

## 第十二章 控制系统仿真

控制系统仿真一般采用如下两种形式进行仿真分析：

1. 针对单输入—单输出系统的信号流图形式
2. 面向多输入—多输出系统的状态空间形式

控制系统仿真的主要研究内容是通过系统的数学模型和计算方法，编写程序运算语句，使之能自动求解各环节变量的动态变化情况，从而得到关于系统输出和所需要的中间各变量的有关数据、曲线等，以实现对控制系统性能指标的分析与设计。

一般地来说，实现控制系统的仿真有以下几个步骤：

1. 根据建立的数学模型以及计算机精度和时间等要求，确定采用的数值计算方法；
2. 将数学模型按算法要求通过分解、综合、等效变换等方法转换成适于在计算机上运行的公式、方程等；
3. 用合适的开发语言进行算法编程和实现；
4. 通过上机运行调试，不断加以改进，使之正确反映系统各项动态性能指标，并得到理想的仿真结果。

围绕以上步骤，控制系统仿真近年来不断发展，不断更新，基于 MATLAB 语言开发的专门应用于控制系统分析与设计的工具箱，对控制系统仿真技术的发展及应用起到巨大的推动作用。

本章主要围绕着控制系统仿真实实现的问题，研究仿真的几种常用方法，主要包括：基于状态空间法的系统仿真、非线性系统的仿真以及离散系统的仿真，并重点阐述了 Simulink 动态仿真软件的应用。希望通过本章的学习使大家能够较系统地了解目前控制系统仿真领域的研究方法和实现手段，并从中掌握基本的系统仿真实实现的技巧和能力。

### 12.1 状态空间法仿真

控制系统的动态模型经常是转化成以状态方程的形式给出的，一般采用四阶龙格-库塔数值积分方程算法进行求解与分析仿真，这就是状态空间法仿真的基本方法。

#### 12.1.1 四阶龙格-库塔法

闭环控制系统最常见的两种描述方式为：传递函数法和状态空间法，而且这两种方法之间可以相互转换。如果系统是由传递函数来描述的，则应用第八章的模型转换方法，可以方便地将传递函数表达式转换成状态空间表达式。

已知系统的状态方程为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (12-1)$$

其中  $A$ 、 $B$ 、 $C$  为系统的系数矩阵，由式 (12-1) 可知系统为一阶微分方程组的矩阵表达式，因此采用四阶龙格-库塔法进行求解和仿真，其求解步骤和方法如下：

- 1、由  $\dot{x} = Ax + Bu$ ，可知  $f(t, x) = Ax + Bu$ ；
- 2、根据四阶龙格-库塔法的递推公式：

$$\begin{cases} k_1 = f(t_n, x_n) \\ k_2 = f(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1) \\ k_3 = f(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2) \\ k_4 = f(t_n + h, x_n + hk_3) \\ x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases} \quad (12-2)$$

其中  $k_1, k_2, k_3, k_4$  为对应  $n$  维状态空间变量  $x = [x_1, x_2, \dots, x_n]^T$  在  $t_{n+1}$  时刻的四组导数，每组为  $n$  维列向量。

3、由 (12-2) 式可知  $t_{n+1}$  时刻的状态为  $x_{n+1}$ ，并得到

$$y_{n+1} = Cx_{n+1} \quad (12-3)$$

根据 (12-2) 式和 (12-3) 式，取  $n = 0, 1, 2, \dots, N$  不断递推，便可得到所需时刻各点的状态变量  $x(t_n)$  和输出量  $y(t_n)$ 。

## 12.1.2 闭环模型的建立

典型闭环控制系统的方框图如图 12.1.1 所示。

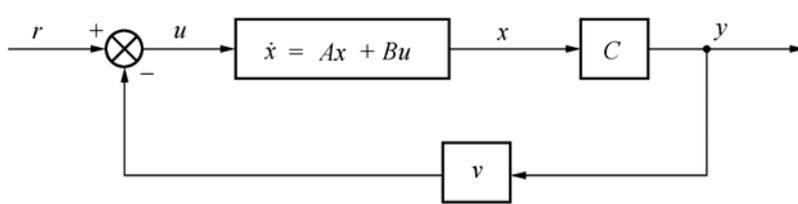


图 12.1.1 典型闭环控制系统的方框图

对单输入单输出系统  $r, u, y, v$  均为标量，由图 12.1.1 可知  $u = r - vy$ ，将其代入 (12-1) 式得

$$\dot{x} = Ax + B(r - vy)$$

又由  $y = Cx$

可得到系统的闭环状态方程：

$$\dot{x} = (A - BvC)x + Br = A_b x + Br \quad (12-4)$$

其中  $A_b = A - BvC$ ，称为系统闭环系数矩阵，而输入矩阵  $B$  和输出矩阵  $C$  保持不变。

ode4() 函数实现上述算法，其程序框图见图 12.1.2。

【调用格式】

$$[t, y] = \text{ode4}(A, B, C, D, x_0, h, r, v, t_0, t_f)$$

【说明】 $\{A, B, C, D\}$  为系统的系数矩阵， $x_0$  为状态向量初值， $h$  为仿真步长， $r$  为输入信号的幅值， $v$  为反馈系数， $t_0$  为仿真的起始时间， $t_f$  终止时间， $y$  为输出量，ode4() 函数参见附录 D：D-12-1。

MATLAB 中的 ode45() 函数可实现四阶/五阶龙格-库塔算法。

【调用格式】

`[t,y]=ode45('f',tspa,x0)`

【说明】`f`为定义的常微分方程函数名,`tspa`为起止时间向量,`x0`为初始状态向量,`y`为输出向量,`t`为仿真时间。

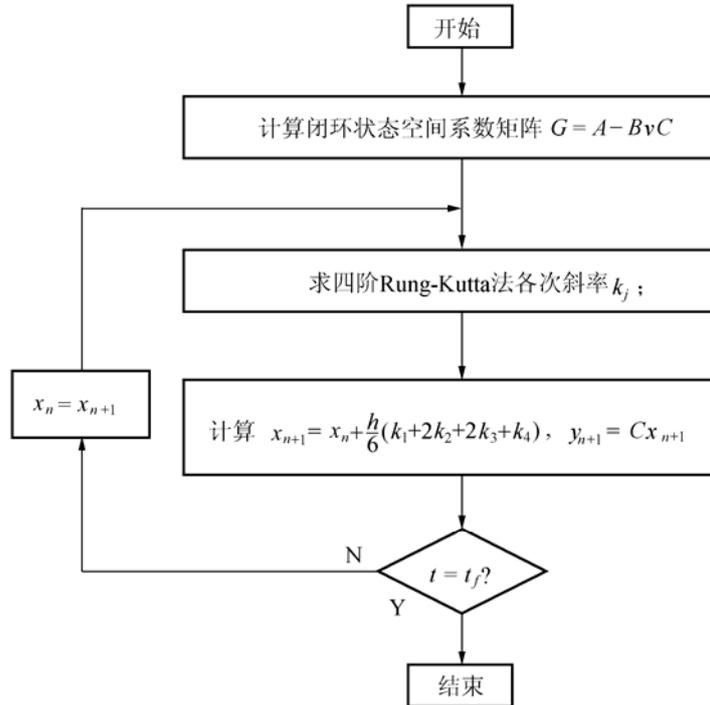


图12.1.2 四阶龙格-库塔法仿真程序框图

例12.1.1系统的开环传递函数为：

$$G(s) = \frac{k(2s + 1)}{s^2(0.25s + 1)^2}$$

在零初始条件下，当输入信号的幅值为1时，试绘制单位负反馈系统的仿真曲线。

解：运行如下程序，并得到如图 12.1.3 所示的仿真曲线。

```

k=1;num=[2*k,k];den=conv([1,0,0],conv([.25,1],[.25,1]));
[A,B,C,D]=tf2ss(num,den);
x0=[0,0,0,0];v=1;t0=0;tf=10;h=0.25;r=1;
[t,y]=ode4(A,B,C,D,x0,h,r,v,t0,tf);
plot(t,y),grid on
    
```

若要分析开环放大系数 `k`、反馈系数 `v` 的变化对系统的影响作用时，只需分别对 `k`、`v` 重新赋值，其余参数不变，再次运行上述程序即可。

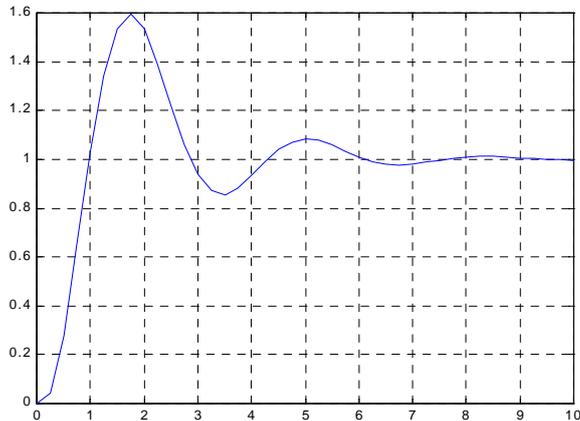


图 12.1.3 系统的仿真曲线

例 12.1.2 系统的状态方程为：

$$\dot{x} = \begin{bmatrix} -14 & 9 & 10 \\ 12 & -9 & 10 \\ 24 & -24 & -18 \end{bmatrix} x + \begin{bmatrix} 3 \\ 4 \\ 12 \end{bmatrix} u$$

在零初始条件下，阶跃信号的输入幅值为 100，试应用状态空间法对系统进行仿真。

解：应用 ode45（）函数，运行下面的程序，可得到仿真曲线如图 12.1.4 所示。

1、定义状态方程的函数

```
function sys=ab(t,x)
u=100;
sys=[-14*x(1)+9*x(2)+10*x(3)+3*u;12*x(1)-9*x(2)+10*x(3)+4*u;...
24*x(1)-24*x(2)-18*x(3)+12*u];
```

2、应用四阶/五阶龙格-库塔算法对状态方程求解

```
x0=[0,0,0]',tspa=[0,2];
[t,y]=ode45('ab',tspa,x0);
plot(t,y),grid
```

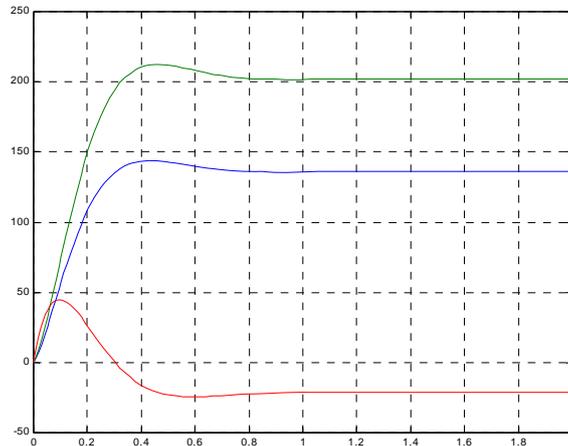


图12.1.4 系统的仿真曲线

## 12.2 非线性环节的仿真

若系统中含有非线性元件，人们为了简化问题，常采用非线性环节线性化的方法将其理想化，并按线性系统理论对其进行分析、设计。然而对于存在明显非线性的系统，则必须按照非线性系统进行研究，因此通过对典型非线性环节的分析，我们给出了其相应的 MATLAB 仿真程序函数，在对非线性系统进行分析与设计时可调用这些函数，从而达到对非线性系统仿真的目的。

### 12.2.1 饱和非线性环节仿真

由图 12.2.1 所示饱和非线性环节的数学表达式：

$$u_c = \begin{cases} -s_1 & u_r \leq -s_1 \\ u_r & -s_1 < u_r < s_1 \\ s_1 & u_r \geq s_1 \end{cases} \quad (12-5)$$

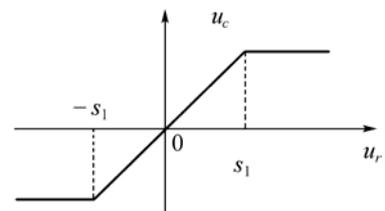


图 12.2.1 饱和非线性环节

根据式 (12-5) 编写了相应的MATLAB函数，其程序框图见图12.2.2。

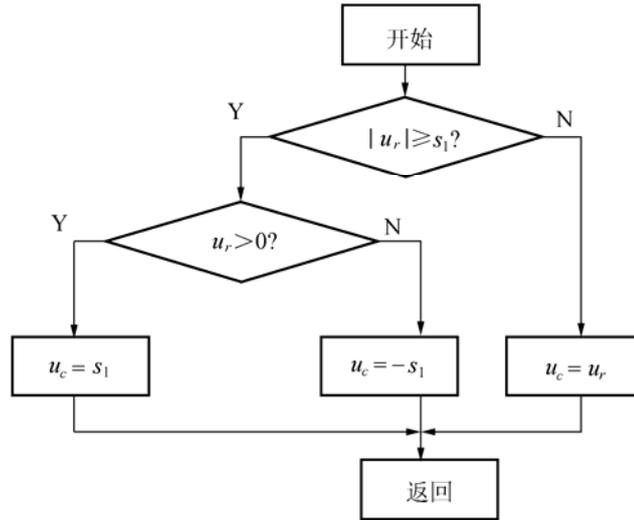


图12.2.2饱和非线性环节仿真程序框图

【调用格式】

$uc = \text{Saturation}(ur, s1)$

【说明】 $ur$ 为输入向量, $s1$ 为非线性环节的饱和值， $uc$ 为输出向量， $\text{Saturation}()$ 函数参见附录D: D-12-2。

## 12.2.2 死区非线性环节仿真

由图 12.2.3 所示死区非线性环节的数学表达式：

$$u_c = \begin{cases} u_r + s_1 & u_r \leq -s_1 \\ 0 & -s_1 < u_r < s_1 \\ u_r - s_1 & u_r \geq s_1 \end{cases} \quad (12-6)$$

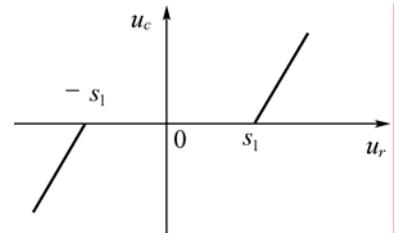


图 12.2.3 死区非线性环节

根据式 (12-6) 编写了相应的MATLAB函数，其程序框图见图12.2.4。

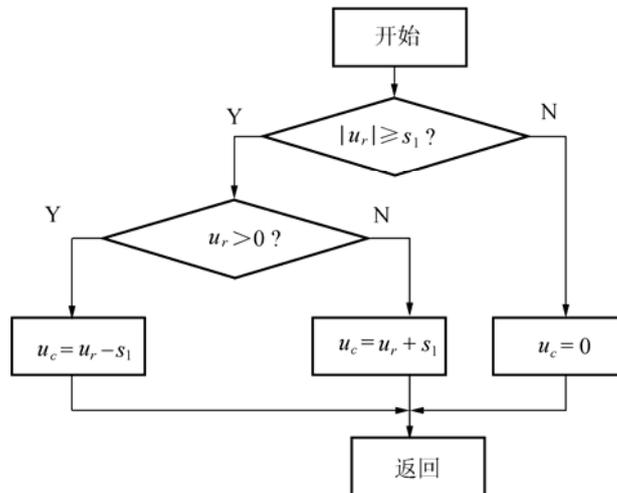


图 12.2.4 死非线性环节仿真程序框图

【调用格式】

$uc = \text{DeadZone}(ur, s1)$

【说明】 $s1$ 为死区非线性环节的死区值，其它参数定义与 $\text{Saturation}()$ 函数相同， $\text{DeadZone}()$ 函数参见附录D: D-12-3。

### 12.2.3 间隙非线性环节仿真

由图 12.2.5 所示间隙非线性环节的数学表达式：

$$u_c = \begin{cases} u_r - s_1 & \dot{u}_r > 0 \text{ 且 } \dot{u}_c > 0 \\ u_r + s_1 & \dot{u}_r < 0 \text{ 且 } \dot{u}_c < 0 \\ u_{cs} & \dot{u}_r < 0 \text{ 且 } \dot{u}_c = 0 \\ u_{cs} & \dot{u}_r > 0 \text{ 且 } \dot{u}_c = 0 \end{cases} \quad (12-7)$$

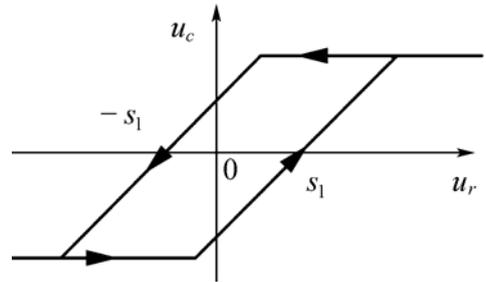


图 12.2.5 间隙非线性环节

根据式 (12-7) 编写了相应的MATLAB函数，其程序框图见图12.2.6

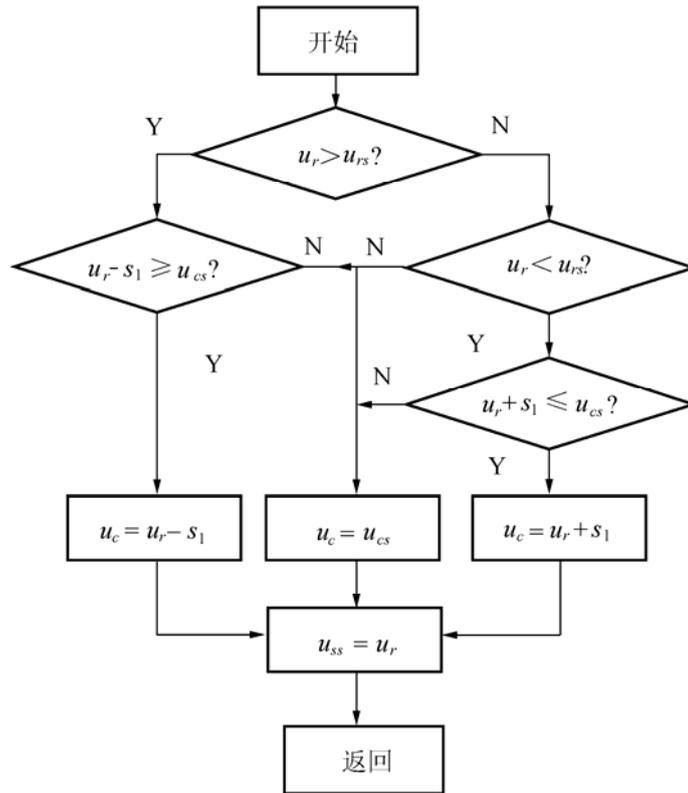


图 12.2.6 间隙非线性环节仿真程序框图

【调用格式】

$$[uc,uss]=backlash(urs,ur,ucs,s1)$$

【说明】s1为环节的间隙宽度值，urs，ucs为ur，uc前一时刻的值，uss为下次运算保留的输入值，其它参数定义与Saturation()函数相同，backlash()函数参见附录D：D-12-4。

### 12.3 离散系统仿真

目前，离散系统的最广泛应用形式是以数字计算机为控制器的所谓数字控制系统，其方框图如图 12.3.1 (a) 所示。模拟信号经过采样开关和 A/D 转换器，按一定的采样周期 T 转换为数字信号，经计算机或其它数字控制器处理后，再经采样保持器和 D/A 转换器将数字信号转换为模拟信号来控制被控对象，以实现数字控制，图 12.3.1 (b) 是对图 12.3.1 (a) 的简化。

离散系统的数学模型一般用差分方程和离散状态方程来描述。

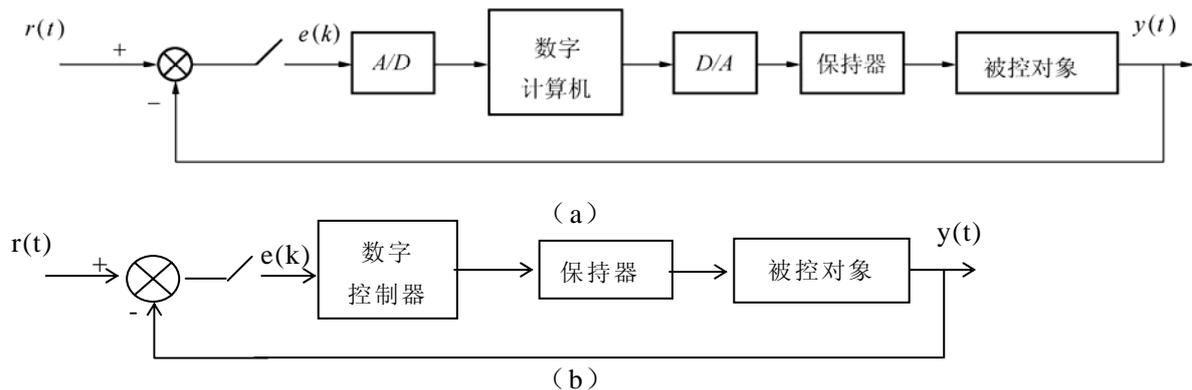


图 12.3.1 数字控制系统方框图

### 12.3.1 差分方程法

连续系统的动态性能是由微分方程组描述的，而离散控制系统对系统中的变量的测量是不连续的，只能测得这些变量在采样时刻  $0、T、2T、3T、\dots nT$  的数值，因此，离散控制系统的动态性能是用差分方程描述的，其系统仿真的步骤如下：

- 1、根据系统的结构图，在适当位置加设虚拟采样开关和保持器；
- 2、将原系统转换成状态空间形式，并按指定的采样周期，依照离散化方法，将系统离散化，并得到离散化的状态方程，即系统的差分方程

$$\begin{cases} x(k+1) = Gx(k) + Hu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \quad k = 0, 1, 2, \dots \quad (12-8)$$

- 3、输入系统初始化参数，并根据差分方程编写仿真程序。
- 根据式 (12-8) 编写了相应的函数，其程序框图见图12.3.2。

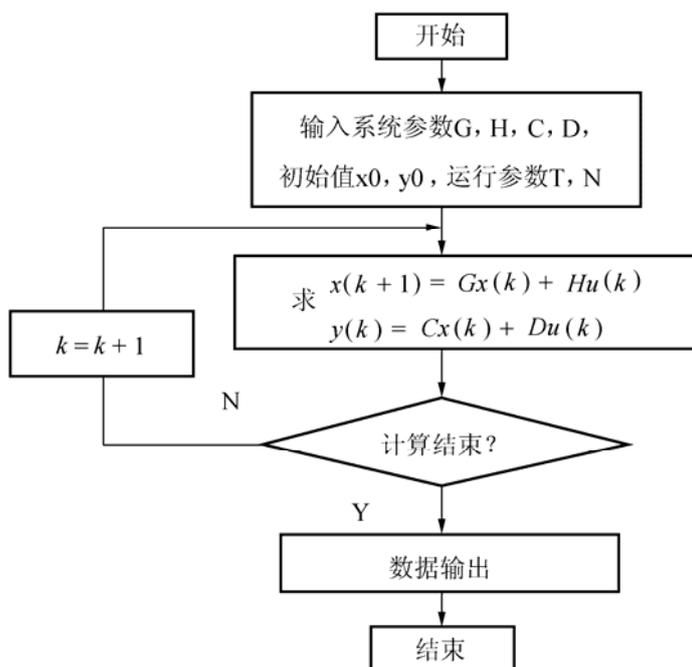


图 12.3.2 差分方程法仿真程序框图

【调用格式】

`[t,xx]=diffstate(G,H,x0,u0,N,T)`

【说明】{G,H}为差分方程的系统矩阵，x0为初始状态值，u0输入向量，N为仿真点数，T为采样周期，t采样时间序列，xx为状态向量，diffstate（）函数参见附录D：D-12-5。

例12.3.1 定常离散系统的状态方程为：

$$x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.16 & -1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

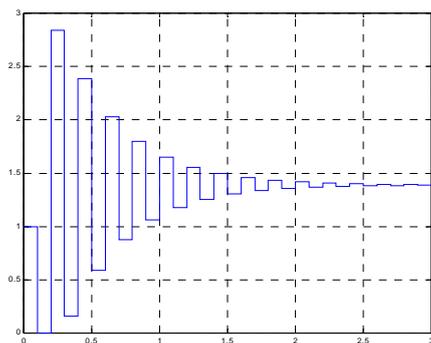
给定初始状态为  $u(k)=1, x(0) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ ，试求解  $x(k)$ ，并绘制其仿真曲线。

解：运行下面的程序，并得到离散系统的仿真曲线如图12.3.3所示

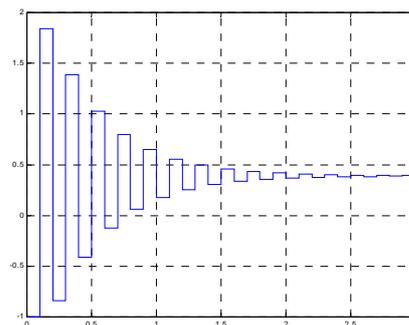
```
G=[0,1;-0.16,-1];H=[1,1]';
x0=[1,-1]';u0=1;N=30;T=0.1;
[t,xx]=diffstate(G,H,x0,u0,N,T);xx,
yk1=[1,0]*xx;yk2=[0,1]*xx;
stairs(t,yk1),grid on,
figure,stairs(t,yk2),grid on
```

运行结果为：

```
xx = 1.0000    0    2.8400    0.1600    2.3856    0.5888    2.0295...
      -1.0000  1.8400   -0.8400    1.3856    0.4112    1.0295   -0.1237...
```



(a)  $x_1(k)$ 的仿真曲线



(b)  $x_2(k)$ 的仿真曲线

图 12.3.3 离散系统的仿真曲线

## 12.3.2 Z 变换法

Z 变换是分析设计离散系统的重要工具之一，它在离散系统中的作用与拉氏变换在连续系统中的作用是相似的。

在控制系统中的大部分 MATLAB 命令，在离散控制系统中都有对应，通常以字母 d 开头，其用法与格式与连续控制系统几乎相同。我们采用前面章节对离散系统分析中的介绍的函数，直接应用 Z 变换法对离散系统进行仿真。

例 12.3.2 系统如图 12.3.4 所示，试求零初始条件下系统的开环和闭环离散化状态方程，并绘制系统的单位阶跃响应曲线。

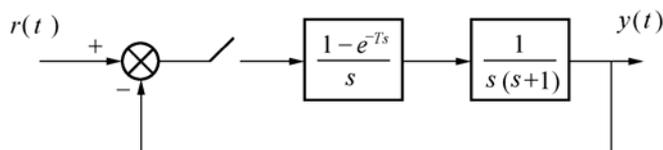


图 12.3.4 离散系统的方框图

解：1、求系统开环和闭环离散化状态方程

```
num=[1];den=conv([1,0],[1,1]);
```

```
[A,B,C,D]=tf2ss(num,den);
```

```
g=ss(A,B,C,D);
```

```
N=fliplr(eye(2));
```

```
g=ss2ss(g,N);
```

```
[A,B,C,D]=ssdata(g);
```

```
T=0.1;[G,H]=c2d(A,B,T),
```

```
G=G-H*C,H=H,C=C,D=D;
```

运行结果：

```
G=1.0000 0.0952
```

```
0 0.9048
```

```
H=0.0048 C = 0.9952 0.0952
```

```
0.0952 -0.0952 0.9048
```

即系统的开环离散化状态方程为：

$$x(k+1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.9048 \end{bmatrix} x(k) + \begin{bmatrix} 0.0048 \\ 0.0952 \end{bmatrix} u(k)$$

系统的闭环离散化状态方程为

$$x(k+1) = \begin{bmatrix} 0.9952 & 0.0952 \\ -0.0952 & 0.9048 \end{bmatrix} x(k) + \begin{bmatrix} 0.0048 \\ 0.0952 \end{bmatrix} u(k)$$

2、绘制闭环离散化系统的仿真曲线

```
dstep(G,H,C,D), grid on
```

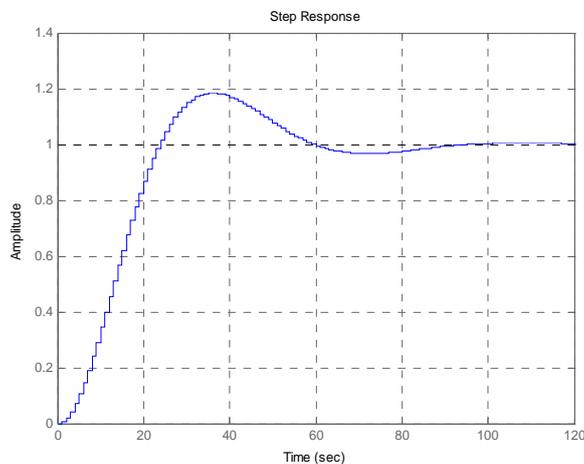


图12.3.5离散系统的仿真曲线

## 12.4 典型环节的 Simulink 仿真

在 MATLAB 环境下，可以直接使用 sim()函数实现 Simulink 仿真。

【调用格式】

```
[t,x,y]=sim('model',timespan,option,ut)
```

【说明】‘model’为 Simulink 生成的模型文件名；timespan 为仿真时间设置，可指定终止时间和起止时间；其余参数为可选，option 是用于设置初始条件、步长与容许误差等值，ut 为外部输入信号。

在 Simulink 中系统默认采用变步长的四阶/五阶龙格-库塔积分算法即 ode45 ( ) 函数进行仿真，若采用其他算法可在算法设置中进行选择。

## 12.4.1 带延迟环节系统的 Simulink 仿真

**例 12.4.1** 系统的 Simulink 结构图 (t1241.mdl) 如图 12.4.1 所示，其中，延迟环节的延迟时间为 1 秒，试绘制系统的仿真曲线。

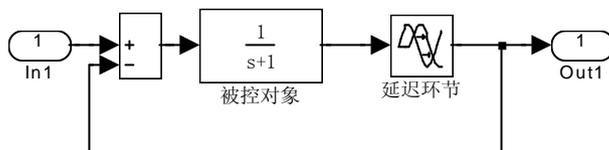


图 12.4.1 带延迟环节的系统 Simulink 结构图

解：运行下面的程序，得到系统的仿真曲线如图 12.4.2 所示。

```
t=[0:0.1:9.9]';ut=[t,ones(size(t))];
[tt,xx,yy]=sim('t1241',8,[],ut);
plot(tt,yy);grid on,
```

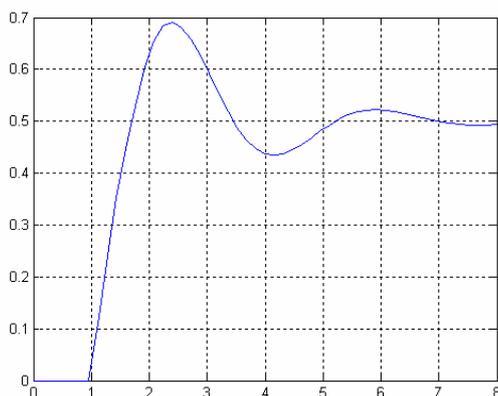


图12.4.2 带延迟环节系统的仿真曲线

**例 12.4.2** 钢铁厂车间加热炉传递函数与温度传感器及其变送器的传递函数模型分别为：

$$G_{01} = \frac{9.9}{120s+1} e^{-80s}, \quad G_{02} = \frac{0.107}{10s+1}$$

设定控制所用的 PID 调节器传递函数为

$$G_c = \frac{9286s^2 + 240s + 1.5}{521s^2 + 145s}$$

试对系统的 PID 控制进行分析、设计与仿真。

解：1、在 Simulink 的 Library 窗口中选择 **【File】|【New】**，打开一个新的工作空间；

2、分别将信号源库、输出方式库、数学运算库、线性环节库中的 Step、Scope、Transfer Fcn、Transport Delay 以及 Sum 各功能模块拖至工作平台；

3、按系统要求将各模块加以连接，如图 12.4.3 所示，并对各模块进行参数设置，如设置滞后环节的延迟时间为 80 秒；

4、运行 **【Simulation】|【Parameters】**，对系统的仿真参数进行设置，如仿真时间 2000 秒，算法为变步长 ode45；

5、执行 **【Simulation】|【Start】**，双击 Scope 图标可得到系统的仿真曲线见图 12.4.4。

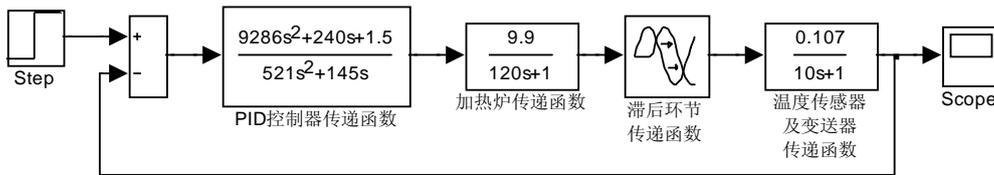


图 12.4.3 PID 系统的 Simulink 结构图

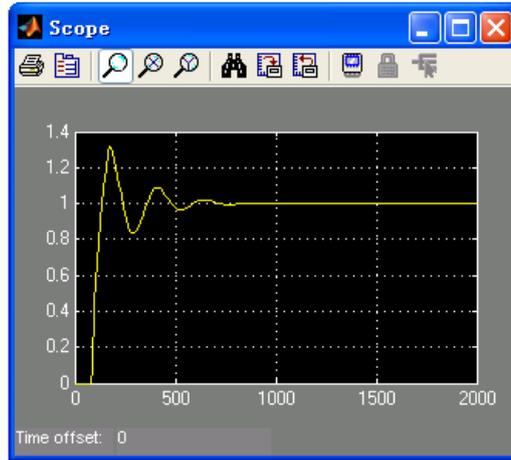


图 12.4.4 PID 系统仿真曲线

## 12.4.2 含非线性环节系统的 Simulink 仿真

**例12.4.3** 含饱和和非线性环节的系统方框图如图12.4.5所示，初始状态为零，饱和非线性环节的饱和值 $c=0.5$ ，试对系统含有饱和和非线性环节前后进行仿真，并绘制其单位阶跃响应曲线。

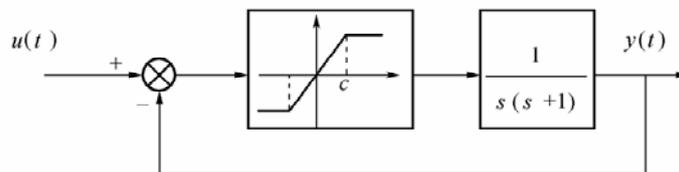


图 12.4.5 含有饱和和非线性系统的方框图

解：1、绘制系统含有饱和和非线性环节前后的Simulink结构图（t1246.mdl和t1247.mdl），如图12.4.6、图12.4.7所示。

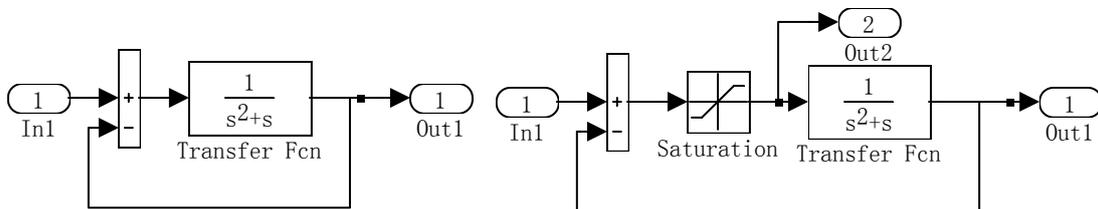


图12.4.6原系统的Simulink结构图

图12.4.7含饱和和非线性环节系统的Simulink结构图

2、运行下面的程序，得到系统含有饱和和非线性环节前后的单位阶跃响应曲线如图12.4.8所示，其中虚线表示图12.4.6所示单位阶跃响应曲线，实线表示图12.4.7所示单位阶跃响应曲线，图12.4.9为饱和和非线性环节输出仿真曲线。

```

t=[0:0.1:9.9]';ut=[t,ones(size(t))];
[tt,xx,yy]=sim('t1246',10,[],ut);
plot(tt,yy,'--'),grid,hold on,
[tt,xx,yy]=sim(' t1247',10,[],ut);
plot(tt,yy(:,1)),hold off,

```

figure,plot(tt,yy(:,2)),grid

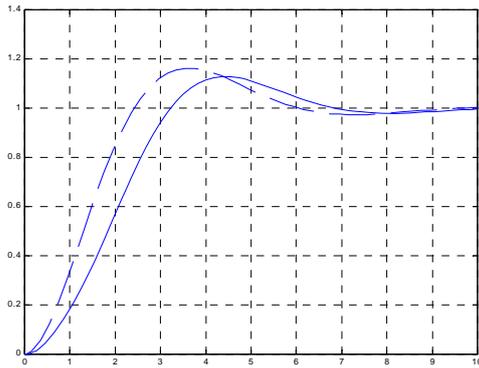


图12.4.8 含饱和和非线性环节的单位阶跃响应曲线

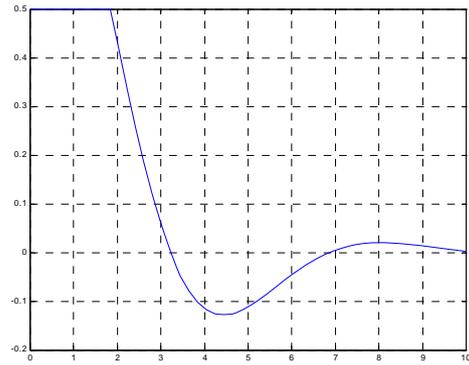


图12.4.9饱和和非线性环节输出仿真曲线

例 12.4.4 含间隙非线性环节的系统方框图如图 12.4.10 所示，设输入幅值为 10，间隙非线性性的宽度为 1，试对包含非线性环节前后的系统进行仿真。

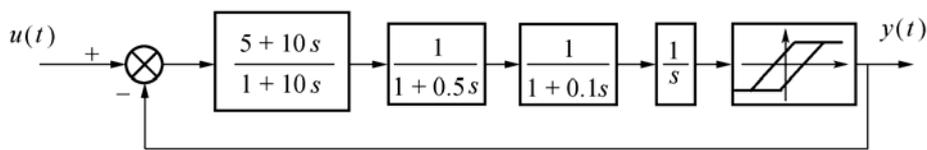


图 12.4.10 间隙非线性环节的系统方框图

- 解：1、在Simulink的Library窗口中选择【File】|【New】，打开一个新的工作空间；
- 2、分别从信号源库、输出方式库、数学运算库、线性环节库和非线性环节库中将Step、Scope、Transfer Fcn、Backlash、Mux以及Sum各功能模块拖至工作平台；
- 3、将各模块按系统要求加以连接，并设置各模块的参数，如设置“Backlash”模块的间隙宽度为1，如图12.4.11所示；
- 4、运行【Simulation】|【Parameters】，对系统的仿真参数进行设置，如设仿真时间为10秒，算法选择默认方式；
- 5、执行【Simulation】|【Start】，双击Scope可得到加入间隙非线性环节前后系统的仿真曲线见图12.4.12。

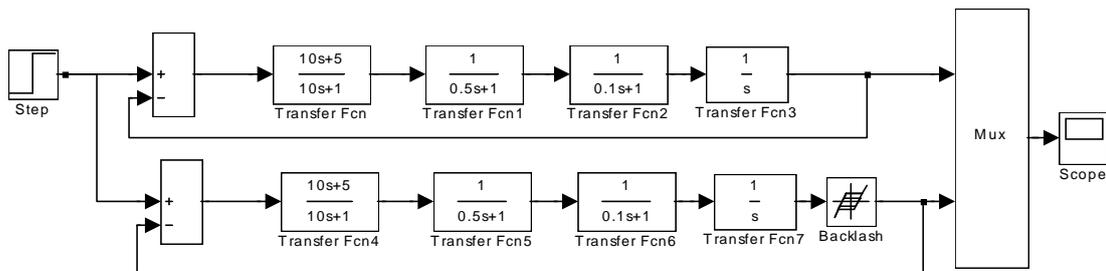


图 12.4.11 系统包含间隙非线性环节前后的 Simulink 的结构图

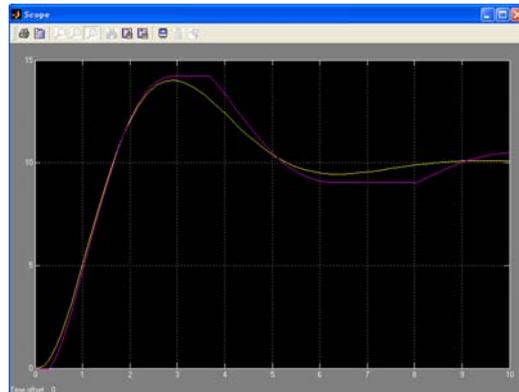


图 12.4.12 包含间隙非线性环节前后系统的仿真曲线

## 12.4.3 离散系统的 Simulink 仿真

例 12.4.4 系统的 Simulink 结构图 (t12413.mdl) 如图 12.4.13 所示, 文件名为 t12413.mdl, 试对离散线性系统进行仿真并求取其频域指标。

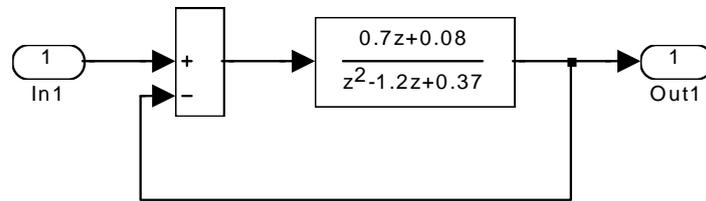


图 12.4.13 二阶离散线性系统的 Simulink 结构图

解: 运行下面的程序, 得到离散系统的仿真曲线如图12.4.14所示

```
t=[0:0.1:9.9]';ut=[t,ones(size(t))];  
[tt,xx,yy]=sim('t12413',3,[],ut);  
stairs(tt,yy),grid,  
num=[0.7 0.08];den=[1 -1.2 0.37];  
[mag,phase,w]=dbode(num,den,0.1);  
[gm,pm,wcg,wcp]=margin(mag,phase,w)
```

运行结果:

gm = 4.1452, pm = 41.6575, wcg =31.4159, wcp =10.4075

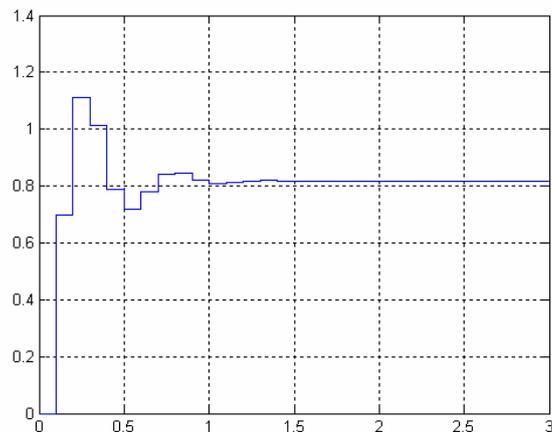


图 12.4.14 二阶离散系统的阶跃响应曲线

## 小 结

本章从控制系统的仿真形式入手, 叙述了对控制系统仿真的常用方法, 并给出了相应的 MATLAB 仿真实现的程序。

1、系统的数学模型无论是传递函数或是状态空间形式, 都可应用状态空间法对系统进行仿真和分析, 使用的技巧与认识需要不断应用, 才能真正学习和领会控制系统仿真的原理与实质内容;

3、当系统中含有非线性、离散等环节时, 可通过环节定义将它们编写为 MATLAB 函数, 这不仅有利于仿真的实现, 而且可反复应用。

4、应用 Simulink 进行系统的动态仿真, 不仅适用于连续线性系统, 而且使离散系统、非线性系统及多输入—多输出系统的仿真变得简单、方便。

5、在使用 Simulink 中, 需要注意合理地选择仿真方法和步长, 如果选择不当, 可能导致系统发散或精度过低。

