

中国地质大学

自动控制原理II：线性系统分析与设计

实验报告

**学 院：**

**课 程：**

**指导老师：**

**学 号：**

**班 级：**

**姓 名：**

**二○二一年十一月二十二日**

**目录**

[第一章 简介部分 3](#_Toc31942)

[1.1实验目标 3](#_Toc19426)

[1.2实验内容 3](#_Toc27568)

[1.3实验平台 3](#_Toc13140)

[第二章 系统模型部分实验 3](#_Toc15343)

[2.1实验目标 3](#_Toc17784)

[2.2实验内容1 （作业一第一题） 3](#_Toc14594)

[2.2.1 实验要求 3](#_Toc32611)

[2.2.2 实验程序 4](#_Toc5362)

[2.2.3 实验结果 5](#_Toc30494)

[2.3 实验内容2 （作业一第二题） 6](#_Toc4417)

[2.3.1 实验要求 6](#_Toc17749)

[2.3.2 实验程序 7](#_Toc5201)

[2.3.3 实验结果 8](#_Toc19875)

[2.4 小结 9](#_Toc14095)

[第三章 系统分析部分实验 9](#_Toc25106)

[3.1 实验目标 9](#_Toc29925)

[3.2 实验内容1 （作业一第一题） 10](#_Toc25544)

[3.2.1 实验要求 10](#_Toc14396)

[3.2.2 实验程序 10](#_Toc20987)

[3.2.3 实验结果 11](#_Toc4120)

[3.3 实验内容2 （作业一第二题） 14](#_Toc417)

[3.3.1 实验要求 14](#_Toc32663)

[3.3.2 实验程序 14](#_Toc19315)

[3.3.3 实验结果 15](#_Toc7765)

[3.4 实验内容3 （作业一第三题） 16](#_Toc11070)

[3.4.1 实验要求 16](#_Toc14596)

[3.4.2 实验程序 16](#_Toc25246)

[3.4.3 实验结果 17](#_Toc23485)

[3.5 小结 18](#_Toc651)

[第四章系统设计部分实验 18](#_Toc15311)

[4.1 实验目标 18](#_Toc6760)

[4.2 实验内容1 18](#_Toc25884)

[4.2.1 实验要求 18](#_Toc117)

[4.2.2 实验程序 19](#_Toc28480)

[4.2.3 实验结果 21](#_Toc1738)

[4.3 小结 25](#_Toc24859)

[第五章实验总结与体会 26](#_Toc18766)

[5.1实验总结 26](#_Toc23920)

[5.2实验体会 26](#_Toc25673)

# 简介部分

## 1.1 实验目标

①掌握基于MATLAB软件的系统建模/分析/设计常用方法

②加深对线性系统分析与设计理论知识的理解与掌握

## 1.2 实验内容

①《线性系统分析与设计》理论知识、系统分析/设计/仿真一些方法

②三大部分：系统模型部分、系统分析部分、系统设计部

## 1.3 实验平台

MATLAB

# 系统模型部分实验

## 2.1 实验目标

①在MATLAB中实现线性系统各类数学模型表示的相互转化

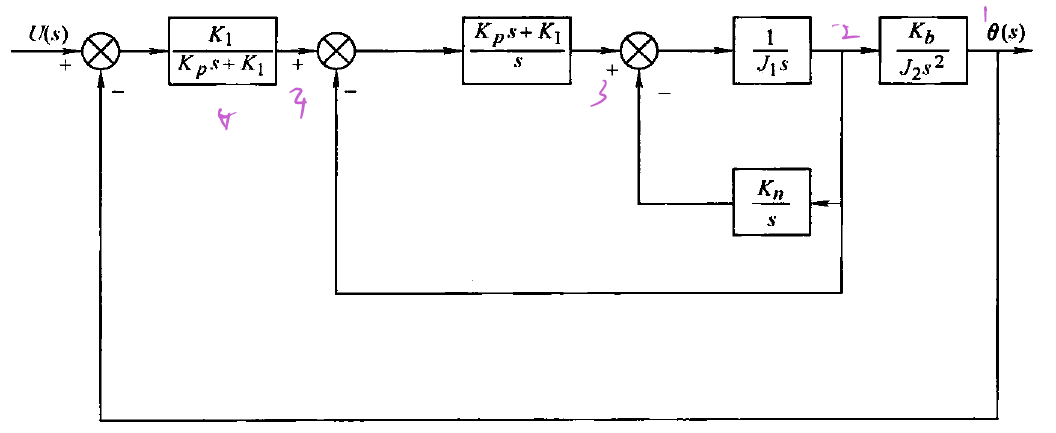
②在MATLAB中实现多个子系统的不同连接合并

## 2.2 实验内容1 （作业一第一题）

### 2.2.1 实验要求

➢ 利用MATLAB给出如下系统的 传递函数模型、状态空间模型

➢ 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

实验代码

%% 作业1

%Kb=3 Kp=4 J1=2 J2=1 K1=5 Kn=6

clc;clear;

sys1=tf([3],[1,0,0]);

sys2=tf([1],[2,0]);

sys3=tf([6],[1,0]);

sys4=tf([4,5],[1,0]);

sys5=tf([5],[4,5]);

sys6=feedback(sys2,sys3,-1);

sys7=series(sys4,sys6);

sys8=feedback(sys7,1,-1);

sys9=series(sys5,sys8);

sys10=series(sys9,sys1);

sys=feedback(sys10,1,-1)

num=[60,75,0];

den=[8,26,64,55,60,75,0];

format rat

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

simulink图

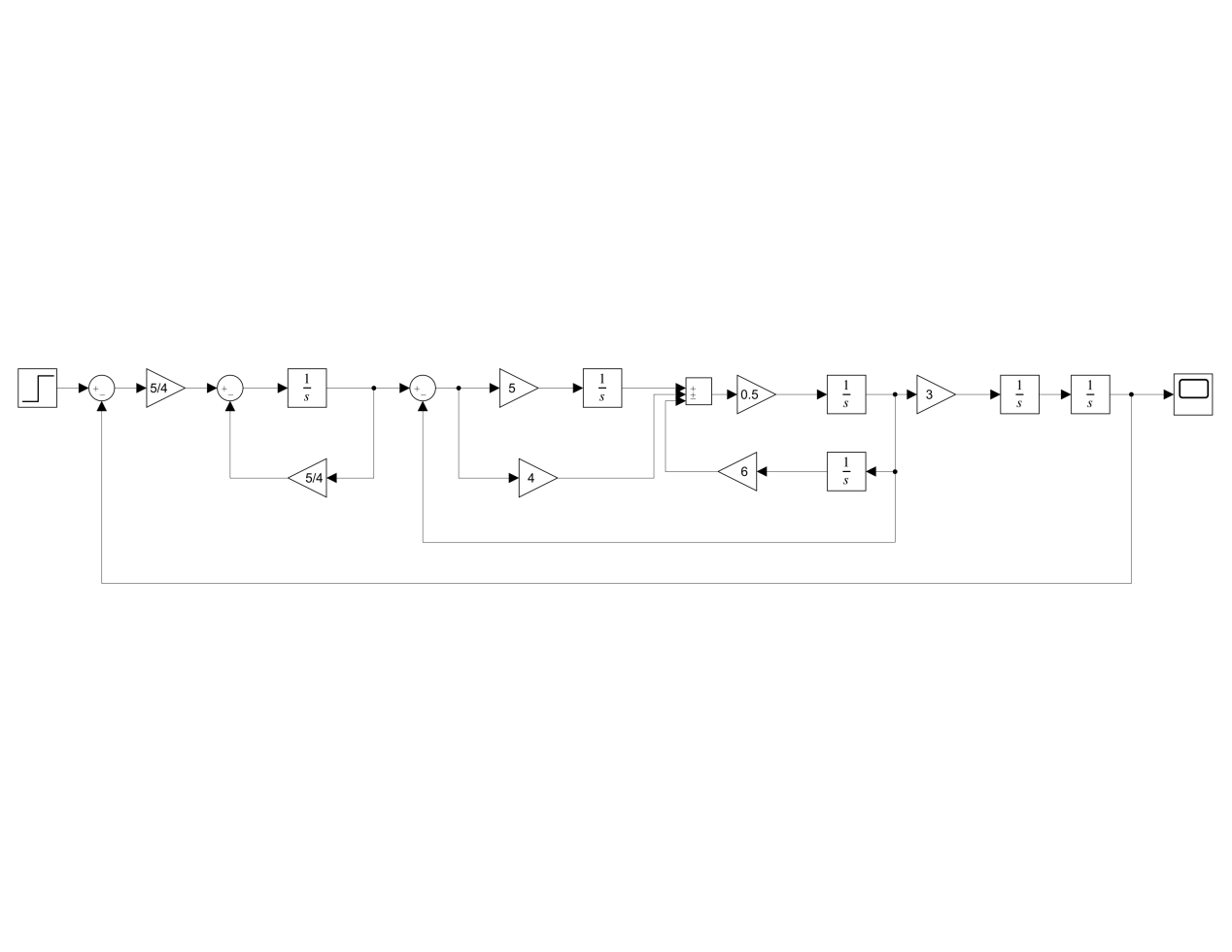


图 1传递函数模型

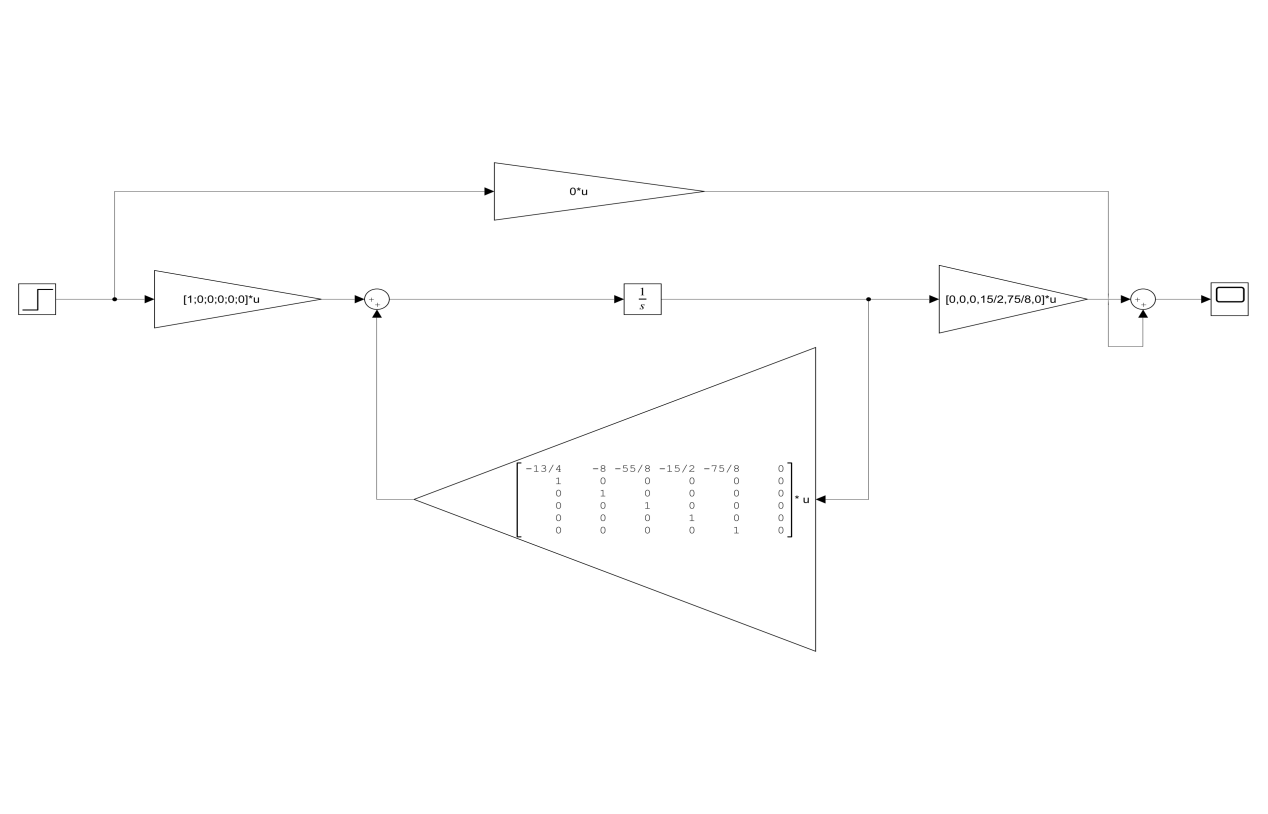


图 2结构图模型

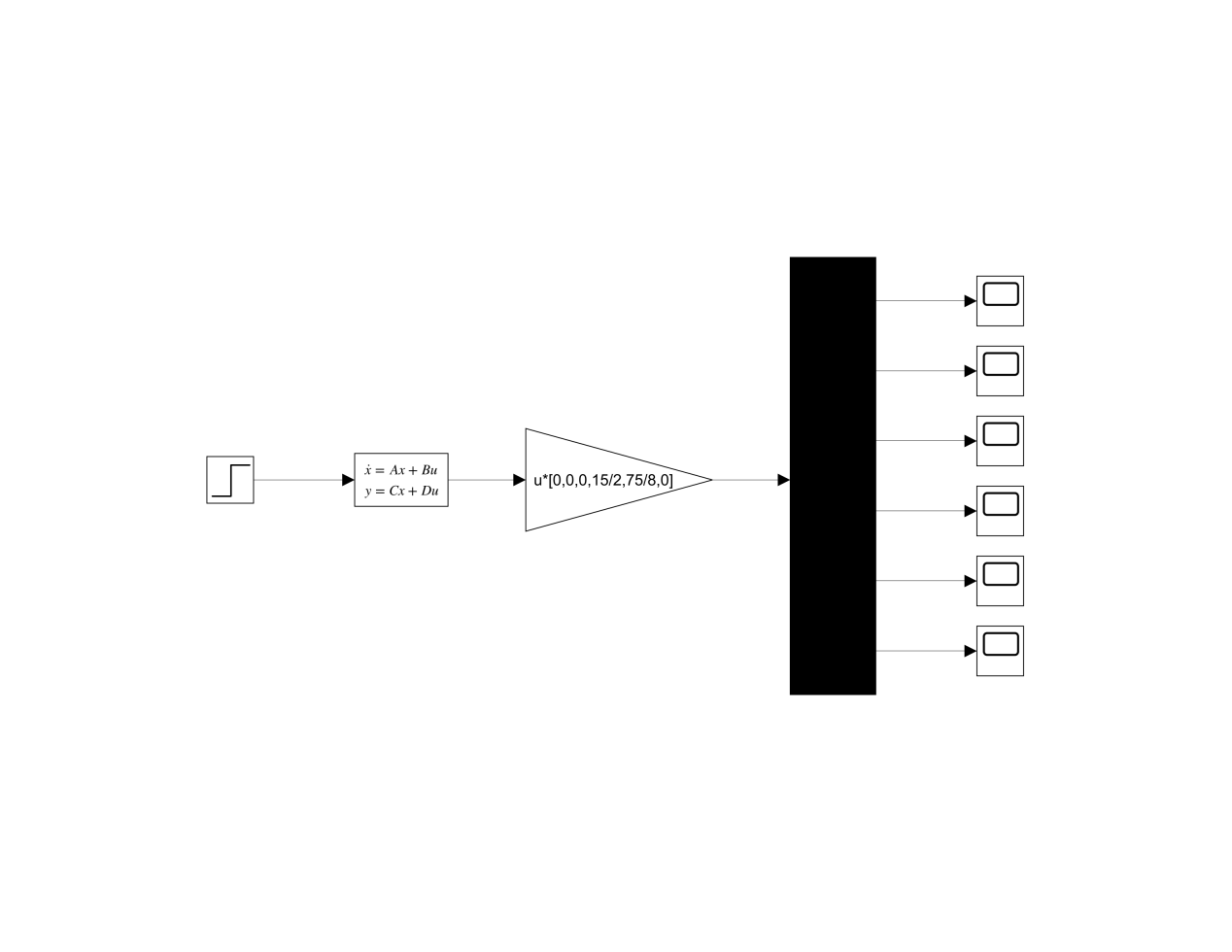


图 3状态空间模型

### 2.2.3 实验结果

程序输出结果

sys =

60 s^2 + 75 s

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

8 s^6 + 26 s^5 + 64 s^4 + 55 s^3 + 60 s^2 + 75 s

Continuous-time transfer function.

A =

-13/4 -8 -55/8 -15/2 -75/8 0

1 0 0 0 0 0

0 1 0 0 0 0

0 0 1 0 0 0

0 0 0 1 0 0

0 0 0 0 1 0

B =

1

0

0

0

0

0

C =

0 0 0 15/2 75/8 0

D =

0

simulink输出结果

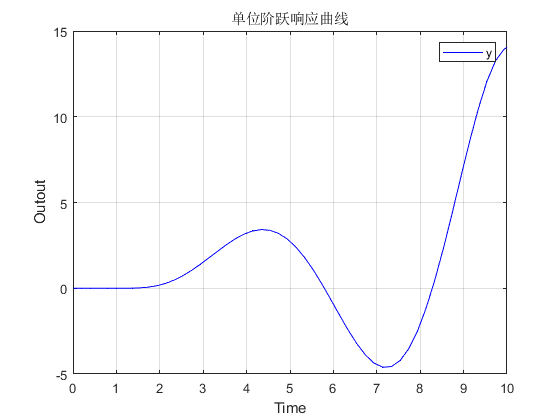


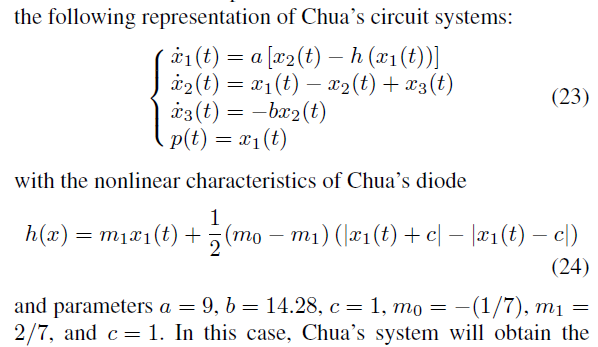
图 4 单位阶跃响应曲线图

## 2.3 实验内容2 （作业一第二题）

### 2.3.1 实验要求

➢ 利用simulink搭建如下系统结构框图

➢ 选初始条件[-0.2; 0.3; 0.7]，观察状态响应



### 2.3.2 实验程序

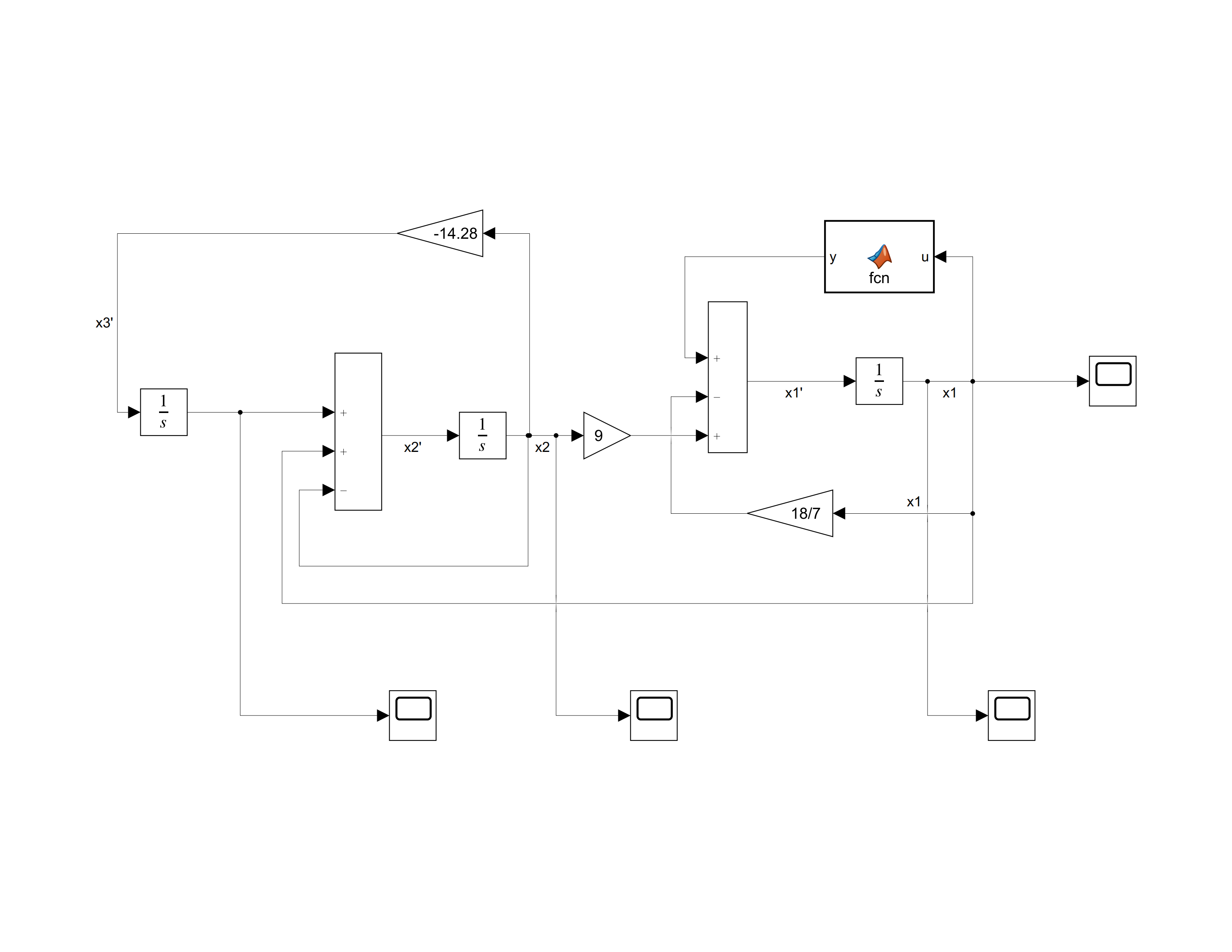


图 5 作业一第二题simulink图

### 2.3.3 实验结果

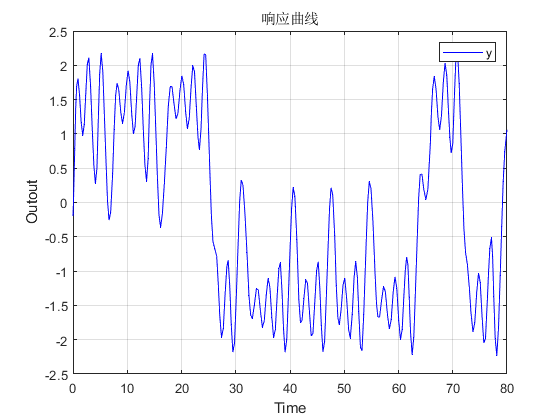


图 6 系统响应曲线图

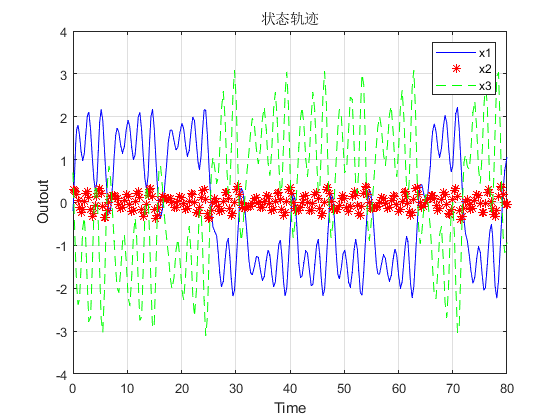


图 7 系统状态轨迹图

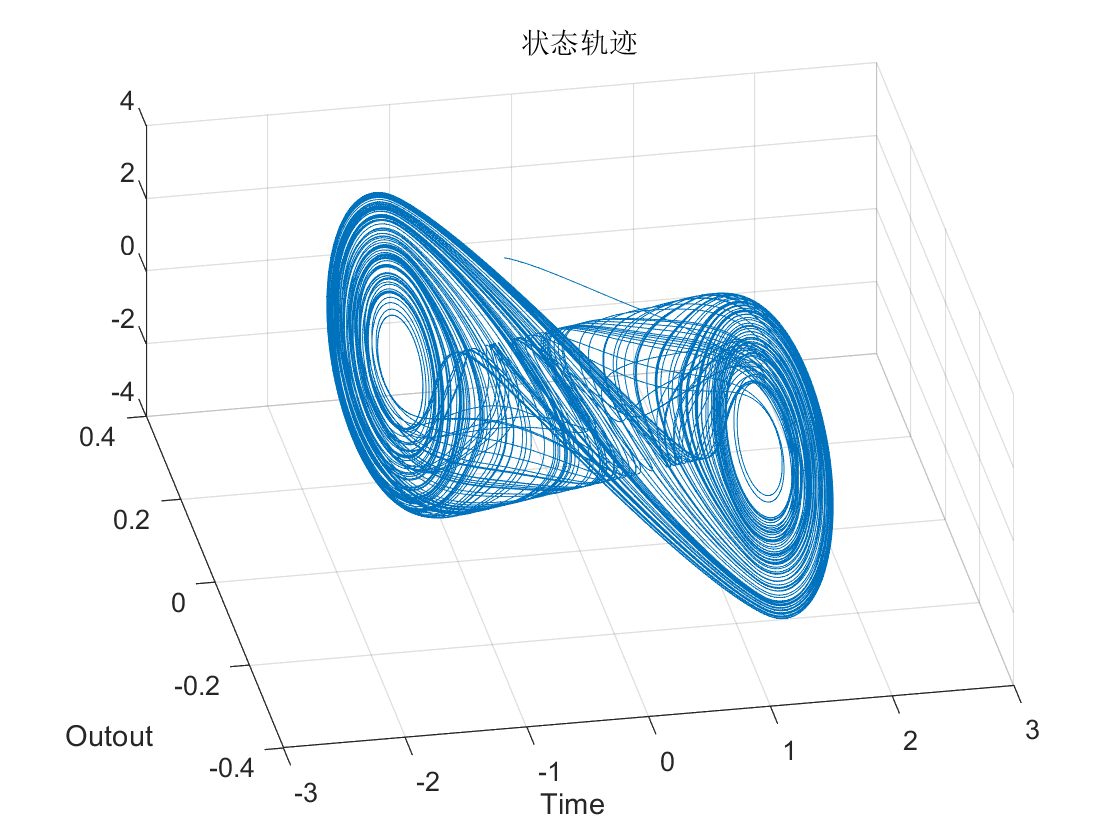


图 8 系统状态轨迹三维图

## 2.4 小结

通过该部分的实验，学习了MATLAB和SIMULINK的使用方法，完成了代码的编写以及系统结构框图的制作，学习了各类传递函数的表示方法。同时深入了解了系统结构框图的一步一步由里到外的搭建方法,加深了对一些没有接触过的代码的认识。

使用SIMULINK搭建状态空间图时，要先找到所有的状态变量的位置，在通过这些状态变量，辐射出所有的关系，这样搭建迅速不会错乱，更容易一次搭建成功。

# 第三章 系统分析部分实验

## 3.1 实验目标

①系统解的获取，即系统响应曲线的绘制

②使用MATLAB对线性定常系统能控性、能观性的判定，及其相关的坐标变换

③使用MATLAB对线性定常系统稳定性判定

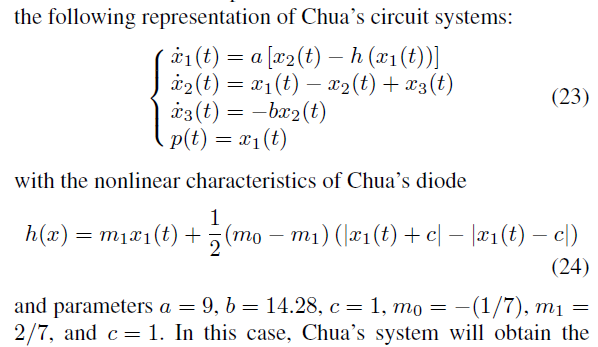
## 3.2 实验内容1（作业一第一题）

### 3.2.1 实验要求

➢ 选择两组初值[-0.1,0.1,0.2][1,2,3]

➢ 绘制如下系统的系统响应曲线

➢ 绘制如下系统的状态轨迹



### 3.2.2 实验程序

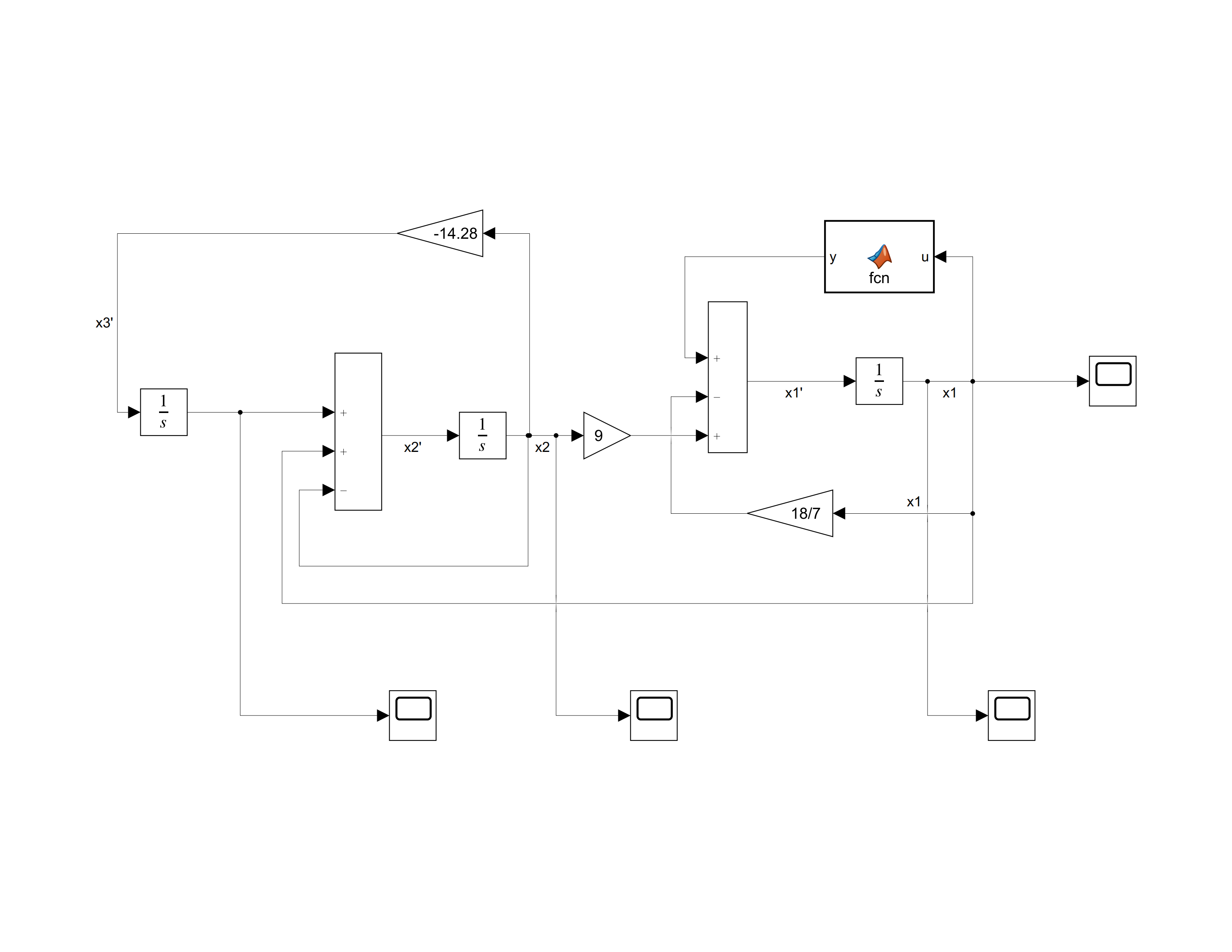


图 9 simulink图

3.2.3 实验结果

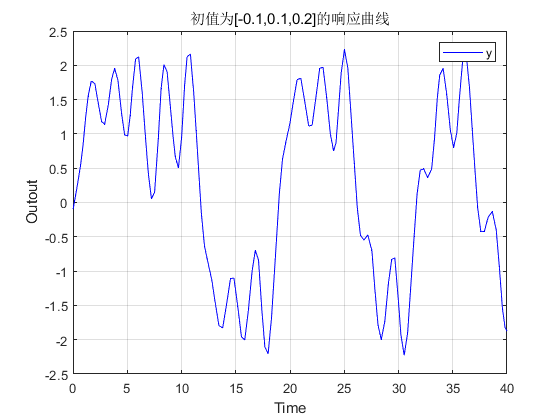


图 10 初值为[-0.1,0.1,0.2]的响应曲线

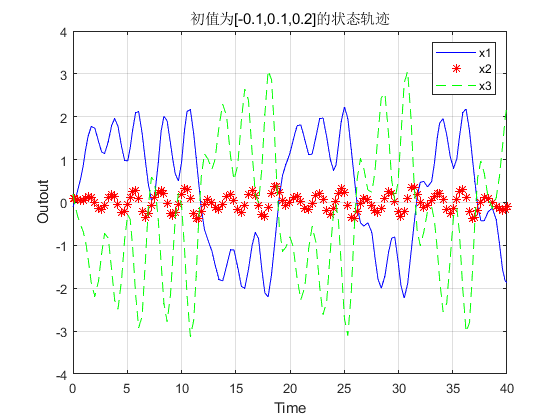


图 11 初值为[-0.1,0.1,0.2]的状态轨迹

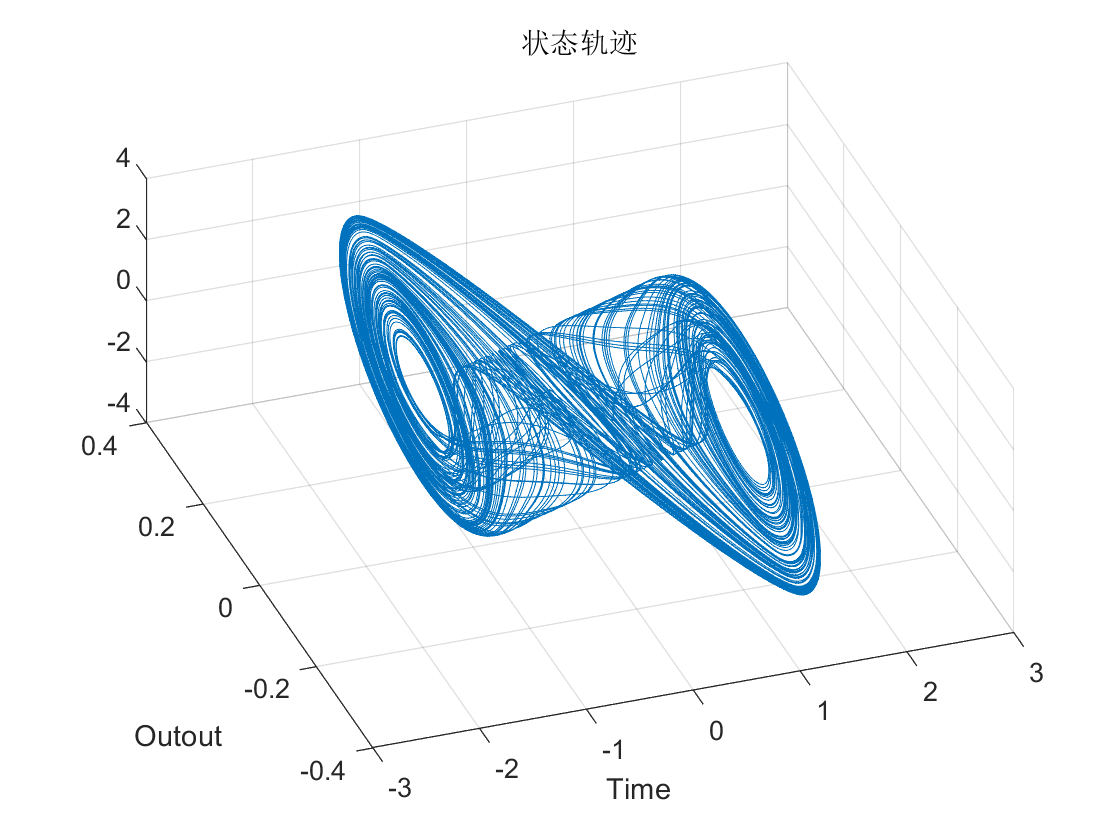


图 12 初值为[-0.1,0.1,0.2]的状态轨迹三维图

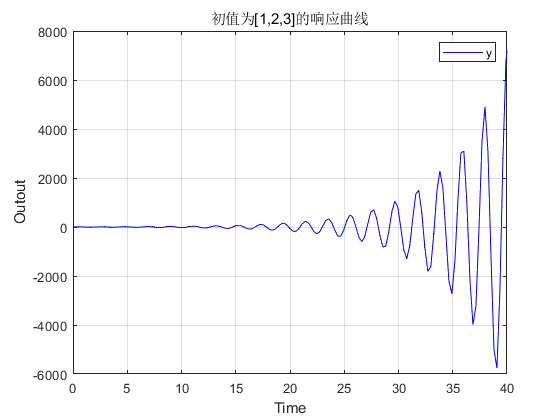


图 13 初值为[1,2,3]的响应曲线

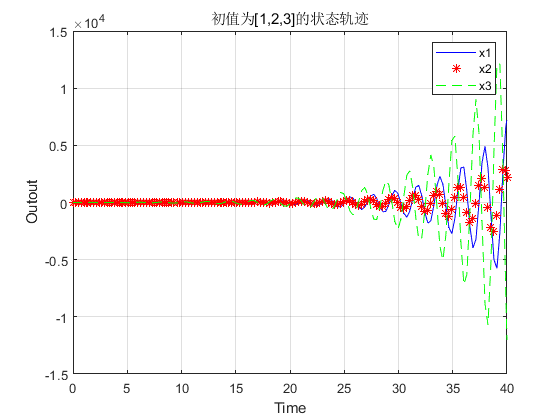


图 14 初值为[1,2,3]的状态轨迹

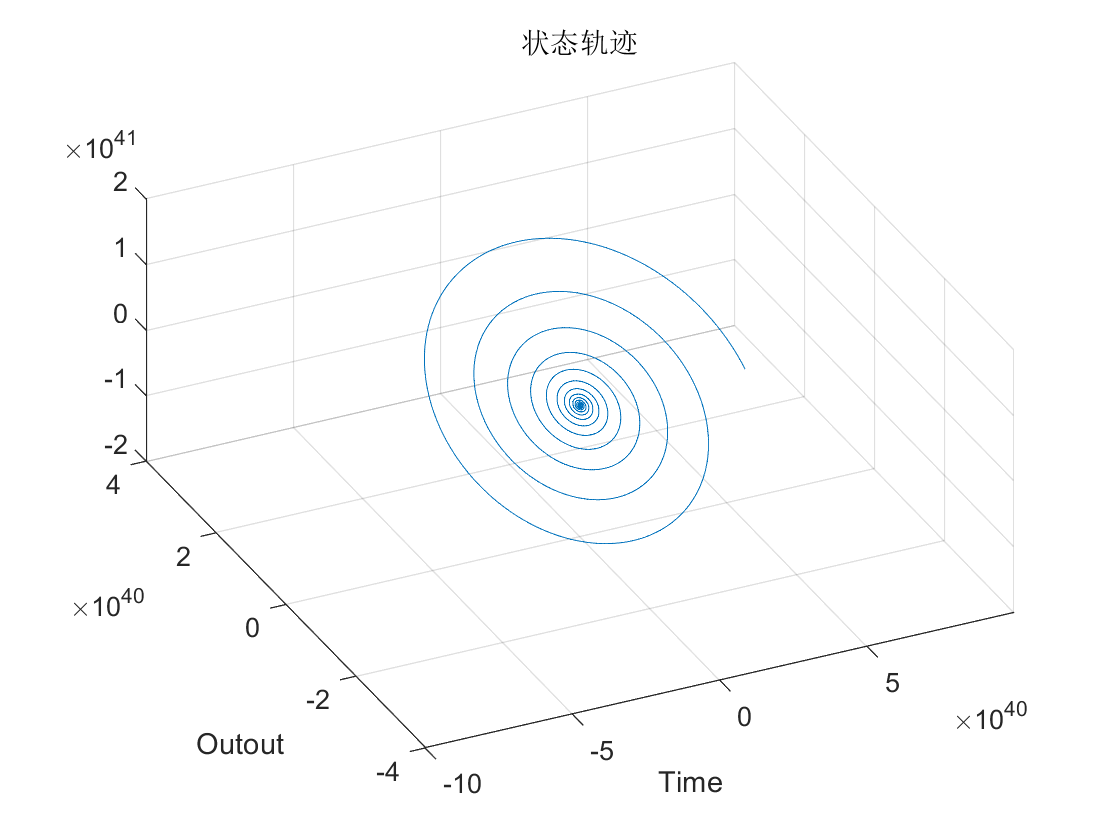
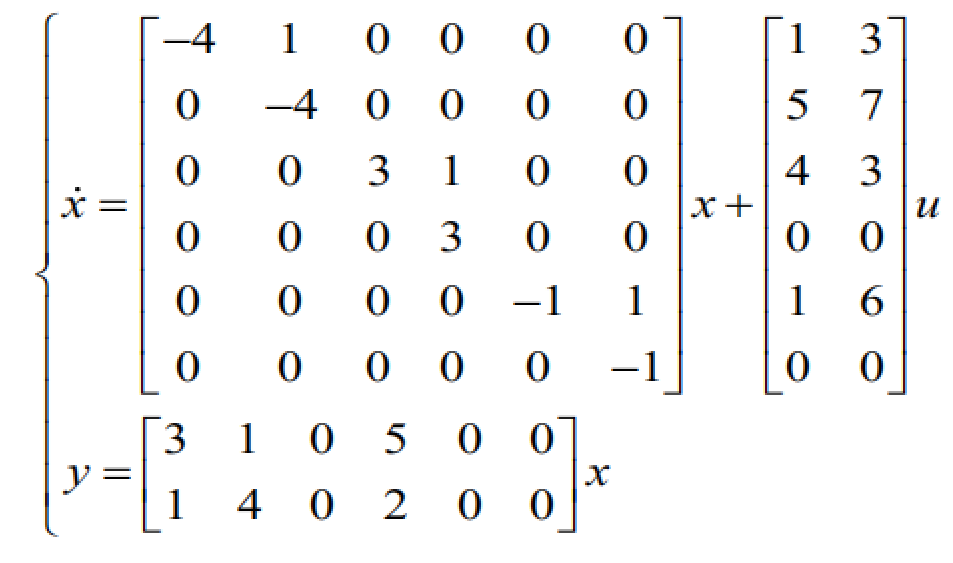


图 15 初值为[1,2,3]的状态轨迹三维图

## 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

%% 作业一

clc;clear

format long

A=[-4,1,0,0,0,0;0,-4,0,0,0,0;0,0,3,1,0,0;0,0,0,3,0,0;0,0,0,0,-1,1;0,0,0,0,0,-1];

B=[1,3;5,7;4,3;0,0;1,6;0,0];

C=[3,1,0,5,0,0;1,4,0,2,0,0];

n=size(A,1);

Qc=ctrb(A,B);

Qo=obsv(A,C);

m=rank(Qc)-1;

if rank(Qc)==n

str='系统完全能控'

else

str='系统不完全能控'

%进行能控性分解

[A1,B1,C1,T,k]=ctrbf(A,B,C)

Ac=A1(6-m:6,6-m:6)

Bc=B1(6-m:6,1:2)

Cc=C1(1:2,6-m:6)

%判断能控性子系统是否完全能观

nc=size(Ac,1);

Qcson=obsv(Ac,Cc);

mc=rank(Qcson);

%进行能观性分解

[A2,B2,C2,T2,k2]=obsvf(Ac,Bc,Cc);

Aco=A2(size(Ac,1)-mc+1:size(Ac,1),size(Ac,1)-mc+1:size(Ac,2));

Bco=B2(size(Bc,1)-mc+1:size(Bc,1),1:size(Bc,2));

Cco=C2(1:size(Cc,1),size(Cc,2)-mc+1:size(Cc,2));

end

format long

%对最终的结果进行判断

no=size(Aco,1);

Qco=ctrb(Aco,Bco);

Qoc=obsv(Aco,Cco);

if rank(Qco)==no

if rank(Qoc)==no

str='系统能控能观'

Aco

Bco

Cco

else

str='系统能控不能观'

end

else

str='系统不能控'

end

### 3.3.3 实验结果

Aco =

-4.447213595499962 0.723606797749979

-0.276393202250022 -3.552786404500047

Bco =

-1.778004752243629 -1.128165359777817

4.778985153879336 7.531748994821685

Cco =

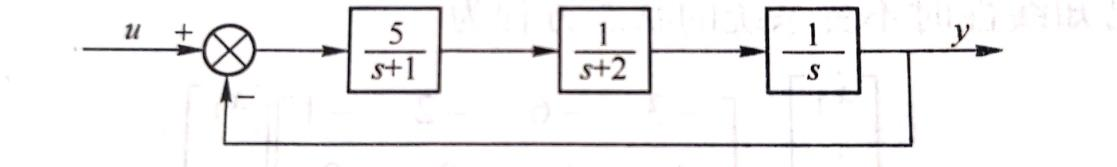
2.026221312936986 2.427844144709443

-1.252273640124496 3.928334345527295

## 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

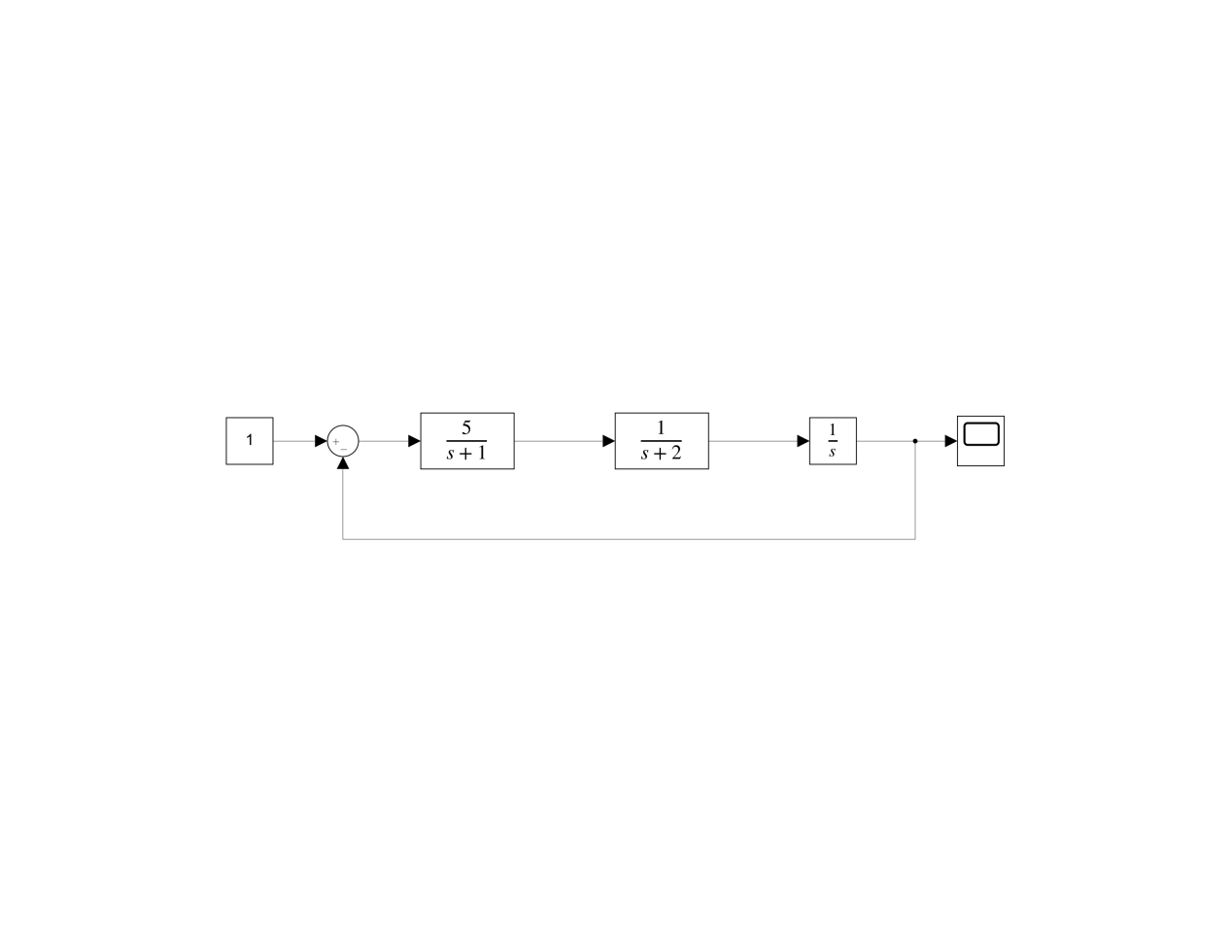


图 16 分析部分作业一第三题simulink图

%%

clear;

clc;

%%%LMI 判断稳定性

disp('1.用LMI YALMIP工具箱判断')

[A,B,C,D]=linmod('assign1\_3');

%描述待求的LMI

P=sdpvar(size(A,1),size(A,1),'symmetric');% 给出待求矩阵 symmetric为对称矩阵

Fcond=[P>0,A'\*P+P\*A<0];% 列出所有待求LMI

%求解LMI

ops=sdpsettings('verbose',0,'solver','sedumi');% 设置求解环境

diagnostics=solvesdp(Fcond,[],ops);% 迭代求解

[m,p]=checkset(Fcond);% 返回求解结果

tmin=min(m);% 验证是否满足

if tmin>0

disp('System is stable')

else

disp('System is unstable')

end

%%%特征值判断稳定性

disp('2.特征值判断稳定性')

Lambda=eig(A)

%%%能量的方法判断稳定性

disp('3.能量的方法判断稳定性')

P=eye(size(A,1));

Q=-P\*A-A'\*P;

try chol(Q);

disp('系统稳定')

catch ME

disp('无法判断')

end

%%%lyap（）函数

disp('4.lyap()函数判断')

Q=eye(size(A,1));

P=lyap(A,Q);

try chol(Q);

disp('系统稳定')

catch ME

disp('系统不稳定')

end

### 3.4.3 实验结果

1.用LMI YALMIP工具箱判断

System is stable

2.特征值判断稳定性

Lambda =

-0.047919570432540 + 1.311248044077123i

-0.047919570432540 - 1.311248044077123i

-2.904160859134920 + 0.000000000000000i

3.能量的方法判断稳定性

无法判断

4.lyap()函数判断

系统稳定

## 3.5 小结

在该部分的实验中，我学习了如何使用MATLAB绘制状态轨迹曲线、系统响应曲线；学习了如何使用MATLAB判断系统的能控能观性、如何进行能控、能观、能控能观性分解；在作业一的第三题中，学习了LMI YALMIP工具箱的使用，学会了如何导入外部工具箱，以及各类判断系统稳定性的方法。

在验证能控能观性以及对能控能观性进行分解时，子系统的能控能观性分解所获得的K值在转换为原式的K值要注意0所补的位置要正确。系统稳定性验证时，要注意理解希尔维斯判据的应用，才可以更好的理解老师所给的实例代码，从而自己设计系统稳定性判据的仿真。

在使用能量的方法判断稳定性（设P求Q）时，需要注意这种方法只能判断求解出Q为正定矩阵的情况，并给出“系统稳定”的结论，但无法判断Q为半正定切不恒为0的情况。使用lyap()函数求解时，可以明确给出“系统稳定”或“系统不稳定”的结论，是因为课本上说明了这种判断方法是充要条件。

在使用SIMULINK中的scope将数据导出到工作空间时，我发现做出的三维状态轨迹图非常的不美观，通过网上搜索和询问同学后得知可以将求解器改为ode3，并减小最小步长可以解决这一问题。使用plot3()可以将图片画成三维。

# 第四章系统设计部分实验

## 4.1 实验目标

①了解系统设计一般流程

②典型综合问题：镇定、极点配置、状态估计、基于状态观测器的状态反馈

③掌握常用控制器增益计算方法：LMI、极点配置

## 4.2 实验内容1

### 4.2.1 实验要求

➢ 设计状态反馈控制器，极点P1

➢ 设计状态观测器，极点P2

➢ 设计基于状态观测器的状态反馈，极点P1

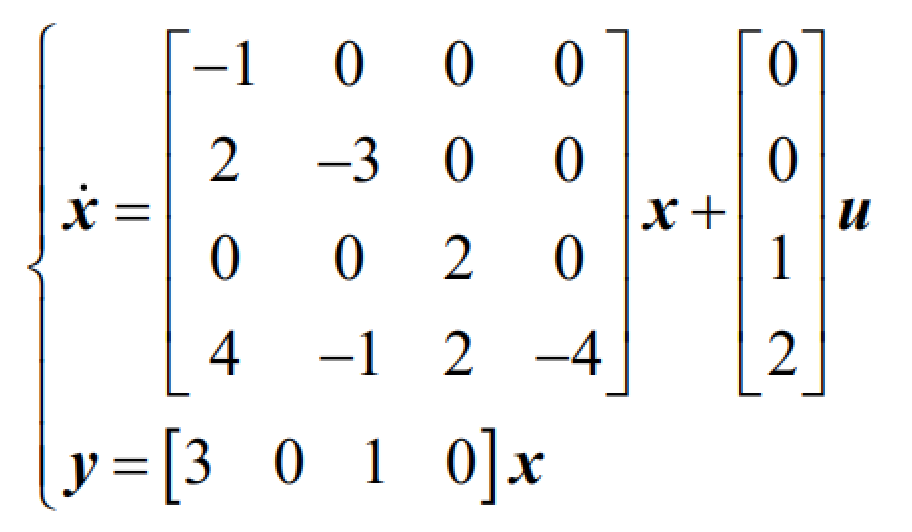
➢ P1, P2在合理条件下任意选

➢ 验证状态反馈控制效果

➢ 验证状态观测器效果

➢ 验证基于状态观测器的状态反馈控制效果

➢ 比较上述两类状态反馈的效果



### 4.2.2 实验程序

%%

%%%%%%%%%%%%%

%设计部分作业一

clc

clear

A=[-1 0 0 0;2 -3 0 0;0 0 2 0;4 -1 2 -4];

B=[0 0 1 2]';

C=[3 0 1 0];

D=[0];

n=size(A,1);

%设计状态反馈控制器 极点P1

if rank(ctrb(A,B)) == n

disp('系统能控，可通过状态反馈任意配置极点')

else

disp('系统不完全能控，不可通过状态反馈任意配置极点')

disp('对系统进行能控性分解')

m=rank(ctrb(A,B));

[Ac1,Bc1,Cc1,Tc1,Ko1]=ctrbf(A,B,C);

Ac=Ac1(m+1:size(Ac1,1),m+1:size(Ac1,1));

Bc=Bc1(m+1:size(Bc1,1));

Cc=Cc1(m+1:size(Cc1,2));

Acn=Ac1(1:m,1:m);

if max(eig(Acn))<0

disp('不能控子系统渐进稳定，系统可以被镇定')

P1=[-5,-6];

K1=acker(Ac,Bc,P1);

eig(Ac-Bc\*K1);

%变换

K=[0 0 K1(1) K1(2)]\*Tc1;

eig(A-B\*K)

ABK=A-B\*K

K

else

disp('系统不可被镇定')

end

end

%设计状态观测器 极点P2

if rank(obsv(A,C))==n

disp('系统能观，可任意配置观测器极点')

else

disp('系统不完全能观，不可任意配置观测器极点')

disp('对系统进行能观性分解')

o=rank(obsv(A,C));

[Ao1,Bo1,Co1,To1,Ko1]=obsvf(A,B,C);

Ao=Ao1(o+1:size(A,1),o+1:size(A,1));

Bo=Bo1(o+1:size(B,1));

Co=Co1(o+1:size(C,2));

Aon=Ao1(1:m,1:m);

if max(eig(Aon))<0

disp('不能观子系统渐进稳定，可以设计状态观测器')

P2=[-1,-2];

Ko1=acker(Ao',Co',P2);

Go1=Ko1';

Go=[0 0 Go1(1) Go1(2)]';

G=inv(To1)\*Go;

AGC=A-G\*C;

eig(AGC);

G

else

disp('不能控子系统不稳定')

end

end

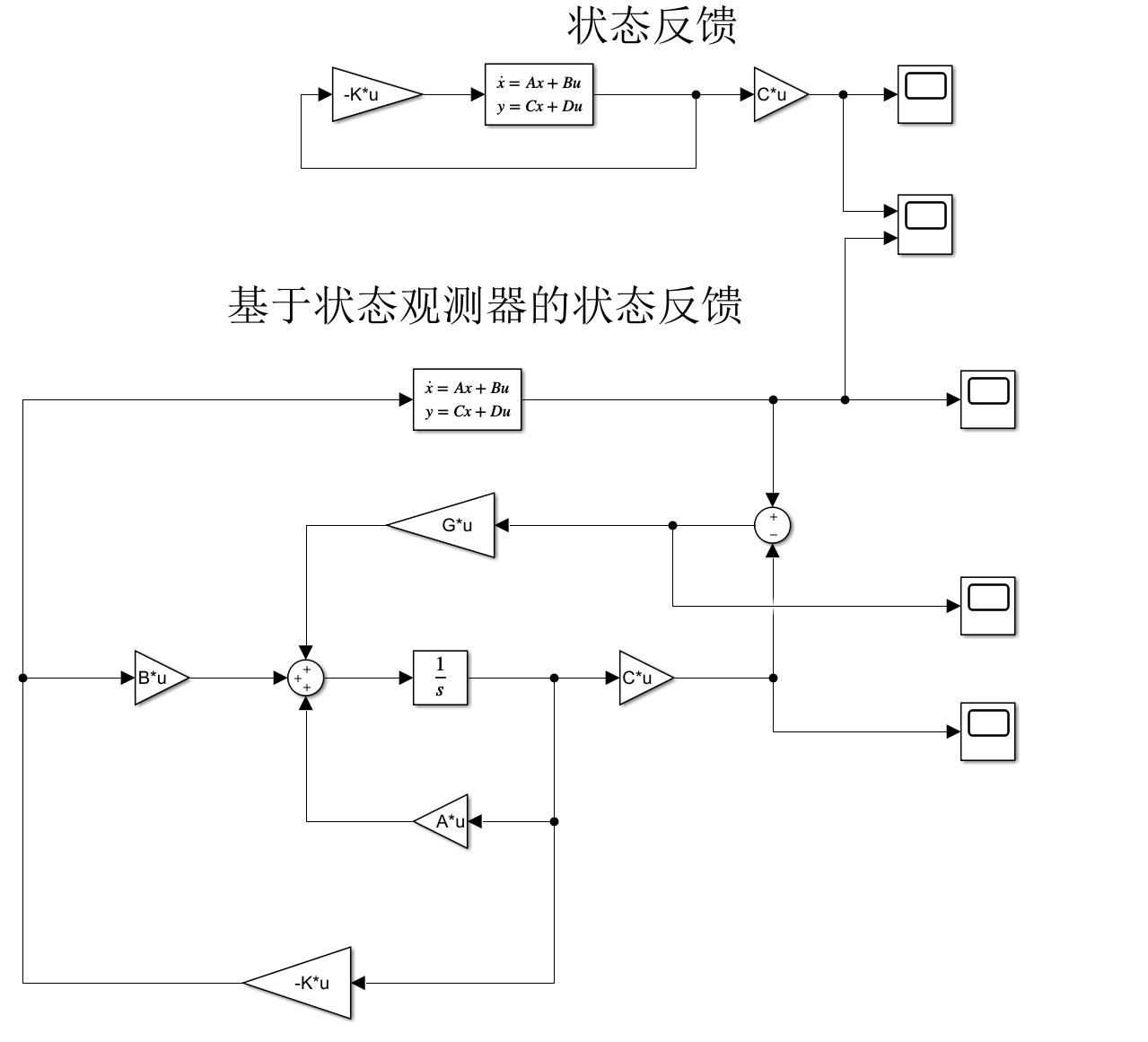


图 17 设计部分作业一simulink图

### 4.2.3 实验结果

系统不完全能控，不可通过状态反馈任意配置极点

对系统进行能控性分解

不能控子系统渐进稳定，系统可以被镇定

ans =

-5.0000

-6.0000

-3.0000

-1.0000

ABK =

-1.0000 0 0 0

2.0000 -3.0000 0 0

-0.0000 0 -7.4000 0.2000

4.0000 -1.0000 -16.8000 -3.6000

K =

0.0000 0 9.4000 -0.2000

系统不完全能观，不可任意配置观测器极点

对系统进行能观性分解

不能观子系统渐进稳定，可以设计状态观测器

G =

-0.0000

0

4.0000

0.0000

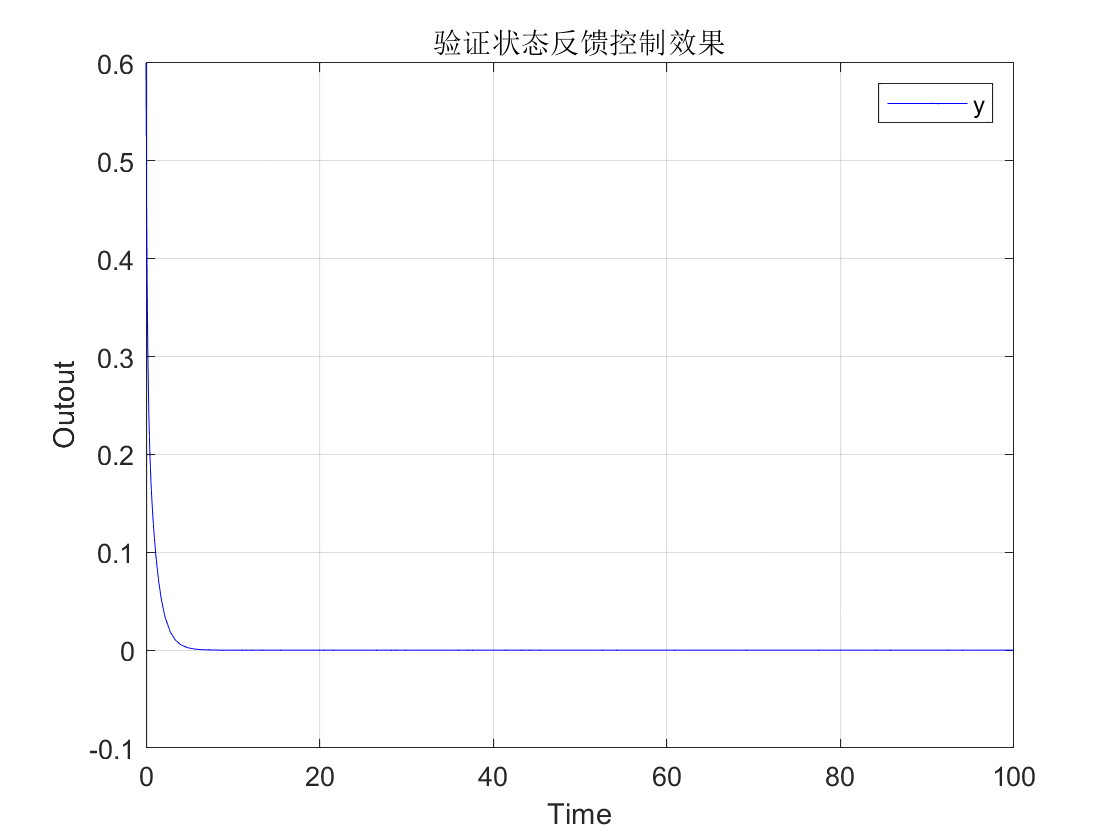


图 18 验证状态反馈控制效果

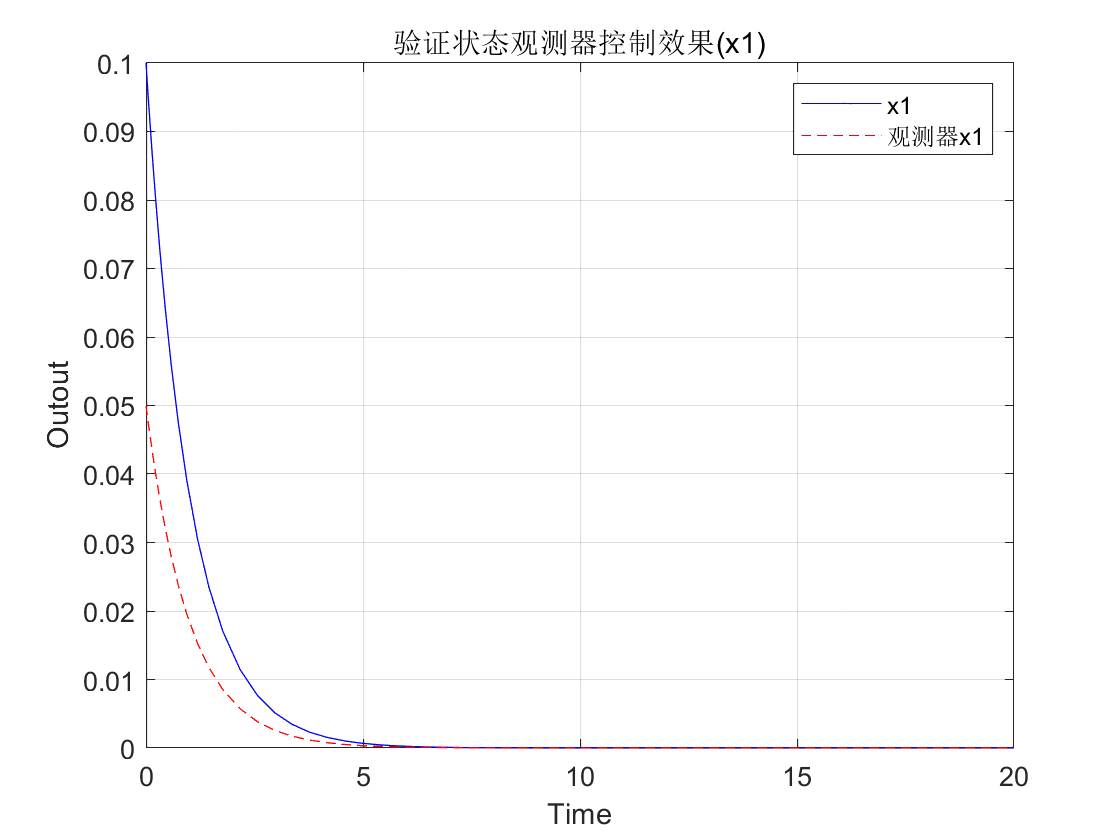


图 19 验证状态观测器控制效果（x1）

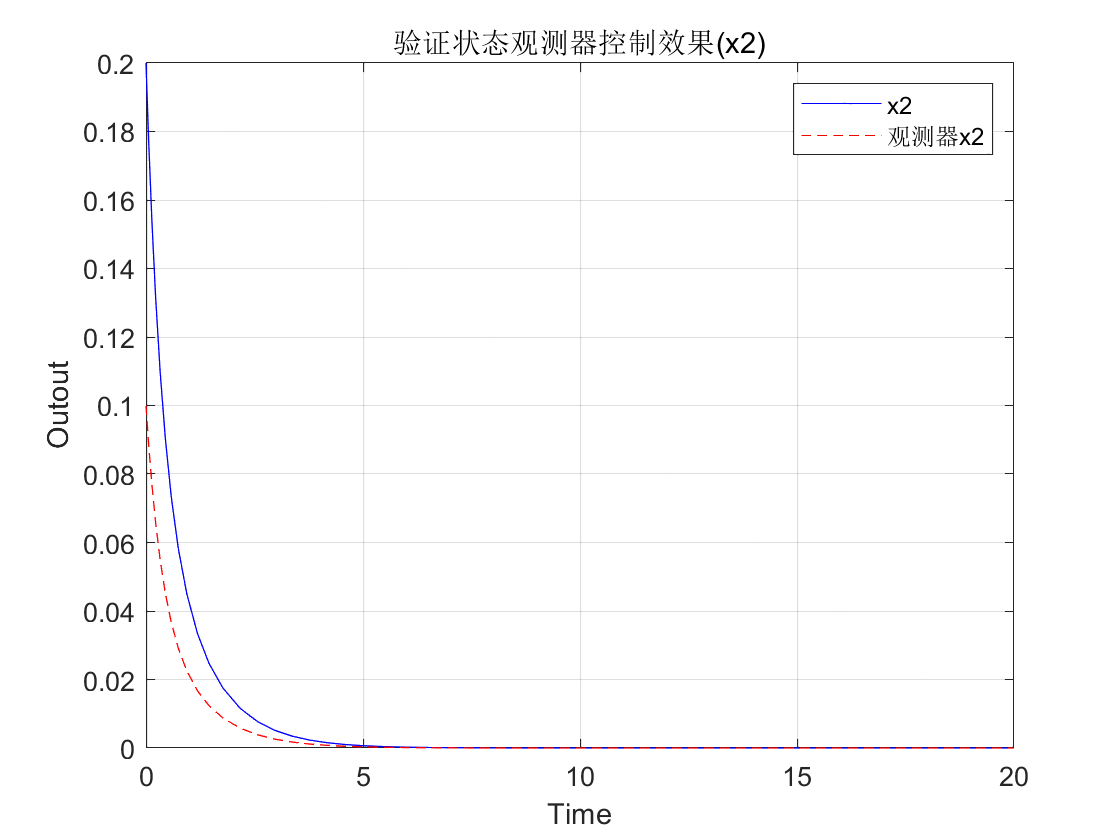


图 20 验证状态观测器控制效果（x2）

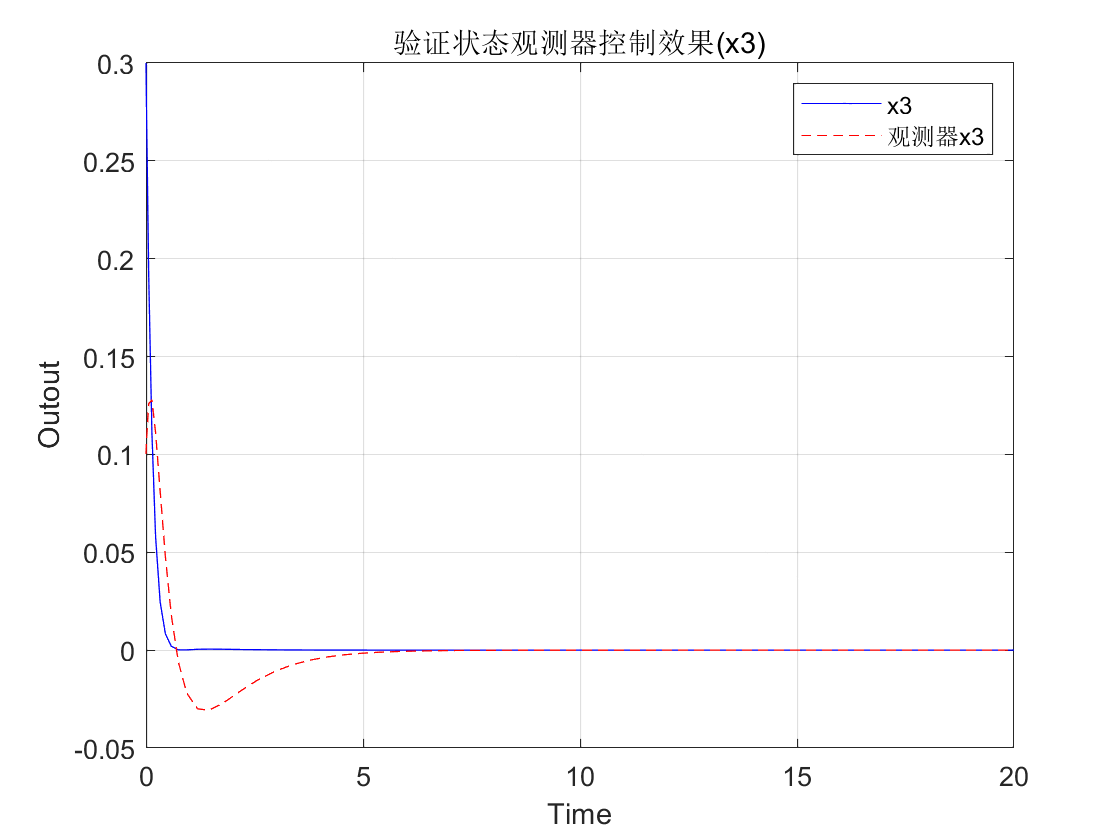


图 21 验证状态观测器控制效果（x3）

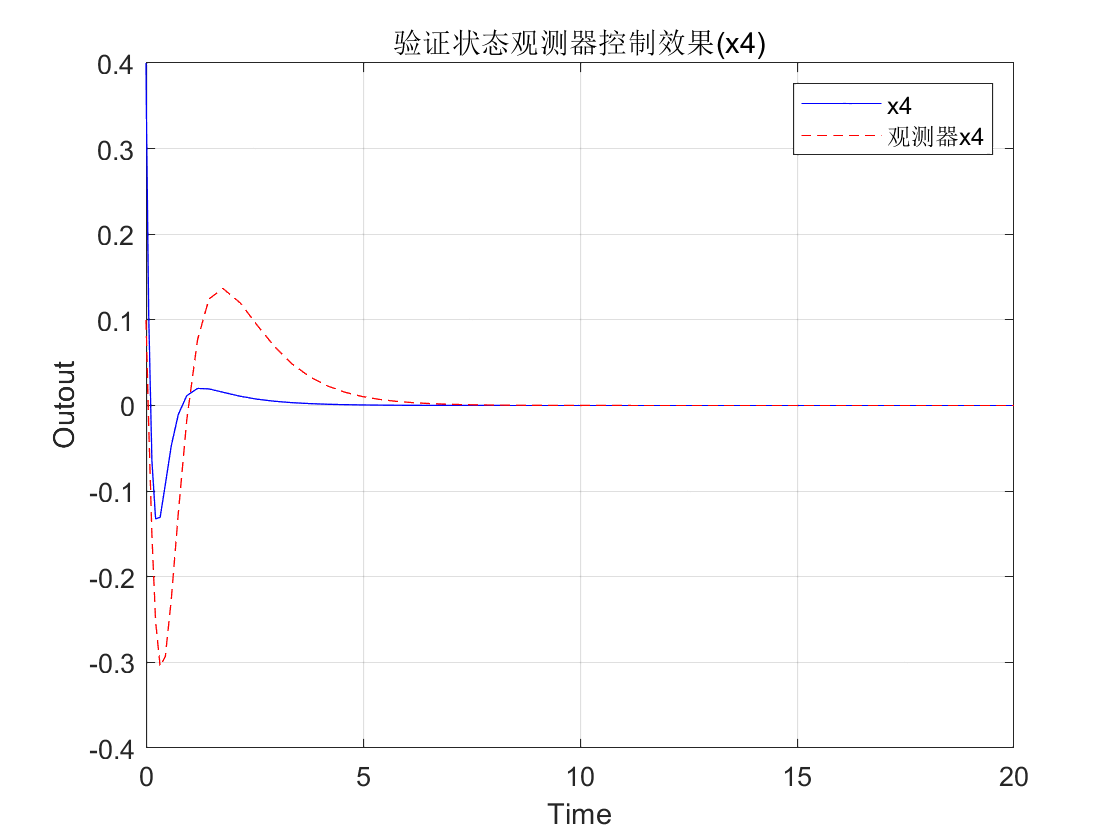


图 22 验证状态观测器控制效果（x4）

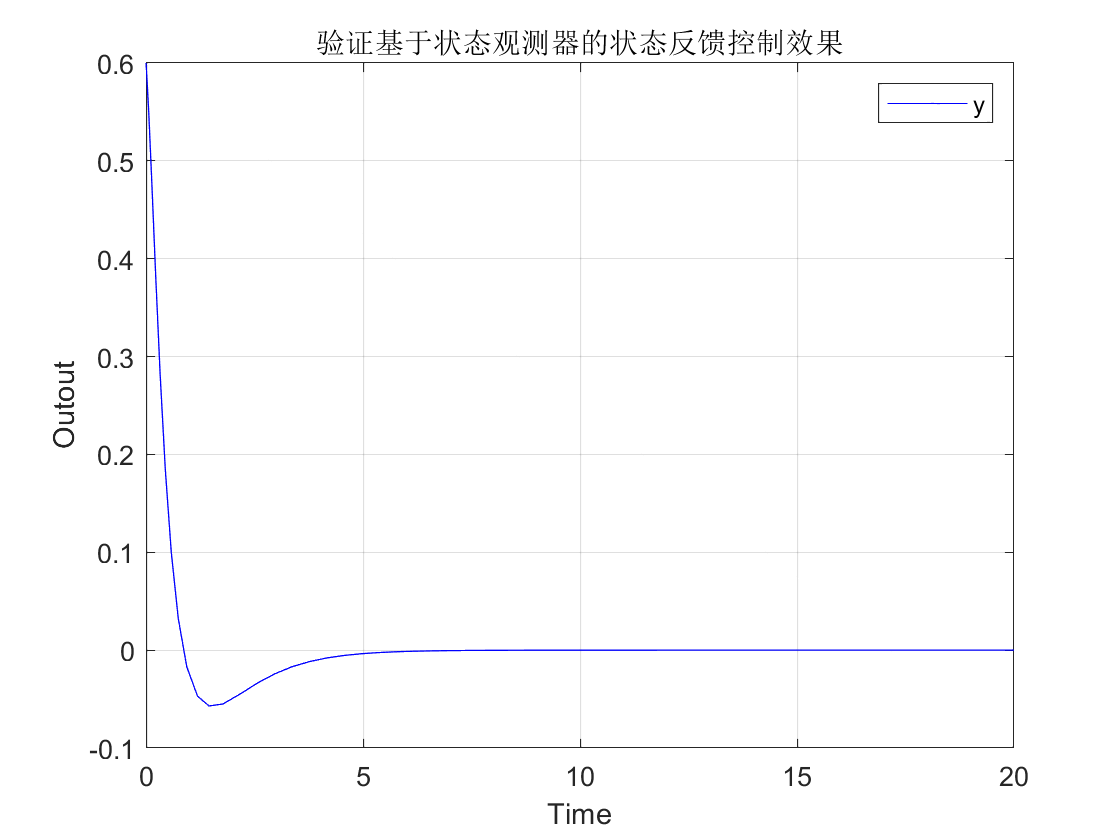


图 23 验证基于状态观测器的状态反馈控制效果

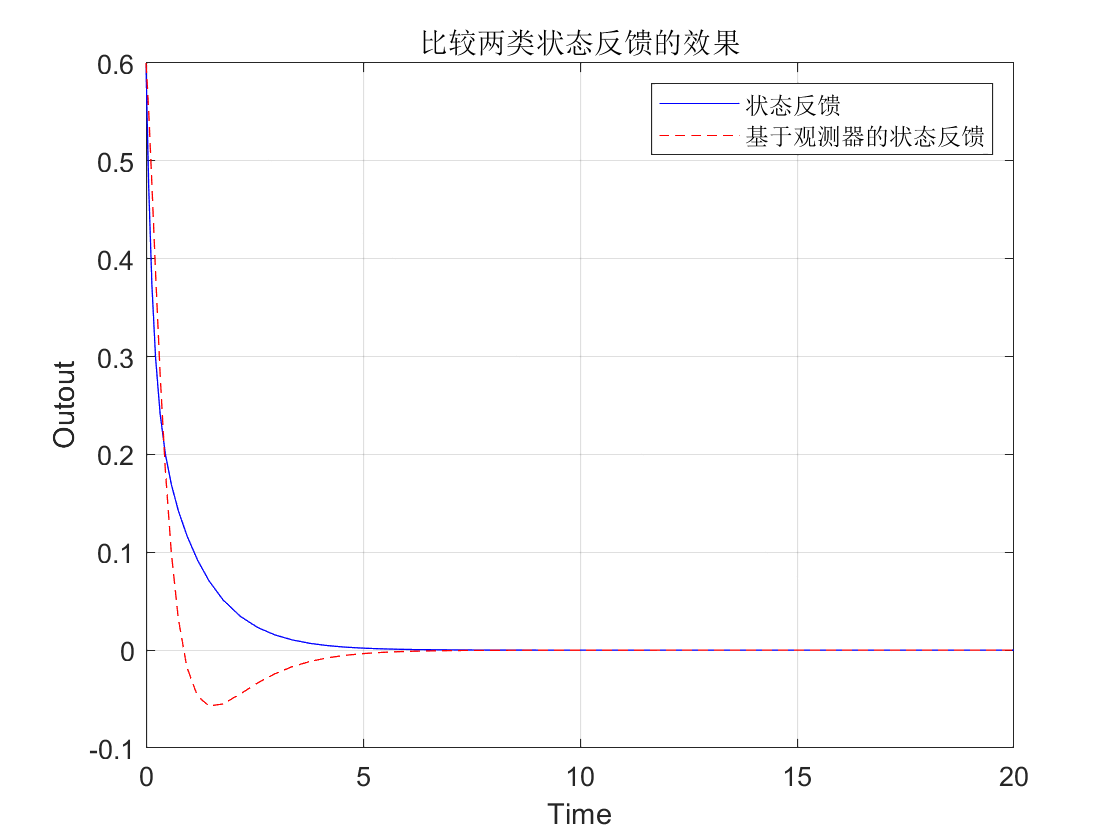


图 24 比较两类状态反馈的效果

## 4.3 小结

在寻找子系统框图时，要注意MATLAB中能控或能观子系统和不能控或不能观子系统的寻找位置与书中不一致。系统的综合设计中，状态反馈控制器的设计，要注意子系统K反推回原式时的补0的位置。

求解G的步骤与求解K的步骤是相似的，均可用acker/place()函数求解，但是要注意的是，求解过程中需要用AT和CT，这是因为求解K需要系统是完全能控的，求解G需要系统完全能观，两类系统是对偶关系，故需要使用AT和CT进行求解。求解后得到的G’矩阵是行向量，需要转化为列向量，并需要按照要求补0。

状态观测器的设计中，与状态反馈控制器的MATLAB代码很相似，但是在SIMULINK仿真中，要注意初值的设定，以及链接时改变gain的种类。基于状态观测器的状态反馈中，首先要计算出观测器的G值，在此基础上加装状态反馈。设计SIMULINK验证时，要注意接线的方向，否则会出现K值报错的情况。并且，在验证观测器效果时，观测器的初值和原系统的初值不能是相同的，不然无法验证观测器是否有效。

# 第五章实验总结与体会

## 5.1 实验总结

在本次实验中，我基本掌握了MATLAB在现代控制方面的基础用法，例如各类传递函数的使用与转化，以及MATLAB对现代控制知识点中的验证。在本次实验中，我首次学习使用了SIMIULINK模块，在SIMULINK中系统框图的搭建有许多非常细节的点，例如运行数据在SIMULINK中的映射、状态向量维数的相匹配都要注意、如何将scope数据导出到工作空间、如何修改求解器。在本次实验中，我认为最有意思的是，代码与仿真相结合，实验过程中，曾出现过多次报错，例如代码原理的错误、错误输入中午符号、在SIMULINK中使用scope输出数据时出现命名相同的问题、由于系统是发散的SIMIULINK在运行时超出限幅等。通过在网上搜索各类资料以及问过老师同学后，终将这些问题一一解决。

## 5.2 实验体会

在实验过程中，我明白了同学之间相互交流、分享经验的重要性，在与同学交流的过程中，可以博采众长，吸收他人的实验思路，代码思路，给自己的思考增添一些方向。此外，我们需要善于利用网络资源，在CSDN、GITHUB、百度、B站、谷歌中都有许多有用且免费的资源。在向别人提问前，自己先通过网络学习是很有必要的。可以说，在MATLAB代码编写过程中，所遇到的错误、所遇到的bug，都会有“网友”遇到过，在网络中都可以找到相应的回答。

总的来说，这次的实验非常有意义，当我们在看课本时，抽象且繁多的概念充斥眼前，很难在所有的概念中找到最基本的步骤，所以这次的实验，在一步步的代码编写中，我更加理解了现控概念的执行步骤和解题步骤，为我的线控学习提供了很好的基础。