

《控制系统仿真》

MATLAB课程设计报告



学院： 自动化学院

专业： 自动化专业

班级： 2016211404

姓名： 张晓媛

学号： 2016211780

2019年1月5日

控制系统仿真大作业

摘 要

本控制系统仿真大作业主要利用MatLab和Simulink对控制系统进行数学建模与性能分析，包括MatLab中的数据定义，文件读取与写入，函数建立，图像绘制；Simulink中子系统封装；PID控制器参数整定，运用MATLAB程序设计和Simulink仿真，可视校正PID参数，通过曲线观察调整结果。

关键词 MatLab Simulink 子系统封装 PID

**目 录**

[第一章 基本操作 1](#_Toc534464689)

[1.1数据定义 1](#_Toc534464690)

[1.2数据读取与绘图 2](#_Toc534464691)

[1.3函数构建与绘图 2](#_Toc534464692)

[1.4系统函数构建及其Bode图绘制 4](#_Toc534464693)

[第二章 子系统封装 5](#_Toc534464694)

[2.1用Simulink 建立系统 5](#_Toc534464695)

[2.2观察输入输出波形 6](#_Toc534464696)

[2.3 数据导出 7](#_Toc534464697)

[第三章 PID控制器参数整定 9](#_Toc534464698)

[3.1工程整定法确定参数 9](#_Toc534464699)

[3.2模型仿真 13](#_Toc534464700)

[3.2.1 P控制器 13](#_Toc534464701)

[3.2.2 PI控制器 14](#_Toc534464702)

[3.2.3 PID控制器 15](#_Toc534464703)

[3.3 M文件设计及性能指标求取 17](#_Toc534464704)

[第四章 总结 18](#_Toc534464705)

[第五章 参考文献 19](#_Toc534464706)

**第一章 基本操作**

1.1数据定义

定义一组数组，第一列为时间t（t为等差数列，）；第二列为与t对应的201个幅值数据，作为信号幅值；第三列为按s的降幂排列的传递函数分子系数；第四列为按s的降幂排列的分母系数。第三列、第四列的数据不能超过5个。并将所有数据保存到文件data.dat。

生成四行数据逐行写入：第一行数据，由公差为1的等差数列构成的一维数组；tp表示幅值，使用random()函数生成泊松分布的一组伪随机数；num、den分别保存传递函数的分子、分母系数。将四组数据写进data.dat文件中

MatLab代码：

t = 0:1:200;

tp=random('poisson',0:200,1,201);

num=[1 3 4 5];

den=[2 3 5 3 6];

fp = fopen('E:\控制系统与仿真\780\1\data.dat','wt');

fprintf(fp, '%4d', t);

fprintf(fp,'\n');

fprintf(fp, '%4d', tp);

fprintf(fp,'\n');

fprintf(fp,'%4d',num);

fprintf(fp,'\n');

fprintf(fp,'%4d',den);

fclose(fp)

生成的date.dat文件部分截图如图1所示。

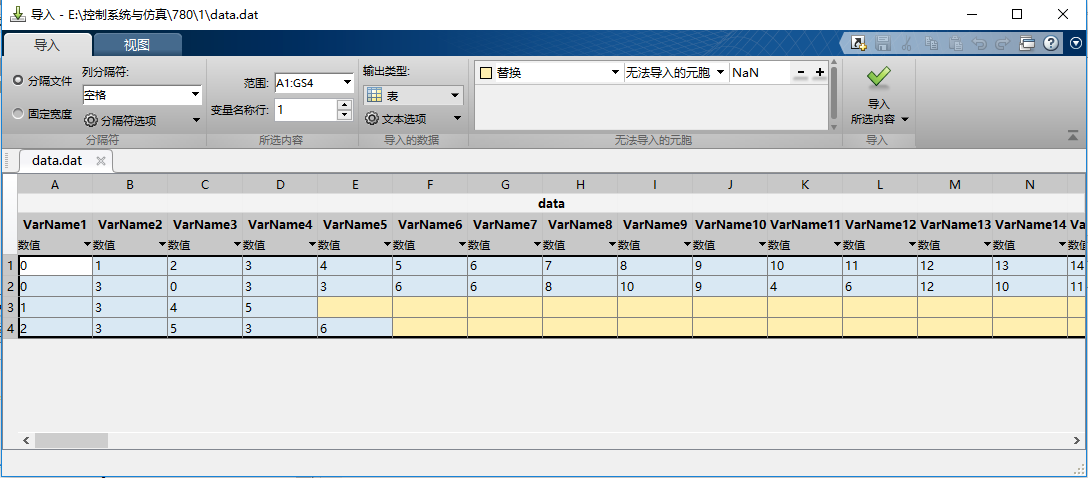


图1 data.dat文件部分截图

1.2数据读取与绘图

读入data.dat数据，画出的时域波形。

分别读出data.dat文件中的第一行和第二行数据（t和幅值），利用plot()函数绘制出f1-t曲线图。

MatLab代码：

fp=fopen('E:\控制系统与仿真\780\1\data.dat','rt');

t=fscanf(fp,'%d',201); %读第一行

f1=fscanf(fp,'%f',201); %读第二行

fclose(fp);

plot(t,f1);

title('f1-t曲线图');

xlabel('t');

ylabel('f1');

画出的f1-t时域波形图如图2所示。

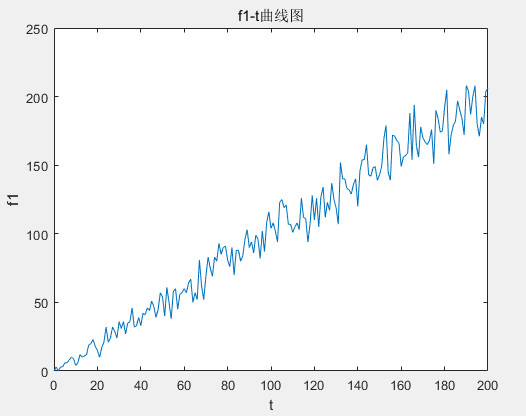


图2 f1(t)-t曲线图

1.3函数构建与绘图

求取，将结果保存到result.mat文件，画出其时域波形。

利用循环求f2，plot()函数绘制f2(t)-t曲线图。

MatLab代码：

for n=1:201

if n <10

f2(n)=2\*f1(n);

elseif (n<=100)&(abs(f1(n))<15)

f2(n)=f1(n)^2;

else

f2(n)=f1(n-3);

end

end

plot(t,f2)

title('f2-t曲线图');

xlabel('t');

ylabel('f2(t)');

fp=fopen('E:\控制系统与仿真\780\1\result.mat','wt');

fprintf(fp,'%4d',f2);

fclose(fp)

画出的f1-t时域波形图如图3所示。

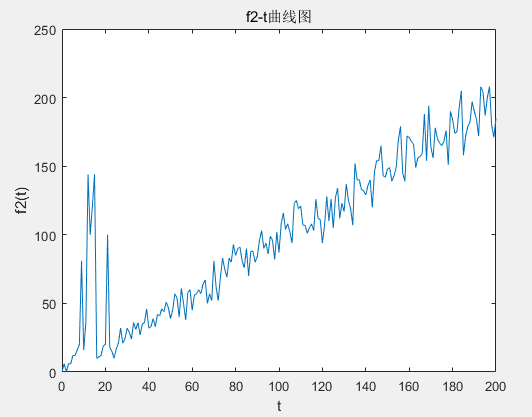


图3 f2(t)-t曲线图

1.4系统函数构建及其Bode图绘制

按 data.dat中的第三列、第四列，求取其对应的传递函数，绘制其bode图。

打开data.dat文件，由于fscanf()函数不能直接读取第三第四行，故先读出前两行402个数据给f，再读出传递函数的分子分母。求出对应的传递函数Gs，绘制出bode图。

MatLab代码：

fp = fopen('E:\控制系统与仿真\780\1\data.dat','rt');

f=fscanf(fp,'%f',402);

num=fscanf(fp,'%f',4);

den=fscanf(fp,'%f',5);

fclose(fp);

num1=ctranspose(num);

den1=ctranspose(den);

Gs=tf(num1,den1)

bode(Gs)

输出传递函数为：

3 s^4 + 5 s^3 + 7 s^2 + 4 s + 2

sys = -------------------------------

2 s^4 + 7 s^3 + 9 s^2 + 1

画出的bode图如图4所示：

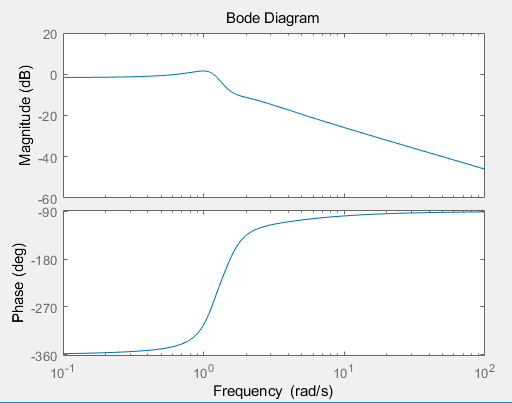


图4 Bode图

第二章 子系统封装

2.1用Simulink 建立系统

建立系统方程， 其中x 为输入，y为输出，a,b,c。

常数。并对该系统进行封装，且能通过对话框修改a, b, c的值。

Simulink建立做给方程系统如图5 所示：

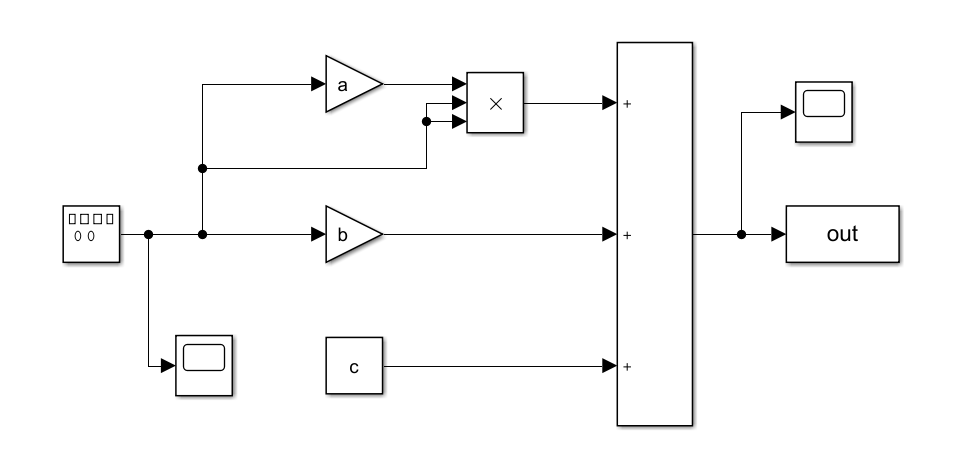


图5 Simulink子系统模型

封装后系统模型如图6所示：

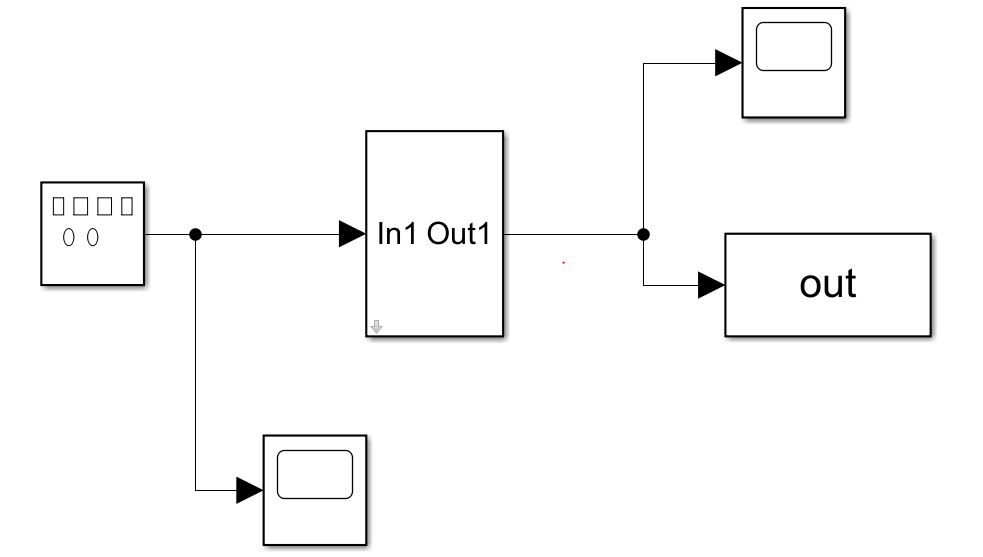


图6 封装后系统模型

设置动态参数可调的界面如图7所示：

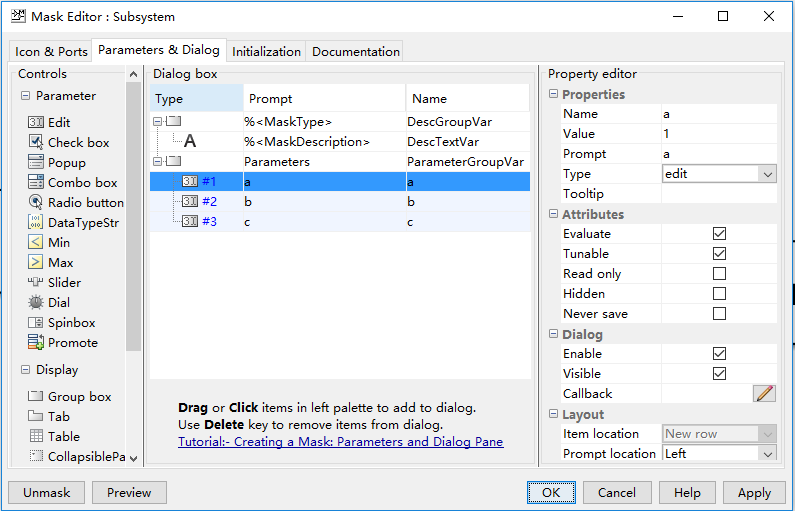


图7 设置子系统参数

双击封装子系统弹出对话框修改参数a、b、c，修改子系统参数界面如图8所示：

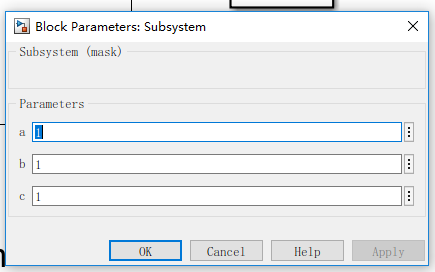


图8 修改子系统参数界面

2.2观察输入输出波形

输入x为幅度为5、频率为0.25Hz的锯齿波，采用示波器显示输出y及输入x的波形。

双击输入框，设置输入波形参数，如图9所示：

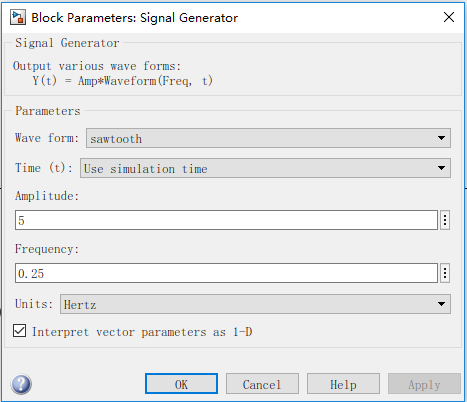


图9 输入波形参数设置界面

双击示波器模块，观察输入输出波形，输入波形如图10所示，输出波形如图11所示：

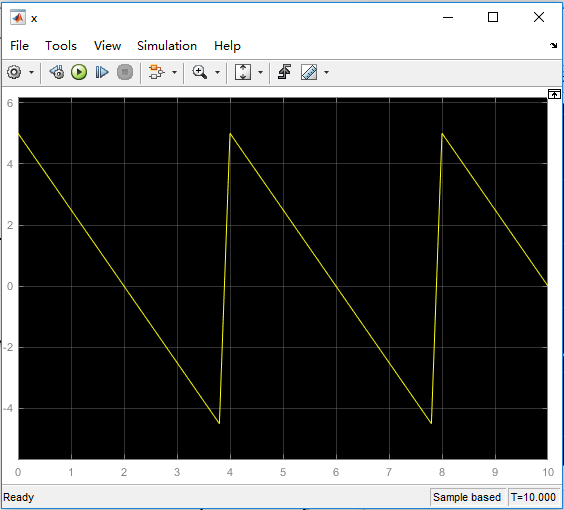


图10 输入锯齿波波形图

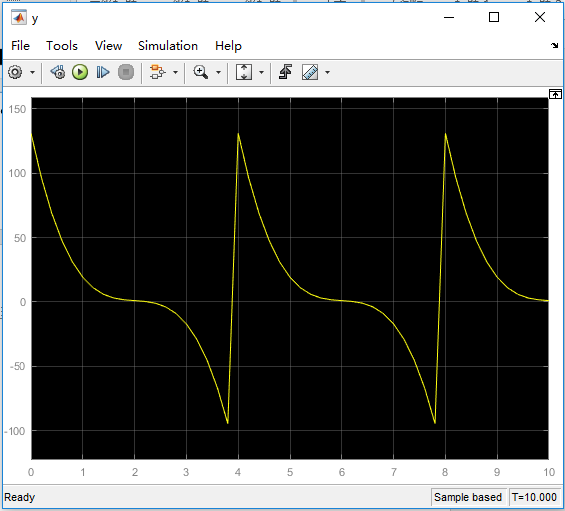


图11 输出波形图

2.3 数据导出

将输入x、输出y，导入到工作空间，并采用plot命令，将两个波形在同一波形窗口显示，带网格线、图例。

数据可通过simout组件返回到工作空间，将输入输出均返回工作空间，如图12所示：

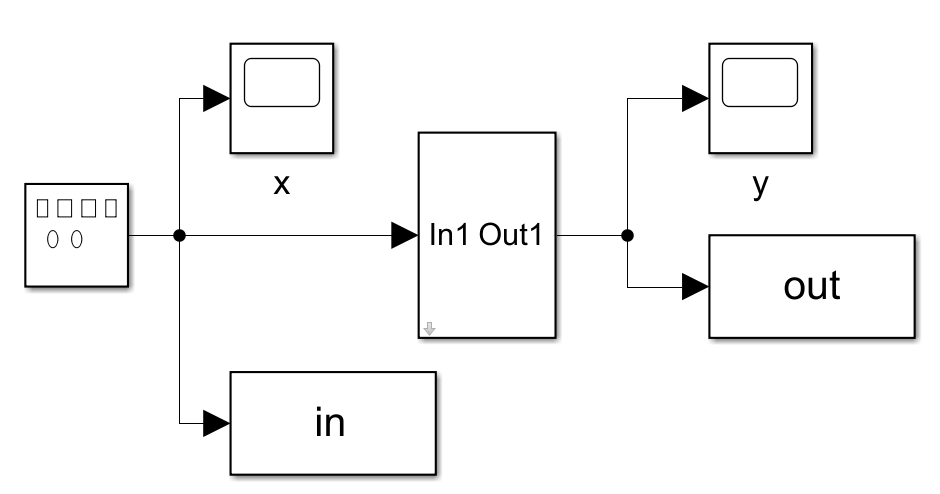


图12 数据通过simout组件返回工作空间

MatLab代码：

x=in;

y=out;

plot(x,'r');

hold on;

plot(y,'k')

hold on;

text(1,50,'y');

text(0.5,-5,'x');

运行后输出结果曲线如图13所示：

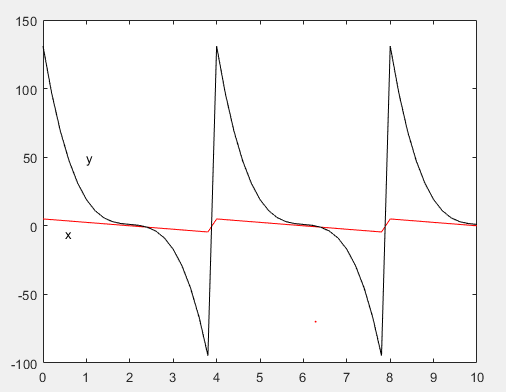


图13 输入输出曲线图

导出数据x，y如图14，15所示：

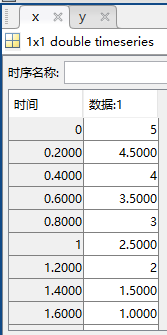


图14 导出的x部分数据

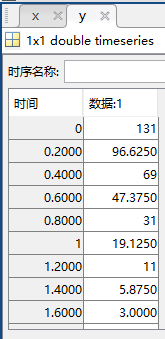


图15 导出的y部分数据

第三章 PID控制器参数整定

3.1工程整定法确定参数

某控制系统的开环传递函数为，采用单位负反馈。利用工程整定法设置合适的P，PI，PID 控制器参数。

Simulink建立原系统模型，如图16所示：

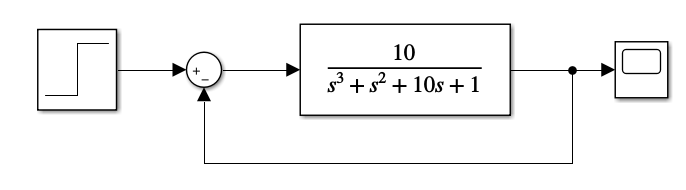


图 16 原系统模型

运行，观察阶跃响应图像曲线，可知该系统不稳定，输出波形图像如图17所示：

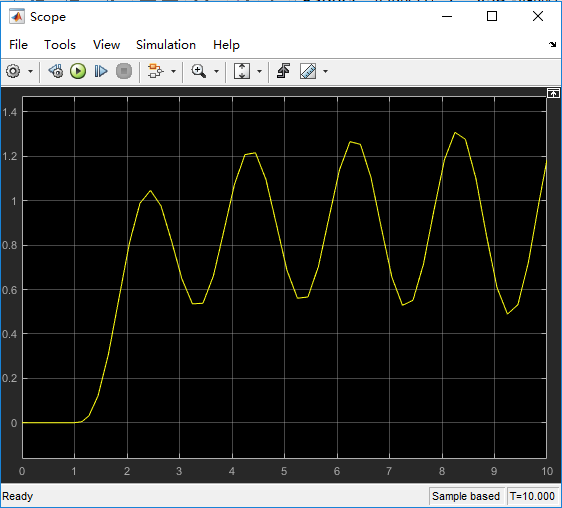


图17 原系统阶跃响应波形图

故引入PID控制器，如图18所示：

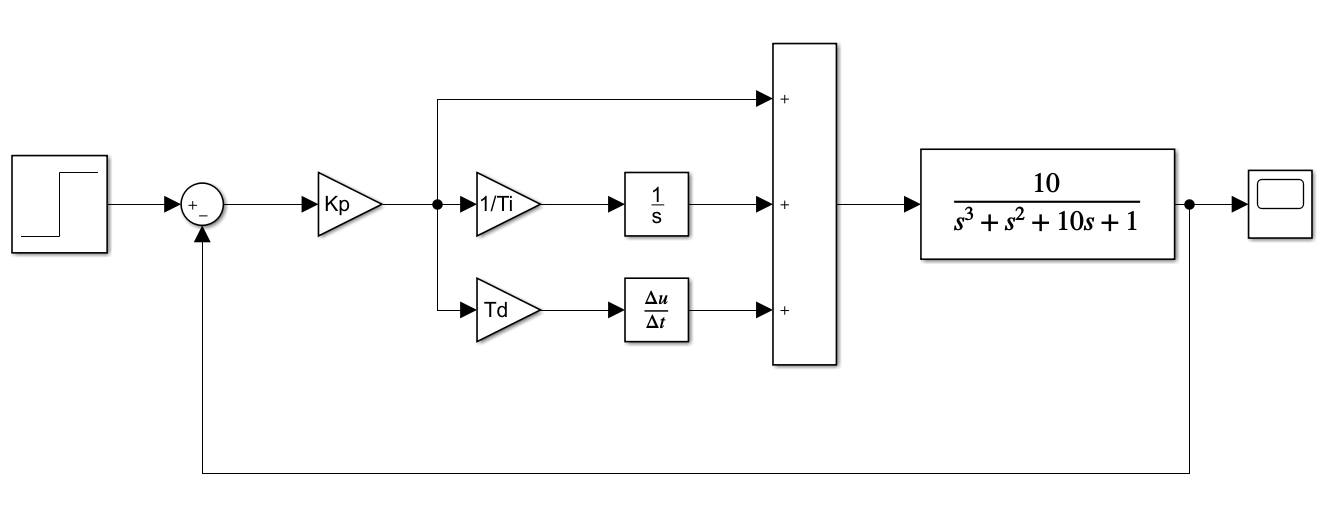


图18 引入PID控制器的系统模型

将控制器进行封装，如图19所示：

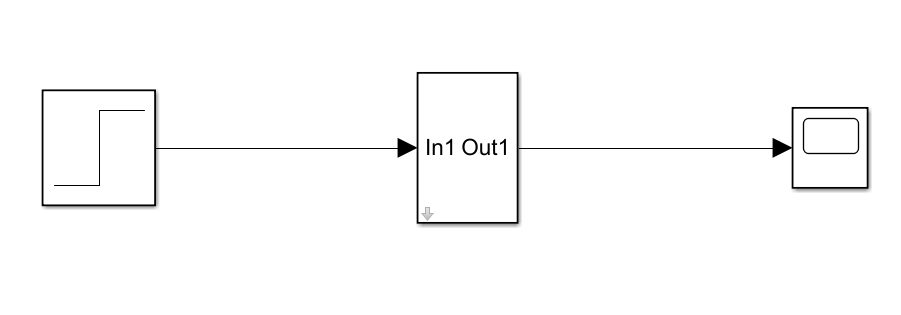


图19 封装后系统

双击子系统模块，可调节三个参数，如图20所示：

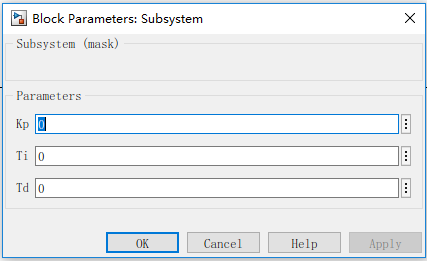


图20 参数调节窗口

采用Z-N法进行PID调整。

令Ti=∞，Td=0，在只有Kp的控制系统下，令Kp由0逐渐增大，直到系统的输出首次呈现持续的等幅振荡，此时的比例系数称为临界增益，用Kc表示，并记下振荡周期Tc。按照表1的整定公式确定PID控制器的参数。

表1 经验公式

|  |  |  |  |
| --- | --- | --- | --- |
| 控制规律 | Kc(Kp) | Ti | Td |
| P | 0.5Kc |  |  |
| PI | 0.45Kc | 0.83Tc |  |
| PID | 0.6Kc | 0.5Tc | 0.12Tc |

逐个调试Kp值，若衰减则应把比例度继续减小，若发散则应把比例度放大。经测试，当Kp=0.9时（如图21所示），系统的输出呈现持续的等幅振荡，如图22所示:

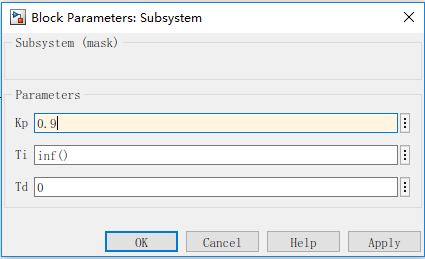


图21 等幅振荡时参数设置界面

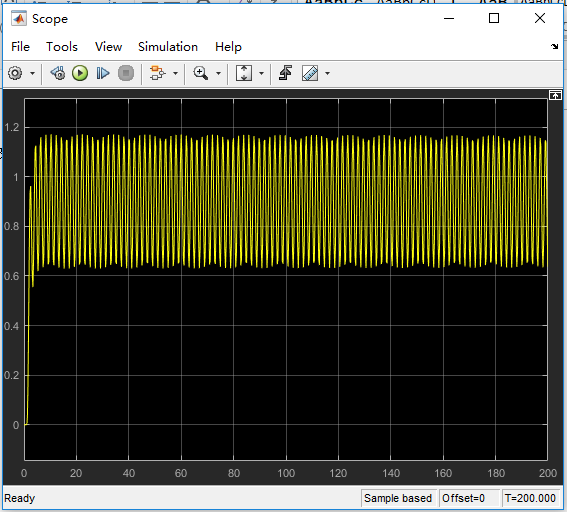


图22 系统输出等幅振荡波形图

由Kc=0.9求Tc，MatLab代码如下：

num=9;

den=[1 1 10 1];

G1=tf(num,den);

G=feedback(G1,1);

step(G)

系统临界震荡波形如图23所示，取10个周期两点，计算震荡周期Tc=1.9s。

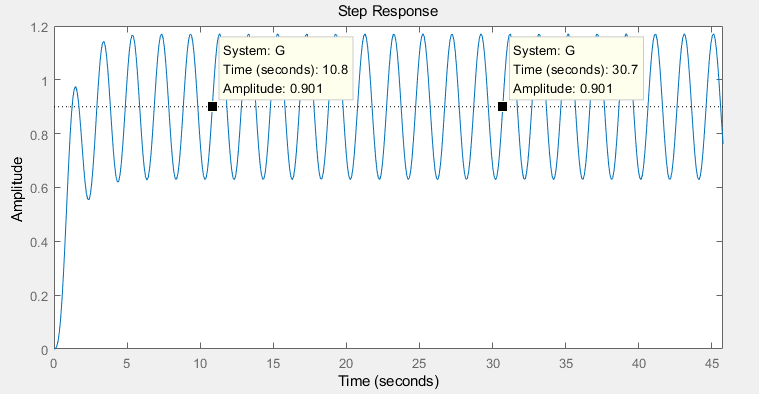


图23 系统临界震荡波形和周期计算图

将Kc=0.9，Tc=1.9s带入经验公式计算得到表2，确定各个参数：

表2 PID确定参数表

|  |  |  |  |
| --- | --- | --- | --- |
| 控制规律 | Kp | Ti | Td |
| P | 0.45 |  |  |
| PI | 0.405 | 1.577 |  |
| PID | 0.54 | 0.95 | 0.228 |

3.2模型仿真

将仿真时间定为300秒，绘制整定后的系统的单位阶跃响应曲线。

* + 1. P控制器

参数设置如图24所示：

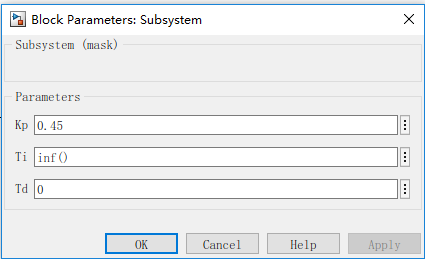


图24 P控制器参数设置

系统单位阶跃响应图像如图25所示:

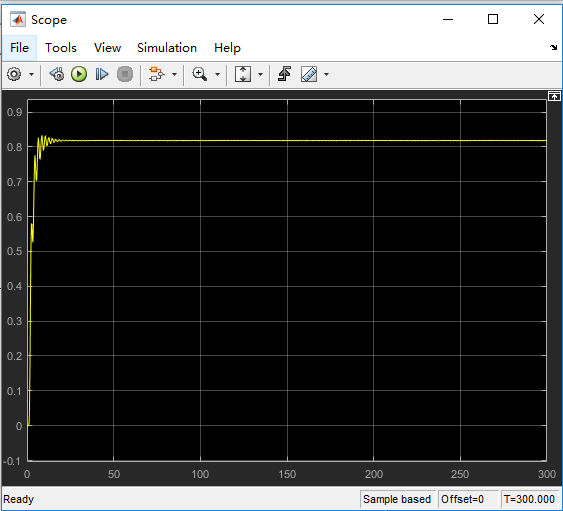


图25 P控制器系统单位阶跃响应曲线

由图可以看出，经过调整Kp，系统超调量减小，调节时间减小，但由于Kp小于1，系统存在较大的稳态误差，P值的改变只改变信号的增益而不影响其相位。经调整P值参数后，系统的超调量和调节时间有很大的改善，但Kp<1，稳态误差太大。

* + 1. PI控制器

参数设置如图26所示：

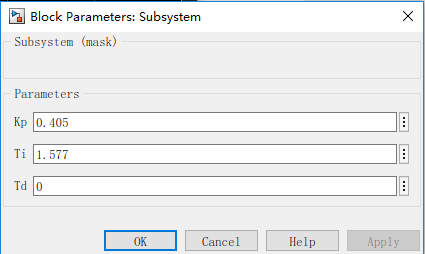


图26 PI控制器参数设置

系统单位阶跃响应图像如图27所示：

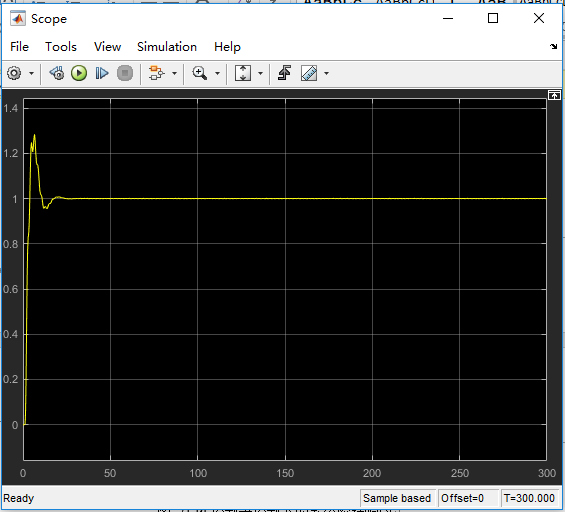


图27 PI控制器系统单位阶跃响应曲线

在P控制的基础上加入I控制，形成PI控制，系统稳态误差有所减少，但是超调量和调节时间有所加大。 PI控制可适用于对系统稳态误差要求精度高，但是对于调节时间要求精度低的实际模型。

* + 1. PID控制器

参数设置如图28所示：

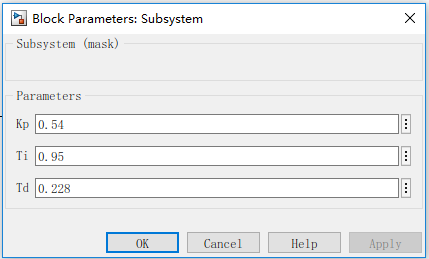


图28 PID控制器参数设置

系统单位阶跃响应图像如图29所示：



图29 PID控制器系统单位阶跃响应曲线

观察发现，此时的超调量大于40%，为减小超调，需要增加Ti。经多次测试，设置PID控制器参数如图30所示：

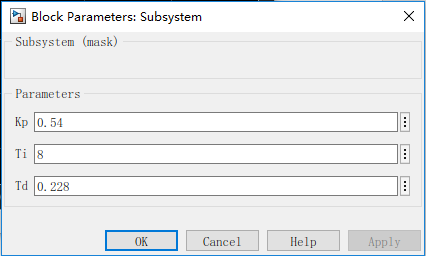


图30 重新调节的PID参数

此时的系统单位阶跃响应输出波形如图31所示：

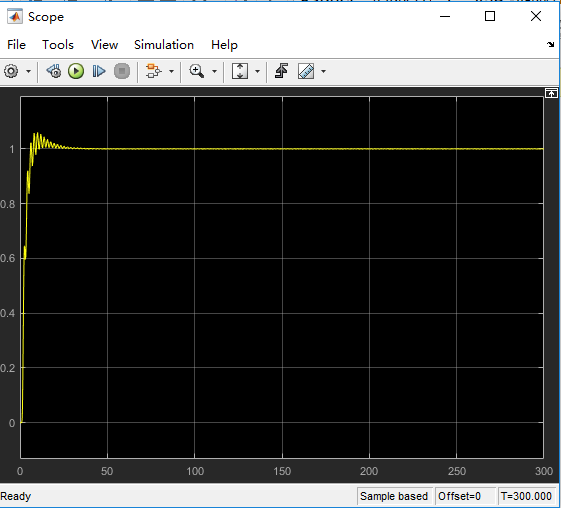


图31 重新调整参数PID控制器系统单位阶跃响应曲线

通过用工程整定法对PID参数的确定，我们体会到：P控制器只改变信号的增益，而不影响其相位。Kp与稳态误差成反比，可通过增大Kp减小稳态误差，从而提高系统的控制精度，但会降低系统的动态性能，甚至可能造成闭环系统不稳定，所以，很少单独使用P控制规律。而PI控制器在系统中增加一个位于原点的开环极点和一个位于S左半平面的开环零点，改善系统的稳态性能。只要积分时间常数足够大，PI控制器对系统稳定性的不利影响可大为减弱。PID控制器则在系统中增加一个位于原点的开环极点和两个负实零点，具有提高系统稳态性能的优点，且在提高系统动态性能方面有更大的优越性。

3.3 M文件设计及性能指标求取

采用编程法求取系统阶跃响应性能指标：超调量、调节时间。

此时，Kp=0.54,Ti=8,Td=0.228，首先计算此时的系统开环传递函数，MatLab代码如下：

s=tf('s');

Gpid=0.54\*(1+1/(8\*s)+0.228\*s);

num=10;

den=[1 1 10 1];

G0=tf(num,den);

Gc=G0\*Gpid;

G=feedback(Gc,1)

输出得到系统传递函数为：

9.85 s^2 + 43.2 s + 5.4

G = ----------------------------------------

8 s^4 + 8 s^3 + 89.85 s^2 + 51.2 s + 5.4

再求该系统的超调量和调节时间，MatLab代码如下：

y=step(G)

%-------------------------求超调量

fprintf('稳态响应终值为：');

c=dcgain(G)

fprintf('阶跃响应最大值及对应时间为：');

[Y,t]=max(y)

fprintf('超调量为：');

chaotiao=(Y-c)/c

%---------------------求调节时间

tt=0:0.00001:100;

yy=step(G,tt);

i=length(tt);

while(yy(i)>0.98\*c)&(yy(i)<1.02\*c);

i=i-1;

end

ts=tt(i)

输出结果为：

稳态响应终值为：

c = 1

阶跃响应最大值及对应时间为：

Y = 1.0415

t = 94

超调量为：

chaotiao = 0.0415

ts = 14.7265

可知，调节时间为14.7265s，超调量为4.15%，系统单位阶跃响应如图32所示：

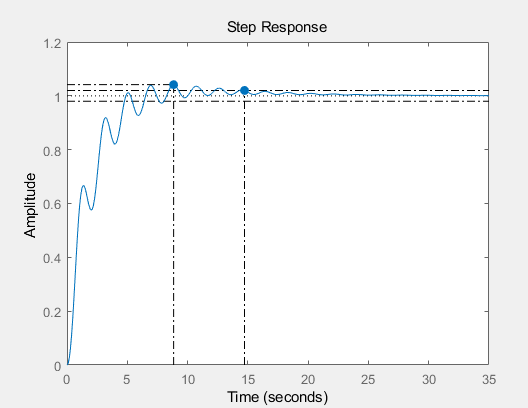


图32 系统单位阶跃响应曲线图

第四章 总结

虽然本次作业选择的是基础题目，没有自选控制系统进行分析，但是依然有很大的收获，将课堂知识进行实践是一个很有趣的事情，对于课堂上一知半解的东西，在实践中得到了更深刻的认识和理解，知识的记忆也更加深刻，同时，即便是基本题目也难免有自己不会的地方，这时网络和课外书籍就成为最重要的资源，锻炼了自己查找和解决问题的能力，收获颇丰。

第五章 参考文献

[1] 薛定宇. 控制系统仿真与计算机辅助设计[M]. 北京: 机械工业出版社, 2008: 13-274

[2] 胡寿松. 自动控制原理[M]. 北京:科学出版社，2002年