

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 《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](#_Toc25842)

[二、 开环控制 4](#_Toc25651)

[1.PLC开环控制中的U0 4](#_Toc1381)

[2.PLC开环控制核心程序 4](#_Toc16095)

[三、 比例闭环 5](#_Toc3444)

[1.比例闭环核心程序 5](#_Toc31259)

[2.实验结果分析与结论 6](#_Toc3749)

[四、 比例积分 7](#_Toc22132)

[1.SPD指令确定新周期的方法 7](#_Toc28652)

[2.比例积分核心代码 8](#_Toc3400)

[3.实验结果分析与结论 9](#_Toc669)

[4.P、I参数整定 13](#_Toc23127)

[五、 P、I、D三个参数的整定以及对系统的作用 14](#_Toc3112)

[1.P、I、D三个参数的整定方法 14](#_Toc27426)

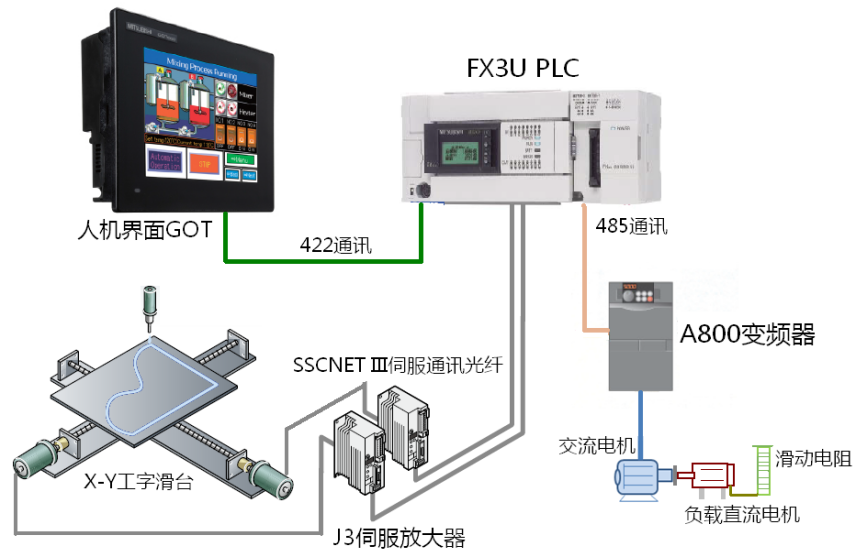
[2.P、I、D参数对系统的作用 16](#_Toc21151)

[六、 本课程的心得和建议 16](#_Toc10715)

[附录 17](#_Toc27201)

## 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

## 开环控制

### 1.PLC开环控制中的U0

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U0，其中电机转速N的计算公式为：

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：

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U0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U0的计算公式应为：

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，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：



## 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，以实现对电机转速的实时调节，计算公式为：

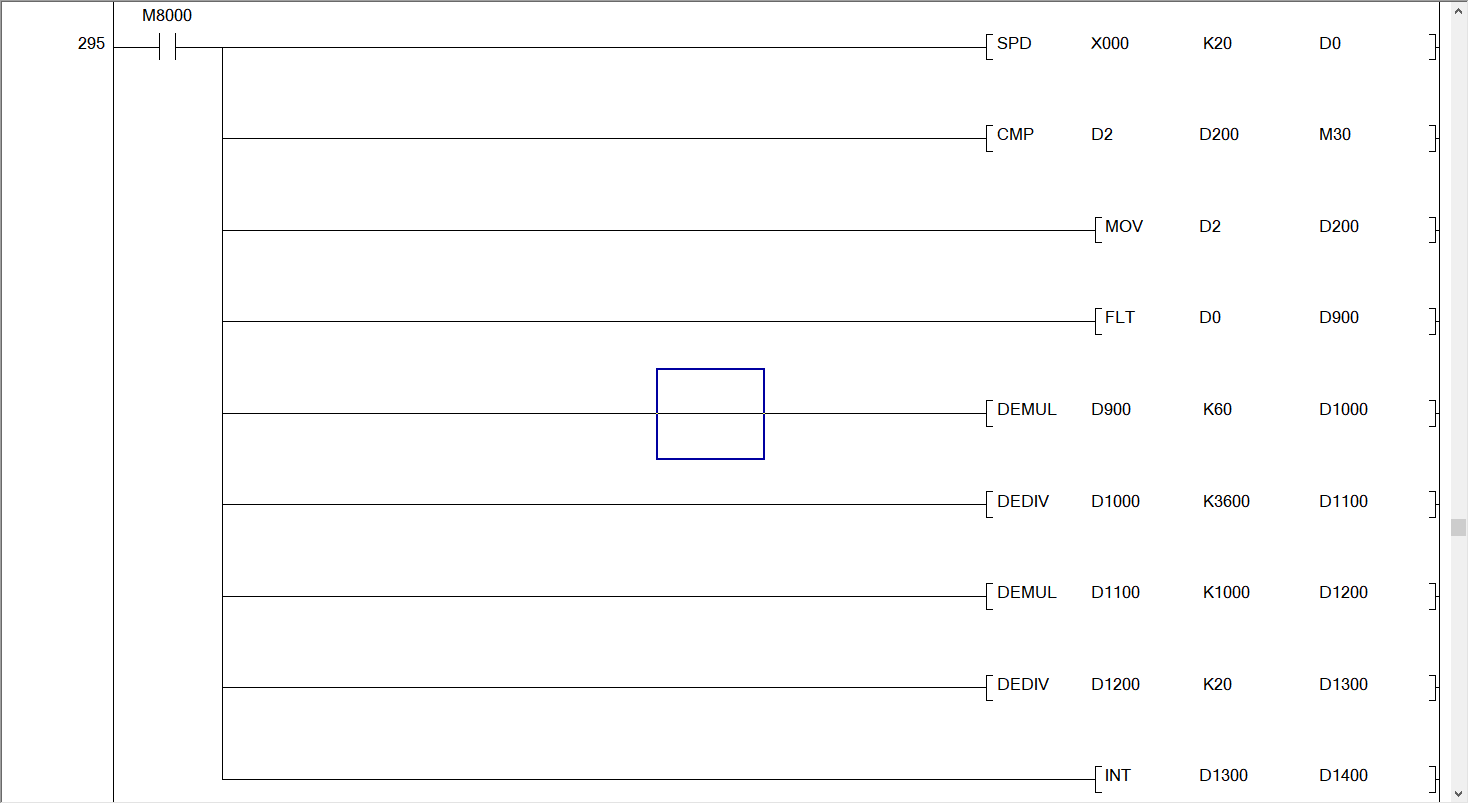
其中，U0为我们期望的电机转速N换算成的输入频率，Kp为比例系数，e = N - n，n为编码器检测到的电机的实际转速。又因为SPD指令每隔时间T对编码器数值进行读取一次，实际转速的获取并不是连续的，而是离散化的，因此我们需要采用离散化的PID控制器，在这里，离散化的比例闭环控制计算公式为：

### 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：

其中，D0是编码器记录的数值，T=20ms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D0，将数值保存在寄存器D0。将D0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Kp值，将Kp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 得到U0保存在寄存器D2300中，最后将U0与Kpe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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kp | 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 |

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Kp对电机转速的作用，实验表格如下所示：

从表格我们可以看到：

①当kp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kp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kp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。

## 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：

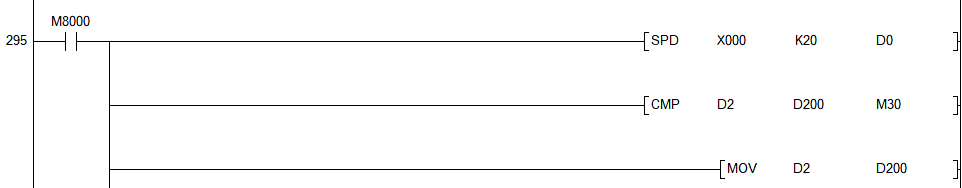
由于积分环节的原理是对e进行积分，因此需要记录每时每刻转速的设定值与实际值之间的误差e，又因为SPD指令对编码器记录数值的采样间隔为T，获取到的实际转速是离散化的，因此我们需要采用离散化的pid控制器。在这里，我们采用的是位置式pid控制器，它的计算公式为：

### 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~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寄存器保存的值在一个采样周期内近似于线性下降，当D2寄存器保存的值降到0后，会重新触发SPD指令，使得新周期开始。因此在SPD的指令里，D2寄存器起到一个计时器的作用，当D2寄存器中保存的值降至0后，会开始一个新周期，根据这个原理，我们可以设计一个新周期的检测程序，具体代码如下图所示：

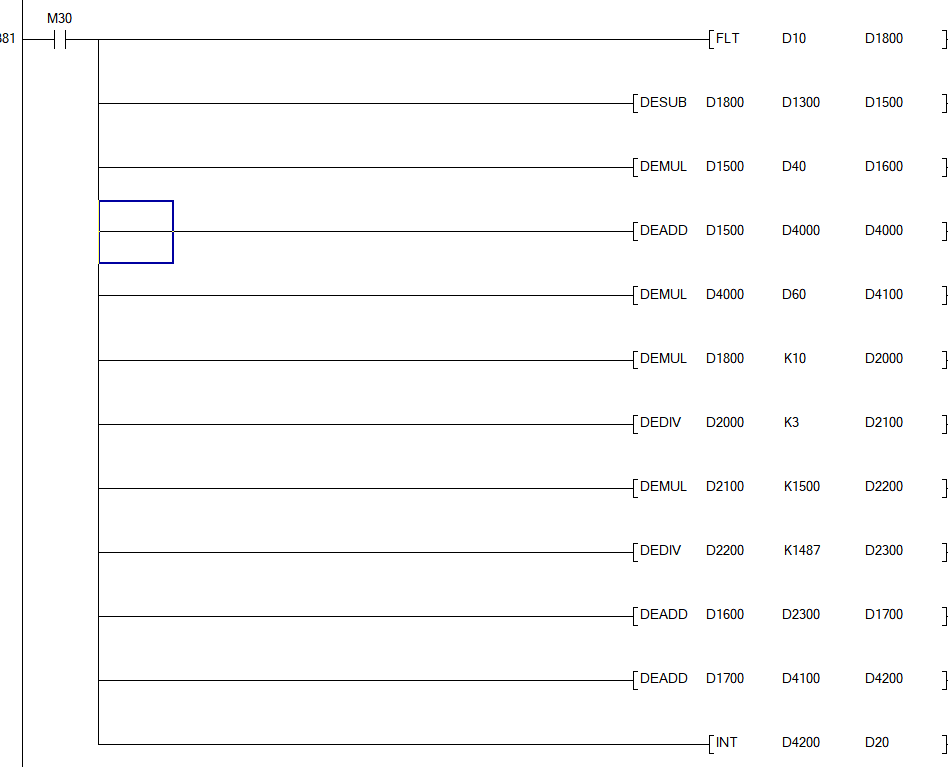


采样器新周期检测程序

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)

### 2.比例积分核心代码

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(Kp)得到Kpe，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(Ki)，得到。再通过一系列的变换得到设定值N对应的输入频率f=U0，最后将三者相加，得到，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地分析Kp、Ki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ki固定为0.10，当kp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kp都能保证静态误差为零。

②当设定转速值为575时，保持kp固定为0.50，当ki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ki=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

|  |  |  |  |
| --- | --- | --- | --- |
| Kp | Ki | 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ki固定为0.10，当kp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kp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kp固定为0.50，当ki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ki=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ki=0.10，kp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kp=0.50，ki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ki=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~+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。

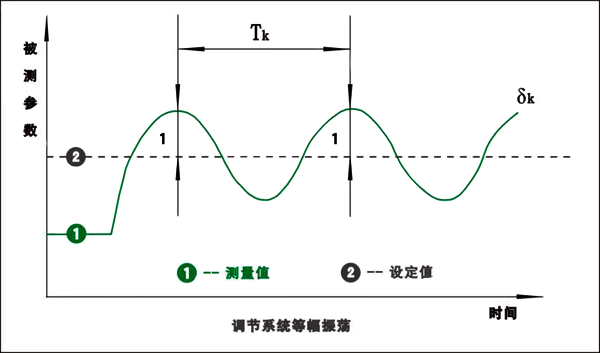
③此外，当ki固定时，增大kp能够加快系统的响应速度、减少静差，但是会增大超调量，降低系统稳定性，导致调节时间变长。但是适当增大kp也可能会减少调节时间，使输出更快地稳定在给定值。

④当kp固定时，ki增大能消除系统静差，但若 ki 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ki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δk，被调参数的工作周期为为临界周期Tk。

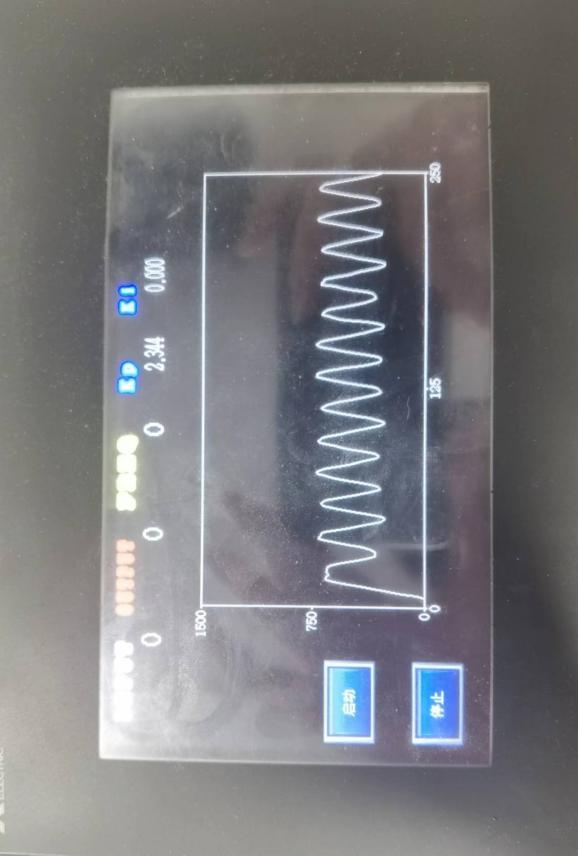


临界比例度法示意图

我们的具体做法如下：

①切断I、D控制，即令Ki、Kd=0。

②逐渐增大Kp，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Kp值为Kp=2.344，如下图所示：



Kp=2.344时的输出曲线图

③取Kp=0.8\*2.344=1.8752≈1.875，逐渐增大Ki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：Kp=1.875，Ki=0.180，输出曲线如下图所示。



Kp=1.875，Ki=0.18的输出曲线图

## 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

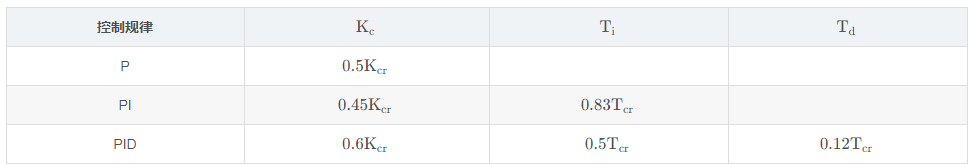
①等幅振荡法

步骤：

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

（2）将Kc由小到大变化，对应于某一Kc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Tcr、控制器增益为Kcr ，再根据控制器类型选择以下PID参数。



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

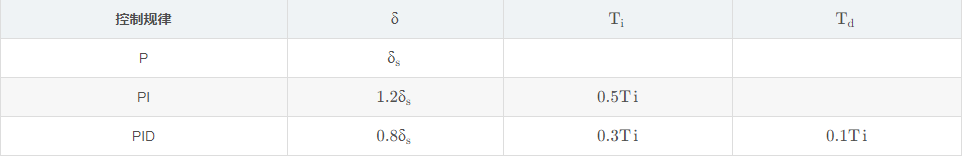
一看二调多分析，调节质量不会低

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Tr，其操作步骤同上。

（3）根据衰减比例度s和衰减周期Ts、Tr按表进行计算，求出各参数值。



4:1衰减曲线法PID参数选取表



10:1衰减曲线法PID参数选取表

④响应曲线法

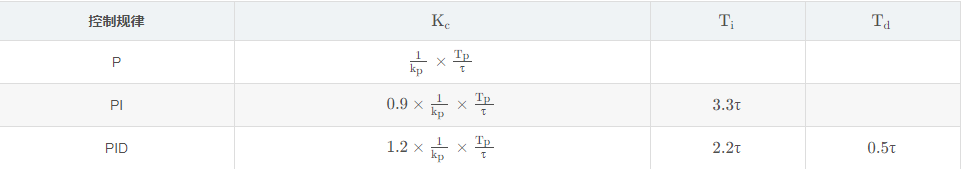
响应曲线法PID参数整定步骤：

（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：



Ziegler-Nichols参数整定表

其中，kp=，τ为纯滞后时间，Tp为惯性时间常数。

### 2.P、I、D参数对系统的作用

比例参数（Kp）的变化会影响系统的响应速度和稳定性。Kp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。Kp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（Ki）的变化会影响系统的精确性和过渡过程。Ki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。Ki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（Kd）的变化会影响系统的抗干扰能力和噪声敏感性。Kd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。Kd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。

## 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≈20ms |
| 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\*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Kp的值 |
| D1600 | 保存Kpe的值 |
| D4000 | 保存的值 |
| D60 | 保存Ki的值 |
| D4100 | 保存Ki的值 |
| D2000 | 保存N\*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\*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=U0的值 |
| D1700 | 保存U0+Kpe的值 |
| D4200 | 保存U0+Kpe+Ki的值 |