** **

**自动控制原理Ⅱ：线性系统分析与设计**

**实验报告**

|  |  |  |
| --- | --- | --- |
| **指导老师** | ： | 张传科 |
| **姓名** | ： |  |
| **班级** | ： |  |
| **学号** | ： |  |

二〇二一年十月

**目录**

[第一章 实验简介 1](#_Toc89822320)

[1.1 实验目标 1](#_Toc89822321)

[1.2 实验内容 1](#_Toc89822322)

[1.3 实验平台 1](#_Toc89822323)

[第二章 线性系统模型部分实验 2](#_Toc89822324)

[2.1 实验目标 2](#_Toc89822325)

[2.2 实验内容1（作业一第一题） 2](#_Toc89822326)

[2.2.1 实验要求 2](#_Toc89822327)

[2.2.2 实验程序 2](#_Toc89822328)

[2.2.3 实验结果 3](#_Toc89822329)

[2.3 实验内容2（作业一第二题） 6](#_Toc89822330)

[2.3.1 实验要求 6](#_Toc89822331)

[2.3.2 实验程序 7](#_Toc89822332)

[2.3.3 实验结果 7](#_Toc89822333)

[2.4 小结 8](#_Toc89822334)

[第三章 线性系统分析部分实验 9](#_Toc89822335)

[3.1 实验目标 9](#_Toc89822336)

[3.2 实验内容1（作业一第一题） 9](#_Toc89822337)

[3.2.1 实验要求 9](#_Toc89822338)

[3.2.2 实验程序 9](#_Toc89822339)

[3.2.3 实验结果 10](#_Toc89822340)

[3.3 实验内容2（作业一第二题） 12](#_Toc89822341)

[3.3.1 实验要求 12](#_Toc89822342)

[3.3.2 实验程序 12](#_Toc89822343)

[3.3.3 实验结果 14](#_Toc89822344)

[3.4 实验内容3（作业一第三题） 16](#_Toc89822345)

[3.4.1 实验要求 16](#_Toc89822346)

[3.4.2 实验程序 16](#_Toc89822347)

[3.4.3 实验结果 18](#_Toc89822348)

[3.5 小结 21](#_Toc89822349)

[第四章 线性系统设计部分实验 23](#_Toc89822350)

[4.1 实验目标 23](#_Toc89822351)

[4.2 实验内容1（作业一第一部分） 23](#_Toc89822352)

[4.2.1 实验要求 23](#_Toc89822353)

[4.2.2 实验程序 23](#_Toc89822354)

[4.2.3 实验结果 25](#_Toc89822355)

[4.3 实验内容2（作业一第二部分） 27](#_Toc89822356)

[4.3.1 实验要求 27](#_Toc89822357)

[4.3.2 实验程序 28](#_Toc89822358)

[4.3.3 实验结果 30](#_Toc89822359)

[4.4 实验内容3（作业一第三部分） 30](#_Toc89822360)

[4.4.1 实验要求 30](#_Toc89822361)

[4.4.2 实验程序 30](#_Toc89822362)

[4.4.3 实验结果 33](#_Toc89822363)

[4.5 小结 33](#_Toc89822364)

[第五章 实验总结与体会 35](#_Toc89822365)

[5.1 实验总结 35](#_Toc89822366)

[5.2 实验体会 35](#_Toc89822367)

# 第一章 实验简介

## 1.1 实验目标

1.加深对线性系统分析与设计理论知识的理解与掌握

2.掌握基于MATLAB软件的系统建模/分析/设计常用方法

## 1.2 实验内容

1.系统模型部分

2.系统分析部分

3.系统设计部分

## 1.3 实验平台

MATLAB R2020b

# 第二章 线性系统模型部分实验

## 2.1 实验目标

1.线性系统各类数学模型的表示与相互转化。

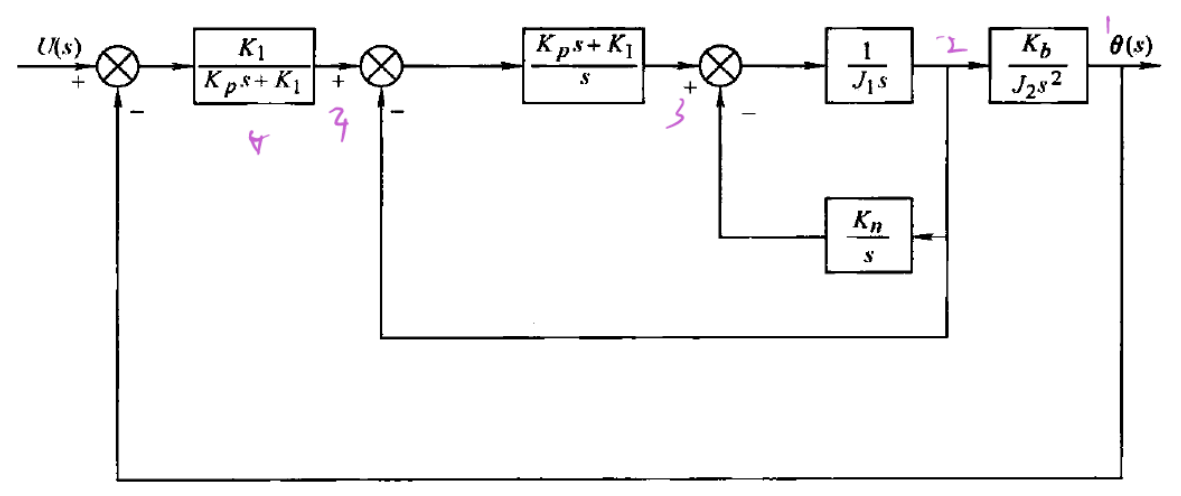
2.多个子系统的连接合并。

## 2.2 实验内容1（作业一第一题）

### 2.2.1 实验要求

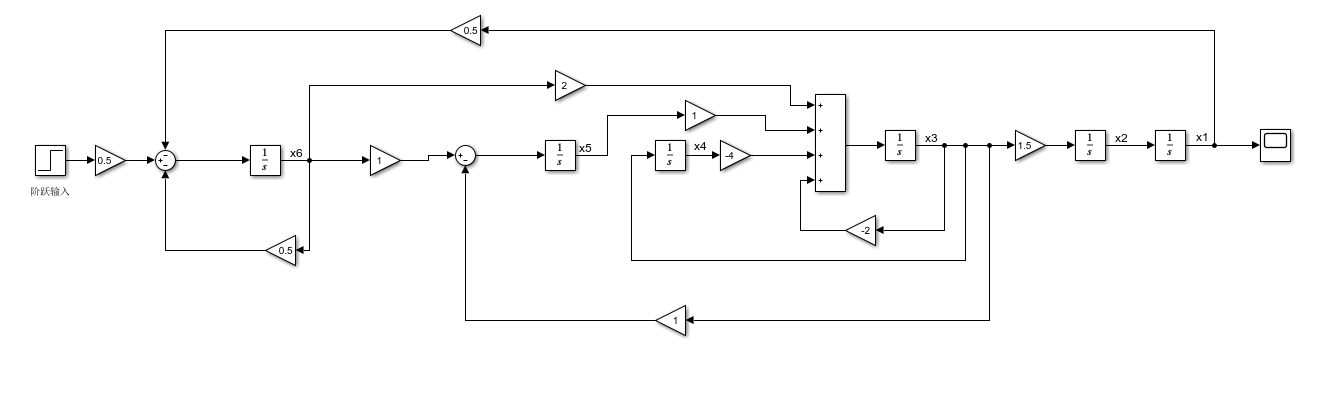
（1）利用MATLAB给出如下系统的传递函数模型、状态空间模型

（2）利用simulink搭建各类模型，并观察其单位阶跃响应（相关参数自由取值）



### 2.2.2 实验程序

1. %%  K1=1，Kp=2,Kb=3,Kn=4,J1=1,J2=2
2. clc;clear;
3. sys1=tf([1],[1 0]); %创建各子系统的传函模型
4. sys2=tf([4],[1 0]);
5. sys3=tf([2 1],[1 0]);
6. sys4=tf([1],[1]);
7. sys5=tf([1],[2,1]);
8. sys6=tf([3],[2 0 0]);
9. sys\_a=feedback(sys1,sys2,-1); %对子系统进行串连、反馈连接
10. sys\_b=series(sys3,sys\_a);
11. sys\_c=feedback(sys\_b,sys4,-1);
12. sys\_d=series(sys\_c,sys5);
13. sys\_e=series(sys\_d,sys6);
14. sys\_all=feedback(sys\_e,sys4,-1) %得到整个系统的传函
15. sysr=minreal(sys\_all) %零极点对消，得到能控能观子系统
16. num\_all=sys\_all.num{1}; %花括号{}进行索引来访问元胞的内容,提取传函的分子
17. den\_all=sys\_all.den{1}; %提取传函的分母
18. [A,B,C,D]=tf2ss(num\_all,den\_all)  %系统状态空间表达式
19. numr=sysr.num{1}; %花括号{}进行索引来访问元胞的内容,提取传函的分子
20. denr=sysr.den{1}; %提取传函的分母
21. [Ar,Br,Cr,Dr]=tf2ss(numr,denr)  %系统状态空间表达式



### 2.2.3 实验结果

sys\_all =

6 s^2 + 3 s

----------------------------------------------

4 s^6 + 10 s^5 + 24 s^4 + 10 s^3 + 6 s^2 + 3 s

Continuous-time transfer function.

sysr =

1.5

--------------------------------------

s^4 + 2 s^3 + 5 s^2 + 2.22e-15 s + 1.5

Continuous-time transfer function.

A =

-2.5000 -6.0000 -2.5000 -1.5000 -0.7500 0

1.0000 0 0 0 0 0

0 1.0000 0 0 0 0

0 0 1.0000 0 0 0

0 0 0 1.0000 0 0

0 0 0 0 1.0000 0

B =

1

0

0

0

0

0

C =

0 0 0 1.5000 0.7500 0

D =

0

Ar =

-2.0000 -5.0000 -0.0000 -1.5000

1.0000 0 0 0

0 1.0000 0 0

0 0 1.0000 0

Br =

1

0

0

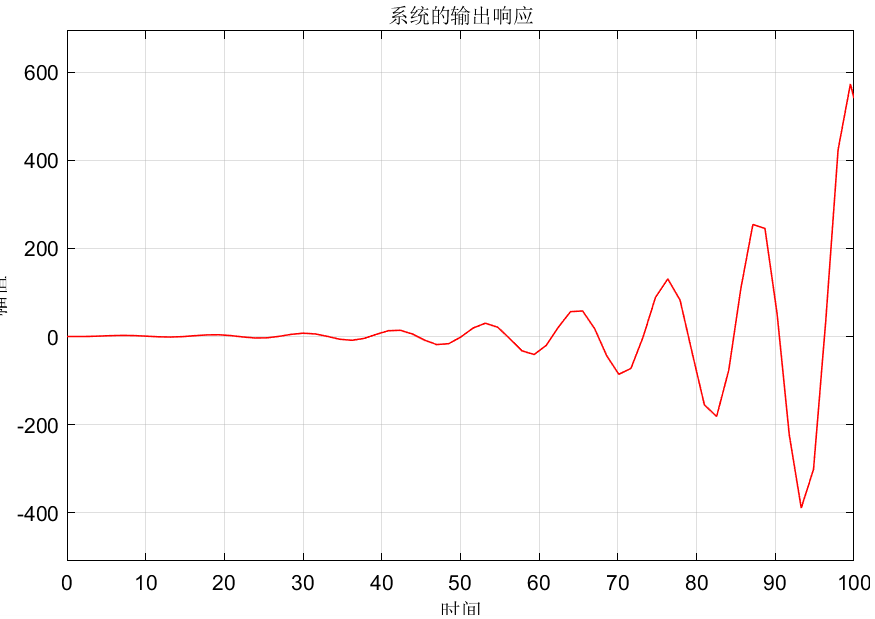
0

Cr =

0 0 0 1.5000

Dr =

0



## 2.3 实验内容2（作业一第二题）

### 2.3.1 实验要求

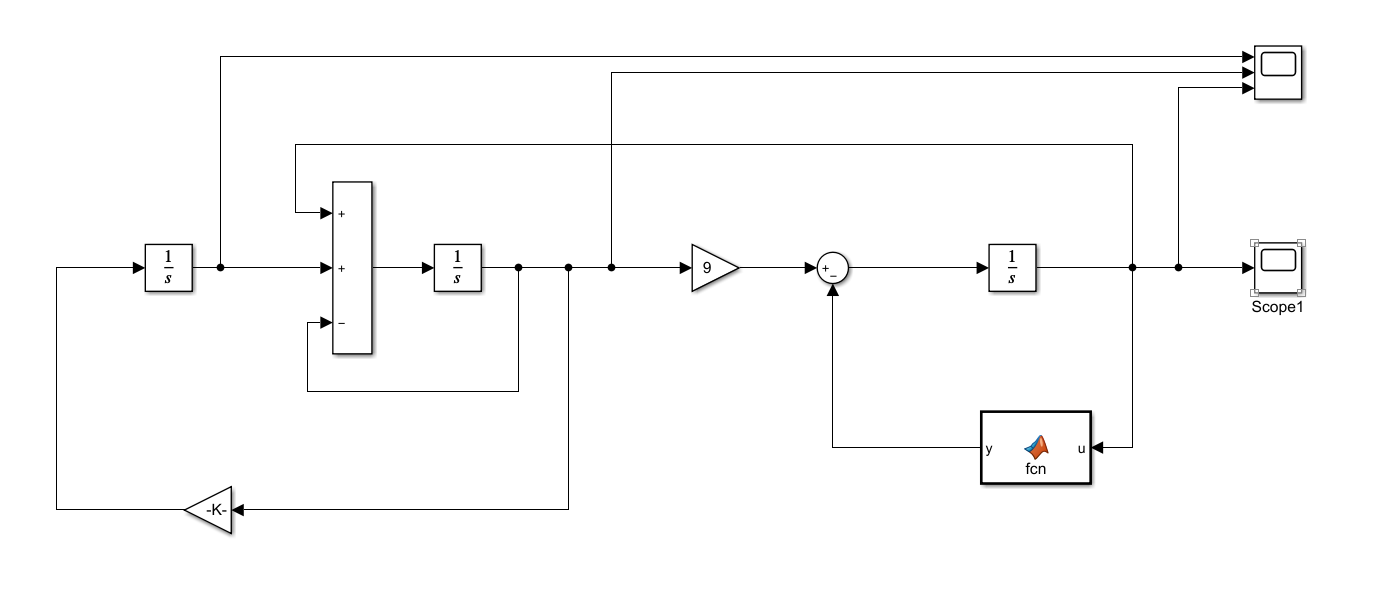
利用simulink搭建如下系统的结构框图，选初始条件[-0.2; 0.3; 0.7]，观察状态响应。

the following representation of Chua’s circuit systems:

with the nonlinear characteristis of Chua’s diode

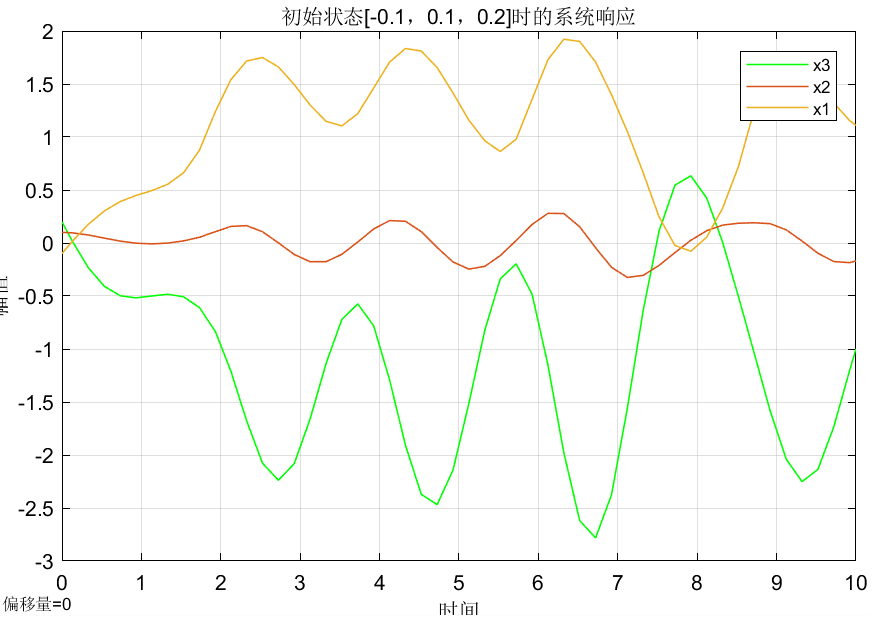
and parameters a=9, b=14.28, c=1, m0=-(1/7), m1=2/7, and c=1.

### 2.3.2 实验程序



1. function y = fcn(u)
2. y=9\*u\*((2/7)\*u+(3/14)\*abs(u-1)-(3/14)\*abs(u+1));

### 2.3.3 实验结果



## 2.4 小结

通过实验中的操作，进一步加深了对LTI系统的状态空间模型、传递函数模型、以及二者之间的相互转换的方法。在观察自己建立的传递函数模型之后，才发现原来matlab把各子系统连接得到的传递函数还存在零极点对消，直接继续下一步得到的状态空间表达式就不是最小实现，于是我使用了minreal()函数来进行零极点对消，然后再转换成状态空间表达式。

第一次使用了Simulink，直接被MATLAB的这个组件的强大与便捷震撼到了，原来对系统的建模和分析不是只能通过编写代码这种“原始”的方法来实现，原来还有这么直观方便的工具，MATLAB不愧是当今三大数学软件之一。Simulink搭建框图的使用方法恰好很符合我们自动化人的习惯，因为平时经常画框图分析系统，这种方式既直观又高效，不禁让人感慨，真值得把simulink好好学一学。刚上手的时候比较懵，主要通过看张老师的讲解视频和Mathworks官网上的视频教程、帮助文档来学习Simulink，尤其是老师的讲解紧贴具体项目，官网的教程全面但偏基础，一起用正好。

# 第三章 线性系统分析部分实验

## 3.1 实验目标

学会用matlab完成以下内容：

①求状态空间表达式的解（绘制系统的响应曲线）；

②LTI系统的能控性/能观性的判断、结构分解；

③LTI系统的稳定性判断；

## 3.2 实验内容1（作业一第一题）

### 3.2.1 实验要求

1.选择两组初值[-0.1,0.1,0.2]、[1,2,3]

2.绘制如下系统的系统响应曲线

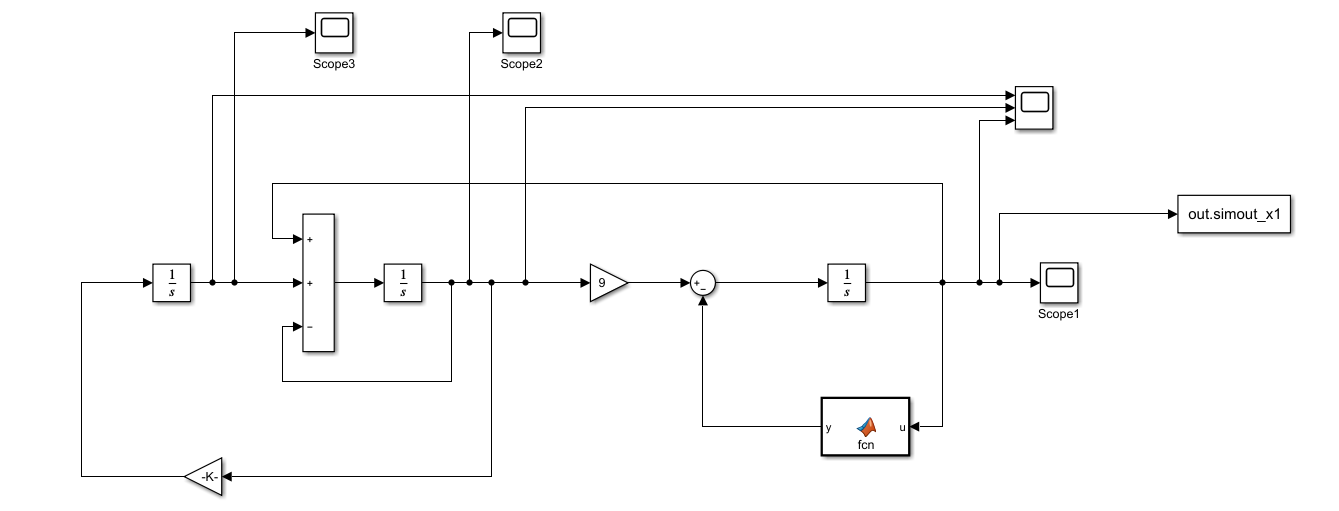
3.绘制如下系统的状态轨迹

the following representation of Chua’s circuit systems:

with the nonlinear characteristis of Chua’s diode

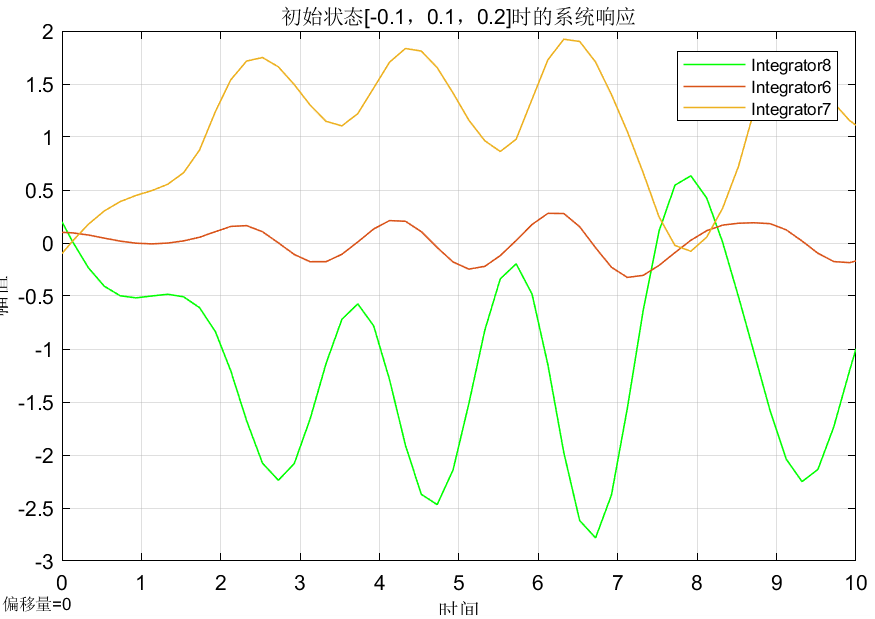
and parameters a=9, b=14.28, c=1, m0=-(1/7), m1=2/7, and c=1.

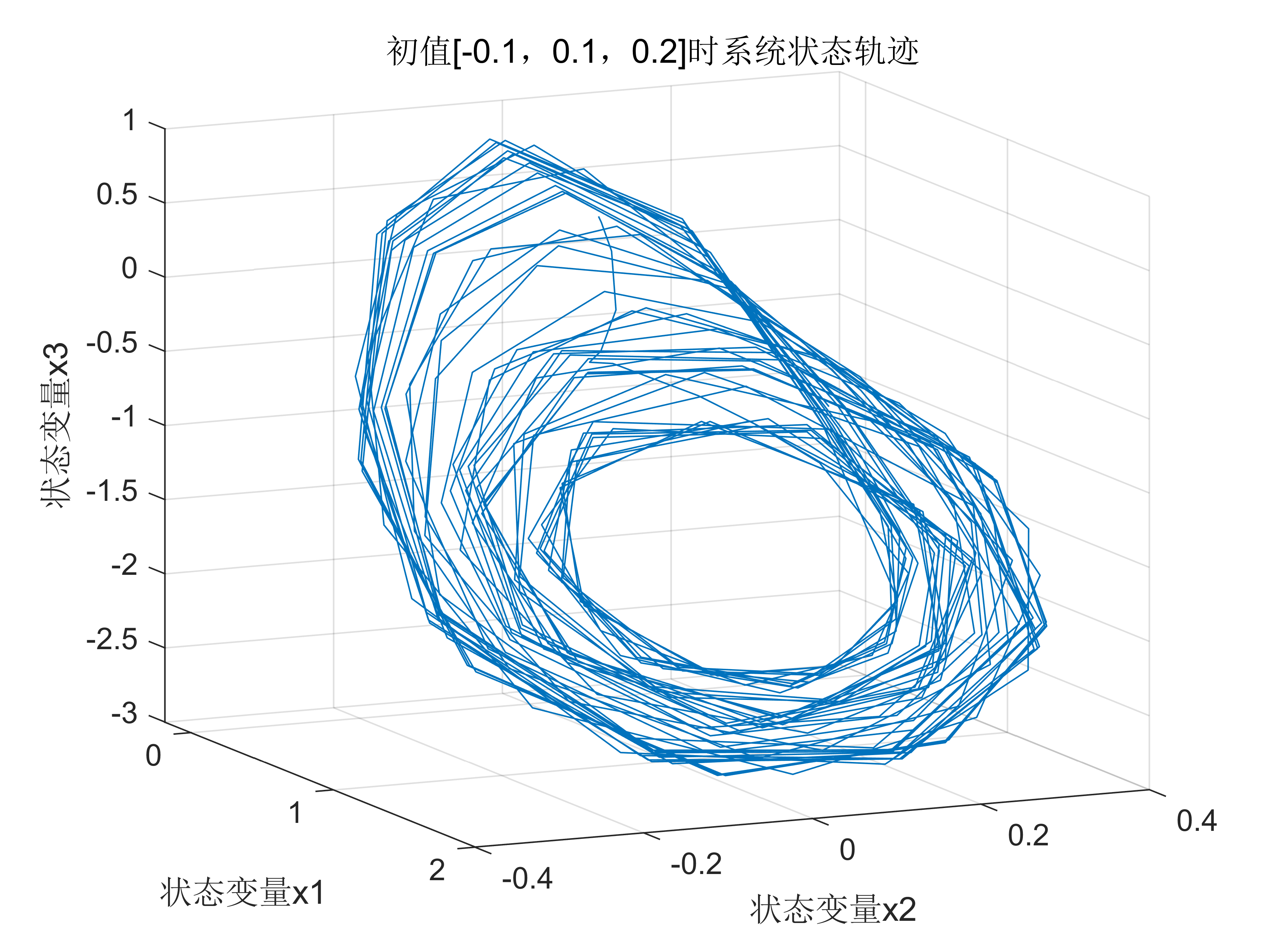
### 3.2.2 实验程序

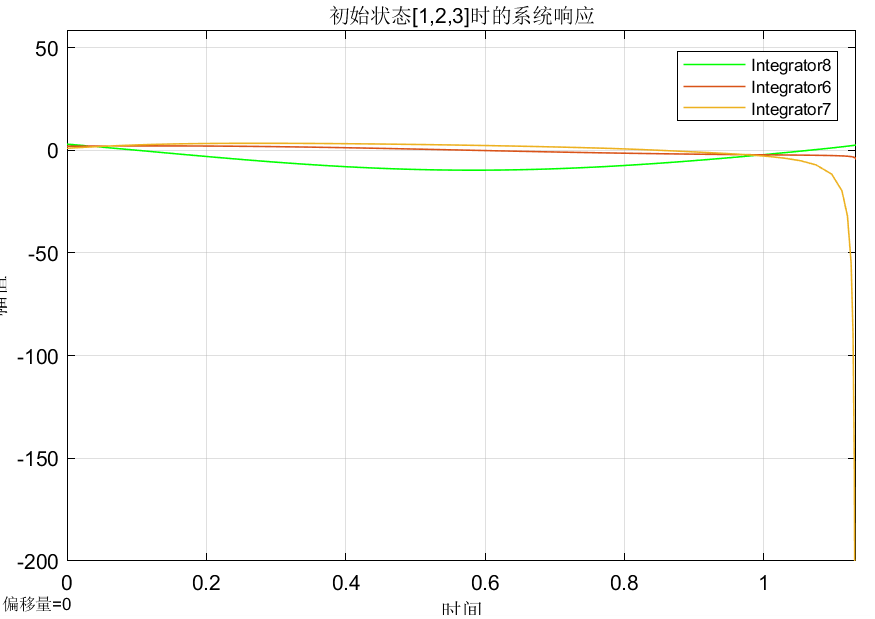


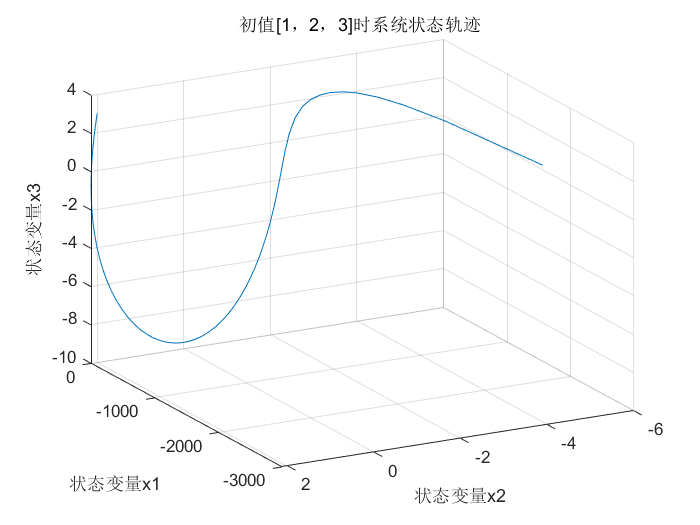
1. figure (1); %以图窗 1 作为当前图窗
2. plot3(out.ScopeData\_x1.signals.values,out.ScopeData\_x2.signals.values,out.ScopeData\_x3.signals.values)
3. grid on; %绘制网格线
4. title('初值[-0.1，0.1，0.2]时系统状态轨迹 '); %设置图标题
5. %title('初值[1，2，3]时系统状态轨迹 '); %设置图标题
6. xlabel('状态变量x1');ylabel('状态变量x2');zlabel('状态变量x3'); %设置横轴、纵轴、竖轴标签

### 3.2.3 实验结果









## 3.3 实验内容2（作业一第二题）

### 3.3.1 实验要求

判断如下系统的能控能观性，若不完全能控且不完全能观，求其能控能观子系统

### 3.3.2 实验程序

1. %% 2 判断如下系统的能控能观性，若不完全能控且不完全能观，求其能控能观子系统
2. clear;clc
3. fprintf('判断如下系统的能控能观性，若不完全能控且不完全能观，求其能控能观子系统');
4. A=[-4 1 0 0 0 0;
5. 0 -4 0 0 0 0;
6. 0 0 3 1 0 0;
7. 0 0 0 3 0 0;
8. 0 0 0 0 -1 1;
9. 0 0 0 0 0 -1]
10. B=[1 3;
11. 5 7;
12. 4 3;
13. 0 0;
14. 1 6;
15. 0 0;]
16. C=[3 1 0 5 0 0;
17. 1 4 0 2 0 0;]
18. n=size(A,1);
19. M=ctrb(A,B);%能控性矩阵
20. N=obsv(A,C);%能观性矩阵
21. fprintf('系统维数n=');disp(size(A,1));
22. fprintf('rank(M)=');disp(rank(M));fprintf('rank(N)=');disp(rank(N));%能控子系统4维，能观子系统3维
23. if (rank(M)==n)&&(rank(N)==n)
24. fprintf('所以系统能控能观\n');
25. else if rank(M)==n
26. fprintf('所以系统能控不能观\n');
27. else if rank(N)==n
28. fprintf('所以系统不能控能观\n');
29. else
30. fprintf('所以系统不能控不能观\n');
31. end
32. end
33. end
34. [Ac0,Bc0,Cc0,Tc0,kc0] = ctrbf(A,B,C);   %对系统按能控性分解，t为变换阵，k为各子系统的秩;
35. Ac1=rot90(Ac0,2); %将ABC矩阵变换回教材上的标准形式
36. Bc1=rot90(Bc0,2); %将数组 A B C 按逆时针方向旋转 k\*90 度
37. Cc1=rot90(Cc0,2);
38. fprintf('能控子系统:Xc''=Ac\*X1+A12\*X2+Bc\*u \t y=C1X1 \n');
39. Ac=Ac1(1:rank(M),1:rank(M)) %从 A 矩阵中提取出 Ac
40. A12=Ac1(1:rank(M),(rank(M)+1):end)
41. Bc=Bc1(1:rank(M),:)
42. Cc=Cc1(:,1:rank(M))
43. disp('下面再对能控子系统进行能观性分解');
44. n1=size(Ac);   %能控子系统的维数 n1
45. [Ao0,Bo0,Co0,To0,ko0] = obsvf(Ac,Bc,Cc);   %对系统按能观性分解，t为变换阵，k为各子系统的秩;
46. Ao1=rot90(Ao0,2); %将ABC矩阵变换回教材上的标准形式
47. Bo1=rot90(Bo0,2); %将数组 A B C 按逆时针方向旋转 k\*90 度
48. Co1=rot90(Co0,2);
49. disp('能控能观子系统: Xco''=A11\*X1+B1\*u  y=C1\*X1');
50. A11=Ao1(1:rank(N),1:rank(N)) %从 Ac 矩阵中提取出A11
51. B1=Bo1(1:rank(N),:)
52. C1=Co1(:,1:rank(N))

### 3.3.3 实验结果

系统维数n= 6

rank(M)= 4

rank(N)= 3

所以系统不能控不能观

能控子系统:Xc'=Ac\*X1+A12\*X2+Bc\*u y=C1X1

Ac =

-2.0033 0.7127 -2.0818 1.0215

0.4902 0.1595 -2.3817 -0.1821

-2.4450 -2.2992 -0.8482 -1.0761

0.3681 -0.3915 -1.6602 -3.3080

A12 =

-0.3961 -0.4845

-0.6144 0.7228

0.6634 0.2830

-0.1596 -0.4034

Bc =

-10.0250 -6.0024

1.5809 -2.6403

-0.0000 0

0.0000 0.0000

Cc =

-3.1995 -0.6800 -2.2964 1.0135

-1.5266 0.4405 0.1531 2.7299

下面再对能控子系统进行能观性分解

能控能观子系统: Xco'=A11\*X1+B1\*u y=C1\*X1

A11 =

-3.5528 0.2764 0.0000

-0.7236 -4.4472 0.0000

0.0000 0.0000 2.8808

B1 =

4.7790 7.5317

1.7780 1.1282

4.1126 3.9909

C1 =

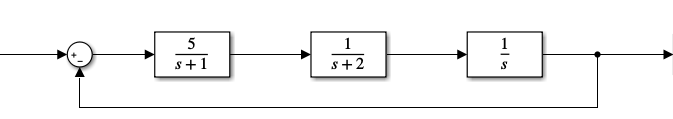
2.4278 -2.0262 0

3.9283 1.2523 -0.0000

## 3.4 实验内容3（作业一第三题）

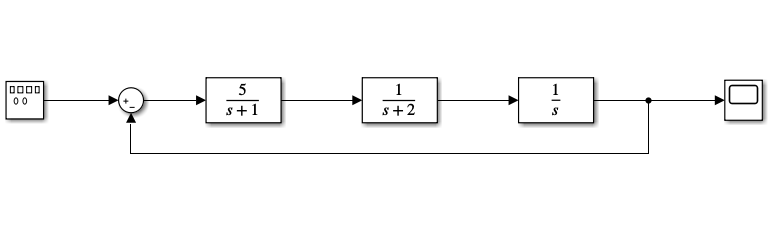
### 3.4.1 实验要求

判断如下系统的稳定性：



### 3.4.2 实验程序

1. %% 判断如下系统的稳定性
2. clc;clear
3. sys1=tf([5],[1 1]);
4. sys2=tf([1],[1 2]);
5. sys3=tf([1],[1 0]);
6. sys4=tf([1],[1]);
7. sysa=series(sys1,sys2);
8. sysb=series(sysa,sys3);
9. sysall=feedback(sysb,sys4,-1)   %整个系统的传递函数
10. %%%%%%%%%%%%%%%%李亚普洛夫第一法，间接法%%%%%%%%%%%%%%%%%%%
11. [A,B,C,D]=tf2ss(sysall.num{1},sysall.den{1})
12. lamda=eig(A)
13. fprintf('由输出可知，矩阵A的特征值均具有负实部，所以系统是渐近稳定的。\n');
14. %%%%%%%%%②李亚普洛夫第二法，直接法，思路2,用lyap()函数%%%%%%%
15. disp('下面用李亚普洛夫第二法（直接法），思路2,用lyap()函数来解：');
16. Q=eye(size(A,1))   %返回一个 n×n 的单位矩阵（先确定能量递减）
17. P=lyap(A,Q)
18. sign=2;
19. for x=1:size(P)      %希尔维斯特判据判定P是否正定
20. detx=det(P(1:x,1:x))
21. if detx<=0
22. sign=0;
23. else
24. sign=1;
25. end
26. end
27. if sign==1      %输出结果，先给Q解出P，再判断P的正定是充要条件
28. fprintf('P的各阶顺序主子式均>0，所以矩阵P正定，该系统稳定！\n\n');
29. else
30. disp('判断出错！');
31. end
32. %%%%%%%③李亚普洛夫第二法(直接法)，思路2,不用lyap()函数%%%%%%%
33. disp('下面用李亚普洛夫第二法（直接法），思路2,不用lyap()函数来解：');
34. P=eye(size(A,1))%先选择一个正定的P
35. Q=-P\*A-A'\*P    %然后由李亚普洛夫方程算出Q
36. sign2=2;
37. for x=1:size(Q)
38. detxx=det(Q(1:x,1:x))
39. if detxx<=0
40. sign2=0;
41. end
42. end
43. %\*\*注意：先给P解出Q，再判断Q的正定是充分条件，若不正定则不能得出否定的结果！！！\*\*%
44. if sign2==2
45. fprintf('各阶顺序主子式均>0，所以矩阵Q正定，该系统稳定！\n\n');
46. else
47. fprintf('Q的各阶顺序主子式中有≤0的，所以矩阵Q不正定，无法判断系统是否稳定！\n');
48. end



### 3.4.3 实验结果

sysall =

5

---------------------

s^3 + 3 s^2 + 2 s + 5

Continuous-time transfer function.

A =

-3 -2 -5

1 0 0

0 1 0

B =

1

0

0

C =

0 0 5

D =

0

lamda =

-2.9042 + 0.0000i

-0.0479 + 1.3112i

-0.0479 - 1.3112i

由输出可知，矩阵A的特征值均具有负实部，所以系统是渐近稳定的。

下面用李亚普洛夫第二法（直接法），思路2,用lyap()函数来解：

Q =

1 0 0

0 1 0

0 0 1

P =

23.0000 -0.5000 -13.5000

-0.5000 13.5000 -0.5000

-13.5000 -0.5000 8.2000

detx =

23.0000

detx =

310.2500

detx =

71.1750

P的各阶顺序主子式均>0，所以矩阵P正定，该系统稳定！

下面用李亚普洛夫第二法（直接法），思路2,不用lyap()函数来解：

P =

1 0 0

0 1 0

0 0 1

Q =

6 1 5

1 0 -1

5 -1 0

detxx =

6

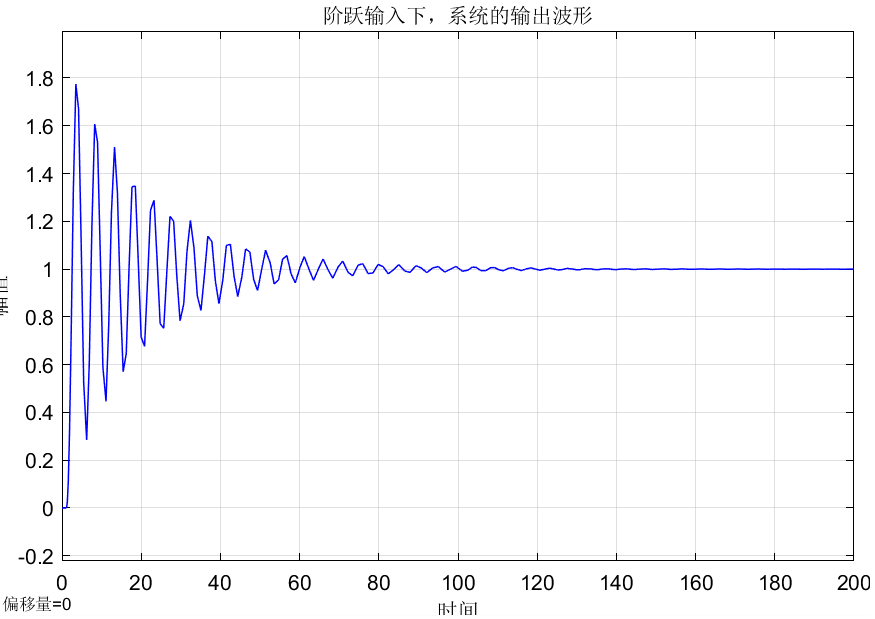
detxx =

-1

detxx =

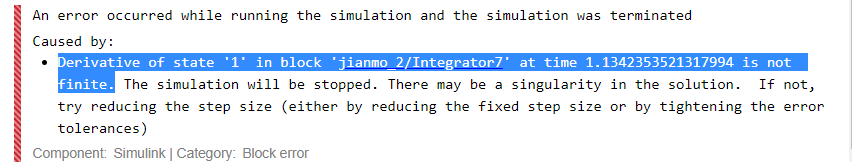
-16

Q的各阶顺序主子式中有≤0的，所以矩阵Q不正定，无法判断系统是否稳定！

****

## 3.5 小结

第一题本来在初值[-0.1，0.1，0.2]的情况下，Simulink仿真程序能顺利运行，但把初值设置为[1,2,3]之后报了一个错误：



根据提示文字可知，在时间1.134235多秒时状态1成了无限的值，也就是说系统是发散的，在这个时刻超出了Simulink的输出限幅，查阅百度可知，解决方法一般有以下几种：

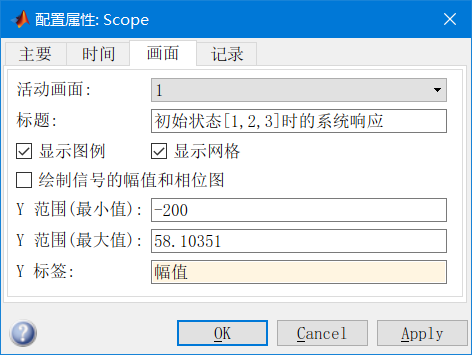
1、在simulation下拉菜单中congfiguration parameters减小步长

2、给积分模块赋予初值，取一个极小数（如0.0001），不影响全局

3、如果在仿真模块中含有控制器，可以在控制器的输出值后面增加一个限幅环节，这样可防止出现刚开始仿真时就出现过冲的现象。

不过我在这里采取了最简单的一种——减小仿真运行时间，把“Stop Time”设置为1.133（小于错误信息中报告的时间），这样就不会报错了。

还是第一题，我发现simulink的波形显示图窗没有标题、也不是像matlab脚本那样写命令来设置图窗的各种细节，一番摸索之后我在图窗上沿选项框“视图”—“配置属性”/“样式”中到了进行相关设置的地方，自行设置了标题、背景色、曲线颜色、坐标轴范围、坐标轴标签等，提升了仿真的效果。



题目三中，在编程区我用了三种方法来判断系统稳定性，加深了我对李雅普诺夫方法，尤其是李雅普诺夫第二法的理解和认识，课堂上讲的都是先给出正定的Q，再解李雅普诺夫方程得到P，然后判断P的正定性；在这里我知道了原来还可以先给出正定的P，然后直接由李雅普诺夫方程一个等式就算出Q，再判断Q的正定性，不过这种方法只是充分条件！如果Q不正定则还要用其他方法辅助判断，可能不能一次就得出结果，我想也正是因为它的这个不足，所以理论课上老师为了节约时间就没讲这种方法。所以后续我又通过Simulink搭框图观察输出波形来验证了结果，在阶跃输入下，系统响应收敛于0，说明这个系统确实是稳定的。

# 第四章 线性系统设计部分实验

## 4.1 实验目标

1.了解系统设计一般流程

2.典型综合问题：极点配置、镇定设计、状态观测器设计、基于状态观测器的状态反馈设计

3.常用控制器增益计算方法：LMI、极点配置

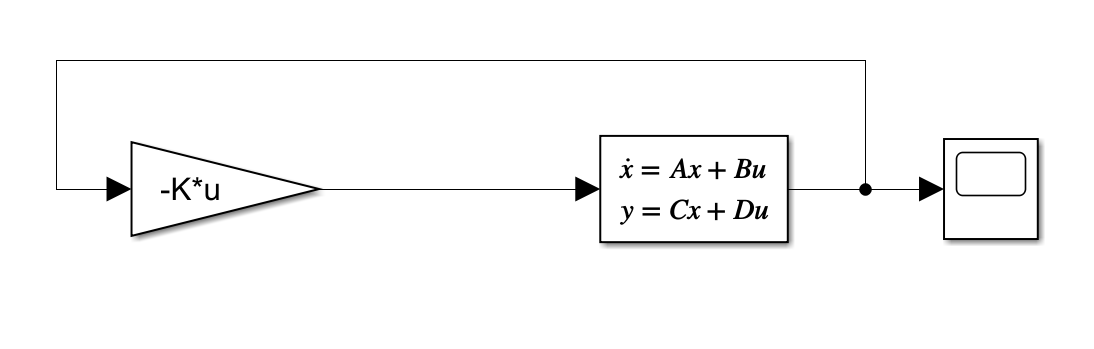
## 4.2 实验内容1（作业一第一部分）

### 4.2.1 实验要求

对如下系统，设计状态反馈控制器，极点P1（合理条件下任意选），验证状态反馈控制效果。

### 4.2.2 实验程序

1. %% ①设计状态反馈控制器，极点P1,验证状态反馈控制效果
2. clc;clear
3. A=[-1 0 0 0;
4. 2 -3 0 0;
5. 0 0 2 0;
6. 4 -1 2 -4;]
7. B=[0;0;1;2]
8. C=[3 0 1 0]
9. D=zeros(1,1)
10. Init=[-0.1 0.1 0.2 -0.3];%初始条件
11. lamda\_A=eig(A);
12. fprintf('原系统特征值为：lamda\_A=');disp(lamda\_A');
13. fprintf('系统矩阵特征根中含有正实部，故系统不能渐近稳定。\n');
14. n=size(A,1);
15. M=ctrb(A,B)%能控性矩阵
16. N=obsv(A,C)%能观性矩阵
17. fprintf('系统维数n=');disp(size(A,1));
18. fprintf('rank(M)=');disp(rank(M));
19. fprintf('rank(N)=');disp(rank(N));
20. if (rank(M)==n)&&(rank(N)==n)
21. fprintf('所以系统能控能观\n');
22. else if rank(M)==n
23. fprintf('所以系统能控不能观\n');
24. else if rank(N)==n
25. fprintf('所以系统不能控能观\n');
26. else
27. fprintf('所以系统不能控不能观\n');
28. end
29. end
30. end
31. % %%----①设计状态反馈控制器，极点P1➢验证状态反馈控制效果------
32. [A1,B1,C1,T,k] = ctrbf(A,B,C);   %对系统按能控性分解，t为变换阵，k为各子系统的秩;
33. Anc=A1(1:2,1:2); %提取不能控子系统
34. %判断不能控子系统是否渐进稳定
35. Q=eye(size(Anc,1));
36. P=lyap(Anc,Q);
37. det1=det(P(1,1));
38. det2=det(P(1:2,1:2));
39. Det=[det1;det2];
40. if min(Det)>0
41. fprintf('不能控子系统渐进稳定，系统状态反馈能镇定\n\n');
42. else
43. fprintf('不能控子系统不稳定');
44. end
45. %对能控子系统进行极点配置
46. %matlab内嵌函数place（无重根）、acker（有重根)
47. Ac=A1(3:4,3:4);%提取能控子系统
48. Bc=B1(3:4,1:1);
49. Cc=C1(1:1,3:4);
50. P1=[-11 -10];   %能控子系统的期望极点P1\_（二维）
51. fprintf('给能控子系统配置的极点是：');disp(P1);
52. K\_=place(Ac,Bc,P1);%matlab里的K=教材里的-K
53. K=[0 0 K\_]\*T;
54. fprintf('状态反馈后的系统为( (A-BK),B,C )  \n');
55. fprintf('A-B\*K = \n');disp((A-B\*K)');
56. lamda\_A\_BK=eig(A-B\*K);
57. fprintf('原本的系统特征值为');lamda\_A=eig(A)
58. fprintf('状态反馈之后的系统特征值为lamda\_(A-BK)=\n');disp(lamda\_A\_BK);
59. fprintf('极点配置成功，系统渐近稳定，验证通过。\n');



### 4.2.3 实验结果

原系统特征值为：lamda\_A= -4 -3 -1 2

系统矩阵特征根中含有正实部，故系统不能渐近稳定。

M =

0 0 0 0

0 0 0 0

1 2 4 8

2 -6 28 -104

N =

3 0 1 0

-3 0 2 0

3 0 4 0

-3 0 8 0

系统维数n= 4

rank(M)= 2

rank(N)= 2

所以系统不能控不能观

不能控子系统渐进稳定，系统状态反馈能镇定

给能控子系统配置的极点是： -11 -10

状态反馈后的系统为( (A-BK),B,C )

A-B\*K =

-1.0000 2.0000 -0.0000 4.0000

0 -3.0000 0 -1.0000

0 0 -25.4000 -52.8000

0 0 4.2000 4.4000

原本的系统特征值为

lamda\_A =

-4

-3

-1

2

状态反馈之后的系统特征值为lamda\_(A-BK)=

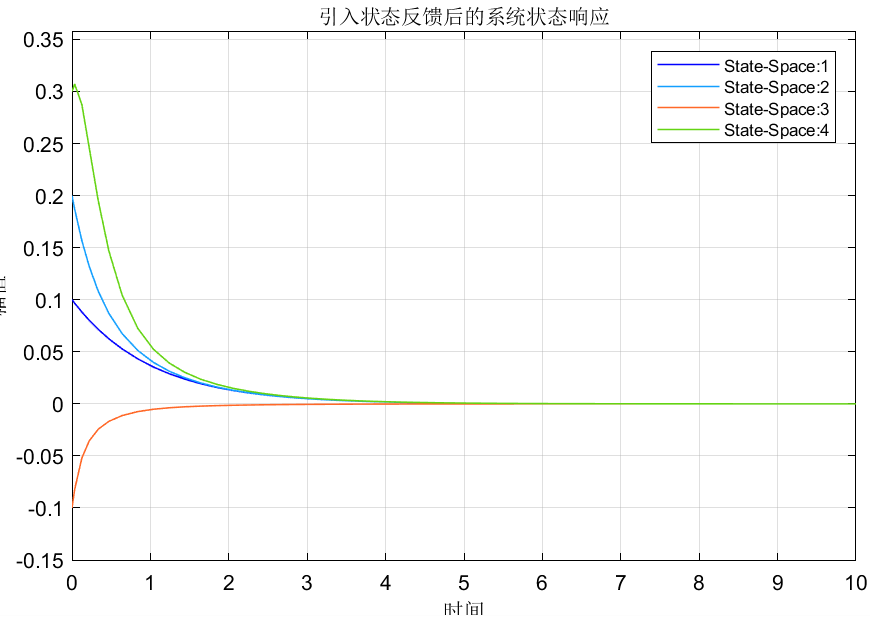
-10.0000

-11.0000

-3.0000

-1.0000

极点配置成功，系统渐近稳定，验证通过。



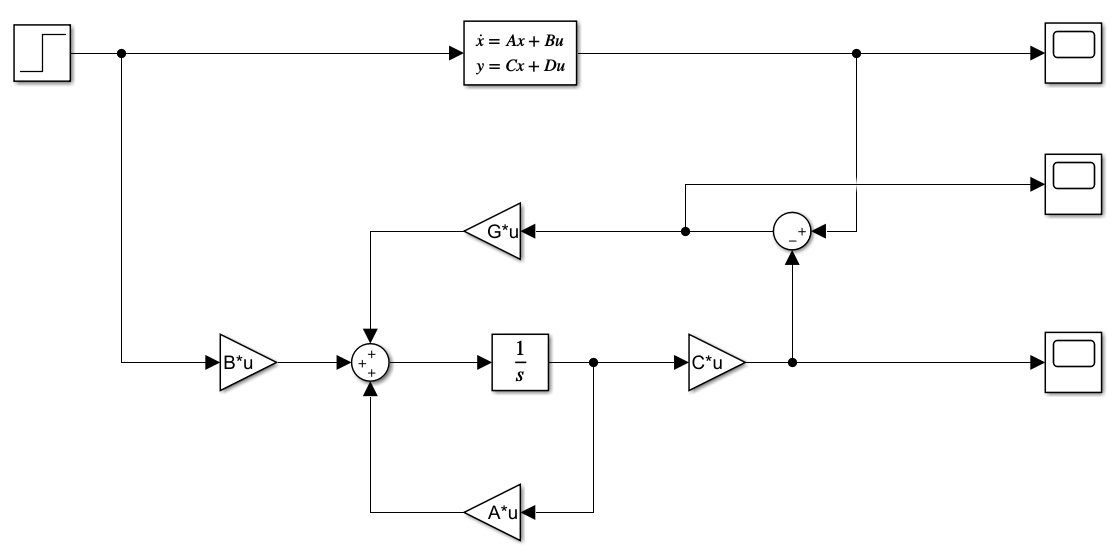
## 4.3 实验内容2（作业一第二部分）

### 4.3.1 实验要求

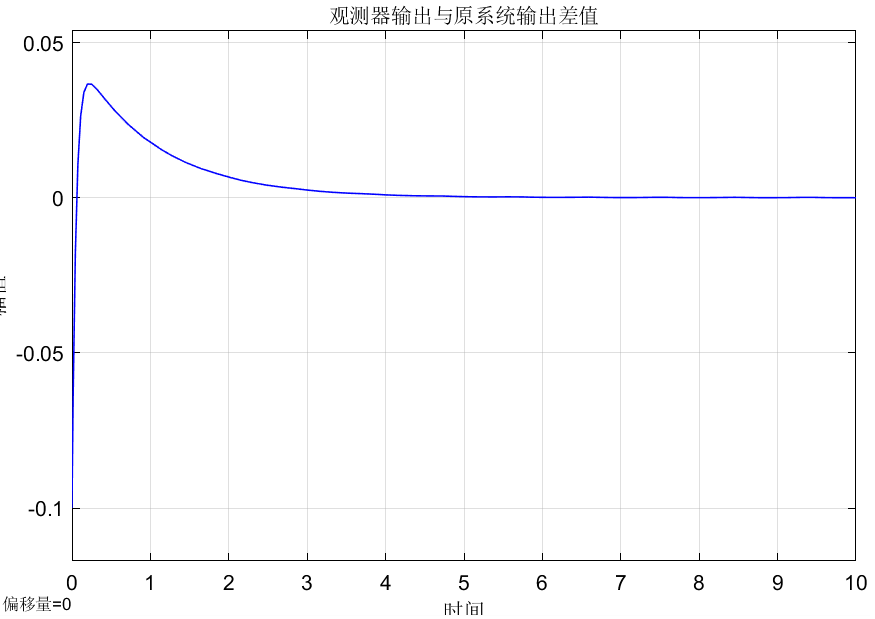
对如下系统，设计状态观测器，极点P2（合理条件下任意选），验证状态观测器效果。

### 4.3.2 实验程序

1. %%-②设计状态观测器，极点P2，验证状态观测器效果-
2. clc;clear
3. A=[-1 0 0 0;
4. 2 -3 0 0;
5. 0 0 2 0;
6. 4 -1 2 -4;]
7. B=[0;0;1;2]
8. C=[3 0 1 0]
9. D=zeros(1,1)
10. Init=[0.1 0.2 0.3 0.44];%初始条件
11. lamda\_A=eig(A);
12. fprintf('原系统特征值为：lamda\_A=');disp(lamda\_A');
13. fprintf('系统矩阵特征根中含有正实部，故系统不能渐近稳定。\n');
14. n=size(A,1);
15. M=ctrb(A,B)%能控性矩阵
16. N=obsv(A,C)%能观性矩阵
17. fprintf('上一问中已求出系统不能控不能观');
18. fprintf('因为原系统不完全能观，所以其不能观子系统必须渐近稳定才能设计观测器,下面进行判断：\n');
19. [A1,B1,C1,T,k] = obsvf(A,B,C);%能观性分解
20. Ano=A1(1:2,1:2);%提取不能观子系统
21. Q=eye(size(Ano,1));%判定不能观子系统是否稳定性
22. P=lyap(Ano,Q);
23. det1=det(P(1,1));
24. det2=det(P(1:2,1:2));
25. Det=[det1;det2];
26. if min(Det)>0
27. fprintf('不能观子系统渐进稳定，所以系统的状态观测器存在。');
28. else
29. fprintf('不能观子系统不稳定');
30. end
31. %对能观子系统进行极点配置
32. Ao=A1(3:4,3:4)%提取能观子系统
33. Bo=B1(3:4,1:1);
34. Co=C1(1:1,3:4);
35. P2\_=[-6;-6.6];%观测器能观部分的期望极点
36. K=place(Ao',Co',P2\_);
37. G\_=K';
38. fprintf('状态观测器的状态方程为：~x''=(A-GC)~x+Gy+Bu  \n');
39. G=T\*[0;0;G\_] %将反馈矩阵进行还原
40. lamda\_Ao\_GC=eig(Ao-G\_\*Co)
41. lamda\_A\_GC=eig(A-G\*C)
42. fprintf('说明极点配置成功，且观测器渐近稳定，验证通过。\n');



### 4.3.3 实验结果



## 4.4 实验内容3（作业一第三部分）

### 4.4.1 实验要求

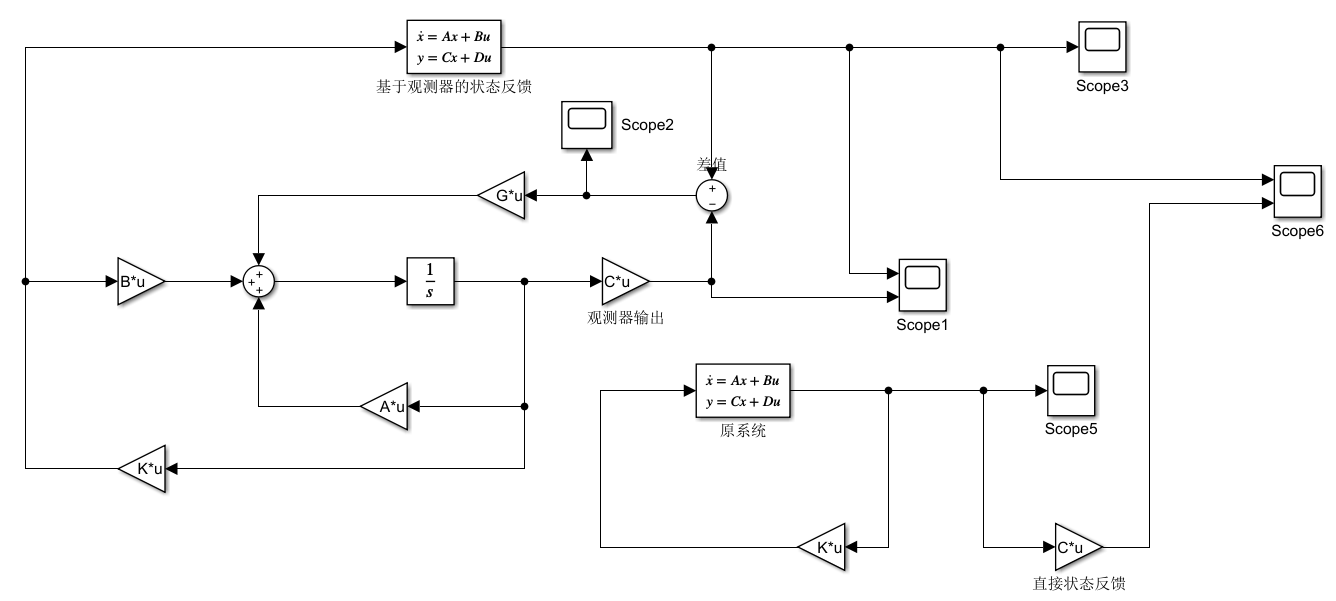
1.对如下系统，设计基于状态观测器的状态反馈，极点P1（合理条件下任选）

2.验证基于状态观测器的状态反馈控制效果。

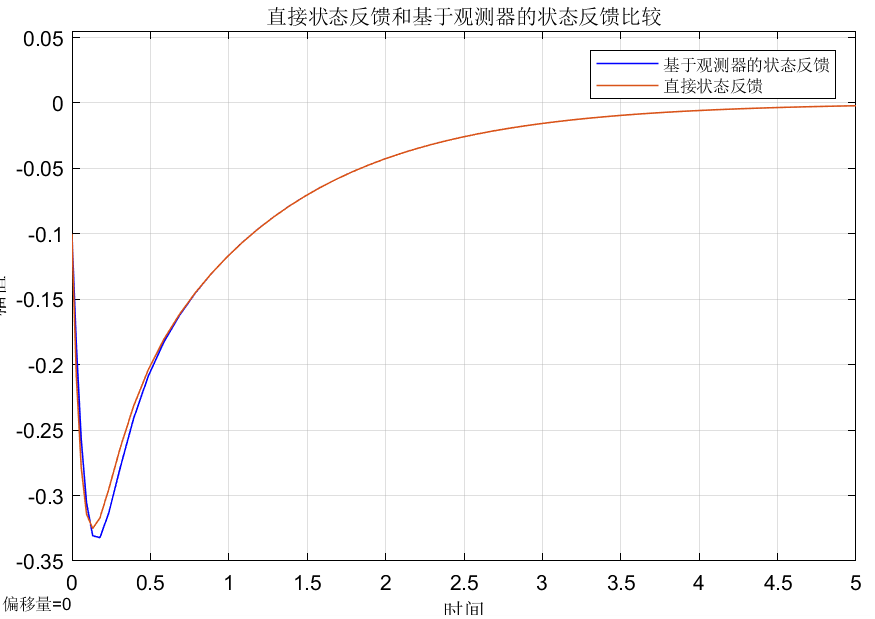
3.比较上述两类状态反馈的效果

### 4.4.2 实验程序

1. %% --③设计基于状态观测器的状态反馈，验证基于状态观测器的状态反馈控制效果--
2. clc;clear
3. A=[-1 0 0 0;
4. 2 -3 0 0;
5. 0 0 2 0;
6. 4 -1 2 -4;]
7. B=[0;0;1;2]
8. C=[3 0 1 0];D=[0]
9. Init=[-0.1 0.1 0.2 -0.05];%初始条件
10. lamda\_A=eig(A);
11. fprintf('原系统特征值为：lamda\_A=');disp(lamda\_A');
12. fprintf('系统矩阵特征根中含有正实部，故系统不能渐近稳定。\n');
13. n=size(A,1);
14. M=ctrb(A,B)%能控性矩阵
15. N=obsv(A,C)%能观性矩阵
16. fprintf('系统维数n=');disp(size(A,1));
17. fprintf('rank(M)=');disp(rank(M));
18. fprintf('rank(N)=');disp(rank(N));
19. if (rank(M)==n)&&(rank(N)==n)
20. fprintf('所以系统能控能观\n');
21. else if rank(M)==n
22. fprintf('所以系统能控不能观\n');
23. else if rank(N)==n
24. fprintf('所以系统不能控能观\n');
25. else
26. fprintf('所以系统不能控不能观\n');
27. end
28. end
29. end
30. fprintf('但根据分离原理，我们可以对能控能观子系统分别设计K和G\n');
31. %设计状态反馈增益K
32. [A1,B1,C1,T,k] = ctrbf(A,B,C);%能控性分解
33. Anc=A1(1:2,1:2); %提取不能控子系统
34. Q=eye(size(Anc,1)); %判断不能控子系统稳定性
35. P=lyap(Anc,Q);
36. det1=det(P(1,1));
37. det2=det(P(1:2,1:2));
38. Det=[det1;det2];
39. if min(Det)>0
40. fprintf('不能控子系统渐进稳定，系统状态反馈能镇定\n\n');
41. else
42. fprintf('不能控子系统不稳定');
43. end
44. Ac=A1(3:4,3:4); %对能控子系统进行极点配置
45. Bc=B1(3:4,1:1);
46. Cc=C1(1:1,3:4);
47. P1=[-11 -10];   %能控子系统的期望极点P1\_
48. fprintf('给能控子系统配的极点是：');disp(P1);
49. K\_=place(Ac,Bc,P1);
50. K=[0 0 K\_]\*T;
51. %设计状态观测器增益矩阵G
52. [A1,B1,C1,T,k] = obsvf(A,B,C);%能观性分解
53. Ano=A1(1:2,1:2);%提取不能观子系统
54. Q=eye(size(Ano,1));%判定不能观子系统稳定性
55. P=lyap(Ano,Q);
56. det1=det(P(1,1));
57. det2=det(P(1:2,1:2));
58. Det=[det1;det2];
59. if min(Det)>0
60. fprintf('不能观子系统渐进稳定，所以系统的状态观测器存在。');
61. else
62. fprintf('不能观子系统不稳定');
63. end
64. %对能观子系统进行极点配置
65. Ao=A1(3:4,3:4)%提取能观子系统
66. Bo=B1(3:4,1:1);
67. Co=C1(1:1,3:4);
68. P2\_=[-6;-6.6];%观测器能观部分的期望极点
69. G0=place(Ao',Co',P2\_);
70. G\_=G0';
71. fprintf('状态观测器的状态方程为：~x''=(A-GC)~x+Gy+Bu  \n');
72. G=T\*[0;0;G\_] %将反馈矩阵进行还原



### 4.4.3 实验结果



## 4.5 小结

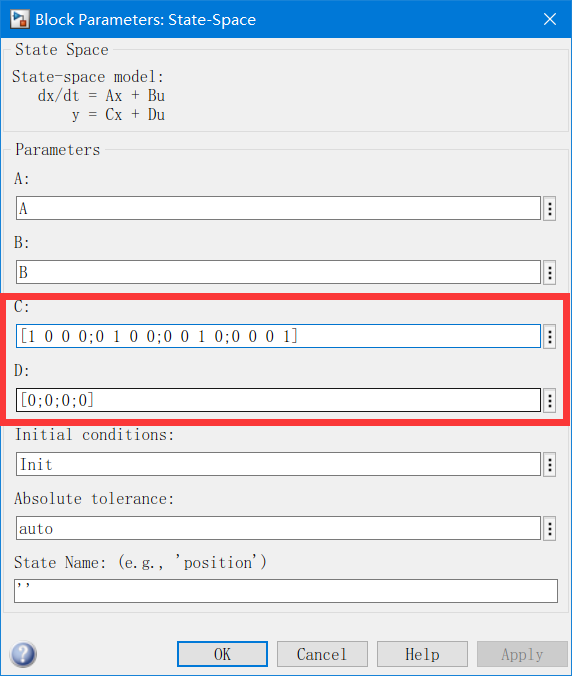
Simulink仿真有几个要特别注意的地方：

①环节和环节的维数匹配问题

②设置初值

③增益的正负号

④输出的是状态还是输出量，如果是要直接进行状态反馈，那就要输出状态，这时C要设置为n阶单位阵，D要设置为n行1列的0矩阵。



另外发现，观测器初值和系统初值相差较大时，基于观测器的状态反馈系统输出和直接的状态反馈系统输出可能不一定在横轴同一侧，而是一个往上偏，一个往下偏，最终都趋于同一稳定状态。

# 第五章 实验总结与体会

## 5.1 实验总结

通过此次实验，我学到了如何用maltab（结合simulink）解决线性系统分析与设计中的问题，从定性分析（稳定性、能控性、能观性）到定量分析（求状态空间表达式的解、坐标变换、结构分解）再到综合设计（极点配置、镇定设计、观测器设计、利用状态观测器设计状态反馈），可以看出现代控制理论的这些内容是一脉相承、一环扣一环的，其实都指向同一个最终目标——设计控制系统，从而满足各种工业生产环节的应用需求。

本次的实验与以往最大的不同就是matlab脚本编程与simulink的配合使用，工作空间（Workspace）是二者进行数据交换的桥梁：脚本先运行，在工作空间中产生数据对象，然后在simnlink中可以调用，比如设置状态空间表达式模型的4个矩阵；也可以先在simulink中运行框图，产生的数据存到工作空间后就可以在脚本编程时调用。在使用时要注意二者的这种联系，很多出错也是因它而起，比如A\B\C\D矩阵的值设置不对、维数不对等，所以当simulink仿真报错的时候，就要重点检查用于联系二者的数据对象。

## 5.2 实验体会

纸上得来终觉浅，单纯的理论课学习显然是不够的，尽管在课堂上对于线性系统的分析设计方法有了基本的理解，但是在面临具体的问题时还是难免力不从心，就是那种脑袋会了但手还不会的情况。matlab仿真特别直观、非常形象，借助这一工具能很好地加快对知识的吸收、加深理解，对提升学习效果很有帮助。

想到一点建议：相比结课后一次把实验做完，一边上理论课一边做实验可能效果会更好一点，每一章的作业改为由课后习题和matlab仿真两部分构成，习题巩固理论基础，作为matlab仿真的前提；matlab实操具体又形象，反过来又加深了对理论的理解，两者相互促进、相得益彰。

总之，虽然由于个人编程水平的限制以及对知识的掌握不够牢固，在实验过程中遇到了许许多多、各种各样的困难，最终结果也有一些不足之处，但值得庆幸的是我并没有放弃，而是通过努力查资料、看视频、问同学问老师然后自己钻研一一解决了问题，我希望能把事情尽力做到最好！这也是我一直信奉的处世态度。努力编写自己的程序，学到的都是自己的，经过本次实验，我对学好控制理论更有信心了。

最后，对老师超级热情又超级耐心的答疑和同学们的热心帮助表示衷心的感谢，是各位的帮助使我更顺利地完成了本次实验！