

自动控制原理 II：线性系统分析与设计

实验报告

二〇二一年十二月

目录

第一章 实验简介.....	1
1.1 实验目标.....	1
1.2 实验内容.....	1
1.3 实验平台.....	1
第二章 系统模型部分实验.....	1
2.1 实验目标.....	1
2.2 实验内容 1.....	1
2.2.1 实验要求.....	1
2.2.2 实验程序.....	2
2.2.3 实验结果.....	3
2.3 实验内容 2.....	4
2.3.1 实验要求.....	4
2.3.2 实验程序.....	4
2.3.3 实验结果.....	5
2.4 小结.....	6
第三章 系统分析部分实验.....	7
3.1 实验目标.....	7
3.2 实验内容 1.....	7
3.2.1 实验要求.....	7
3.2.2 实验程序.....	7
3.2.3 实验结果.....	7
3.3 实验内容 2.....	8
3.3.1 实验要求.....	8
3.3.2 实验程序.....	9
3.3.3 实验结果.....	11
3.4 实验内容 3.....	11
3.4.1 实验要求.....	11
3.4.2 实验程序.....	12
3.4.3 实验结果.....	13
3.5 小结.....	14
第四章 系统设计部分实验.....	15
4.1 实验目标.....	15
4.2 实验内容 1.....	15
4.2.1 实验要求.....	15
4.2.2 实验程序.....	15
4.2.3 实验结果.....	20
4.3 小结.....	23
第五章 实验总结与体会.....	24
5.1 实验总结.....	24
5.2 实验体会.....	24

第一章 实验简介

1.1 实验目标

- 加深对线性系统分析与设计理论知识的理解与掌握
- 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

- 《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
- 三大部分：系统模型部分、系统分析部分、系统设计部分

1.3 实验平台

基于 MATLAB 软件实现

第二章 系统模型部分实验

2.1 实验目标

- 学会 MATLAB 的简单实用
- 学会线性系统各类数学模型的表示
- 线性系统各类数学模型之间的相互转化
- 子系统的连接合并

2.2 实验内容 1

2.2.1 实验要求

- 利用 MATLAB 给出图 2.1 所示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

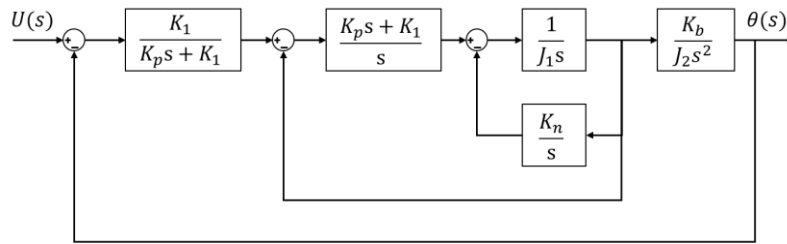


图 2.1 系统模型部分作业一内容 1 系统结构图

2.2.2 实验程序

根据实验要求中系统结构图选取参数如表 2.1 所示:

表 2.1 系统模型部分作业一内容 1 参数选择

参数	值
K_1	1
K_p	1
K_n	1
K_b	1
J_1	1
J_2	1

按照要求利用 MATLAB 求解系统传递函数与状态空间表达式。

```
%% 参数选定
KI = 1;
KP = 1;
KN = 1;
J1 = 1;
J2 = 1;
KB = 1;
%% 计算传递函数
sys1 = tf(1,[J1 0]);
sys2 = tf(KN,[1 0]);
loop1 = feedback(sys1,sys2); %最内环传递函数
sys3 = tf([KP KI],[1 0]);
loop2 = feedback(sys3*loop1,1); %次内环传递函数
sys4 = tf(KI,[KP KI]);
sys5 = tf(KB,[J2 0 0]);
onesys = tf(1,1);
system = feedback(sys4*loop2*sys5,onesys,-1) %系统传递函数
num = system.num{1};
den = system.den{1};
%% 计算状态空间表达式
[A,B,C,D] = tf2ss(num,den); %ABCD 对应系统状态空间表达式 ABCD
```

随后在 Simulink 按照代码运行结果搭建模型如图 2.2 所示, 其中 State-Space

模块参数由 Matlab 代码运行结果给出：

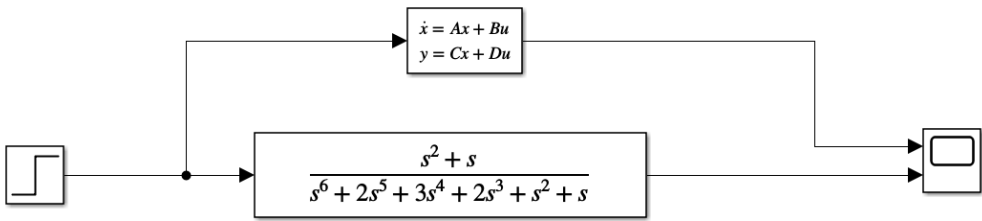


图 2.2 系统模型部分作业一内容 1 Simulink 仿真示意图

2.2.3 实验结果

运行 Matlab 程序代码可以直接得出系统传递函数如下所示：

```
system =  
  
      s^2 + s  
-----  
s^6 + 2 s^5 + 3 s^4 + 2 s^3 + s^2 + s  
Continuous-time transfer function.
```

同时也可以获得系统的状态空间表达式如下所示：

```
A = 6x6  
    -2    -3    -2    -1    -1     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 = 6x1  
     1  
     0  
     0  
     0  
     0  
     0  
  
C = 1x6  
     0     0     0     1     1     0  
  
D = 0
```

根据计算出的 A、B、C、D 矩阵可以利用 Simulink 进行传递函数模型何状态空间模型的比较如图 2.3 所示：

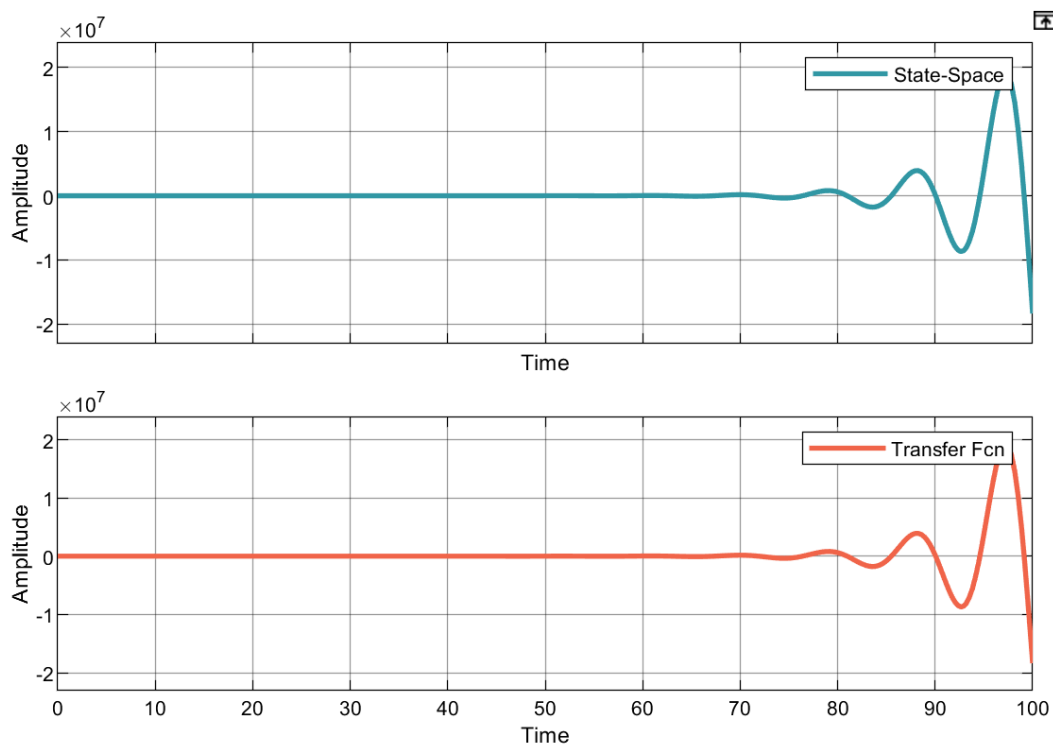


图 2.3 系统模型部分作业一内容 1 状态空间模型与传递函数模型响应曲线对比图

2.3 实验内容 2

2.3.1 实验要求

- 利用 Simulink 搭建如下公式系统结构框图
- 选取初始条件 $[-0.2;0.3;0.7]$ ，观察状态相应

$$\begin{cases} \dot{x}_1(t) = a[x_2(t) - h(x_1(t))] \\ \dot{x}_2(t) = x_1(t) - x_2(t) + x_3(t) \\ \dot{x}_3(t) = -bx_2(t) \\ p(t) = x_1(t) \end{cases}$$

其中

$$h(x) = m_1 x_1(t) + \frac{1}{2}(m_0 - m_1)(|x_1(t) + c| - |x_1(t) - c|)$$

参数满足：

$$a = 9, b = 14.28, c = 1, m_0 = -\left(\frac{1}{7}\right), m_1 = \frac{2}{7}$$

2.3.2 实验程序

根据题目要求搭建 Simulink 仿真平台如图 2.4 所示：

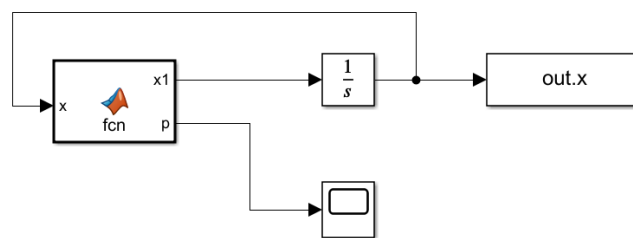


图 2.4 系统模型部分作业一内容 2 Simulink 仿真示意图

其中 MATLAB Function 代码编写如下：

```
%% MATLAB Function 代码
function [x1,p] = fcn(x)
    x1 = zeros(3,1);
    a = 9;
    b = 14.28;
    c = 1;
    m0 = -1/7;
    m1 = 2/7;
    h = m1*x(1)+1/2*(m0-m1)*(abs(x(1)+c)-abs(x(1)-c));
    x1(1) = a*(x(2)-h);
    x1(2) = x(1)-x(2)+x(3);
    x1(3) = -b*x(2);
    p = x(1);
```

通过将数据导至工作区利用 plot 函数进行制图：

```
%% 绘图代码
plot(out.x.Time,out.x.Data(:,1),out.x.Time,out.x.Data(:,2),out.x.Time,o
ut.x.Data(:,3))
```

2.3.3 实验结果

图 2.5 展示了系统三个状态随时间变化的响应曲线，均呈现无周期性的振荡状态。

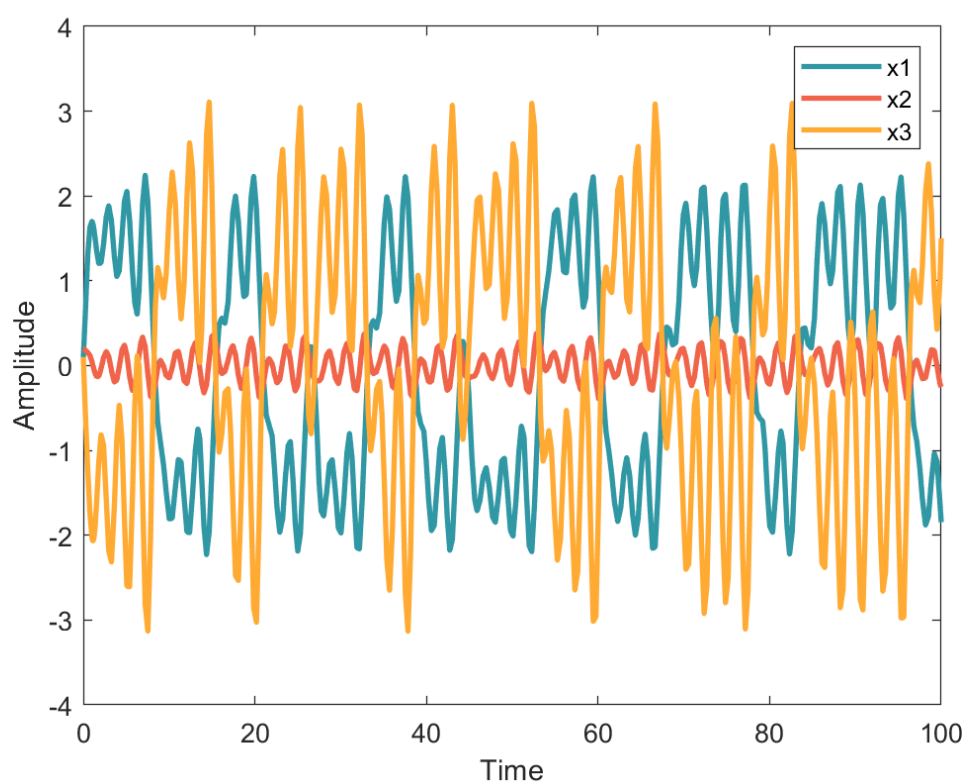


图 2.5 系统模型部分作业一内容 2 状态响应结果图

2.4 小结

- 在从结构框图到状态空间表达式的转换过程中发现，相同的传递函数可能得出不同的状态空间表达式，说明了状态空间表达式的非唯一性。
- 实验内容 2 所仿真系统为蔡氏混沌电路的仿真模型，初步尝试可以发现系统输出无规律可言同时不同的初始状态会导致系统收敛与发散。

第三章 系统分析部分实验

3.1 实验目标

- 系统解的获取，即系统响应曲线的绘制
- 线性定常系统能控性、能观性的判断，及其相关的坐标变换
- 线性定常系统稳定性判断

3.2 实验内容 1

3.2.1 实验要求

同系统模型部分实验内容 2 系统，选择两组初值 $[-0.1, 0.1, 0.2]$, $[1, 2, 3]$

- 绘制系统响应曲线
- 绘制系统状态轨迹

3.2.2 实验程序

Simulink 仿真平台与图 2.3 中搭建平台相同，仅仅通过修改初值即可。其中系统响应曲线可以直接由 Scope 模块获取，状态轨迹可以将系统状态数据输出至工作区，绘图代码如下：

```
%% 绘制状态轨迹
figure(1)
subplot(1,2,1)
plot3(out.x.Data(:,1),out.x.Data(:,2),out.x.Data(:,3))
subplot(1,2,2)
plot3(out.x1.Data(:,1),out.x1.Data(:,2),out.x1.Data(:,3))
```

3.2.3 实验结果

实验结果如图 3.1, 3.2 所示，其中当初始状态为 $[-0.1, 0.1, 0.2]$ 时，系统响应曲线无周期振荡，其状态轨迹呈现双圆环状曲线；当初始状态为 $[1, 2, 3]$ 时，系统不稳定其系统响应曲线发散，状态轨迹呈现按照圆环状不断向外拓展圆环的形式。

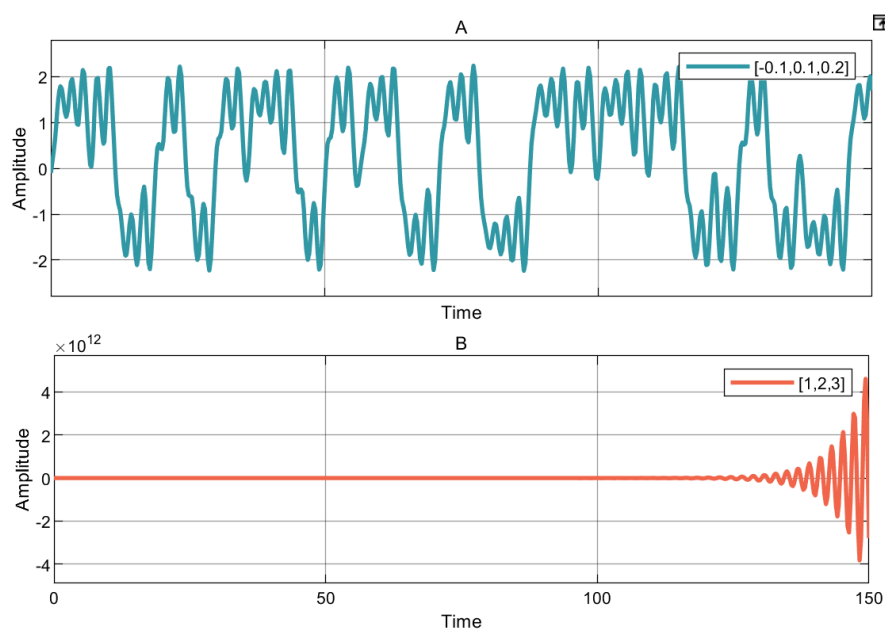


图 3.1 系统分析部分内容 1 系统响应曲线. A:初值 $[-0.1, 0.1, 0.2]$, B:初值 $[1, 2, 3]$

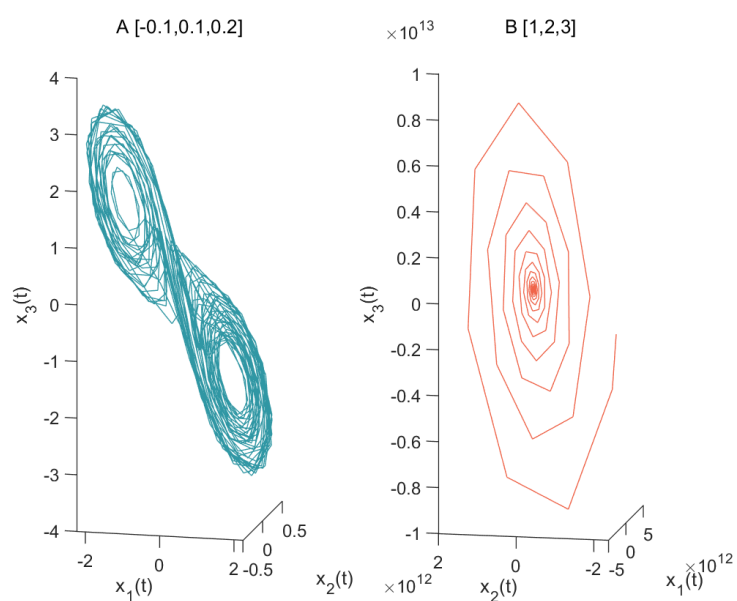


图 3.2 系统分析部分内容 1 状态轨迹. A:初值 $[-0.1, 0.1, 0.2]$, B:初值 $[1, 2, 3]$

3.3 实验内容 2

3.3.1 实验要求

判断如下系统的能控能观性，若不能完全能控且不完全能观，求其能控能观子系统。

$$\left\{ \begin{array}{l} \dot{x} = \begin{bmatrix} -4 & 1 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} x + \begin{bmatrix} 1 & 3 \\ 5 & 7 \\ 4 & 3 \\ 0 & 0 \\ 1 & 6 \\ 0 & 0 \end{bmatrix} u \\ y = \begin{bmatrix} 3 & 1 & 0 & 5 & 0 & 0 \\ 1 & 4 & 0 & 2 & 0 & 0 \end{bmatrix} x \end{array} \right.$$

3.3.2 实验程序

```

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];
%% 能控能观检验
Qc = ctrb(A,B);
Qo = obsv(A,C);
rc = rank(Qc);
ro = rank(Qo);
L = size(A);
if rc==L
    disp("系统能控")
else
    disp("系统不完全能控")
end
if ro==L
    disp("系统能观")
else
    disp("系统不完全能观")
end
% 先按照能观性分解
[Achat,Bchat,Cchat,Pc,Kc] = ctrbf(A,B,C)
% 抽取不能控子系统进行能观性分解
Ac1 = Achat([1,L(1)-rc],[1,L(1)-rc]);
Bc1 = Bchat([1,L(1)-rc,:])*0;
Cc1 = Cchat(:,[1,L(1)-rc]);
% 判断其能观性能
Qo1 = obsv(Ac1,Cc1);
ro1 = rank(Qo1);
L1 = size(Ac1);
if ro1==L1
    disp("系统能观")
else
    disp("系统不完全能观")

```

```

end
[Ac1ohat,Bc1ohat,Cc1ohat,Pc1o,Kc1o] = obsvf(Ac1,Bc1,Cc1)
% 抽取其子系统, 先抽取不能控但能观察部分
if ro1==1
    A33 = Ac1ohat(L1(1),L1(1));
    A44 = Ac1ohat([L1(1)-ro1],[L1(1)-ro1]);
    A43 = Ac1ohat(L1(1),[L1(1)-ro1]);
else
    A33 = Ac1ohat(L1(1)-ro1+1:L1(1),L1(1)-ro1+1:L1(1));
    A44 = Ac1ohat(1:L1(1)-ro1,1:L1(1)-ro1);
    A43 = Ac1ohat(L1(1)-ro1+1:L1(1),1:L1(1)-ro1);
end
C3 = Cc1ohat(:,L1(1)-ro1+1:L1(1))
% 现在对能控子系统进行能观性分解, 需要提取 A1 子系统
Ac0 = Achat(L(1)-rc+1:L(1),L(1)-rc+1:L(1));
Bc0 = Bchat(L(1)-rc+1:L(1),:);
Cc0 = Cchat(:,L(1)-rc+1:L(1));
% 判断其能观性能
Qo0 = obsv(Ac0,Cc0);
ro0 = rank(Qo0);
L0 = size(Ac0);
if ro0==L0
    disp("系统能观")
else
    disp("系统不完全能观")
end
[Ac0ohat,Bc0ohat,Cc0ohat,Pc0o,Kc0o] = obsvf(Ac0,Bc0,Cc0);
%抽取其子系统, 先抽取不能控但能观察部分
A11 = Ac0ohat(L0(1)-ro0+1:L0(1),L0(1)-ro0+1:L0(1))
A22 = Ac0ohat(1:L0(1)-ro0,1:L0(1)-ro0);
A21 = Ac0ohat(1:L0(1)-ro0,L0(1)-ro0+1:L0(1));
B1 = Bc0ohat(L0(1)-ro0+1:L0(1),:);
B2 = Bc0ohat(1:L0(1)-ro0,:);
C1 = Cc0ohat(:,L0(1)-ro0+1:L0(1));
% 现在对能控子系统进行能观性分解, 需要提取 A2 子系统
A2 = Achat(L(1)-rc+1:L(1),1:L(1)-rc);
A_2 = Pc0o*A2*inv(Pc1o);
% 抽取其子系统
A13 = A_2(3:4,2);
A24 = A_2(1:2,1);
A23 = A_2(1:2,2);
%组合系统
AA = [A11,zeros(2),A13,[0;0];
      A21 A22 A23 A24];

```

```
[0,0],[0,0],A33,0;  
[0,0],[0,0],A43,A44;]  
BB = [B1;B2;[0,0];[0,0]]  
C = [C1,zeros(2),C3,[0;0]]
```

3.3.3 实验结果

针对系统使用了能控能观分解，经过三次分解，分别求出 A11, A22 等相关子系统矩阵，最终再组合为分解后的矩阵，如下所示：

```
AA = 6x6  
-4.4472    0.7236         0         0   -0.0000         0  
-0.2764   -3.5528         0         0   -0.0000         0  
 0.0000   -0.0000    0.9981    2.0000    0.7068    0.7074  
 0.0000   -0.0000    2.0000    1.0019    0.7074   -0.7068  
         0         0         0         0    3.0000         0  
         0         0         0         0   -0.0000   -1.0000  
BB = 6x2  
-1.7780   -1.1282  
 4.7790    7.5317  
-3.5345   -6.3650  
-2.1230    2.1183  
         0         0  
         0         0  
C = 2x6  
 2.0262    2.4278         0         0   -5.0000         0  
-1.2523    3.9283         0         0   -2.0000         0
```

上述分解结果中，能控能观性维度为 2*2，能控不能观性维度为 2*2，不能控能观性维度为 1*1，不能控不能观性维度为 1*1。以上四个子系统从上到下依次排列。

3.4 实验内容 3

3.4.1 实验要求

判断如下系统的稳定性：

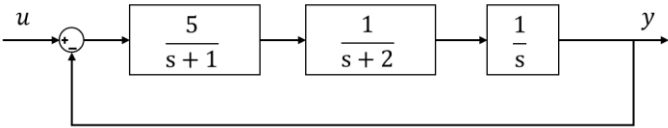


图 3.3 系统分析部分内容 2 系统框图

3.4.2 实验程序

首先, 根据题目给定的系统模型建立状态空间模型。

```
%% 建立系统模型
sys1 = tf(5,[1 1]);
sys2 = tf(1,[1 2]);
sys3 = tf(1,[1 0]);
sys = series(sys1,sys2);
sysa = series(sys,sys3);
onesys = tf(1,1);
loop = feedback(sysa,onesys,-1);
num = loop.num{1}
den = loop.den{1}
[A,B,C,D] = tf2ss(num,den)
```

根据李雅普诺夫第一方法判断特征根实部如下:

```
% 1.李雅普诺夫第一方法: 判断特征根
Lambda = eig(A)
judge = real(Lambda)<0
if(all(judge(:)==1))
    disp("系统稳定")
else
    disp("系统不稳定")
end
```

利用李雅普诺夫第二法, 认为提供正定矩阵 P , 判断矩阵 Q 的正定性:

```
% 2.直接法: 正能量是否衰减?
P = eye(size(A,1));
Q = -P*A-A'*P;
% 2.1 判断矩阵 Q 的正定性, 采用特征值判断, 但是
% 总感觉与方法一有所相似, 没必要
Lambda1 = eig(Q)
judge1 = real(Lambda1)>0
if(all(judge1(:)==1))
    disp("系统稳定")
else
    disp("此方法无法判断")
end

% 2.2 判断矩阵的顺序主子式
Det = zeros(1,size(A,1));
for k = 1:size(A,1)
    Det(1,k) = det(Q(1:k,1:k))
end
```

```

if min(Det)>0
    disp('系统稳定')
else
    disp('此方法无法判断') %要求对于任意 Q 求 P
end

```

同样可以先指定 Q 来反求 P 并判断 P 的正定性：

```

% 3.指定 Q 反求 P
% 注意 A 要转置，不转置也可以，求特征值或者顺序主子式效果一样
A = A';
Q = eye(size(A,1));
P = lyap(A,Q);
Lambda2 = eig(Q);
judge2 = real(Lambda2)>0
if(all(judge2(:)==1))
    disp("系统稳定")
else
    disp("系统不稳定")
end

```

最后，利用线性矩阵不等式的方法，判断两个不等式同时满足的可能性：

```

% 4.YALMIP toolbox, LMI:线性矩阵不等式
P = sdpvar(size(A,1),size(A,1),'symmetric');%建立矩阵
Fcond = [P>0,A'*P+P*A<0];%待求解 LMI

ops = sdpsettings('verbose',0,'solver','sedumi');
diagnostics = solvesdp(Fcond,[],ops);
[m,p] = checkset(Fcond);
tmin = min(m);
% 残差为正代表解可行即存在 P
if tmin>0
    disp('系统稳定')
else
    disp('系统不稳定')
end

```

3.4.3 实验结果

对应方法一，求得特征根与结果如下：

```

Lambda = 3×1 complex
-2.9042 + 0.0000i
-0.0479 + 1.3112i
-0.0479 - 1.3112i

```

系统稳定

方法二由于判断条件有限，无法判断最终结果，但是不能证明系统不稳定，方法三与方法四得出结果均为系统稳定。

3.5 小结

- MATLAB 中的能控能观分解与书本中讲解的分解方法存在着形式上的差别，其中 T 矩阵存在转置关系，相应的完全能观完全能空等分块矩阵的所在方位也需要注意。
- 判断系统稳定性可以直接由特征根所在平面位置判断，通过李雅普诺夫第二方法判断是，若是通过确定 P 矩阵求得 Q 未能得到稳定结果，不能直接下结论系统不稳定而是需要通过其他判断方法进行进一步分析。
- YALMIP 工具箱在使用过程中可能针对人为给定的 LMI 方程进行约束条件的加强从而给出提示，实验发现这并不影响最终结果。

第四章 系统设计部分实验

4.1 实验目标

- 了解系统设计一般流程
- 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
- 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

针对如下系统进行相应设计：

$$\begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 2 & -3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 4 & -1 & 2 & -4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} u \\ \mathbf{y} = [3 \ 0 \ 1 \ 0] \mathbf{x} \end{cases}$$

- 设计状态反馈控制器，极点自选
- 设计状态观测器，极点自选
- 设计基于状态观测器的状态反馈，极点自选
- 验证状态反馈控制效果
- 验证基于状态观测器的状态反馈控制系统
- 比较上述两类状态反馈的效果

4.2.2 实验程序

4.2.2.1 状态反馈设计与验证

首先，根据题目给定系统建立状态空间表达式，并分析系统能控性，针对不能控子系统需要满足镇定条件才能够进行状态反馈设计。

```
% 根据给定建立系统
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);
%%能控性判断
Qc=ctrb(A,B);
nc=rank(Qc);
if rank(Qc)==n
    disp('系统完全能控')
else
    disp('系统不完全能控')
    [A1,B1,C1,T1,k1]=ctrbf(A,B,C);
    Auc=A1((1:(n-nc)),(1:(n-nc)));
    eig(Auc);
    if max(real(eig(Auc)))>0
        disp('由于不可控部分特征根为正，通过状态反馈不能使系统镇定')
    else
        disp('不可控部分渐进稳定，通过状态反馈能使系统镇定，且系统不可控部分特征根为: ')
        eig(Auc)
    end
end
end

```

若系统不能控，则对能控子系统进行提取，将极点配置为-4，-3，-2，-1。

```

%%能控性提取
disp('原系统能控性部分提取如下: ')
Ac = A1((n-nc+1):n,(n-nc+1):n)
Bc = B1((n-nc+1):n,:)
Cc = C1(:,(n-nc+1):n)
disp('能控部分配置以下极点: ')
P1=[-4 -2]
disp('能控部分的状态反馈增益为')
K1=place(Ac,Bc,P1)

%%对于原系统而言状态反馈增益计算
disp('原系统的状态反馈增益为')
K1=[0 0 K1];
K1=K1*T1
ABK=A-B*K1
%%配置检验
disp('经过状态反馈后，系统特征根如下: ')
eig(A-B*K1)

```

针对所求得的增益矩阵可以在 Simulink 中进行相关验证，仿真平台搭建如下，由于 MATLAB 编程中 K 的正负与书本不符需要在实际搭建过程中改为负反馈。

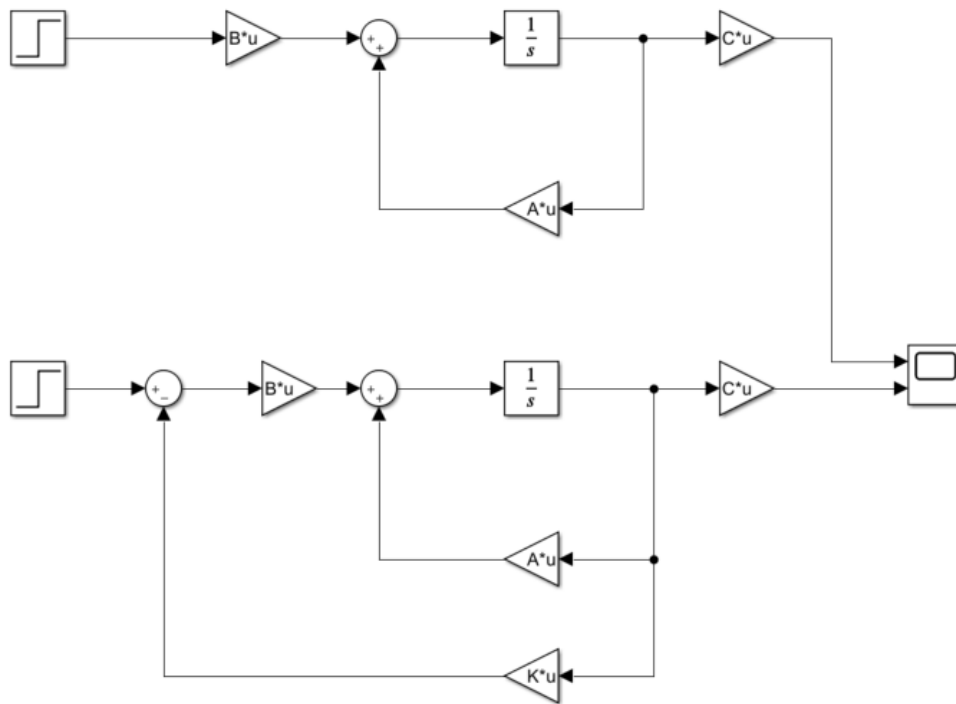


图 4.1 系统设计部分内容 1 状态反馈仿真平台

4.2.2.2 基于状态观测器的状态反馈设计与验证

状态观测器的状态反馈具有分离定理，可以将观测器与状态反馈分开设计，其中状态反馈部分可以直接采用上一部分的 K 矩阵，在此部分可以直接对反馈矩阵 G 进行相关设计。能够设计状态观测器的状态反馈要求系统不能观部分镇定，首先需要判断系统能观性并在此基础上进行相关设计，将极点配置为-4, -3, -2, -1。

```
%能观性检验
Qo=obsv(A,C);
no=rank(Qo);
if no==n
    str = '系统完全能观'
else
    str = '系统不完全能观'
    [A2,B2,C2,T2,k2]=obsvf(A,B,C);
    Auo=A2((1:(n-no)),(1:(n-no)));
    eig(Auo);
    if max(real(eig(Auo)))>0
        disp('由于不能观部分特征根为正，通过状态观测器不能使系统镇定')
    else
```

```

disp('不能观部分渐进稳定，通过状态观测器能使系统镇定，且系统不能观部分特
征根为：')
eig(Auo)
end
end
%%能观性提取
disp('原系统能观性部分提取如下：')
Ao = A2((n-no+1):n,(n-no+1):n)
Bo = B2((n-no+1):n,:)
Co = C2(:,(n-no+1):n)
disp('能观部分配置以下极点：')
P2=[-1 -2]
Ao1=Ao';
Bo1=Co';
K2=acker(Ao1,Bo1,P2);
G=K2'
G=[0;0;G]
G = inv(T2)*G;
AGC=A-G*C
%%配置检验
disp('经过状态观测器后，系统特征根如下：')
eig(AGC)

```

在此基础上用 Simulink 搭建相应模型如下图所示，其中图 4.2 用于验证状态观测器的观测效果，图 4.3 用于验证基于状态观测器反馈的最终效果。

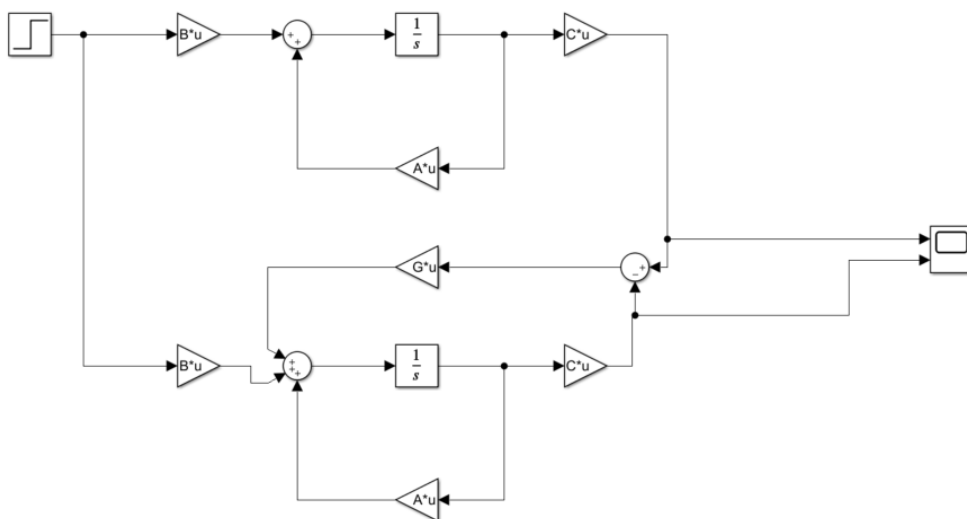


图 4.2 系统设计部分内容 1 状态观测器设计仿真平台

在图 4.2 所示仿真平台基础上搭建状态观测器反馈仿真，修改跟随系统的初始条件为[0;0;2;1]。

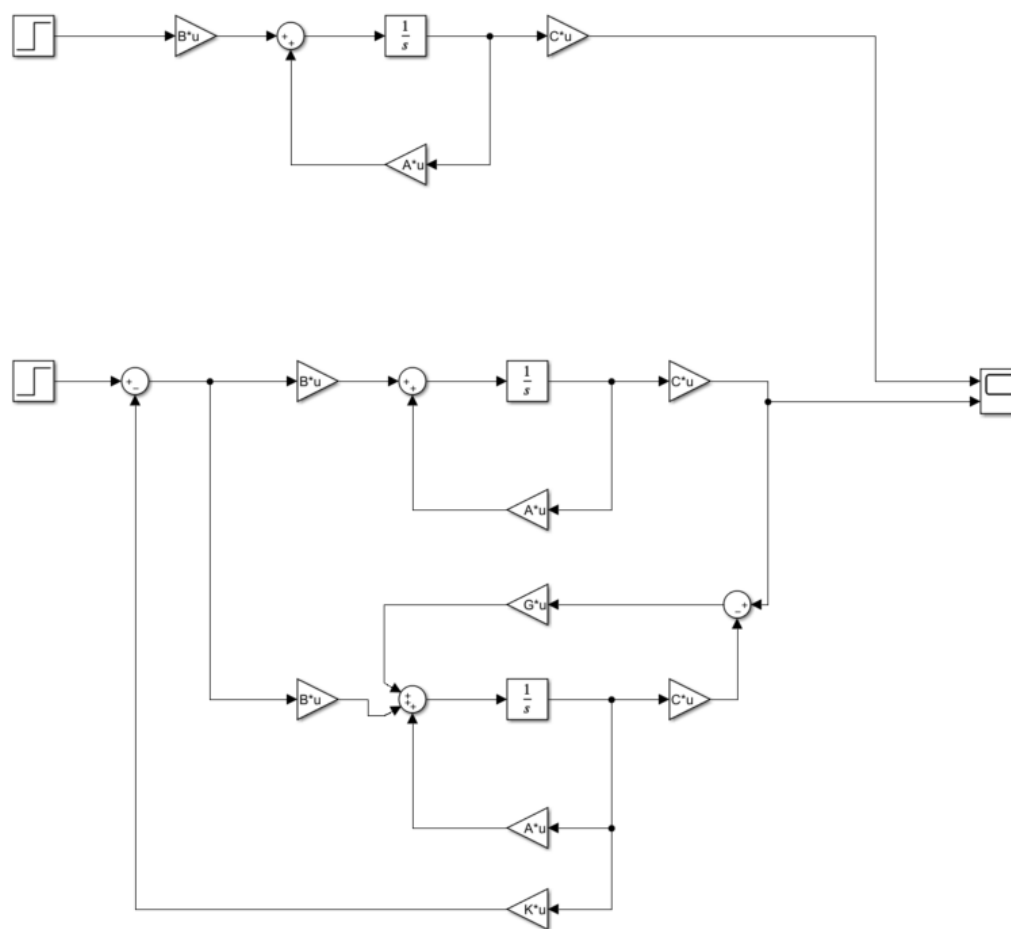


图 4.3 系统设计部分内容 1 基于状态观测器的状态反馈设计仿真平台

4.2.2.3 基于状态观测器的状态反馈与状态反馈效果对比

当两种系统的极点均配置为-4, -3, -2, -1 时, 可以对比两个系统的输出曲线来判断反馈效果, 搭建实验平台如下图所示, 其中状态观测器反馈中观测器初始状态为 $[0;0;2;1]$ 。

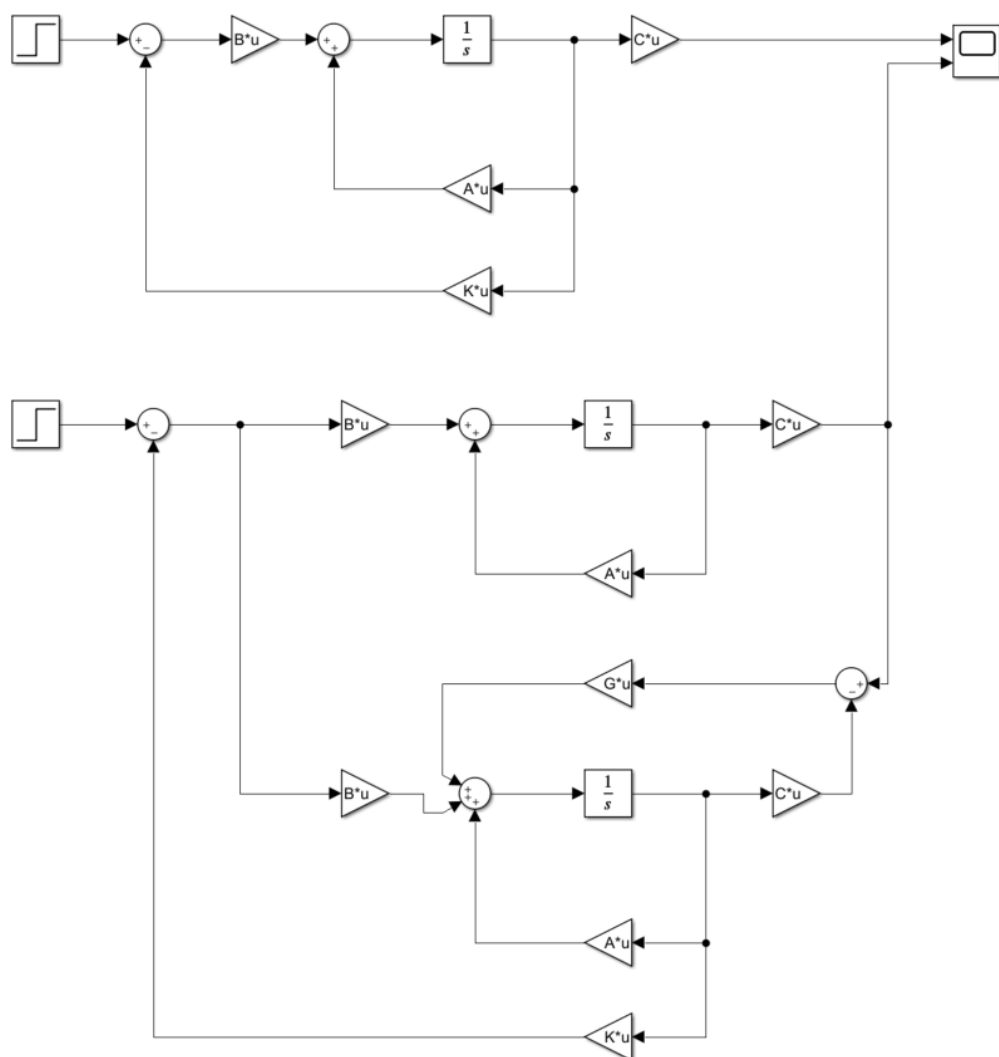


图 4.4 系统设计部分内容 1 设计系统对比仿真平台

4.2.3 实验结果

4.2.3.1 状态反馈验证结果

运行 MATLAB 代码可以得出设计的 K 矩阵为：

K = 1×4			
0.0000	0	4.0000	-0.0000

针对设计的 K 矩阵放入仿真平台可以发现原先系统为不稳定系统利用状态反馈将极点配置至左半平面后系统稳定。

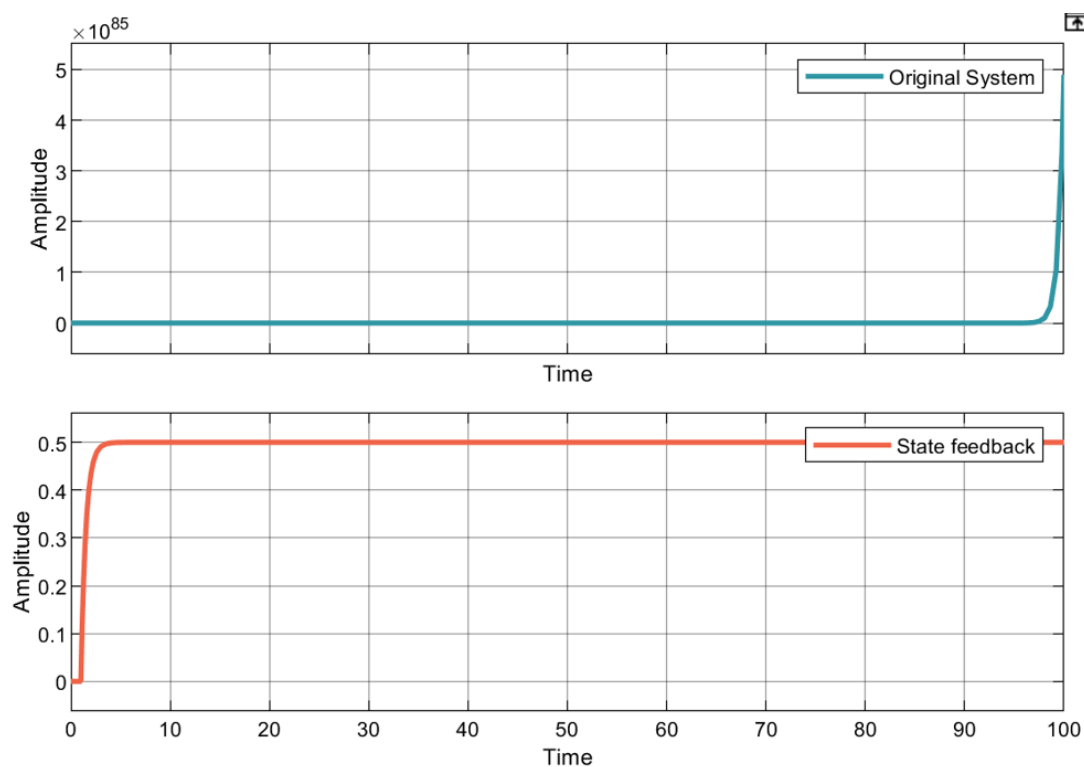


图 4.5 系统设计部分内容 1 状态反馈验证结果

4.2.3.2 基于状态观测器的状态反馈验证结果

运行 MATLAB 代码可以得出设计的 K 矩阵为：

G = 4×1

0
0
4
0

针对设计的 G 矩阵放入仿真平台可以发现所设计的状态观测器可以较好跟随系统。

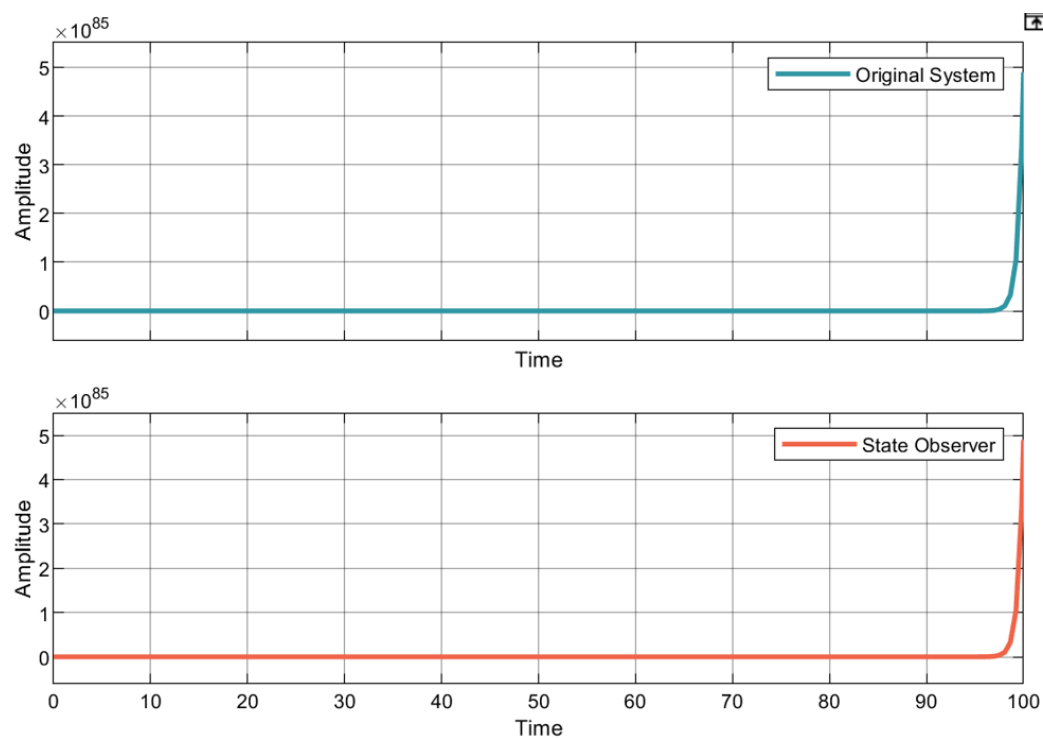


图 4.6 系统设计部分内容 1 状态观测器跟随效果验证

利用状态观测器设计状态反馈，从图中可以看出系统最终趋于稳定。

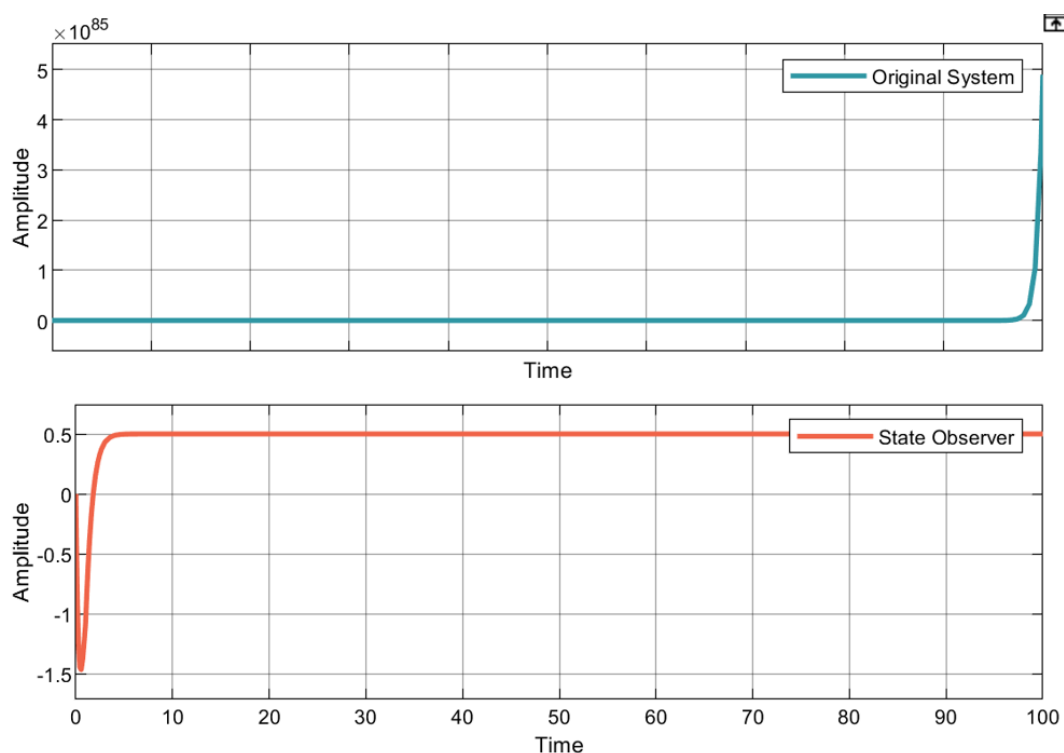


图 4.7 系统设计部分内容 1 基于状态观测器状态反馈效果验证

4.2.3.3 基于状态观测器的状态反馈与状态反馈效果对比结果

通过将两个状态反馈系统输出接入同一个示波器对比可以发现，当 $t \rightarrow \infty$ 时，

带观测器的状态反馈系统与直接状态反馈系统完全等价，但是在系统开始相应阶段由于初始条件的不同可以看出两者略有不同。

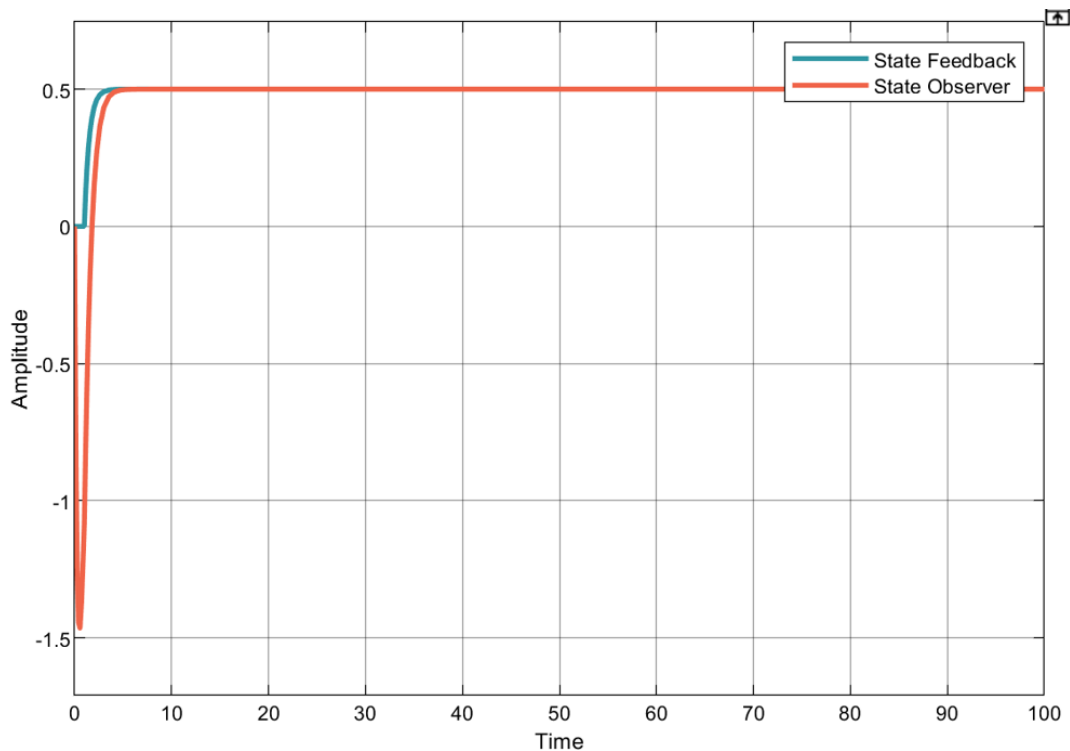


图 4.8 系统设计部分内容 1 基于状态观测器的状态反馈与状态反馈效果对比结果

4.3 小结

- 在状态反馈的设计过程中需要注意判断系统能控性，若系统存在不完全能控子系统需要判断其镇定性同时设计出来的反馈矩阵 \mathbf{K} 与书本所求存在正负相反关系，可以在 Simulink 中直接修改为负反馈（书中为正反馈）。
- 在设计状态观测器时需要判断系统的能观性，若系统存在不完全能观子系统需要判断其镇定性，在设计时需要对观测器的初始状态进行相应修改，实际使用时存在不清楚系统初始条件的情况，因此状态观测器在设置初始条件时尽可能与原系统不同才能看出最终效果。

第五章 实验总结与体会

5.1 实验总结

通过此次实验，对书本知识进行了进一步的巩固，学习了状态空间方程的建立以及转换，并在此基础上判断系统稳定性、能控性、能观性，最后针对需求对系统进行设计，包括状态反馈和状态观测器的相关部分。

除此之外也对 MATLAB 有了更加深入的使用体会，学习了 YAMIP 工具箱的基础使用方法，认识到了 MATLAB 函数与课本教学内容之间可能存在的差异问题，能够通过查阅资料自行矫正。

5.2 实验体会

现代控制理论是我比较喜欢的一门课程，精简而不简单。在学习过程中与学习过的经典控制理论相互比较，不断会有新的体悟。课程中的数学推导采用的矩阵运算也实现了对于线性代数课程所学知识的工程应用。很享受跟着老师的思维在课堂上针对每一个定理进行相关的推导，让原本难以理解的概念以数字的形式更加直观的展现在眼前。

在实验过程中，深刻体会到了实验的重要性，能够针对课程知识进行一定的复习，并在此基础上进行一定程度的拓展，尤其是在能控能观分解部分的代码编写过程中，原本其实并不了解每个分块矩阵的相对含义，在程序编程中需要对每个分块的含义非常清楚，通过实验过程也是进一步巩固了知识的掌握。

在整个课程的学习过程中，印象很深的是张老师愿意花时间在实验过程中进行现场指导，针对我们提出的问题能够立刻解答。同时，助教在每次作业批改后指出注意点，这一点确实帮助颇多。最后要感谢老师和助教一个学期的指导和帮助！！