河海大學
HOHAI UNIVERSITY

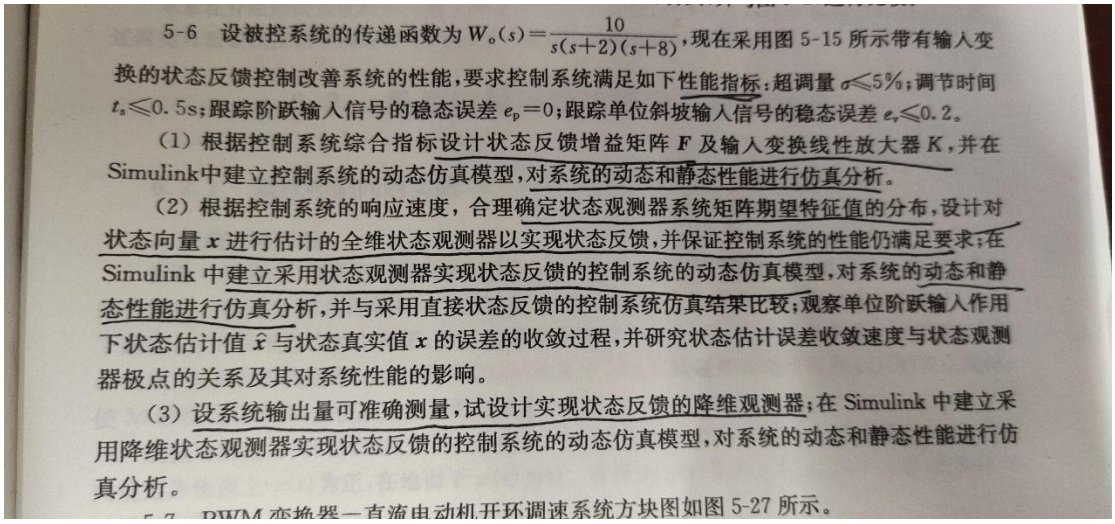
现代控制理论
上机实验报告

实验内容：	状态反馈与状态观测器
学　　院：	能源与电气学院
专　　业：	自动化
年　　级：	2019 级
学　　号：	1905010134
报 告 人：	刘晨阳
时　　间：	2022.6.20

目录

一、 题目及要求

1. 题目：



5-6 设被控系统的传递函数为 $W_o(s)=\frac{10}{s(s+2)(s+8)}$，现在采用图 5-15 所示带有输入变换的状态反馈控制改善系统的性能，要求控制系统满足如下性能指标：超调量 $\sigma\leqslant 5\%$；调节时间 $t_s\leqslant 0.5$s；跟踪阶跃输入信号的稳态误差 $e_p=0$；跟踪单位斜坡输入信号的稳态误差 $e_v\leqslant 0.2$。

（1）根据控制系统综合指标设计状态反馈增益矩阵 $\boldsymbol{F}$ 及输入变换线性放大器 K，并在 Simulink中建立控制系统的动态仿真模型，对系统的动态和静态性能进行仿真分析。

（2）根据控制系统的响应速度，合理确定状态观测器系统矩阵期望特征值的分布，设计对状态向量 $\boldsymbol{x}$ 进行估计的全维状态观测器以实现状态反馈，并保证控制系统的性能仍满足要求；在 Simulink 中建立采用状态观测器实现状态反馈的控制系统的动态仿真模型，对系统的动态和静态性能进行仿真分析，并与采用直接状态反馈的控制系统仿真结果比较；观察单位阶跃输入作用下状态估计值 $\hat{\boldsymbol{x}}$ 与状态真实值 $\boldsymbol{x}$ 的误差的收敛过程，并研究状态估计误差收敛速度与状态观测器极点的关系及其对系统性能的影响。

（3）设系统输出量可准确测量，试设计实现状态反馈的降维观测器；在 Simulink 中建立采用降维状态观测器实现状态反馈的控制系统的动态仿真模型，对系统的动态和静态性能进行仿真分析。

5-7 PWM 变换器—直流电动机开环调速系统方块图如图 5-27 所示。

图 1 题目

2. 重点：

1) 利用 Matlab/Simulink 完成，熟悉仿真软件应用。

2) 练习分析系统的动态性能和静态性能。

3) 关注极点对系统的影响。

3. 目标：

1) 尽量用 Matlab/Simulink 完成所有计算、评价、绘图工作，扩展程序的适用性，提高程序的鲁棒性。

2) 利用仿真工具多调试，多观察，多利用理论知识进行分析，完成所有问题。

3) 把分析平台做好，以后想研究某个控制问题，十分方便。

二、 第一问

4. 理论解决思路

能控性证明：

系统完全能控，是系统可以通过状态反馈任意配置闭环极点的充要条件。对于证明系统的能控性，非常常用的做法是采用“秩判据”。即计算 $Q_c = \left[B, AB, \ldots, A^{n-1}B\right]$，其中 $n = Length(A)$，如果 $Rank(Q_c) = n$ 则系统能控，否则不完全能控。

极点的选取

状态反馈的最终目的便是改变系统的闭环极点，从而改变系统的静态、动态特性。极点能起到这么大的作用就源于其与系统运动模态的对应。闭环系统的极点就是系统特征方程或者传递函数特征方程的零点。通过拉普拉斯反变化，我们很容易可以知道负实部的极点代表着指数收敛的运动模态，正实部的极点代表着指数发散的运动模态，含虚部的共轭极点代表着正弦震荡的运动模态。

据此，我认为，我们应试所有极点均具有负实部，且负实部越小越好，虚部越小越好。这样可以保证系统尽快的收敛，且过程中震荡较小。

F 的求取：

最直观的 F 求法就是待定系数并解联立方程。

设状态反馈增益矩阵 $F = [f_1 f_2 \ldots f_n]$，则闭环系统 $\sum(A - BF, B, C)$ 的特征多项式 $P_F(s) = det[sI - (A - BF)]$ 即可得到。同时根据预设极点，便可得到期望特征多项式 $p^*(s)$。令 $P_F(s) = p^*(s)$，则可求得状态反馈增益矩阵 F。

K 的求取

为了提高系统的稳态精度，基于状态空间综合法的一种简单方法就是除了按

极点配置法确定状态反馈增益矩阵 F，还引入输入变换线性放大器 K。

此时闭环传递函数为 $W_{FK}=C(SI-A+BF)^{-1}BK$ ，对单位阶跃信号的跟踪误差为 $e_{pFK}=1-C(-A+BF)^{-1}BK$ 。令误差为零，可求得 $K=B^{-1}(-A+BF)C^{-1}$.

5. 软件编程实现

本工程实现了根据极点自动解算状态反馈参数，并发送至 Simulink 自动开启仿真，并根据仿真结果绘图并分析动态静态性能。详情见附录。

```
%1. 求开环、闭环传递函数与状态空间
num = 10;
den = conv([1,0],conv([1,2],[1,8]));          %题目信息
Go = tf(num,den);                             %开环传递函数
[A,B,C,D] = tf2ss(num,den);                   %开环状态空间

Gc = feedback(Go,1);                          %闭环传递函数
[num1,den1] = tfdata(Gc,'v');
[Ac,Bc,Cc,Dc] = tf2ss(num1,den1);             %闭环状态空间
```

图 2 系统模型转换

```
function y = controlable(A,B)
    Qc = ctrb(A,B);
    ctrl = rank(Qc);
    n = length(A);
    if ctrl == n
        disp('能控');
        y = 1;
    else
        disp('不能控');
        y = 0;
    end
```

图 3 能控性判别

```
function [F,K] = FK_Get(A,B,C,D,P)
    F = acker(A,B,P);
    K = B\(-A + B*F)/C;
end
```

图 4 参数求取

```
%4. 仿真结果
    open_system('lcy56_s1');
    source_state = 1;                                          %1: 阶跃输入; 0: 斜坡输入
    set_param('lcy56_s1/Constant','Value',num2str(source_state));
    set_param('lcy56_s1/Gain10','Gain',num2str(F(1)));
    set_param('lcy56_s1/Gain11','Gain',num2str(F(2)));
    set_param('lcy56_s1/Gain12','Gain',num2str(F(3)));
    set_param('lcy56_s1/Gain13','Gain',num2str(K));
    sim('lcy56_s1',[0,6]);                    %开启仿真
    load('lcy56_1mat.mat');                    %读取仿真结果
```

图 5 开启仿真

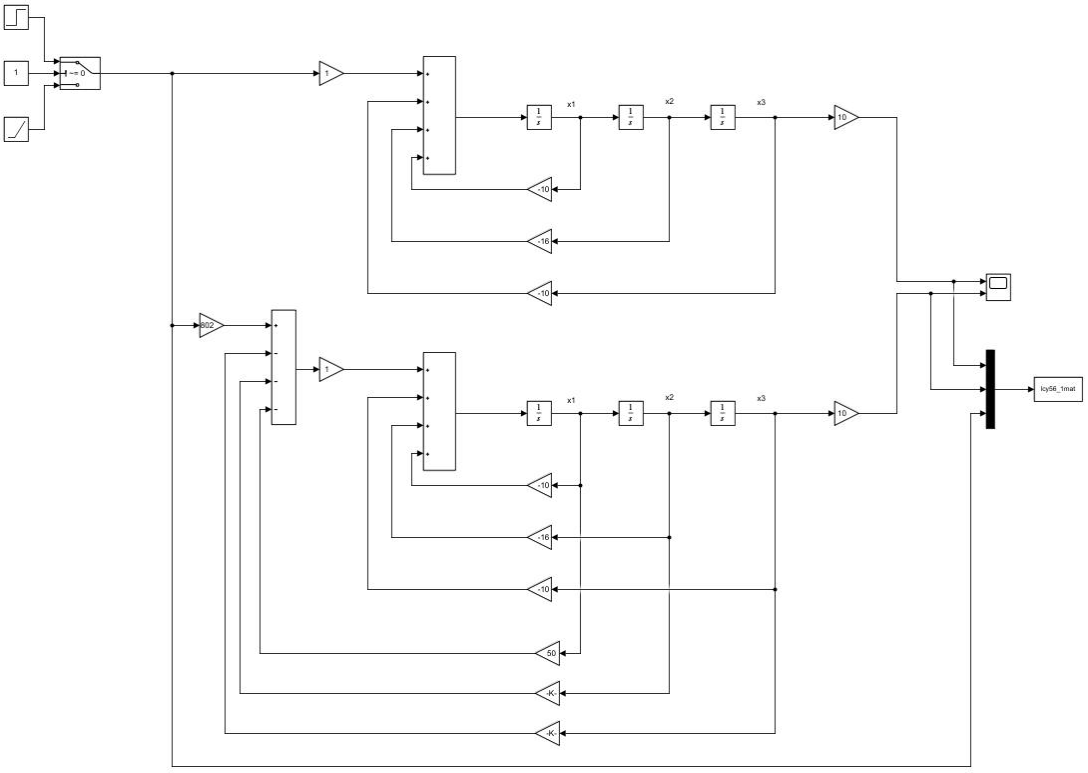


图 6 仿真模型

特性分析部分太长，详情参见附录。

6. 系统性能演示

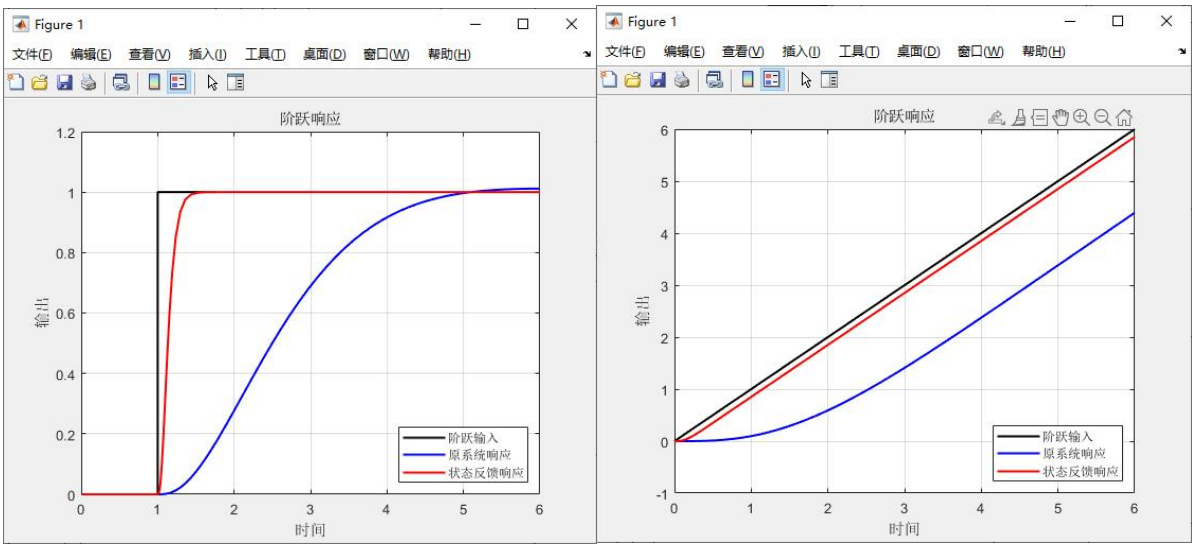


图 7 系统阶跃响应对比图

由图 7 定性可知，状态反馈大大改善了系统的静态特性和动态特性。

```
设计要求:
超调量<=0.05
调节时间<=0.5s
阶跃响应稳态误差=0
斜坡响应稳态误差<=0.2

能控
状态反馈后闭环系统极点为:
 -20.0000 + 1.0000i
 -20.0000 - 1.0000i
 -20.0000 + 0.0000i

原系统调整时间为3.295
状态反馈系统调整时间为0.361
原系统稳态误差为0.06125
状态反馈系统稳态误差为0.02415
原系统峰值时间为6.000
状态反馈系统峰值时间为4.175
原系统超调量为0.012
状态反馈系统超调量为0.000
```

命令行窗口

```
设计要求:
超调量<=0.05
调节时间<=0.5s
阶跃响应稳态误差=0
斜坡响应稳态误差<=0.2

能控
状态反馈后闭环系统极点为:
 -20.0000 + 1.0000i
 -20.0000 - 1.0000i
 -20.0000 + 0.0000i

原系统稳态误差为1.61256
状态反馈系统稳态误差为0.14975
fx >>
```

图 8 静态动态性能自动分析功能

由图 8 可知，阶跃响应时，状态反馈后系统超调量为 0.012，小于 0.05 的要求，

调节实践为 0.361s，小于 0.5s 的要求，阶跃响应稳态误差为 0，满足 0 的要求；斜坡响应时，稳态误差为 0.15，小于 0.2 的要求。

7. 系统性能分析

由于设计 K 时，我们已证明系统稳态误差必然为 0，选取含正实部极点必然导致系统不稳定，因此此处不再赘述。

我们首先讨论是否要设计共轭极点的问题。我认为，共轭极点会带来震荡，应当避免，但不少同学在选取极点时参照书中例题，选取了-1+-1.7j 的共轭极点，并令剩下的极点为非主导极点。我采用一个常见的极点配置：$-1-1.7j,-1+1.7j,-6$，进行验证。发现系统出现了超调震荡，且调整时间上升到了近 3s。我在我原本的极点配置 $-20+j,-20-j,-20$ 上做出改动：$-20+10j,-20-10j,-20$，尽管虚部变换很大，但结果变化不多。因此，我认为虚部对该系统动态性能影响不大，应着重保证负实部足够小，以获得更快的响应速度和对虚部引起的震荡的更大的抑制能力。

但我仍对共轭极点对系统的影响感到好奇，毕竟很多系统都存在该现象。我在作业后会继续探索，如还有困惑，会向老师请教。

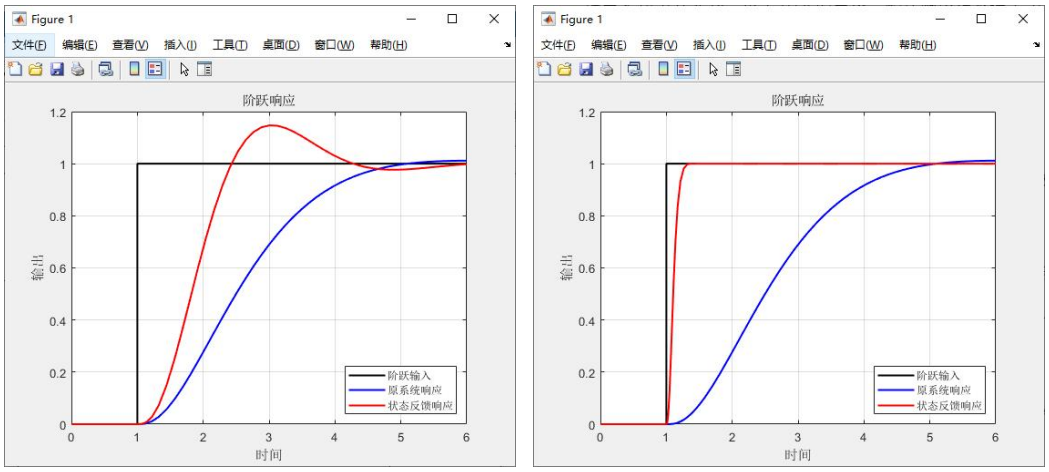


图 9 先后两次实验的阶跃响应

三、 第二问

8. 理论解决思路

分离特性

由观测器构成状态反馈的复合系统的特征值由相互独立的两部分组成:

- 直接状态反馈系统的系统矩阵 $A-BF$ 的 n 个特征值;
- 状态观测器系统矩阵 $A-GC$ 的 n 个特征值。

分离特性保证了 F 和 G 可不耦合地单独进行设计。但要注意，状态观测的收敛速度要大于控制的收敛速度，一般快 2~5 倍。

能观性证明

根据对偶性质，能观性证明与能控性证明十分类似。根据 $Q_o=\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix}$，的秩与系统矩阵的维数是否相等判断系统是否能观。

G 的求取

根据对偶性质，G 的求取和 F 的求取大同小异，在此不做赘述。

9. 软件编程实现

```
function y = observable(A,C)
    Qo = obsv(A,C);
    Rank_ob = rank(Qo);
    n = length(A);
    if Rank_ob == n
        disp('能观');
        y = 1;
    else
        disp('不能观');
        y = 0;
    end
```

图 10 能观性判别

```
function [G] = G_Get(A,B,C,D,P)
    Gt = acker(A',C',P);
    G = Gt';
end
```

图 11 G 的求取

```
%7. 状态估计分析
figure(2);
plot(state_data(1,:),state_data(2,:),'k','LineWidth',1.5);    %x1估计误差
hold on;
plot(state_data(1,:),state_data(3,:),'b','LineWidth',1.5);    %x1估计误差
hold on;
plot(state_data(1,:),state_data(4,:),'r','LineWidth',1.5);    %x1估计误差
hold on;
Plot_Client22();                                              %绘制坐标轴图例等
end
```

图 12 状态估计分析

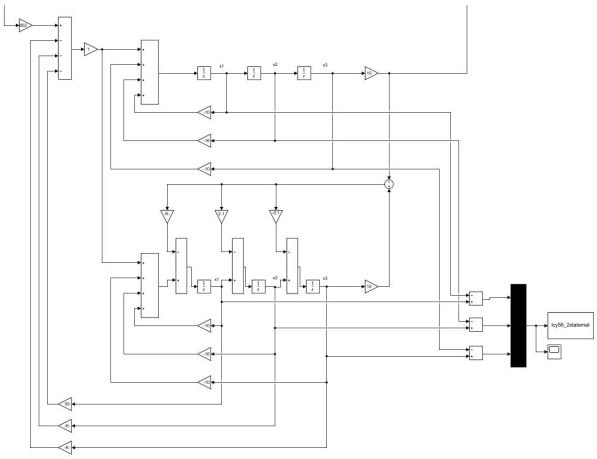


图 13 部分仿真

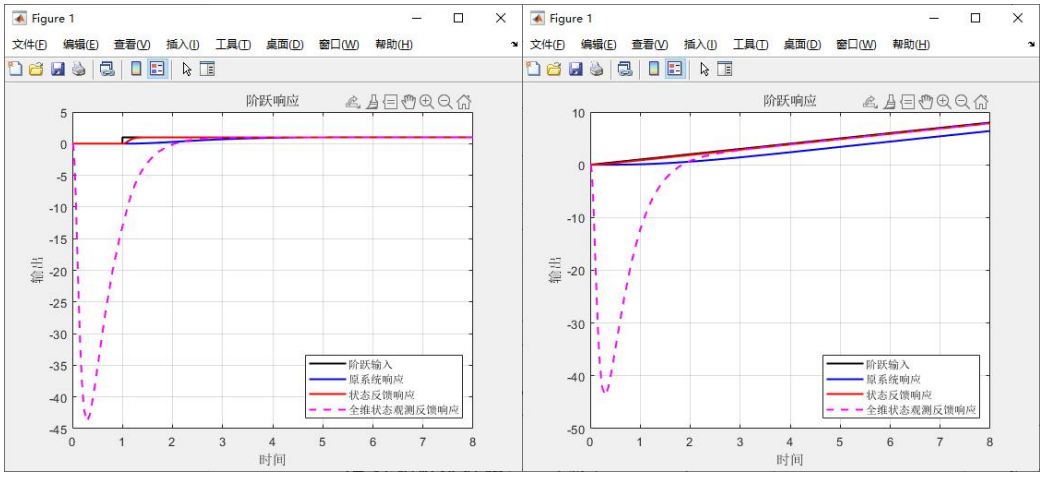


图 14 阶跃相应与斜坡响应曲线

10. 系统性能演示

由图 14 可定性看出虽然全维状态观测器下，虽然状态反馈也能无差控制，但前期偏差过大，相较直接进行状态反馈，仍有很大提升空间。主要原因在于状态观测器收敛需要时间，而我也设置了状态的初始误差（1，3，5）。

当我去掉初始初始误差，并在状态反馈极点保持为 $-20+j,-20-j,-20$ 的条件下，讲观测器极点由 $-3,-3,-3$ 改为 $-100,-100,-100$ 后，情况就非常完美了：

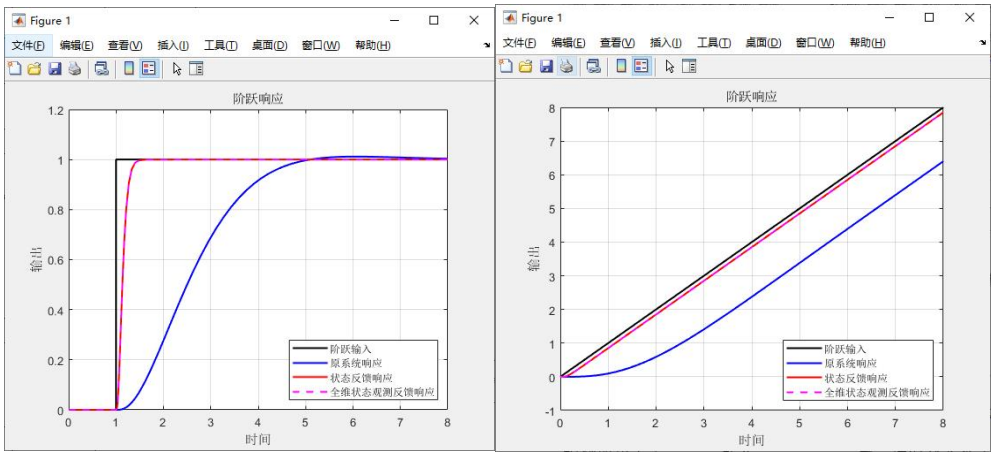


图 15 加快状态估计后的响应曲线

由图可见，全维观测器在配置得当时，效果与直接状态反馈几乎无异。

图 16 自动稳态动态分析功能

由图 16 可知，系统阶跃响应下超调量为 0，调节时间为 0.33s，稳态误差接近于 0；斜坡响应下，系统稳态误差为 0.15。均达到设定要求。

11. 系统性能分析

系统性能与直接状态反馈相近，均与反馈闭环的极点配置关系密切。极点越小，收敛越快。

但要注意的是，一定要保证状态观测器收敛要比控制器快很多，否则会出现图 14 所示情况。

本新能分析所依据数据均来自性能参数自动求取功能。对于静态特性，我主要关注稳态误差。我首先会找到响应进入 5%误差带的部分，然后求其误差平均值获得稳态误差。对于动态特性，我通过查找最大值得到超调量和峰值时间，并根据求稳态误差时记录的误差带索引，求得调整时间。具体程序详见附录。

12. 直接状态反馈比较

当状态观测器收敛不够快时，性能明显不如直接反馈。

当状态观测器收敛足够快时，性能与直接反馈相差无二。

在图 14，15 部分已做分析。

实际工程中，很多很多状态量无法直接测量或成本太高，如果能配置好全维状态观测器参数，就能实现低成本高收益的大好事。当然实际工程中，面对有些状态量测量存在干扰的情况，也可采用卡尔曼滤波等滤波算法，从数理统计的方向去考虑解决方案。

13. 误差收敛与极点关系

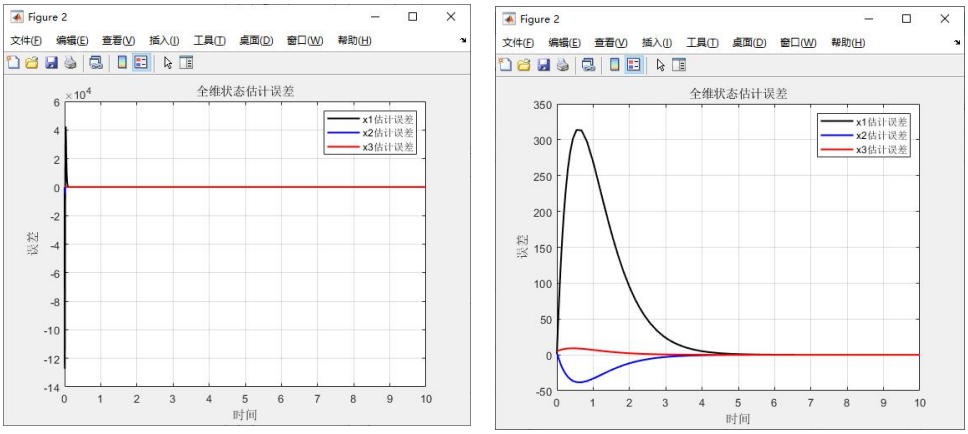


图 17 误差收敛情况对比

在观测器极点为 $-100,-100,-100$ 时，即使初始误差为 10，30，50，也能很快收敛，如图 17 左。相反，在极点为-2，-2，-2 时，即使初始误差为 1，3，5. 收敛也花费了很长时间。

因此正常的状态观测不会因存在初始偏差就丧失作用，优秀的观测，能在出现大偏差后仍能快速收敛，给控制器提供可靠信息。

四、 第三问

14. 理论解决思路

降维状态观测器的求取：

构造降维状态观测器的意义在于：一些状态可由输出反映，不必消耗额外资源。这种情况下，可以对没被输出反应的状态针对性设计观测器，降低系统成本。

- 首先要计算降维观测器的观测维数：能观性判别矩阵维数减去输出矩阵维数。
- 然后构造线性非奇异变换矩阵，对系统进行分解，分解出需要观测的子系统。

- 对此子系统设计全维状态观测器。
- 最后还原之前的线性非奇异变换。

15. 软件编程实现

```
T = [1,0,0;0,1,0;0,0,10];
INVT = inv(T);
Ad = T*Ac/T;
Bd = T*Bc;
Cd = Cc/T;
AD = mat2cell(Ad,[2,1],[2,1]);
Ad11 = AD{1,1};Ad12 = AD{1,2};Ad21 = AD{2,1};Ad22 = AD{2,2};
Bd1 = Bd(1:2,1);Bd2 = Bd(3,1);
Gd = Gd_Get(Ad11,Ad21,Pd);
Aw = Ad11 - Gd*Ad21;
Bu = Bd1 - Gd*Bd2;
By = (Ad11 - Gd*Ad21)*Gd + Ad12 - Gd*Ad22;
Cw = [1;1];
Dy = Gd;
DY = [Dy;1];
CW = [Cw;0];
RemidyY = T\DY;
RemidvW = T\CW;
```

```
function [Gd] = Gd_Get(A11,A21,P)
    Gt = acker(A11',A21',P);
    Gd = Gt';
end
```

图 18 降维观测器增益矩阵求取

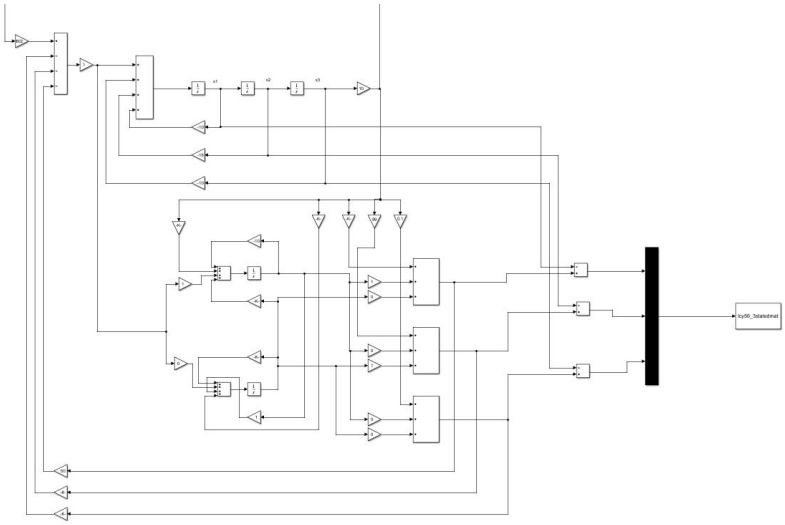


图 19 部分仿真

16. 系统性能演示

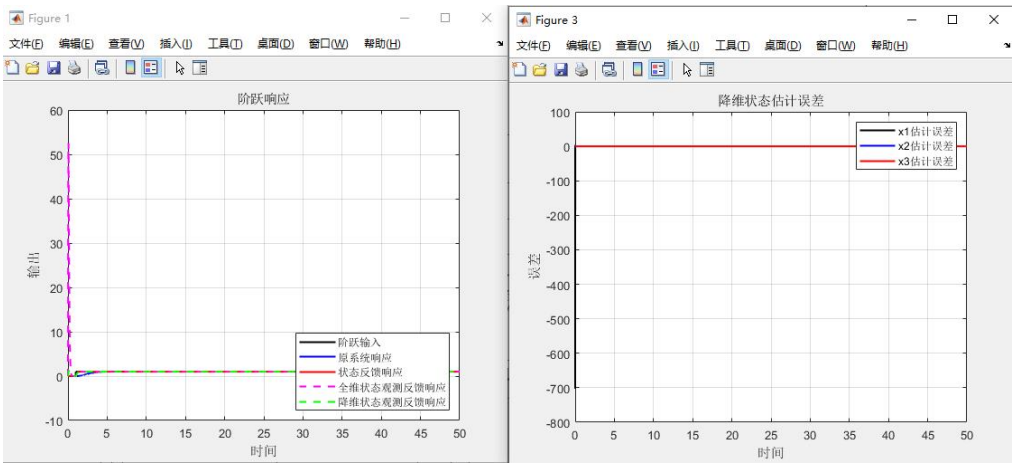


图 20 阶跃响应及状态估计

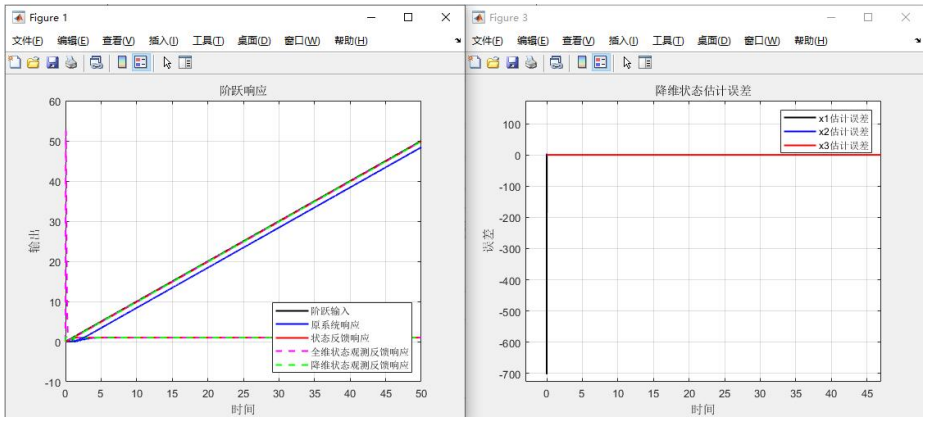


图 21 单位斜坡响应及状态估计

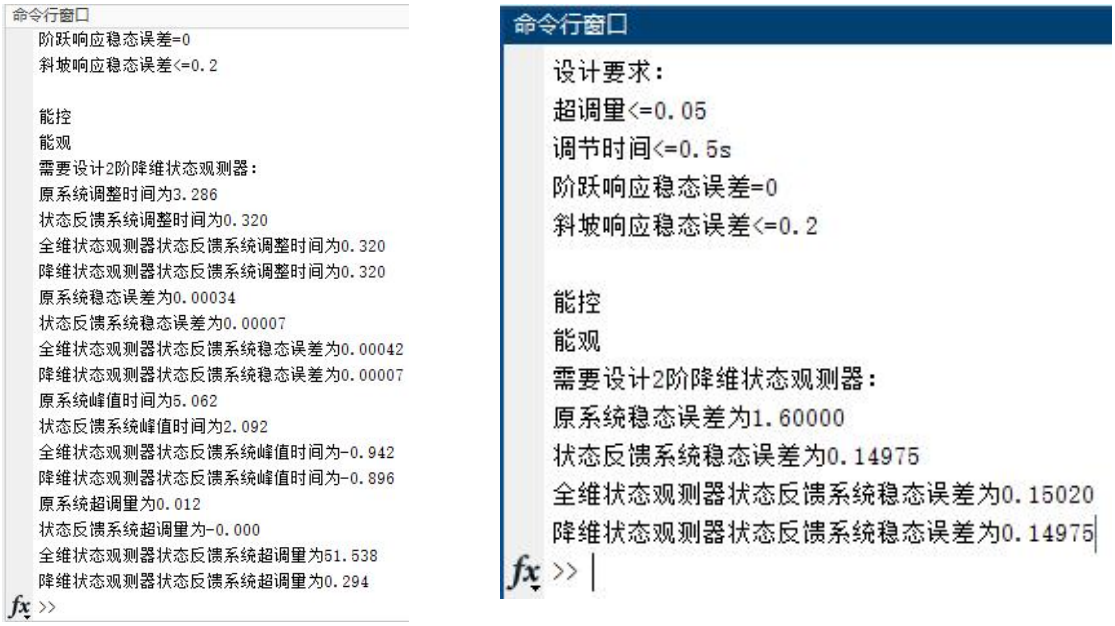


图 22 性能指标自动计算功能

由图 22 数据知，系统新能均已达到设定指标。

17. 系统性能分析

降维观测器和全维观测器一样，必须要收敛的足够快才能为控制器提供有效信息。

这里我尝试一个零极点，看会给系统带来多大影响：

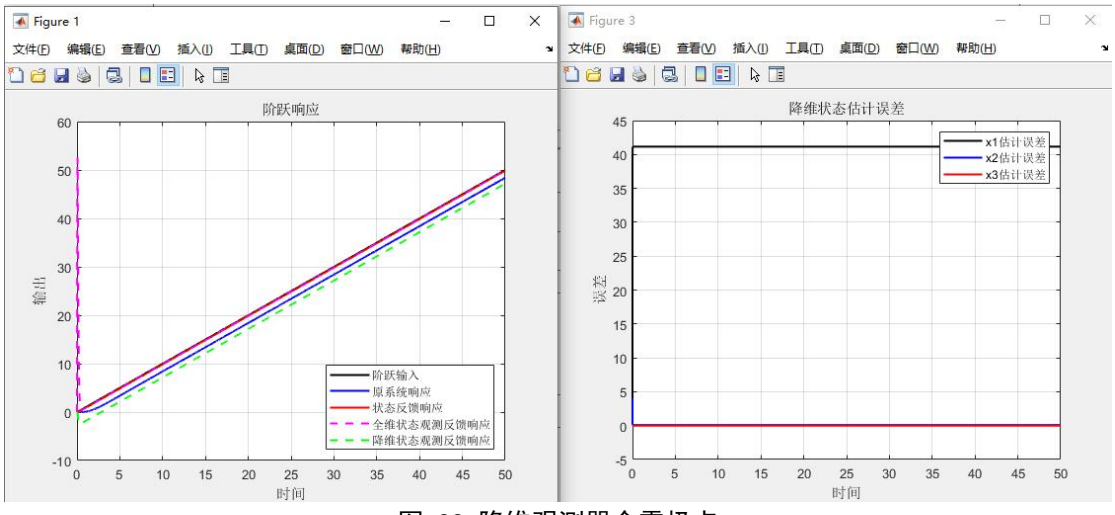


图 23 降维观测器含零极点

由图 23 知，零极点，导致了状态观测的不收敛，从而导致了稳态误差的存在。

如果存在干扰，系统甚至可能不稳定。

五、 收获及待解决问题

18. 收获

最大的收获还是熟悉了 matlab/simulink 使用。之前一直很苦恼 simulink 不好处理仿真数据，而且一些参数手动修改十分麻烦。这次我画了很大力气做好了 matlab，simulink 的联动。让我能够给他最简单的指令后，他 matlab 就能直接完成各项参数的解算，并直接送给 simulink 仿真，并将仿真数据传回给 matlab 进行进一步解析和显示。我相信这对于我后面希望进一步探究控制理论有很大帮助。

还有一个收获就是通过调参数，对于极点对系统的影响有了更深刻的认识。

19. 问题

共轭极点的意义

二阶最小阻尼比的意义和参考价值

极点真的是越小越好吗

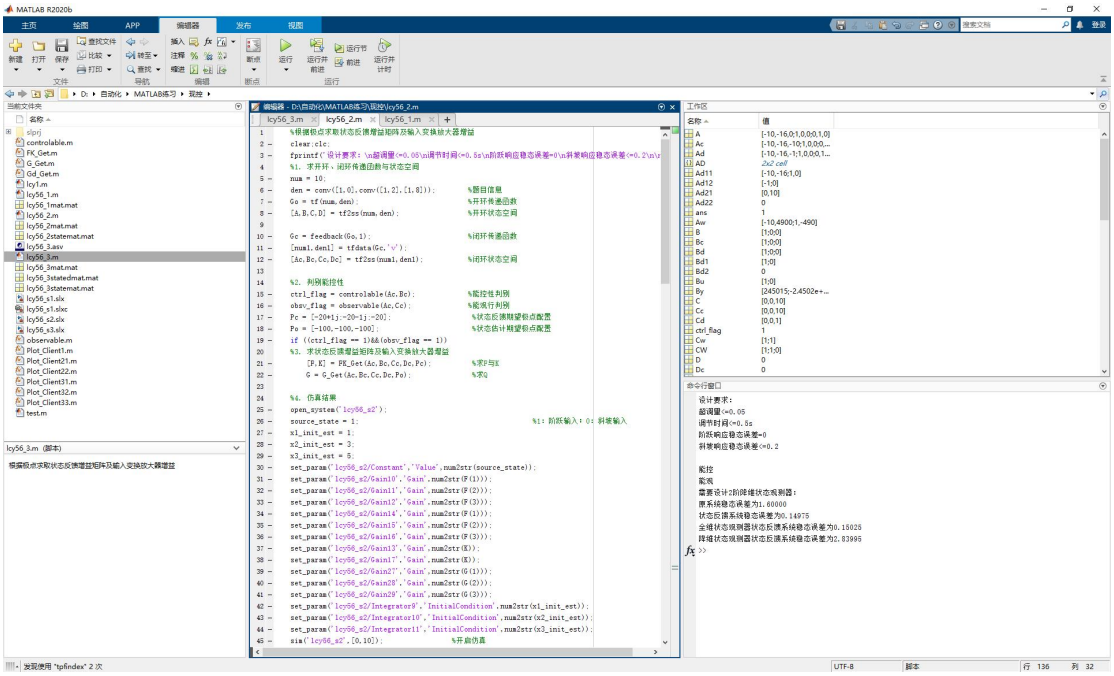
对于这些问题我会继续探索

六、 附录

三道题的完整工程（含.m,.slx.mat）放在如下网址：

https://app.mediatrack.cn/player/1538664167253213184

工程概览：



第二题 matlab 程序:

```
clear;clc;
num = 10;
den = conv([1,0],conv([1,2],[1,8]));        %ÌâÄ¿ÐÅÏ¢
Go = tf(num,den);                          %¿ª»·´«µÝº¯Êý
[A,B,C,D] = tf2ss(num,den);                 %¿ª»·×´Ì¬¿Õ¼ä

Gc = feedback(Go,1);                       %±Õ»·´«µÝº¯Êý
[num1,den1] = tfdata(Gc,'v');
[Ac,Bc,Cc,Dc] = tf2ss(num1,den1);           %±Õ»·×´Ì¬¿Õ¼ä

ctrl_flag = controlable(Ac,Bc);            %ÄÜ¿ØÐÔÅÐ±ð
obsv_flag = observable(Ac,Cc);             %ÄÜ¹ÛÐÔÅÐ±ð
Pc = [-20+1j;-20-1j;-20];                   %×´Ì¬·´À¡ÆÚÍû¼«µãÅäÖÃ
Po = [-100,-100,-100];                      %×´Ì¬¹À¼ÆÆÚÍû¼«µãÅäÖÃ
if ((ctrl_flag == 1)&&(obsv_flag == 1))
    [F,K] = FK_Get(Ac,Bc,Cc,Dc,Pc);          %ÇóFÓëK
    G = G_Get(Ac,Bc,Cc,Dc,Po);               %ÇóQ

open_system('lcy56_s2');
source_state =
1;                                         %1£º¼×Ô¼ÊäÈë£»0£ºÐ±ÆÂÊäÈë
x1_init_est = 1;
x2_init_est = 3;
x3_init_est = 5;
set_param('lcy56_s2/Constant','Value',num2str(source_state));
```

```
set_param('lcy56_s2/Gain10','Gain',num2str(F(1)));
set_param('lcy56_s2/Gain11','Gain',num2str(F(2)));
set_param('lcy56_s2/Gain12','Gain',num2str(F(3)));
set_param('lcy56_s2/Gain14','Gain',num2str(F(1)));
set_param('lcy56_s2/Gain15','Gain',num2str(F(2)));
set_param('lcy56_s2/Gain16','Gain',num2str(F(3)));
set_param('lcy56_s2/Gain13','Gain',num2str(K));
set_param('lcy56_s2/Gain17','Gain',num2str(K));
set_param('lcy56_s2/Gain27','Gain',num2str(G(1)));
set_param('lcy56_s2/Gain28','Gain',num2str(G(2)));
set_param('lcy56_s2/Gain29','Gain',num2str(G(3)));
set_param('lcy56_s2/Integrator9','InitialCondition',num2str(x1_init_est));
set_param('lcy56_s2/Integrator10','InitialCondition',num2str(x2_init_est));
set_param('lcy56_s2/Integrator11','InitialCondition',num2str(x3_init_est));
sim('lcy56_s2',[0,10]);                %¿ªÆô·ÂÕæ
load('lcy56_2mat.mat');                %¶ÁÈ¡·ÂÕæ½á¹û
load('lcy56_2statemat.mat');                %¶ÁÈ¡·ÂÕæ½á¹û
figure(1);
plot(step_data(1,:),step_data(4,:),'k','LineWidth',1.5);    %½×Ô¾ÊäÈë
hold on;
plot(step_data(1,:),step_data(2,:),'b','LineWidth',1.5);    %Ô-ÏµÍ³½×
hold on;
plot(step_data(1,:),step_data(3,:),'r','LineWidth',1.5);    %×´Ì¬·´À¡
hold on;
plot(step_data(1,:),step_data(5,:),'m--','LineWidth',1.5);    %È«Î¬×´Ì¬¹Û²â
Æ÷×´Ì¬·´À¡ÏµÍ³½×Ô¾ÏìÓ¦
hold on;
Plot_Client21();                                %»æÖÆ×ø±êÖáÍ¼À

    Total_length = length(step_data);                    %ÐòÁÐ³¤¶È
    Total_time = step_data(1,Total_length);                    %×ÜÊ±³¤

    if source_state == 1
        stable_error = 1*0.05;

        stable_index_c = Total_length;
        stable_index_f = Total_length;
        stable_index_gf = Total_length;
        for i = 2:1:Total_length
            tempc_now = step_data(2,i);
            tempc_last = step_data(2,i-1);
            tempf_now = step_data(3,i);
            tempf_last = step_data(3,i-1);
            tempgf_now = step_data(5,i);
```

```
            tempgf_last = step_data(5,i-1);
            if
((abs(tempc_now-1)<=stable_error)&&(abs(tempc_last-1)>stable_error))
                stable_index_c = i;
                stable_time_c = step_data(1,stable_index_c)-1;
            end
            if
((abs(tempf_now-1)<=stable_error)&&(abs(tempf_last-1)>stable_error))
                stable_index_f = i;
                stable_time_f = step_data(1,stable_index_f)-1;
            end
            if
((abs(tempgf_now-1)<=stable_error)&&(abs(tempgf_last-1)>stable_error))
                stable_index_gf = i;
                stable_time_gf = step_data(1,stable_index_gf)-1;
            end
        end
        fprintf('Ô-ÏµÍ³µ÷ÕûÊ±¼äÎª%.3f\n',stable_time_c);
        fprintf('×´Ì¬·´À¡ÏµÍ³µ÷ÕûÊ±¼äÎª%.3f\n',stable_time_f);

fprintf('È«Î¬×´Ì¬¹Û²âÆ÷×´Ì¬·´À¡ÏµÍ³µ÷ÕûÊ±¼äÎª%.3f\n',stable_time_gf);

        sum_c = 0;
        sum_f = 0;
        sum_gf = 0;
        for i = stable_index_c:1:Total_length
            sum_c = sum_c + step_data(2,i);
        end
        stable_value_c = sum_c/(Total_length-stable_index_c);
        stable_error_c = abs(stable_value_c - 1);
        for i = stable_index_f:1:Total_length
            sum_f = sum_f + step_data(3,i);
        end
        stable_value_f = sum_f/(Total_length-stable_index_f);
        stable_error_f = abs(stable_value_f - 1);
        for i = stable_index_gf:1:Total_length
            sum_gf = sum_gf + step_data(4,i);
        end
        stable_value_gf = sum_gf/(Total_length-stable_index_gf);
        stable_error_gf = abs(stable_value_gf - 1);
    else
        error_sum_c = 0;
        error_sum_f = 0;
        error_sum_gf = 0;
```

```
        for i = Total_length-9:1:Total_length
            error_now_c = step_data(4,i) - step_data(2,i);
            error_now_f = step_data(4,i) - step_data(3,i);
            error_now_gf = step_data(4,i) - step_data(5,i);
            error_sum_c = error_sum_c + error_now_c;
            error_sum_f = error_sum_f + error_now_f;
            error_sum_gf = error_sum_gf + error_now_gf;
        end
        stable_error_c = error_sum_c/10;
        stable_error_f = error_sum_f/10;
        stable_error_gf = error_sum_gf/10;
    end
    fprintf('Ô-ÏµÍ³ÎÈÌ¬Îó²îÎª%.5f\n',stable_error_c);
    fprintf('×´Ì¬·´À¡ÏµÍ³ÎÈÌ¬Îó²îÎª%.5f\n',stable_error_f);
    fprintf('È«Î¬×´Ì¬¹Û²âÆ÷×´Ì¬·´À¡ÏµÍ³ÎÈÌ¬Îó²îÎª%.5f\n',stable_error_gf);

    if source_state == 1
        [Mpc,tpcindex] = max(step_data(2,:));
        tpc = step_data(1,tpcindex);
        [Mpf,tpfindex] = max(step_data(3,:));
        tpf = step_data(1,tpfindex);
        [Mpgf,tpgfindex] = max(step_data(5,:));
        tpgf = step_data(1,tpgfindex);
        fprintf('Ô-ÏµÍ³·åÖµÊ±¼äÎª%.3f\n',tpc-1);
        fprintf('×´Ì¬·´À¡ÏµÍ³·åÖµÊ±¼äÎª%.3f\n',tpf-1);
        fprintf('È«Î¬×´Ì¬¹Û²âÆ÷×´Ì¬·´À¡ÏµÍ³·åÖµÊ±¼äÎª%.3f\n',tpgf-1);
        fprintf('Ô-ÏµÍ³³¬µ÷Á¿Îª%.3f\n',Mpc-1);
        fprintf('×´Ì¬·´À¡ÏµÍ³³¬µ÷Á¿Îª%.3f\n',Mpf-1);
        fprintf('È«Î¬×´Ì¬¹Û²âÆ÷×´Ì¬·´À¡ÏµÍ³³¬µ÷Á¿Îª%.3f\n',Mpgf-1);
    end

figure(2);
plot(state_data(1,:),state_data(2,:),'k','LineWidth',1.5);    %x1¹À¼Æ
hold on;
plot(state_data(1,:),state_data(3,:),'b','LineWidth',1.5);    %x1¹À¼Æ
hold on;
plot(state_data(1,:),state_data(4,:),'r','LineWidth',1.5);    %x1¹À¼Æ
hold on;
Plot_Client22();                                              %»æÖÆ×ø±êÖáÍ¼À
end
```