过程控制综合实践 I

PLC-DCS 综合设计实习报告

课程名称：过程控制综合实践 I：PLC-DCS 综合设计实习

老　　师：何王勇

班　　级：220211

学　　院：未来技术学院

学　　号：20211003337

姓　　名：曾康慧

小组序号：A4

实习日期：2023 年 12 月

目录

一、实习目标和任务简述

1. 实习目标

该课程以现代过程控制原理与应用技术为基础，运用 PLC（可编程逻辑控制器）技术进行 DCS（集散控制系统）架构下过程控制系统的设计和集成，完成以案例为设计目标的过程控制系统分析、设计、控制、投运。主要包括：**过程控制系统工艺分析、过程控制系统方案设计、过程检测和执行通道配置、PLC 控制程序设计、系统参数整定和投运**。通过实践教学，达到学生在借助现代先进控制工具和软件基础上，能灵活运用《现代过程控制原理与应用技术》理论知识**在 DCS 框架下运用现代化工具 PLC 进行复杂过程对象的控制系统设计**，培养学生解决复杂控制问题的能力。同时培养学生在**工程实践中的沟通、团队协作和项目管理能力，并考虑具体实施中的安全、健康、法律、文化和环境等因素约束条件**；拓宽学生的知识面，培养学生分析问题、解决问题的能力，启动学生的创造性思维。

具体课程目标如下：

(1) 了解案例问题的系统组成、控制需求、控制任务、控制工艺以及将要实施的控制系和技术的先进性情况；

(2) 能灵活综合运用所学知识，在自学 PLC 技术基础上开展控制系统的系统建模、控制工艺分析和控制方案设计与讨论，培养具备依据控制需求确定最佳工程实施方案和技术路线的能力；

(3) 培养具备运用 PLC 技术进行 DCS 结构下复杂过程控制系统的设计及投运的能力，并完成具体实施工作，如控制策略选取、系统搭建、软件设计、系统投运；

(4) 掌握 DCS 系统下过程控制系统设计、集成、投运的一般步骤和方法，培养具备 PID 单回路控和复杂控制系统的控制程序开发，参数整定和系统投运，过程检测和控制通道调校，安全连锁设计，DCS 系统下控制系统工程实施的能力；

(5) 充分发挥团队协作能力开展复杂控制任务的分解和协同工作，在考虑解决方案对社会、健康、安全、法律以及文化的影响基础上进行不断分析、评判和优化，具备利用 PLC 技术构建集散系统，并具有考虑非技术因素下的对系统进行全面优化设计的能力。

2. 设计内容

以实验室水箱实验装置为实验对象，PLC 作为现场控制站控制单元，触摸屏作为现场操作监控单元，构建基于 PLC 的 DCS 单站（多站）系统。开展涉及如下内容的设计工作：

(1) 系统结构组成与电气原理认识，含水路流通管路、传感器和执行器特性和性能、接线理图；

(2) 完成单容水箱、扰动的系统辨识；以 PLC 提取数据，Matlab 中进行模型辨识，给出被控对象的传递函数。

(3) 开展任意参数的**单回路**（压力、流量、液位），**串级**（液位-压力、液位-流量、流量-压力、），**前馈-反馈、比值控制。**要求：1-画出 PI&D（管道工艺流程图）方案（需要进行综合分析，选取较好的控制方式）、2-给出程序设

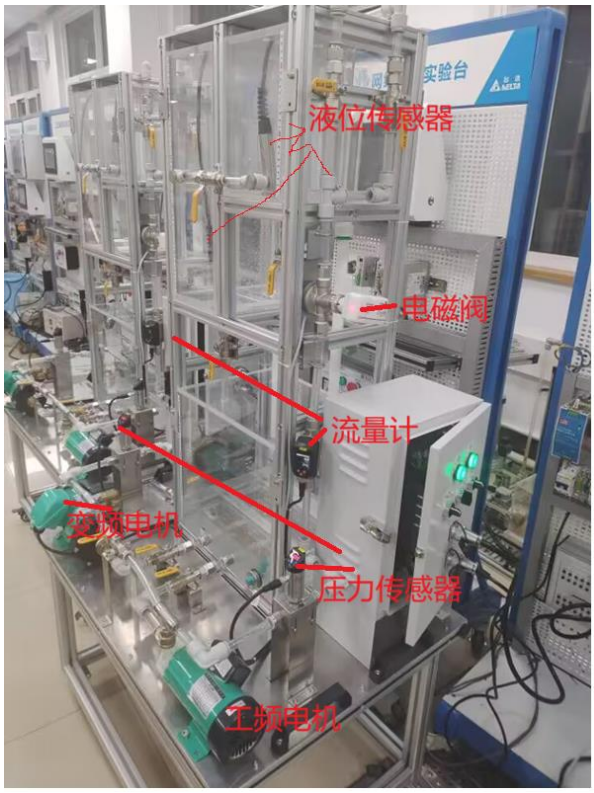
计思路、3-给出 PID 参数调整　结果和参数变化曲线图（扰动作用下 4：1 曲线，要求被控参数和阀门/变频器度曲线一起展示）。**5-其中（2）中的对象辨识结果，要求在其中 1 类控制中要有所体现（Matlab 中虚拟调试 PID 结果与实际控制 PID 结果需要进行对比）。控制系统设计中需要团队协作和沟通，组织肩负项目管理作用。**

(4) 给出简单图形界面

二、控制系统简介

1. 可变多容水箱设备和执行器

该水箱设备的主要由三个水箱容器、一个工频泵（定频泵）、一个变频泵、一个电动调节阀和若干手动调节阀组成。



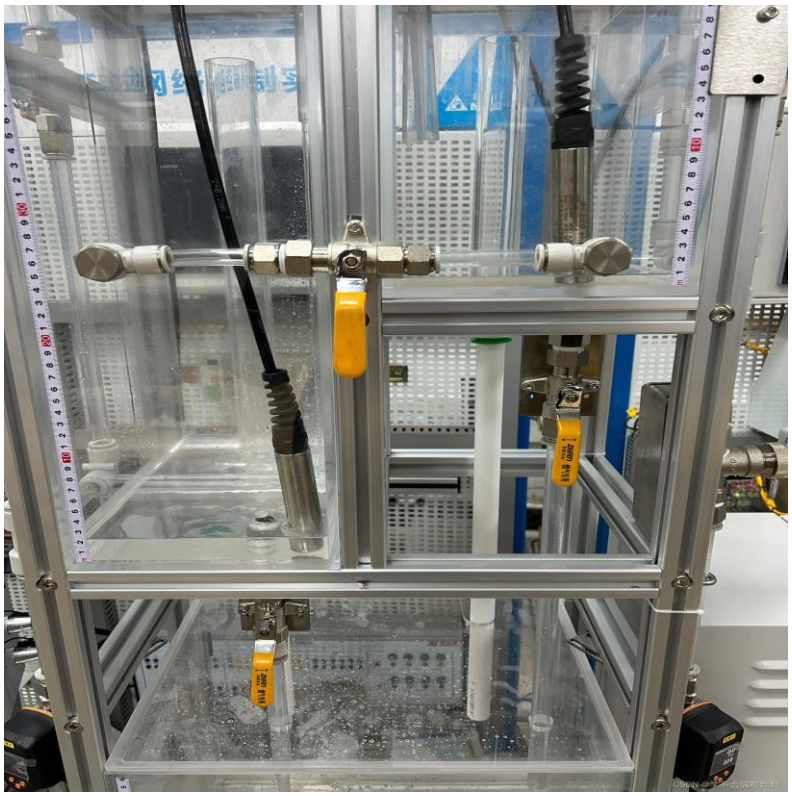
物理器件绑定至 PLC 软件上如图所示：

Physical View

Name	L..	Position	Version	Description
5AP1130.156C-000			2.0.1.0	AP1130 TFT C FHD 15.6in T
5PPC3100_KBU1_000		SY1	2.2.5.0	PPC3100 i3-7100U 2C 2.4/2.1GHz 3MB 15W
ETH1 --> GPOS		IF3		Ethernet
ETH2		IF4		Ethernet
ETHinternal		IF9		Ethernet
USB1 --> GPOS		IF5		Universal Serial Bus
USB2 --> GPOS		IF6		Universal Serial Bus
USB3 --> GPOS		IF7		Universal Serial Bus
USB4		IF8		Universal Serial Bus
		SL1		
5AC901.IPLK-00		SL2	1.8.0.1	APC910 IF POWERLINK SRAM
PLK		IF1		POWERLINK
X20BB81		ST1	1.1.0.0	X20 Bus Base for bus controller or hub, 1 Subslot
X20BC1083		SL1	2.14.0.0	X20 Bus Controller POWERLINK, 2x IF
X20PS9400		PS1	1.8.0.0	24 VDC power supply module for BC, internal IO supply and bus
X20IF10E3_1		SS1	1.9.0.0	X20 Interface PROFINET RT Slave (DTM)
Profinet (DTM)		IF1		
X2X		IF1		X2X Link Interface
X20DM9324		ST2	1.2.0.0	8 Digital Inputs 24 VDC, Sink, 4 Outputs 24 VDC / 0.5 A, Source
X20AI8321		ST3	1.5.0.0	X20 Analog 8xI, 0..20 mA, 12 Bit, 1ms
X20AO4622		ST4	1.3.0.0	4 Outputs ±10V / 0 to 20mA / 4 to 20mA, 12 bit
X20CS1030		ST5	1.8.0.0	Interface Modul RS422 / RS485
Serial		IF1		Communication Port
X20AI2622		ST6	1.2.0.0	X20 Analog 2xI, +/-10V/0..20 mA, 12 Bit
		SL3		

1.1 水箱容器

水箱容器有三个，其中最下方的为蓄水池，只承担蓄水功能，如果水箱整体水较少时，需要向蓄水池中加水；上方右侧为水箱容器 1，其水箱最高；上方左侧为水箱容器 2，其水箱底部稍低于水箱容器 1，当容器 1（右水箱）和容器 2（左水箱）之间手动阀打开时，两水箱将不会独立，容器 1 中的水会因为重力而一部分向左水箱流入，形成负载效应。



1.2 工频泵

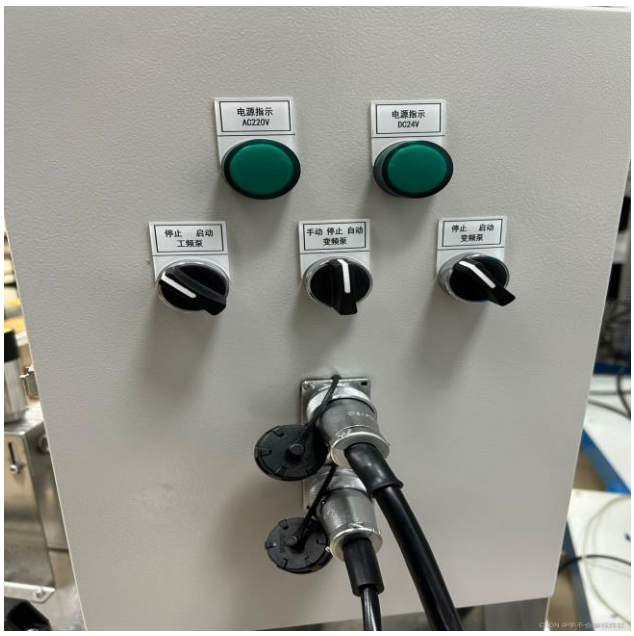
工频泵位于多容水箱设备右侧，与底部水箱相连接。频率一定，开关只可通过右侧接线盒的工频泵开关手动开启（导致了无法利用控制启停 PWM 的方法来控制抽水效率）。

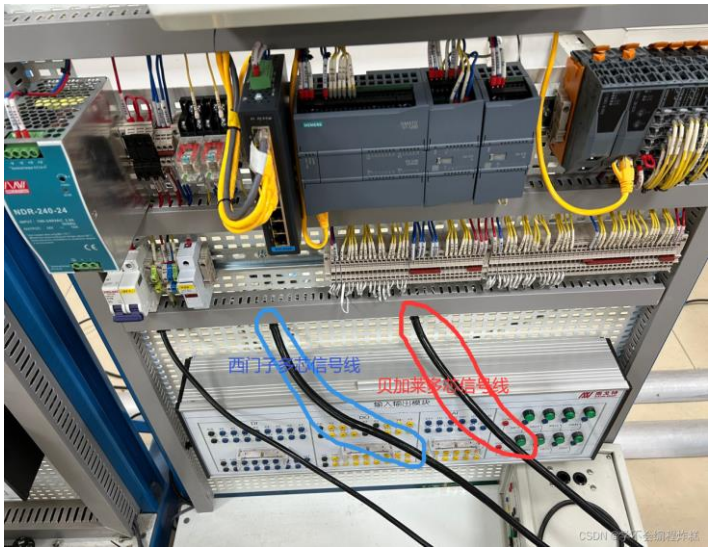
1.3 变频泵

变频泵位于多容水箱设备左侧，也与底部蓄水箱相连接，但是变频泵频率可改变，因此可以通过控制变频泵工作频率直接控制抽水效率，额定频率最高为50Hz。

1.4 水箱操作盒

水箱操作盒位于多容水箱右侧，其中有三个旋转开关，分别控制工频泵和变频泵。另外还有两根多芯信号线（上面那根粗的多芯信号线可更换为西门子控制器接线或者贝加莱控制器接线），西门子控制器接线为中间那根，贝加莱控制器接线为右侧那根（选择不同控制器同学记得检查和更改接线），如下图所示：





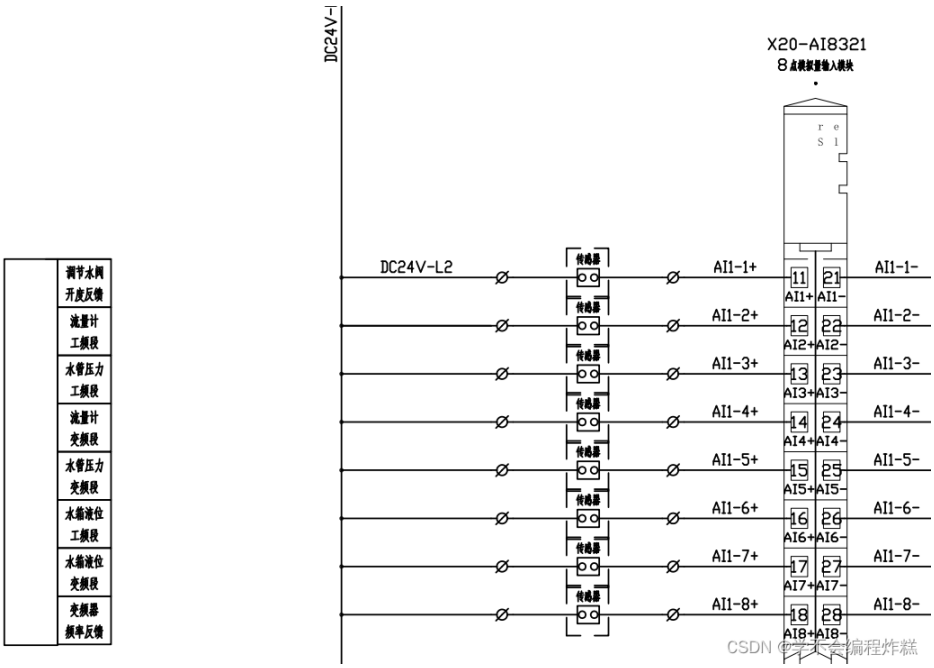
1.5 电动调节阀

电动调节阀位于多容水箱设备右侧侧面中部，与工频泵相距较近，与工频泵相组合，可以受控改变该调节阀开度，从而控制抽水效率。

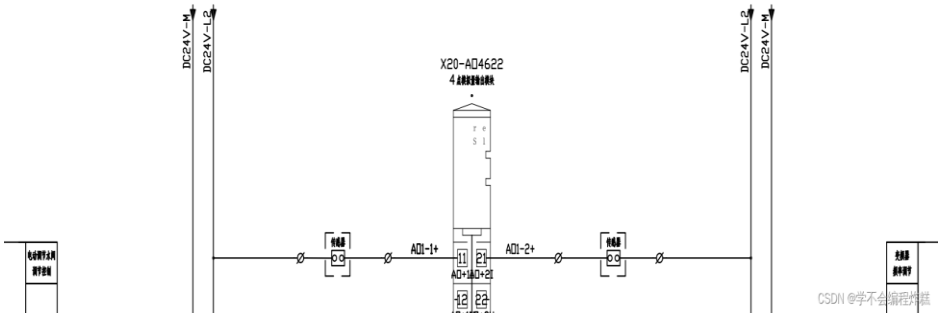
2. 传感器与信号输入输出

水箱设备中具有很多传感器负责将真实物理量转换为电信号 4-20mA 进行输出，比如液位传感器、流量传感器等。该部分将结合水箱接线图和贝加莱 AutomationStudio 软件 IOmapping 进行展开。

下图为 X20-AI8321 模块的部分接线图示，AI 代表模拟量输入。

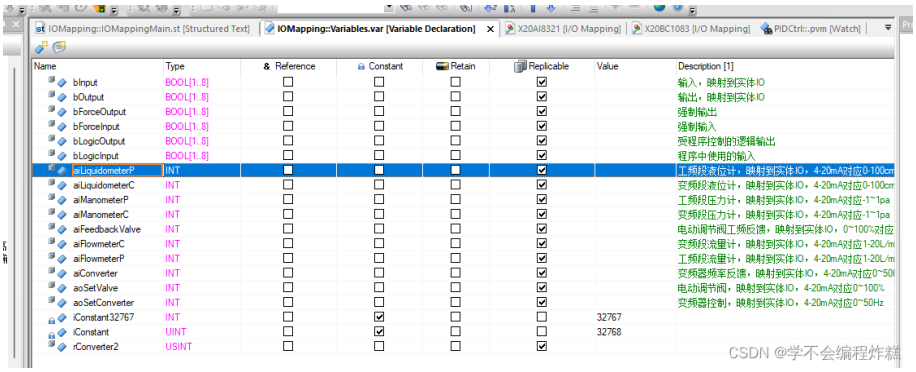


下图为 X20-AD4622 模块的部分接线图示，AO 表示模拟量输出。

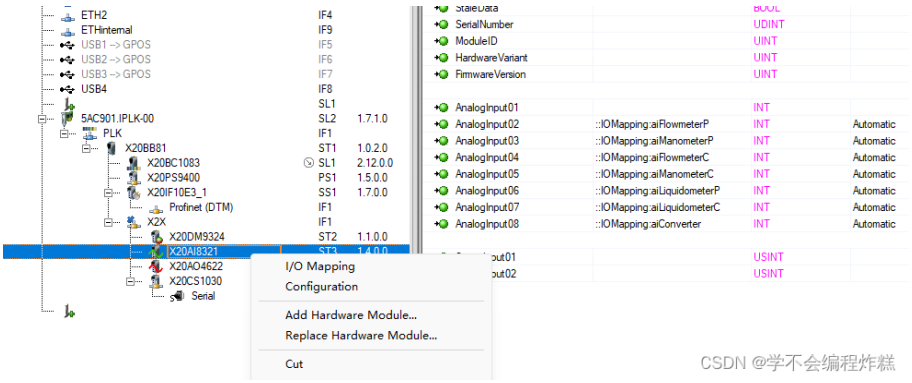


2.1 液位传感器和信号

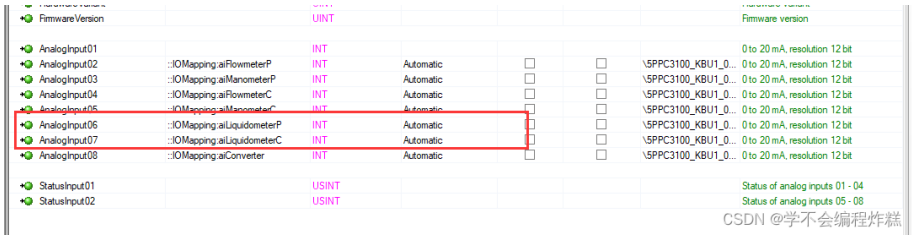
将对于的液位传感器输入变量，绑定到 AutomationStudio 软件硬件组态中 X20-AI8421 模块的 IOmapping 中。如下图所示：



以下是 X20-AI8321 模块 IOmapping 配置操作

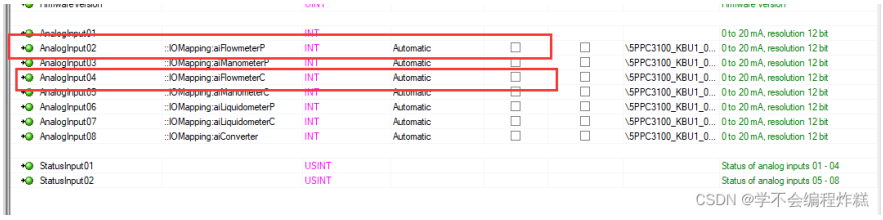


如下即为液位传感器输入信号绑定位置：



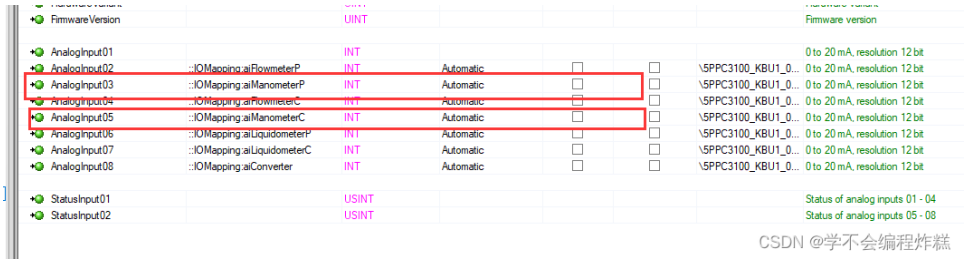
2.2 流量传感器和信号

同理，将流量传感器输入变量绑定到 X20-AI8321 模块 IOmapping 的第 2、4 位。



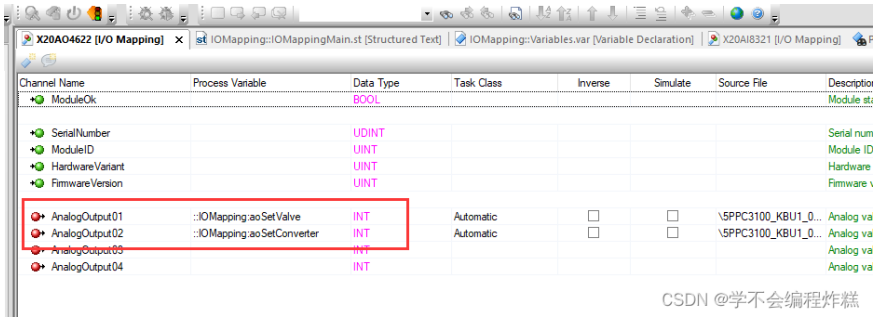
2.3 压力传感器和信号

同理，将压力传感器输入变量绑定到 X20-AI8321 模块 IOmapping 的第 3、5 位。



2.4 变频泵频率输出和调节阀控制输出

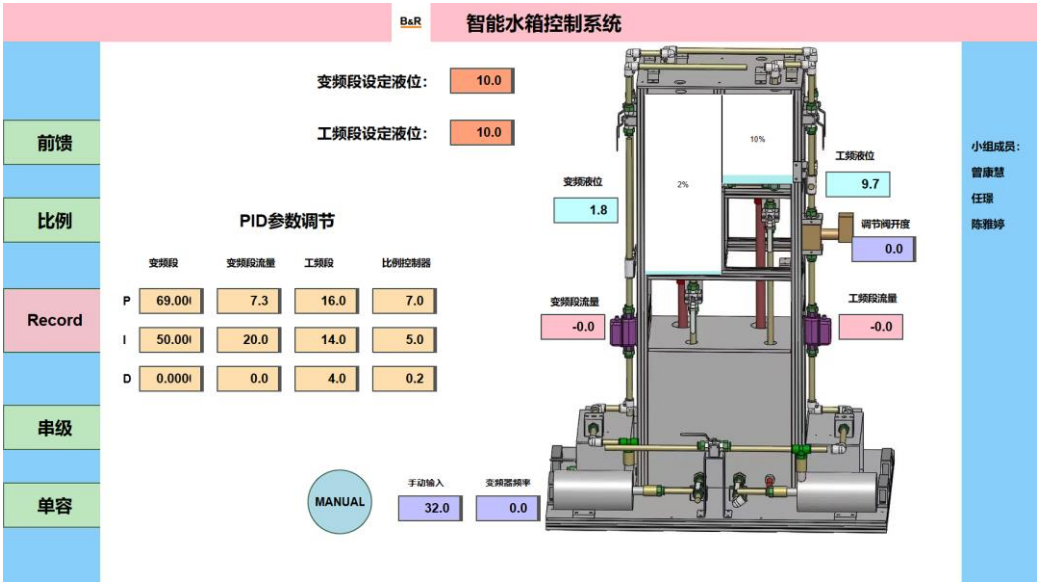
从 X20-AD4622 模块的接线图示可以看出，我们需要将调节阀控制输出和变频泵频率输出分别绑定到 AS 软件 X20-AD4622 模块的第 1、2 位。

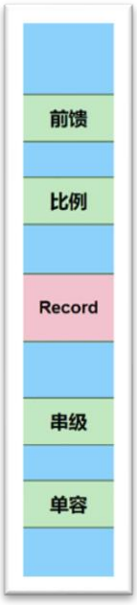


三、控制系统设计及运行结果

1. 系统 UI 设计

主界面：





左边的**菜单栏**：

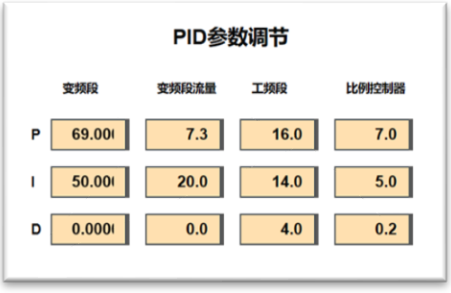
——>实现前馈-反馈控制

——>实现前馈-反馈控制

——>数据记录页面

——>实现串级控制

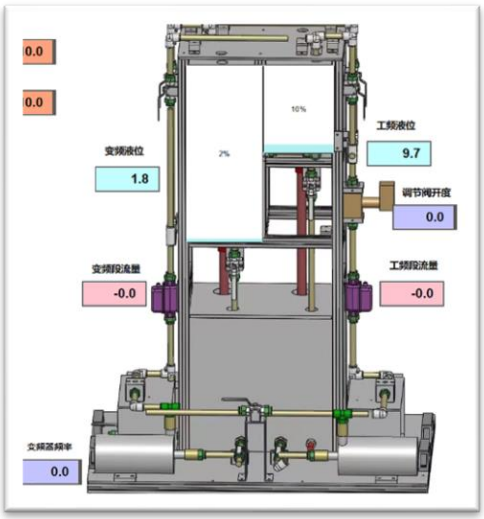
——>实现单容控制



PID 调节模块：

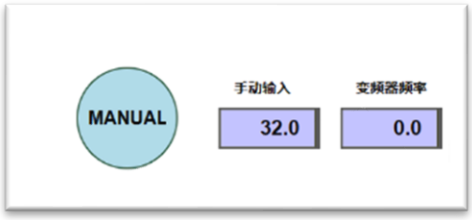
可在主页面直接修改以下 PID 控制器参数：

1. 变频段主回路控制器
2. 变频段副回路控制器
3. 工频段控制器
4. 比值控制器



水箱监控模块：

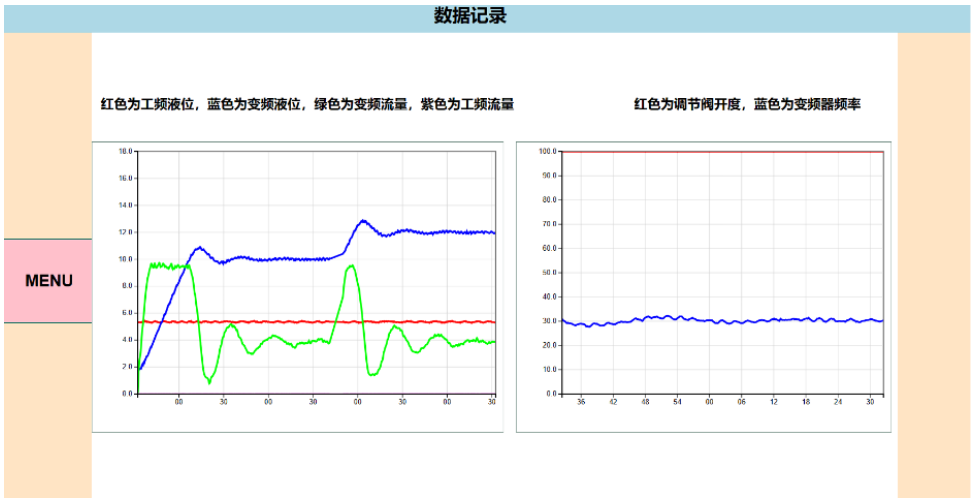
可实时监控水箱的变频段和工频段的液位、流量以及变频器频率，调节阀开度



手动输入变频器频率

可改变变频段和工频段的设定液位

数据监控页面：



左图：红色为工频液位，蓝色为变频液位，绿色为变频流量，紫色为工频流量

右图：红色为调节阀开度，蓝色为变频器频率

（本次实习使用变频段水箱，即左图蓝线）

2. 单容水箱控制

2.1 实验原理

2.1.1 单容水箱动态特性分析

整个单容水箱实验台包含两个水箱。位于下方的水箱为储水箱，位于上方的水箱称为单容水箱，是整个实验的被控对象。单容水箱实验台简单示意如图 1：

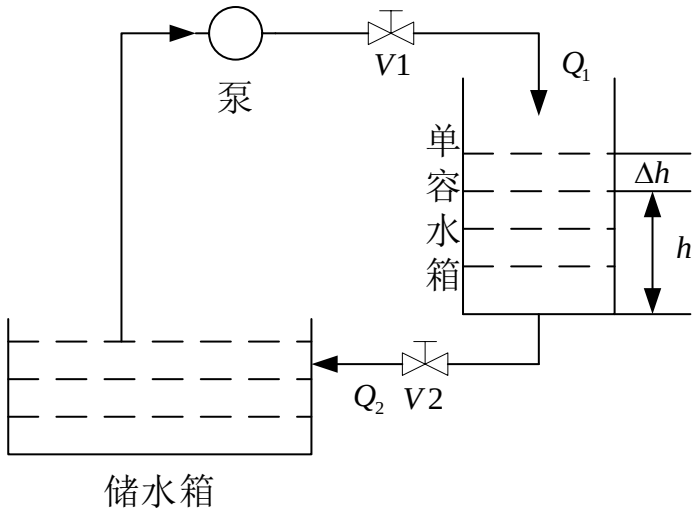


图 1：单容水箱实验台示意简图

图 1 中，包含两个手动阀$V1$和$V2$，两个阀开度一般固定不变。整个实验过程中，储水箱中的水通过水泵的抽取经过手动阀$V1$，以一定流量Q_1流入单容水箱；单容水箱中的水又经过手动阀$V2$，以一定流量Q_2回到储水箱；

可以知道，当手动阀开度不变时，进入单容水箱水的流量Q_1与水泵的转速相关，即通过控制水泵的转速来控制单容水箱液位高度h。实际上水泵是由电机驱动，而电机又是由变频器变频驱动。因此在本实验中，我们的被控对象是单容水箱，被控变量是单容水箱液位高度h，控制变量则是变频器工作频率。

此外，还需要说明的是单容水箱是具有自衡能力的。简单来讲，无论水泵的转速有多大，流量Q_1有多大，经过一段时间后，$Q_1 = Q_2$，单容水箱液位高度h会稳定在一个值附近。

接下来对单容水箱的数学模型进行推导。

根据物料平衡关系得，

在平衡状态时：

$$Q_{10} = Q_{20} \tag{0.1}$$

动态时则有：

$$Q_1 - Q_2 = \frac{dV}{dt} \tag{0.2}$$

式中，V 为单容水箱的储水体积，$\frac{dV}{dt}$ 为单容水箱储水体积的变化率，它与 h 的关系为 $dV = Adh$，即：

$$\frac{dV}{dt} = A\frac{dh}{dt} \tag{0.3}$$

式中，A 为单容水箱的底面积。将式(1.3)代入式(1.2)得：

$$Q_1 - Q_2 = A\frac{dh}{dt} \tag{0.4}$$

基于 $Q_2 = \frac{h}{R_s}$，R_s 为手动阀 $V2$ 的液阻，则上式可改写为 $Q_1 - \frac{h}{R_s} = A\frac{dh}{dt}$，即：

$$AR_s\frac{dh}{dt} + h = R_sQ_1 \tag{0.5}$$

令 $K = R_s$；$T = AR_s$；将上式进行拉氏变换，得到单容水箱传递函数：

$$\frac{H(s)}{Q_1(s)} = \frac{K}{Ts+1} \tag{0.6}$$

考虑整个过程的时滞，则单容水箱的传递函数进一步表示为：

$$\frac{H(s)}{Q_1(s)} = \frac{K}{Ts+1}e^{-\tau s} \tag{0.7}$$

进一步的，由于水泵电机转速与流量 Q_1 成正比，而水泵电机转速与变频器输出频率也成正比，因此变频器输出频率与单容水箱液位的传递函数如下：

$$G(s) = \frac{H(s)}{U(s)} = \frac{K}{Ts+1}e^{-\tau s} \tag{0.8}$$

2.1.2 PID 控制器原理

工程实际中，应用最广泛的调节器控制规律为比例、积分、微分控制，简称 PID 控制。它的优点是结构简单、稳定性好、不依赖精确的数学模型。当控制理论的其他技术难以实施时，系统控制器的结构与参数就依赖于现场调试与经验，

PID 在这样的情况下应用十分简单可靠。当对一个系统不完全了解，或是不能通过有效的测量手段对系统参数进行测量时，PID 控制技术是最适合使用的。PID 控制器就是利用系统的误差，利用比例、积分、微分计算出控制量进行控制。

比例环节用于加快系统的响应速度，提高系统的调节精度。KP 越大，系统的响应速度越快，系统的调节精度越高，但是随之而来的是超调量增大，甚至导致系统不稳定。KP 过小，会降低调节精度，使响应速度变慢，从而延长调节时间，使系统特性被破坏。积分环节主要用来消除静差，TI 越小，系统的静差消除越快，但是 TI 过小，在响应过程的初期会产生积分饱和现象，从而引起响应过程的超调量较大。如果 TI 过大，系统的静差将难以消除，会影响系统的调节精度。微分环节能改善系统的动态特性，可以在响应过程中抑制偏差的变化，对偏差的变化有提前预报的能力。但是 TD 过大，会使响应过程提前停止，从而延长调节时间，降低系统的抗干扰性能。

$$u(t)=K_c e(t)+\frac{K_c}{T_I}\int_0^t e(\tau)d(\tau)+K_c T_D\frac{de(t)}{d(t)}$$

2.1.3 PID 整定方式

下面介绍两种常用的 PID 参数整定方式，分别为临界比例度法与衰减曲线法。

临界比例度法

a. 先将控制器的积分时间 TI 置于最大，微分时间 TD 置零，比例带系数 KP 置为较大的数值，使系统投入闭环运行。
b. 等系统运行稳定后，对设定值施加一个阶跃扰动，并减小 Kp 直到系统出现等幅振荡为止。记录此时的 Kp 与等幅振荡周期 Tk。根据记录的数据与经验公式计算出控制器参数。

控制规律	δ	T_I	T_D
P	$2\delta_k$	-	-
PI	$2.2\delta_k$	$0.85T_k$	-
PID	$1.66\delta_k$	$0.5T_k$	$0.125T_k$

衰减曲线法

衰减曲线法与临界比例度法类似，不同的是衰减曲线法不需要出现等幅振荡过程。

a. 先将控制器的积分时间 TI 置于最大，微分时间 TD 置零，比例带系数 KP 置为较大的数值，使系统投入闭环运行。
b. 等系统运行稳定后，对设定值施加一个阶跃扰动，然后观察系统的响应。若响应震荡衰减太快，就减小比例带系数；反之则增大比例带系数。如此反复，直至系统出现衰减比为 n = 4 : 1 或 n = 10 : 1 的震荡过程，记录此时的比例带系数 Ks，以及相应的衰减震荡周期 Ts 或者输出响应的上升时间 Tp。

衰减率ψ	控制规律	δ	T_I	T_D
0.75	P	δ_s	-	-
	PI	$1.2\delta_s$	$0.5T_s$	-
	PID	$0.8\delta_s$	$0.3T_s$	$0.1T_s$
0.90	P	δ_s	-	-
	PI	$1.2\delta_s$	$2T_r$	-
	PID	$0.8\delta_s$	$1.2T_r$	$0.4T_r$

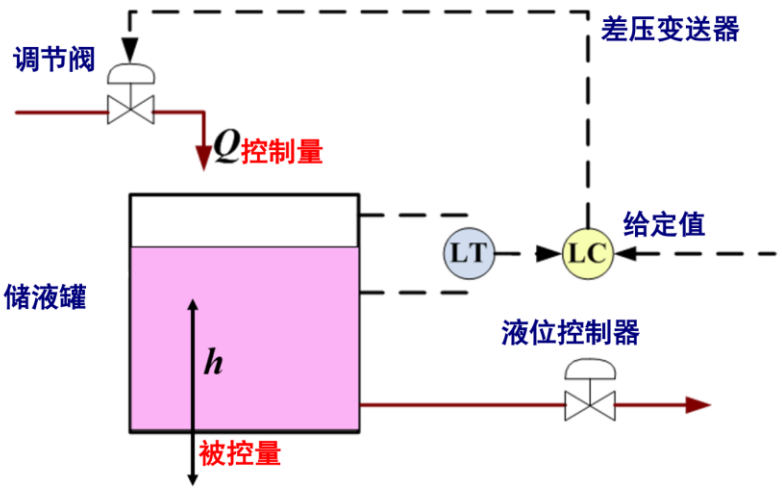
本次我们采用临界比例度法整定参数。

2.1.4 系统分析

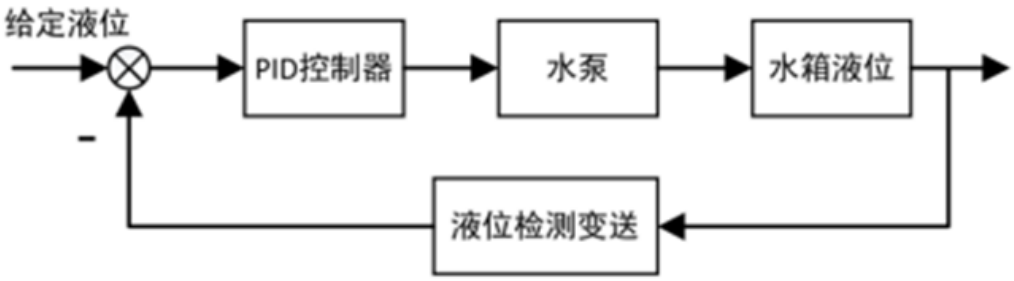
单回路控系统最制中最简单的一种形式，用一个调节器，也有一个闭环。能满足大对数的工艺生产要求，因此，它是一种最基本的、使用最广泛的控制系统它是由被控对象、执行器、调节器和测量变送器组成一个单闭环控制系统。系统的给定量是某一定值，要求系统的被控制量稳定至给定量。由于这种系统结构简单，性能较好，调试方使等优点，故在工业生产中已被广泛应用。

本实验中，被控对象为下水箱液位，控制对象为进水量，因此，选择单回路控制即可满足要求。

单回路的工艺流程图如图所示：



单回路的系统框图如图所示：



系统的最终设计目标是稳定水箱液位。

分析有：

控制器：PID 控制器（正作用）

控制对象：进水量

被控对象：水箱液位高度。

传感器：液位变送器、运输通道中的压力、流量变送器

执行器：变频泵（气开）

2.2 系统辨识

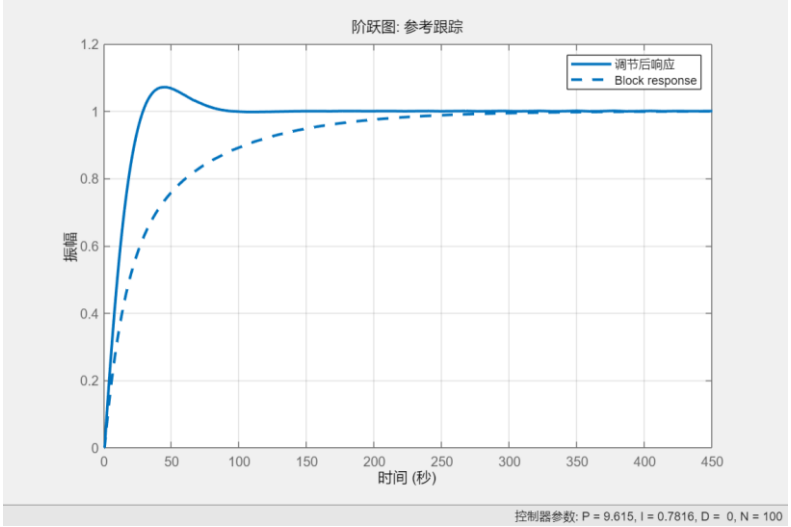
在水箱液面为零时，变频器一定的频率给水箱进水，即给水箱施加一个阶跃，等待液面稳定，并记录数据。

用 Matlab 自带的系统辨识工具箱辨识得：

传递函数模型：

```
tf1 =

  从输入 "u1" 到输出 "y1":
                0.006244
  exp(-0.4*s) * -----------
                s + 0.04291

名称: tf1
Continuous-time identified transfer function.

Parameterization:
   Number of poles: 1   Number of zeros: 0
   Number of free coefficients: 2
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 86.23% (stability enforced)
FPE: 0.007961, MSE: 0.007923
```

Simulink 虚拟调试 pid：



性能和稳健性

	Tuned	Block
上升时间	20.3 秒	20.3 秒
稳定时间	73.4 秒	73.4 秒
超调	7.12 %	7.12 %
峰值	1.07	1.07
增益裕度	36.3 dB @ 3.9 rad/s	36.3 dB @ 3.9 rad/s
相位裕度	70.8 deg @ 0.0764 rad/s	70.8 deg @ 0.0764 rad/s
闭环稳定性	稳定	稳定

2.3 实际系统应用及 PID 参数整定

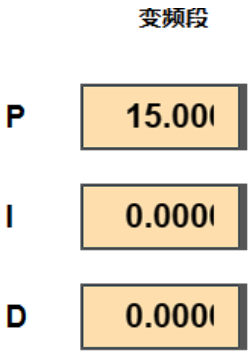
ST 程序代码：

```
IF single THEN //变频单容

    MTBasicsPID_1.EnableTracking:=0;
    MTBasicsPID_1.Enable         :=1;
    MTBasicsPID_1.PIDParameters  := ParaPID1;
    MTBasicsPID_1.SetValue       := rLevelSettingC;
    MTBasicsPID_1.ActValue       := rLiquidometerC;
    MTBasicsPID_1.MinOut         :=24.5;
    MTBasicsPID_1.MaxOut         :=50;
    rSetConverter                :=MTBasicsPID_1.Out;

    IF manual THEN
        MTBasicsPID_1.EnableTracking :=1;
    END IF
```

默认设定液位为 10 cm

2.3.1 纯 P 控制

PID 参数：

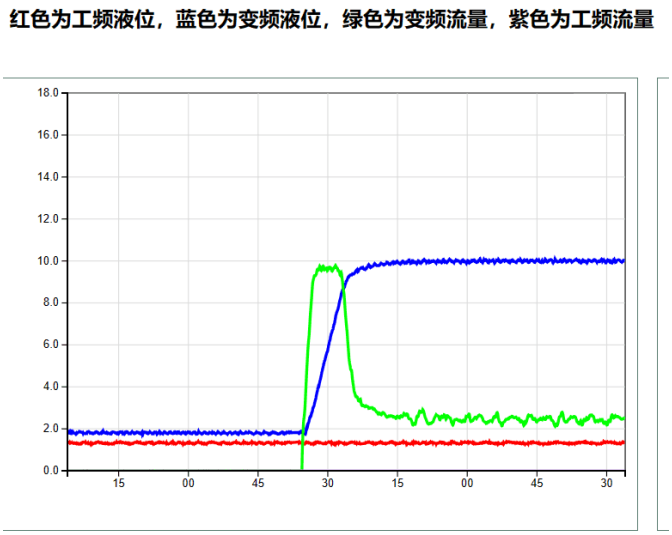


P： 15

I： 0

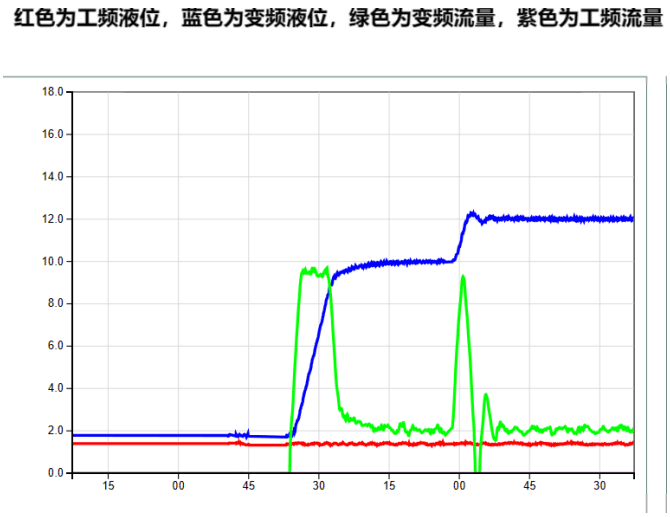
D：0.0

纯 P 控制效果（无扰）：



可以看出响应无超调

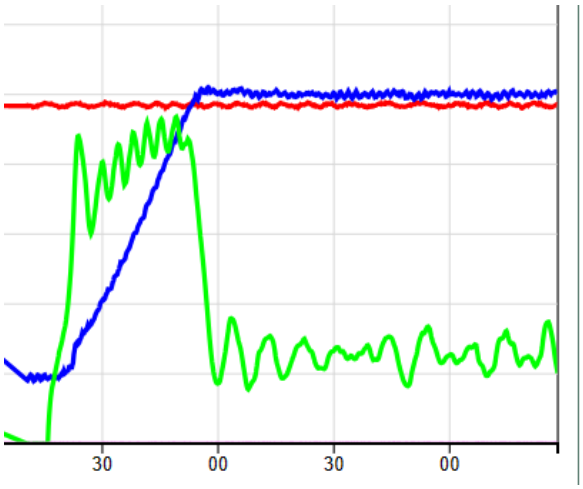
纯 P 控制效果（+扰动，设定液位 10cm—>12cm）：



加了扰动后有些许超调

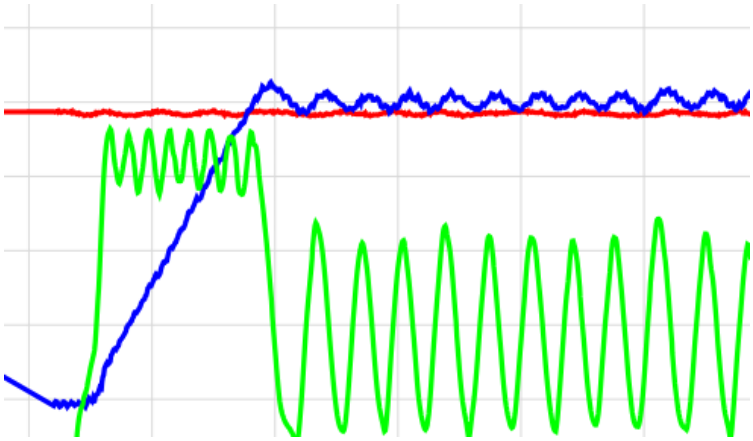
增大 P 参数后的效果：

增加到 P=16：



可以看出有微小超调

增加到 P=18（有超调）：



可以看出此时有超调和等幅振荡

基于此记录临界比例带δ_K和等幅振荡周期T_K，接下来进行临界比例度法整定参数

结果分析：

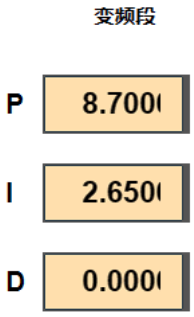
1. 增大 P 参数可以加快系统的响应速度，减小稳态误差。
2. 但 P 参数的取值会影响系统的稳定性，过大的 P 参数可能导致系统在运行过程中产生振荡，使系统稳定性变差。

2.3.2 PI 控制

控制规律	δ	T_I	T_D
P	$2\delta_k$	-	-
PI	$2.2\delta_k$	$0.85T_k$	-
PID	$1.66\delta_k$	$0.5T_k$	$0.125T_k$

利用临界比例度法整定参数后得到 PI 参数：

PI 参数：

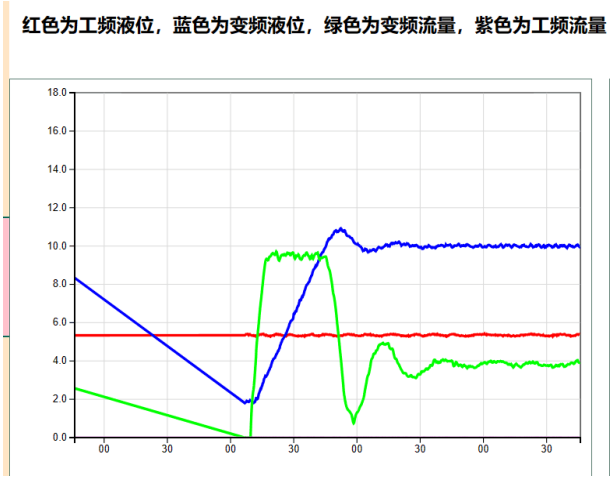


P： 8.7

I： 2.65

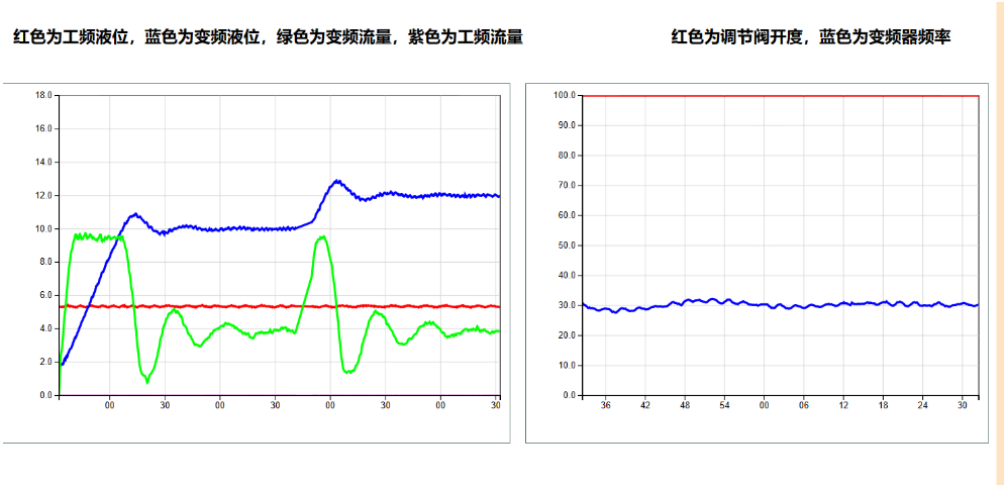
D：0.0

PI 控制效果（无扰）：

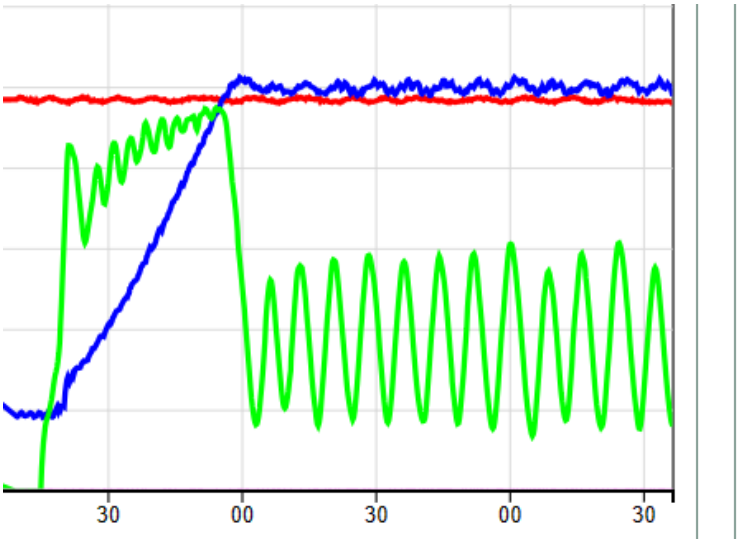


可以看出有超调，但稳定较快，消除了稳态误差

PI 控制效果（有扰动，设定液位 10cm—>12cm）



参数 P 不变，同时增加参数 I 为 4 时的响应：



增加参数 I 后有微小震荡

结果分析：

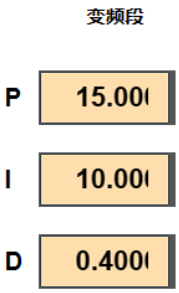
1. 相比于上一组纯 P 控制，这组整定参数后的 PI 控制响应有超调，更迅速，消除了稳态误差。因为加入积分控制后，消除了系统稳态误差。因为偏差出现时，比例作用迅速反应输入的变化，积分作用使输出逐渐增加，最终消除稳态误差，可以达到比较好的控制效果。

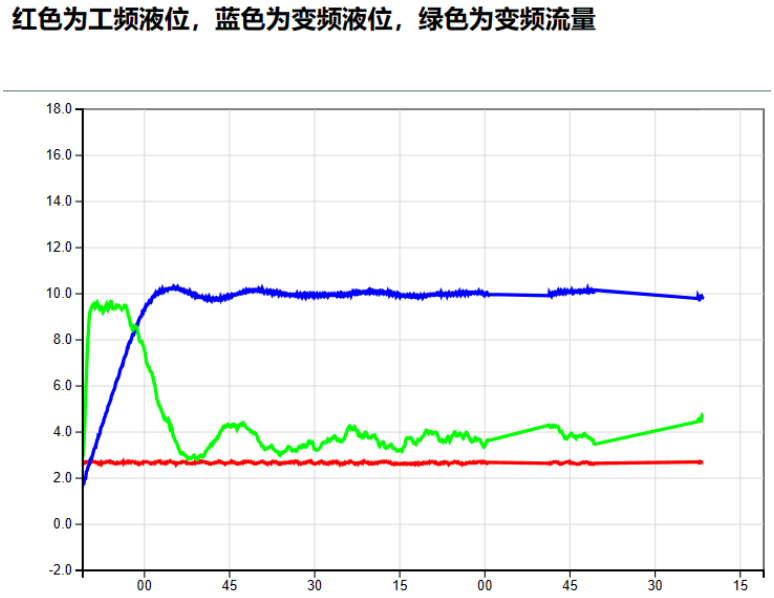
2. 但是 TI 越大，积分作用越弱，可以有效降低超调量，提高稳定性。TI 减小，积分作用增强，系统振荡加剧,达到稳态的过渡时间也逐渐加长。

2.2.3 PID 控制

利用临界比例度法整定参数后得到 PID 参数：

PID 参数：





结果分析：

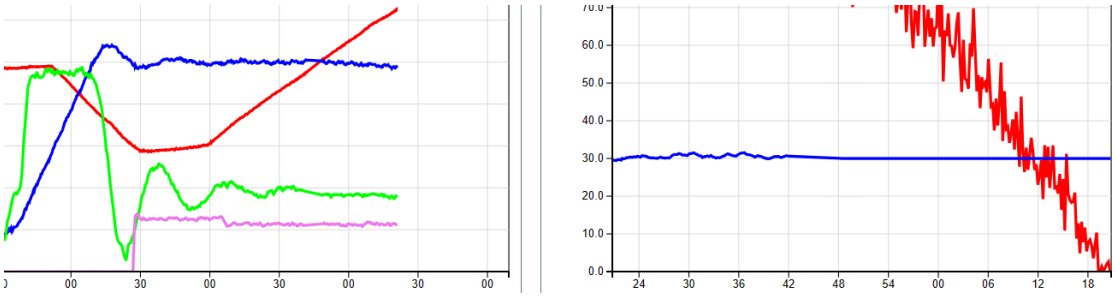
1. 相比于上一组 PI 控制，这组参数整定后的 PID 控制的响应有通过积分作用消除误差，通过微分控制可缩小超调量、加快系统响应。
2. 随着 Td 值的增大，系统超调量逐渐减小，动态特征有改善。是综合了 PI 控制和 PD 控制长处并去除其短处的控制。
3. 但是微分控制对于一些噪声过于敏锐，可能会变频泵的 pwm 输出频率起伏过大从而损坏仪器。

2.3.4 手动模式自动模式无扰切换：

按下“MANUAL”按钮并手动输入期望频率后，即可进入手动模式

再次按下按钮即可回到 pid 自动调频模式

在稳定时切换至手动模式：



结果分析：

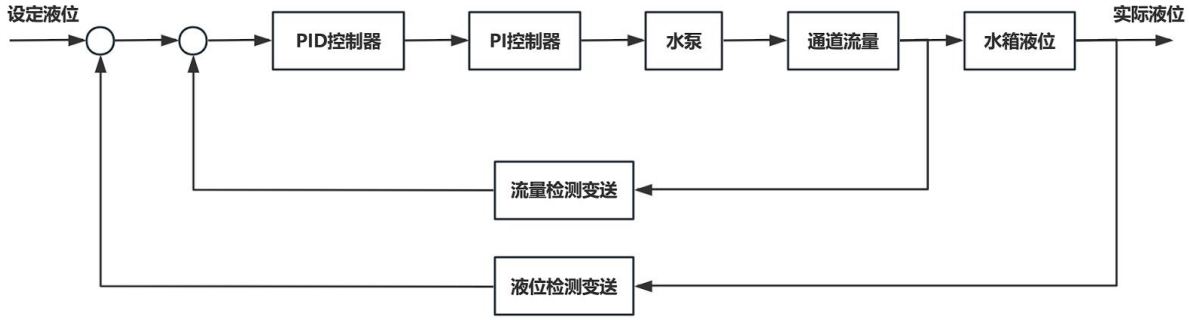
右图可看出当 42s 时切至手动模式时变频器频率（右图蓝线）已固定为 30

3. 流量-液位串级控制

3.1 实验原理

只使用单闭环 PID 控制器来控制液位的稳定，面临着运输管道内流体压力不稳定的问题。

为了解决这一问题，第二阶段我们考虑使用流量变送器进行副回路设计，通过内环控制器来迅速抑制运输通道内流量变化对系统产生的干扰，最终构成流量-液位串级控制系统来对系统进行更加精准的控制。系统框图如图所示。



分析得：
副过程：正
主过程：正
主控制器：PID 控制器（负作用）
副控制器：PI 控制器（负作用）
主被控量：水箱液位
副被控量：通道流量
控制量：进水量
执行器：变频泵
传感器：流量、液位检测变送器

串级控制系统的特点:

(1)改善了过程的动态特性;

(2)能及时克服进入副回路的各种二次扰动，提高了系统抗扰动能力;

(3)提高了系统的鲁棒性;

(4)具有一定的自适应能力。

3.2 串级 PID 参数整定方式

串级控制常用的整定方法有逐步逼近法、两步整定法、一步整定法。

1、逐步逼近法

先断开主环，按单回路控制系统整定副控制器，将结果应用于副控制器。将副回路当做主回路一个环节，再整定主控制器，然后循环进行且逐步逼近主、副控制器的最佳值。

2、两步整定法

先整定副控制器，主副控制器均纯比例作用，主控制器比例度 100%，4:1 衰减曲线法整定副回路得δ_{2s}和T_{2s}；应用副控制器整定结果，将副回路当做主回

路一个环节，再整定主控制器，得到主控制器满足 4:1 衰减过程的δ_{1s}，和T_{1s}。最后按经验公式计算出主、副控制器的整定参数并微调。

3、一步整定法

根据经验先确定副控制器比例度（参考值取 P=5），然后按单回路控制系统整定方法整定主控制器参数

在实习中我们使用两步整定法进行串级 PID 整定

3.3 实际系统应用

ST 程序代码如下：

```
IF cascade THEN //变频流量串级

    MTBasicsPID_1.Enable        :=1;
    MTBasicsPID_1.PIDParameters := ParaPID1;
    MTBasicsPID_1.SetValue      := rLevelSettingC;
    MTBasicsPID_1.ActValue      := rLiquidometerC;

    MTBasicsPID_1.MinOut        :=0;
    MTBasicsPID_1.MaxOut        :=10;

    MTBasicsPID_2.Enable        :=1;
    MTBasicsPID_2.PIDParameters := ParaPID2;
    MTBasicsPID_2.SetValue      := MTBasicsPID_1.Out;
    MTBasicsPID_2.ActValue      := rFlowmeterC;

    MTBasicsPID_2.MinOut        :=24.5;
    MTBasicsPID_2.MaxOut        :=50;
    rSetConverter               :=MTBasicsPID_2.Out;

END_IF;
```

3.3.1 主、副控制器参数的选择

分析：

主控制器起定值控制作用， 副控制器起随动控制作用

主控制器选用 PI 或 PID 控制规律

- 主参数是工艺主要指标， 允许波动的范围小， 一般要求无余差

副控制器选择 PI 控制规律

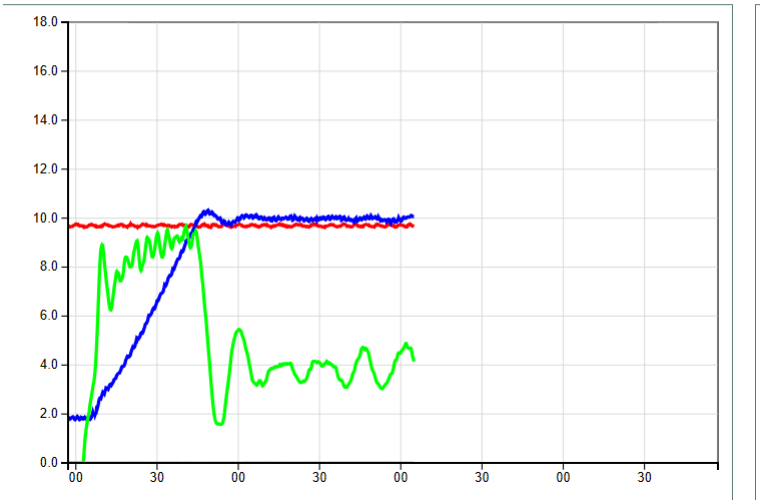
- 副参数为了主参数控制质量， 允许有余差
- 引入积分控制， 会延长控制过程， 减弱副回路的快速作用
- 引入微分控制， 因副回路本身起着快速作用， 再引入微分作用会使调节阀动作过大， 对控制不利，故不采用。

使用两步整定法整定得主副控制器 PID 参数：

	主控制器	副控制器
	变频段	变频段流量
P	20.00	10.0
I	20.00	10.0
D	0.500	0.0

3.3.2 串级控制效果

红色为工频液位，蓝色为变频液位，绿色为变频流量，紫色为工频流量



结果分析：

相比于单回路控制系统，该串级系统能迅速克服副回路干扰，使控制性能和抗干扰能力的综合指标有明显的提高，响应速度更快，超调量更小。

单回路和串级系统的比较：

单回路控制系统是指只有一个被调参数、一个控制器和一个反馈回路的控制系统。

单回路控制系统的优点是结构简单、易于实现和调节。

但缺点是对扰动的抑制能力较弱、调节速度较慢、稳态误差较大。

串级控制系统是指在单回路控制系统的基础上增加了一个副回路，使得主回路的输出作为副回路的给定值，副回路的输出作为主回路的控制输入。

串级控制系统的优点是对进入副回路的扰动具有较强的克服能力、可以改善对象特性、提高调节速度和精度、具有一定的自适应能力。

但缺点是结构复杂、仪表多、参数整定难。

因此，相比单回路控制系统，串级控制系统对设定值扰动的控制过程动态品质更好，但也更复杂。

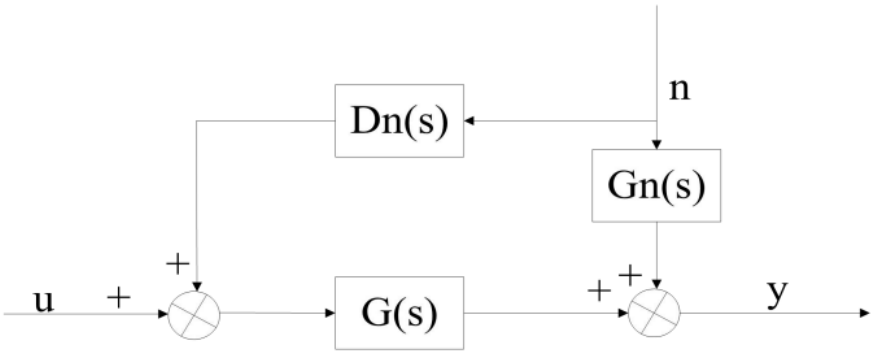
4. 前馈-反馈控制

4.1 实验原理

反馈控制系统中，反馈按偏差控制。即在干扰的作用下，被控量先偏离给定值，然后调节器才按偏差产生控制作用去抵消干扰的影响。如果干扰不断施加，则系统总是跟在干扰作用后面波动，从而不可避免的存在稳态位置跟踪误差。

前馈控制是按扰动量进行补偿的开环控制，即当系统扰动出现时，按照扰动量的大小直接产生校正作用。前馈控制在理论上可以完全消除扰动引起的偏差。

前馈控制的结构图如下图。



Gn(s)是被控对象扰动通道的传递函数，Dn(s)是前馈控制器的传递函数，G(s)为被控对象控制通道传递函数，n、u、y 分别为扰动量、控制量和输出量。

若使前馈控制作用完全补偿扰动作用，则:

$$Dn(s)G(s)+Gn(s)=0$$

$$Dn(s)=-Gn(s)/G(s)$$

在实际应用中，因为前馈控制为一个开环系统，因此常常采用反馈+前馈的复合控制方式，这样既有前馈控制及时、又有反馈控制精确的特点。

从理论上看，前馈调节能依据干扰值的大小，在被调参数偏离给定值之前进行控制，使被调量始终保持在给定值上。而实现完全补偿，在很多情况下只有理论意义，实际上做不到;同时，在工业对象中，存在许多扰动因素，我们只能选择一两个主要的扰动进行补偿，而其余的扰动仍会使被调量发生偏差。

前馈-反馈控制系统将前馈与反馈结合起来，选择对象中主要的一些干扰作为前馈信号，对其它引起被调参数变化的各种干扰则采用反馈调节系统来克服，从而充分利用了这两种调节作用的优点，使调节质量进一步提高。前馈-反馈控制系统的整定，一般先反馈，后前馈，且二者基本独立。

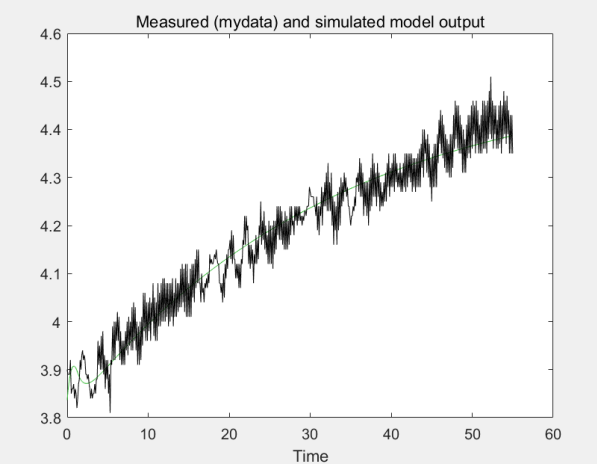
静态前馈不考虑干扰作用被控变量的动态过程，仅保证系统在稳态的补偿作用。

则静态前馈控制器的传递函数:Dn(s)=-Kf，具有比例特性，Kw 为对象干扰通道与控制通道的静态放大系数之比。

4.2 扰动辨识及前馈-反馈仿真

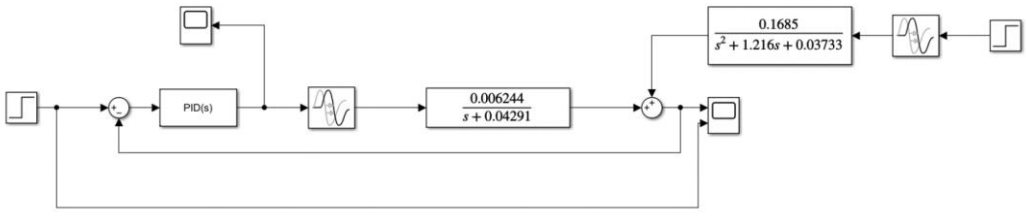
当变频水箱液位稳定在 10cm 时，工频水箱跟变频水箱连通，给变频水箱加入 1cm 水位的扰动，即可测不可控的扰动，记录该扰动过程，用 Matlab 自带的系统辨识工具箱辨识该扰动得：

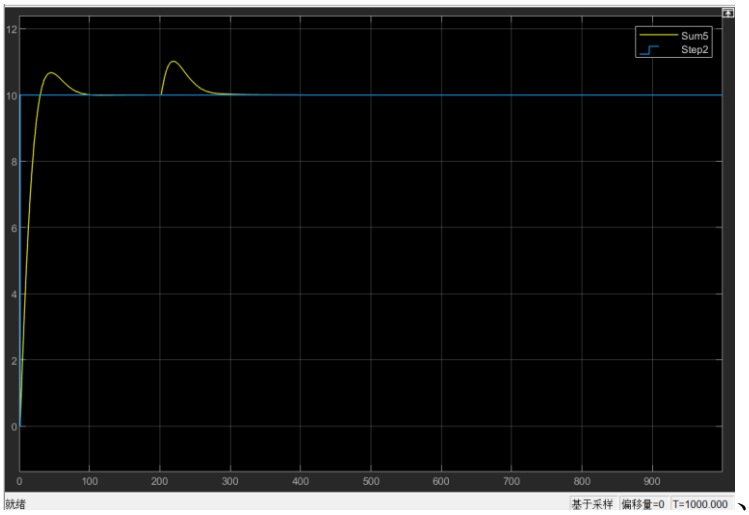
```
tf2 =

  从输入 "u1" 到输出 "y1":
                        0.1685
  exp(-1.2*s) * ------------------------
                s^2 + 1.216 s + 0.03733

名称: tf2
Continuous-time identified transfer function.

Parameterization:
   Number of poles: 2   Number of zeros: 0
   Number of free coefficients: 3
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 73.71% (stability enforced)
FPE: 0.00193, MSE: 0.001895
```



simulink 设计前馈-反馈控制器：

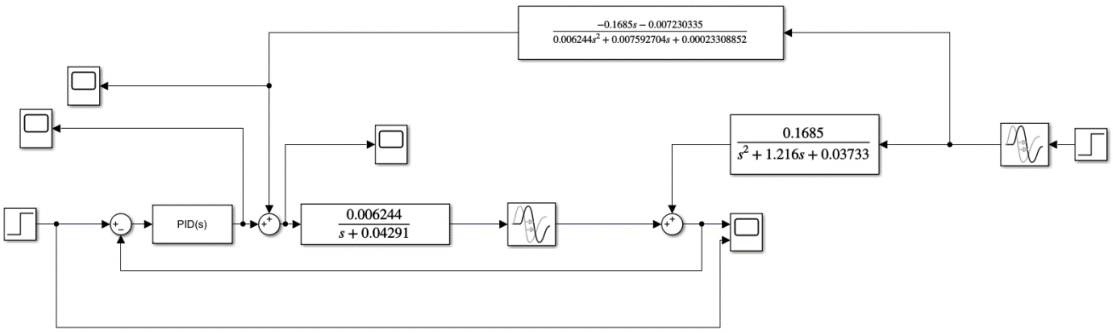
1、反馈+扰动：

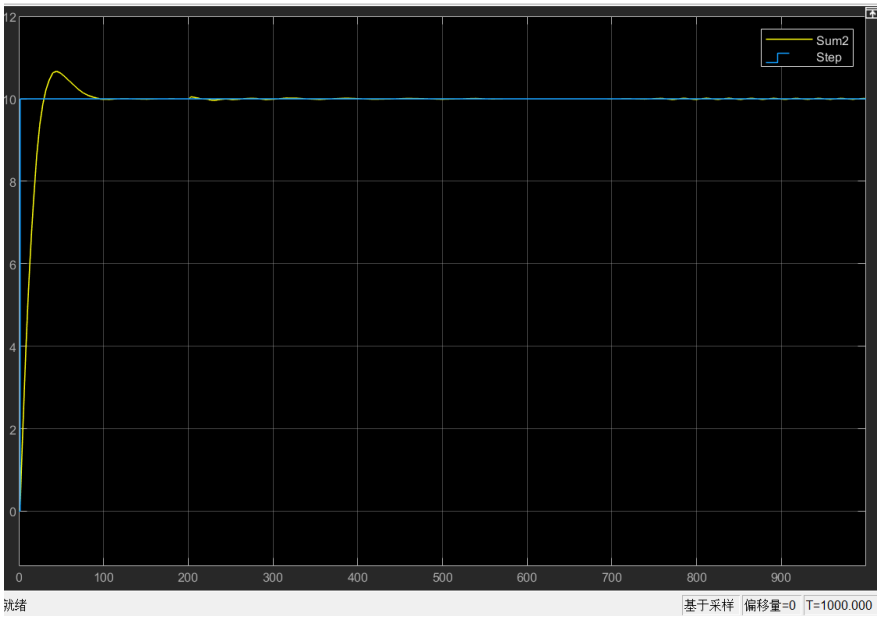




可以看到第二个扰动非常明显

2、前馈-反馈+扰动：





可以看到第二个扰动被前馈控制器抵消了

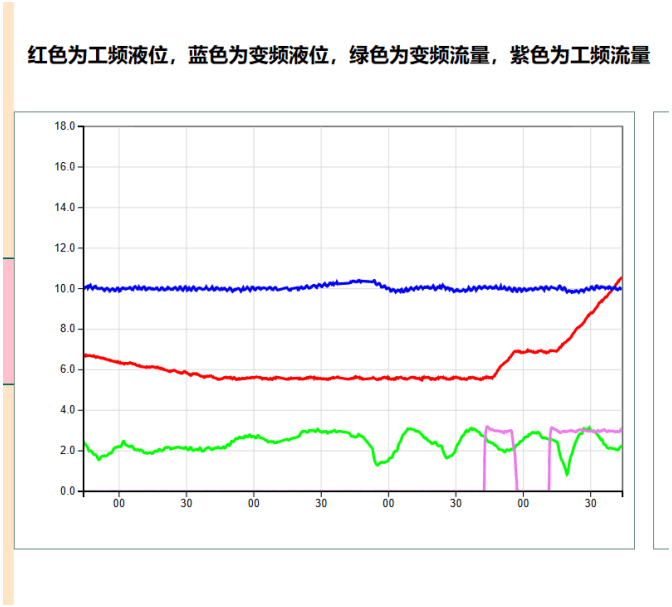
4.3 实际运行结果

ST 程序代码如下：

```
IF feedforward  THEN    //前馈

    out :=(MTBasicsPID_1.Out-2);

    IF (MTBasicsPID_1.Out-2)<0  THEN
        out:=0;
    END_IF

    rSetConverter                :=out;

END_IF;
```

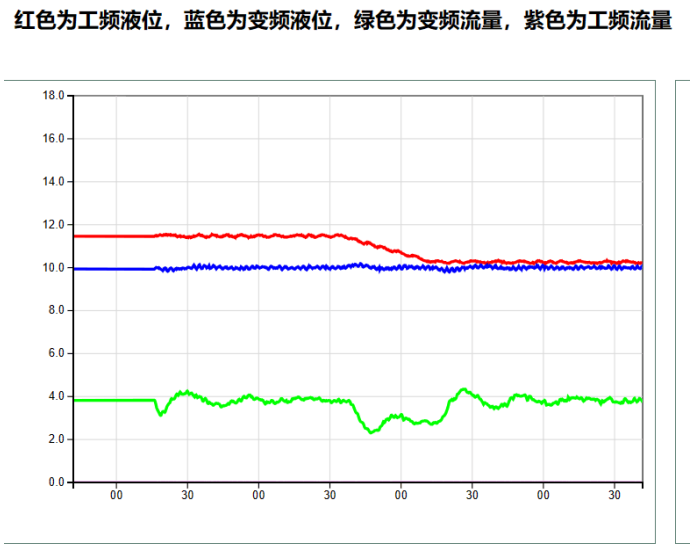
实现效果：当变频水箱液位稳定在 10cm 时，工频水箱跟变频水箱连通，给变频水箱加入 1cm 水位的扰动，用前馈控制把扰动抵消。

未加前馈控制器的效果：



工频水箱（红线）给变频水箱（左边蓝线）加入 1cm 水位的扰动，可以看出变频液位（左边蓝线）在加入扰动后有些许起伏。

加了前馈控制器的效果：



结果分析：

1. 可以看出加了前馈控制器后，工频水箱（红线）给变频水箱（左边蓝线）加入 1cm 水位的扰动，变频液位（左边蓝线）在加入扰动后十分平稳。
2. 由实验结果可知，在扰动的作用下，前馈-反馈控制系统能更快地达到稳态，且扰动引起超调量更小。这是因为反馈控制系统在被控量出现偏差后才进行调节，调节作用在干扰作用之后。前馈调节则将干扰测量出来并直接引入调节装置，在干扰为对实际输出产生影响的时候就对其进行了处理，对于干扰的克服比反馈控制及时。
3. 前馈系统使用中存在的问题：

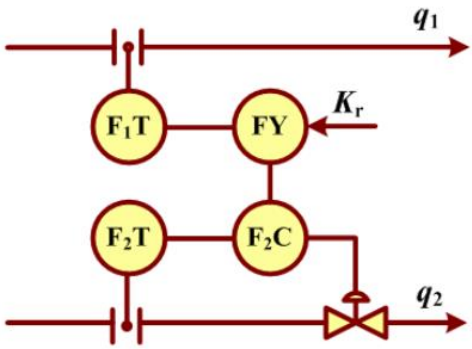
前馈控制使用的调节器是根据被控对象的特点来确定调节规律的前馈调节器，即需要专用调节器，不具有广泛的适用性。

前馈控制只能克服所测量的干扰，若干扰量不可测量，就无法使用有效的前馈。

工业对象存在多个扰动，若均设置前馈控制器会大幅度提高成本。

5. 比例控制

5.1 实验原理



比值控制系统：一种物料随另一种按一定比例变化的控制系统
主物料/主动量：处于比值控制中的主导地位，即工频段流量 q1
从物料/从动量：按主物料进行配比，即变频段流量 q2
比值控制系统：要求从动量 L2 与主动量 L1 成一定的比值关系，即满足：

$$\frac{q2}{q1} = K = 2$$

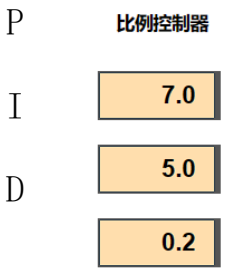
5.2 实际系统应用

ST 程序代码如下：

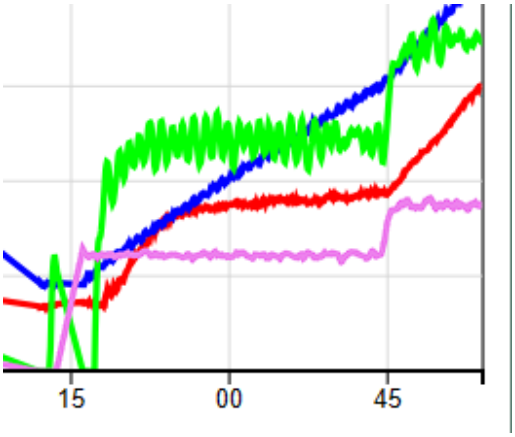
```
//变频水箱 流量比值控制
    IF DOUBLE  THEN
        //主回路
        rSetValve                   :=60;
        //副回路
        MTBasicsPID_1.Enable        := 1;
        MTBasicsPID_1.PIDParameters := ParaPID1;
        MTBasicsPID_1.SetValue      := 2*rFlowmeterP;
        MTBasicsPID_1.ActValue      := rFlowmeterC;
        MTBasicsPID_1.MinOut        := 30;
        MTBasicsPID_1.MaxOut        := 50;
        rSetConverter               := MTBasicsPID_1.Out;
    ELSIF DOUBLE = 0 THEN
        MTBasicsPID_1.Enable        :=0;

    END_IF;
```

实现变频流量=2*工频流量

参数：



实际效果：



结果分析：

可见变频流量（绿线）=2*工频流量（粉线），实现了比值控制。

四、实习心得与体会

这次实习时间跨度非常长，从 10 月初的贝加莱软件培训，到 11 月的预习报告，再到 12 月的正式实习，整整三个月，我对于 PLC 从迷茫慢慢转为熟悉。

我还记得我一开始学习使用贝加莱软件的时候真的挺不知所措的，不知道各个部分是干什么的，对于组态和程序之间的关系也是一知半解，但随着实习的开展，把一个个物理器件绑定到贝加莱软件上，自己编写了第一个简单的程序，克服了一个又一个问题后，当看着水箱的变频器开始顺利运作后，我才真正理解了 PLC 的底层逻辑。

但变频器能运作还不够，我们还得让液面稳定在设定值，这时候就得用 PID 控制了，在过程控制系统课上，安剑奇老师讲 PID 原理讲得很好，虽然在 Simulink 上可以做 PID 仿真，但是当我把 PID 真正运用到水箱上时，才发现仿真跟实际还是有非常大的区别的，比如当我们把水箱系统用 Matlab 辨识出来后用 Simulink 做 PID 仿真，当我们用衰减曲线法整定出一个比较完美的参数后将其应用到实际水箱上时，我们惊奇地发现实际的曲线跟仿真曲线相比要差一些，这可能是由于我们系统辨识所辨识出来的模型拟合度只有 90%的缘故。

虽然实习过程中遇到了数不尽的困难，但是关关难过关关过，遇到问题不要怕，解决问题就是了，最终，我们解决了一个又一个问题后已经能够熟练使用贝加莱软件，并且能灵活运用《现代过程控制原理与应用技术》理论知识在 DCS 框架下进行串级、前馈-反馈、比值控制等复杂过程对象的控制系统设计。

由于各科期末考试临近，再加上水箱的实验放水进水耗时很长，我在实习的过程中经常会陷入一种比较焦虑的状态，希望尽快地将实习完成，进而有更多的时间投入到期末复习上。但是往往越是着急，就越容易制造更多的 BUG 使得实习效率低下，所以说过控实习也是一个磨练人意志的过程，如何及时调整心态、调整设计思路？如何在有限的时间内尽可能的设计出更好的控制思路？如何适当的放弃部分功能，不死磕 BUG？这些都是我在该阶段实习中所学习到的。

虽然大部分想法能得到的较好的完成，但是有些想法没有被实现，例如预测控制和解耦控制，这些由于时间和个人能力问题没能完成，还是比较遗憾的。

总的来说，我对该阶段实习的完成情况是比较满意的，设计之初的所有想法我们都有尝试去实现过，在工作量上问心无愧。但庆幸的是，最终我们还是很好地完成了已有的单回路、串级、前馈-反馈、比值控制。

整个实习过程收获满满，学到了很多，感谢何王勇老师，胡杰老师和助教们的付出。

参考文献

[1] 郭一楠,等.过程控制系统.北京:机械工业出版社.2009.1

代码附录

IOMapping.st:

```
PROGRAM _CYCLIC

    //模拟量输入
    rLiquidometerP :=-1+aiLiquidometerP/327.67;
    rLiquidometerC :=-1+aiLiquidometerC/327.67;

    rFeedbackValve :=aiFeedbackValve/327.67;
    rFlowmeterP         :=aiFlowmeterP/327.67/5.0;
    rManometerP         :=(aiManometerP-16384.0)/16384.0;
    rFlowmeterC         :=aiFlowmeterC/327.67/5.0;
    rManometerC         :=(aiManometerC-16384.0)/16384.0;
    rFeedbackConverter      :=aiConverter/327.67/2.0;

    //模拟量输出 aoSetValve  aoConverter
    aoSetValve          :=REAL_TO_INT(rSetValve*iConstant32767/100.0);
    aoSetConverter :=REAL_TO_INT(rSetConverter*2.0*iConstant32767/100.0);
    IF rSetConverter<0 THEN
        rSetConverter:=0;
    END_IF;

    //复位所有强制输入输出

    //*****************输入**************************
    bLogicInput[1]:=bForceInput[1] OR bInput[1];
    bLogicInput[2]:=bForceInput[2] OR bInput[2];
    bLogicInput[3]:=bForceInput[3] OR bInput[3];
    bLogicInput[4]:=bForceInput[4] OR bInput[4];
    bLogicInput[5]:=bForceInput[5] OR bInput[5];
    bLogicInput[6]:=bForceInput[6] OR bInput[6];
    bLogicInput[7]:=bForceInput[7] OR bInput[7];
    bLogicInput[8]:=bForceInput[8] OR bInput[8];

    bFloatSwitch   :=bInput[1];
    bPumpAutoOrM   :=bInput[2];
    bPumpFeedback  :=bInput[3];

    //*****************输出**************************
```

```
    bOutput[1]:=bForceOutput[1] OR bLogicOutput[1];
    bOutput[2]:=bForceOutput[2] OR bLogicOutput[2];
    bOutput[3]:=bForceOutput[3] OR bLogicOutput[3];
    bOutput[4]:=bForceOutput[4] OR bLogicOutput[4];

    bLogicOutput[1] := bPumpCtrl;


END_PROGRAM
```

PID_Ctrl:

```
PROGRAM _CYCLIC

    IF single THEN //变频单容

        MTBasicsPID_1.EnableTracking:=0;
        MTBasicsPID_1.Enable         :=1;
        MTBasicsPID_1.PIDParameters := ParaPID1;
        MTBasicsPID_1.SetValue       := rLevelSettingC;
        MTBasicsPID_1.ActValue       := rLiquidometerC;
        MTBasicsPID_1.MinOut         :=24.5;
        MTBasicsPID_1.MaxOut         :=50;
        rSetConverter                :=MTBasicsPID_1.Out;

        IF manual THEN
            MTBasicsPID_1.EnableTracking :=1;
        END_IF

        IF feedforward  THEN   //前馈

            out :=(MTBasicsPID_1.Out-2);

            IF (MTBasicsPID_1.Out-2)<0  THEN
                out:=0;
            END_IF

            rSetConverter               :=out;

        END_IF;

    END_IF;
```

```
IF cascade THEN //变频流量串级

     MTBasicsPID_1.Enable         :=1;
     MTBasicsPID_1.PIDParameters := ParaPID1;
     MTBasicsPID_1.SetValue       := rLevelSettingC;
     MTBasicsPID_1.ActValue       := rLiquidometerC;

     MTBasicsPID_1.MinOut         :=0;
     MTBasicsPID_1.MaxOut         :=10;

     MTBasicsPID_2.Enable         :=1;
     MTBasicsPID_2.PIDParameters := ParaPID2;
     MTBasicsPID_2.SetValue       := MTBasicsPID_1.Out;
     MTBasicsPID_2.ActValue       := rFlowmeterC;

     MTBasicsPID_2.MinOut         :=24.5;
     MTBasicsPID_2.MaxOut         :=50;
     rSetConverter                :=MTBasicsPID_2.Out;

END_IF;

IF (single=0) AND (cascade =0) THEN
     MTBasicsPID_1.Enable         :=0;
     MTBasicsPID_2.Enable         :=0;
END_IF;
(*
IF UPDATE THEN  //参数更新
     MTBasicsPID_1.Enable     :=1;
     MTBasicsPID_2.Enable     :=1;
ELSE
          MTBasicsPID_1.Enable :=0;
          MTBasicsPID_2.Enable :=0;
END_IF;
*)



MTBasicsPID_1();
MTBasicsPID_2();


     (**********调用功能块**********)
```

```
END_PROGRAM
```

PID_Double.st:

```
PROGRAM _CYCLIC
    //工频单容
    IF single THEN
        MTBasicsPID_0.Enable         :=1;
        MTBasicsPID_0.PIDParameters := ParaPID;
        MTBasicsPID_0.SetValue       := rLevelSettingP;
        MTBasicsPID_0.ActValue       := rLiquidometerP;

        MTBasicsPID_0.MinOut         :=0;
        MTBasicsPID_0.MaxOut         :=100;
        rSetValve                    :=MTBasicsPID_0.Out;


    ELSIF single = 0 THEN
        MTBasicsPID_0.Enable         :=0;
    END_IF;



//变频水箱 流量比例控制

    IF DOUBLE  THEN
        //主回路

        rSetValve                    :=60;

        //副回路
        MTBasicsPID_1.Enable         := 1;
        MTBasicsPID_1.PIDParameters := ParaPID1;
        MTBasicsPID_1.SetValue       := 2*rFlowmeterP;
        MTBasicsPID_1.ActValue       := rFlowmeterC;
        MTBasicsPID_1.MinOut         := 30;
        MTBasicsPID_1.MaxOut         := 50;
        rSetConverter                := MTBasicsPID_1.Out;

    ELSIF DOUBLE = 0 THEN
        MTBasicsPID_1.Enable         :=0;
```

```
    END_IF;

    (*

    IF (single =0) AND (DOUBLE =0) AND (cascade =0) THEN
        rSetConverter := 0;
        rSetValve     := 0;
    END_IF;
    (*
    IF UPDATE THEN  //参数更新
        MTBasicsPID_1.Enable     :=1;
        MTBasicsPID_0.Enable     :=1;
    ELSE
        MTBasicsPID_1.Enable :=0;
        MTBasicsPID_0.Enable :=0;
    END_IF;
    *)
    MTBasicsPID_0();
    MTBasicsPID_1();


END_PROGRAM
```

Data_manage.st:

```
PROGRAM _CYCLIC

    //数据记录
    FOR i:=0 TO 1 DO
        MpDataRegParCon[i].MpLink    :=ADR(gDataRecorder);
        MpDataRegParCon[i].Enable := TRUE;
    END_FOR;


    MpDataRegParCon[0].PVName := ADR('rSetConverter');
    MpDataRegParCon[1].PVName := ADR('rLiquidometerC');
    FOR i:=0 TO 1 DO
        MpDataRegParCon[i]();
    END_FOR;

    MpDataRecorder_0.MpLink := ADR(gDataRecorder);
    MpDataRecorder_0.Enable := TRUE;
```

```
    MpDataRecorder_0.DeviceName := ADR('USER');
    MpDataRecorder_0.RecordMode := mpDATA_RECORD_MODE_TIME;
//  MpDataRecorder_0.Record := 1;

    MpDataRecorder_0();


END_PROGRAM
```