


实验一

信号的时域描述与运算





目 录

1

funtool命令

2

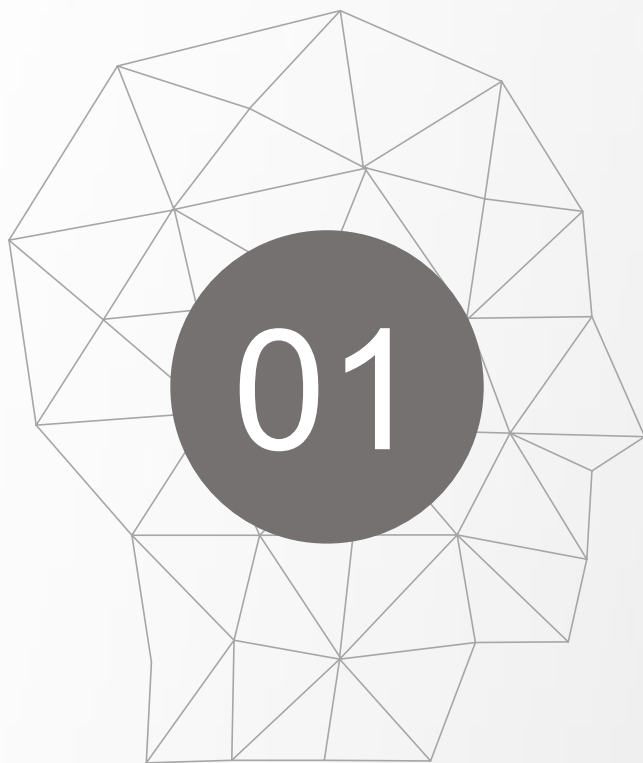
matlab符号函数工具包

3

数值计算

4

符号运算

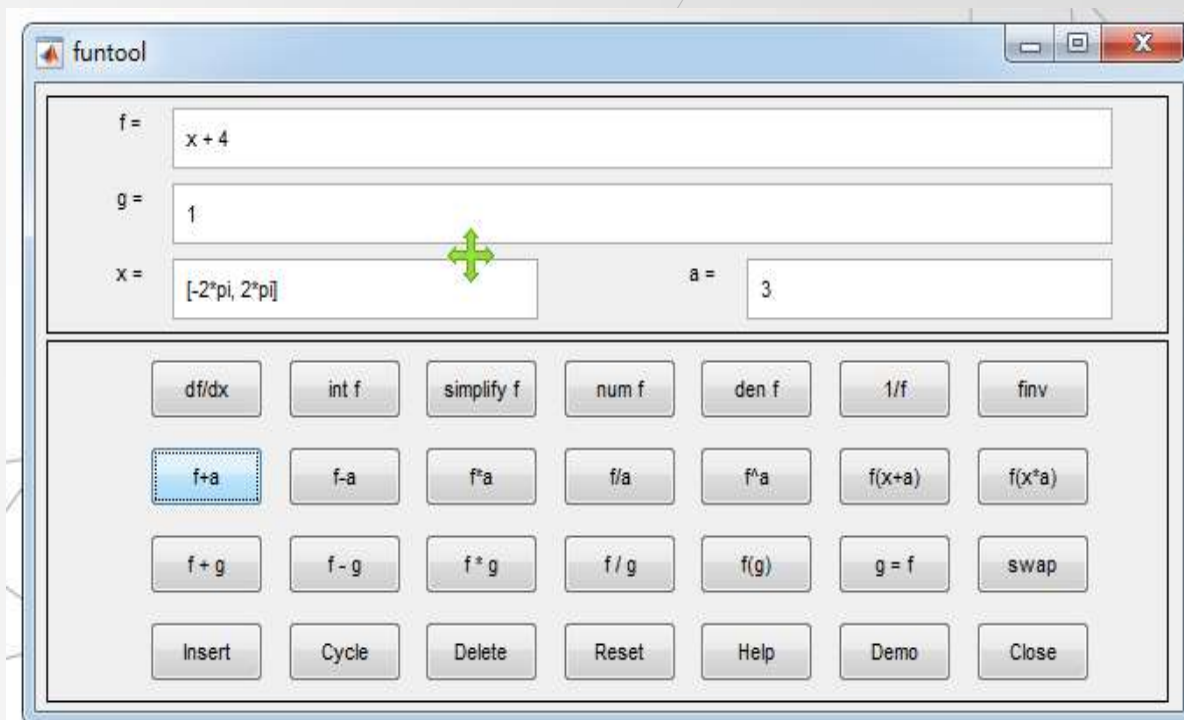


funtool命令

如何使用funtool命令

◆ 第一步：双击matlab软件图标，打开matlab软件，可以看到matlab软件的界面，输入“funtool”，点击enter键，打开函数图像计算器

◆ 第二步：点击funtool窗口中的第二行按钮的“f+a”，表示计算 $f(x)+a$ 。左上角的窗口会显示函数图像。



如何使用funtool命令

- ◆ 第三步，点击funtool窗口中的第二行按钮的“f-a”，表示计算 $f(x)-a$ 。左上角的窗口会显示函数图像。
- ◆ 第四步，依次点击funtool窗口中的第二行按钮的“f*a”、‘f/a’、‘f^a’等。左上角的窗口会显示相应的函数图像。





matlab符号函数工具包

什么是符号运算?

在数学应用在，常常需要做求导数、极限、微分的运算，matlab称这些运算为符号运算，通过调用符号函数工具箱来实现。matlab的符号函数工具箱包括微积分运算，解方程，线性代数，化简和代换等运算。



符号运算的常用功能

◆ 解方程：

(1) syms函数：

调用格式：syms var1 var2 var3...

功能：用于生成多个符号对象

注意：变量之间使用空格

(2) solve函数：

调用格式：S=solve (f1,f2,f3... x1,x2...)

功能：求方程f关于变量x的解

```
>>A=sym(' [a,b;c,d] ')\nA =\n[ a, b]\n[ c, d]\n>>syms a b c d\nA=[a b;c d]\nA =\n[ a, b]\n[ c, d]
```

```
>> syms x\neqn = sin(x) == 1;\nsolx = solve(eqn,x)\n\nsolx =\n\npi/2
```


符号运算的常用功能

◆ 解微分方程：

dsolve函数：

调用格式：dsolve('S', 's1', 's2', ..., 'x')

功能：解微分方程组S，初始条件为s1, s2..., 自变量为x.

注意：① 若不指定自变量为x，则默认自变量为t

② **Dy**表示微分，在**D**后面带有数字，则表示多阶导数，如**D2y**

③ 如果不指定初值，则最后的结果会包含常数项，如**C1,C2**

Solve the equation $\frac{dy}{dt} = ay$ with the condition $y(0) = 5$.

```
>> syms y(t) a
eqn = diff(y,t) == a*y;
cond = y(0) == 5;
ySol(t) = dsolve(eqn, cond)

ySol(t) =

5*exp(a*t)
```



数值计算功能

数值计算的功能

数值计算【numerical computation】
有效使用数字计算机求数学问题近似解
的方法与过程。与符号计算相比，数值
计算在科研和工程中应用更加广泛。



数值计算的常用功能

◆ 求近似数值积分。

trapz函数：

调用格式：S=trapz(x, y)

功能：求函数y关于x的积分。

```
>> X = 0:pi/100:pi;  
Y = sin(X);  
Q = trapz(X, Y)  
  
Q =  
  
1.9998
```

◆ 求近似导数

eps函数：表示的是一个数可以分辨的最小精度。

eps(N) 其中N表示数字N的最小精度，

当N越大时，eps(N)越大，

eps不是一个固定的值。

当它越靠近0时eps越小。eps的默认精度为

```
A>> eps  
  
ans =  
  
2.2204e-16
```

```
>> d = eps(10.0)  
  
d =  
  
1.7764e-15
```

找出从10.0到下一个最大的双精度数的距离



符号运算

符号计算是对未赋值的符号对象（常数、变量、表达式）进行运算和处理。

符号表达式的建立

1、创建符号常量

符号常量是不含变量的符号表达式，用sym命令来创建符号常量。

语法：sym('常量')

```
>> a=sym('sin(2)')  
a =  
sin(2)
```

2、(1) 创建符号变量和表达式

sym('表达式')

(2) 创建多个符号变量

syms('a','b','c','x')

或syms a b c x

```
>>A=sym(' [a,b;c,d]')  
A =  
[ a, b]  
[ c, d]  
>>syms a b c d  
A=[a b;c d]  
A =  
[ a, b]  
[ c, d]
```

符号表达式的建立

3、符号级数

(1) symsum函数（符号表达式的级数求和）

语法：symsum(s, x, a, b)

说明：x为自变量，s为符号表达式，[a, b]为参数x的取值范围。

(2) taylor函数

语法：taylor(F, x, n)

说明：x为自变量，F为符号表达式，对F进行泰勒级数展开至n项，n省略则默认展开前五项。

```
>> syms k
S1 = symsum(k^2, k, 0, 10)

S1 =

385
```

```
>> syms x
taylor(exp(x))

ans =

x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1
```

实验二

连续时间信号的时域运算





目 录

1 基本信号时域表示

2 平移、翻转、尺度变换

3 卷积 (conv函数)



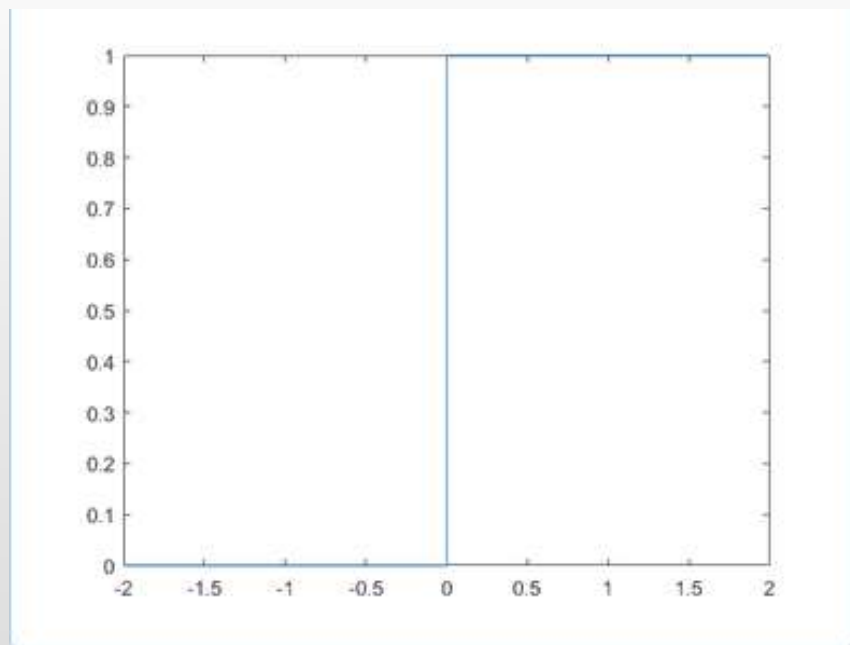
基本信号时域表示

用matlab实现如下信号表示

1、用 matlab 实现如下信号表示：单位阶跃信号（heaviside 函数），采样函数（sinc 函数），门函数（rectpuls 函数），三角脉冲（tripuls 函数），周期方波（square 函数），锯齿波（sawtooth 函数）。

用matlab实现如下信号表示

1.单位阶跃信号 (heaviside函数)



程序主要思想：

```
syms x
```

```
fplot(heaviside(x), [-2, 2]);
```

注：fplot函数

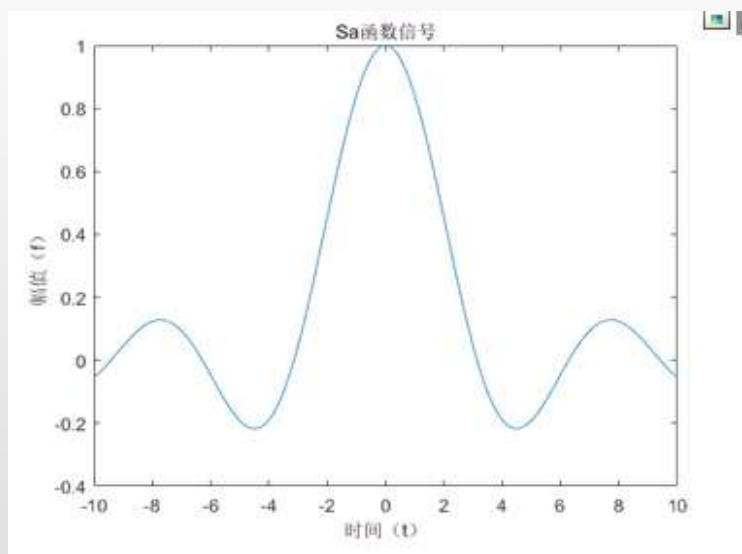
调用格式：fplot(y,[a,b]) %精确绘图

fplot与plot的区别：

plot针对任意变量x和y都可以绘制，而fplot则需要实现定义函数，用function命令定义函数，然后再fplot命令中引用函数的名字。

用matlab实现如下信号表示

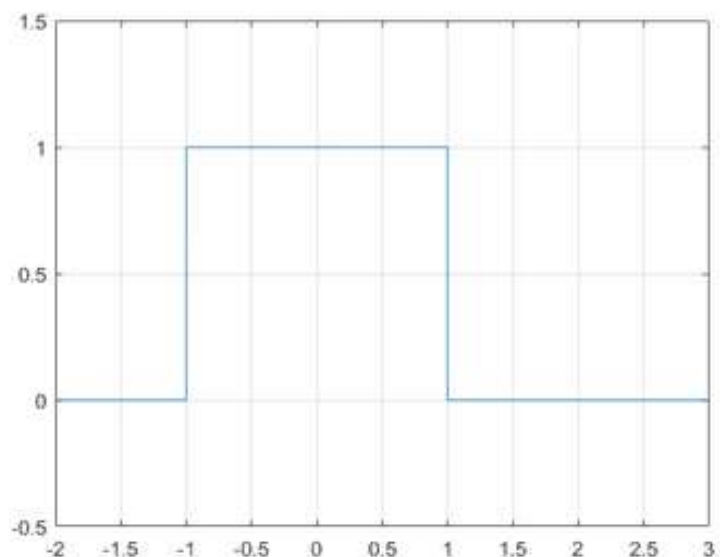
2.采样函数（sinc函数）



```
t= -10:1/500:10;  
x=sinc(t/pi);  
plot(t,x);  
title('Sa函数信号');  
xlabel('时间 (t) ');  
ylabel('幅值 (f) ');
```

用matlab实现如下信号表示

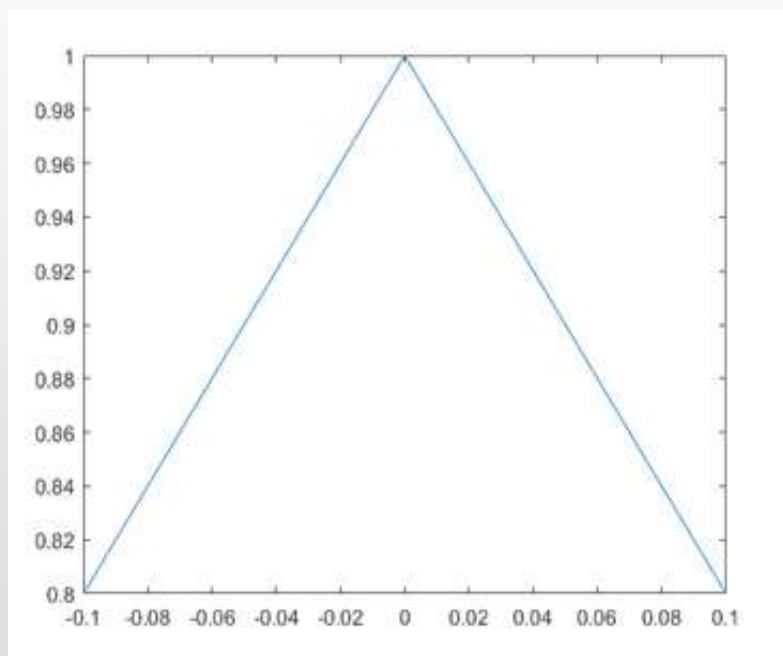
3.门函数 (rectpuls函数)



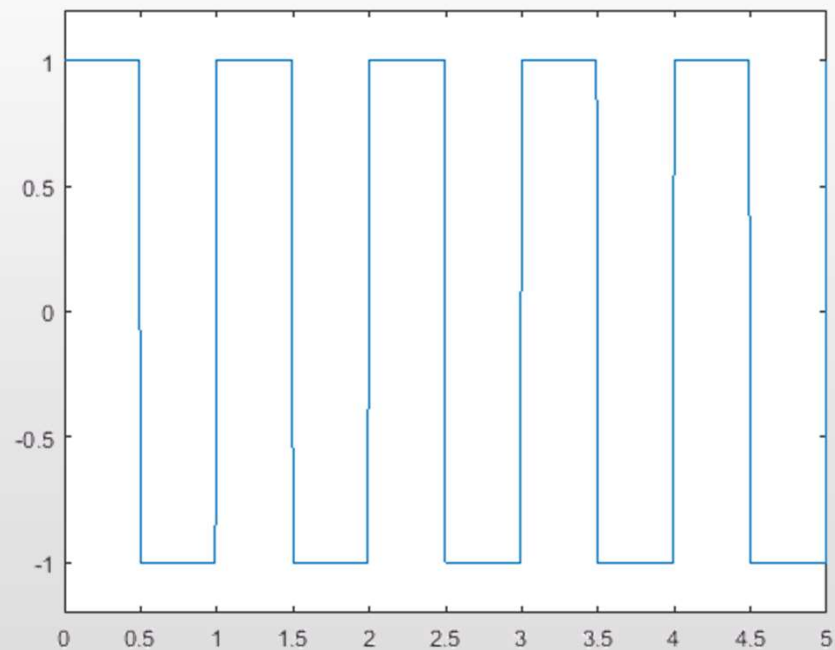
```
width=2;           %设置宽度
t=-2:0.001:3;      %横坐标的取值
ft=rectpuls(t,width); %调用格式为rectpuls(x,y)
plot(t,ft);
grid on;           %显示网格
ylim([-0.5 1.5]); %ylim(limits) 设置当前坐标区或图的 y
                  坐标轴范围。将 limits 指定为 [ymin ymax] 形式的二元素向量,
                  其中 ymax 大于 ymin。
```

用matlab实现如下信号表示

4.三角脉冲 (tripuls函数)

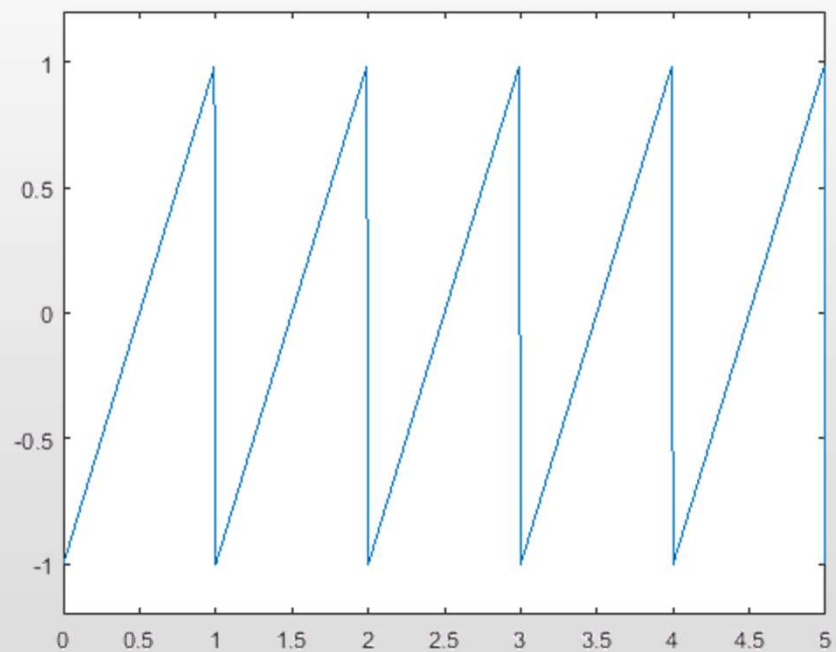


5.周期方波 (square函数)



用matlab实现如下信号表示

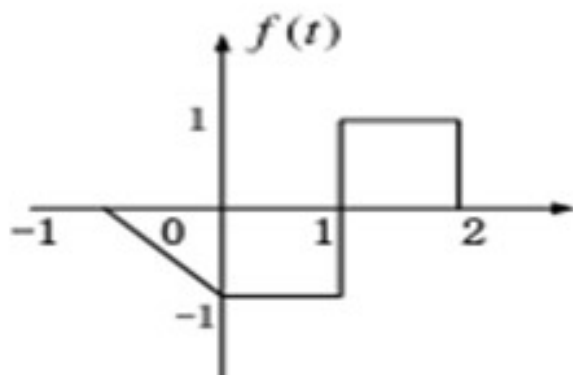
6. 锯齿波 (sawtooth函数)





平移、翻转、尺度变换

2、利用 matlab 编程实现如下图所示信号 $f(t)$ 的变化： $f(-t)$ 、 $f(t+2)$ 、 $f(-t+2)$ 和 $f(2-2t)$ ，画出相应图形。

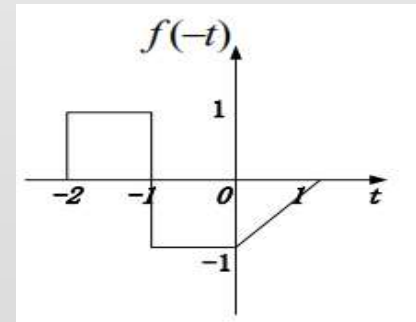
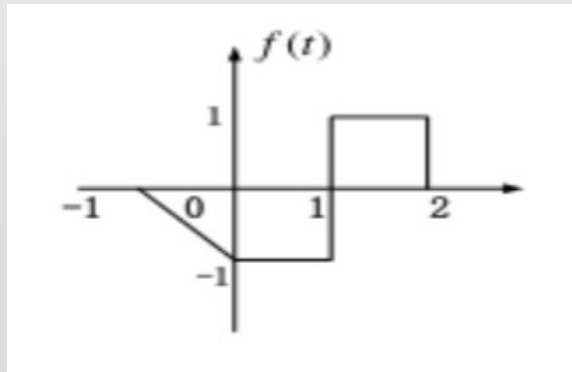


变换步骤： (1) 翻转 → 展缩 → 平移
(2) 翻转 → 平移 → 展缩
(3) 展缩 → 翻转 → 平移
.....



以 $f(t)$ 翻转为 $f(-t)$ 为例

```
t0=-5:0.01:5;           %定义横坐标的取值范围以及递进区间
t=t0;                    %将t0赋值给t
ft=(-t-1).*(heaviside(t+1)-heaviside(t))-(heaviside(t)-heaviside(t-1))+(heaviside(t-1)-heaviside(t-2));
figure(1)
subplot(1,4,1)          %subplot (m,n,i) 是指把图形窗口分为m*n个子图, 并在第i个子图中画图
plot(t0,ft);
xlabel('t');
ylabel('f(t)');
hold on;
t1= -t;                  %翻转, 将t翻转为-t, 定义为t1
t=t1;                    %将翻转后的新变量赋值给t
ft=(-t-1).*(heaviside(t+1)-heaviside(t))-(heaviside(t)-heaviside(t-1))+(heaviside(t-1)-heaviside(t-2));
figure(1)
subplot(1,4,2)
plot(t0,ft);
xlabel('t');
ylabel('f(-t)');
```



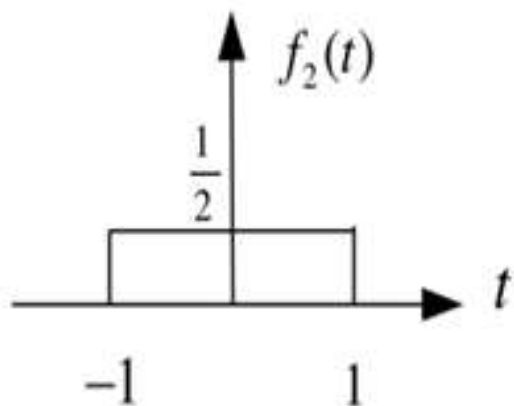
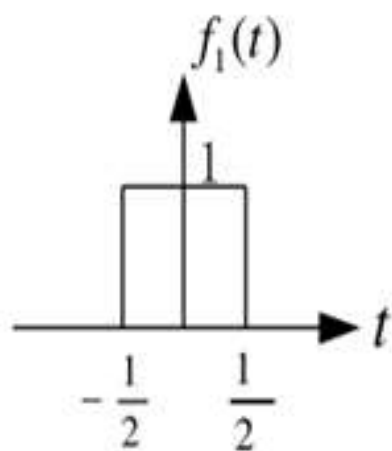


卷积 (conv函数)

3、利用 matlab 实现连续时间信号的卷积图解法。

— 编程实现卷积积分图解法，求如图所示两矩形脉冲信号的卷积

$$f(t) = f_1(t) * f_2(t)$$



编程思路：

- (1) 首先我们要设定 t 的取值范围，对不同的信号我们 设定的 t 的取值是不同的，大家可以根据具体的信号来取值。例如 $t_1 = -2:0.01:3$
- (2) 同第2题，用阶跃函数来生成矩形波信号 f_1 和 f_2 ；
- (3) 对 f_1 和 f_2 进行卷积计算，使用的命令为conv函数，
调用格式： $y = \text{conv}(f_1, f_2)$
- (4) 分别画出 f_1, f_2, y 的图形。
- (5) 如果大家出来的卷积 y 在图中的纵坐标非常大，在图中不好观察，大家可以将 y 乘以一个 步长 T ，我们关心的是图形的形状，增益是可以忽略的。

实验三

连续时间系统的时域和s域分析





涉及的知识点:

- ✓ 冲激响应 (impulse函数)
- ✓ 阶跃响应 (step函数)
- ✓ 零状态响应 (lsim,conv函数)
- ✓ Simulink仿真
- ✓ 响应分析

1、利用 Matlab 计算并绘制以下系统的单位冲激响应和单位阶跃响应。

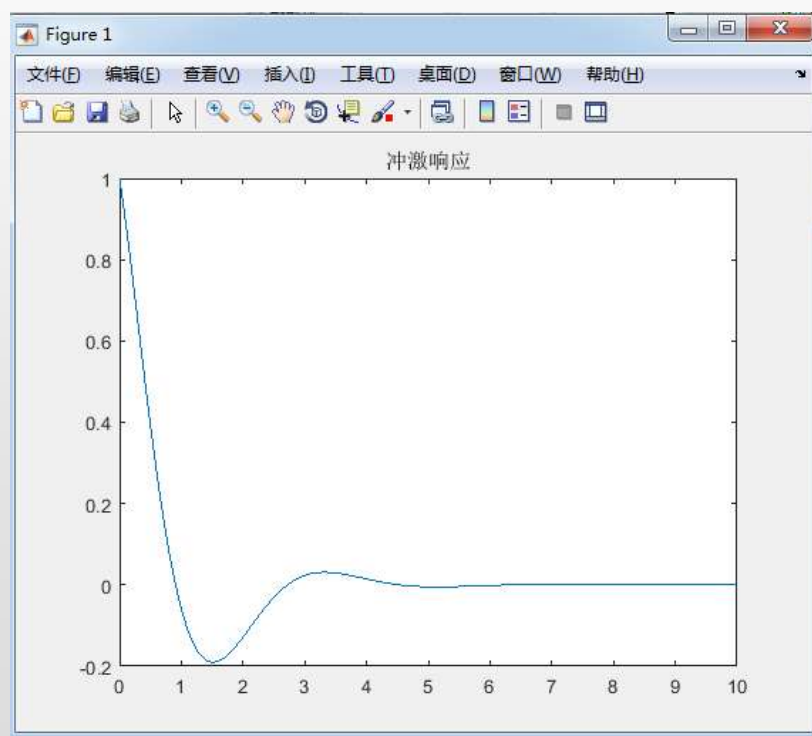
$$2y''(t) + 7y'(t) + 3y(t) = f(t); \quad y''(t) + 3y'(t) + 2y(t) = f'(t) + f(t)$$

分析：

- 单位冲激响应 (impulse函数)
- 单位阶跃响应 (step函数)
- 用tf函数建立传递函数模型,

例1.利用Matlab计算并绘制以下系统的单位冲激响应和单位阶跃响应。

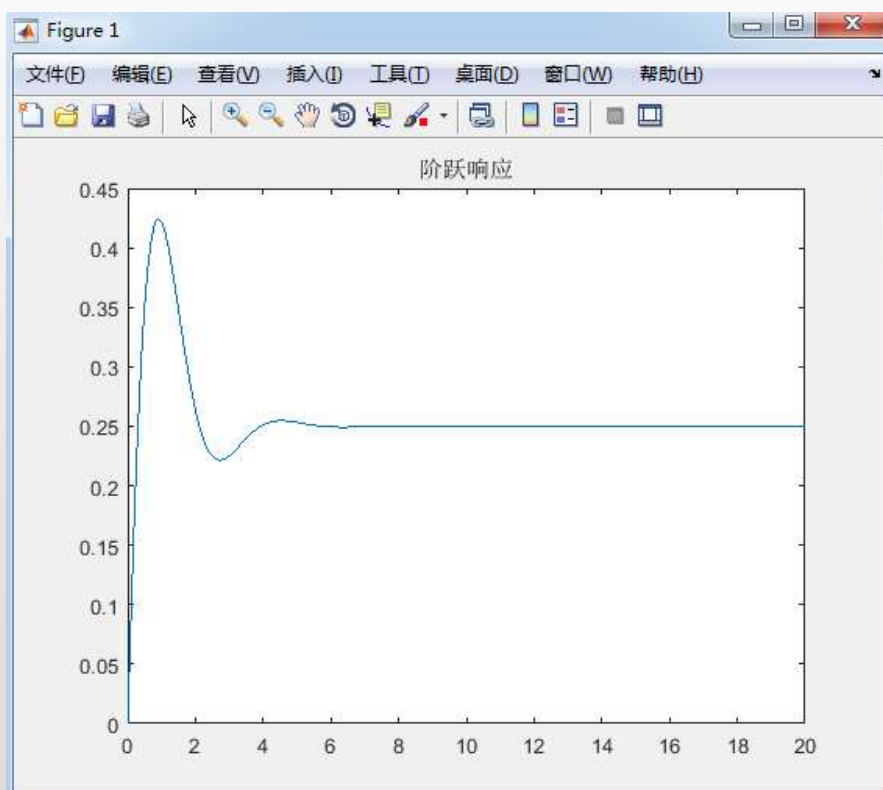
$$y''(t)+2y'(t)+4y(t)=f'(t)+f(t)$$



```
sys=tf([1,1],[1,2,4]);  
t=0:0.1:10;  
y=impz(sys,t);  
plot(t,y);  
title('冲激响应')
```

例1.利用Matlab计算并绘制以下系统的单位冲激响应和单位阶跃响应。

$$y''(t)+2y'(t)+4y(t)=f'(t)+f(t)$$



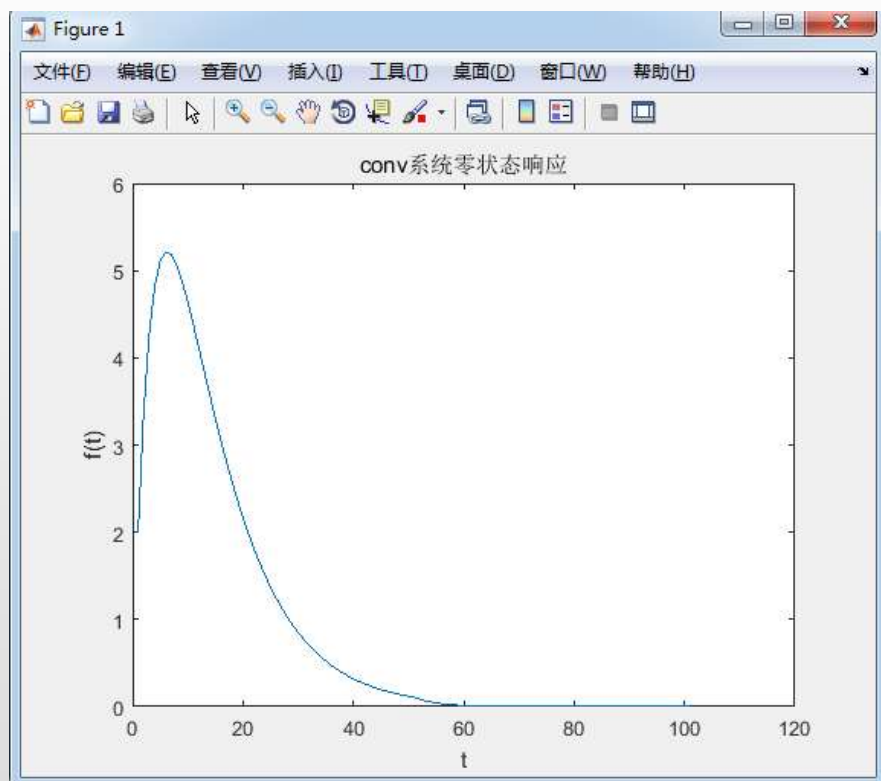
```
sys=tf([1,1],[1,2,4]);  
t=0:0.1:20;  
y=step(sys,t);  
plot(t,y);  
title('阶跃响应')
```

2、已知某 LTI 系统的单位冲激响应为 $h(t) = (e^{-t} + e^{-2t})u(t)$ ，指定输入激励为 $f(t) = e^{-t}u(t)$ ，分别利用 Matlab 中的 lsim 函数和 conv 函数计算系统的零状态响应。

注意：

- 1.使用lsim函数
- 2.使用conv函数

例2 已知某LTI系统的单位冲激响应为 $h(t) = (e^{-2t} + e^{-3t})u(t)$ ，输入激励为 $f(t) = e^{-t}u(t)$ ，用Matlab 中的conv函数计算系统的零状态响应。



```
>> t=0:0.1:5;  
h=exp(-2*t)+exp(-3*t);  
f=exp(-t);  
yf=conv(h, f);  
figure(1);  
plot(yf);  
title('conv系统零状态响应');  
xlabel('t');  
ylabel('f(t)');
```

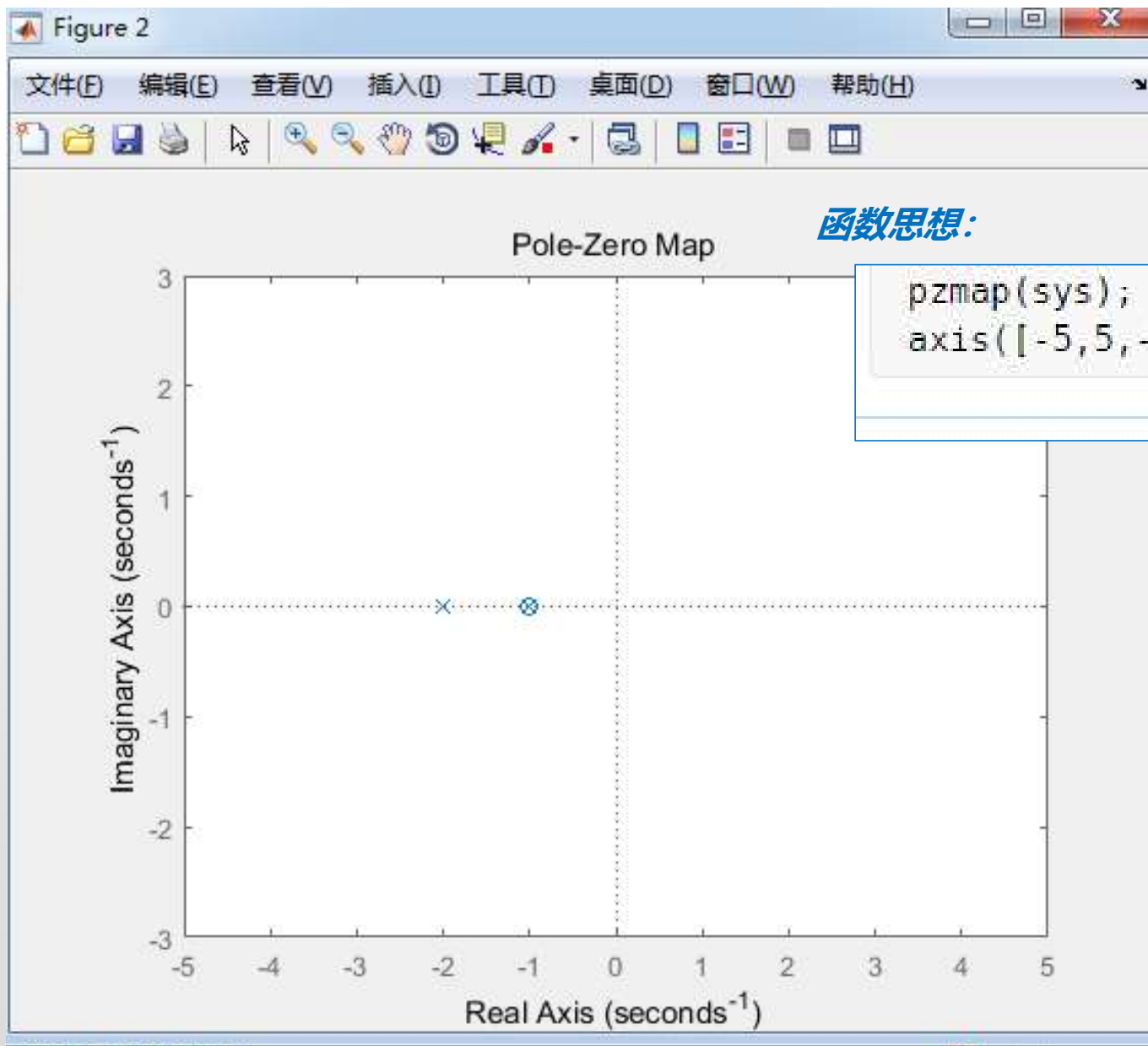
已经给出了conv函数的用法，那关于利用lsim函数计算系统的零状态响应，大家下去查找lsim函数的用法。



3、求 $y''(t) + 3y'(t) + 2y(t) = f'(t) + f(t)$ 的系统函数并判断系统的稳定性。

对于连续系统：

- I. 先通过拉普拉斯变换求出系统函数 $H(S)$ ，令 $H(S)$ 分母表达式的值为0，求出的值就是系统函数的极点；
 - II. 若系统函数 $H(s)$ 全部极点落在 S 域左平面，则系统稳定
 - III. 若系统函数 $H(s)$ 有极点落在 S 域右半平面，或在虚轴上具有二阶以上的极点，则该系统不稳定。
 - IV. 若系统函数 $H(s)$ 无极点落在 S 域右半平面，但在虚轴上有一阶极点，则该系统临界稳定。
- 注意：系统函数的分子多项式的阶次，不高于分母多项式的阶次。

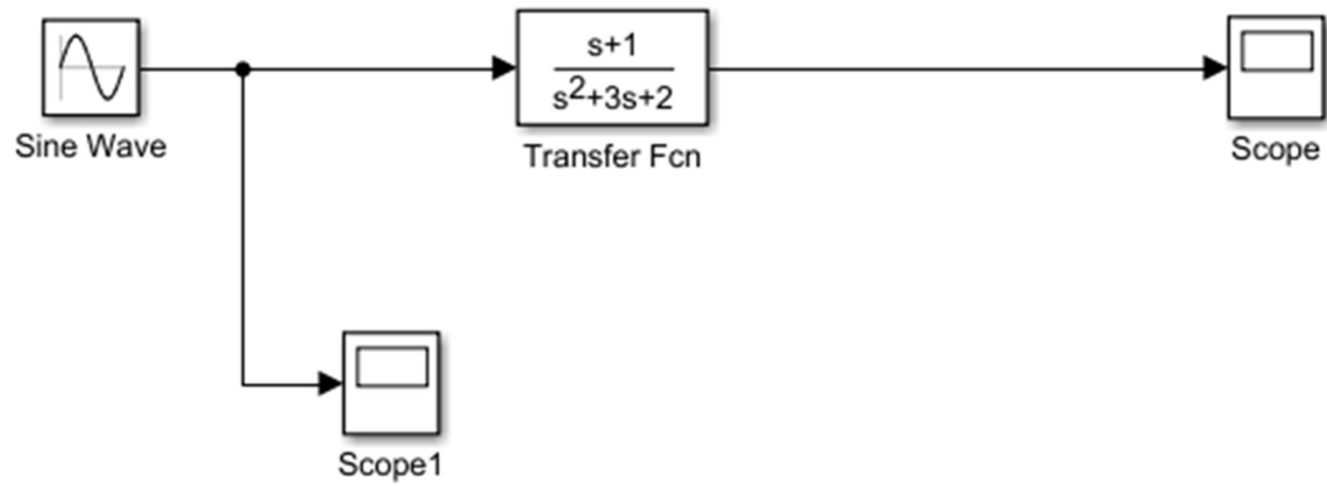


函数思想:

```
pzmap(sys);  
axis([-5,5,-3,3]);
```


4、通过 $y''(t) + 3y'(t) + 2y(t) = f'(t) + f(t)$ 微分方程，进行系统函数的
simulink 系统搭建及响应分析。✦作业里正弦发生器为 $\sin(2x)$

接下来是matlab演示如何使用simulink进行系统搭建



响应分析：

1. 峰值时间
2. 超调量
3. 上升时间
4. 调节时间

```
% 峰值时间tp
sys=tf([0,1],[2,7,3]);
[y,t]=step(sys);%%y是输出变量响应值,t是仿真时间
[Y,k]=max(y);%%Y是峰值, k是响应时间
tp=t(k);
% 超调量
C=dcgain(sys);%%dcgain函数求取系统的终值
[Y,k]=max(y);
sita=100*(Y-C)/C;
```

```
% 上升时间tr  
C=dcgain(sys);%%dcgain函数求取系统的终值  
n=1;  
while y(n)<0.1*C%%输出变量的响应值小于0.1倍的系统终值  
n=n+1  
end  
m=1;  
while y(m)<0.9*C%%输出变量的响应值小于0.9倍的系统终值  
m=m+1;  
end  
tr=t(m)-t(n);%%得出m和n的值，代入仿真时间函数
```

```
% % 调节时间  
[y,t]=step(sys);  
C=dcgain(sys);  
i=length(t);%%length仿真时间的向量长度  
while(y(i)>0.98*C)&(y(i)<1.02*C)  
    i=i-1  
end  
ts=t(i);
```

实验四

基于傅里叶级数展开的有限项波形
和成的波形呈现及误差分析



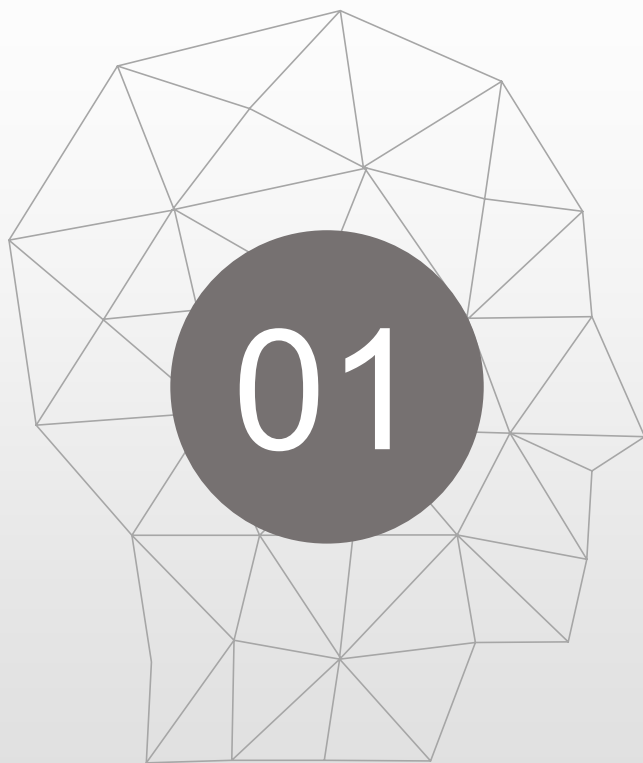


目 录

1 用matlab求函数的正反傅里叶变换

2 quad函数

3 傅里叶级数展开, 误差分析



用matlab求函数的正反傅里叶变换

◆fourier函数:

(1) $F = \text{fourier}(f)$: 他是符号函数 f 的Fouier变换, 默认返回的是关于 w 的函数。

(2) $F = \text{fourier}(f, v)$: 他返回函数 F 是关于符号对象 v 的函数, 而不是默认的 w , 例

如
$$F(v) = \int_{-\infty}^{\infty} f(t) e^{-jvt} dt$$

(3) $F = \text{fourier}(f, u, v)$: 是对关于 u 的函数 f 进行变换, 返回函数 F 是关于 v 的函

数, 即
$$F(v) = \int_{-\infty}^{\infty} f(t) e^{-jvu} du$$

◆ifourier函数:

(1) $f = \text{ifourier}(F)$

(2) $f = \text{ifourier}(F, u)$

(3) $f = \text{ifourier}(F, u, v)$

注: 函数 $\text{fourier}()$ 和 $\text{ifourier}()$ 都是接受由 sym 函数所定义的符号变量或者符号表达式。

1、用 Matlab 求下列问题：

(1) 利用 Matlab 符号运算 `fourier` 函数和 `ifourier` 函数，以符号形式画出门函数的傅里叶正反变换。

(2) 利用 matlab 的 `fft` 函数和 `ifft` 函数求冲激函数的傅里叶正反变换。

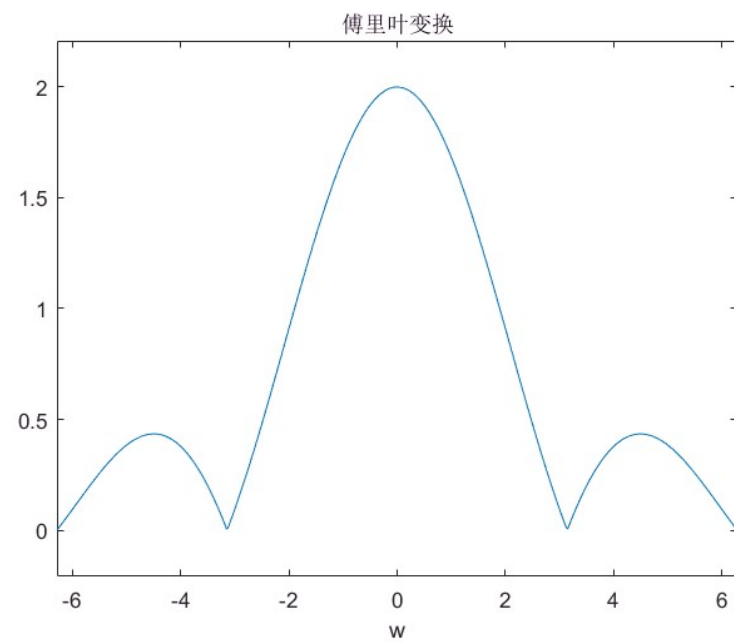
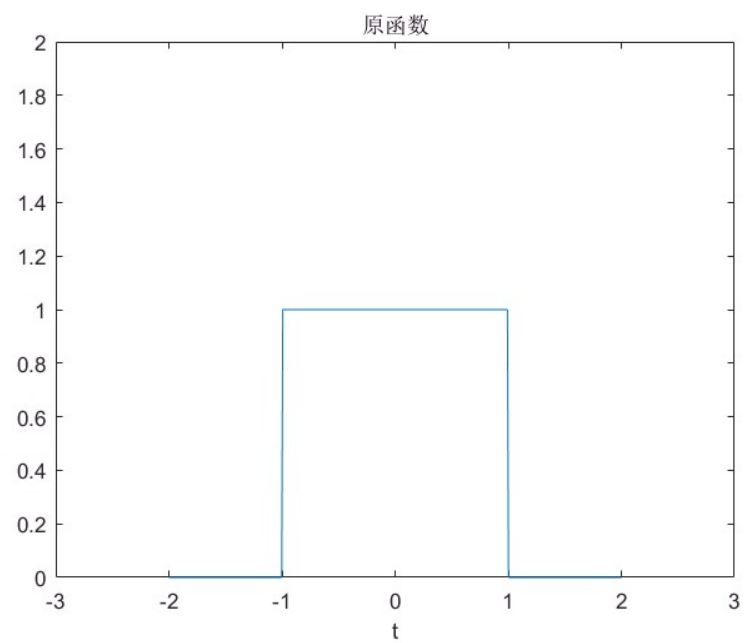
以门函数的傅里叶正变换为例：

```
syms t;           %首先定义一个符号变量t
y=heaviside(t+1)-heaviside(t-1); %画出门函数的图像
ezplot(y,[-2,2]);
axis([-3 3 0 2]); %axis( [xmin xmax ymin ymax] ), 设置当前坐标轴 x轴 和 y轴的限制范围
title('原函数');
F=fourier(y);      %求函数y的傅里叶变换
figure;
ezplot(abs(F));    % abs()对于复数是取模值，对于实数是取绝对值
title('傅里叶变换');
```

`ezplot()`函数：

绘制用符号运算表示法表示的连续信号的波形。

调用格式： `ezplot(f,[xmin,xmax])`



fourier函数和ifourier函数的局限性

- (1) 如果返回函数中有如单位冲激函数等项, 那么用ezplot将无法做图。
- (2) 对某些信号求变换时, 其返回函数可能包含一些不能直接用符号表达的式子, 那么也不能作图。
- (3) $f(t)$ 连续, 但经过抽样所获得的信号是多组离散的数值 $f(n)$,那么无法表示成符号表达式, 那么不能用fourier函数进行处理。

第二问:用IFFT和 FFT函数求单位冲激响应的傅里叶变换思路:

- (1) 首先写出单位冲激信号的表达式 x ;
- (2) 调用 $F=\text{fft}(x)$, 通过画图语句就可以画出图像。



quad函数

2、利用 Matlab 的 quad 函数用数值分析的方法近似计算三角波信号

$$x(t) = \begin{cases} 1-|t| & |t| \leq 1 \\ 0 & |t| > 1 \end{cases} \text{ 的频谱;}$$

```
function y = sf(t,w)
```

```
y=(abs(t)<=1).*(1-abs(t)).*exp(-j*w*t);
```

%这两句要做成函数，放到m文件中

```
w=linspace(-6*pi,6*pi,512); %linspace是Matlab中的均分计算指令,,linspace(x1,x2,N),其中x1为起始值,  
                             x2为终点值, N为元素个数
```

```
N=length(w); %返回到该行数中的最大的值
```

```
X=zeros(1,N); %生成一个1行N列的零矩阵
```

```
for k=1:N
```

```
X(k)=quad(@sf,-1,1,[],[],w(k)); %用来计算积分的函数，在已经建立.m函数时，在调用该函数的时候要加  
                                @符号
```

```
end
```

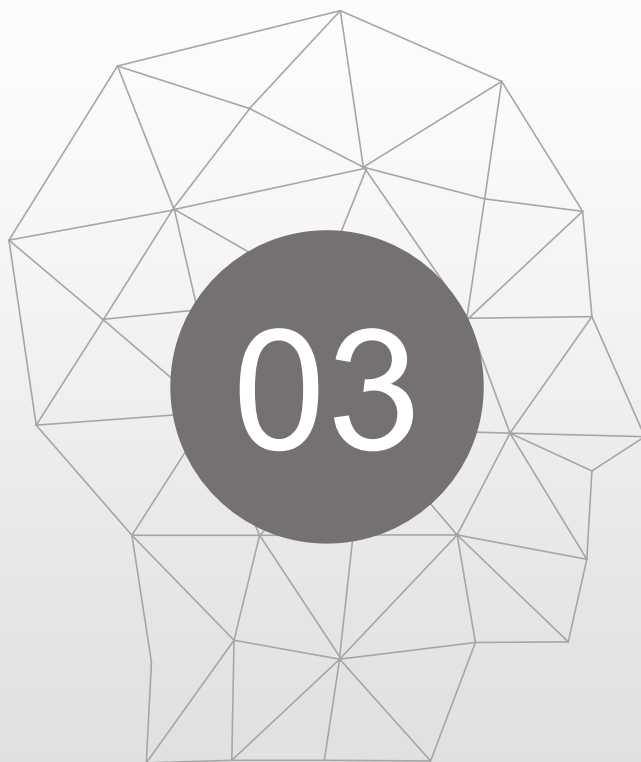
```
subplot(121);
```

```
plot(w,real(X));
```

```
xlabel('\omega');
```

```
ylabel('X(j\omega)');
```

```
title('Spectrum of Triangle wave');
```

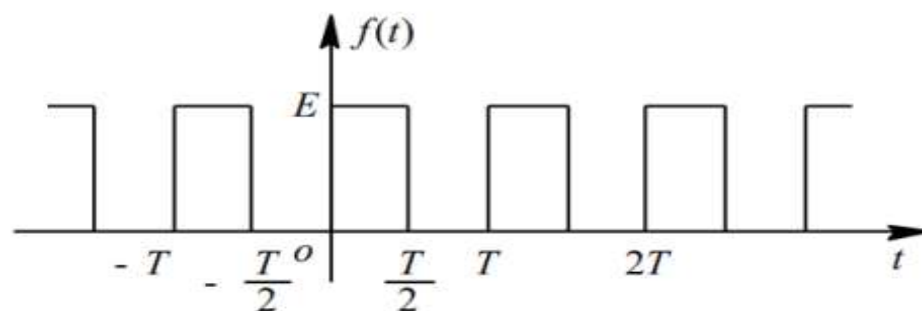



傅里叶级数展开，误差分析

3、利用 Matlab 编程计算如下图所示周期信号 $f(t)$ 的如下问题：

(1) 分别画出其傅里叶级数展开式前三项之和、前十项之和的图形，并分别计算其和原信号之间的均方误差，分析图形和误差变化的原因；其中 $E=1$ 。

(2) 编程画出傅里叶级数的项数 (<100) 和与均方误差之间的关系曲线，并在图中标出均方误差小于等于 0.01 时所需要的项数。



$$\begin{aligned} f(t) &= \frac{E}{2} + \sum_{k=1}^{\infty} \frac{2E}{k\pi} \sin(k\omega_1 t) \\ &= \frac{E}{2} + \frac{2E}{\pi} \left[\sin(\omega_1 t) + \frac{1}{3} \sin(3\omega_1 t) + \frac{1}{5} \sin(5\omega_1 t) + \dots \right] \end{aligned}$$

以画出傅里叶级数的前三项为例。

```
T=20;
```

```
w=2*pi/T;
```

```
t=0:0.01:20;
```

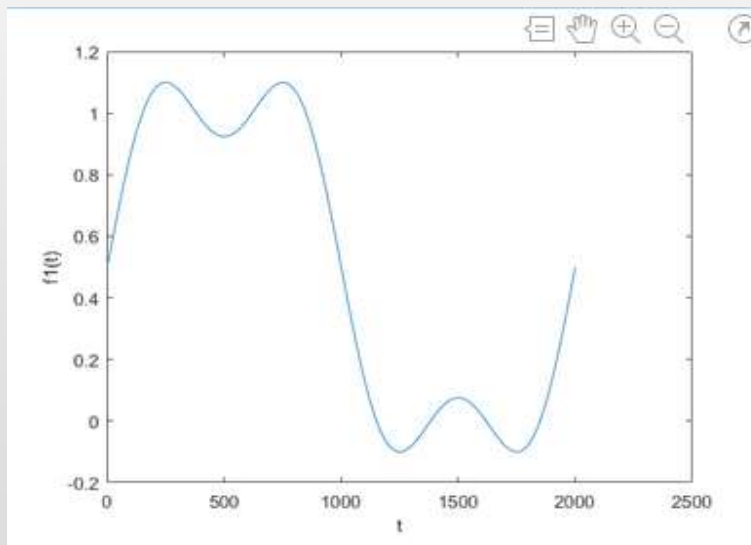
```
f1=1/2+2/pi*sin(w*t)+2/pi*1/3*sin(3*w*t);
```

 %傅里叶级数展开式前三项的和

```
plot(f1);
```

```
xlabel('t');
```

```
ylabel('f1(t)');
```



如何进行均方误差的计算？

根据均方误差的计算公式，可得其均方误差为：

$$\begin{aligned}\overline{\varepsilon^2} &= \frac{1}{T} \left[\int_0^T f^2(t) dt - (a_0)^2 T - \sum_{j=1}^n b_j^2 \frac{T}{2} \right] \\ &= \frac{1}{T} \left[\int_0^{\frac{T}{2}} dt - \left(\frac{1}{2} \right)^2 T - \frac{T}{2} \sum_{j=1}^n b_j^2 \right] \\ &= \frac{1}{4} - \frac{1}{2} \sum_{j=1}^n b_j^2\end{aligned}$$

如何进行均方误差的计算？

当只有直流和基波时：
$$\overline{\varepsilon^2} = \frac{1}{4} - \frac{1}{2} \left(\frac{2}{\pi} \right)^2 = 0.047$$

当取直流、基波和三次谐波时：
$$\overline{\varepsilon^2} = \frac{1}{4} - \frac{1}{2} \left(\frac{2}{\pi} \right)^2 - \frac{1}{2} \left(\frac{2}{3\pi} \right)^2 = 0.0246$$

当取直流、一、三、五次谐波时：
$$\overline{\varepsilon^2} = \frac{1}{4} - \frac{1}{2} \left(\frac{2}{\pi} \right)^2 - \frac{1}{2} \left(\frac{2}{3\pi} \right)^2 - \frac{1}{2} \left(\frac{2}{5\pi} \right)^2 = 0.0165$$

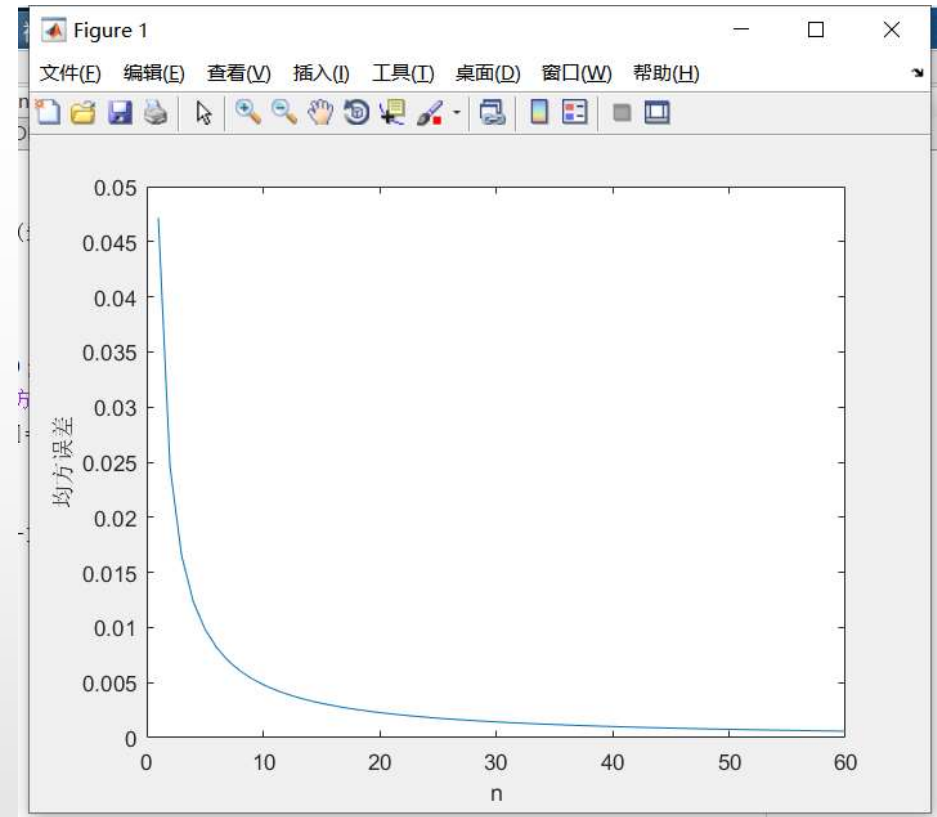
当取直流、一、三、五、七次谐波时：

$$\overline{\varepsilon^2} = \frac{1}{4} - \frac{1}{2} \left(\frac{2}{\pi} \right)^2 - \frac{1}{2} \left(\frac{2}{3\pi} \right)^2 - \frac{1}{2} \left(\frac{2}{5\pi} \right)^2 - \frac{1}{2} \left(\frac{2}{7\pi} \right)^2 = 0.01236$$

以上结果表明，选取的项数越多，合成波形和理想波形之间的均方误差就越小。我们可以依据给定的误差要求，选取适当的谐波数，实现不同周期信号的合成。

用matlab程序来实现均方误差

```
clear;
for i=1:60
a(i)=cumtpt(i);
end
x=1:1:60;
plot(x,a);
xlabel('n');
ylabel('均方误差');
function[s]=cumtpt(n)
s=0.25;                %均方误差的第一项
for i=1:n
a=(2/((2*i-1)*3.14))*(2/((2*i-1)*3.14))*0.5; %均方误差的余项
s=s-a;                %每循环一次产生一个新的s
end
end
```



第二问中的标出均方误差小于等于0.01时所需要的项数

- (1) 首先根据第一问画出前100项的均方误差图。
- (2) 我们可以从图中粗略看出当 n 小于0.01时对应的项数。
- (3) 更精确的看哪一项是具体小于0.01, 我们可以点击工作区的 a , 则每项都会对应一个均方误差的值。

实验五

信号时域采样与恢复







采样定理的理论基础

MATLAB实现连续信号采样与重构的理论基础（一）

1.连续时间信号

连续信号是指自变量的取值范围是连续的，且对于一切自变量的取值，除了有若干个不连续的点以外，信号都有确定的值与之对应。那严格来说，MATLAB并不能处理连续信号，而是用等时间间隔点的样值来近似表示连续信号。当取样时间间隔足够小的时候，这些离散的样值就能较好地近似连续信号。



MATLAB实现连续信号采样与重构的理论基础（二）

2.信号的采样

模拟信号经过A/D变换转换为数字信号的过程称为采样，信号采样后其频谱产生了周期延拓，每隔一个采样频率 f_s ，重复出现一次。为保证采样后信号的频谱形状不失真，采样频率必须大于信号中最高频率成分的两倍，这称之为采样定理。

时域采样定理从采样信号 $f_s(t)$ 恢复原信号 $f(t)$ 必须满足两个条件：

- (1) $f(t)$ 必须是带限信号，其频谱函数在 $|\omega| > \omega_m$ 各处为零；
- (2) 取样频率不能过低，必须 $\omega_s > 2\omega_m$ （或 $f_s > 2f_m$ ），这时频谱不发生混叠。取样频率要足够大，采得的样值要足够多，才能恢复原信号。

注：带限信号是在某个频率区间内有值，在这个区间之外就是0的信号。由于 $sa(t)$ 函数不是严格的带限信号，其带宽可根据一定的精度要求做一近似

MATLAB实现连续信号采样与重构的理论基础 (三)

3.信号的重构

设信号 $f(t)$ 被采样后形成的采样信号 $f_s(t)$,信号的重构是指 $f_s(t)$ 经过内插处理后,恢复出原来信号 $f(t)$ 的过程, 又称信号恢复。若设 $f(t)$ 是带限信号, 带宽为 w_m ,经采样后的频谱为 $F(jw)$ 。设采样频率 $w_s \gg 2w_m$, $F(jw)$ 是以 w 为周期的谱线。

举例:

设 $f(t) = \text{sinc}(t) = \frac{\sin(t)}{t}$, 其 $F(jw)$ 为:

$$F(jw) = \begin{cases} \pi, & |w| < 1 \\ 0, & |w| > 1 \end{cases}$$

即 $f(t)$ 的带宽为 $w_m=1$, 为了由 $f(t)$ 的采样信号 $f_s(t)$ 不失真地重构 $f(t)$, 由时域采样定理得采样间隔 $T_s < \frac{\pi}{w_m} = \pi$, T_s 成为奈奎斯特间隔。

若 $T_s = \pi$ 时, 成为临界采样

若 $T_s = 1.5\pi$ 时, 称为欠采样, 那这时候是低于奈奎斯特采样频率还是高于呢?

选取一个理想低通滤波器与 $F_s(j\omega)$ 相乘，得到的频谱即为原信号的频谱 $F(j\omega)$ 。该滤波器的频率特性 $H(j\omega) = \begin{cases} T_s, & \omega < \omega_c \\ 0, & \omega > \omega_c \end{cases}$ ，(其中 ω_c 即截止频率满足 $\omega_m \leq \omega_c \leq \omega_s/2$)

$$f(t) = h(t) * f_s(t)$$

其中 $f_s(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} f(nT_s) \delta(t - nT_s)$,

$$h(t) = F^{-1}[H(j\omega)] = T_s \frac{\omega_c}{\pi} \text{sa}(\omega_c t)$$

整理可得： $f(t) = f_s(t) * T_s \frac{\omega_c}{\pi} \text{sa}(\omega_c t) = T_s \frac{\omega_c}{\pi} \sum_{n=-\infty}^{\infty} f(nT_s) \text{sa}[\omega_c(t - nT_s)]$,
这个就是信号重构时用到的公式。

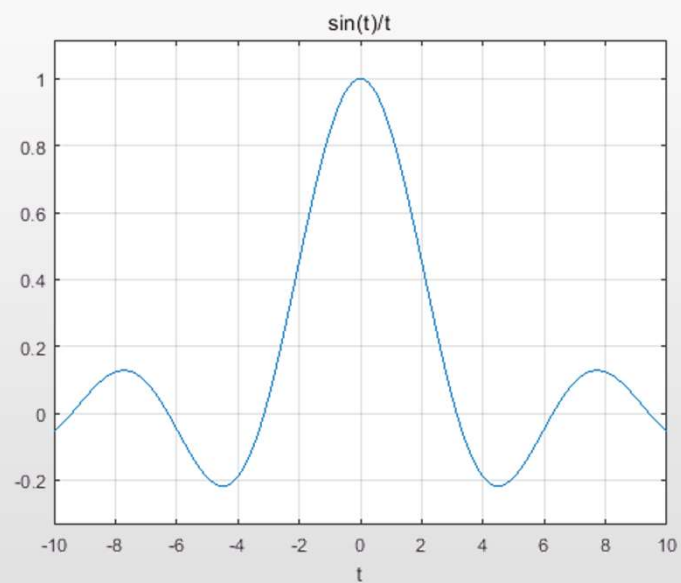


程序讲解

已知信号 $f(t)=Sa(t)$ ，编程解决如下问题： +

(1) 画出其波形图;

```
t=-10:0.01:10;  
f='sin(t)/t';  
ezplot(f,t);  
grid on;
```



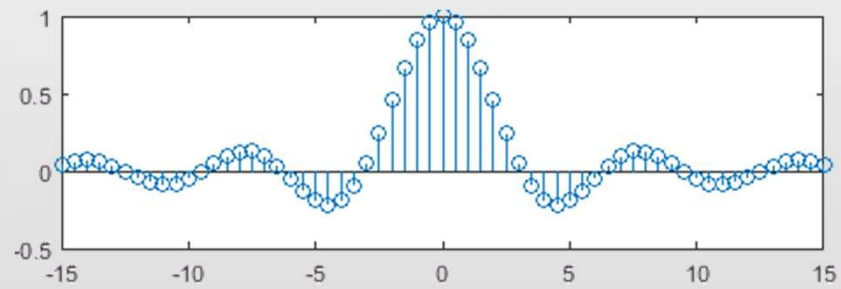
(2) 选择奈奎斯特采样频率对其离散化，画出离散化后的波形图

程序思想：

```
t1=-15:0.5:15;
```

```
f1=sinc(t1/pi);
```

```
stem函数;
```



(3) 选取合适的理想低通滤波器由 $f(t)$ 的采样值恢复 $f(t)$ ，并画出恢复后的波形图；

程序重点：

1. $w_m = ?$, $w_c = w_m$, $T_s = ?$

2. `n=-100:100;`
`nTs=n*Ts;`
`f=sinc(nTs/pi);`
`Dt=0.005;`
`t=-15:Dt:15;`

```
fa=f*Ts*wc/pi*sinc((wc/pi)*(ones(length(nTs),1)*t-nTs'*ones(1,length(t)))));
```

(4) 第一问:

选择低于奈奎斯特采样频率的采样率对其离散化, 画出采样信号的频谱图;

分析:

1. 离散化
2. 采样信号的频谱图

频谱图的程序主要思想:

```
y=fft(f);  
GF=abs(y);  
h=(0:length(y)-1)'*fs/length(y);  
subplot(312);  
plot(h, GF);  
xlabel('频率(Hz)');  
ylabel('幅值');
```

分析:

- 1.恢复原信号
- 2.画出恢复信号与原信号的误差图

(4) 第二问:

选择合适的理想低通滤波器恢复原信号, 画出其波形, 计算其和原信号之间的均方误差;

恢复原信号, 程序重点:

$$\omega_m = ?, \omega_c = \omega_m, T_s = ?$$

求误差, 程序重点:

```
error=abs(fa-sinc(t/pi));
```

(5) 分析并合理解释以上仿真结果。

可以从以下几个方面，结合仿真结果，进行分析

1. 采样定理
2. 离散化
3. 选择低于奈奎斯特采样频率的采样率对其离散化
4. 从离散信号恢复原信号
5. 恢复信号与原信号的误差

实验六

模拟滤波器设计

Butterworth滤波器设计



特点

巴特沃斯滤波器的特点是通频带内的频率响应曲线最大限度平坦，没有纹波，而在阻频带则逐渐下降为零。巴特沃斯滤波器是三大原型模拟低通滤波器之一，其他的模拟滤波器(如高通、带通和带阻滤波器)可以由三大原型低通滤波器通过简单的频率变换来实现。

实现巴特沃斯滤波器的命令包括三个：buttord、buttap和butter

① buttord：用来求巴特沃斯滤波器的阶数

$[N, Wn] = \text{buttord}(Wp, Ws, Rp, Rs, 's')$ ：给定通带边界频率 Wp 和阻带边界频率 $Ws(\text{rad/s})$ 、通带波动 $Rp(\text{dB})$ 和最小阻带衰减 $Rs(\text{dB})$ ，求满足指标的模拟低通滤波器的最低阶数 N 和3 dB边界频率 Wn 。's'表示模拟域。

特点

② buttap: 用来在给定滤波器阶数时, 计算滤波器的零极点和增益因子

[z, p, k] = buttap(N): 给定阶数N时, 计算归一化的巴特沃斯滤波器的零极点和增益因子, 其3 dB的边界频率为1。z是一个空向量, p为N个元素的向量, k为增益因子。系统函数为

$$H_a(s) = \frac{k}{(s - p_1)(s - p_2) \cdots (s - p_N)} \quad (1)$$

③ butter: 在给定阶数和边界频率时, 设计滤波器, 格式为

[B, A] = butter[N, Wn, 'type', 's']: 给定模拟滤波器的阶数N和边界角频率Wn, 设计N阶模拟巴特沃斯低通滤波器。计算得到的B和A分别为滤波器的系统函数的分子和分母多项式的系数。当设计低通滤波器时, 'type'可缺省。

第一题

- 1、利用 Matlab 的 Butterworth 方法设计一个低通滤波器，通带截止频率为 100Hz，通带衰减为 1dB，阻带截止频率为 200Hz，阻带衰减为 15dB，采样频率为 500HZ。

程序

```
wp=100*2*pi; %通带截止频率
ws=200*2*pi; %阻带截止频率
ap=1;        %通带最大衰减
as=15;        %阻带最小衰减
Fs=500;       %采样频率
%选择滤波器的最小阶数
[N,Wc]=buttord(wp,ws,ap,as,'s'); %N为符合要求的滤波器的最小阶数，Wc为巴特沃斯模拟低通滤波器3db截止频率，'s'表示模拟域，即表示wp,ws,wc都使用角频率
[Z,P,K]=buttap(N); %返回一个n阶，巴特沃斯型归一化的模拟低通滤波器的零极点增益模型
[A,B,C,D]=zp2ss(Z,P,K); %输入参数：Z,P,K分别表示系统的零极点增益模型的零点、极点和增益
功能：将系统的零极点增益模型转换为状态空间模型
```

程序

%低通到低通的转变

[AT,BT,CT,DT]=lp2lp(A,B,C,D,Wc); %将用状态方程表示截止频率为1rad/s的模拟低通滤波器变换为截止频率为Wc的模拟低通滤波器

[num1,den1]=ss2tf(AT,BT,CT,DT); %返回系统传递函数模型， num1为系统传递函数分子系数组成向量， den1为分母系数向量

[num2,den2]=bilinear(num1,den1,500); %双线性变换法将模拟滤波器转换为数字滤波器

%绘制幅频响应曲线

[H,W]=freqz(num2,den2); %[H, W] = freqz (b, a, n) 返回n点复频响应矢量H和n点的频率向量w。 b和a为系统传递函数的分子和分母的系数向量。如果n没有指定，默认为512。

plot(W*Fs/(2*pi),abs(H));grid;

xlabel('频率/Hz');

ylabel('幅值');

第二题

2、利用 Matlab 设计一个 3dB 带宽为 10kHz，过渡带小于等于 1kHz 的 Butterworth 高通滤波器。

程序

```
FS=20;
F1=10;
Fh=11;
wp1=(F1*2*pi)/FS;      %频率做归一化处理
ws1=(Fh*2*pi)/FS;%
[n,Wn]=buttord(wp1,ws1,1,25,'s');
[b,a]=butter(n,Wn,'s');  %n为滤波器的阶数，Wn代表滤波器的截止频率，这两个参数都可由buttord函数来确定
[H,W]=freqz(b,a,256);   %返回256点复频响应矢量H和256点的频率向量w
plot(W*FS/(2*pi),abs(H));
grid;
xlabel('频率/kHz');ylabel('幅度');
```

Thank you

