控 制 原 理 实 验

实验一 Matlab介绍

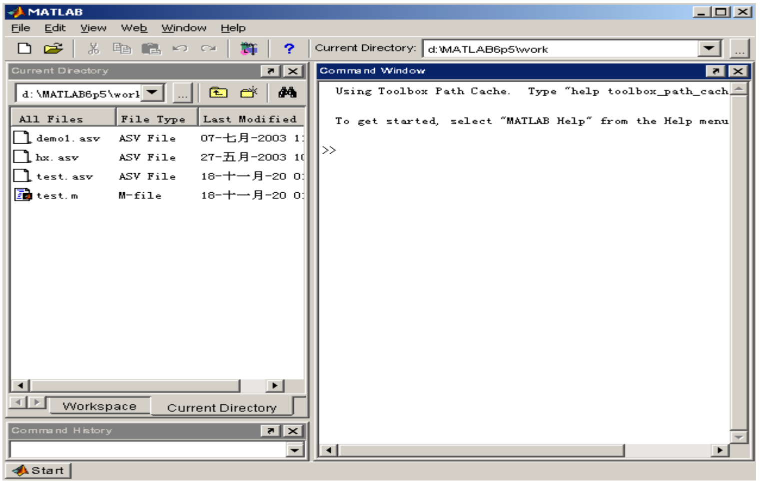
1.1 Matlab简介

Matlab是美国**Mathworks**公司推出的一套高性能的数值分析和计算软件，它将矩阵运算、数值分析、图形处理、编程技术结合在一起，为用户提供了一个强有力的科学及工程问题分析计算和程序设计的工具。

Matlab语言可以被认为是一种解释性语言，用户可以在**Matlab**的工作空间中输入一个指令，也可以在编辑器中编写应用程序，应用程序执行时，**Matlab**软件对其中的命令和函数进行翻译，然后在**Matlab**环境中对它进行处理，最后返回结果。

实验一 Matlab介绍

1.2 Matlab的桌面环境



实验一 Matlab介绍

1.3 Matlab基本指令和用法

1) 数值、变量和表达式

- 数值一般采用十进制数表示：**45 -103 3.56 2.348e8 2.2e-1**
- 变量名、函数名是对字母大小写敏感的，变量名的第一个字符必须是英文字母，最多可以包含**31**个字符，变量名中不能包含标点、空格字符 .
- **Matlab**中存在一些固定的变量，如**eps**（相对精度或机器零阈值）、**pi**（π）**Inf**（+∞）、**NaN**、**realmax**、**realmin**等
- 表达式遵循日常中的习惯写法，支持复数的使用，虚数符号**i**或**j**。如 **z=3+4i**

实验一 Matlab介绍

1.3 Matlab基本指令和用法

2）向量运算

构造向量：用“：”产生不同的向量

>> x=1:4　　结果：　x =　1　2　3　4

>> y=0:pi/4:pi　结果：　y =　0　0.7854　1.5708　2.3562　3.1416

构造矩阵：A=[1 2 3;4 5 6;7 8 9]

矩阵的分行输入

```
A=[1,2,3
   4,5,6
   7,8,9]
```

实验一 Matlab介绍

1.3 Matlab基本指令和用法

3）向量的下标

作用：对矩阵的行列子矩阵处理时使用；也可以用来产生向量。

如：A(3,1)表示A矩阵第三行第一列的元素。

下标可以是向量：

A(1:5,3) 指A中前5行对应第三列元素组成的5×1子矩阵。

使用“：”代替下标：可以表示所有的行或列。

子矩阵的赋值语句：

>> A(:,[2 3])=B(:,1:2)

实验一 Matlab介绍

1.3 Matlab基本指令和用法

```
>> A=[1:5;6:10;11:15;16:20;21:25]
A =
    1    2    3    4    5
    6    7    8    9   10
   11   12   13   14   15
   16   17   18   19   20
   21   22   23   24   25
>> A(3,1)
ans =
   11
```

```
>> A(1:5,3)

ans =

     3
     8
    13
    18
    23
```

实验一 Matlab介绍

1.3 Matlab基本指令和用法

4）矩阵的简单运算

空矩阵：**x=[]**产生一个**0**×**0**的矩阵

矩阵的转置：**>> B=A'**

矩阵的乘法：用**"*"**表示，矩阵与标量相乘表示矩阵中的每个元素都和标量相乘 。**>> x'*y**，**>> A*2**

矩阵的逆：**inv(A)**

矩阵的点乘（除）：对应元素相乘（除）

矩阵的除法：两种不同的矩阵除法符号**"/"**和**"\"**分别表示右除和左除，**A\B=inv(A)*B**，**B/A=B*inv(A)**

实验一 Matlab介绍

1.3 Matlab基本指令和用法

```
>> A=[1:5;6:10;11:15;16:20;21:25]
A =
    1    2    3    4    5
    6    7    8    9   10
   11   12   13   14   15
   16   17   18   19   20
   21   22   23   24   25
>> A'
ans =
    1    6   11   16   21
    2    7   12   17   22
    3    8   13   18   23
    4    9   14   19   24
    5   10   15   20   25
```

```
>> A*2
ans =
    2    4    6    8   10
   12   14   16   18   20
   22   24   26   28   30
   32   34   36   38   40
   42   44   46   48   50
>> B=[1 2 3 4]
B =
    1    2    3    4
>> B.*[5 6 7 8]
ans =
    5   12   21   32
```

实验一 Matlab介绍

1.3 Matlab基本指令和用法

矩阵的乘方：**A^P**表示**A**的**P**次方。如果**P**不是整数，则计算涉及特征值特征向量问题

创建矩阵的函数：创建某些特殊的矩阵

eye(size(A))：产生与**A**矩阵同阶的单位矩阵

zeros（）和**ones**（）：产生**0**和**1**的矩阵

rand（）：产生随机元素的矩阵

diag()、**triu()**、**tril()**：创建对角、上三角、下三角矩阵

size()：显示一个包含两个元素的向量：矩阵的行与列的个数。函数**length()**返回向量的长度或矩阵行数和列数的最大值

实验一 Matlab介绍
1.3 Matlab基本指令和用法

方阵的行列式： ***det***

矩阵的迹： ***trace***

矩阵的秩： ***rank***

矩阵和向量的范数

norm 欧几里德范数

***norm*(*x*,inf)** 无穷范数

矩阵函数

expm ***logm*** ***sqrtm***

实验一 Matlab介绍
1.3 Matlab基本指令和用法

5）矩阵的特殊运算

矩阵的三角分解：将一个方阵表示为一个上三角阵（**U**）和一个下三角阵（**L**）的乘积（**LU**分解） **[L,U]=lu(X)**

矩阵的正交变换：分解为正交矩阵（**Q**）和上三角矩阵（**R**）的乘积（**QR**分解） **[Q,R]=qr(A)**

奇异值分解：**[U,S,V]=svd(A) A=U*S*V’**

矩阵的特征值：**eig(A)**以列向量形式返回特征值，**[X,D]=eig(A)**返回特征值和特征向量 ，**D**为特征值对角阵，特征向量**X**。

实验一 Matlab介绍

1.3 Matlab基本指令和用法

6）数据的图形显示

二维图形的绘制：

>> **plot(x,y,‘-’x,z)** %绘制二维曲线

>> **grid** %图形加网格线

>> **xlabel('X')** %输出X轴说明

>> **ylabel('Y')** %输出Y轴说明

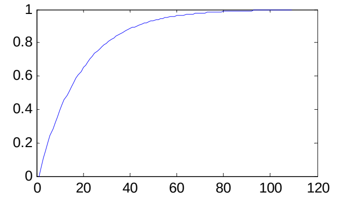
>> **title('sine and cosine curve')** %在图形上方加上图形说明

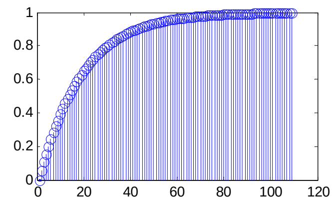
>> **text(2.5,0.7,'sin(x)')** %在适当位置为图形加上注释

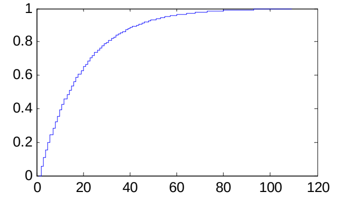
实验一 Matlab介绍

1.3 Matlab基本指令和用法

```
>> sys=tf([1],[10 1]);
>> y=step(sys);
>> plot(y)
>> stem(y)
>> stairs(y)
```







实验一 Matlab介绍

1.3 Matlab基本指令和用法

特殊的二维图形

- **polar** 画极坐标形式函数**r = f (θ)**的极坐标图

用法如下：

polar(theta,rho,LineSpec)

- **pie** 用**x**中的数据画一饼形图

实验一 Matlab介绍

1.3 Matlab基本指令和用法

7）命令窗口基本指令

cd：设置当前工作目录 **clf**：清除图形窗口

clc：清除命令窗口中的显示内容

clear：清除**matlab**工作空间中所有或指定的变量

dir：列出指定目录下的文件和子目录清单

edit：打开**M**文件编辑器

exit(quit)：关闭（退出）**matlab**

more：在命令窗口中分页显示输出的内容

pack：整理**matlab**的内存碎块 **type**：显示指定文件的内容

实验一 Matlab介绍

1.4 Matlab控制流结构

1)循环语句

```
for x=1:4          while expression
   …                  …
end                end
```

2) 选择语句

```
if expression
   …
else
   …
end
```

实验一 Matlab介绍

1.4 Matlab控制流结构

3) 条件判断语句

```
switch x
    case result1
    …
    case result2
    …
    …
    case resultn
    …
    otherwise
    …
end
```

实验一 Matlab介绍

1.4 Matlab控制流结构

4) 其他常用控制指令

input：将**Matlab**的控制权暂交给用户。用户通过键盘输入相应的数值或者表达式。

keyboard：将控制权交给键盘。输入各种合法的**Matlab**指令

pause/pause(n)：暂时停止文件的执行。用户按任意键，文件继续执行

break,continue：**break**指令将使包含有该指令的**for,while**循环结构的循环过程立即终止，**continue**指令将使包含有该指令的**for, while**循环结构的本次循环过程结束，立即进行下一次的循环过程。

实验一 Matlab介绍

1.5 关系运算和逻辑运算

运算符：<（小于）、<=（小于等于）、>（大于）>=（大于等于）、==（等于）、~=（不等于）

逻辑运算：**&**（与）、|（或）、~（非）（非零元素都表示为真）

关系函数和逻辑函数：

all(x)：检查x是否全为1　　**isempty(x)**：检查x是否为空阵

any(x)：检查x是否有不为零的元素

find(x)：找出非零元素的位置标识

isinf(x)：检查x是否为无穷大　　**isstr(x)**：检查x是否为字符串

实验一 Matlab介绍

1.6 工作空间（workspace）

1）基本指令who和whos

列出**matlab**工作空间中驻留的变量清单，二者不同的是**whos**在给出驻留的变量名的同时，还给出各个变量的维数和属性（类型等），如 **Name Size Bytes Class**

R 31x31 7688 double array

2）工作空间的保存

保存/载入部分变量：**load**和**save** 。保存所有变量：**“File: Save workspace As”**。装载所有变量：**“File: Load workspace”** 。如：

save FileName v1 v2

实验一 Matlab介绍

1.7 路径的设置

方法一：将用户目录设置为当前目录

cd d:\mydir 或者 **pwd d:\mydir**

方法二：将用户目录设置为**Matlab**的搜索路径中

path(path,’d:\mydir’)

用这种方法设置的目录信息只在当前环境中有效，一旦**Matlab**重新启动，以前的设置就无效。

方法三：可以用路径浏览器进行修改

在命令窗口中使用**pathtool**，或者在桌面环境中选择**“File:Set Path”**菜单项。

保存搜索路径的修改，则下次启动**Matlab**，不需要重新进行修改 或者

实验一 Matlab介绍

1.8 M文件的编写与调试

1）M文件的编写

启动编译器的方法

- ✓ 在命令窗口输入指令**edit**，**edit filename** （打开已有的**M**文件）
- ✓ 单击桌面工具条上的新建/打开文件图标

在桌面菜单中选择**“File: New”**或者**“File: Open”**， 再选择**“M-File”**

- ➢ 编写**M**文件

可以自定义函数：**function y=average(x)**

可接受输入参数并返回输出参数，其内的变量不占用**Matlab**工作空间，第一行包含**function**

注： **M**文件的调用以文件名为准

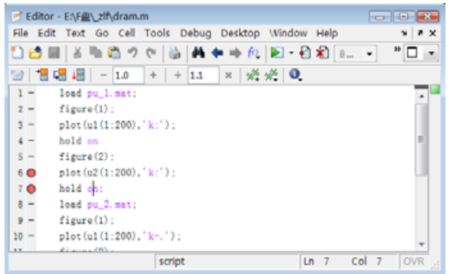
%为**Matlab**的注释符，其后的语句不执行（只对当前行有效）。

实验一 Matlab介绍

1.8 M文件的编写与调试

2） M文件的调试

- ➢ 断点设置：红色断点标志
- ➢ 调试：单步执行程序，也可直接跳到下一断点处
- ➢ 中间变量的观察：**“Tool: Option: General: Show Datatype”**被选中的情况下



实验一 Matlab介绍

1.9 变量的作用范围

1）局部变量

存在于函数内部的中间变量，其作用范围仅仅限于函数本身。缺省状态都是局部变量 。

2）全局变量

通过**global**指令，**Matlab**允许不同的函数空间以及基本工作空间共享一个变量。

对全局变量的声明必须在该变量使用之前进行声明，一般放在函数文件或**M**文件的开头

实验一 Matlab介绍

1.10 帮助

方法一：在命令窗口中直接运行**help**指令，获取有关指令的使用方法

方法二：运行帮助浏览器

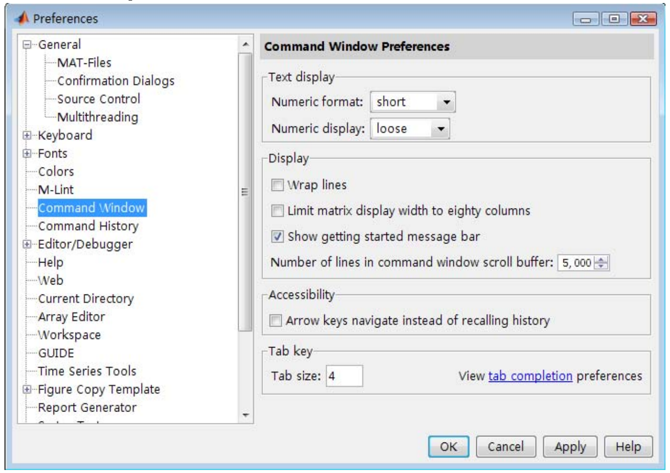
方法三：查看**PDF**帮助文档

Matlab的**toolbox**目录下任何子目录中的**content.m**文件描述了该目录下所有函数的主要功能。

实验一 Matlab介绍

1.11 精度设置

菜单：**File\preferences**



实验二 Matlab数值计算功能

2.1 多项式

1）多项式表达与创建

Matlab用行矢量表示多项式系数，其中各元素按照降幂顺序排列

POLY 将根系数转换为多项式系数

例：将多项式**(x-6)(x-3)(x-8)**表示为系数形式

```
>> a=[6 3 8];          %写成根矢量
>> pa=poly(a)          %求出系数矢量
 pa =
     1  -17   90  -144
```

实验二 Matlab数值计算功能

2.1 多项式

2）多项式的乘/除

c=conv(a,b) 多项式乘法

[q,r]=deconv(c,a) 多项式除法（**q**为商，**r**为余数）

3）其他常用的多项式运算命令

pa=polyval(p,s)：按数组运算规则计算给定**s**时多项式**p**（系数矢量）的值

[r,p,k]=residue(b,a)：部分分式展开，**b**、**a**分别为分子分母多项式系数矢量，**r**、**p**、**k**分别是留数、极点和直项矢量

p=polyfit(x,y,n)：用**n**阶多项式拟合**x**、**y**矢量给定的数据

polyder(p)：多项式微分

实验二 Matlab数值计算功能

2.2 特征值和特征向量

两种调用方式：

- **e=eig(a)**：其中**e**是一个包含特征值的矢量
- **[v,d]=eig(a)**：其中**v**是一个与**a**相同的**n**×**n**阶矩阵，它的每一列是矩阵**a**的一个特征值所对应的特征矢量，**d**为对角阵，其对角元素即为矩阵**a**的特征值。

```
>> a=[1 2;0 4];
>> eig(a)
ans =
     1
     4
```

实验三　拉氏变换与Z变换

3.1　函数表示方法

syms 定义符号变量名：

```
>> syms a b c
>> f=a+b+c
f =
a+b+c
```

实验三　拉氏变换与Z变换

3.2 Laplace拉氏变换

L = laplace(F) 系统F（缺省独立变量为t）的拉氏变换，缺省返回为s的函数，如果 **F = F(s)**，那么LAPLACE返回为t的函数**L = L(t).**

L = laplace(F,t) 返回L为t的函数，而不是缺省时s的函数

L = laplace(F,w,z) 返回L为z的函数，而不是缺省时s的函数积分变量为w

```
syms a s t w x
laplace(t^5)          returns   120/s^6
laplace(exp(a*s))     returns   1/(t-a)
laplace(sin(w*x),t)   returns   w/(t^2+w^2)
laplace(cos(x*w),w,t) returns   t/(t^2+x^2)
```

实验三　拉氏变换与Z变换

3.2 Laplace拉氏变换

求拉氏变换　$e^{-bt}\cos(at+c)$

Matlab程序:

```
>>syms a t b c
>> laplace(exp(-b*t)*cos(a*t+c))
  ans =
      ((s+b)*cos(c)-a*sin(c))/((s+b)^2+a^2)
```

$$\frac{(s+b)\cos(c)-a\sin(c)}{(s+b)^2+a^2}$$

实验三　拉氏变换与Z变换

3.3 Laplace反变换

F = ilaplace(L) 系统L（缺省独立变量为s）的拉氏逆变换，缺省返回为t的函数，如果 **L = L(t)**，那么**LAPLACE**返回为s的函数**L = L(s).**

F = ilaplace(L,y) 返回F为y的函数，而不是缺省时的t的函数

F = ilaplace(L,y,x) 返回F为x的函数，而不是缺省时的t的函数。积分变量为y

```
syms s t w x y
ilaplace(1/(s-1))            returns   exp(t)
ilaplace(1/(t^2+1))          returns   sin(x)
ilaplace(y/(y^2 + w^2),y,x)  returns   cos(w*x)
ilaplace(s/s)                returns   Dirac(t)
```

实验三　拉氏变换与Z变换

3.3 Laplace反变换

求拉氏反变换

$$\frac{s+d}{(s+a)(s+b)(s+c)}$$

```
>> syms s a b c d
>> ilaplace((s+d)/((s+a)*(s+b)*(s+c)))
ans =
   1/(a-b)/(a-c)*exp(-a*t)*d-1/(a-b)/(a-c)*exp(-a*t)*a-1/(b-c)/(a-b)*exp(-b*t)*d
   +1/(b-c)/(a-b)*exp(-b*t)*b-1/(b-c)/(a-c)*exp(-c*t)*c+1/(b-c)/(a-c)*exp(-c*t)*d
```

$$\frac{e^{-at}d}{(a-b)(a-c)}-\frac{e^{-at}a}{(a-b)(a-c)}-\frac{e^{-bt}d}{(b-c)(a-b)}+\frac{e^{-bt}b}{(b-c)(a-b)}-\frac{e^{-ct}c}{(b-c)(a-c)}+\frac{e^{-ct}d}{(b-c)(a-c)}$$

实验三　拉氏变换与Z变换

3.4 Z变换

F = ztrans(f) 表示函数**f(n)**的**Z**变换,返回为**z**的函数**F(z)**，**f = f(n) => F = F(z).**

F = ztrans(f,w) 返回**F**是**w**的函数

F = ztrans(f,k,w) 表示**f**是变量**k**的函数，返回**F(w)**

应用到连续系统时，必须首先将连续系统离散化，例：**sin(wt)**写为**sin(wTn)**，然后用**ztrans**进行**Z**变换。

```
>> ztrans((t*T)^2/2)
ans =
1/2*T^2*z*(z+1)/(z-1)^3
```

实验三　拉氏变换与Z变换

3.4　Z反变换

f = iztrans(F) ——表示F(z)的反变换，缺省时返回f为n的函数f(n)，F = F(z) => f = f(n).如果F = F(n), 那么返回f为k的函数: f = f(k).

f = iztrans(F,k) 表示返回f为k的函数（而不是缺省时的n的函数）

f = iztrans(F,w,k) ——表示F为w的函数，返回f为k的函数

```
>>syms z x k
>> iztrans(z/(z-2))
ans =
2^n
>> iztrans(exp(x/z),z,k)
ans =
x^k/k!
```

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

连续系统：

$$H(s)=\frac{num(s)}{den(s)}=\frac{b_1s^m+b_2s^{m-1}+\cdots+b_{m+1}}{a_1s^n+a_2s^{n-1}+\cdots+a_{n+1}}$$ 分子分母 num/den

$$H(s)=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$ 零极点增益 [z,p,k]

$$\begin{cases}\dot{x}=ax+bu\\ y=cx+du\end{cases}$$ 状态方程（a,b,c,d）

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

离散系统：

$$H(z)=\frac{num(z)}{den(z)}=\frac{b_1 z^m+b_2 z^{m-1}+\cdots+b_{m+1}}{a_1 z^n+a_2 z^{n-1}+\cdots+a_{n+1}}$$ 分子分母 num/den

$$H(z)=k\frac{(z-z_1)(z-z_2)\cdots(z-z_m)}{(z-p_1)(z-p_2)\cdots(z-p_n)}$$ 零极点增益 [z,p,k]

$$\begin{cases}x(k+1)=ax(k)+bu(k)\\ y(k)=cx(k)+du(k)\end{cases}$$ 状态方程（a,b,c,d）

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

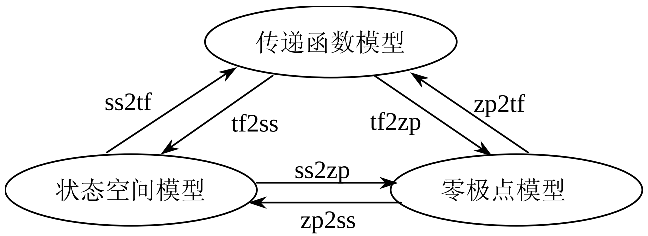
模型建立和模型转换函数：

函数名称	功能
filt	生成 DSP 形式的离散传递函数（z^{-1} 形式）
ss	生成状态空间模型或者转换成状态空间模型
tf	生成传递函数模型或者转换成传递函数模型
zpk	生成零极点模型或者转换成零极点模型
tfdata	获取传递函数模型数据
zpkdata	获取零极点模型数据
ssdata	获取状态空间模型数据

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

也可以用函数tf2ss, ss2tf, zp2tf, tf2zp, ss2zp, zp2ss



实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

SYS = tf(NUM,DEN)——生成连续时间函数，分子**num**，分母**den**

SYS = tf(NUM,DEN,TS)——生成离散传递函数，**Ts**为采样周期，**Ts**＝－**1**表示采样时间不定

SYS = tf(SYS)——将任意形式的模型转换为传递函数

SYS = tf(NUM,DEN,'inputdelay',T)——生成有纯滞后的传递函数

```
>> sys1=tf([1 2],[1 2 1])
Transfer function:
   s + 2
-------------
s^2 + 2 s + 1
>> sys1=tf({1,2;3,4},{[1 1],[2 1];[3 1],[4 1]});  %生成传递函数阵
```

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

s = tf('s') ——表示传递函数是s域的

```
>> s=tf('s');
>> H = (s+1)/(s^2+3*s+1)
Transfer function:
   s + 1
-------------
s^2 + 3 s + 1
```

这种方法当分子分母多项式的系数不容易获得时（如分子分母是乘积形式的）是非常有效的

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

SYS = ss(A,B,C,D)——生成连续状态方程，A、B、C、D表示状态方程的相应矩阵。

SYS = ss(A,B,C,D,Ts)——生成离散状态方程。Ts为采样时间，Ts＝－1表示采样时间不定

SYS = ss(SYS)——将任意形式的模型转换为状态方程

SYS = zpk(Z,P,K)——生成连续零极点增益模型，零点Z，极点P，增益K。

SYS = zpk(Z,P,K,Ts)——生成离散零极点增益模型，Ts为采样周期，Ts＝－1表示采样时间不定

SYS = zpk(SYS)——将任意形式的模型转换为零极点增益模型

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

提取传递函数的信息：

[num,den] =tfdata(SYS,‘v’)——对于**SISO**连续/离散系统，下同

[num,den,Ts,Td] =tfdata(SYS,‘v’)——Ts和**Td**为采样周期和输入时延

[z,p,k]=zpkdata(sys,’v’)

[z,p,k,Ts,Td]=zpkdata(sys,’v’)

[a,b,c,d]=ssdata(sys)

[a,b,c,d,Ts,Td]=ssdata(sys)

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

1）传递函数到状态方程

[A,B,C,D] = ss2tf(NUM,DEN), 传递函数转换为状态方程

对于离散系统，必须保证分子和分母多项式的长度相同

$$G(s)=\frac{s+7}{s^2+4s+3}$$

```
num=[0 1 7];
den=[1 4 3];
[a,b,c,d]=tf2ss(num,den)
也可以：
sys=tf(num,den);
sys1=ss(sys)
```

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

2）状态方程转换到传递函数

[NUM,DEN] = ss2tf(A,B,C,D,iu)，　状态方程转换为传递函数

$$x(k+1)=\begin{bmatrix}0.6 & 0.233\\ -0.466 & -0.097\end{bmatrix}x(k)+\begin{bmatrix}0.2\\ 0.233\end{bmatrix}u(k)$$
$$y(k)=\begin{bmatrix}1 & 0\end{bmatrix}x(k)$$

```
a=[0.6 0.233;-0.466 -0.097];
b=[0.2;0.233];
c=[1 0];
d=0;
[num,den]=ss2tf(a,b,c,d,1);
sys=tf(num,den,-1)
或者：
sys=ss(a,b,c,d,-1);
sys1=tf(sys)
```

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

3）传递函数的部分分式展开

$$\frac{B(s)}{A(s)}=\frac{\text{num}}{\text{den}}=\frac{b(1)s^n+b(2)s^{n-1}+\cdots+b(n)}{a(1)s^n+a(2)s^{n-1}+\cdots+a(n)}$$

行向量**num**和**den**表示传递函数的分子和分母的系数

num=[*b*(1)　*b*(2)　... *b*(n)]　　　**den=[*a*(1)　*a*(2)　... *a*(n)]**

[r, p, k]=residue(num, den)---- 将求出两个多项式***B(s)***和***A(s)***之比的部分分式展开留数、极点和直接项。

$$\frac{B(s)}{A(s)}=\frac{r(1)}{s-p(1)}+\frac{r(2)}{s-p(2)}+\cdots+=\frac{r(n)}{s-p(n)}+k(s)$$

实验四　控制系统模型

4.1 传递函数和状态方程的生成（离散和连续）

例: 试求以下传递函数的部分分式表达式

$$\frac{B(s)}{A(s)}=\frac{2s^3+5s^2+3s+6}{s^3+6s^2+11s+6}$$

```
num=[2 5 3 6]
den=[1  6 11 6]
[r, p, k]=residue(num, den)
```

运行结果为:

```
r =
   -6.0000
   -4.0000
    3.0000
p =
   -3.0000
   -2.0000
   -1.0000
k =2
```

$$\frac{B(s)}{A(s)}=\frac{2s^3+5s^2+3s+6}{(s+1)(s+2)(s+3)}=\frac{3}{s+1}+\frac{-4}{s+2}+\frac{-6}{s+3}+2$$

实验四　控制系统模型

4.2 连续模型与离散模型的转换

函数名称	功能
c2d	连续时间系统离散化
d2c	离散时间系统连续化

1）连续系统的离散化

SYSD = c2d(SYSC,Ts,METHOD)——将连续模型转换为离散模型，**METHOD**缺省为采用零阶保持器的方法，**Ts**为采样周期。

Method: 'zoh'——采用零阶保持器　**'foh'**——采用一阶保持器

'tustin'——采用双线形（**tustin**）逼近方法

'prewarp'——采用改进的**tustin**方法

'matched'——采用**SISO**系统的零极点匹配法。

例：

>>sysc=ss([0 1;0 –2],[0;1],[1 0],0) %生成连续状态空间模型

>>sysd=c2d(sysc,1); %连续状态方程离散化，采样周期为1秒

实验四　控制系统模型

4.2　连续模型与离散模型的转换

2) 离散时间系统连续化

sysc=d2c(sysd,method)——离散对象的连续化

Method:　**'zoh'**——采用零阶保持器

'tustin'——采用双线形（**tustin**）逼近方法

'prewarp'——采用改进的**tustin**方法

'matched'——采用**SISO**系统的零极点匹配法。具有接近**1**的极点的情况。

zoh法不适合系统具有**z=0**的极点的情况，对于具有负实数极点的系统，该方法将增加系统的阶。

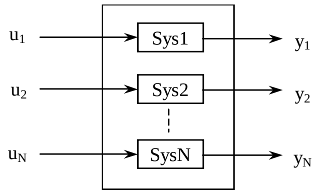
Tustin法不适合系统具有**z=1**

实验四　控制系统模型

4.3　典型连接

常见的典型连接有串连、并联、反馈等

1) 多个系统的组合



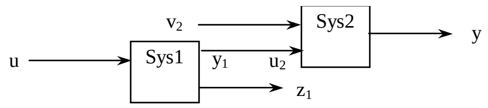
SYS = APPEND(SYS1,SYS2, ...)

——对多个**LTI**系统的组合（将输入和输出组合起来）。

实验四　控制系统模型

4.3　典型连接

2) 系统的串联实现

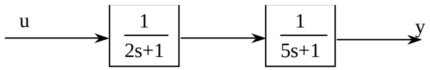


sys=series(sys1,sys2)——返回两个系统**sys1**和**sys2**的串联系统。两个子系统必须连续时间系统或者具有相同采样周期的离散时间系统。

sys=series(sys1,sys2,outputs1,inputs2)——**outputs1**和**inputs2**用于指定**sys1**的部分输出与**sys2**的部分输入进行连接。

实验四　控制系统模型

4.3　典型连接



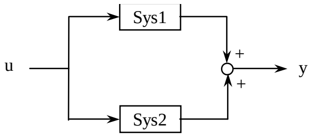

```
>> s=tf('s');
>> sys1=1/(2*s+1);
>> sys2=1/(5*s+1);
>> sys=series(sys1,sys2)
 Transfer function:
            1
     ---------------
    10 s^2 + 7 s + 1
```

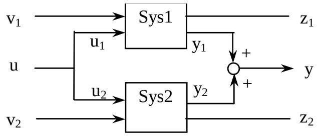
$$sys = \frac{1}{10s^2 + 7s + 1}$$

实验四　控制系统模型

4.3　典型连接

3）系统的并联



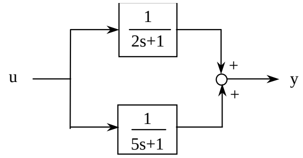


sys=parallel(sys1,sys2)——返回两个系统并联连接系统，两个子系统必须连续时间系统或者具有相同采样周期的离散时间系统。

sys=parallel(sys1,sys2,inp1,inp2,out1,out2)——**inp1**和**inp2**分别表示两个系统连接在一起的输入端，**out1**和**out2**中分别指定要做相加的输出端编号。

实验四　控制系统模型

4.3　典型连接



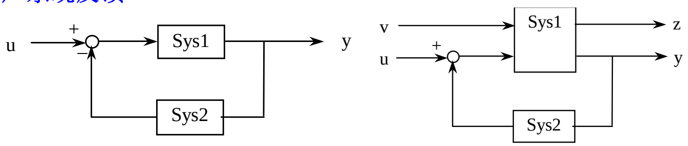

```
>> parallel(sys1,sys2)
 Transfer function:
   7 s + 2
----------------
10 s^2 + 7 s + 1
```

$$sys=\frac{7s+1}{10s^2+7s+1}$$

实验四　控制系统模型

4.3　典型连接

4）系统反馈



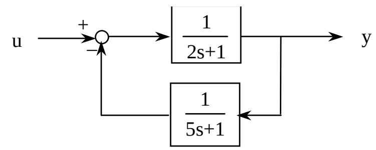
sys=feedback(sys1,sys2)——返回**sys1**和**sys2**的反馈连接系统**sys**，反馈为负反馈。两个子系统必须连续时间系统或者具有相同采样周期的离散时间系统。

sys=feedback(sys1,sys2,sign)——定义反馈形式**sign**，**sign=+1**表示正反馈，**sign=-1**表示负反馈。

sys=feedback(sys1,sys2,feedin,feedout,sign)——将**sys1**的指定输出**feedout**连接到**sys2**的输入，**sys2**的输出连接到**sys1**的指定输入**feedin**，以此构成闭环系统

实验四　控制系统模型

4.3　典型连接



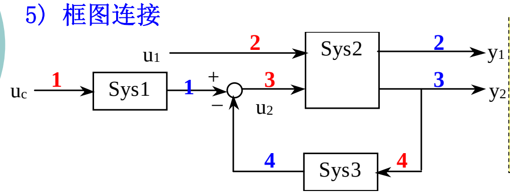

```
>> feedback(sys1,sys2)
   Transfer function:
         5 s + 1
     ----------------
      10 s^2 + 7 s + 2
```

$$sys = \frac{5s+1}{10s^2+7s+2}$$

实验四　控制系统模型

4.3　典型连接

5）框图连接

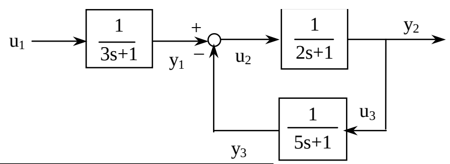



```
sys=append(sys1,sys2,sys3);
Q=[3 1 -4;4 3 0];
inputs=[2 1];
outputs=[2 3];
```

sysc=connect(sys,Q,inputs,outputs)——框图建模，**sys**为由**append**生成的无连接对角方块系统，**Q矩阵用于指定系统sys的内部连接关系，其中矩阵的每一行对应一个输入，其第一个元素为输入编号，其后为连接该输入的输出编号，如采用负连接，则以负值表示。****inputs**和**outputs**用于指定无连接系统中的某些输入/输出保留作为外部的输入输出

实验四　控制系统模型

4.3　典型连接




```
>> sys1=1/(3*s+1);
>> sys2=1/(2*s+1);
>> sys3=1/(5*s+1);
>> sys=append(sys1,sys2,sys3);
>> Q=[2 1 -3;3 2 0];
>> inputs=1;
>> outputs=2;
>> sysc=connect(sys,Q,inputs,outputs)
```

```
Transfer function:
        0.1667 s + 0.03333
   -----------------------------------------
s^3 + 1.033 s^2 + 0.4333 s + 0.06667
```

$$sys=\frac{5s+1}{30s^3+31s^2+13s+2}$$

实验五 控制系统分析

5.1 特征方程特征根

连续系统的特征根在左半平面，离散系统特征根在z的单位圆内，则系统稳定。连续系统的零点在左半平面，离散系统零点在z的单位圆内，则系统是最小相位的

pole(sys), eig(A)——计算系统的极点，所有极点都位于系统左半平面，则系统是稳定的；否则系统不稳定，返回极点。

[V,D]=eig(A)——矩阵的特征值和特征向量。

poly(A)——求矩阵的特征多项式。当**A**为向量时，**A**中的元素为所求得的多项式的根。

polyeig(A)——求矩阵的特征值。

roots(B)——求多项式的根。

Z = zero(SYS) ——求传递函数的零点。

实验五 控制系统分析

5.2 控制系统时域分析

控制系统最常用的时域分析方法是：当输入信号为单位阶跃和单位脉冲信号时的系统输出响应。

函数名称	功能
step	单位阶跃响应
impulse	单位脉冲响应
intial	零输入响应
lsim	任意输入响应

离散系统响应可在上述函数名称前加**d**，例：**dstep**，也可以采用上述函数，对应的系统采用离散形式。

实验五　控制系统分析

5.2　控制系统时域分析

step(SYS)——线性时不变模型sys的单位阶跃响应。

step(SYS,TFINAL) ——连续系统响应时间t=0到t=TFINAL。离散系统TFINAL表示采样步数

step(SYS,T) ——用户定义仿真时间向量。对离散模型，T为Ti:Ts:Tf ，Ts为采样时间，对于连续模型，T为Ti:dt:Tf，dt为离散化时的采样时间，阶跃输入t=0开始。

initial(SYS,X0)——X0为初始状态的状态方程零输入响应。

initial(SYS,X0,TFINAL)， initial(SYS,X0,T)中参数同上。

lsim(SYS,U,T)——任意输入的响应，输入由U和T表示，T，U的列表示输入个数，U的行u(i)表示T(i)时刻对应的行：t = 0:0.01:5; u = sin(t); lsim(sys,u,t)，对于离散系统，U按照sys的采样周期采样，如果sys的采样周期不定，则T可省略，或者[]表示。

lsim(SYS,U,T,X0) 非零初始状态X0。

实验五　控制系统分析

5.2　控制系统时域分析

例：求三阶系统的单位阶跃响应和单位脉冲响应及零输入响应和零极点图，设初始状态为[1 1 -1]'.

$$H(s)=\frac{5(s^2+5s+6)}{s^3+6s^2+10s+8}$$

```
num=[5 25 30];
den=[1 6 10 8];
x0=[1 1 -1]';
sys=tf(num,den);
figure(1);
subplot(2,2,1);
step(sys)
```

```
subplot(2,2,2)
impulse(sys)
subplot(2,2,3)
initial(ss(sys),x0)
subplot(2,2,4)
pzmap(sys)
```

实验五　控制系统分析

5.2　控制系统时域分析