北京大學

本科实验报告

课程名称：　自动控制理论

姓　　名：　金镇雄

学　　院：　元培

　　系：　智能科学系

专　　业：　智能科学与技术

年　　级：　19

学　　号：　1900094619

2022 年　　4 月　　10 日

实验一、PID 控制实验

一、 实验目的

- 了解 PID 控制器的原理
- 了解并使用 Ziegler-Nichols 方法
- 了解 PID 控制的各组件对系统稳定性的影响

二、 实验器材

PC 机、MATLAB、任一实际系统的传递函数

三、 实验原理

在工程实际中，应用最为广泛的调节器控制规律为比例（Proportional）、积分（Integral）、微分（Derivative）控制，简称 PID 控制。PID 控制是三种反馈控制：比例控制，积分控制与微分控制的统称。根据控制对象和应用条件，可以采用这三种控制的部分组合，即 P 控制，PI 控制，PD 控制或者是三者的组合，即真正意义上的 PID 控制。

PID 一般算式及模型控制规律如下：

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$

其中 $u(t)$ 为控制器的输出； $e(t)$ 为偏差，即期望值与实际值的差值；K_p为控制器的放大系数，即比例系数；K_i为积分系数；K_d为微分系数。

比例控制是最简单的控制方式之一，其控制器的输出与输入误差信号成比例关系。比例系数K_p直接决定控制作用的强弱，加大K_p可以减少系统的稳态误差，提高系统的动态响应速度，但K_p过大会使动态质量变坏，引起被控制量振荡甚至导致闭环系统不稳定。当只有比例控制时系统输出会存在稳态误差。

在积分控制中，控制器的输出与输入误差信号的积分成正比关系。系统在进入稳态后会存在稳态误差，为了消除稳态误差，在控制器中必须引入“积分项”。积分项对误差取决于时间的积分，随着时间的增加，积分项会增大。这样，即便误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大使稳态误差进一步减小，直到接近于零。但积分控制将使系统的动态过程变慢，而且过强的积分作用使系统的超调量增大，从而系统的稳定性变坏。

在微分控制中，控制器的输出与输入误差信号的微分成正比关系。在克服误差的调节过程中可能会出现振荡甚至失稳。解决的办法是使抑制误差的作用的变化“超前”。它能预测误差变化的趋势，产生超前的校正作用，有助于减少超调，克服振荡，使系统趋于稳定，并加快系统的动作速度，减少调节时间，从而改善系统的动态性能。

Ziegler-Nichols 方法是一种整定 PID 控制器，探索其控制参数的方法。其调试方式为，首先将积分和微分增益设置为 0，然后比例增益从零开始逐渐增加，直到到达极限增益K_U，此时控制器输出值以恒定值振荡。K_U和振荡周期T_U根据不同的类型，按下表中的方式来设置比例、积分和微分增益。

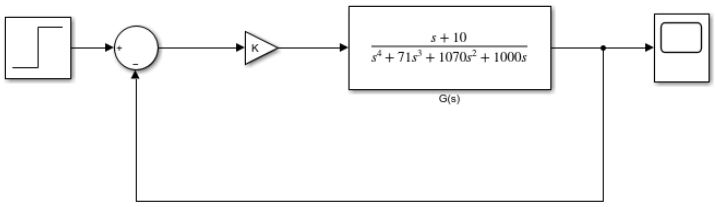
Ziegler–Nichols method[1]

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$0.80T_u$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$0.125T_u$	–	$0.10K_uT_u$
classic PID[2]	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_uT_u$
Pessen Integral Rule[2]	$0.7K_u$	$0.4T_u$	$0.15T_u$	$1.75K_u/T_u$	$0.105K_uT_u$
some overshoot[2]	$0.33\bar{3}K_u$	$0.50T_u$	$0.33\bar{3}T_u$	$0.66\bar{6}K_u/T_u$	$0.1\bar{1}K_uT_u$
no overshoot[2]	$0.20K_u$	$0.50T_u$	$0.33\bar{3}T_u$	$0.40K_u/T_u$	$0.066\bar{6}K_uT_u$

四、实验内容

1. 选定系统

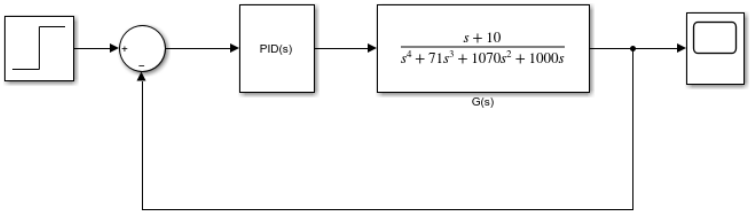
本实验中使用了电梯模型。



系统的传递函数为$G(s)=\frac{K(s+10)}{s^4+71s^3+1070s^2+1000s}$。

2. 引入 PID 控制器

由于原系统包含 P 控制，在比例控制器处并联积分控制器或微分控制器，即可实现 PI 控制、PD 控制以及 PID 控制。



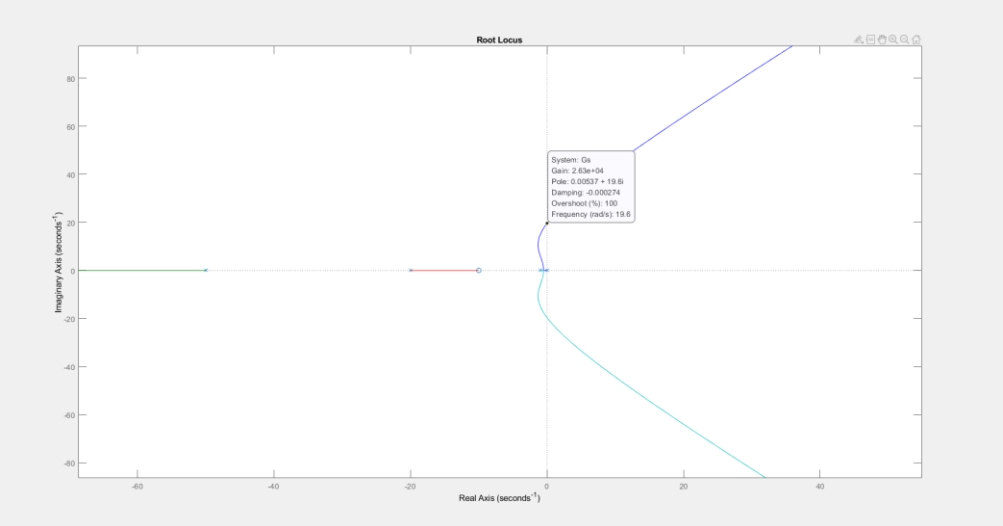
3. **用 Ziegler-Nichols 方法探索控制参数**

A. **令$K_I = K_D = 0$，求临界增益K_U**

根轨迹与虚轴的交点处所对应的 K 值即是划分系统工作于稳态或非稳态的临界值。因此，求临界增益K_U只需观察系统的根轨迹。根轨迹与虚轴的交点取决于如下特征方程：

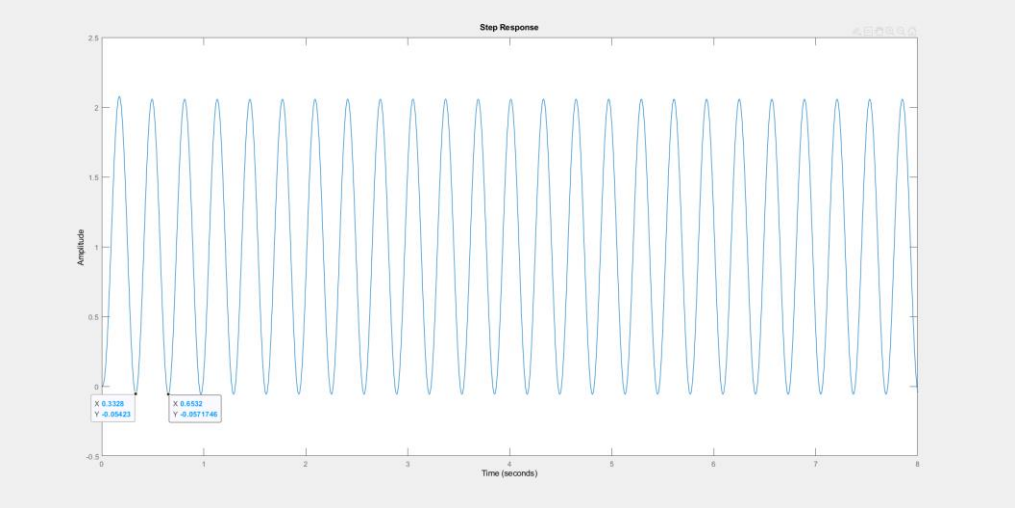
$$1 + F(j\omega)A(j\omega) = 0，\ F(s)A(s) = \frac{N(s)}{D(s)} = K\frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)}$$

使用 MATLAB 中 rlocus 函数即可得到系统的根轨迹，在图上点击根轨迹与虚轴的交点处可得K_U值：26399.796。



B. **求临界振荡周期T_U**

求系统的阶跃响应和临界振荡周期T_U，求得T_U=0.3024。



C. 计算K_P、K_I、K_D

根据K_U和T_U值，按照 Ziegler-Nichols 方法参数整定公式，计算K_P、K_I、K_D。

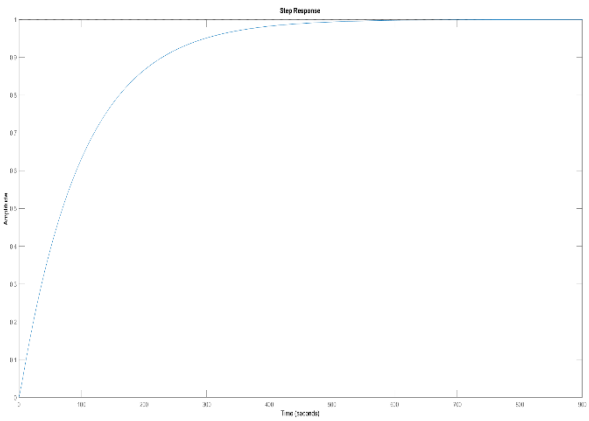
	K_P	K_I	K_D
P 控制	1319.9898	0	0
PI 控制	11879.9082	46347.9564	0
PD 控制	21119.8368	0	845.8495
PID 控制	15839.8776	98875.6404	634.3870978

五、实验结果与分析

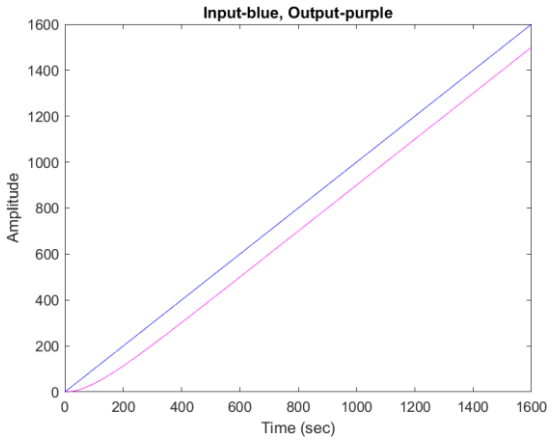
1. 原系统（$K_P = 1$）

$K_P = 1$时，阶跃响应的超调量、上升时间、调节时间以及峰值时间如下：

K_P	1
K_I	0
K_D	0
超调量	0
上升时间	217.5612
调节时间	388.3686
峰值时间	1044.3



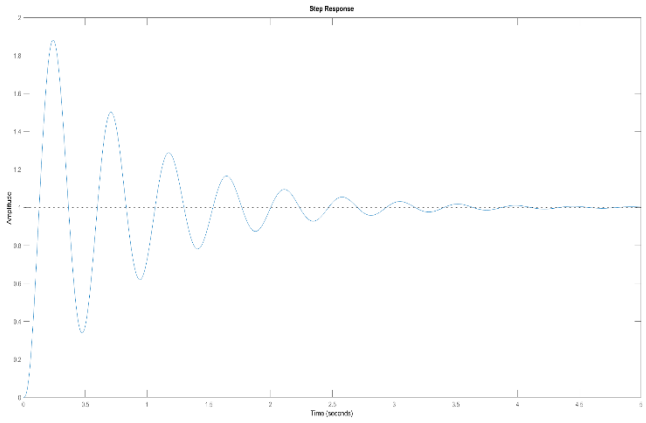
可见，系统的阶跃响应中不出现振荡和超调量，可是达到稳定需要很长时间，不适合于电梯模型。



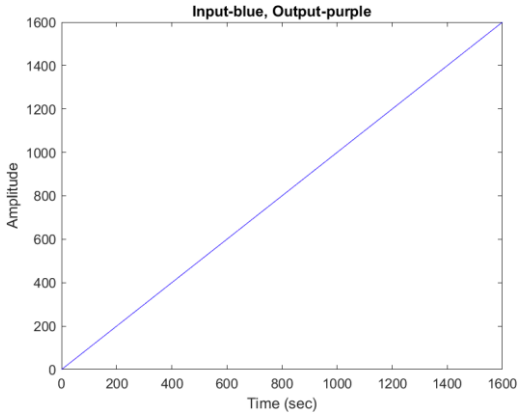
上图中反对角线上的紫色直线为输入信号，粉色直线为系统的输出。可见，系统稳定后，输入信号与输出信号不一致，存在稳态误差。

2. P 控制

K_P	1319.9898
K_I	0
K_D	0
超调量	88.2427
上升时间	0.0766
调节时间	3.3251
峰值时间	0.2398



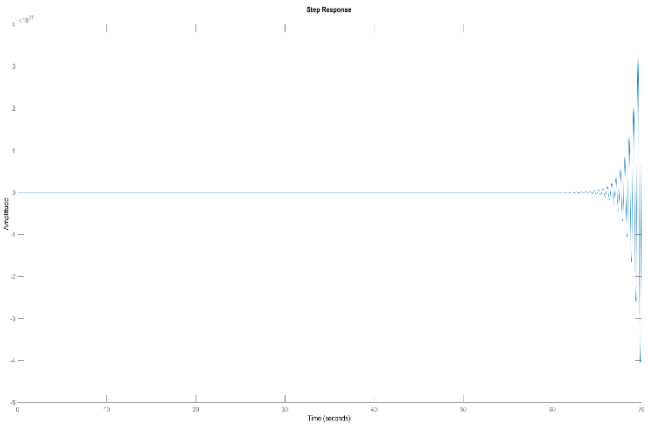
可见，引入$K_P = 1319.99$的 P 控制器后，调节时间减少到 3.3s，与$K_P = 11$ 时相比，系统的动态响应速度有明显的提高。然而，系统出现了振荡，且超调量达到了 88.24%。在实际应用中，超调量过大会导致安全问题，仅引入 P 控制器不满足需求。



由上图可见，系统的输入和响应几乎一致，加大K_p基本上消除了系统的稳态误差。

3. PI 控制

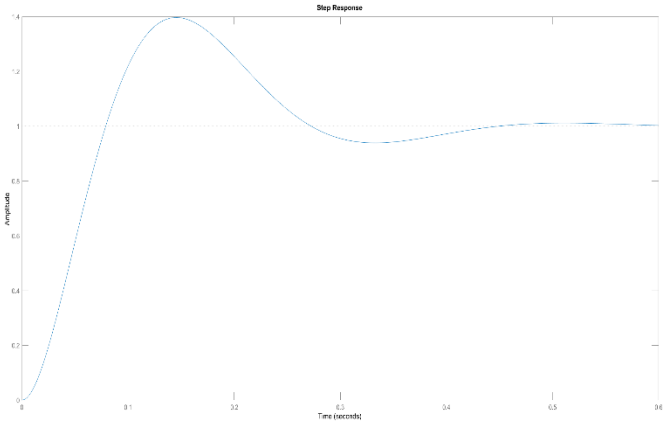
K_P	11879.9082
K_I	46347.9564
K_D	0
超调量	Nan
上升时间	Nan
调节时间	Nan
峰值时间	Inf



引入 PI 控制后，由于系统的阶次提高，系统的稳定性下降，调解过程中出现失稳。其实，该系统中 P 控制器已基本上能消除稳态误差，不需要引入积分器。

4. **PD 控制**

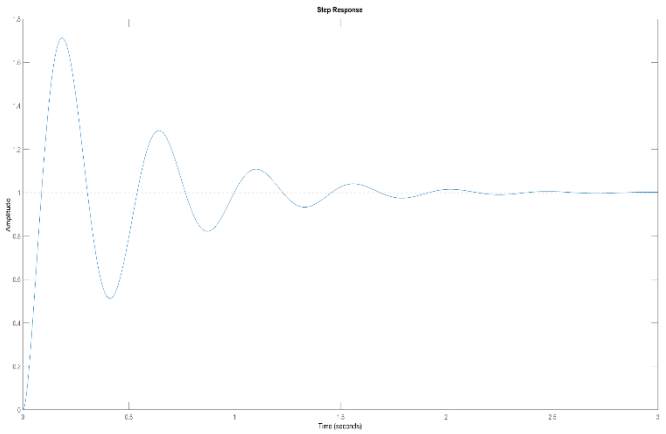
K_P	21119.8368
K_I	0
K_D	845.8495
超调量	39.7020
上升时间	0.0543
调节时间	0.4119
峰值时间	0.1461



在 P 控制的基础上加微分控制后，超调量从 88.2%减少到 39.7%，调节时间从 3.3s 减少到 0.4s。可见，微分控制有助于减少超调，克服振荡，使系统趋于稳定，并加快系统的动作速度，减少调节时间，从而改善系统的动态性能。

5. **PID 控制**

K_P	15839.8776
K_I	98875.6404
K_D	634.3870978
超调量	71.2028
上升时间	0.0607
调节时间	1.8386
峰值时间	0.1844

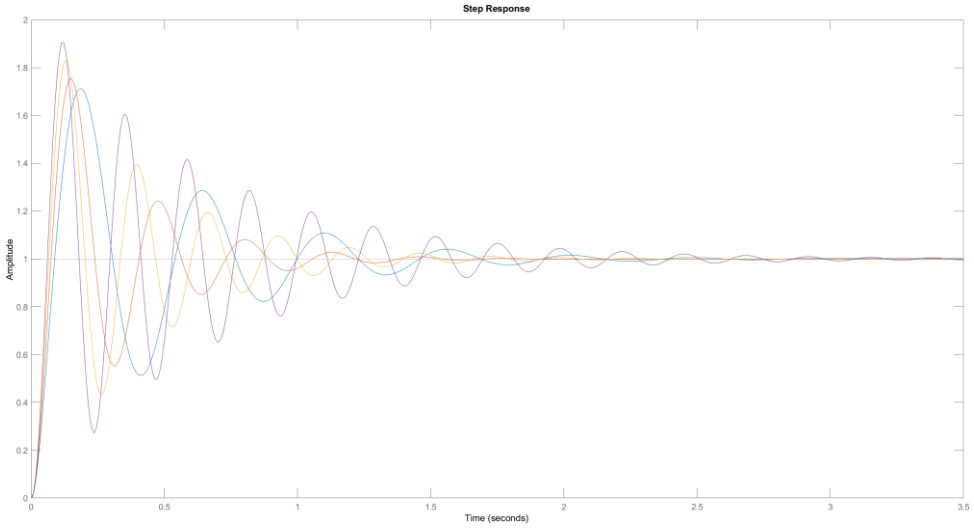


可见，引入 PID 控制器与 PD 控制器相比，调节时间和超调量均增大，系统的动态性能更差，不适合于实际的电梯系统。

6. **PID 控制：调节参数K_P**

K_P	15839.8776	25839.8776	35839.8776
K_I	98875.6404	98875.6404	98875.6404
K_D	634.3870	634.3870	634.3870
超调量	71.2028	75.3802	83.1437
上升时间	0.0607	0.0500	0.0437
调节时间	1.8386	1.1666	1.4827
峰值时间	0.1844	0.1450	0.1304

调节参数K_P，与公式得到的 PID 控制器对比

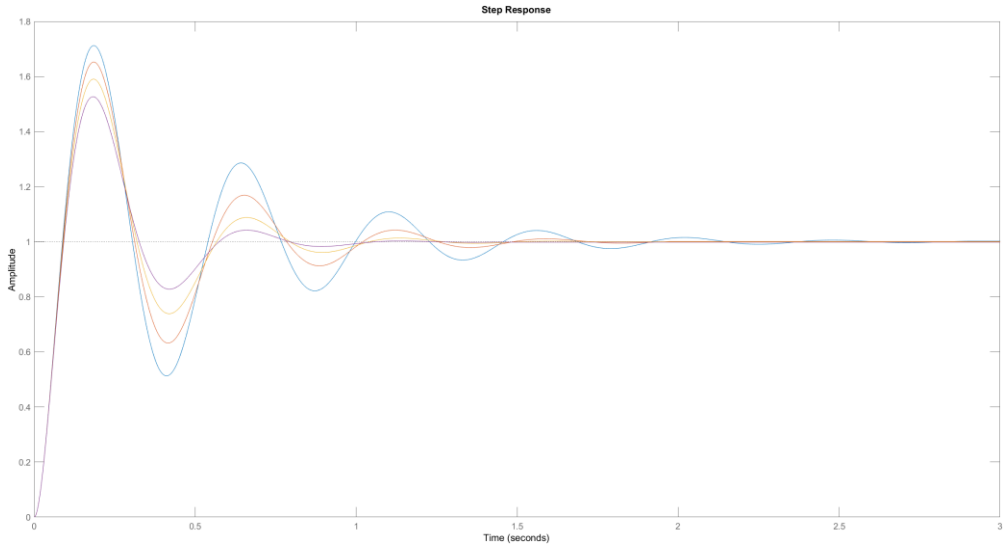


可见，比例系数K_P增加，稳态误差减少，超调量增大，振荡性更强。

7. PID 控制：调节参数K_I

K_P	15839.8776	15839.8776	15839.8776
K_I	98875.6404	78875.6404	58875.6404
K_D	634.3870	634.3870	634.3870
超调量	71.2028	65.2215	59.0331
上升时间	0.0607	0.0616	0.0625
调节时间	1.8386	1.3810	0.9813
峰值时间	0.1844	0.1857	0.1850

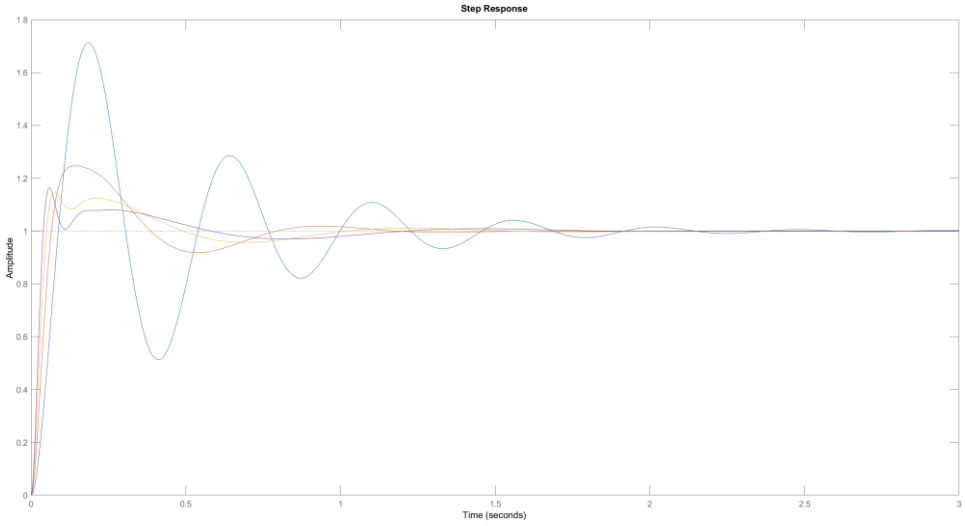
调节参数K_P，与公式得到的 PID 控制器对比



可见，K_I增加时，调节时间增加，响应速度变慢，超调量变大，振荡变得厉害。

8. PID 控制：调节参数K_D

K_P	15839.8776	15839.8776	15839.8776
K_I	98875.6404	98875.6404	98875.6404
K_D	634.3870	1634.3870	2634.3870
超调量	71.2028	24.7089	14.8682
上升时间	0.0607	0.0451	0.0336
调节时间	1.8386	0.1400	0.8937
峰值时间	0.1844	0.1400	0.0767



可见，微分增益K_D增加，阻尼增大，振荡减少，超调量减少。结果发现，$K_P = 15839.88$，$K_I = 98875.64$，$K_D = 2634.39$ 的 PID 控制器最适合于实际的电梯系统，其超调量不到 15%，调节时间也比较短。

六、讨论

1. PID 的优缺点

优点：具有原理简单，易于实现，适用面广，控制参数相互独立，参数的选定比较简单。

缺点： P、I、D 三者之间是线性组合关系，导致系统总是会出现“超调”、“震荡”等问题，而现有的数学工具还是不足以支撑我们找到一个“通解”。体现在实际的应用中，由于被控过程往往机理复杂，具有高度非线性、时变不确定性和纯滞后等特点，特别是在噪声、负载扰动等因素的影响下，过程参数甚至模型结构均会随时间和工作环境的变化而变化，最终导致系统无法满足控制需求。

2. PID 控制器参数整定方法

从本次实验可知，Ziegler-Nichols 方法不是万能的，以自行调整得到的K_P、K_I、K_D

为参数的 PID 控制的动态性能优于公式得到的 PID 控制。

在实际应用中，PID 控制器参数整定方法有：理论计算整定法和工程整定法。理论计算整定法主要是依据系统的数学模型，经过理论计算确定控制器参数。这种方法所得到的计算数据未必可以直接用，还必须通过工程实际进行调整和修改。工程整定方法主要依赖工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。PID 控制器参数的工程整定方法，主要有临界比例法、反应曲线法和衰减法。

七、代码

```
clear;
clc;
close all;
%%  Define Plant

n=[1 10];                  % Defining Numerator
d=[1 71 1070 1000 0];       % Defining Denominator
Gs=tf(n,d)                 % Transfer function=
Ts=feedback(Gs,1)

t=0:0.1:1600;
u = t;                       % Input
[y,t,~] = lsim(Ts,u,t); % Output
figure;
plot(t,y,'m',t,u,'b');
xlabel('Time (sec)'); ylabel('Amplitude');
title('Input-blue, Output-purple');

%figure;
%rlocus(Gs);        %根轨迹
%figure;
%step(Ts);
%stepinfo(Ts)

%%  Add PID Contoller

Ku = 26399.796;
Tu = 0.6532-0.3328;

c_type = ["P" "PI" "PD" "PID"]
for i=1:4
    [Kp,Ki,Kd] = ZieglerNichols(Ku,Tu,c_type(i));
    Gc = pid(Kp,Ki,Kd);
    Ts = feedback(Gs*Gc,1);
    figure;
    step(Ts);
    title(c_type(i));
    stepinfo(Ts)
end
```

```
%% Ziegler-Nichols Method

function [varargout] = ZieglerNichols(varargin)

%ZIEGLERNICHOLS Computes PID gains using Ziegler-Nichols
%
%   [KP, KI, KD] = ZIEGLERNICHOLS(KU, TU, TYPE) Computes the KP, KI, and KD gains
%   for a PID controller using the Ziegler-Nichols method.
%
%   TYPE:
%       'PID'
%       'P'
%       'PI'
%       'PD'
%       'PessenIntegrationRule'
%       'SomeOvershoot'
%       'NoOvershoot'
%
%INPUT:     -KU:    Ultimate gain that leads to steady oscillations
%           -TU:    Oscillation period (seconds)
%           -TYPE:  char array denoting the type of desired control
%
%OUTPUT:    -KP:    Proporational gain
%           -KI:    Integral gain
%           -KD:    Derivative gain
switch nargin
    case 3
        %User supplies all inputs
        KU      = varargin{1};
        TU      = varargin{2};
        TYPE    = varargin{3};

    case 2
        %Assume ClassicPID
        KU      = varargin{1};
        TU      = varargin{2};
        TYPE    = 'PID';

    otherwise
        error('Invalid number of inputs');
end

assert(TU > 0, 'TU should be a positive value')

if(strcmp(TYPE,'PID')==1)
    KP = 0.6*KU;
    Ti = TU/2;
    Td = TU/8;

elseif(strcmp(TYPE,'P')==1)
    KP = 0.5*KU;
```

```
    Ti = NaN;
    Td = NaN;

elseif(strcmp(TYPE,'PI')==1)
    KP = 0.45*KU;
    Ti = 0.8*TU;
    Td = NaN;

elseif(strcmp(TYPE,'PD')==1)
    KP = 0.8*KU;
    Ti = NaN;
    Td = TU/8;

elseif(strcmp(TYPE,'PessenIntegrationRule')==1)
    KP = 0.7*KU;
    Ti = 2*TU/5;
    Td = 3*TU/20;

elseif(strcmp(TYPE,'SomeOvershoot')==1)
    KP = KU/3;
    Ti = TU/2;
    Td = TU/3;

elseif(strcmp(TYPE,'NoOvershoot')==1)
    KP = 0.2*KU;
    Ti = TU/2;
    Td = TU/3;

else
    error('Unsupported TYPE')
end

%Compute KI and KD based on KP, TI, and TD
KI = KP/Ti;
KD = Td*KP;

%If KI or KD are NaN, change to 0 so it is compatible with simulations
if(isnan(KI))
    KI = 0;
end

if(isnan(KD))
    KD = 0;
end

%Package outputs
varargout{1} = KP;
varargout{2} = KI;
varargout{3} = KD;
end
```