

《信号分析与处理》实验部分：

例1-37：

源代码：

```
%两个矩形脉冲的卷积
%t1 t2 t3 分别表示f1 f2 f1和f2卷积的时间
close all;clear;clc;
tspan=0.01; %采样间隔
t1=0:tspan:3.5; %时间向量
t1_len=length(t1);
t2=0:tspan:3.5;
t2_len=length(t2);
t3=0:tspan:(t1_len+t1_len-2)*tspan; %卷积

%生成题目所要求的两个信号(这里直接写成矩形信号的形式)
f1=[zeros(1,length([0:tspan:(1-
0.01)])),3*ones(1,length([1:tspan:2])),zeros(1,length([2.01:tspan:3.5]))];
f2=[zeros(1,length([0:tspan:(1-
0.01)])),1*ones(1,length([1:tspan:3])),zeros(1,length([3.01:tspan:3.5]))];

%两矩形信号进行卷积
w=conv(f1,f2);
w=w*tspan;

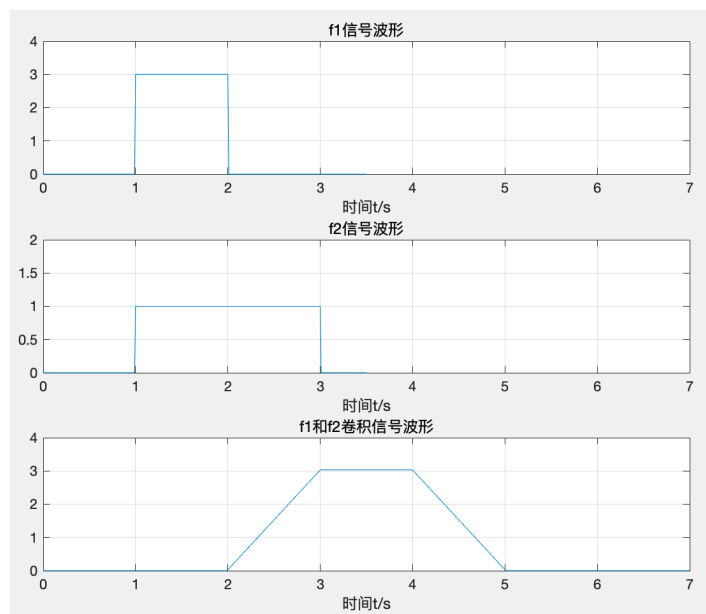
%绘制f1信号波形
subplot(3,1,1);
plot(t1,f1);
title('f1信号波形')
grid on;
xlabel('时间t/s');
axis([0 7 0 4]) %设置坐标范围

%绘制f2信号波形
subplot(3,1,2);
plot(t2,f2);
title('f2信号波形')
grid on;
xlabel('时间t/s');
axis([0 7 0 2]) %设置坐标范围

%绘制f1和f2卷积信号波形
subplot(3,1,3);
plot(t3,w);
title('f1和f2卷积信号波形')
grid on;
```

```
xlabel('时间t/s');
```

输出结果：



心得体会：

本课题的输出要求是两个矩形脉冲的卷积：

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t - \tau)$$

在matlab中，使用conv()函数进行卷积，值得注意的是，我们需要确定卷积之后的时间长度应是两个矩形信号时间之和。

设置采样间隔tspan=0.01，我们能生成一个关于时间的序列，对于这个时间序列，使用ones()和zeros()函数就能生成题目要求的两个矩形函数，最后，设置好绘图的坐标轴范围和标签，就能直观的输出两个矩形脉冲的卷积。

在输出中，有两个最初的矩形信号波形，以及两个矩形信号卷积之后的信号波形。

例1-39：

源代码：

```
%绘制信号幅度频谱
close all;clear;clc;
syms t v w x;          %定义变量

%表示题目所给的表达式
% heaviside(x)函数表示阶跃函数 returns the value 0 for x < 0, 1 for x > 0, and 1/2 for x = 0.
% x=1/2*exp(-1*t)*sym('heaviside(t)');      %例题中的表达式
x=sym(1/2*exp(-1*t)*heaviside(t-1));        %题目修改后的表达式,matlab版本太高，去掉单引号
```

`%傅立叶变化`

```
F = fourier(x);
```

`%绘制原信号波形`

```
subplot(2,1,1);
```

```
ezplot(x);      %用ezplot绘制函数，可用fplot代替，matlab版本太高，去掉单引号
```

```
%fplot(x);
```

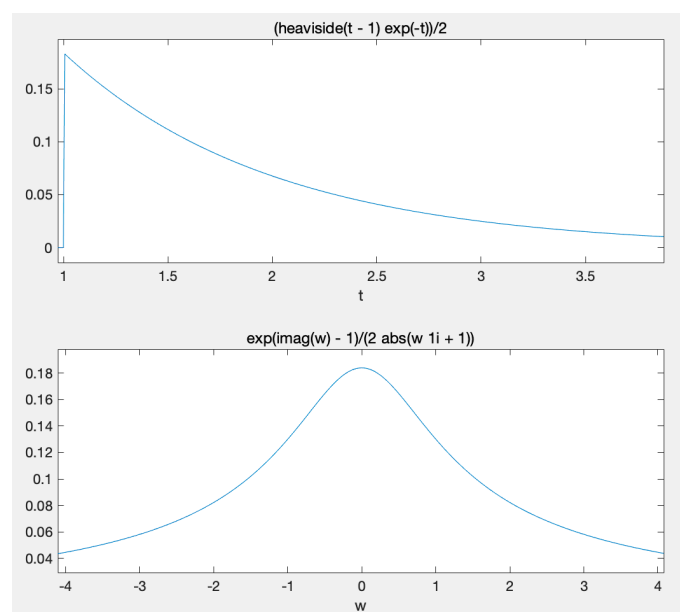
`%绘制傅立叶变化之后的波形`

```
subplot(2,1,2);
```

```
ezplot(abs(F)); %用ezplot绘制函数，可用fplot代替，matlab版本太高，去掉单引号
```

```
%fplot('abs(F)');
```

输出结果：



心得体会：

本课题要求输出信号的幅度频谱，首先，先用时域信号的方式表示出信号： $x = \text{sym}(1/2 \exp(-1t) * \text{heaviside}(t-1))$ ，然后对时域信号进行傅立叶变化， $F = \text{fourier}(x)$ ，最后再使用`ezplot()`函数绘制出时域信号的图像和频域信号的图像。值得注意的一点是，不同版本的matlab中`sym`的用法存在区别。

如图输出了经过平移后的信号，以及经过傅立叶变化后的幅度频谱。

例1-42：

源代码：

`%求信号的频谱傅立叶反变化，给出表达式`

```
close all;clear;clc;
```

```
syms t w a;      %定义变量
```

```
Fw = sym(2*a/(w^2+a^2)); %题目表达式
```

```
ft=ifourier(Fw,w,t) %傅立叶反变化
```

输出结果：

```
ft =  
  
(a*exp(-abs(t)*(a^2)^(1/2)))/(a^2)^(1/2)
```

心得体会：

本课题要求使用matlab输出信号的频谱傅立叶反变化,给出表达式。Matlab提供了ifourier()函数求解傅立叶反变化，其中的变量需要用sym命令说明为符号变量，虽然ifourier()函数用方便，但是存在很多局限性，比如含有冲击项的信号无法使用ezplot绘制出来。

傅里叶反变化公式：

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(w)e^{iwt} dw$$

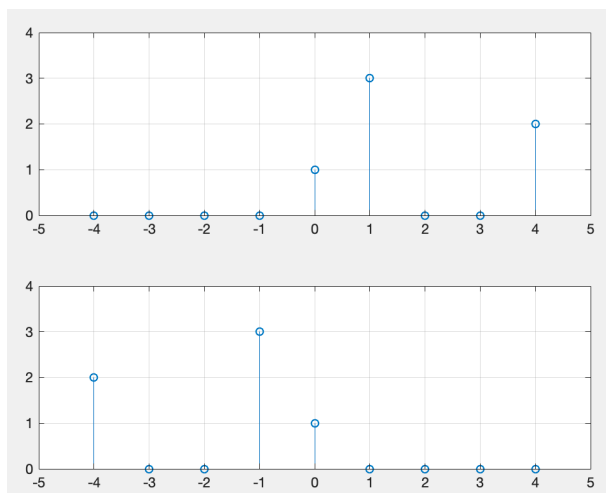
如图输出了傅立叶反变化的表达式。

例2-41：

源代码：

```
%离散信号翻转，注意有序列长度  
%离散信号的反转可用fliplr()函数实现  
close all;clear;clc;  
n=0:8; %定义序列长度  
nn=n-4; %生成对称x轴坐标  
%x=[zeros(1,4),1,2,0,0,3,]; %书中例子的信号  
x=[zeros(1,4),1,3,0,0,2,]; %题目的信号  
y=fliplr(x);  
  
figure(1); %打开画图1  
subplot(2,1,1);  
stem(nn,x); %画出原信号的火柴梗图  
axis([-5 5 0 4]); %坐标轴范围  
grid on;  
  
subplot(2,1,2);  
stem(nn,y); %画出反转信号的火柴梗图  
axis([-5 5 0 4]); %坐标轴范围  
grid on;
```

输出结果：



心得体会：

本课题要求离散信号的反转。在matlab中，可以使用fliplr()函数实现对信号的方法，需要注意的是，反转必须按照信号的中心，所以在生成坐标范围的时候，需要对称的生成。再就是离散信号的生成，我们可以使用序列的方法实现，如 $x=[\text{zeros}(1,4),1,3,0,0,2,]$ ，对称点先使用0代替。

在输出的图中，可以看到，信号经过翻转后，关于0点对称。

例2-42：

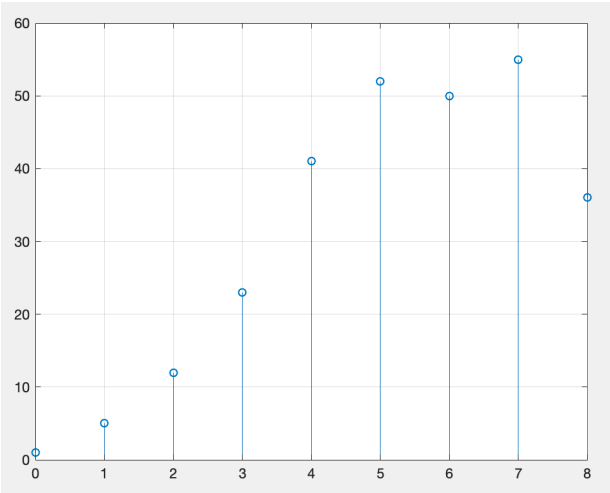
源代码：

```
% 两个离散信号的卷积
% 补充一下subplot
close all;clear;clc;
%两个离散信号的坐标轴长度N,M, 以及卷积信号的长度L
N=5;
M=5;
L=N+M-1;

%例题离散信号
%x=[1,2,3,4,5];
%h=[1,2,1,3,4];
%题目离散信号
x=[1,2,1,3,4];
h=[1,3,5,7,9];

y=conv(x,h);
n=0:(L-1);
stem(n,y); %火柴梗图
grid on;
```

输出结果：



心得体会：

本课题要求实现离散信号的卷积输出。在连续信号的时候使用的conv()函数依然可以用来实现离散信号的卷积，使用方法是两个有限长度序列向量作为参数，最后的输出只有卷积的结果，没有卷积的取值范围，但是根据离散信号的卷积特性，可以知道卷积信号结果的长度为length(x)+length(h)-1。

离散信号卷积输出：

$$y(n) = x(n) * h(n) = \sum_{m=-\infty}^{\infty} x(m)h(n - m)$$

如图，我们设置了卷积信号的长度为（5+5-1），并且合适的设置了输出型信号的上下幅值，展示了两个离散信号卷积之后的信号波形。

例2-44：

源代码：

```
% FFT 采样 幅度谱
close all;clear;clc;
k=16;    %采样频率
n1=[0:1:19];    %采样点20个
xa1=sin(2*pi*n1/k);

%绘制原始连续信号
figure(1);
subplot(1,2,1)
stem(n1,xa1)
xlabel('t/T');ylabel('x(n)');
title("20个采样点信号");

%{
xk1=fft(xa1);
```

```

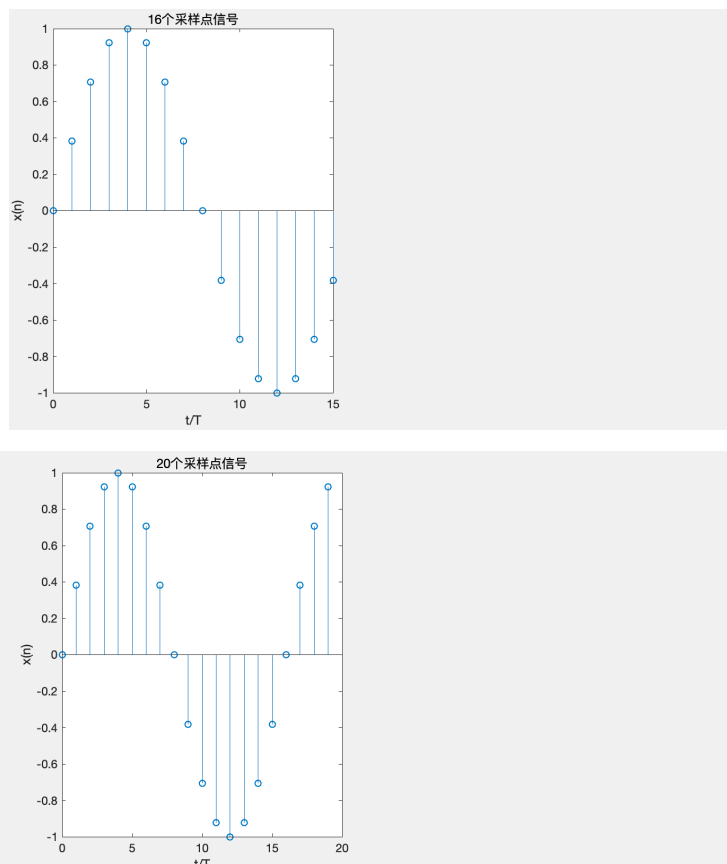
xk1=abs(xk1);
subplot(1,2,2)
stem(n1,xk1)
xlabel('k');ylabel('X(k)');
title("20个采样点傅立叶幅值");
%}

n2=[0:1:15];
xa2=sin(2*pi*n2/k);
figure(2)
subplot(1,2,1)
stem(n2,xa2)
xlabel('t/T');ylabel('x(n)');
title("16个采样点信号");

%{
xk2=fft(xa1);
xk2=abs(xk2);
subplot(1,2,2)
stem(n1,xk2)
xlabel('k');ylabel('X(k)');
title("16个采样点傅立叶幅值");
%}

```

输出结果：



心得体会：

本课题要求利用FFT快速傅立叶变化，观察输入信号的幅度谱。快速傅立叶变化（FFT）极大的减少了傅立叶变化的计算时间和运算压力，使得离散傅立叶变化（DFT）在信号处理中得到真正广泛的应用，在matlab中，实现快速傅立叶变化的函数主要有fft()和ifft()。

按照采样频率 $f_s=16f_a$ 进行采样，分别取20个和16个采样点坐标，可以得到上图所示的离散的采样信号。接着使用fft()函数，实现了离散时间信号的傅里叶变化后的幅值，运行时间较快，说明FFT能很高效的实现时间信号的傅里叶变化。

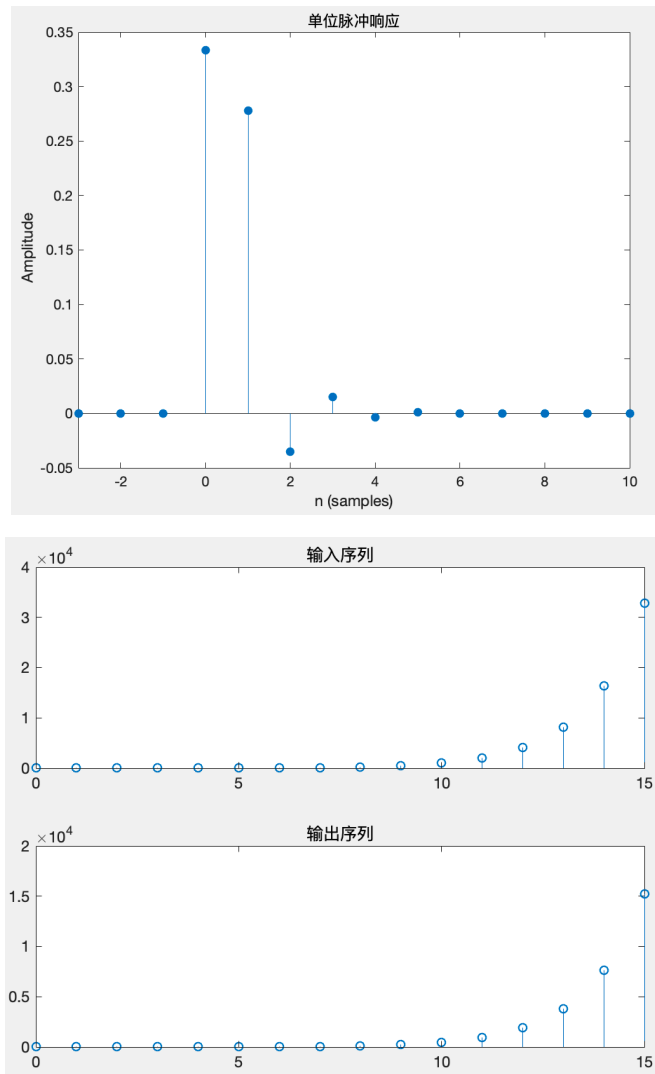
例3-21：

源代码：

```
%系统零输入响应
close all;clear;clc;
%系数向量
a=[3,0.5,-0.1];
b=[1,1,0];
figure(1)
impz(b,a,-3:10);    %时间范围内做图
title('单位脉冲响应');

%零输入响应
k=0:15;    %时间范围
x=2.^k;    %输入序列表达式
%x=(0.5).^k;    %输入序列表达式
y=filter(b,a,x);    %filter()滤波器函数，使得波形平滑
figure(2)
subplot(2,1,1);
stem(k,x)
title('输入序列')
subplot(2,1,2);
stem(k,y)
title('输出序列')
```

输出结果：



心得体会：

本课题研究离散系统的零输入响应。与连续系统类似，matlab提供了用于求解由差分方程所描述的离散系统冲击响应和阶跃响应，并绘制其时域波形的函数impz()和stepz()，以及求解离散系统在任意激励信号序列作用下的响应函数filter()和dlism()：

零输入响应： $y = \text{filter}(\text{num}, \text{den}, x)$

零状态响应： $y = \text{filter}(\text{num}, \text{den}, x, zi)$

全响应： $y = \text{filter}(\text{num}, \text{den}, u, x0)$

如图依次输出了离散系统的单位脉冲响应，输入信号序列，以及输入信号的激励下的零状态响应。

例3-23：

源代码：

```
%利用频率变化，对已知冲激响应的系统输入，求输出
close all;clear;clc;
syms w t;    %定义w和t
h=10*t*exp(-2*t)*heaviside(t);    %heaviside()=u(t)
```

```

H=fourier(h);
[Hn,Hd]=numden(H);    %得到h的分子和分母
Hnum=abs(sym2poly(Hn));    %求得系数向量
Hden=abs(sym2poly(Hd));

%计算频率特性
[Hh,Hw]=freqs(Hnum,Hden,500);
Hh1=abs(Hh);    %幅频特性
Hw1=angle(Hh);    %相频特性

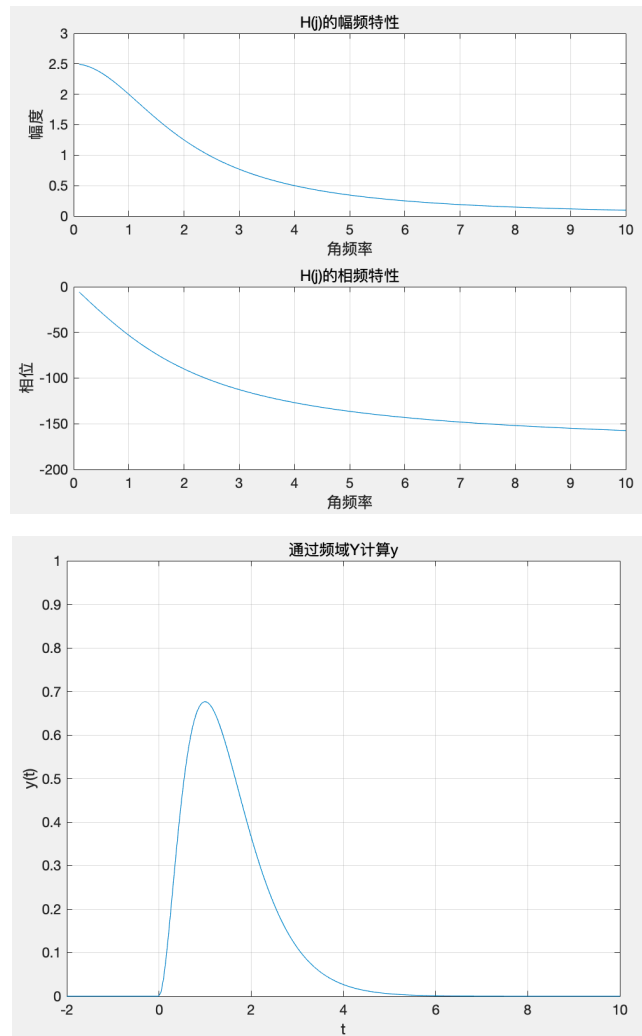
%绘制系统的幅频特性和相频特性
subplot(2,1,1);
plot(Hw,Hh1);
grid on;
xlabel('角频率');
ylabel('幅度');
title('H(j)的幅频特性');
axis([0 10 0 3]);

subplot(2,1,2);
plot(Hw,Hw1*180/pi);
grid on;
xlabel('角频率');
ylabel('相位');
title('H(j)的相频特性');

%绘制输入响应
x=exp(-2*t)*heaviside(t);
X=fourier(x);
Y=X*H;    %频域相乘
y=ifourier(Y);    %傅立叶反变化
figure(2)
ezplot(y,[-4,20]);
axis([-2 10 0 1]);
grid on;
title('通过频域Y计算y');
xlabel('t');
ylabel('y(t)');

```

输出结果：



心得体会：

本课题要求实现连续系统的输入响应。主要的思路是利用fourier()函数，求出系统单位冲击响应和输入信号的傅里叶变化，再通过：

$$Y(\omega) = H(\omega)X(\omega)$$

求出系统输出频域的表达式，最后通过傅里叶反变化就能求出系统输出的时域表达式y(t)。

利用函数abs()和angle(), 得到了上面两幅图，分别表示系统的幅频特性和相频特性。最后设置合适的取值区间，利用函数ifourier()傅里叶反变化得到了系统输出的时域信号。

例4-16：

源代码：

```
% 巴特沃斯滤波器设计
close all;clear;clc;
Wn=400;      %设置通带截止频率
[b,a]=butter(3,Wn,'low','s');

[H,F]=freqs(b,a);
plot(F,20*log10(abs(H)));    %转化为对数的形式
xlabel('频率');
ylabel('幅值');
title('低通滤波器');
axis([0 800 -30 5]);
grid on;
Hs=tf(b,a)
```

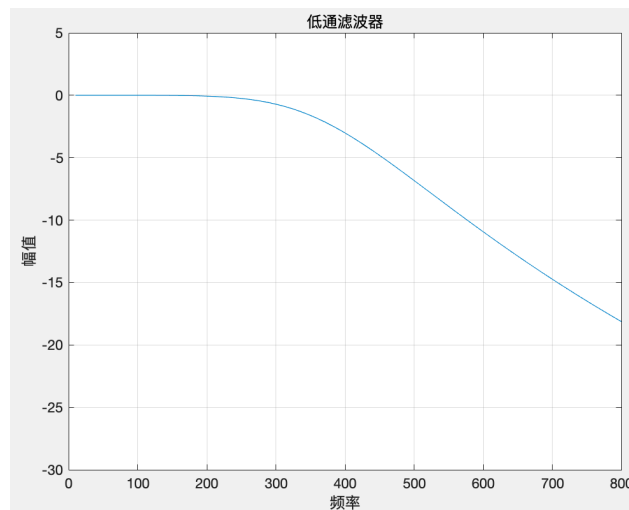
输出结果：

```
Hs =

          6.4e07
-----
s^3 + 800 s^2 + 3.2e05 s + 6.4e07

Continuous-time transfer function.

>>
```



心得体会：

本课题要求实现设计巴特沃斯低通滤波器。在matlab中，已知参数通带截止频率、阻带截止频率、通带波动，阻带最小衰减的情况下，可以使用buttord()函数实现巴特沃斯滤波器。在实验中，我们设置通带截止频率为400Hz，阶数为3。可以这么设计参数：[b,a]=butter(3,Wn,'low','s')

如图输出了模拟滤波器系统函数和巴特沃斯滤波器的幅频特性曲线。可以看到在巴特沃斯低通滤波器的幅频特性曲线中，小于截止频率400Hz前幅值较平稳，信号能够尽可能无损的通过，在f=400Hz这点相比之前平稳的幅值下降了3dB，在这之后幅值迅速下降，对于信号有较好的阻隔作用。

例4-18:

源代码:

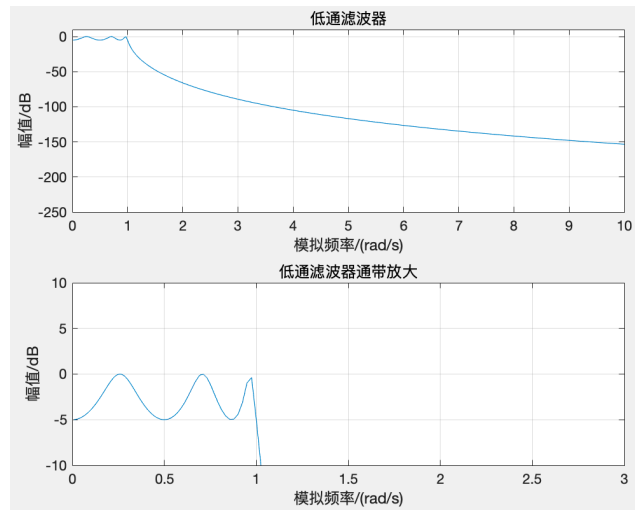
```
%设计切比雪夫滤波器
close all;clear;clc;
[z,p,k]=cheblap(8,4);    %设置切比雪夫参数
[z,p,k]=cheblap(6,5);
[num,den]=zp2tf(z,p,k);
[H,W]=freqs(num,den);
%绘制整个滤波器幅频特性曲线
subplot(2,1,1);
plot(W,20*log10(abs(H)));
xlabel('模拟频率/(rad/s)');
ylabel('幅值/dB');
title('低通滤波器');
axis([0 10 -250 10]);
grid on;
%绘制放大的滤波器幅频特性曲线
subplot(2,1,2);
plot(W,20*log10(abs(H)));
xlabel('模拟频率/(rad/s)');
ylabel('幅值/dB');
title('低通滤波器通带放大');
axis([0 3 -10 10]);
grid on;
%输出系统函数
Hs=tf(num,den)
```

输出结果:

```
Hs =

          0.02125
-----
s^6 + 0.4105 s^5 + 1.584 s^4 + 0.4865 s^3 + 0.6323 s^2 + 0.1124 s + 0.03779

Continuous-time transfer function.
```



心得体会：

本课题要求实现设计切比雪夫低通滤波器。在matlab中可以使用che1ap()、cheb1ord()、cheby1()等函数实现切比雪夫滤波器。在实验中，设置通带波纹为4dB，阶数为6。可以这么设计参数： $[z,p,k]=\text{cheb1ap}(6,5)$

如图输出了模拟滤波器系统函数和切比雪夫滤波器的幅频特性曲线。可以看到在切比雪夫低通滤波器的幅频特性曲线中，小于截止频率1rad/s 前幅值存在波动且处于高位，信号能够尽可能无损的通过，在这之后幅值迅速下降，对于信号有较好的阻隔作用。并且放大了低通滤波器的通带部分，可以看到滤波器的幅值波动不超过5dB。

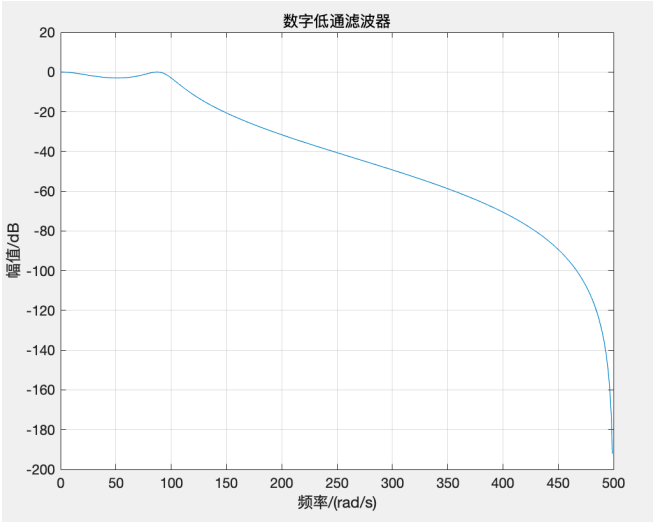
例4-23：

源代码：

```
%设计低通滤波器（切比雪夫）
close all;clear;clc;
%频率归一化
Wp=100/500;
Ws=200/500;
Rp=3;    %通带波纹
Rs=30;   %阻带衰减
[n,Wp]=cheblord(Wp,Ws,Rp,Rs);
[b,a]=cheby1(n,Rp,Wp);
[H,F]=freqz(b,a,512,1000);
plot(F,20*log10(abs(H)));
xlabel('频率/(rad/s)');
ylabel('幅值/dB');
title('数字低通滤波器');
axis([0 500 -200 20]);
grid on;
Hz=tf(b,1,1/1000,'Variable','z^-1')
```

输出结果：

```
Hz =  
  
0.006609 + 0.01983 z^-1 + 0.01983 z^-2 + 0.006609 z^-3  
  
Sample time: 0.001 seconds  
Discrete-time transfer function.
```



心得体会：

本课题中需要实现数字滤波器，常见的有无限冲击响应（IIR）数字滤波器和有限冲击响应（FIR）数字滤波器两种，在matlab中经常使用butter()、cheb1ord()、cheby2()等函数实现两种数字滤波器。在本题中，我们需要实现通带波纹小于3dB、阻带衰减30dB的低通滤波器。于是我们选用cheb1ord()函数实现滤波器的设计。在设计过程中，需要注意的是，我们需要首先对通带截止频率和阻带截止频率归一化，利用[H,F]=freqz(b,a,512,1000)求取滤波器的频率特性，合理设置取值区间绘制幅频特性曲线。

如图输出了模拟滤波器系统函数和设计的低通滤波器的幅频特性曲线。可以看到在低通滤波器的幅频特性曲线中，通带截止频率为100Hz，在阻带频率200Hz到500Hz之间的最小衰减为30dB。

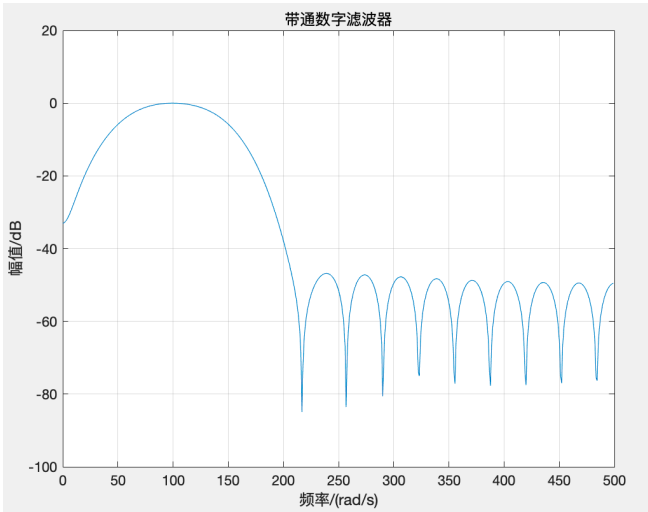
例4-26：

源代码：

```
close all;clear;clc;  
%b=fir1(48,[0.35,0.65]);  
b=fir1(30,[0.1,0.3]);  
[H,F]=freqz(b,1,512,1000);  
plot(F,20*log10(abs(H)));  
xlabel('频率/(rad/s)');  
ylabel('幅值/dB');  
title('带通数字滤波器');  
axis([0 500 -100 20]);  
grid on;  
Hz=tf(b,1,1/1000,'Variable','z^-1')
```

输出结果：

```
Hz =  
  
0.003471 + 0.003221 z^-1 + 0.001499 z^-2 - 0.001654 z^-3 - 0.003435 z^-4 + 2.471e-18 z^-5 + 0.007193 z^-6  
+ 0.007269 z^-7 - 0.01367 z^-8 - 0.05693 z^-9 - 0.1002 z^-10 - 0.1061 z^-11 - 0.04947 z^-12  
+ 0.05676 z^-13 + 0.1611 z^-14 + 0.2045 z^-15 + 0.1611 z^-16 + 0.05676 z^-17 - 0.04947 z^-18  
- 0.1061 z^-19 - 0.1002 z^-20 - 0.05693 z^-21 - 0.01367 z^-22 + 0.007269 z^-23 + 0.007193 z^-  
-24 + 2.471e-18 z^-25 - 0.003435 z^-26 - 0.001654 z^-27 + 0.001499 z^-28 + 0.003221 z^-29 + 0.003471 z^-30
```



心得体会：

本课题要求实现设计一个30阶的FIR带通滤波器，带通频率为 $0.1 \leq \omega \leq 0.3$ ，且采样频率为1000Hz。设计FIR数字滤波器的方法有很多，最常用的是具有线性相频特性的窗函数法。在Matlab中提供了基于窗函数法设计的fir1()和fir2()函数，可用于标准带通滤波器设计，包括低通、高通、带通和带阻数字滤波器。在本课题中可以使用fir1()函数。

FIR 滤波器具有以下主要优点：可以具有精确的线性相位，始终稳定，设计方法通常是线性的，可以在硬件中高效实现，滤波器启动瞬态具有有限持续时间。但是缺点也很明显，要达到同样的性能水平，其所需阶数远高于 IIR 滤波器。相应地，这些滤波器的延迟通常比同等性能的 IIR 滤波器大得多。

上图是输出结果，包含了模拟滤波器系统函数和带通数字滤波器的幅频特性曲线。对比更高阶数的FIR滤波器，能使得通带更加的平稳，阻带衰减的更快，但是在实际工程中，乘法器的位数会随滤波器的阶数的增加而增加，从而硬件规模将变得十分庞大。因此太高阶数的FIR滤波器，不利于实现。