1];

```

```

xn = ones(1,8);
y1n = conv(h1n,xn);
y2n = conv(h2n,xn);
figure
stem(y1n,'filled');
title('h1 系统的输出响应')
figure
stem(y2n,'filled');
title('h2 系统的输出响应')

```

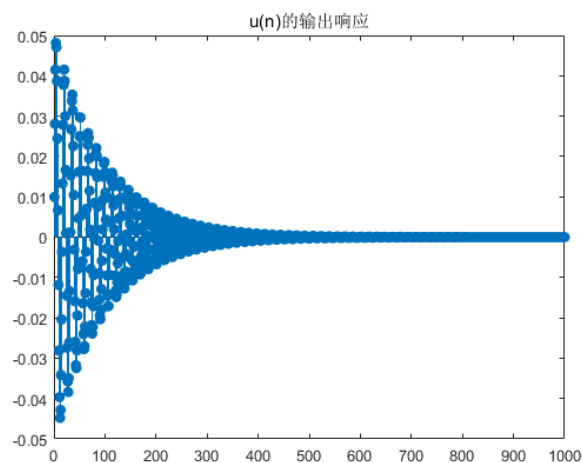


第四题第一小问：

```

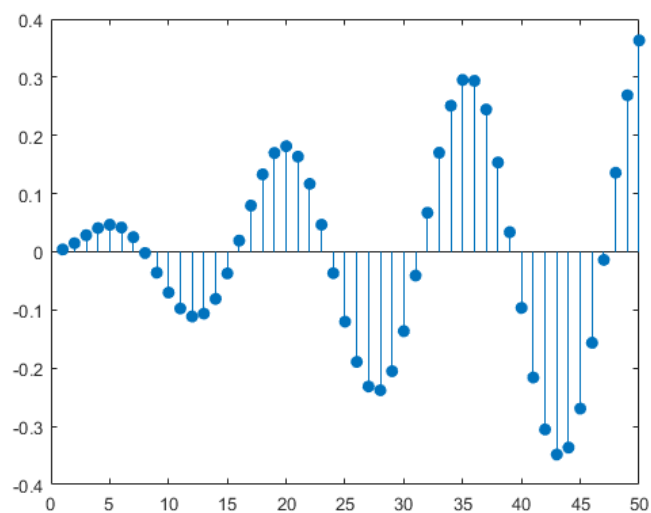
%用 impz()函数计算系统的单位脉冲响应
%使用 conv()函数将输入序列与系统单位进行卷积
a1=[1 -1.8237 0.9801];% y[n]系数
b1=[1/100.49 0 -1/100.49];%x[n]系数
h = impz(b1,a1,1000);
xn = ones(1,1000);%u(n)
yn = conv(h,xn);
figure
stem(yn(1:1000),'filled');
title('u(n)的输出响应')

```



第四题第二小问：

```
%用 impz()函数计算系统的单位脉冲响应
%使用 conv()函数将输入序列与系统单位进行卷积
a1=[1 -1.8237 0.9801];% y[n]系数
b1=[1/100.49 0 -1/100.49];% x[n]系数
h = impz(b1,a1,50);
xn1 = 1:50;
xn = sin(0.014*xn1)+sin(0.4*xn1);
yn = conv(h,xn);
figure
stem(yn(1:50),'filled');
```

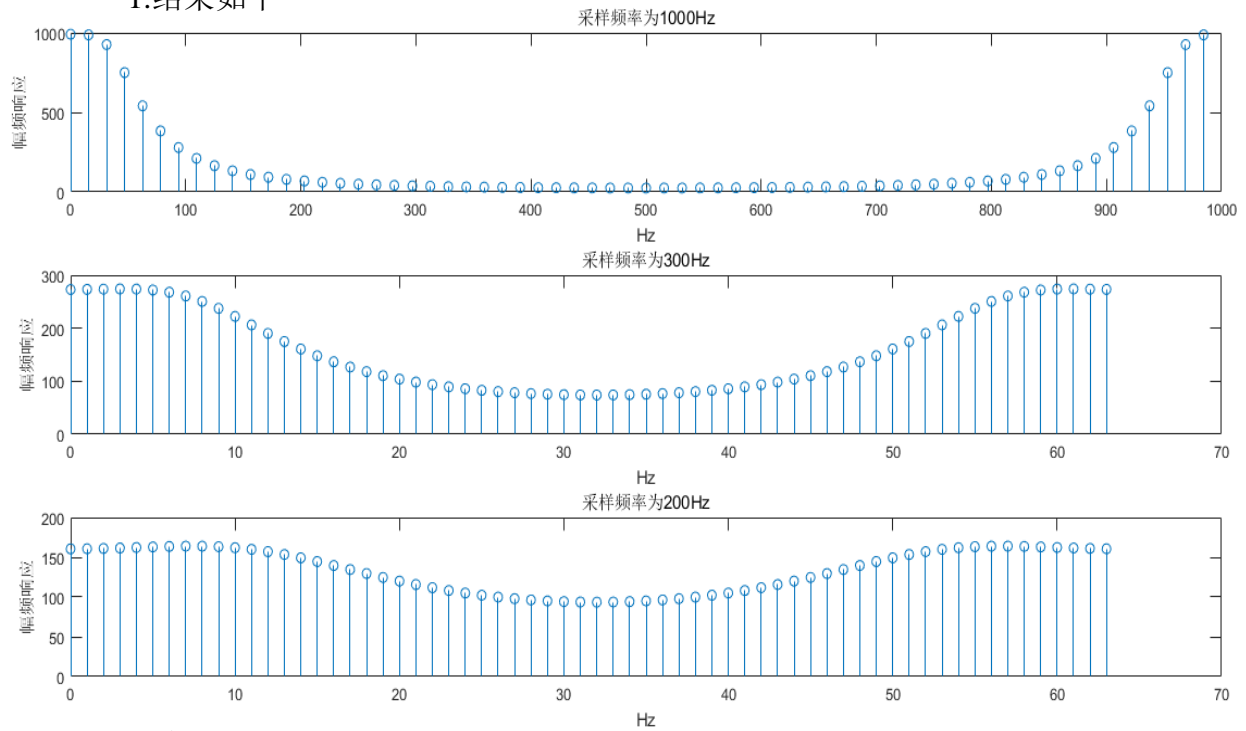


实验二

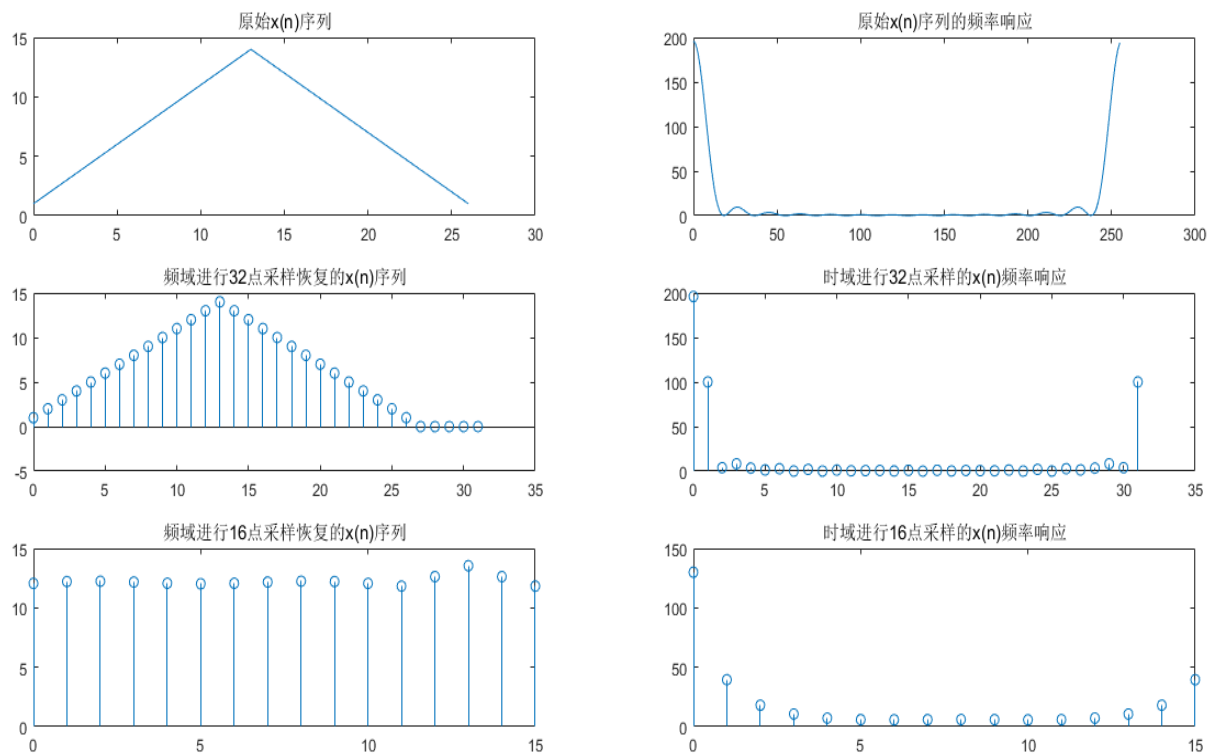
(1) 运行程序，打印要求显示的图形

答:

1.结果如下



2.结果如下



(2) 分析比较实验结果，简述由实验得到的主要结论

1.由第一张图片可知，采样序列的频谱是以采样频率为周期对模拟信号频谱进行的周期延拓。当采样频率较大，为 1000Hz 时频谱混叠很小；当采样频率降低为 300Hz 时，在折叠频率 150Hz 附近频谱混叠很严重；当采样频率为 200Hz 时，在折叠频率 110Hz 附近频谱混叠更加严重。因此在时域抽样时要满足奈奎斯特定理才能使频域不发生混叠。

2.由第二张图片可知，对信号序列 $x(n)$ 的频谱函数在 $[0, 2\pi]$ 上等间隔采样 N 点,再进行 IDFT,得到结果就是原序列 $x(n)$ 以 N 为周期的周期延拓后的主值区间,因此,为保证时域不发生混叠,在频域进行抽样的时候,采样点数 N 要大于时域离散信号的长度 M 。如果频域采样点数没有达到要求,时域会发生混叠失真,得到的时域图像则和原序列不一致。

(3) 简要回答思考题

通过第二问可知，只要先将时域序列进行长度为 N 的周期延拓，然后取主值区间得到新的序列，对该序列进行 N 点 DFT 即可。

(4) 程序清单和有关曲线

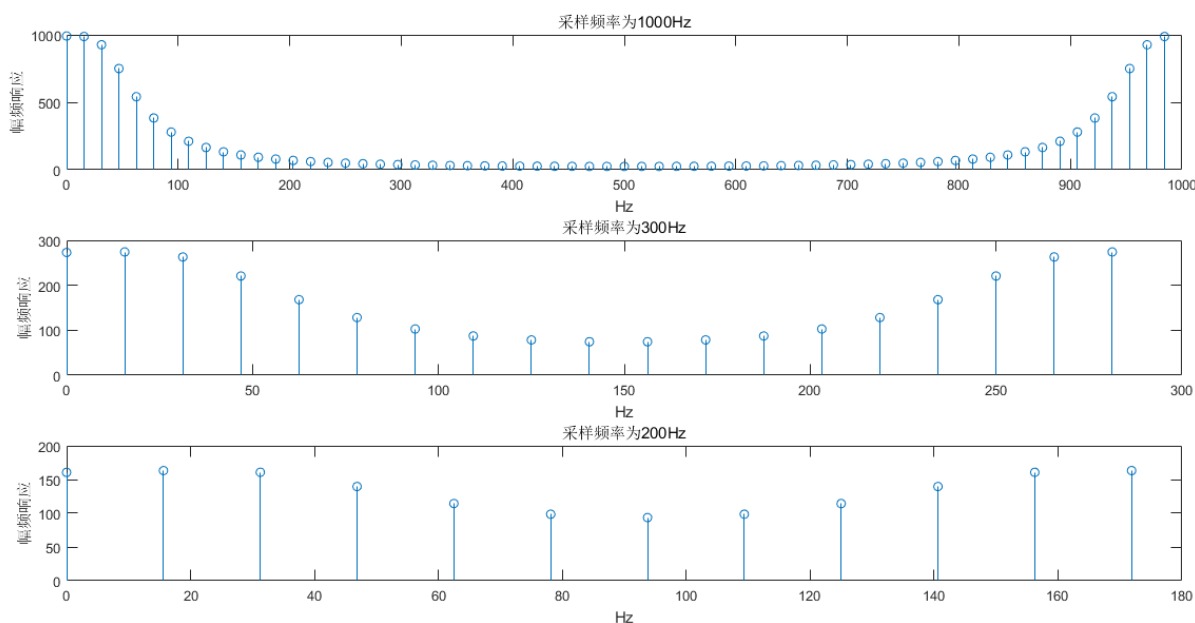
第一问：

```
A = 444.128;
a=50*sqrt(2)*pi;
w0=50*sqrt(2)*pi;
Tp = 64/1000;%观测时间
T1 = 1/1000;
T2 = 1/300;
T3 = 1/200;
N1=0:Tp*1000-1;N2=0:Tp*300-1;N3=0:Tp*200-1;%序列 x 轴 N1, N2, N3
x1=A*exp(-a*N1*T1).*sin(w0*N1*T1);%产生采样序列 x1(n)
x2=A*exp(-a*N2*T2).*sin(w0*N2*T2);%产生采样序列 x2 (n)
x3=A*exp(-a*N3*T3).*sin(w0*N3*T3);%产生采样序列 x3 (n)
%采样频率不同,变换点数不同,N= fs*tp
M1 = Tp*1000;%变换点数
M2 = Tp*300;
M3 = Tp*200;
y1=fft(x1,M1);%采样序列 x1 (n) 的 FFT 变换
y2=fft(x2,M2);%采样序列 x2 (n) 的 FFT 变换
y3=fft(x3,M3);%采样序列 x3 (n) 的 FFT 变换
w1 = (0:M1-1)/Tp;
subplot(3,1,1)
stem(w1 ,abs(y1))%采样频率为 1000Hz
xlabel('Hz')
ylabel('幅频响应')
title('采样频率为 1000Hz')
subplot(3,1,2)
```

```

w2 = (0:M2-1)/Tp;
stem(w2,abs(y2))%采样频率为 300Hz
xlabel('Hz')
ylabel('幅频响应')
title('采样频率为 300Hz')
subplot(3,1,3)
w3 = (0:M3-1)/Tp;
stem(w3,abs(y3))%采样频率为 200Hz
xlabel('Hz')
ylabel('幅频响应')
title('采样频率为 200Hz')

```



第二问:

```

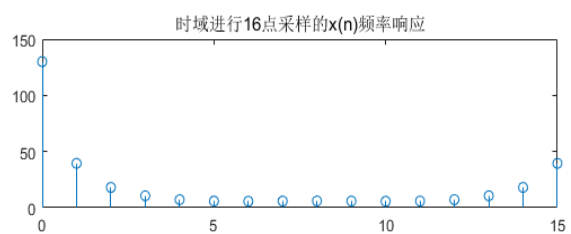
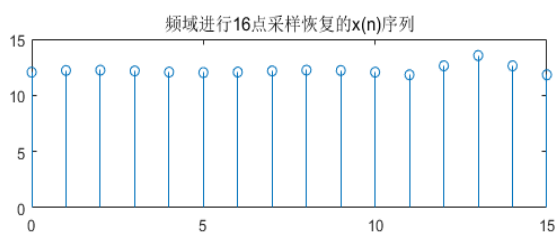
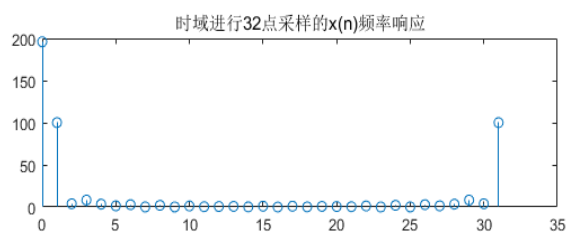
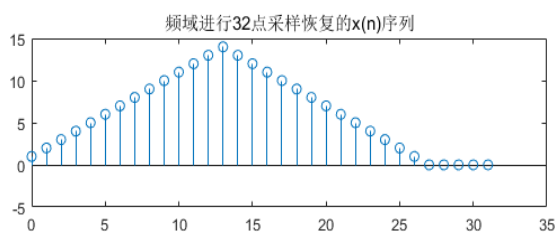
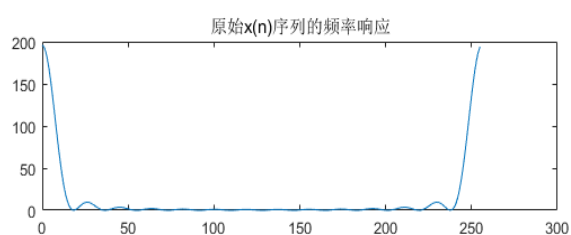
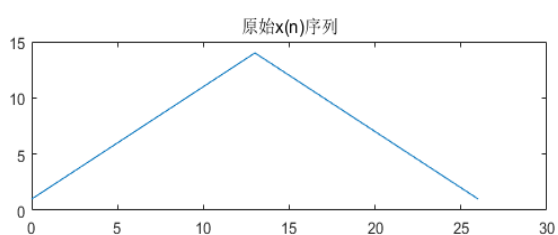
n = 0:26;
xn = (n+1).*(n>=0&n<=13)+(27-n).*(n>=14&n<=26);% x(n)
xk =fft(xn,256)%用点 fft 近似近似频率响应
xk1=fft(xn,32);%32 点 FFT
xk2 = fft(xn,16);%16 点 FFT
xn1 = ifft(xk1,32)% x32n
xn2 = ifft(xk1(1:27),16)% x16n
subplot(3,2,1)
plot(n,xn)%绘制 x(n)
title('原始 x(n)序列');
subplot(3,2,2)
k = 0:255;
plot(k,abs(xk))%默认 x(k)
title('原始 x(n)序列的频率响应');
subplot(3,2,3)

```

```

stem(0:31,xn1)%32 点 xn
title('频域进行 32 点采样恢复的 x(n)序列');
subplot(3,2,4)
stem(0:31,abs(xk1))%32 点 fft
title('时域进行 32 点采样的 x(n)频率响应');
subplot(3,2,5)
stem(0:15,xn2)%16 点 xn
title('频域进行 16 点采样恢复的 x(n)序列');
subplot(3,2,6)
stem(0:15,abs(xk2))%16 点 fft
title('时域进行 16 点采样的 x(n)频率响应');

```



实验三

(1) 完成各个实验任务和要求，附上程序清单和有关曲线

答;

1.

第一题代码如下

```
xn = ones(1,4);
n = 0:7;
x2n = (n+1).*(0<= n&n <=3)+(8-n).*(4 <=n &n<=7);
x3n = (4-n).*(0<= n&n <=3)+(n-3).*(4 <=n &n<=7);
N1 = 8 ; N2 = 16;
k1 = 0:7;k2 = 0:15;
w1 = 2*k1/N1;
w2 = 2*k2/N2;
xk8 = fft(xn,N1);% xn 的 8 点 fft
x2k8 = fft(x2n,N1);% x2n 的 8 点 fft
x3k8 = fft(x3n,N1);% x3n 的 8 点 fft
xk16 = fft(xn,N2);% xn 的 16 点 fft
x2k16 = fft(x2n,N2);% x2n 的 16 点 fft
x3k16 = fft(x3n,N2);% x3n 的 16 点 fft
subplot(3,2,1)
stem(w1,abs(xk8))
xlabel('ω/π')
title('xn 的 8 点 fft')
subplot(3,2,2)
stem(w2,abs(xk16))
xlabel('ω/π')
title('xn 的 16 点 fft')

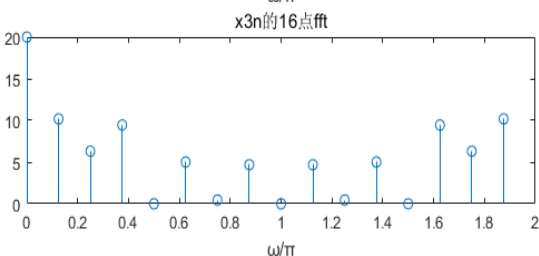
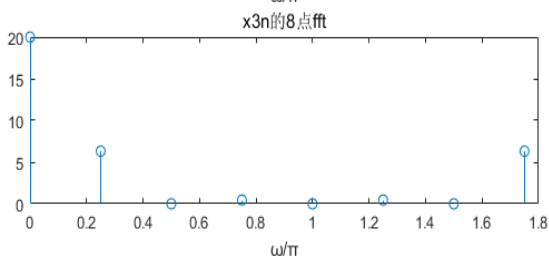
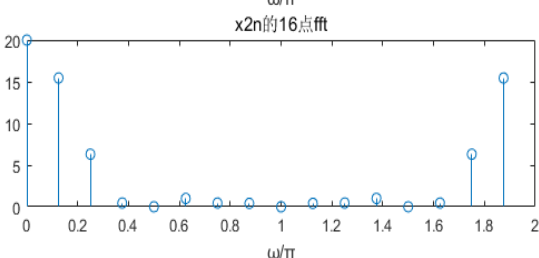
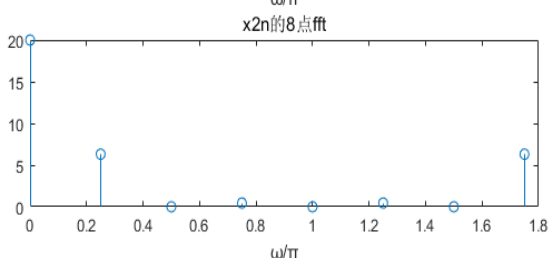
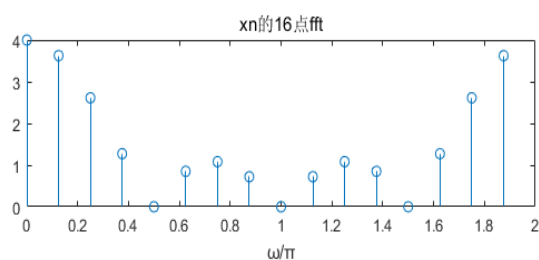
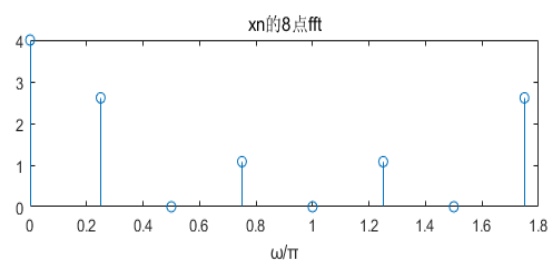
subplot(3,2,3)
stem(w1,abs(x2k8))
xlabel('ω/π')
title('x2n 的 8 点 fft')
subplot(3,2,4)
stem(w2,abs(x2k16))
xlabel('ω/π')
title('x2n 的 16 点 fft')

subplot(3,2,5)
stem(w1,abs(x3k8))
xlabel('ω/π')
title('x3n 的 8 点 fft')
subplot(3,2,6)
```

```

stem(w2,abs(x3k16))
xlabel('ω/π')
title('x3n 的 16 点 fft')

```



结果分析：通过对比发现， $x_1(n)$ 的 8 点 DFT 和 16 点 DFT 正好是 $x_1(n)$ 频域的 8 点抽样和 16 点抽样,另外,发现 $x_2(n)$ 和 $x_3(n)$ 8 点 DFT 频谱的一样,但 16 点 DFT 不一致,这是因为 $x_3(n)$ 可以做到为 $x_2(n)$ 的 8 点循环移位,但做不到 16 点循环移位,故 16 点的 DFT 二者不在一致.

2.

第二题代码如下：

```

n = 0:159;
x4n = cos(pi/4*n);
x5n = cos(pi/4*n)+cos(pi/8*n);
N1 = 8; N2 = 16;
k1 = 0:7;k2 = 0:15;
w1 = 2*k1/N1;
w2 = 2*k2/N2;%归一化数字频率
x4k8 = fft(x4n,8);
x4k16 = fft(x4n,16);
x5k8 = fft(x5n,8);
x5k16 = fft(x5n,16);

```

```

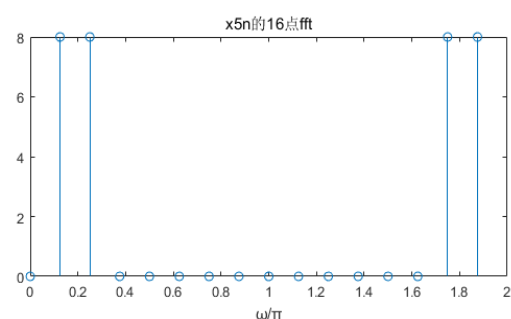
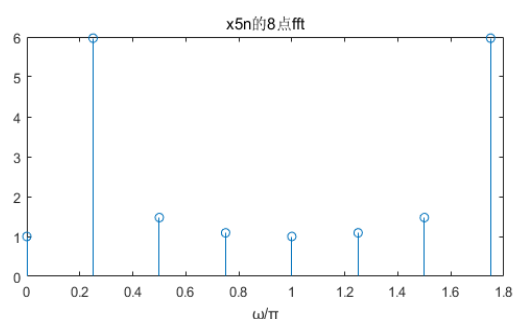
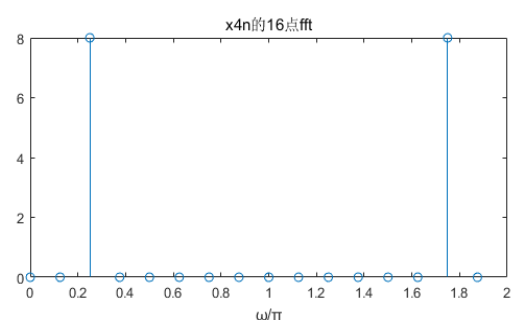
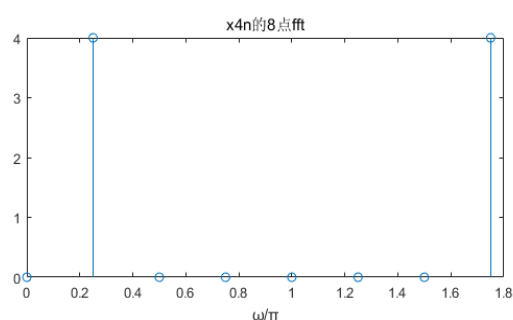
subplot(2,2,1)
stem(w1,abs(x4k8))

```

```

title('x4n 的 8 点 fft')
xlabel('ω/π')
subplot(2,2,2)
stem(w2,abs(x4k16))
title('x4n 的 16 点 fft')
xlabel('ω/π')
subplot(2,2,3)
stem(w1,abs(x5k8))
title('x5n 的 8 点 fft')
xlabel('ω/π')
subplot(2,2,4)
stem(w2,abs(x5k16))
title('x5n 的 16 点 fft')
xlabel('ω/π')

```



结果分析：通过横向对比发现， $x_4(n)$ 的周期为 8 或者 16,所以可以得到正确的谱线, $x_5(n)$ 在 $x_4(n)$ 的时域序列基础上增加了 $\cos(\pi n/8)$,导致其周期变为 16,从而 8 点的 FFT 会导致频谱不正确，只能在 16 点 DFT 上得到正确结果。

3.

第三题代码如下：

```

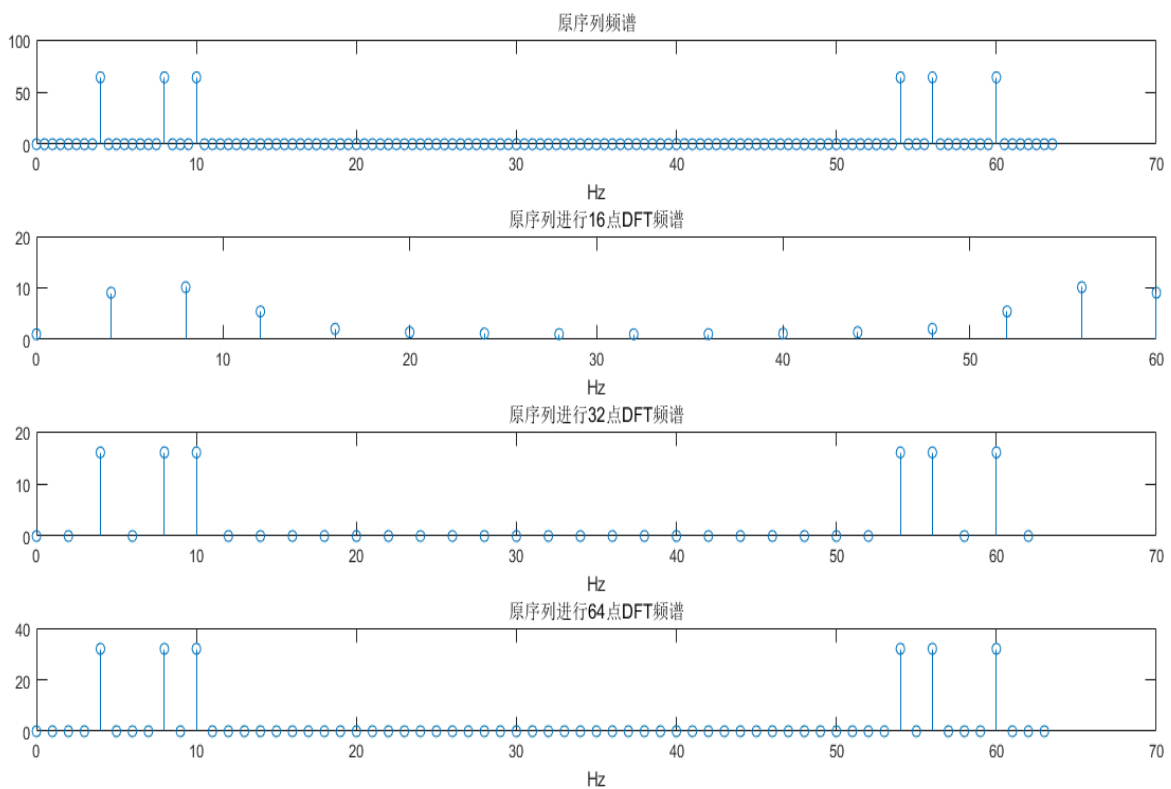
FS = 64;
N0 = 128;
n = 0:127;
x6n = cos(8*pi/FS*n)+cos(16*pi/FS*n)+cos(20*pi/FS*n);
N16 = 16;N32 = 32;N64 = 64;
x6k16 = fft(x6n,N16);
x6k32 = fft(x6n,N32);
x6k64 = fft(x6n,N64);

```

```

x6k = fft(x6n,N0);
k0 = 0:127;k16 = 0:15;
k32 = 0:31;k64 = 0:63;
% 换算成模拟频率
f0 = FS/N0*k0;f16 = FS/N16*k16;
f32 = FS/N32*k32;f64 = FS/N64*k64;
subplot(4,1,1)
stem(f0,abs(x6k))% fft 近似序列的频率响应
title('原序列频谱')
xlabel('Hz')
subplot(4,1,2)
stem(f16,abs(x6k16))
title('原序列进行 16 点 DFT 频谱')
xlabel('Hz')
subplot(4,1,3)
stem(f32,abs(x6k32))
title('原序列进行 32 点 DFT 频谱')
xlabel('Hz')
subplot(4,1,4)
stem(f64,abs(x6k64))
title('原序列进行 64 点 DFT 频谱')
xlabel('Hz')
结果如下：

```



结果分析:

对于 $x_6(t)$ 含有 3 个频率成分, $f_1=4\text{Hz}$, $f_2=8\text{Hz}$, $f_3=10\text{Hz}$, 其部分周期分别为 0.25s, 0.125s, 0.1s, 故 $x_6(t)$ 的周期可以取 0.5s, 采样频率 $F_s = 64\text{Hz}$, 当 N 分别为 16, 32, 64 时, 对应观察时间为 0.25s, 0.5s, 1s, 所以当 $N=16$ 时无法观察到正确的频谱, $N=32$, $N=64$ 时可以观察到正确的频谱, 三根谱线正好位于 $f_1=4\text{Hz}$, $f_2=8\text{Hz}$, $f_3=10\text{Hz}$ 处。

(2) 简要回答思考题

答:

1. 可以先试着进行 N 点的 FFT, 观察频谱, 如果不满足要求, 则可以再次进行 $2N$ 次的 FFT, 直到满足要求。
2. 对于周期信号, 只要保持 N 满足是整数倍周期长度即可, 对于非周期信号。因为 FFT 的分辨率是 $2\pi/N$, 故应使 $N > 2\pi/F$ 。
3. 当 N 等于 8 时, $x_3(n)$ 可以做到为 $x_2(n)$ 的 8 点循环移位, 但做不到 16 点循环移位, 故 16 点的 DFT 二者不在一致。

实验四

(1) 简述实验目的及原理

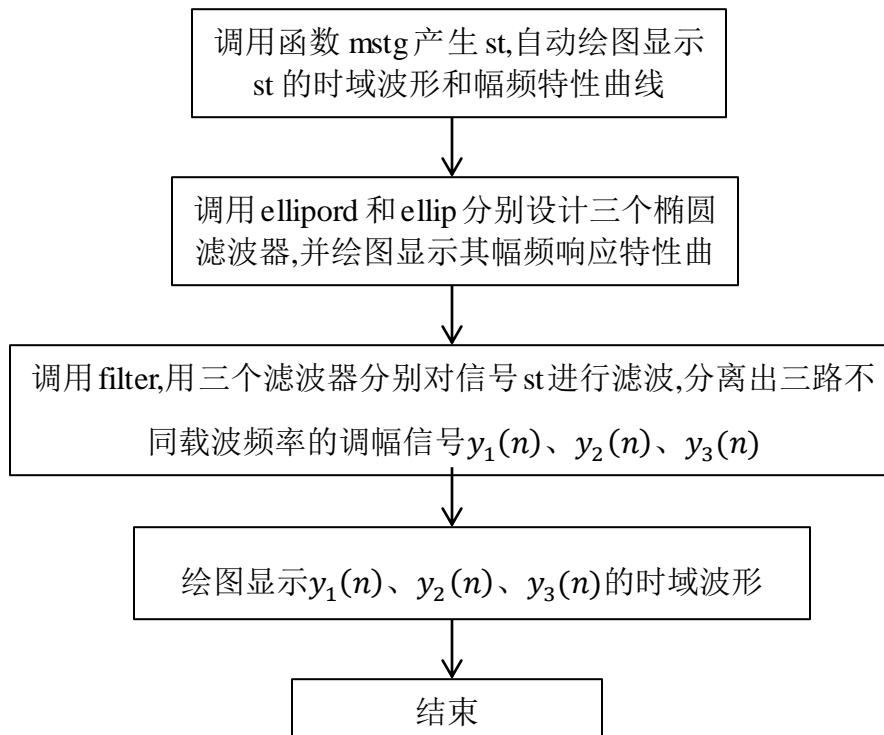
答:

1.实验目的:熟悉使用双线性变换法设计 IIR 数字滤波器的原理和方法,学会调用 MATLAB 信号处理工具箱中滤波器设计各种 IIR 数字滤波器,学会根据要求确定滤波器参数,掌握数字滤波的概念。

2.实验原理:在时域混叠不好分析的信号,将其转换到频域,通过在频域进行滤波操作,可以将信号实现分离。

(2) 画出实验主程序框图, 打印程序清单

程序框图:



程序清单:

```
function st=mstg()
%产生信号序列向量 st, 并显示 st 的时域波形和幅频曲线
%st=mstg 返回三路调幅信号相加形成的混合信号, 长度 N=800
N=800;%N 为信号 st 的长度
Fs=10000;T=1/Fs;Tp=N*T;%采样频率 Fs=10kHz, Tp 为采样时间
t=0:T:(N-1)*T;k=0:N-1;f=k/Tp;
fc1=Fs/10; %第 1 路调幅信号的载波频率 fc1=1000Hz
fm1=fc1/10; %第 1 路调幅信号的调制信号频率 fm1=100Hz
fc2=Fs/20 %第 2 路调幅信号的载波频率 fc2=500Hz
```

```

fm2=fc2/10;      %第2路调幅信号的调制信号频率 fm2=50Hz
fc3=Fs/40;      %第3路调幅信号的载波频率 fe3=250Hz
fm3=fc3/10;      %第3路调幅信号的调制信号频率 fm3=25Hz
xt1=cos(2*pi*fm1*t).*cos(2*pi*fc1*t);    %产生第1路调幅信号
xt2=cos(2*pi*fm2*t).*cos(2*pi*fc2*t);    %产生第2路调幅信号
xt3=cos(2*pi*fm3*t).*cos(2*pi*fc3*t);    %产生第3路调幅信号
st=xt1+xt2+xt3;%三路调幅信号相加
fxt=fft(st,N);%计算信号 st 的频谱
%====以下为绘图部分，绘制 st 的时域波形和幅频特性曲线=====
subplot(2,1,1)
plot(t,st);grid;
xlabel('t/s');ylabel('s(t)');
axis([0,Tp/8,min(st),max(st)]);
title('(a) s(t)的波形')
subplot(2,1,2)
stem(f,abs(fxt)/max(abs(fxt)));grid;
title('(b) s(t)的频谱')
axis([0,Fs/5,0,1.2]);
xlabel('f/Hz');ylabel('幅度')
end

%三路调幅信号的载波频率分别为 250hz 500hz 1000hz 带宽分别为 50hz
100hz 200hz
Fs=10000;%采样频率
T=1/Fs;
st = mstg;
%低通滤波器
fp=300;fs=400;%通带和阻带边界
wp=2*fp/Fs;ws=2*fs/Fs;%数字指标
rp=0.1;rs=60;%通带最大衰减,阻带最小衰减
[N,wp]=ellipord(wp,ws,rp,rs);%调用 ellipord 计算椭圆 DF 阶数 N 和通带截止
频率 p
[B,A]=ellip(N,rp,rs,wp);%调用 ellip 计算椭圆低通 DF 系统函数系数向量 B 和
A
y1 = filter(B,A,st);%滤波
[h,w] = freqz(B,A,1024);%返回滤波器的频率响应和频率坐标轴
m =abs(h);
figure
subplot(3,2,1)
plot(w*Fs/2/pi,20*(log10(m/max(m))));%绘制衰减图
axis([0 1200 -120 10])
set(gca,'XTick',[0:200:1200]);
set(gca,'YTick',[-120:30:10]);%设置坐标轴范围
xlabel('频率');ylabel('衰减幅度')

```

```

grid on;
subplot(3,2,2)
n = 0:length(y1)-1;
t = n*T;
plot(t,y1)
xlabel('时间');title('低通滤波结果')
grid on;
%带通滤波器
fp1=400;fp2=600;%通带边界
fs1=300;fs2=700;%阻带边界
wp=[2*fp1/Fs,2*fp2/Fs];%数字指标
ws=[2*fs1/Fs,2*fs2/Fs];%数字指标
rp=0.1;rs=60;
[N,wp]=ellipord(wp,ws,rp,rs);%调用 ellipord 计算椭圆 DF 阶数 N 和通带截止
频率 wp
[B,A]=ellip(N,rp,rs,wp);%调用 ellip 计算椭圆带通 DF 系统函数系数向量 B 和
A
y2=filter(B,A,st);%滤波
[h,w] = freqz(B,A,1024);%返回滤波器的频率响应和频率坐标轴
m=abs(h);
subplot(3,2,3)
plot(w*Fs/2/pi,20*(log10(m/max(m))));%绘制衰减图
axis([0 1200 -120 10])
set(gca,'XTick',[0:200:1200]);
set(gca,'YTick',[-120:30:10]);%设置坐标轴范围
xlabel('频率');ylabel('衰减幅度')
grid on;
subplot(3,2,4)
n = 0:length(y2)-1;
t = n*T;
plot(t,y2)
xlabel('时间');title('带通滤波结果')
grid on;
%高通滤波器
fp=800;fs=600;%通带边界和阻带边界
wp=2*fp/Fs;ws=2*fs/Fs;
rp=0.1;rs=60;%通带最大衰减,阻带最小衰减
[N,wp]=ellipord(wp,ws,rp,rs);%调用 ellipord 计算椭圆 DF 阶数 N 和通带截止
频率 p
[B,A]=ellip(N,rp,rs,wp,'high');%调用 ellip 计算椭圆高通 DF 系统函数系数向
量 B 和 A
y3 = filter(B,A,st);%滤波
[h,w] = freqz(B,A,1024);%返回滤波器的频率响应和频率坐标轴
m=abs(h);

```

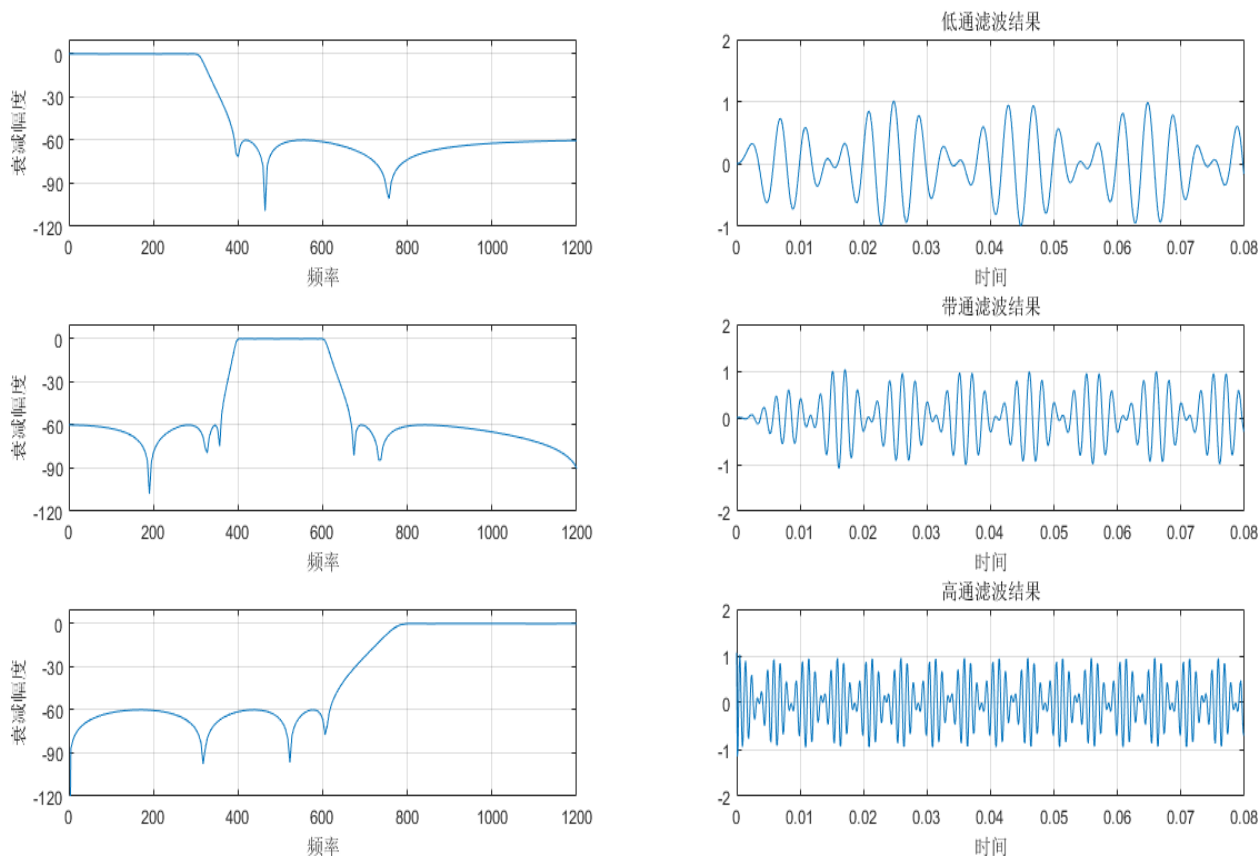
```

subplot(3,2,5)
plot(w*Fs/2/pi,20*(log10(m/max(m))));%绘制衰减图
axis([0 1200 -120 10])
set(gca,'XTick',[0:200:1200]);
set(gca,'YTick',[-120:30:10]);%设置坐标轴范围
xlabel('频率');ylabel('衰减幅度')
grid on;
subplot(3,2,6)
n = 0:length(y3)-1;
t = n*T;
plot(t,y3)
xlabel('时间');title('高通滤波结果')
grid on;

```

(3) 绘制三个分离滤波器的损耗函数曲线

(4) 绘制经过滤波分离的三路抑制载波调幅小号的时域波形



(5) 简要回答思考题

1.

第一路载波频率和调制信号频率分别为：1000Hz，100Hz

第二路载波频率和调制信号频率分别为：500Hz，50Hz

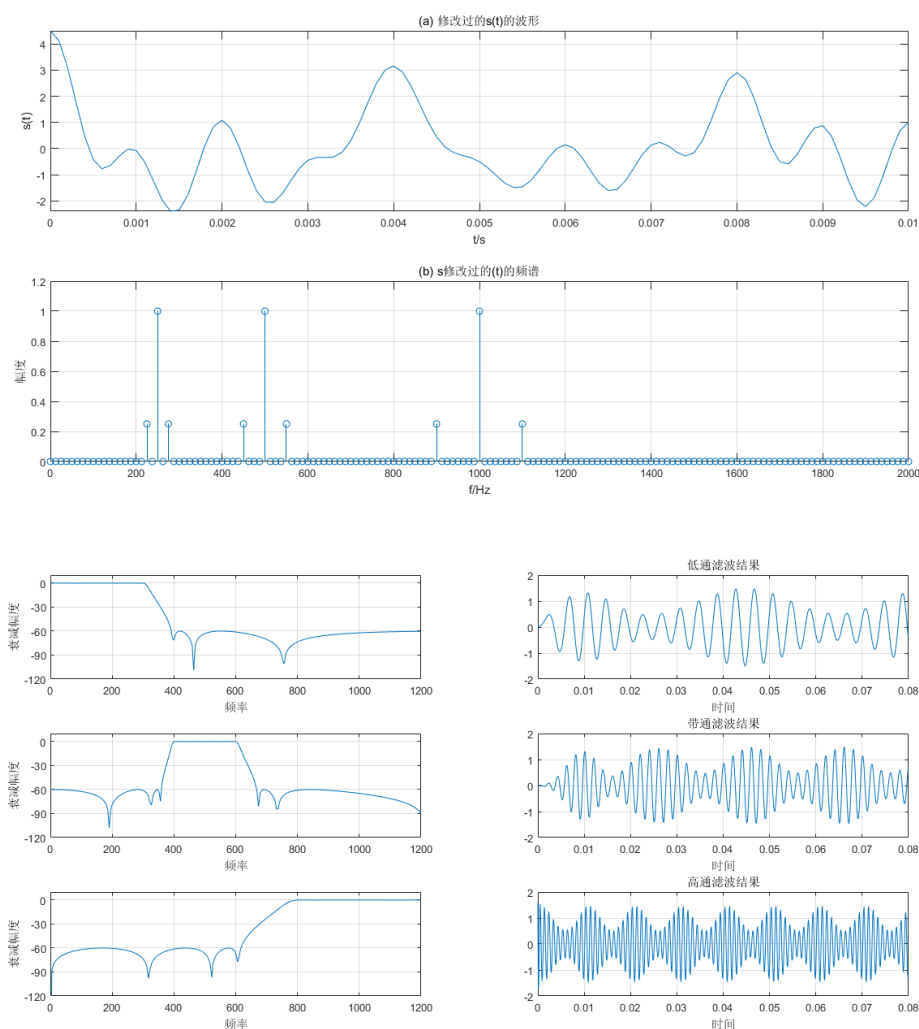
第三路载波频率和调制信号频率分别为：250Hz，25Hz

2.

当 $N = 1800$ 时，无法得到正确的 6 根谱线，原因分析为：生成的 $s(t)$ 函数是周期信号，根据理论只有 N 取周期的整数倍时才能得到清晰的频谱。注意到本题使用 10KHz 时，可以得到完整清晰的频谱，而且 $s(t)$ 函数的各个频率分量均为 25Hz 的倍数，而 $10\text{KHz} = 400 \times 25\text{Hz}$ ，故只要是 N 为 400 的整数倍就一定能得到正确清晰的频谱，故 1800 不行，2000 可以。

3.

经过修改的 mstg 函数及其滤波结果如下：



经过分析，发现频域增加了三路载波频率谱线。

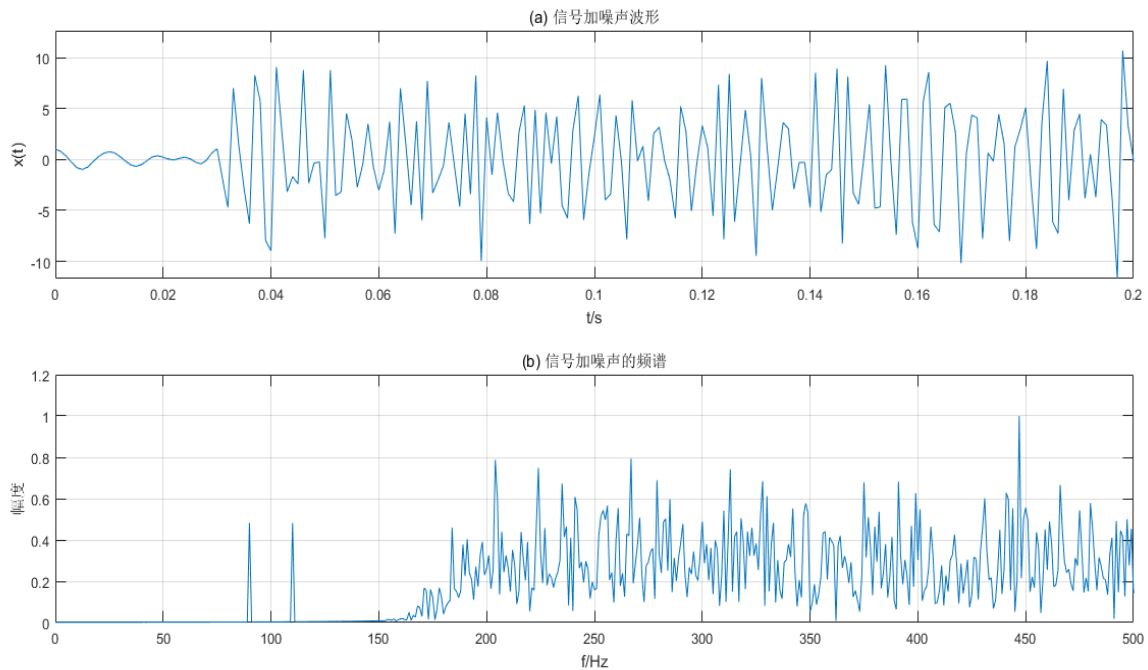
实验五

(1) 对两种设计 FIR 滤波器的方法（窗函数法和等波纹最佳逼近

近法) 进行分析比较, 简述其优缺点

(2) 附程序清单, 打印实验内容要求绘图显示的曲线图

```
加性噪声信号产生函数:
function xt=xtg()
%实验五信号 x(t)产生,并显示信号的幅频特性曲线
%xt=xtg(N)?产生一个长度为 N=1000,有加性高频噪声的单频调幅信号 xt,采
样频率 Fs=1000Hz
%载波频率 fc=Fs/10=100Hz,调制正弦波频率 f0=fc/10=10Hz
N=1000;
Fs=1000;T=1/Fs;Tp=N*T;
t=0:T:(N-1)*T;
fc=Fs/10;f0=fc/10; %载波频率 fc=Fs/10, 单频调制信号频率为 f0=Fc/10;
mt=cos(2*pi*f0*t); %产生单频正弦波调制信号 mt, 频率为 f0
ct=cos(2*pi*fc*t); %产生载波正弦波信号 ct, 频率为 fc
xt=mt.*ct; %相乘产生单频调制信号 xt
nt=2*rand(1,N)-1; %产生随机噪声 nt
%=====设计高通滤波器 hn,用于滤除噪声 nt 中的低频成分,生成高通噪声
=====
fp=150;fs=200;Rp=0.1;As=70; %滤波器指标
fb=[fp,fs];m=[0,1]; %?计算 remezord 函数所需参数 f,m,dev
dev=[10^(-As/20),(10^(Rp/20)-1)/(10^(Rp/20)+1)];
[n,fo,mo,W]=remezord(fb,m,dev,Fs); %?确定 remez 函数所需参数
hn=remez(n,fo,mo,W); %调用 remez 函数进行设计,用于滤除噪声 nt 中的低
频成分
yt=filter(hn,1,10*nt); %滤除随机噪声中低频成分,生成高通噪声 yt
%=====
=====
xt=xt+yt; %噪声加信号
fst=fft(xt,N);
k=0:N-1;f=k/Tp;
subplot(2,1,1);
plot(t,xt);grid;
xlabel('t/s');ylabel('x(t)');
axis([0,Tp/5,min(xt),max(xt)]);
title('(a) 信号加噪声波形')
subplot(2,1,2);
plot(f,abs(fst)/max(abs(fst)));grid;
title('(b) 信号加噪声的频谱')
axis([0,Fs/2,0,1.2]);
xlabel('f/Hz');ylabel('幅度')
end
结果:
```



窗函数设计法:

```
xt=xtg();%产生加性噪声信号
```

```
N = 1000;Fs = 1000;
```

```
fcuts = [0.24 0.3];
```

```
%归一化频率 omega/pi 这里指通带截止频率、阻带起始频率
```

```
mags = [1 0];
```

```
% mags = [1 0]为由 f 指定的各个频带上的幅值向量，一般只有 0 和 1 表示
```

```
devs = [0.01 10^(-2.5)];
```

```
%用于指定各个频带输出滤波器的频率响应与其期望幅值之间的最大输出  
误差或偏差
```

```
[n,Wn,beta,ftype] = kaiserord(fcuts,mags,devs); %计算出凯塞窗 N, beta 的值
```

```
hh= fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');%设计凯塞窗
```

```
[hh1,w1]=freqz(hh,1,1024);%系统频率响应
```

```
figure
```

```
subplot(3,1,1)
```

```
plot(w1*Fs/2/pi,abs(hh1))%%观察窗函数的频率响应
```

```
xlabel('f/Hz');title('窗函数的频率响应');
```

```
subplot(3,1,2)
```

```
plot(w1*Fs/2/pi,20*log10(abs(hh1)))%%绘制衰减图
```

```
xlabel('f/Hz');title('窗函数衰减特性');
```

```
subplot(3,1,3)
```

```
yt=fftfilt(hh,xt); %调用函数 fftfilt 对 xt 滤波
```

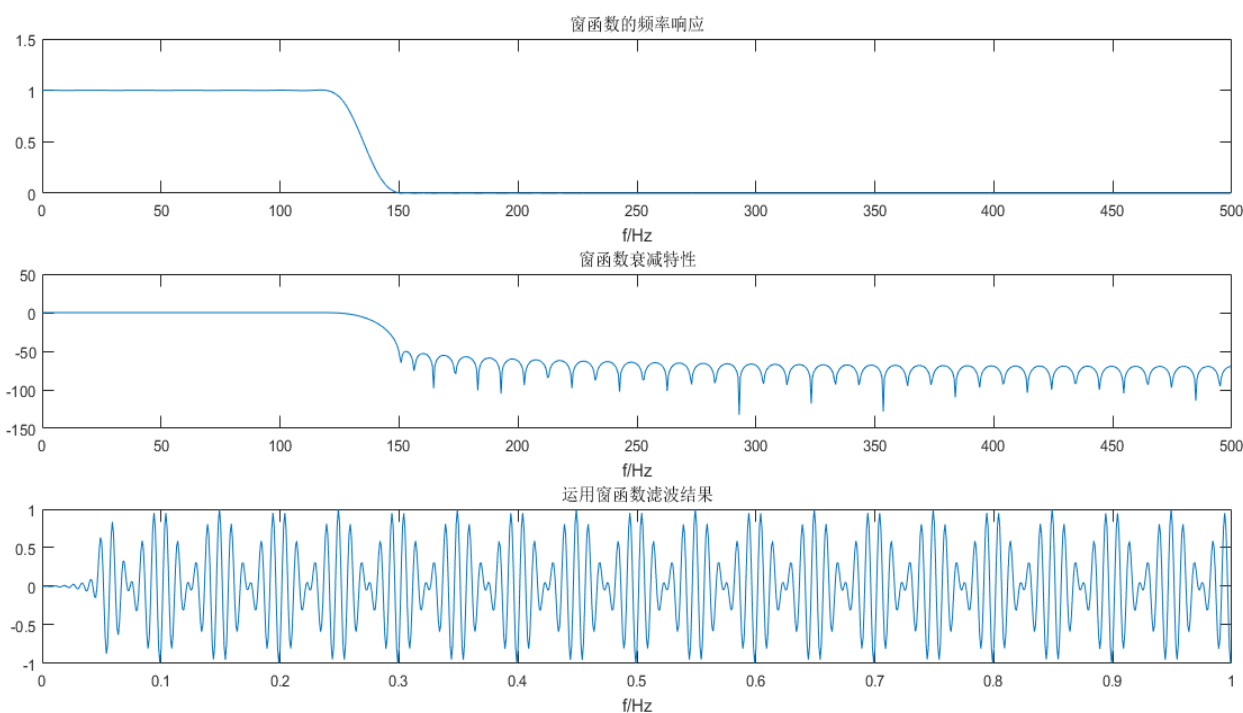
```
T = 1/Fs;
```

```
t=0:T:(N-1)*T;
```

```
plot(t,yt)%观察滤波结果
```

```
xlabel('f/Hz');title('运用窗函数滤波结果');
```


结果:



等波纹滤波器:

```
xt=xtg();%产生加性噪声信号
```

```
N = 1000;Fs = 1000;
```

```
fcuts = [0.24 0.3];
```

```
%归一化频率  $\omega/\pi$  这里指通带截止频率、阻带起始频率
```

```
mags = [1 0];
```

```
% mags = [1 0]为由 f 指定的各个频带上的幅值向量，一般只有 0 和 1 表示
```

```
devs = [0.01 10-2.5];
```

```
%用于指定各个频带输出滤波器的频率响应与其期望幅值之间的最大输出误差或偏差
```

```
[n,fpts,mag,wt]=remezord(fcuts,mags,devs);%估算参数
```

```
h2=remez(n,fpts,mag,wt);
```

```
[hh2,w2]=freqz(h2,1,1024);%系统频率响应
```

```
figure
```

```
subplot(3,1,1)
```

```
plot(w2*Fs/2/pi,abs(hh2))%观察等波纹滤波器的频率响应
```

```
xlabel('f/Hz');title('等波纹滤波器的频率响应');
```

```
subplot(3,1,2)
```

```
plot(w2*Fs/2/pi,20*log10(abs(hh2)))%绘制衰减图
```

```
xlabel('f/Hz');title('等波纹滤波器衰减特性');
```

```
subplot(3,1,3)
```

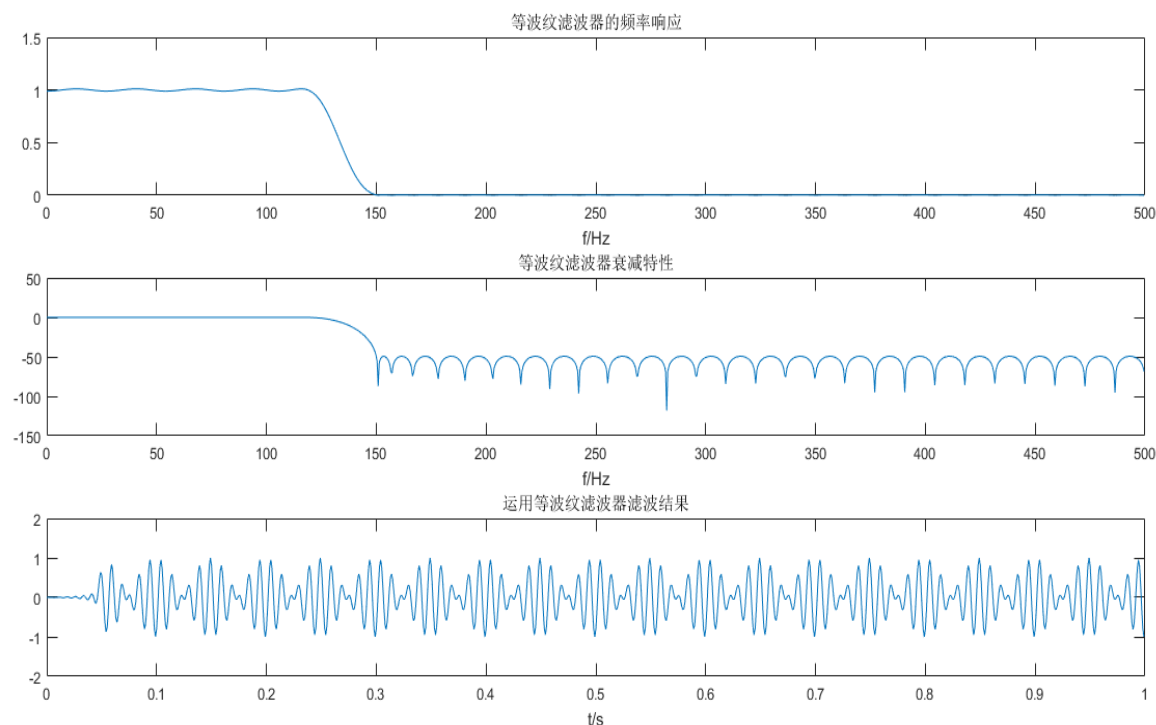
```
yt=ffilt(hh,xt); %调用函数 ffilt 对 xt 滤波
```

```
T = 1/Fs;
```

```
t=0:T:(N-1)*T;
```

```
plot(t,yt)%%观察滤波结果
xlabel('t/s');title('运用等波纹滤波器滤波结果');
```

结果:



(3) 分析总结实验结果

实验发现,两种方法设计的滤波器都能有效地从噪声中提取信号,但窗函数滤波器结束 $n = 98$,等波纹滤波器结束 $n = 75$,显然最佳逼近法设计的滤波器阶数低得多。

(4) 简要回答思考题

1.将问题所提供的模拟频率指标转换为数字频率指标,然后设计合适的窗函数对理想低通滤波器进行截取,然后对信号进行滤波。

$$2. \omega_{c1} = (\omega_{s1} + \omega_{p1})/2, \quad \omega_{cu} = (\omega_{su} + \omega_{pu})/2。$$

3.因为等波纹滤波器设计法通带阻带波纹为等波纹,指标均匀分布,最大误差均匀分布,使得指标相同时,阶数最低。