

北京邮电大学

《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
1.1 数据定义	1
1.2 数据读取与绘图	2
1.3 函数构建与绘图	2
1.4 系统函数构建及其 Bode 图绘制	4
第二章 子系统封装	5
2.1 用 Simulink 建立系统	5
2.2 观察输入输出波形	6
2.3 数据导出	7
第三章 PID 控制器参数整定	9
3.1 工程整定法确定参数	9
3.2 模型仿真	13
3.2.1 P 控制器	13
3.2.2 PI 控制器	14
3.2.3 PID 控制器	15
3.3 M 文件设计及性能指标求取	17
第四章 总结	19
第五章 参考文献	19

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， $0 \leq t \leq 200$)；第二列为与 t 对应的 201 个幅值数据，作为信号 $f_1(t)$ 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 `data.dat` 文件部分截图如图 1 所示。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	data													
	VarName1	VarName2	VarName3	VarName4	VarName5	VarName6	VarName7	VarName8	VarName9	VarName10	VarName11	VarName12	VarName13	VarName14
	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值	数值
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
2	0	3	0	3	3	6	6	8	10	9	4	6	12	10
3	1	3	4	5										
4	2	3	5	3	6									

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 $f_1(t)$ 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 f_1 -t 时域波形图如图 2 所示。

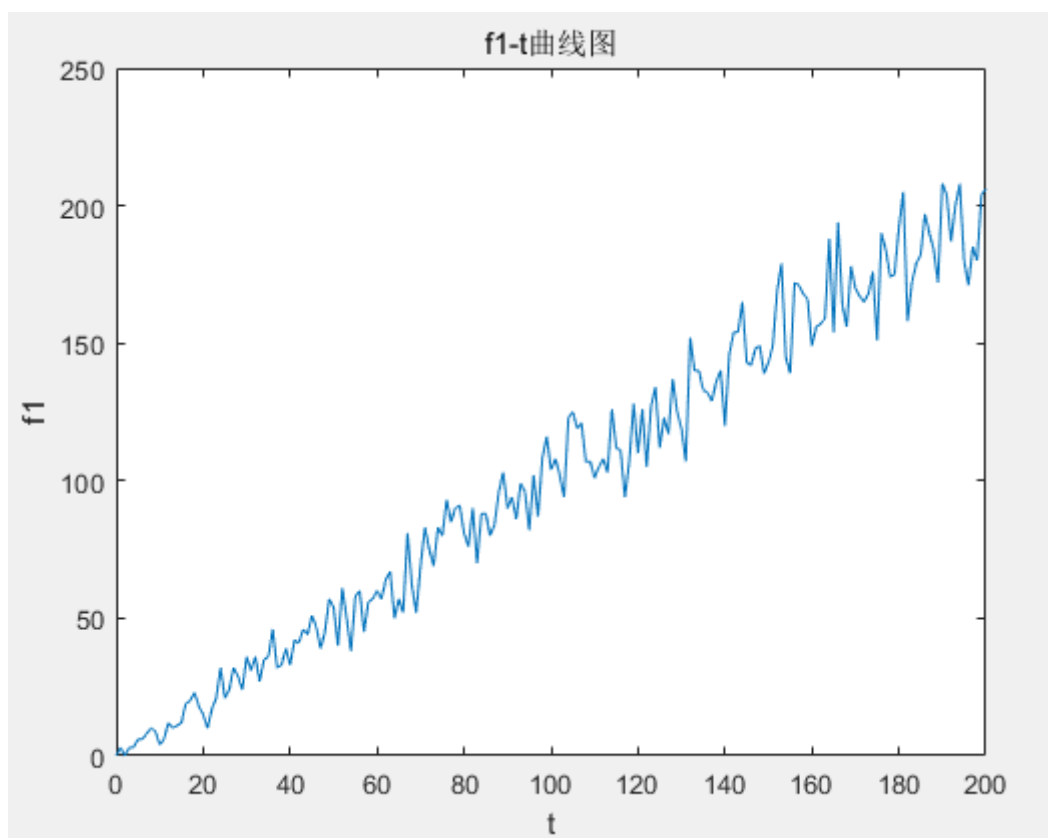


图 2 $f_1(t)$ -t 曲线图

1.3 函数构建与绘图

求取 $f_2(t)$ ，将结果保存到 result.mat 文件，画出其时域波形。

$$f_2(t) = \begin{cases} 2 * f_1(t) & 0 \leq t < 10 \\ f_1^2(t) & 10 \leq t \leq 100 \text{ 且 } |f_1(t)| < 15 \\ f_1(t-3) & \text{其他} \end{cases}$$

利用循环求 f_2 ，plot()函数绘制 $f_2(t)$ -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 f_1 -t 时域波形图如图 3 所示。

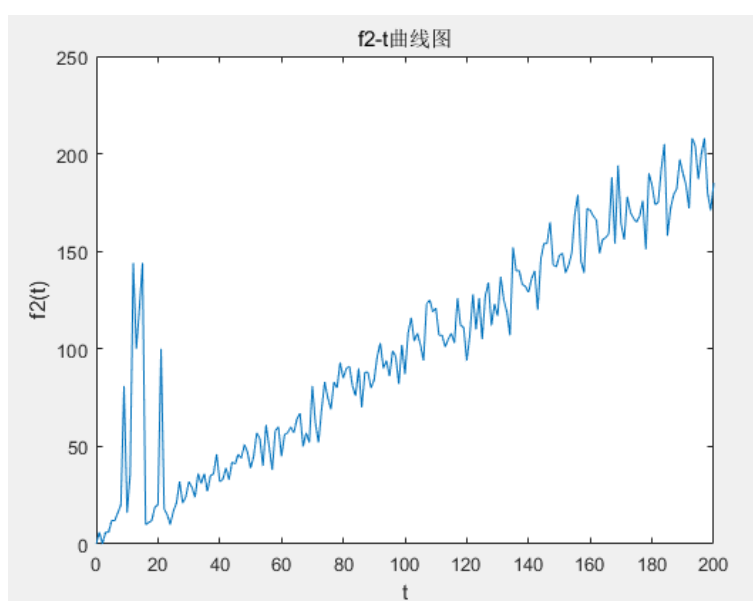


图 3 $f_2(t)$ -t 曲线图

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：

$$\text{sys} = \frac{3s^4 + 5s^3 + 7s^2 + 4s + 2}{2s^4 + 7s^3 + 9s^2 + 1}$$

画出的 bode 图如图 4 所示：

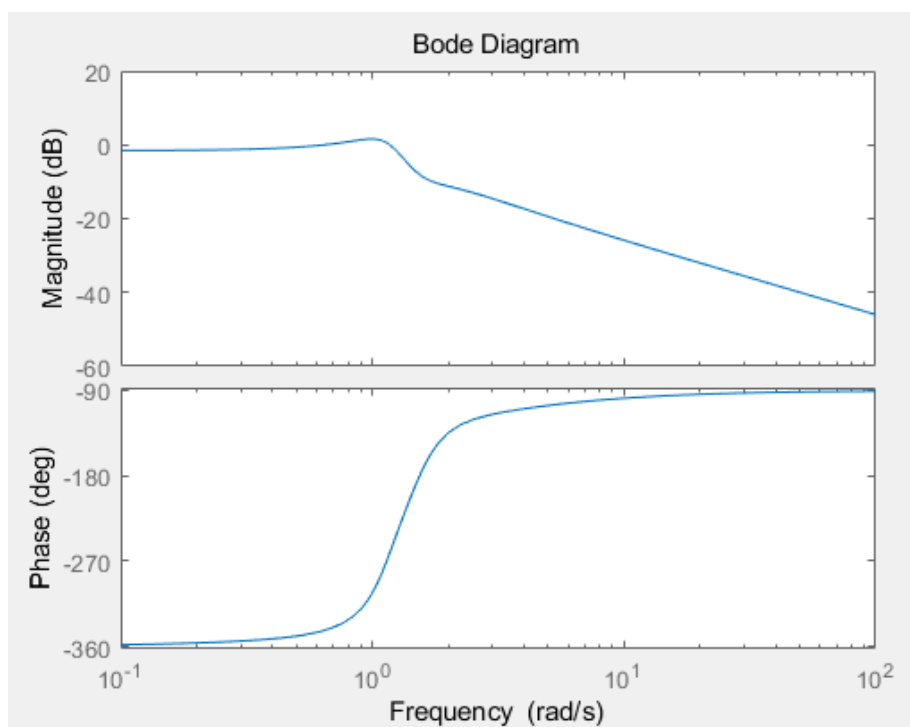


图 4 Bode 图

第二章 子系统封装

2.1 用 Simulink 建立系统

建立系统方程 $y = ax^3 + bx + c$ ，其中 x 为输入， y 为输出， a, b, c 为常数。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：

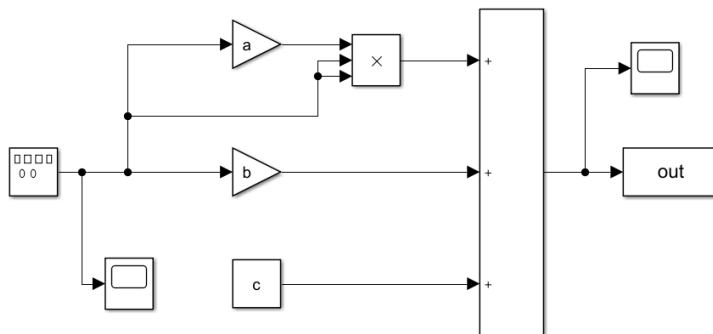


图 5 Simulink 子系统模型

封装后系统模型如图 6 所示：

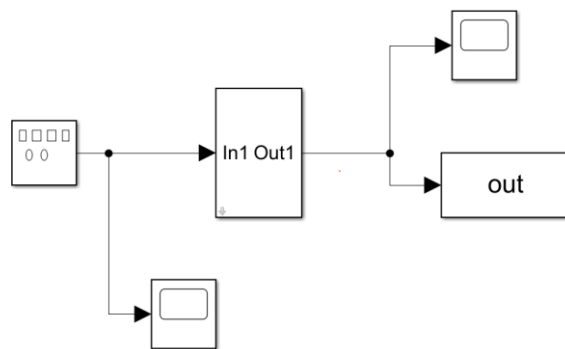


图 6 封装后系统模型

设置动态参数可调的界面如图 7 所示：

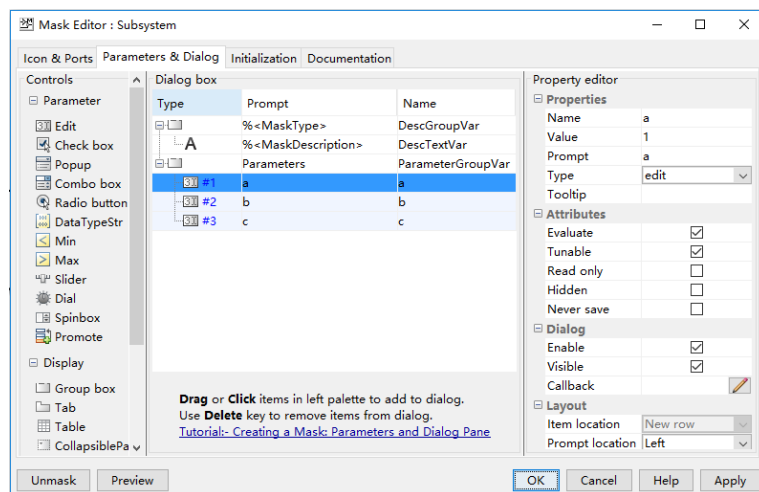


图 7 设置子系统参数

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：

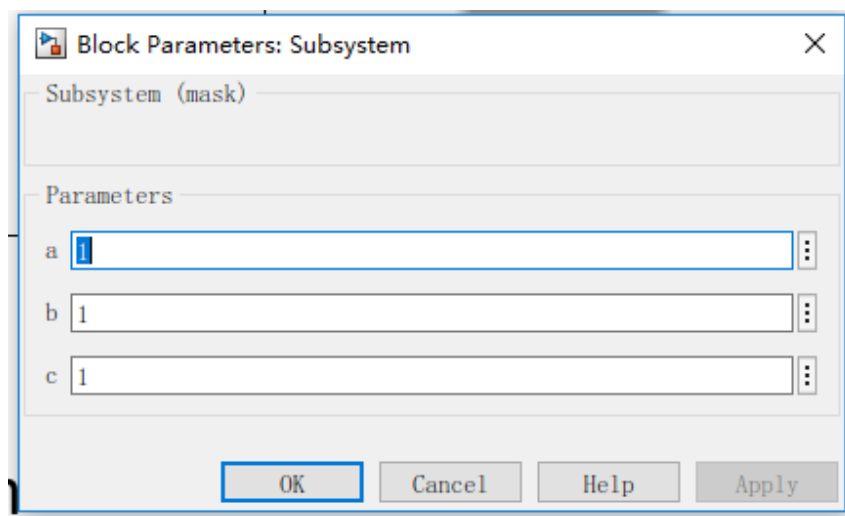


图 8 修改子系统参数界面

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：

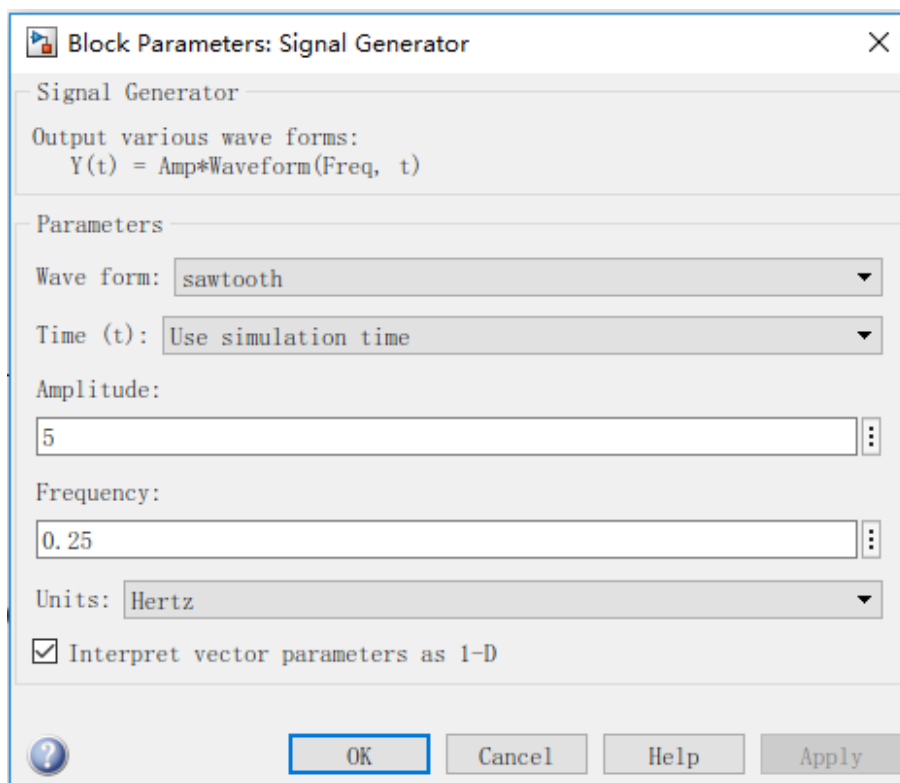


图 9 输入波形参数设置界面

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：

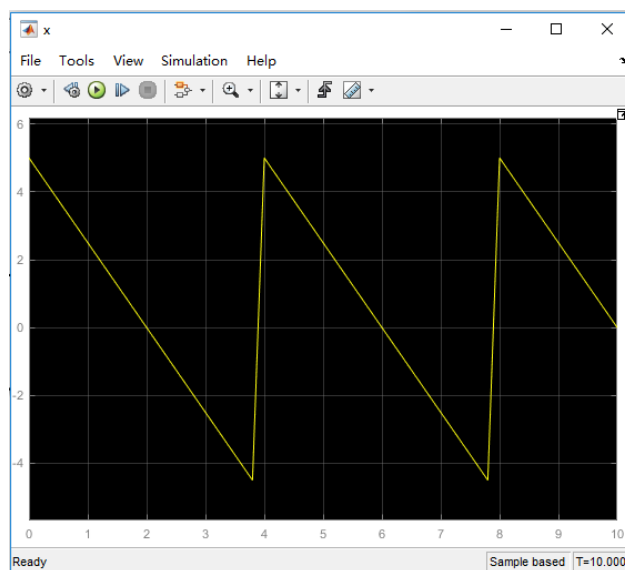


图 10 输入锯齿波波形图

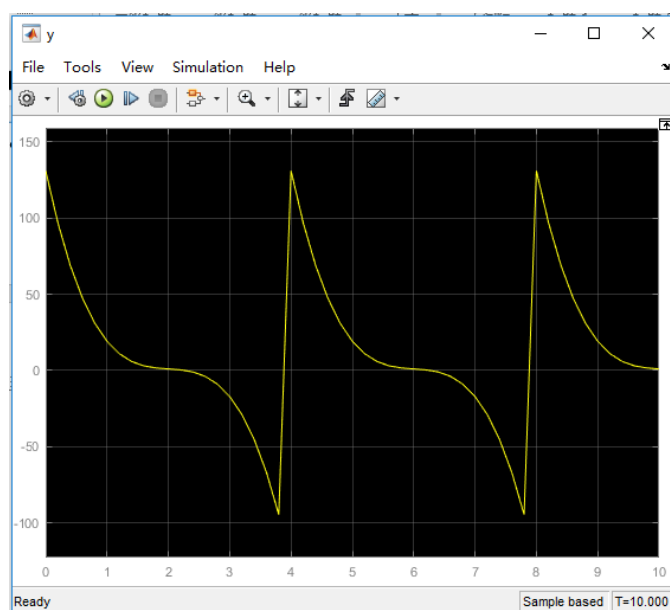


图 11 输出波形图

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：

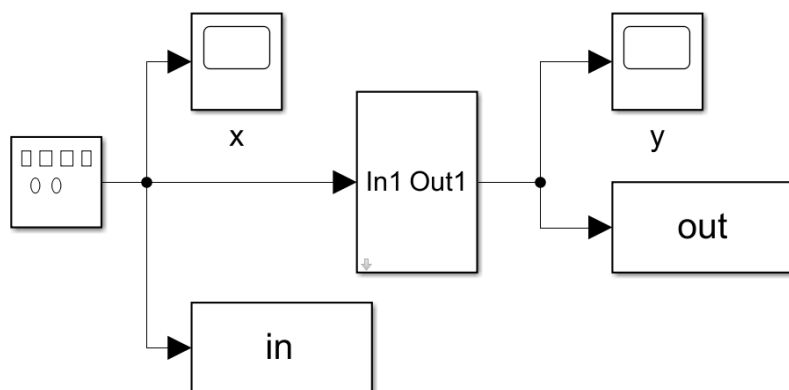


图 12 数据通过 simout 组件返回工作空间

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:

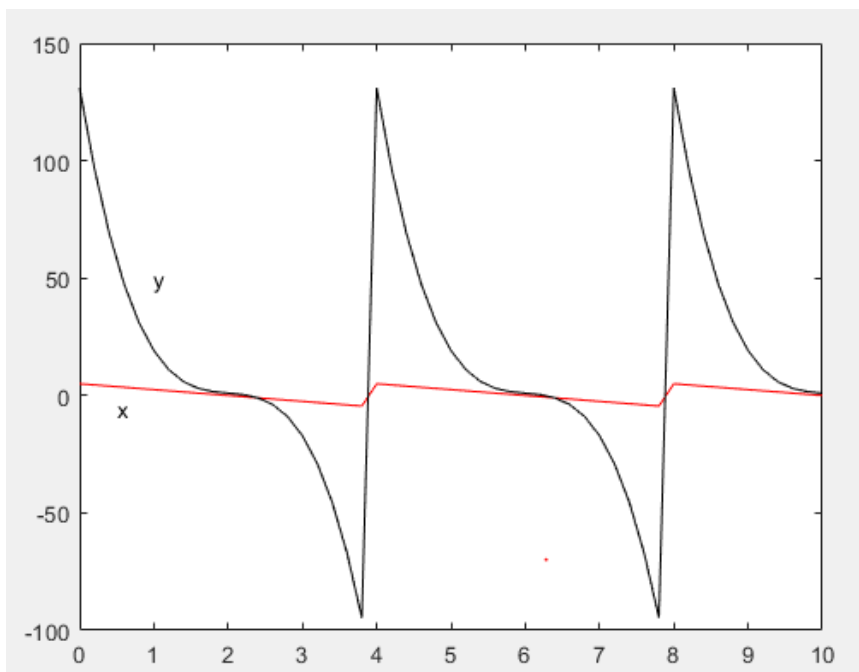
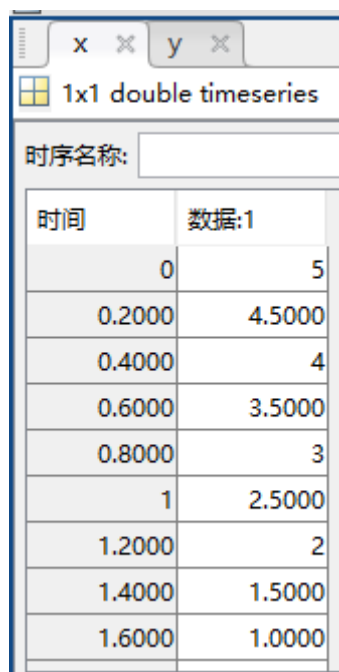


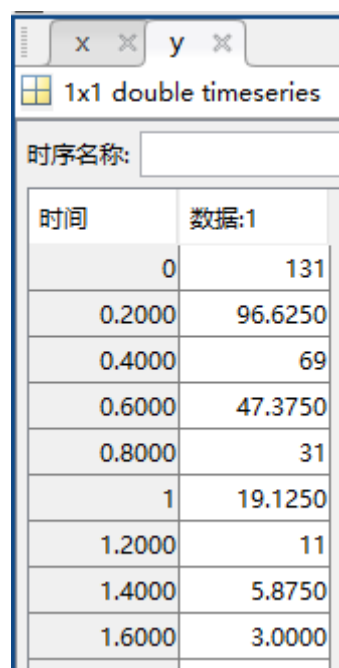
图 13 输入输出曲线图

导出数据 x, y 如图 14, 15 所示:



时间	数据:1
0	5
0.2000	4.5000
0.4000	4
0.6000	3.5000
0.8000	3
1	2.5000
1.2000	2
1.4000	1.5000
1.6000	1.0000

图 14 导出的 x 部分数据



时间	数据:1
0	131
0.2000	96.6250
0.4000	69
0.6000	47.3750
0.8000	31
1	19.1250
1.2000	11
1.4000	5.8750
1.6000	3.0000

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 $G_n(s) = \frac{10}{s^3 + s^2 + 10s + 1}$, 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：

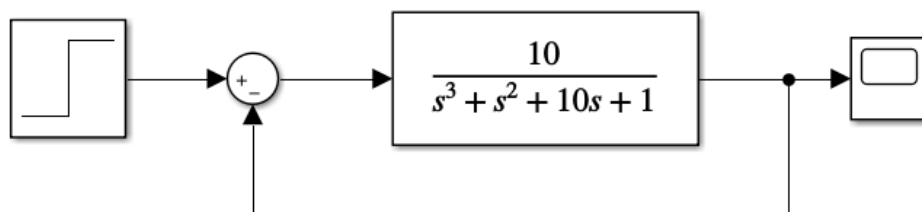


图 16 原系统模型

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：

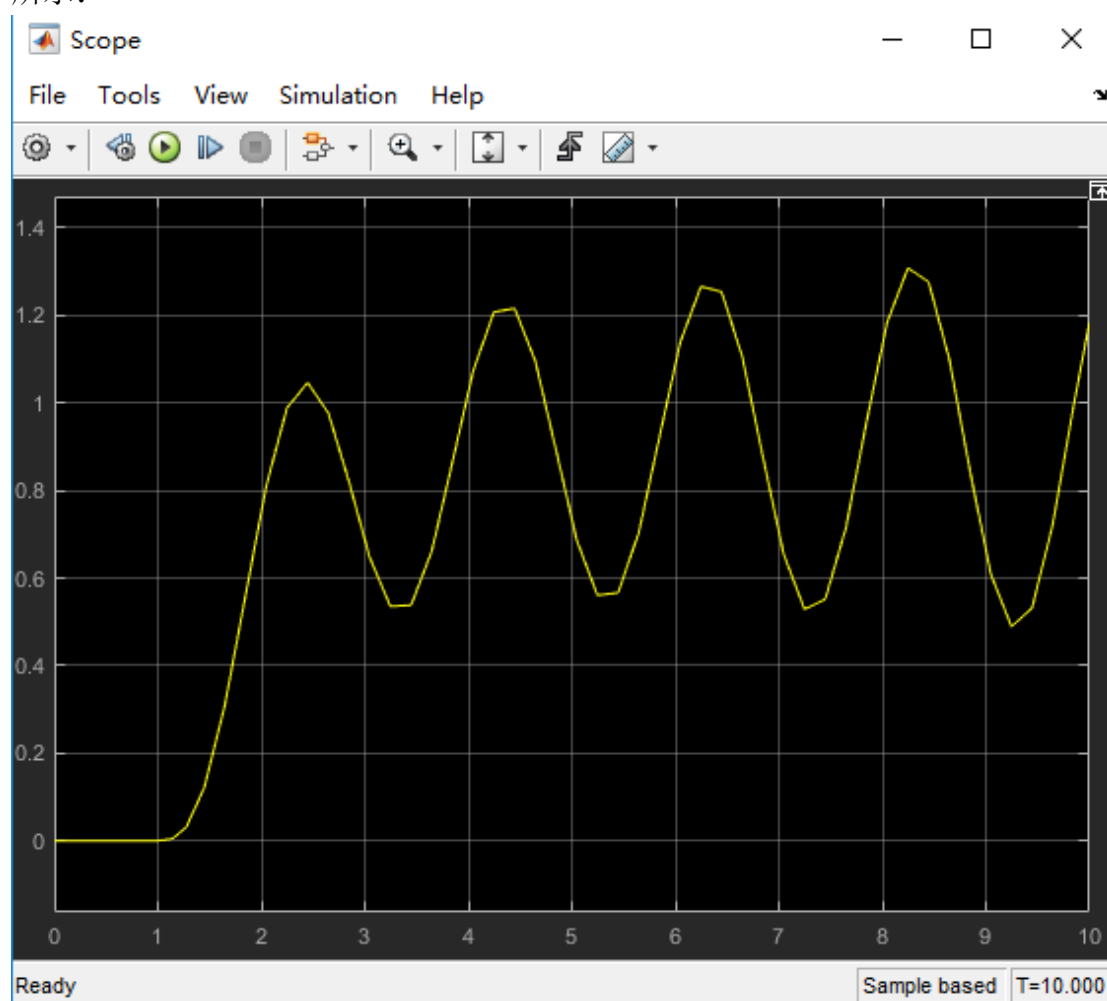


图 17 原系统阶跃响应波形图

故引入 PID 控制器，如图 18 所示：

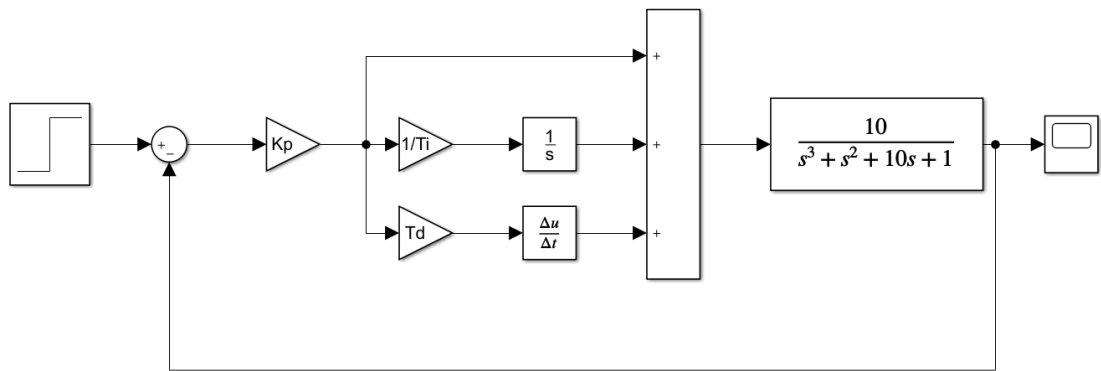


图 18 引入 PID 控制器的系统模型

将控制器进行封装，如图 19 所示：

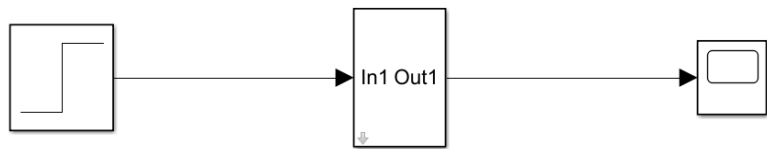


图 19 封装后系统

双击子系统模块，可调节三个参数，如图 20 所示：

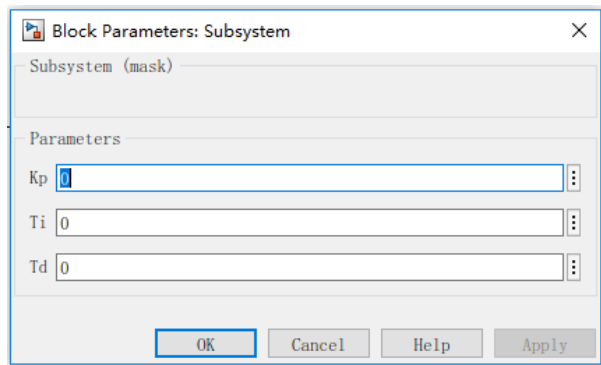


图 20 参数调节窗口

采用 Z-N 法进行 PID 调整。

令 $T_i = \infty$ ， $T_d = 0$ ，在只有 K_p 的控制系统下，令 K_p 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 K_c 表示，并记下振荡周期 T_c 。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	$K_c(K_p)$	T_i	T_d
P	$0.5K_c$		
PI	$0.45K_c$	$0.83T_c$	
PID	$0.6K_c$	$0.5T_c$	$0.12T_c$

逐个调试 K_p 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 $K_p=0.9$ 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:

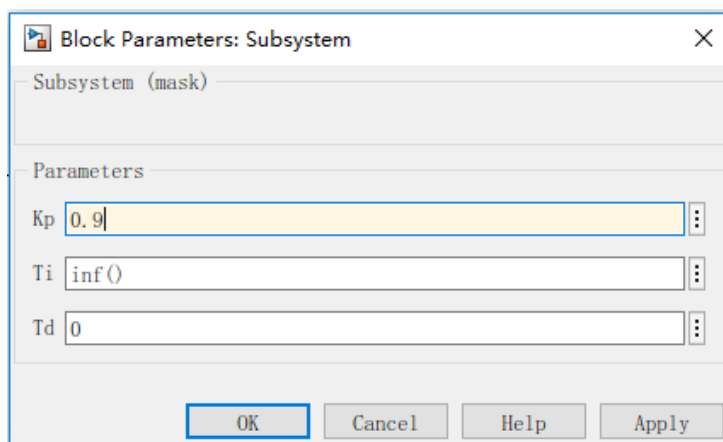


图 21 等幅振荡时参数设置界面

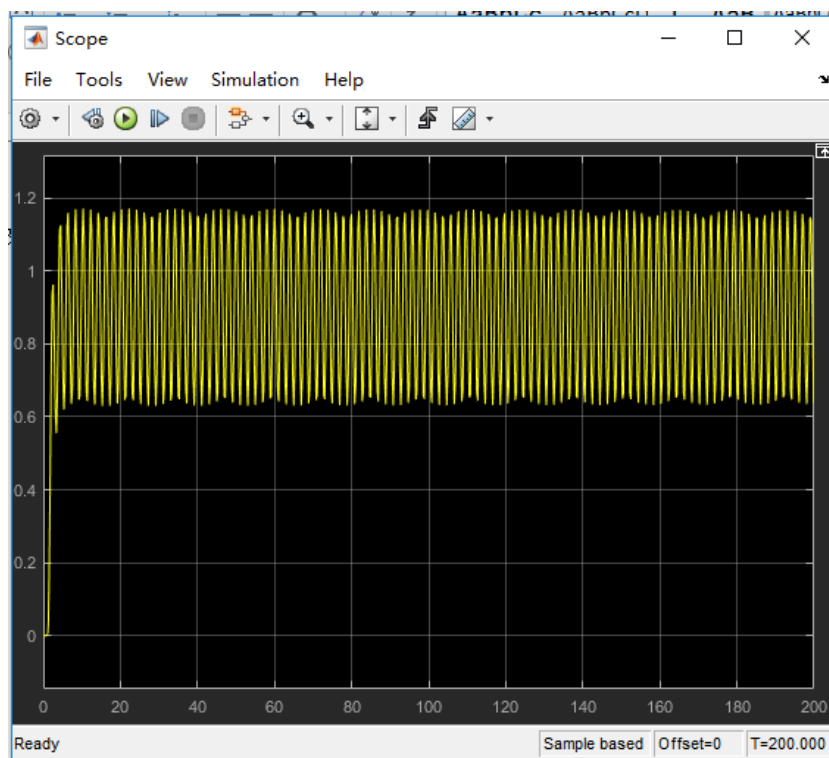


图 22 系统输出等幅振荡波形图

由 $K_c=0.9$ 求 T_c , MATLAB 代码如下:

```
num=9;  
den=[1 1 10 1];  
G1=tf(num,den);  
G=feedback(G1,1);  
step(G)
```

系统临界振荡波形如图 23 所示，取 10 个周期两点，计算振荡周期 $T_c=1.9s$ 。

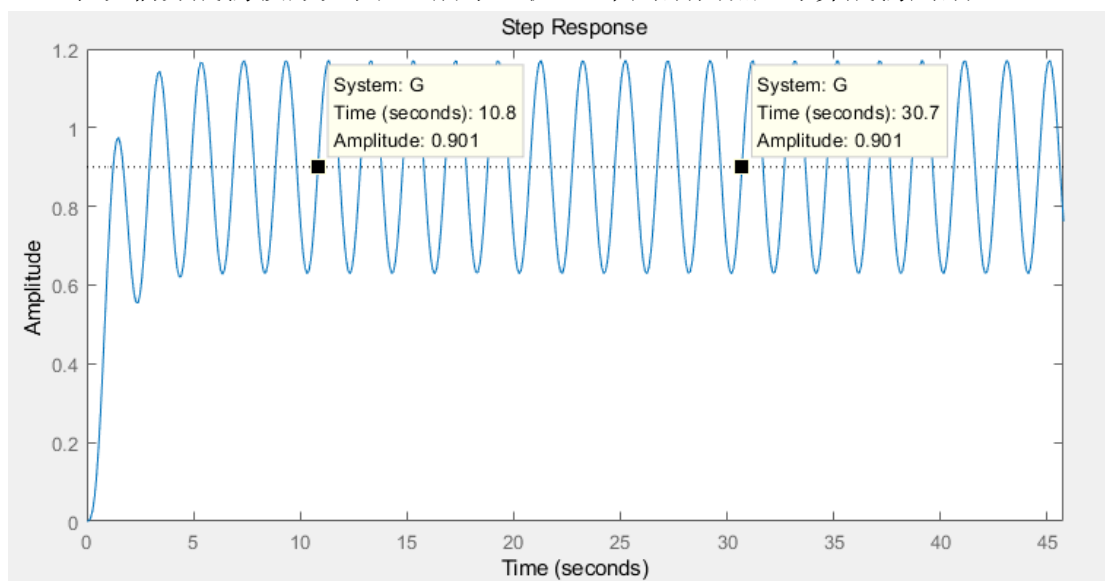


图 23 系统临界振荡波形和周期计算图

将 $K_c=0.9$ ， $T_c=1.9s$ 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	K_p	T_i	T_d
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。

3.2.1 P 控制器

参数设置如图 24 所示：

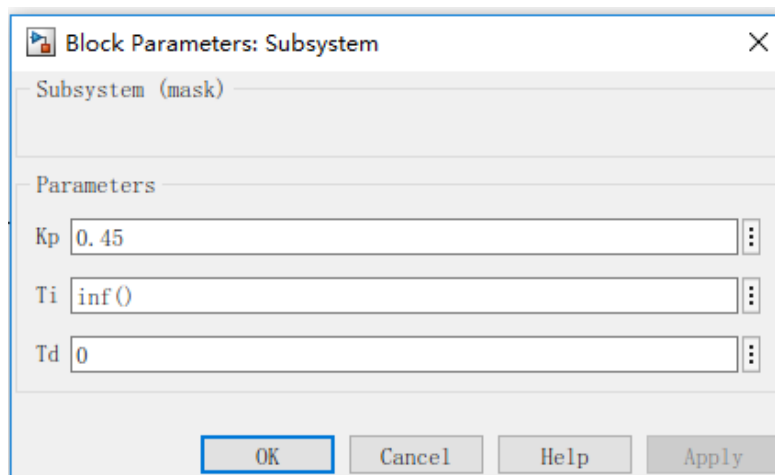


图 24 P 控制器参数设置

系统单位阶跃响应图像如图 25 所示：

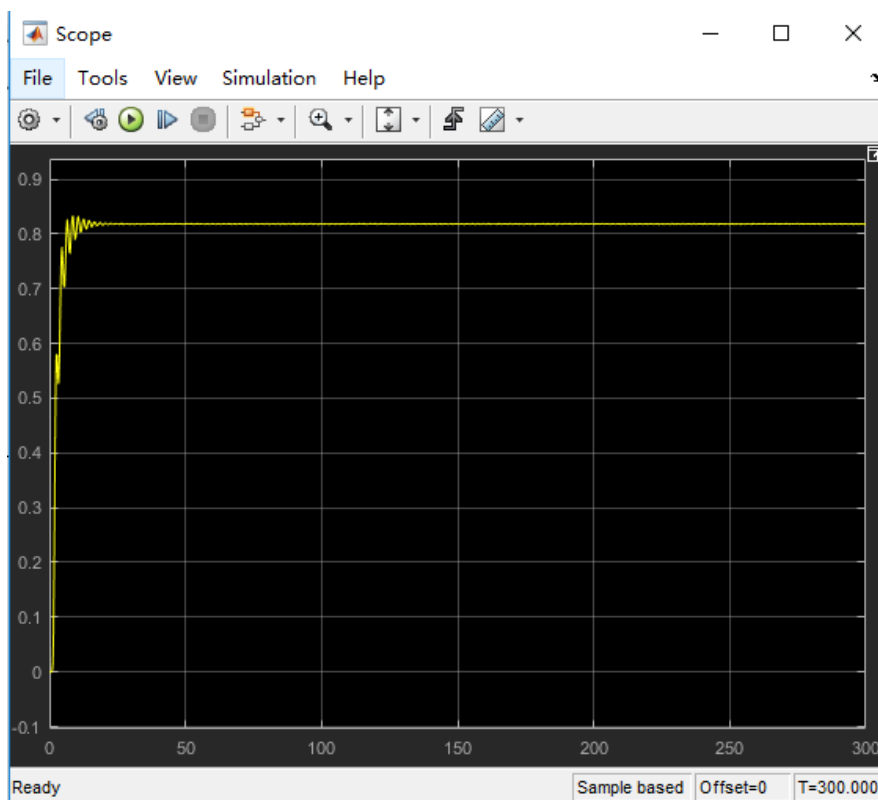


图 25 P 控制器系统单位阶跃响应曲线

由图可以看出，经过调整 K_p ，系统超调量减小，调节时间减小，但由于 K_p 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 $K_p < 1$ ，稳态误差太大。

3.2.2 PI 控制器

参数设置如图 26 所示：

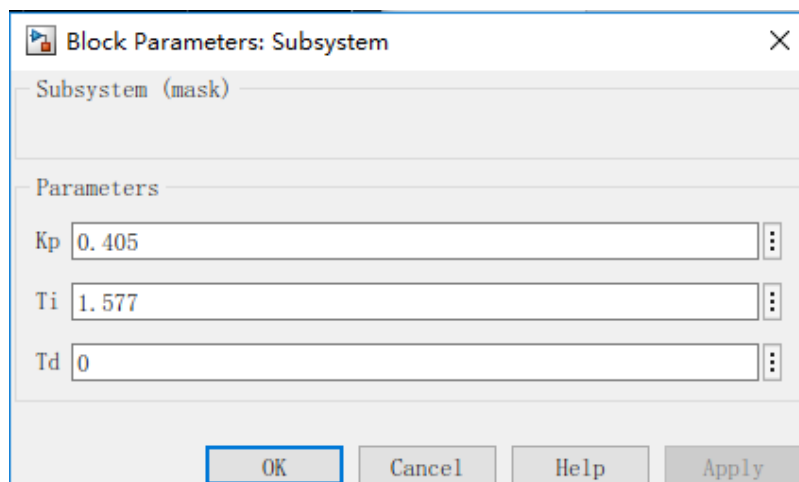


图 26 PI 控制器参数设置

系统单位阶跃响应图像如图 27 所示：

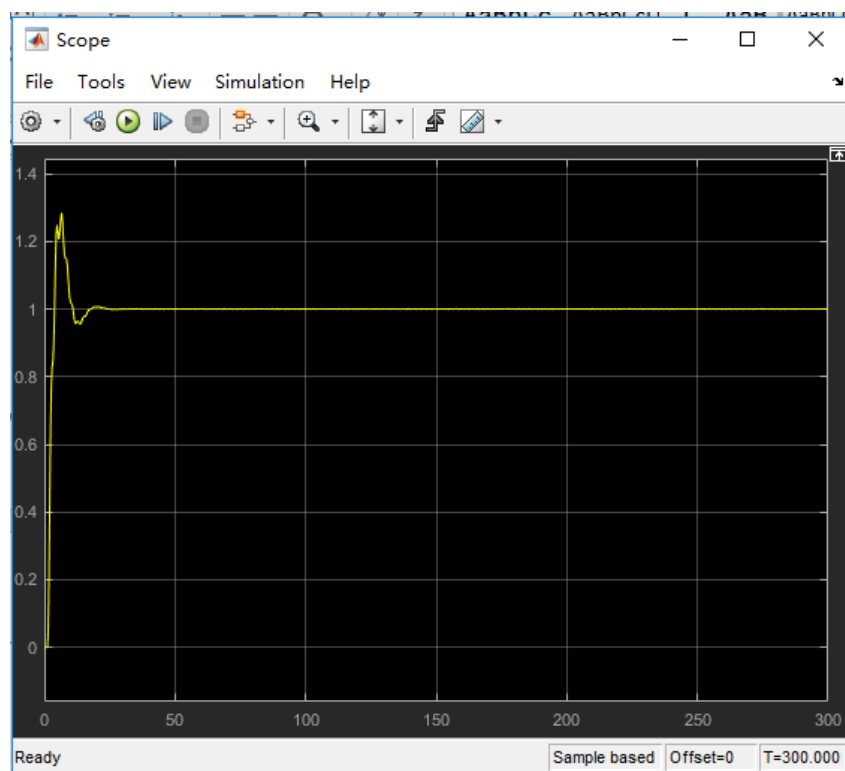


图 27 PI 控制器系统单位阶跃响应曲线

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。

3.2.3 PID 控制器

参数设置如图 28 所示：

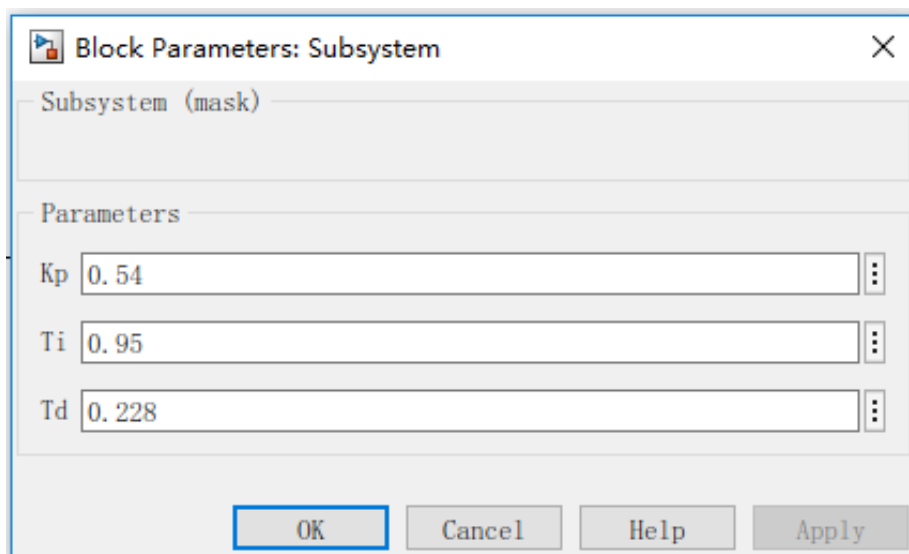


图 28 PID 控制器参数设置

系统单位阶跃响应图像如图 29 所示：

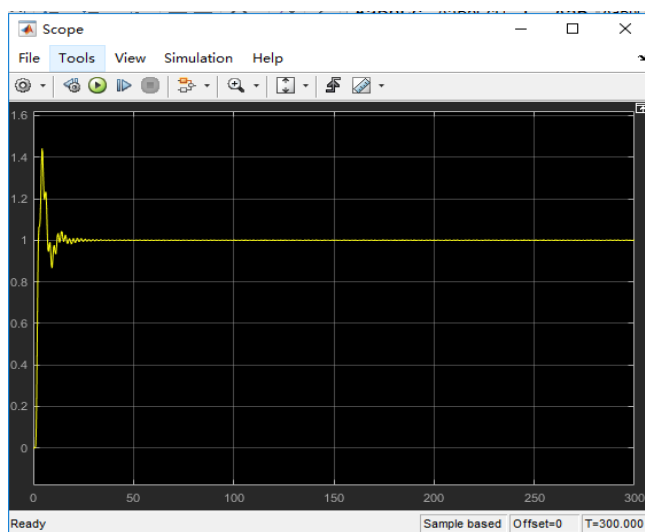


图 29 PID 控制器系统单位阶跃响应曲线

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

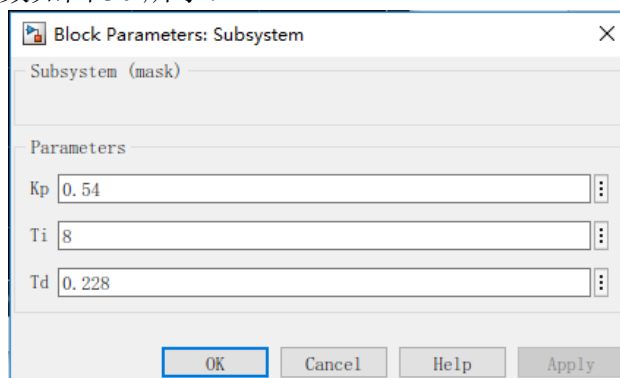


图 30 重新调节的 PID 参数

此时的系统单位阶跃响应输出波形如图 31 所示：

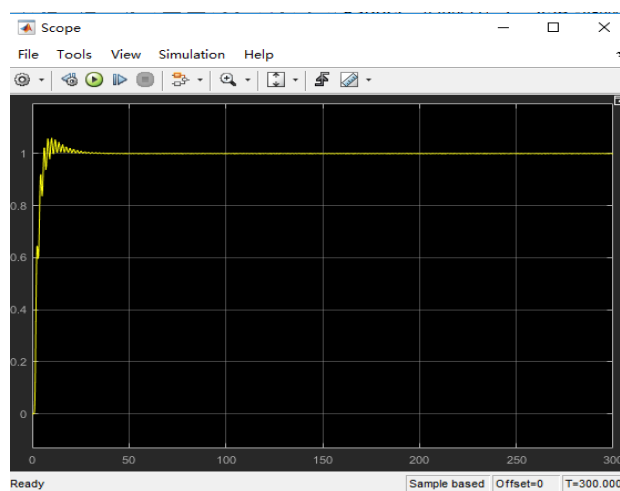


图 31 重新调整参数 PID 控制器系统单位阶跃响应曲线

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。 K_p 与稳态误差成反比，可通过增大 K_p 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， $K_p=0.54, T_i=8, T_d=0.228$ ，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：

$$G = \frac{9.85 s^2 + 43.2 s + 5.4}{8 s^4 + 8 s^3 + 89.85 s^2 + 51.2 s + 5.4}$$

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：

$\text{chaotiao} = 0.0415$

$ts = 14.7265$

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：

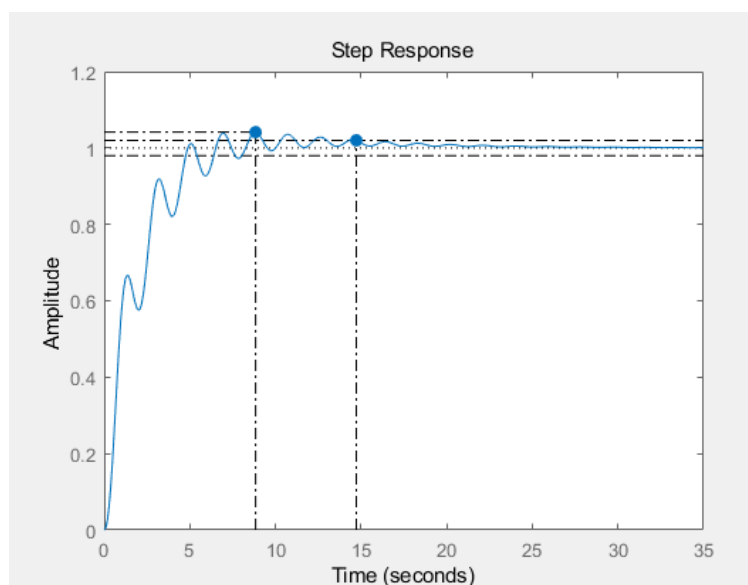


图 32 系统单位阶跃响应曲线图

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