



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY

## 《运动控制课程设计实验报告》

姓 名: 卓 東  
学 号: 520020910169  
小 组 成 员: 李 厚 霖  
学 号:  
指 导 教 师 姓 名: 宫 亮  
专 业 名 称: 自 动 化  
班 级: F2003202

二〇二三年 3 月 30 日



## 目录

一、 硬件选型.....	3
二、 开环控制.....	4
1. PLC 开环控制中的 $U_0$ .....	4
2. PLC 开环控制核心程序.....	4
三、 比例闭环.....	5
1. 比例闭环核心程序.....	5
2. 实验结果分析与结论.....	6
四、 比例积分.....	7
1. SPD 指令确定新周期的方法.....	7
2. 比例积分核心代码.....	8
3. 实验结果分析与结论.....	9
4. P、I 参数整定.....	13
五、 P、I、D 三个参数的整定以及对系统的作用 .....	14
1. P、I、D 三个参数的整定方法.....	14
2. P、I、D 参数对系统的作用.....	16
六、 本课程的心得和建议.....	16
附录 .....	17



## 一、硬件选型

本门课实验通过 PLC 实现异步电机的速度控制，并考察负载对其控制效果的影响。本门课程所使用到的硬件共有变频器、电动机、负载、PLC、反馈编码器、触摸屏。以下介绍每个硬件的详细参数。



PLC 控制系统示意图

(1) PLC

型号：三菱(Mitsubishi) FX3u-32M

(2) 变频器

型号：三菱(Mitsubishi) FR-A800

(3) 电动机

型号：西门子(Siemens) 1LE0001-0DB22-1AA4 Three-phase asynchronous motor

(4) 编码器

型号：光洋(Koyo) TRD-2E3600B-2M

(5) 负载

型号：Saar HK Electronic Limited, 113ZYT220-500 DC motor

(6) 人机界面

型号：三菱(Mitsubishi) GS2107-WTBD



## 二、开环控制

### 1. PLC 开环控制中的 $U_0$

在 PLC 开环控制中，我们将能够调节电机转速  $N$  的输入频率  $f$  设为  $U_0$ ，其中电机转速  $N$  的计算公式为：

$$N = \frac{f \times 60 \times (1 - s)}{p}$$

当我们设定理想转速  $N$  时，可以通过计算公式得到所需的输入频率：

$$U_0 = f = \frac{NP}{60 \times (1 - s)} = N \times \frac{5000}{1500}$$

但是在实际实验中我们发现，由于电机本身的转速具有限制，当输入频率  $f$  达到最大值 5000 时，最高转速只能达到 1485 rpm，因此我们需要校正  $U_0$  的计算公式，由于输入频率  $f$  与电机转速  $N$  成线性关系，校正后的  $U_0$  的计算公式应为：

$$U_0 = f = N \times \frac{5000}{1500} \times \frac{1500}{1485} = N \times \frac{5000}{1485}$$

在校正完毕后，当我们给定电机转速设定值  $N$  后，PLC 系统会自动换算成对应的输入频率  $f$ ，通过  $f$  对电机转速实现控制，从而实现电机的开环控制，使电机转速达到我们设定的转速值。

### 2. PLC 开环控制核心程序

开环控制核心程序如下图所示：



开环控制核心程序

这段程序的作用：当我们在人机界面按下开关(X001 导通)，PLC 程序会将我们设定的电机转速(使用 D10 寄存器进行保存)以 float 形式(float 形式的变量会保留小数，方便进行 DIV 运算)保存在 D14 里，通过校正后的计算公式  $U_0 = N \times \frac{5000}{1485}$ ，得到设定转速对应的输入频率  $f$ ，将对应的频率保存在 D20 寄存器上以显示在人

机界面上，方便我们验证校正公式是否正确。(其他开关和数据寄存器的注释请参考附录)

通过这段代码，我们可以将输入频率传输给电机，使电机进行相应转速的运动：





### 三、比例闭环

在实际实验中我们发现：当电机加上负载时，使用开环控制电机的实际转速不能很好地达到我们的设定值(通常实际值比设定值略低)。为了解决这个问题，我们将开环控制改为闭环控制，使用 PID 控制器对转速进行调节，为了探究 P、I 以及 D 这三个参数对系统的作用，我们分别进行了使用比例闭环和使用比例积分的两次实验。

在比例闭环中，我们电机的输入量  $U$  会随着时间变化，以实现电机转速的实时调节，计算公式为：

$$U(t) = U_0 + K_p e(t)$$

其中， $U_0$  为我们期望的电机转速  $N$  换算成的输入频率， $K_p$  为比例系数， $e = N - n$ ， $n$  为编码器检测到的电机的实际转速。又因为 SPD 指令每隔时间  $T$  对编码器数值进行读取一次，实际转速的获取并不是连续的，而是离散化的，因此我们需要采用离散化的 PID 控制器，在这里，离散化的比例闭环控制计算公式为：

$$U(k) = U_0 + K_p e(k)$$

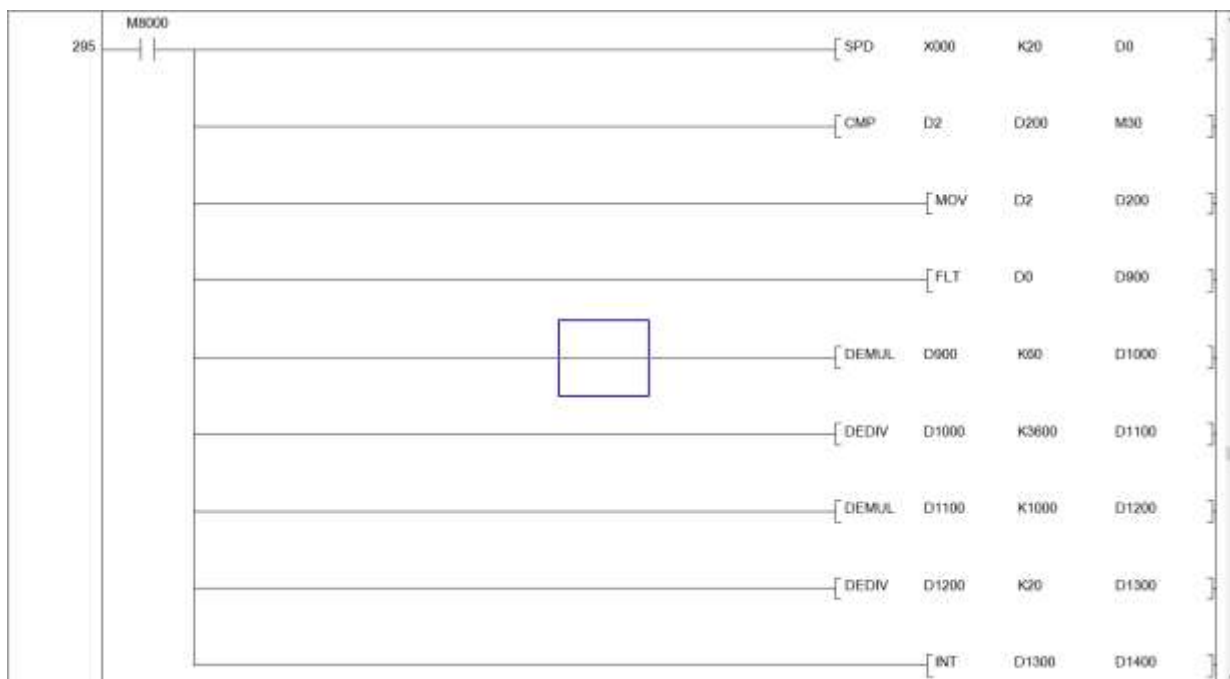
#### 1. 比例闭环核心程序

在进行比例闭环控制之前我们需要通过编码器获取电机实际的转速  $n$ ，转速  $n$  的计算公式为：

$$n = \frac{60 \times D_0}{C \times T} \times 1000$$

其中， $D_0$  是编码器记录的数值， $T=20\text{ms}$  为采样间隔， $C=3600$ 。

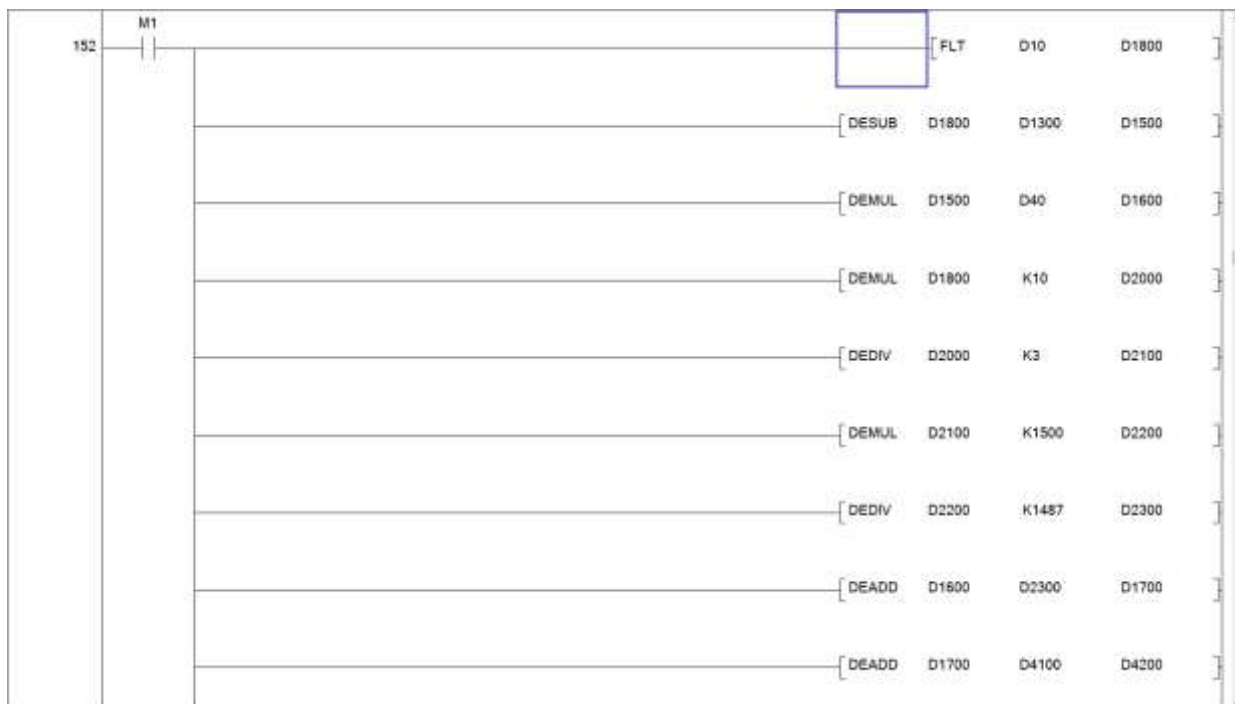
实际转速  $n$  的计算过程如下图所示：



实际转速  $n$  的计算过程

这段程序的作用：通过 SPD 指令每 20ms 采样一次编码器记录的数值  $D_0$ ，将数值保存在寄存器  $D_0$ 。将  $D_0$  以 float 形式(与开环控制中的作用一致)保存在寄存器  $D900$ ，通过计算公式得到测量到的实际转速  $n$  保存在寄存器  $D1300$  里，以 int 形式保存在寄存器  $D1400$  里，以显示在人机界面上，使得我们可以实时得到电机的实际转速，便于我们分析控制器的效果。(其他开关和数据寄存器的注释请参考附录)

比例闭环控制的核心程序如下图所示：



### 比例闭环控制的核心程序

这段代码的作用：首先从 D10 寄存器中获取我们设定的电机转速值 N，以 float 形式保存在 D1800 寄存器中，又因为我们在实际转速 n 的计算过程中将 float 形式的 n 保存在了 D1300 寄存器中，所以这里我们直接相减得到 e，保存在 D1500 寄存器中。通过从寄存器 D40 中提取到我们事先设定好的  $K_p$  值，将  $K_p$  与 e 的乘积保存在 D1600 寄存器中。由于 U 保存的是输入频率 f，因此我们需要对电机转速值 N 进行对应频率的换算，利用公式  $U_0 = f = N \times \frac{5000}{1500} \times \frac{1500}{1485} = N \times \frac{5000}{1485}$  得到  $U_0$  保存在寄存器 D2300 中，最后将  $U_0$  与  $K_p e$  相加得到最终的控制信号 U，保存在寄存器 D1700 里，最后再将 U 输入至电机中。(其他开关和数据寄存器的注释请参考附录)

## 2. 实验结果分析与结论

为了测试 P 控制器的效果，我们设计了几组实验来定性地分析  $K_p$  对电机转速的作用，实验表格如下所示：

$K_p$	设定转速	0 负载	50 负载	100 负载
0	575	574	572	568
0.5		574	573	569
1.5		575	574	570
0	1085	1084	1081	1070
0.5		1085	1081	1072
1.5		1085	1082	1074
0	1415	1414	1409	1395
0.5		1414	1410	1397
1.5		1415	1411	1401



从表格我们可以看到：

①当  $k_p$  和设定转速保持一定时，随着负载增加，实际转速会相较于我们设定的转速值下降，因此我们需要引入闭环控制来保证电机的实际转速不会随着负载的增加发生过多的下降。

②当  $k_p$  保持一定，设定的转速值增大时，对于负载高的电机，它的实际转速会发生较大地下降，即使使用  $k_p$  也不能完全地消除静差，因此对于这种情况，我们需要用到 PI 控制。

③当设定转速和负载保持一定时，使用 P 控制能够显著降低实际转速与设定转速之间的误差，对于 0 负载的情况下，甚至能够做到完全地跟踪设定转速。但是，对于高设定转速、高负载的电机，只使用 P 控制仍然会产生静差，无法完全做到误差为 0。

结论：使用比例闭环控制能够根据设定转速  $N$  与实际转速  $n$  之间的误差  $e$  来实时地调整控制量  $U$ ，从而达到实时控制电机转速的效果。使用比例闭环控制能够显著地减少电机在负载情况下实际转速与设定转速之间的偏差，达到跟踪设定转速的效果。但是，只使用 P 控制无法完全消除在高负载、高转速情况下的误差，因此需要采用 PI 控制。

## 四、比例积分

在实际实验中，我们会发现在电机处于高负载、高转速情况下，只采用比例闭环控制无法完全消除误差，难以实现输出能够追踪输入的效果。为了消除静差，我们需要引入积分控制，采用比例积分环节来控制电机转速。在采用比例积分环节的情况下，电机的控制量  $U$  的计算公式为：

$$U(t) = U_0 + K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

由于积分环节的原理是对  $e$  进行积分，因此需要记录每时每刻转速的设定值与实际值之间的误差  $e$ ，又因为 SPD 指令对编码器记录数值的采样间隔为  $T$ ，获取到的实际转速是离散化的，因此我们需要采用离散化的 pid 控制器。在这里，我们采用的是位置式 pid 控制器，它的计算公式为：

$$U(k) = U_0 + K_p e(k) + K_i \sum_{i=1}^k e(i)$$

### 1. SPD 指令确定新周期的方法

在采用位置式 pid 控制器时，我们会遇到一个问题：如果直接根据程序代码执行周期对误差  $e$  进行累积（即每次代码运行一次就计算一次  $e=N-n$  并与之前的累积相加），又因为程序代码执行完一次的时间极短，当设定值发生阶跃时，电机输出来不及消除误差，会导致误差快速积累，造成积分器迅速饱和，因此两次误差累积之间的时间间隔应该与电机转速采样间隔  $T$  相匹配。但是如果直接使用采样间隔  $T$  的具体数值（如 20ms）作为两次误差累积之间的时间间隔，又由于采样计时器存在一定误差，电机转速  $n$  的采样时间间隔  $T$  不一定为精确的 20ms，使得第二次计算误差时使用的测量值  $n$  可能仍为上一次计算误差时使用的测量值，而不是新周期的测量值，这会导致误差  $e$  不能很好地跟踪设定值与实际值之间实时的误差，使得控制器效果下降。为了保证每一次计算误差  $e$  时，使用的测量值  $n$  都是新周期的测量值，我们根据 SPD 指令的原理来对新周期进行了确定。SPD 代码如下所示：



每次执行 SPD 指令时，实际上会占用  $D0 \sim D2$  一共三个寄存器， $D0$  寄存器保存的是采样的编码器的数值， $D1$  寄存器保存的值在一个采样周期内会随着时间成阶梯式上升，直到新周期开始后从 0 继续开始，而  $D2$  寄

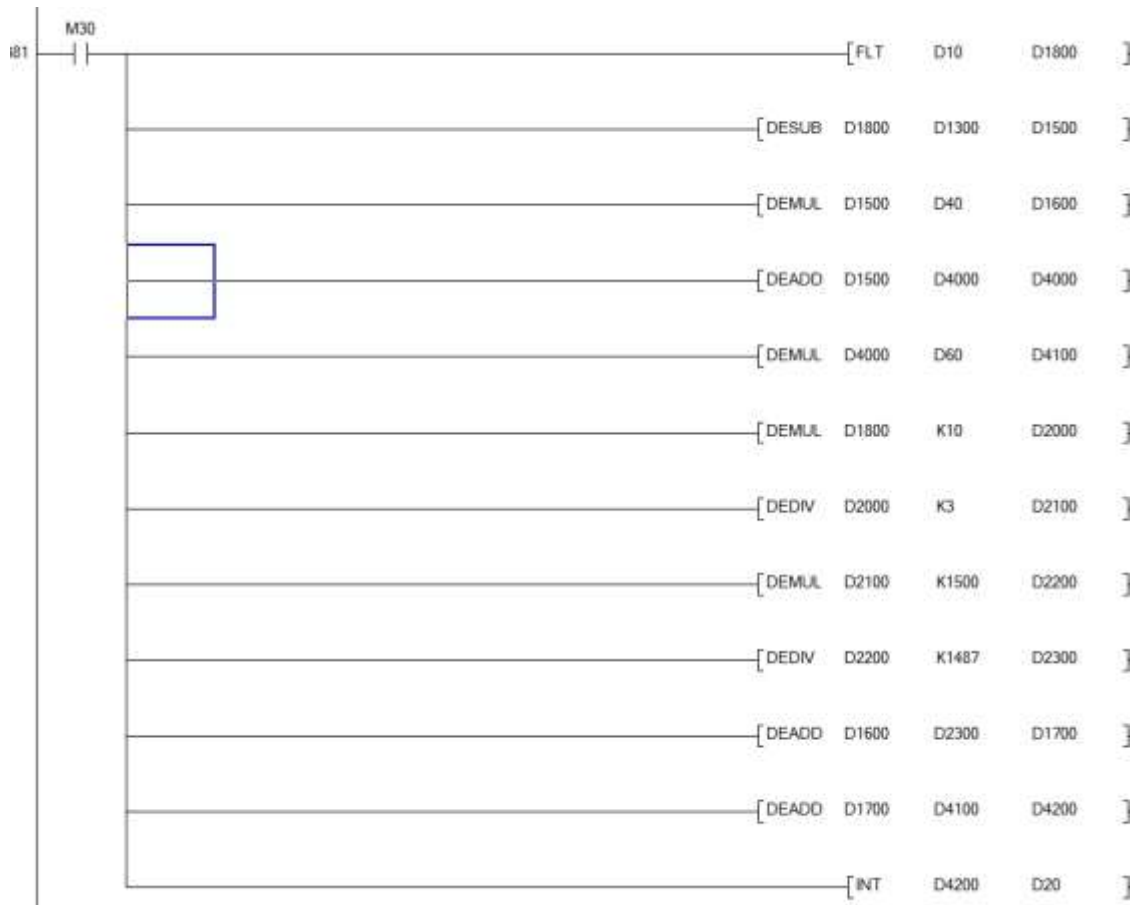




Line	Address	Instruction	Operand 1	Operand 2	Operand 3	Operand 4
295	M8000	[SPD	X000	K20	D0	]
		[CMP	D2	D200	M30	]
		[MOV	D2	D200		]

这段代码的作用为：每 20ms 采样一次编码器的数值(代表一次新周期开始)，比较当前时刻 D2 寄存器的值与上一时刻 D2 寄存器的值(上一时刻 D2 寄存器的值被保存在 D200 寄存器中)，若当前时刻 D2 寄存器的值大，代表新的周期已经开始，使 M30 导通(以进行误差累积任务)，若小于等于则关断 M30(代表新周期还未开始，不能执行误差累积任务)。将当前时刻的值保存在 D200 寄存器中作为下一时刻 D2 寄存器中的值比对的对象。通过这段程序，我们可以实现：每当一个新周期开始时，控制器进行一次误差累积，保证每次计算的误差都是新周期的误差，避免重复使用某一时刻采样的转速值来计算误差。(其他开关和数据寄存器的注释请参考附录)

比例积分控制的核心程序如下图所示:



### 比例积分控制的核心程序



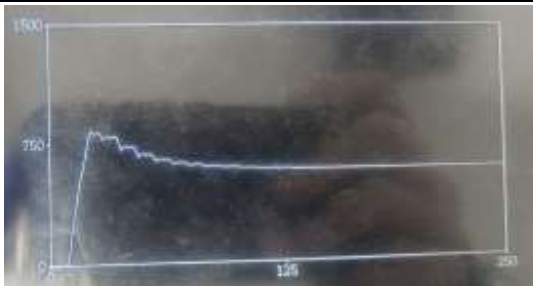
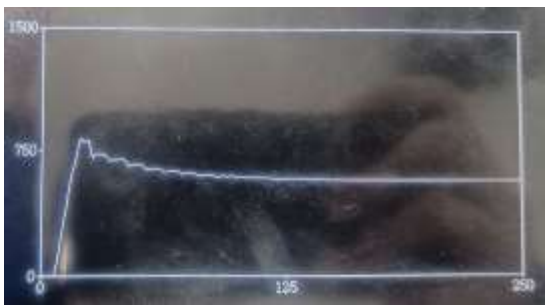



这段代码作用为：当 M30 导通时(新周期开始时)，将 D10 保存的数据(设定的转速值 N)以 float 形式保存在 D1800 寄存器内，将 D1800 寄存器的值(N)与 D1300 寄存器保存的值(n)相减得到误差 e，保存在寄存器 D1500 中。将当前时刻的 e 乘上 D40 寄存器保存的值( $K_p$ )得到  $K_p e$ ，保存在 D1600 寄存器中，将当前时刻的 e 与 D4000 保存的值(当前时刻之前累积的 e 值)相加得到当前时刻累积的 e 值，保存在 D4000 寄存器中。将当前时刻累积的 e 值乘以 D60 寄存器保存的值( $K_i$ )，得到  $K_i \sum_{i=1}^k e(i)$ 。再通过一系列的变换得到设定值 N 对应的输入频率  $f=U_0$ ，最后将三者相加，得到  $U = U_0 + K_p e + K_i \sum_{i=1}^k e(i)$ ，为电机的控制量，保存在 D4200 寄存器中，最终再把 D4200 以 int 形式保存在 D20，作为电机的控制量并显示在人机界面上。


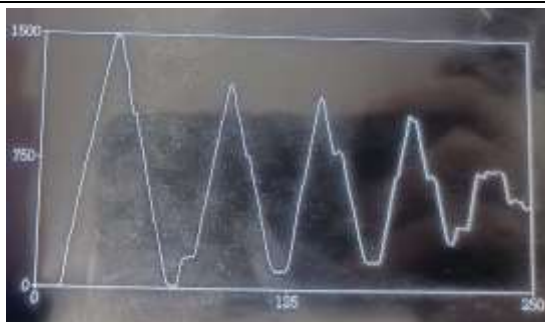
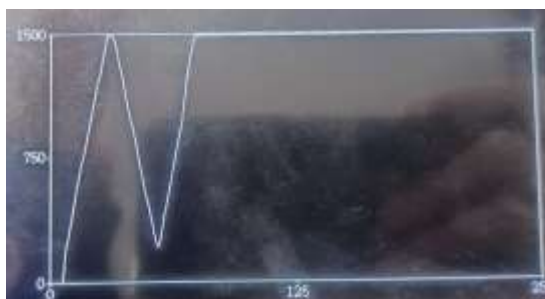
### 3. 实验结果分析与结论

为了测试 PI 控制器的效果，我们设计了几组实验来定性分析  $K_p$ 、 $K_i$  对电机转速的作用，实验表格如下所示：

设定转速 N 为 575

Kp	Ki	转速曲线	实测转速 (稳定后)
0.10	固定		575
0.50			575
1.00			575




固定	0.10		575
	0.50		574
	1.00		1488

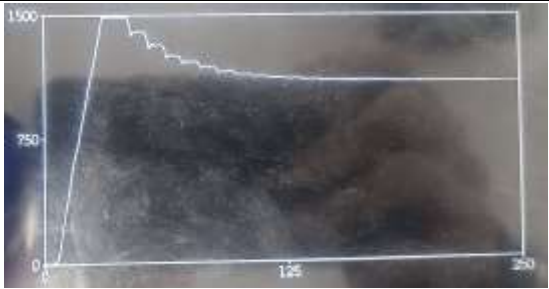
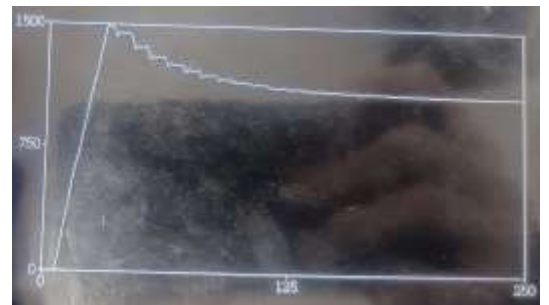

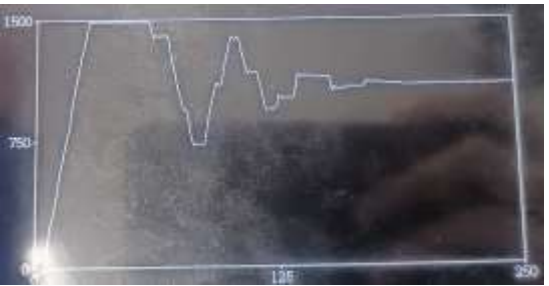
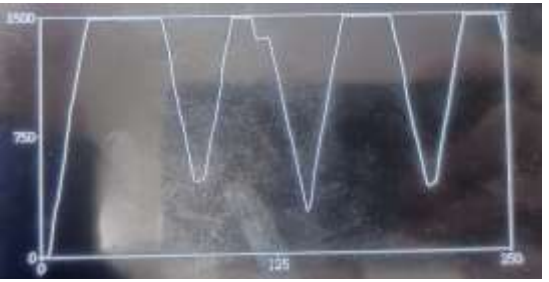
分析：①当设定转速值为 575 时，保持  $k_i$  固定为 0.10，当  $k_p$  增大时，我们可以发现系统的响应速度变快但超调量变大，并且在低转速情况下，三种  $k_p$  都能保证静态误差为零。

②当设定转速值为 575 时，保持  $k_p$  固定为 0.50，当  $k_i$  增大时，我们可以发现系统逐渐变得不稳定且输出曲线开始震荡，当  $k_i=1.00$  时，系统已经变得不稳定，输出曲线开始发散，误差变得很大。

设定转速  $N$  为 1085

$K_p$	$K_i$	转速曲线	实测转速 (稳定后)
0.10	固定		1086



0.50			1085
1.00			1084
固定	0.10		1086
	0.50		1085
	1.00		1486






分析：①当设定转速值为 1085 时，保持  $k_i$  固定为 0.10，当  $k_p$  增大时，我们可以发现系统的响应速度变快并且适当地增大  $k_p$  可以减少调节时间，但是设定转速在 1085 的情况下，由于设定值突然阶跃到一个特别大的值，导致计算的误差  $e$  特别大，使得积分器很快饱和，导致调节时间变长。

②当设定转速值为 1085 时，保持  $k_p$  固定为 0.50，当  $k_i$  增大时，我们可以发现系统逐渐变得不稳定且输出曲线开始震荡，当  $k_i=1.00$  时，系统已经变得不稳定，输出曲线开始发散，误差变得很大。并且设定转速在 1085 的情况下，由于设定值突然阶跃到一个特别大的值，导致计算的误差  $e$  特别大，使得积分



器很快饱和，导致调节时间变长。

设定转速  $N$  为 1415

Kp	Ki	转速曲线	实测转速 (稳定后)
0.10	固定		1415
0.50			1414
1.00			1413
固定	0.10		1414
	0.50		1413



	1.00		1486
--	------	--	------

分析：①当设定转速值为 1415 时，保持  $k_i=0.10$ ， $k_p$  增大。由于设定转速为 1415 已经接近电机转速上限，因此启动时设定值突然阶跃到一个特别大的值，导致计算的误差  $e$  特别大，误差累积的值特别大，使得积分器很快饱和，导致电机长时间以最高转速运行，需要较长时间才能消除偏差回到 1415 的转速，人机界面已经无法显示电机转速曲线从启动到稳定的全过程。

②当设定转速值为 1415 时，保持  $k_p=0.50$ ， $k_i$  增大。由于设定转速为 1415 已经接近电机转速上限，因此启动时设定值突然阶跃到一个特别大的值，导致计算的误差  $e$  特别大，误差累积的值特别大，使得积分器很快饱和，导致电机长时间以最高转速运行，需要较长时间才能消除偏差回到 1415 的转速，人机界面已经无法显示电机转速曲线从启动到稳定的全过程。并且当  $k_i=1.00$  时，电机转速长时间保持在最高转速，我们认为此时并不是积分器饱和的问题，而是系统变得不稳定，输出曲线发散的结果。

结论：①使用 PI 控制器能够消除只使用 P 控制无法消除的静态误差，使实际值在高转速、高负载的情况下仍然能够跟踪我们给定的设定转速，实现静态误差在  $-1\sim+1$  rpm 之内。

②但是使用位置式积分器可能会面临积分饱和的情况，即当设定值发生较大阶跃变化时，此时误差值累积特别快，控制器输出的控制量  $U$  的大小达到电机的极限但仍然不能消除偏差，使得系统输出需要很长时间才达到给定值。针对这种情况，我们可以采用积分分离法或有效偏差法等方法来降低积分器饱和的可能性。

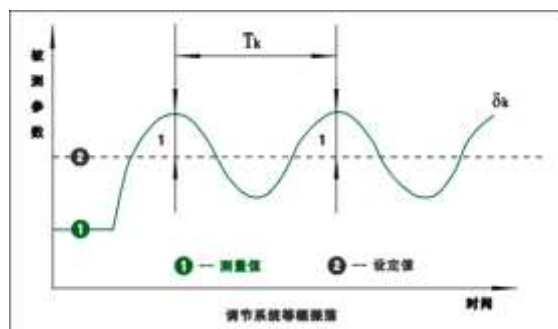
③此外，当  $k_i$  固定时，增大  $k_p$  能够加快系统的响应速度、减少静差，但是会增大超调量，降低系统稳定性，导致调节时间变长。但是适当增大  $k_p$  也可能会减少调节时间，使输出更快地稳定在给定值。

④当  $k_p$  固定时， $k_i$  增大能消除系统静差，但若  $k_i$  太大，则会导致系统逐渐变得不稳定且输出曲线开始震荡甚至发散，若  $k_i$  太小，则控制能力减弱，无法完全消除静差。

#### 4. P、I 参数整定

为了实现较好的控制效果，我们需要对 PI 控制器的参数进行整定，整定的方法很多，有：Ziegler-Nichol 响应曲线法、临界比例度法、衰减曲线法等，在这里我们选取的是临界比例度法。

临界比例度法：一个调节系统，在阶跃干扰作用下，出现既不发散也不衰减的等幅震荡过程，此过程称为等幅振荡过程，如下图所示。此时 PID 调节器的比例度为临界比例度  $\delta_k$ ，被调参数的工作周期为临界周期  $T_k$ 。



临界比例度法示意图

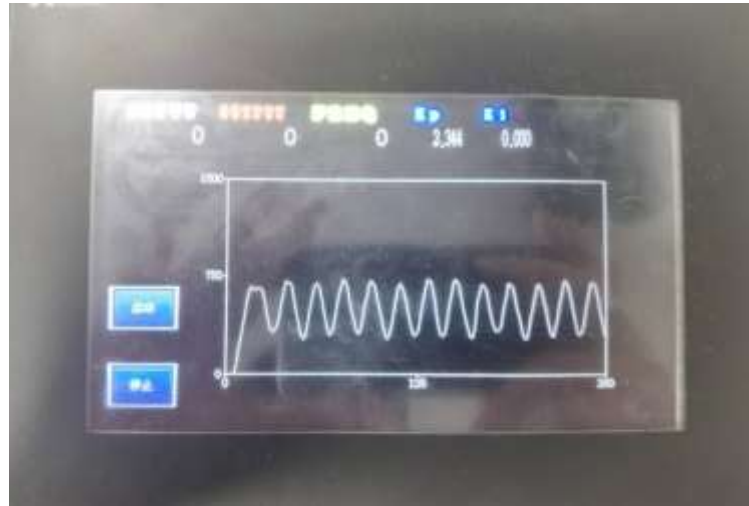




我们的具体做法如下：

①切断 I、D 控制，即令  $K_i$ 、 $K_d=0$ 。

②逐渐增大  $K_p$ ，使得输出曲线发生等幅振荡，在实验中，我们找到的使输出曲线等幅振荡的  $K_p$  值为  $K_p=2.344$ ，如下图所示：



$K_p=2.344$  时的输出曲线图

③取  $K_p=0.8*2.344=1.8752 \approx 1.875$ ，逐渐增大  $K_i$  值，直到获得较好的输出曲线为止，在这里我们组获得的最优的一组参数为： $K_p=1.875$ ， $K_i=0.180$ ，输出曲线如下图所示。



$K_p=1.875$ ， $K_i=0.18$  的输出曲线图

## 五、P、I、D 三个参数的整定以及对系统的作用

### 1. P、I、D 三个参数的整定方法

#### ①等幅振荡法

步骤：

(1) 先切除 PID 控制器中的积分与微分作用，取比例增益  $K_c$  较小值，并投入闭环运行；



- (2) 将  $K_c$  由小到大变化，对应于某一  $K_c$  值作小幅度的设定值阶跃响应，直至产生等幅振荡；  
(3) 设等幅振荡时振荡周期为  $T_{cr}$ 、控制器增益为  $K_{cr}$ ，再根据控制器类型选择以下 PID 参数。

控制规律	$K_c$	$T_i$	$T_d$
P	$0.5K_{cr}$		
PI	$0.45K_{cr}$	$0.83T_{cr}$	
PID	$0.6K_{cr}$	$0.5T_{cr}$	$0.12T_{cr}$

等幅振荡法 PID 参数选取表

## ②完全经验法

这种方法没有任何定量规律可循，凭借的是工程技术人员对控制系统与控制对象的工作机理、工作环境的熟悉，是一种粗糙的调参方法，一些定性的调参准则如下：

参数整定找最佳，从小到大顺序查  
先是比例后积分，最后再把微分加  
曲线振荡很频繁，比例度盘要放大  
曲线漂浮绕大湾，比例度盘往小扳  
曲线偏离回复慢，积分时间往下降  
曲线波动周期长，积分时间再加长  
曲线振荡频率快，先把微分降下来  
动差大来波动慢。微分时间应加长  
理想曲线两个波，前高后低 4 比 1  
一看二调多分析，调节质量不会低

## ③衰减曲线法

(1) 先把积分时间放至最大，微分时间放至零，使控制系统运行，比例度放至较大的适当值，“纯 P 降低比例度”，就是使控制系统按纯比例作用的方式投入运行。然后慢慢地减少比例度，观察调节器的输出及控制过程的波动情况，直到找出 4:1 的衰减过程为止。这一过程就是“找到衰减 4:1”。

(2) 对有些控制对象，用 4:1 的衰减比感觉振荡过强时，这时可采用 10:1 的衰减比。但这时要测量衰减周期是很困难的，可采取测量第一个波峰的上升时间  $T_r$ ，其操作步骤同上。

(3) 根据衰减比例度  $s$  和衰减周期  $T_s$ 、 $T_r$  按表进行计算，求出各参数值。

控制规律	$\delta$	$T_i$	$T_d$
P	$\delta_s$		
PI	$1.2\delta_s$	$0.5T_i$	
PID	$0.8\delta_s$	$0.3T_i$	$0.1T_i$

4:1 衰减曲线法 PID 参数选取表

控制规律	$\delta$	$T_i$	$T_d$
P	$\delta_s$		
PI	$1.2\delta_s$	$2T_r$	
PID	$0.8\delta_s$	$0.3T_r$	$0.1T_r$

10:1 衰减曲线法 PID 参数选取表





#### ④响应曲线法

响应曲线法 PID 参数整定步骤：

- (1) 在手动状态下，改变输入（通常采用阶跃变化），记录被控变量的响应曲线；
- (2) 由开环响应曲线获得单位阶跃响应曲线，并求取“广义对象”的近似模型与模型参数；
- (3) 根据控制器类型与对象模型，选择 PID 参数并投入闭环运行。在运行过程中，可对增益作调整。

由于广义对象的响应曲线可以用“一阶+纯滞后”来近似，所以，如下 Ziegler-Nichols 参数整定方法可以使用：

控制规律	$K_c$	$T_i$	$T_d$
P	$\frac{1}{k_p} \times \frac{T_p}{\tau}$		
PI	$0.9 \times \frac{1}{k_p} \times \frac{T_p}{\tau}$	$3.3\tau$	
PID	$1.2 \times \frac{1}{k_p} \times \frac{T_p}{\tau}$	$2.2\tau$	$0.5\tau$

Ziegler-Nichols 参数整定表

其中， $k_p = \frac{\Delta Output}{\Delta Input}$ ， $\tau$  为纯滞后时间， $T_p$  为惯性时间常数。

## 2. P、I、D 参数对系统的作用

比例参数（ $K_p$ ）的变化会影响系统的响应速度和稳定性。 $K_p$  增大可以加快系统响应速度，减小静差，但可能导致超调量和振荡增大，稳定性变差，调节时间变长。 $K_p$  减小可以降低超调量和振荡，提高稳定性，但可能导致响应速度变慢，静差增大。

积分参数（ $K_i$ ）的变化会影响系统的精确性和过渡过程。 $K_i$  增大可以消除静差，提高精确性，但可能导致过冲和振荡增大甚至发散，响应速度变慢，稳定性变差。 $K_i$  减小可以降低过冲和振荡，加快响应速度，提高稳定性，但可能导致静差增大，精确性降低。

微分参数（ $K_d$ ）的变化会影响系统的抗干扰能力和噪声敏感性。 $K_d$  增大可以抑制超调和振荡，提高抗干扰能力，适用于迟滞或无阻尼系统，但可能放大噪声或干扰，使系统失稳。 $K_d$  减小可以减小噪声或干扰的影响，使系统更平滑和稳定，但可能导致超调和振荡增大，抗干扰能力降低。

## 六、本课程的心得和建议

运动控制系统课程设计是一门综合性的课程，要求我们运用所学的理论知识和实践技能，针对一个具体的控制对象(电机)进行系统设计、硬件选型、仿真分析和调试验证。该课程可以让我们掌握 PLC 控制器的原理以及应用，为我们今后从事相关领域的工作或研究打下坚实的基础。

该课程也有一些难点和挑战，如 PID 控制器的设计和优化、硬件的选型和连接、PID 参数的整定与调试等，需要我们有较强的理论基础和实践能力，以及良好的团队合作和沟通能力。

总之，运动控制系统课程设计是一门有趣而有意义的课程，让我们在实践中学习，在学习中实践，体验到运动控制系统的魅力和价值，为我们今后的发展奠定了坚实的基石。



## 附录

该附录包含代码中出现过的所有寄存器以及开关的注释，具体如下：

元件编号	注释
X001	启动按钮，当按下启动按钮闭合开关 X001，进行开关状态、寄存器数据的初始化。
X002	停止按钮，当按下停止按钮闭合开关 X002，清空其他元件的状态。
M1	开环控制以及比例闭环控制的主程序开关，当电机启动时就处于闭合状态
M30	比例积分控制的主程序开关，每到一个新周期闭合一次
M33	用于停止 V0 计数的开关，当 V0 大于 250 时，闭合 M33
M42	用于防止输入频率低于 0 的开关，当输入频率 f 低于 0 时，闭合 M42，令 f=0
M43	用于防止输入频率高于上限 5000 的开关，当输入频率 f 大于 5000 时，闭合 M43，令 f=5000
V0	用于计数，当按下启动按钮后，每隔 20ms 加 1，将 D1400 中保存的实际转速的 n 值保存在 D3000V0
D0	用于保存当前时刻采样到的编码器的值
D2	用于保存当前时刻 SPD 指令输出到 D2 寄存器中的值
D10	用于保存设定转速 N 的值
D14	开环控制中用于保存设定转速 N 的 float 形式
D20	用于保存控制量(电机输入频率)U=f 的值
D200	用于保存上一周期 D2 寄存器中的值
D3000~D3250	用于保存 0T 到 250T 之间每一时刻的实际转速值 n，以获得输出曲线， $T \approx 20\text{ms}$
D1800	比例积分控制中用于保存设定转速 N 的 float 形式
D1000	用于保存当前时刻采样到的编码器的值的 float 形式
D1100	保存 $D0/60$ 的值
D1200	保存 $D1100 \times 1000$ 的值
D1300	保存 $D1200/20$ 的值，即保存当前时刻检测到的实际转速 $n = \frac{60 \times D_0}{3600 \times 20} \times 1000$ 的最终值
D1400	保存 n 的 int 形式
D1500	保存设定转速值 N 与实际转速值 n 的差值 e
D40	保存 $K_p$ 的值
D1600	保存 $K_p e$ 的值
D4000	保存 $\sum_{i=1}^k e(i)$ 的值
D60	保存 $K_i$ 的值
D4100	保存 $K_i \sum_{i=1}^k e(i)$ 的值
D2000	保存 $N \times 5000$ 的值
D2100	保存 $D2000/1500$ 的值，即保存的是未校正的设定转速对应的频率 f 的值
D2200	保存 $f \times 1500$ 的值
D2300	保存 $D2200/1487$ 的值，即保存的是校正后的设定转速对应的频率 $f=U_0$ 的值
D1700	保存 $U_0 + K_p e$ 的值
D4200	保存 $U_0 + K_p e + K_i \sum_{i=1}^k e(i)$ 的值



上海交通大學  
SHANGHAI JIAO TONG UNIVERSITY