



SIEMENS



SCE 培训资料

Siemens Automation Cooperates with Education | 2017/05

博途 (TIA Portal) 模块 052-300
SIMATIC S7-1500 的 PID 控制器

Cooperates
with Education

Automation

SIEMENS

本培训资料适用于以下 SCE 教育培训产品

SIMATIC 控制系统

- **SIMATIC ET 200SP Open Controller CPU 1515SP PC F 和 HMI RT SW**
订货号: 6ES7677-2FA41-4AB1
- **SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety**
订货号: 6ES7512-1SK00-4AB2
- **SIMATIC CPU 1516F PN/DP Safety**
订货号: 6ES7516-3FN00-4AB2
- **SIMATIC S7 CPU 1516-3 PN/DP**
订货号: 6ES7516-3AN00-4AB3
- **SIMATIC CPU 1512C PN (带软件和 PM 1507)**
订货号: 6ES7512-1CK00-4AB1
- **SIMATIC CPU 1512C PN (带软件、PM 1507 和 CP 1542-5 (PROFIBUS))**
订货号: 6ES7512-1CK00-4AB2
- **SIMATIC CPU 1512C PN (带软件)**
订货号: 6ES7512-1CK00-4AB6
- **SIMATIC CPU 1512C PN (带软件和 CP 1542-5 (PROFIBUS))**
订货号: 6ES7512-1CK00-4AB7

SIMATIC STEP 7 培训软件

- **SIMATIC STEP 7 Professional V14 SP1 - 单独许可证**
订货号: 6ES7822-1AA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - 6 套课堂许可证包**
订货号: 6ES7822-1BA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - 6 套升级版许可证包**
订货号: 6ES7822-1AA04-4YE5
- **SIMATIC STEP 7 Professional V14 SP1 - 20 件套学生许可证**
订货号: 6ES7822-1AC04-4YA5

请注意, 必要时会使用后续培训产品代替本培训产品。

可通过以下网页获得最新的 SCE 可用培训产品概览: [siemens.com/sce/tp](https://www.siemens.com/sce/tp)

培训课程

如需了解各地的 Siemens SCE 培训课程, 请联系当地的 SCE 联系人 [siemens.com/sce/contact](https://www.siemens.com/sce/contact)

有关 SCE 的其它信息

[siemens.com/sce](https://www.siemens.com/sce)

使用说明

通用型自动化解决方案 - 全集成自动化 (TIA) 的培训资料属于“西门子自动化教育合作项目 (SCE)”, 专门用于公共教育机构和研发机构的培训。Siemens AG 对其内容不承担任何担保责任。

本资料仅可用于 Siemens 产品/系统的首次培训。即允许全部或部分复印本资料并当面转交给培训人员, 令其在培训框架范围内使用。允许在公共培训和进修场合出于培训目的转发、复制本资料或传播其内容。

例外情况需经 Siemens AG 的书面许可。联系人：
Roland Scheuerer 先生 roland.scheuerer@siemens.com。

违者须承担赔偿责任。保留包含翻译在内的所有权利，尤其针对申请专利或实用新型登记注册时的权利。

严禁用于工业客户培训课程。我们绝不允许该资料用于商业目的。

感谢德累斯顿工业大学，特别是 Leon Urbas 教授（工程博士）以及 Michael Dziallas 工程公司和全体人员对本 SCE 培训资料制作过程的支持。

目录

1	目标.....	5
2	前提条件.....	5
3	所需的硬件和软件	6
4	闭环控制技术的理论	7
4.1	闭环控制技术的任务	7
4.2	闭环控制回路的组成部分	8
4.3	用于研究受控系统的阶跃函数	10
4.4	带补偿能力的受控系统.....	11
4.4.1	不带延时的比例受控系统	11
4.4.2	带一次延时的比例受控系统.....	12
4.4.3	带两次延时的比例受控系统.....	13
4.4.4	带 n 次延时的比例受控系统	14
4.5	无补偿能力的受控系统.....	15
4.6	连续控制器的基础类型.....	16
4.6.1	比例控制器 (P 控制器)	17
4.6.2	积分控制器 (I 控制器)	19
4.6.3	PI 控制器.....	20
4.6.4	微分控制器 (D 控制器)	21
4.6.5	PID 控制器	21
4.7	借助振荡试验设置控制器	22
4.8	借助 T_u - T_g 近似法设置控制器.....	23
4.8.1	按照齐格勒-尼科尔斯方法 (Ziegler-Nichols) 设置 PI 控制器.....	24
4.8.2	按照 Chien-Hrones-Reswick 方法设置 PI 控制器	24
4.9	数字控制器	25
5	任务要求.....	27
6	规划.....	27
6.1	PID_Compact 控制块.....	27
6.2	技术示意图	28
6.3	分配表	29
7	结构化的逐步式引导指南	30
7.1	取回一个现有项目	30
7.2	在周期性中断块中调用控制器 PID_Compact	32
7.3	保存程序并编译.....	39
7.4	加载程序.....	40
7.5	观测 PID_Compact	41
7.6	PID_Compact 预整定.....	43

7.7	PID_Compact 精整定.....	46
7.8	项目归档.....	49
8	检查清单.....	50
9	更多相关信息	51

SIMATIC S7-1500 的 PID 控制器

1 目标

本章学习的是如何利用 TIA Portal 编程工具在 SIMATIC S7-1500 上使用软件 PID 控制器。

该课程单元介绍了 PID 控制器在 SIMATIC S7-1500 中的调用、互连、配置和整定。其中, 将逐步介绍在博途 (TIA Portal) 中调用 PID 控制器以及将 PID 控制器关联至应用程序的方法。

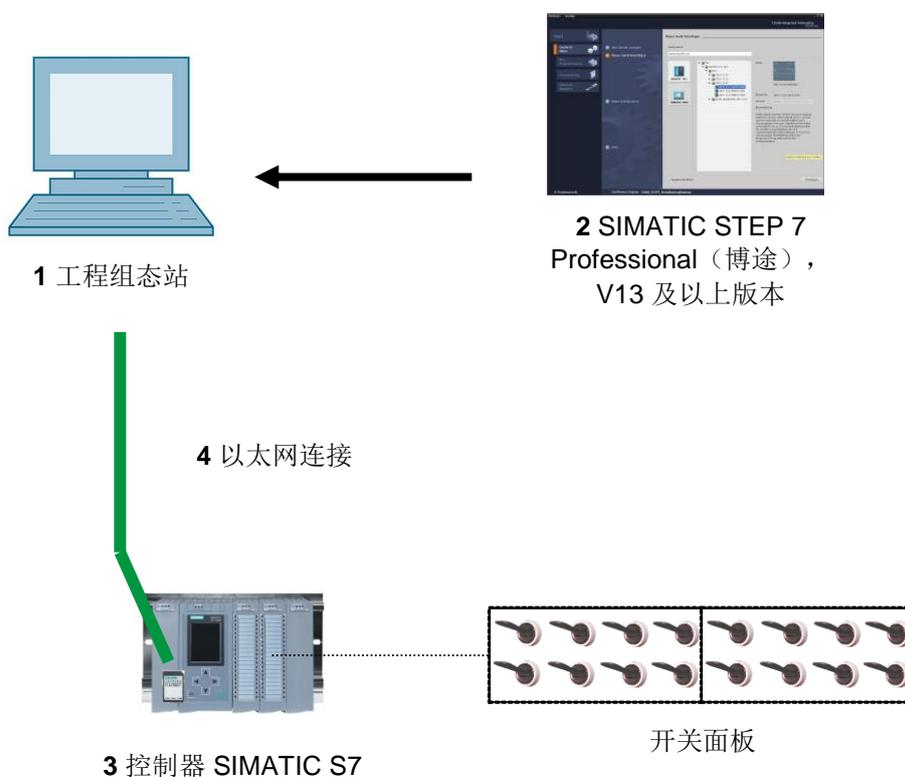
可以使用第 3 章所述的 SIMATIC S7 控制器。

2 前提条件

本章的基础是“SIMATIC S7 CPU1516F-3 PN/DP 的模拟值”一章。为完成本章的学习, 您可能需要重新温习如下项目: “SCE_ZH_032-500_Analog_Values_R1508.zap13”。

3 所需的硬件和软件

- 1 工程组态站：硬件和操作系统是工程组态站的前提
(更多信息参见博途 (TIA Portal) 安装 DVD 里的自述文件)
- 2 博途 (TIA Portal) 中的 SIMATIC STEP 7 Professional 软件 – V13 及以上版本
- 3 控制器 SIMATIC S7-1500/S7-1200/S7-300, 例如 CPU 1516F-3 PN/DP – 固件 V1.6 及以上版本, 带存储卡和 16DI/16DO 以及 2AI/1AO
提示：数字输入端和模拟输入/输出端应布线至开关面板。
- 4 工程组态站和控制器之间的以太网连接



4 闭环控制技术的理论

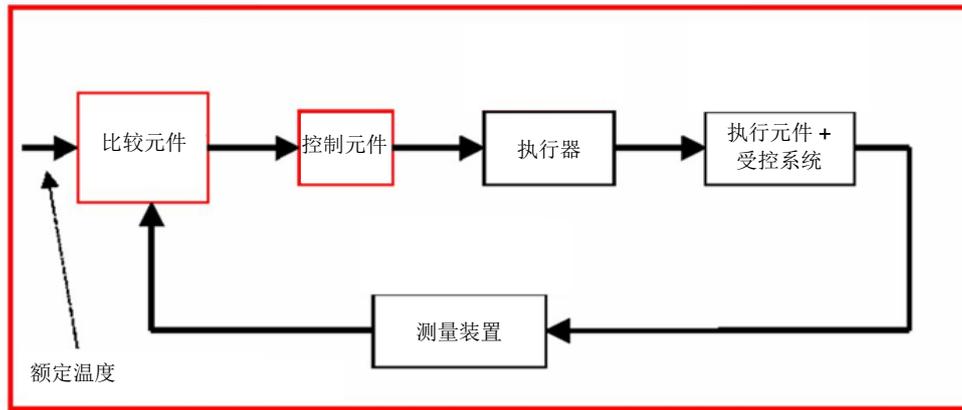
4.1 闭环控制技术的任务

闭环控制是一种基于对变量的测量而不断进行干预，以此创建并保持变量值的过程。

由此产生一个作用流程，该作用流程在一个闭合的环路，即闭环控制回路中实现。由于该过程基于对变量的测量，反过来又会作用于变量自身。

因此，受到控制的变量会接受持续的测量，并与另一个相同类型的预设变量做比较。根据比较结果，通过闭环控制过程使受控变量接近预设变量的值。

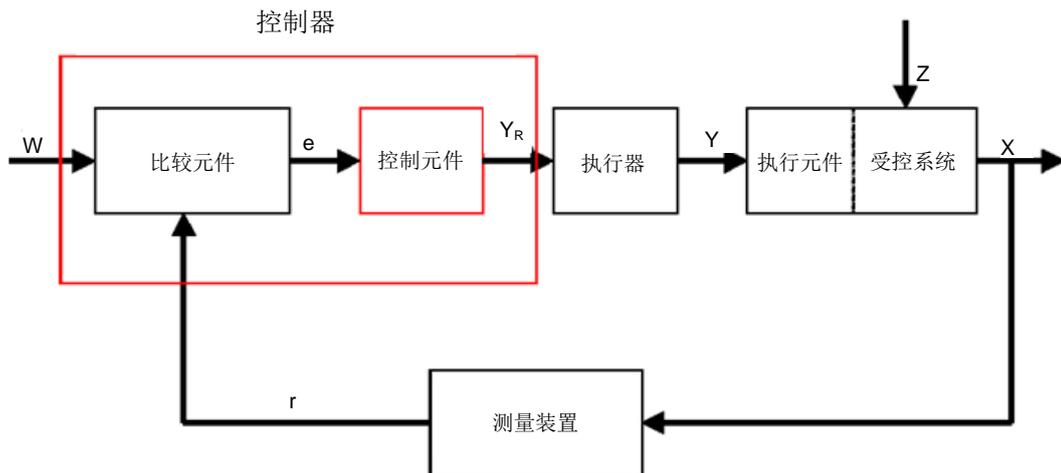
闭环控制流程图



4.2 闭环控制回路的组成部分

下面将分别介绍闭环控制技术中最基础的术语。

首先通过这张图形成大致的了解：



1. 受控变量 x

这是闭环控制的本质“目标”，即那些受到影响，用以维持整个系统目的变量。我们在此以室温为例。受控变量在某一特定时刻产生的瞬时值被称为这一时刻的“实际值”。

2. 反馈变量 r

在闭环控制回路中，受控变量会接受持续的检查，以应对异常变化。与受控变量成比例变化的测量变量被称为反馈变量。以“供暖装置”为例，反馈变量相当于室内温度计的测量电压。

3. 干扰变量 z

干扰变量是无意间对受控变量造成影响并使其偏离当前设定值的变量。在固定值控制的情况下，则有必要检查是否存在干扰变量。在供暖系统中，干扰变量是指外界温度或其他致使室温偏离理想值的变量。

4. 额定值 w

某一时刻的额定值是指在这一时刻受控变量应达到的最佳值。需要注意的是，在从属值控制中，额定值有时会不断变化。例如，额定值有可能是目前所需的室温。

5. 比较元件

这是受控变量的当前测量值与基准变量的瞬时值相比较的一个点。大多数情况下，两个变量均表现为测量电压。两个变量的差值为“控制偏差” e 。该偏差将被传至控制元件进行分析（见下面）。

6. 控制元件

控制元件是闭环控制的核心。它分析控制偏差，即受控变量与当前额定值是否偏差，偏差多少，并将这些信息作为输入变量。从输入变量中推导出“控制器输出变量” Y_R ，该输出变量在最后的的结果中影响受控变量。例如在供暖系统中，控制器输出变量是指混合器电机的电压。

控制元件如何根据控制偏差来确定控制器输出变量的方式方法，是闭环控制的主要标准。

7. 执行器

执行器被称为闭环控制的“执行机构”。它从控制元件中获得以“控制器输出变量”为形式的信息，该信息指出应如何影响受控变量。之后执行器根据该信息使“调节变量”发生变化。在我们的示例中执行器是指混合器电机。

8. 执行元件

这是闭环控制回路中根据调节变量 Y 影响受控变量（直接变大或变小）的元件。在示例中为混合器、供暖导线和散热器的组合。混合器的设置（调节变量）通过混合器电机（执行器）进行并通过水温来影响室温。

9. 受控系统

受控系统是指受控变量所在的系统，例如供暖装置或居室。

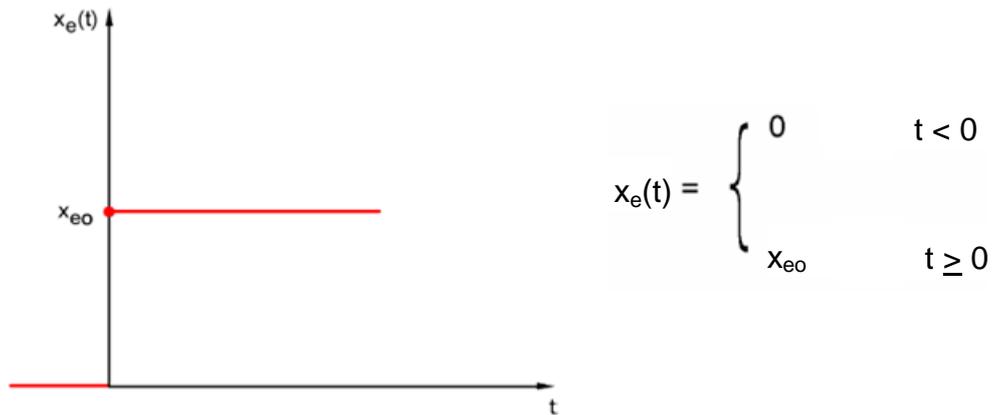
10. 死区时间

死区时间是指从控制器输出变量开始变化到受控系统做出可测反应的一段时间。在示例中，即混合器电机的电压发生变化到室温由此产生可测变化之间的时间。

4.3 用于研究受控系统的阶跃函数

为了研究受控系统、控制器和闭环控制回路的行为，将使用一种统一的函数作为输入信号，即阶跃函数。

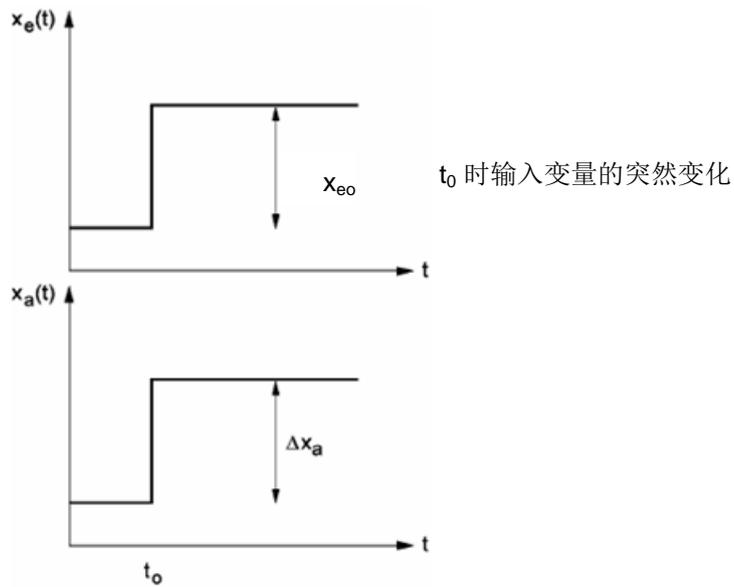
根据研究对象是闭环控制回路的一个元件还是整个闭环控制回路，来决定是将阶跃函数分配给受控变量 $x(t)$ 、调节变量 $y(t)$ 、参考变量 $w(t)$ 还是干扰变量 $z(t)$ 。因此输入信号，即阶跃函数，往往设为 $x_e(t)$ ，而输出信号设为 $x_a(t)$ 。



4.4 带补偿能力的受控系统

4.4.1 不带延时的比例受控系统

该受控系统简称为 P 系统。



受控变量/调节变量:

$$x = K_{ss} \cdot y \quad K_{ss}: \text{调节变量变化的比例值}$$

$$K_{ss} = \frac{\Delta x}{\Delta y} = \tan \alpha$$

受控变量/干扰变量:

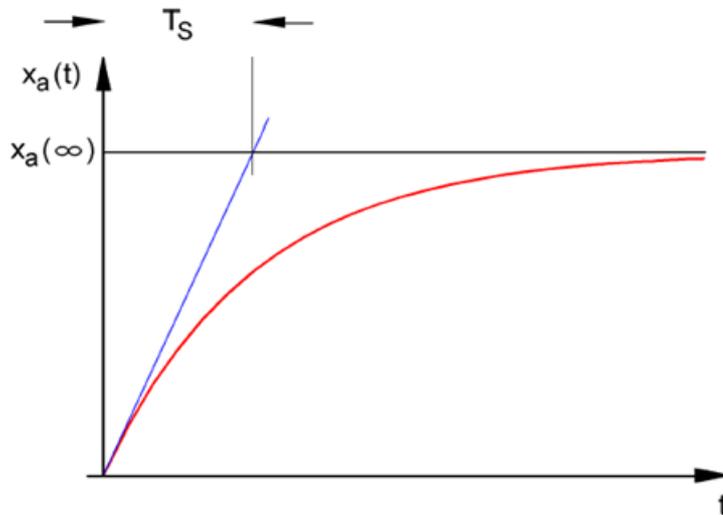
$$x = K_{sz} \cdot z \quad K_{sz}: \text{干扰变量变化的比例值}$$

调节范围: $y_h = y_{\max} - y_{\min}$

受控范围: $x_h = x_{\max} - x_{\min}$

4.4.2 带一次延时的比例受控系统

该受控系统简称为 P-T1 系统。



一般输入信号的微分方程 $x_e(t)$:

$$T_S \cdot \dot{x}_a(t) + x_a(t) = K_{PS} \cdot x_e(t)$$

以阶跃函数作为输入信号的微分方程的解（阶跃响应）：

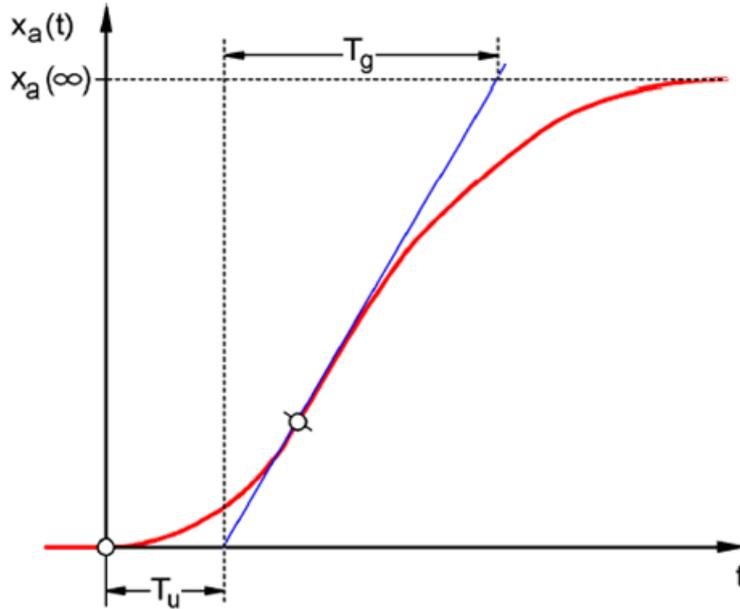
$$x_a(t) = K_{PS} (1 - e^{-t/T_S}) \cdot x_{e0}$$

$$x_a(t = \infty) = K_{PS} \cdot x_{e0}$$

T_S : 时间常量

4.4.3 带两次延时的比例受控系统

该受控系统简称为 P-T2 系统。



T_u: 延迟时间 T_g: 补偿时间

该系统是由通过将两个互不关联的 P-T1 系统串联连接而生成的，这两个时间常量分别为 TS1 和 TS2

P-Tn 系统的可控性:

$$\frac{T_u}{T_g} < \frac{1}{10} \rightarrow \text{易于控制}$$

$$\frac{T_u}{T_g} \approx \frac{1}{6} \rightarrow \text{仍可控}$$

$$\frac{T_u}{T_g} > \frac{1}{3} \rightarrow \text{难以控制}$$

T_u / T_g 比值越高，系统越难以控制

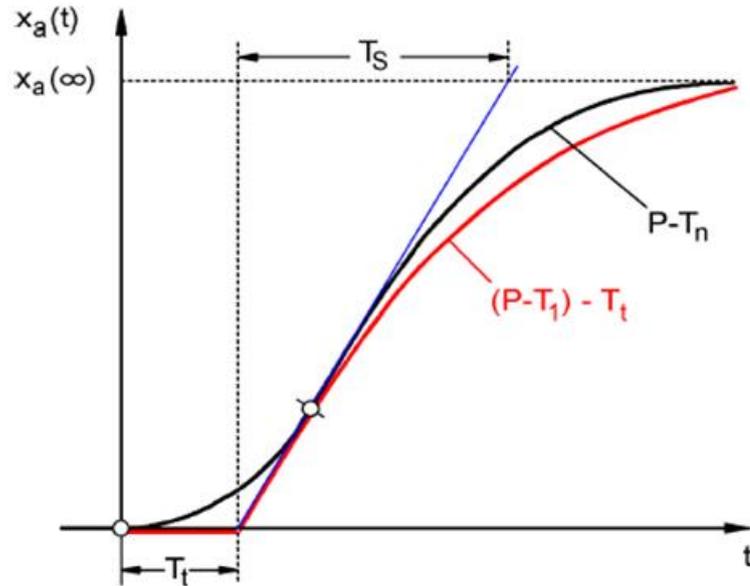
4.4.4 带 n 次延时的比例受控系统

该受控系统简称为 P-T $_n$ 系统。

其时间特性由一个 n 阶微分方程来描述。阶跃响应的曲线类似于 P-T $_2$ 系统的情况。时间特性通过 T_u 和 T_g 描述。

替代：将一个 P-T $_1$ 系统和一个死区时间系统串联起来，便可以替代具有多个延时的受控系统，二者的作用原理差不多。

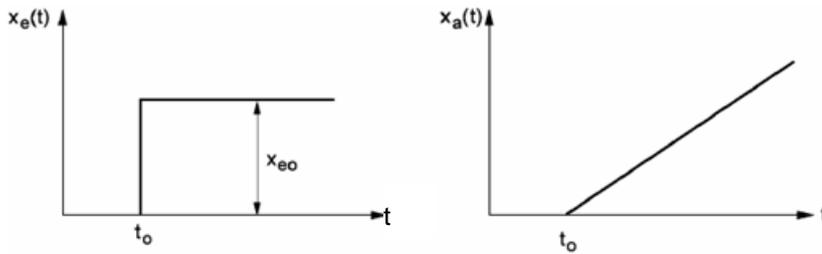
这适用于： $T_t \gg T_u$ 和 $T_S \gg T_g$ 。



4.5 无补偿能力的受控系统

该受控系统简称为 I 系统。

系统出现扰动之后, 受控变量会持续变化, 而不会达到一个固定的最终值。

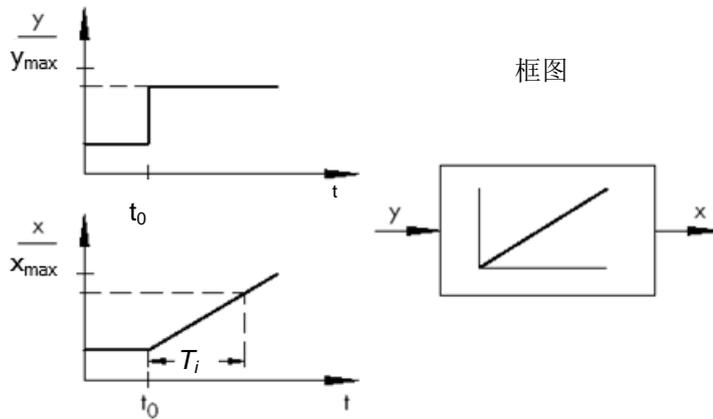


示例: 液位控制

一个带排水的容器, 当注水量和排水量一致时, 便保持恒定的液位高度。若注水或排水的流量发生变化, 则液位上升或下降。注水和排水之间的差值越大, 液位变化得越快。

这个示例说明, 在实际应用中, 积分为通常具有一定的局限性。受控变量只在达到系统决定的极限值前增加或降低: 所谓系统决定极限值, 例如容器溢出或排空, 压力达到最大或最小等。

插图展示了 I 系统在输入变量发生突变时的时间特性以及由此得出的框图。



如果将输入端上的阶跃函数变成一个任意函数 $x(t)$, 则

$$x_a(t) = K_{IS} \int x_e(t) dt \rightarrow \text{对受控系统进行积分}$$

K_{IS} : 受控系统的积分系数

* 插图出自 SAMSON 技术信息 - L102 - 控制器和受控系统, 版本: 2000 年 8 月
(http://www.samson.de/pdf_en/l102en.pdf)

4.6 连续控制器的基础类型

离散的控制只启用/停用一个或两个调节变量，具有简单方便的优势。这种优势不仅体现在控制器本身，同时也体现在执行器和执行元件上，因此比连续控制器更加便宜。

当然，非连续控制器也有一系列不足。例如当接入了大型电动机或冷却机组等大负载时，启动时的高负载峰值会造成例如电源过载。出于这一原因，人们通常并非在“开”和“关”之间进行切换，而是在执行器/执行元件的全功率（“满载”）和明显较小的功率（“基本负载”）之间进行切换。但即便是有了这样的改进，非连续控制器依然不适用于许多应用。比如，让我们假设车辆发动机的转速是非连续受控的。我们只有空转和全速两种选择。可以看出，在突然全速时动力根本无法合理地通过轮胎传递到路面，这样的车辆根本不能在道路交通中行驶。

因此在这一类的应用中，人们使用连续控制器。从理论上讲，控制元件在控制偏差和控制器输出变量之间创建的数学关系在这其中几乎不受限制。在实际应用中，人们分出三种经典的基础类型，下面将详细进行说明。

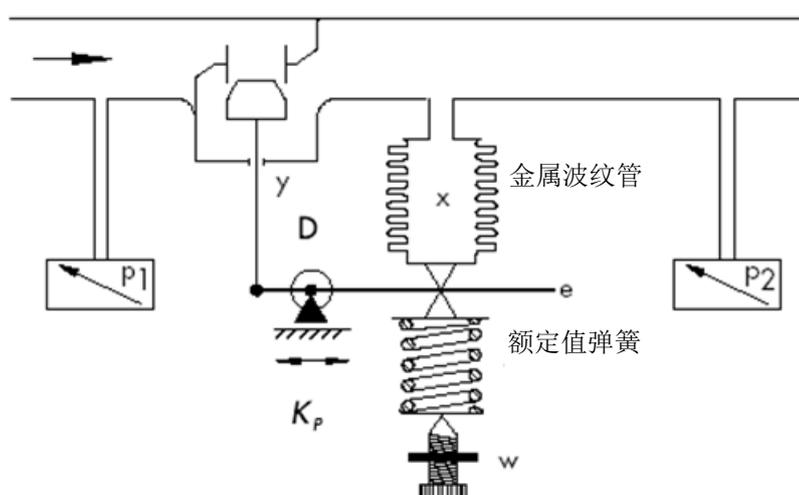
4.6.1 比例控制器 (P 控制器)

在 P 控制器中, 调节变量 y 始终与测得的控制偏差 ($y \sim e$) 成比例。由此得出, P 控制器会对控制偏差做出无延迟的响应, 并且只有存在偏差 e 时才生成调节变量。

图中草绘的比例式压力控制器比较额定值弹簧的弹力 FS 和压力 p_2 在弹性金属波纹管上产生的力 FB 。若力不平衡, 杠杆围绕旋转点 D 转动。阀门位置 \bar{n} 发生变化, 相应地, 受控压力 p_2 也发生变化, 直到重新形成力均衡状态。

突然出现控制偏差时, P 控制器的行为如下图所示。调节变量变动的幅度 y 取决于控制偏差 e 的高度以及比例系数 K_p 的大小。

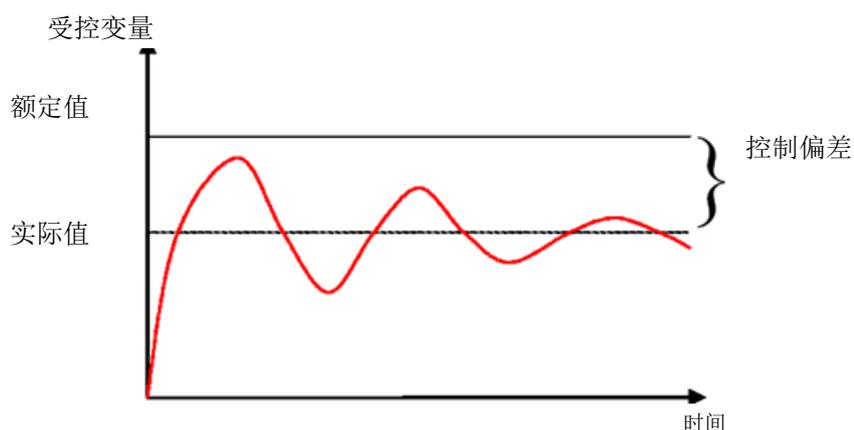
为了使控制偏差尽可能小, 必须选择一个尽可能大的比例系数。系数越大, 控制器的反应越快。但过高的值也会造成控制器过调以及显著的振荡趋势。



$$y = K_p \cdot e$$

* 插图和文字出自 SAMSON 技术信息 - L102 - 控制器和受控系统, 版本: 2000 年 8 月
(http://www.samson.de/pdf_en/l102en.pdf)

下图表现了 P 控制器的特性:



这种控制器类型的优势一方面在于其简易性（在最简单的情况下，控制器的电子系统仅通过一个单纯的电阻便可以实现），另一方面，相对于其他类型来说它的响应相当迅速。

P 控制器最大的不足在于永久性的控制偏差，即便是经过长时间的调节，也无法完全达到额定值。即便是设置了更高的比例系数，也无法改善这种缺点以及仍不够理想的反应速度，而系数过高反而会造成控制器过调，即反应过度。在最不利的情况下，控制器长期处于振荡状态，受控变量由于控制器自身的原因定期与额定值发生偏离，而非由于干扰变量的原因。

这种长期出现控制偏差的问题最好通过接入一个附加的积分控制器来解决。

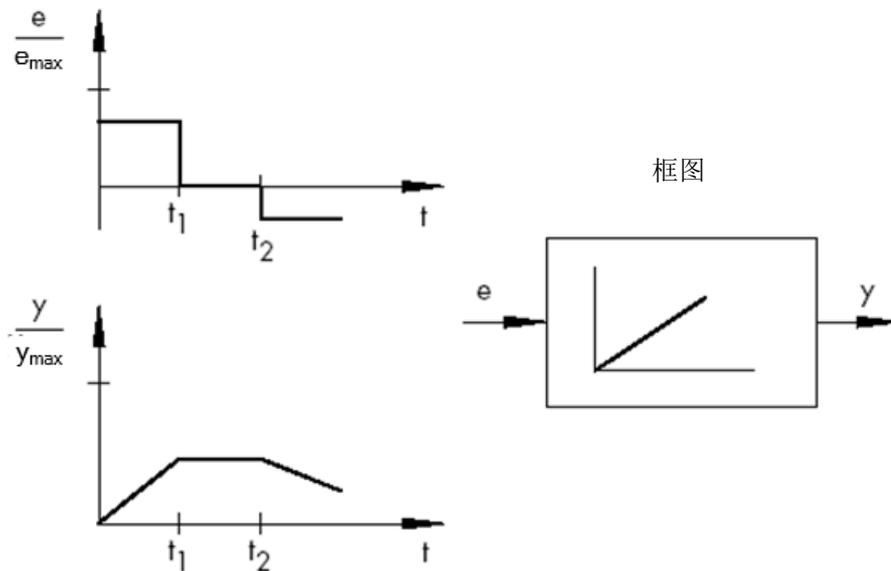
4.6.2 积分控制器 (I 控制器)

使用积分控制器的目的在于全面补偿每个运行点的控制偏差。只要控制偏差不为零，调节变量的大小就会一直变化。只有当参考变量和受控变量一样大时，最迟在调节变量达到其由系统决定的极限值时（ U_{max} 、 P_{max} 等），闭环控制才算进入平稳状态。

该积分操作的数学公式为：调节变量与控制偏差 e 的时间积分成比例：

$$y = K_i \int e dt \quad \text{其中: } K_i = \frac{1}{T_n}$$

调节变量上升（或下降）的速度取决于控制偏差和积分时间。

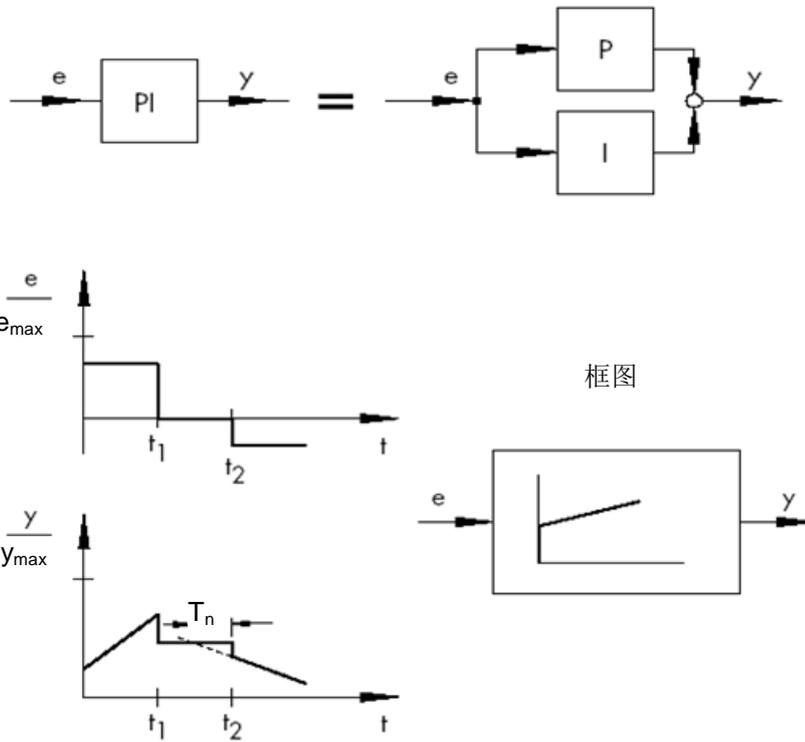


* 插图和文字出自 SAMSON 技术信息 - L102 - 控制器和受控系统, 版本: 2000 年 8 月
(http://www.samson.de/pdf_en/l102en.pdf)

4.6.3 PI 控制器

PI 控制器是一种在实际应用中最常采用的控制器类型。它由一个 P 控制器和一个 I 控制器并联而成。

若设计妥当，它可以结合两种控制器的优势（稳定快速，没有控制偏差），同时又能弥补其不足之处。



时间特性由比例系数 K_p 和重调时间 T_n 表示。基于比例部分，调节变量对各个控制偏差 e 立即作出反应，同时积分部分则会随着时间发生作用。其中 T_n 所代表的时间是指，在这段时间之后，积分部分产生的控制振幅才会与比例部分 (K_p) 从一开始产生的控制振幅相同。如果人们要提高积分部分，必须如同在 I 控制器中那样减少重调时间 T_n 。

控制器设计：

通过设置变量 K_p 和 T_n ，能以牺牲控制的动态特性为代价来降低受控变量的过调量。

PI 控制器的应用范围：不允许出现控制偏差的快速控制回路。

示例：压力控制、温度控制、比例控制。

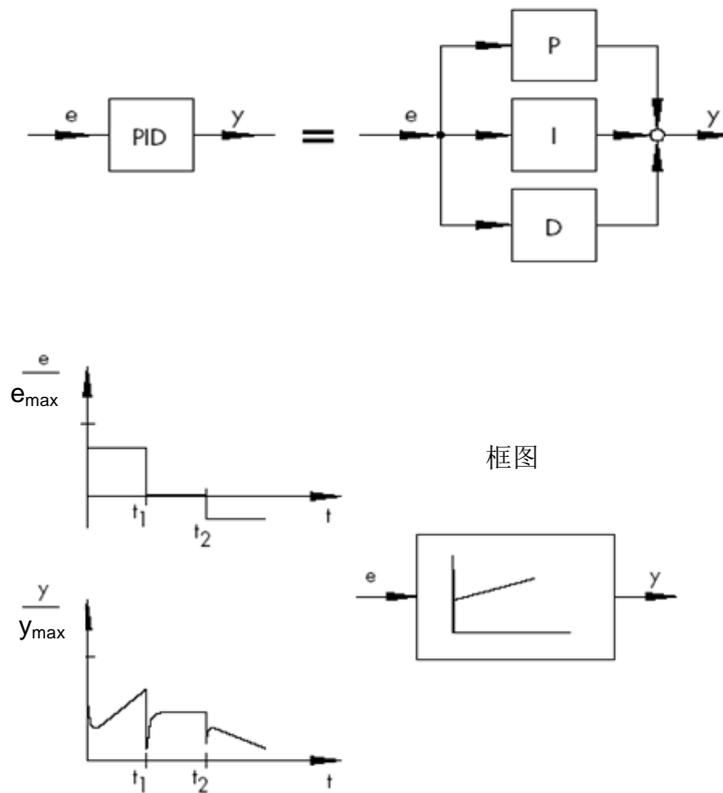
* 插图和文字出自 SAMSON 技术信息 - L102 - 控制器和受控系统，版本：2000 年 8 月
http://www.samson.de/pdf_en/l102en.pdf

4.6.4 微分控制器 (D 控制器)

D 控制器由控制偏差的变化速度得出调节变量，而不是像 P 控制器那样由其振幅得出。因此它比 P 控制器的反应速度更快：甚至是在控制偏差较小时，只要出现振幅变化，它都可以立即产生较大的控制振幅。反之，如果控制偏差的变化速度为零，无论偏差多大，D 控制器都不会识别。因此，在实际工作中，人们几乎不单独使用 D 控制器。它通常与其他控制元件组合使用，大多是连接一个比例控制器。

4.6.5 PID 控制器

为一个 PI 控制器扩展一个微分部分，便得到一个 PID 控制器。在 PD 控制器中，补充一个微分部分的作用在于，在设计得当的情况下，受控变量可以更早地达到其额定值并且更迅速地进入稳定状态。



$$y = K_p \cdot e + K_i \int e dt + K_D \frac{de}{dt} \quad \text{其中: } K_i = \frac{K_p}{T_n}; K_D = K_p \cdot T_V$$

* 插图和文字出自 SAMSON 技术信息 - L102 - 控制器和受控系统，版本：2000 年 8 月
(http://www.samson.de/pdf_en/l102en.pdf)

4.7 借助振荡试验设置控制器

为了实现令人满意的控制效果, 选择一个合适的控制器是十分重要的。同样的, 设置合适的控制参数 K_p 、 T_n 和 T_v , 使其与受控系统相匹配也是必要条件之一。大多数情况下, 人们需要在非常稳固但缓慢的控制方式与异常灵活但不安定的控制行为间取得平衡, 尤其是后者有时还会造成振荡和不稳定。

在面对始终应在相同运行点工作的非线性受控系统时, 例如采用固定值控制方式时, 必须调整控制参数, 使其在这一工作点上与受控系统特性相匹配。若采用从属控制方式 \tilde{n} , 则可以不定固定工作点, 在这种情况下, 控制设置必须确保在整个工作范围内达到足够快速且稳定的控制效果。

实际应用中, 控制器常常按照经验值进行设置。

若不存在经验值, 则必须仔细分析受控系统特性, 以便之后借助各种理论上的或实际中的设计方法来确定适合的控制器参数。

其中一种就是根据齐格勒-尼科尔斯方法 (Ziegler-Nichols) 进行的振荡试验。它设计简便, 并且适用于多种场合。但这种设置方法只能在允许自己的受控变量产生自振的受控系统中运用。

具体方法如下:

- 将控制器的 K_p 和 T_v 设为最小值, T_n 设为最大值 (控制器的效果可能最差)。
- 将受控系统手动调整到所需工作点 (到达控制点)。
- 将控制器的调节变量设为一个手动预设的值并切换到自动模式。
- 不断增加 K_p (减小 X_p), 直至识别到受控变量的谐波振荡。如果可以, 在 K_p 变化期间可小幅度改变控制回路的额定值, 以此激励振荡。
- 记下设定的 K_p 值, 作为临界的比例系数 $K_{p, crit}$ 。确定整个振荡持续的时间 T_{crit} , 必要时通过秒表得出多个振荡的算术平均值。
- 将 $K_{p, crit}$ 和 T_{crit} 的值与表中的因数相乘, 按照得出的值设置控制器的 K_p 、 T_n 和 T_v 。

	K_p	T_n	T_v
P	$0.50 \times K_{p, crit}$	-	-
PI	$0.45 \times K_{p, crit}$	$0.85 \times T_{crit}$	-
PID	$0.59 \times K_{p, crit}$	$0.50 \times T_{crit}$	$0.12 \times T_{crit}$

* 插图和文字出自 SAMSON 技术信息 - L102 - 控制器和受控系统, 版本: 2000 年 8 月 (http://www.samson.de/pdf_en/l102en.pdf)

4.8 借助 T_u - T_g 近似法设置控制器

在此以一个 P-T2 受控系统为例讲解如何设置受控系统。

T_u - T_g 近似法

齐格勒-尼科尔斯方法 (Ziegler-Nichols) 和 Chien-Hrones-Reswick 方法的基础是 T_u - T_g 近似法，它利用受控系统的阶跃响应来确定受控系统的传输系数 K_S 、延迟时间 T_u 和平衡时间 T_g 。

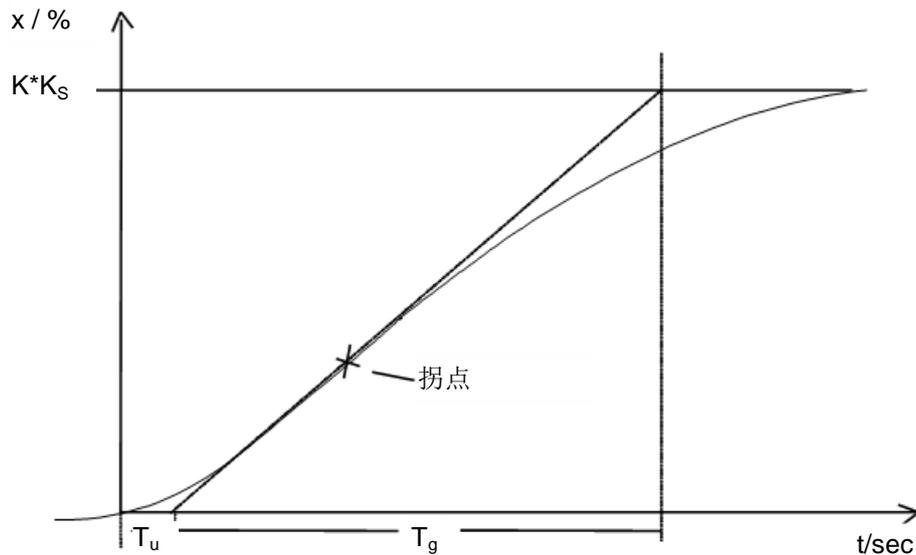
下述设置规则是通过使用模拟计算机仿真在实验基础上确立的。

P- T_N 受控系统可以通过所谓的 T_u - T_g 近似法，即通过对 P-T1-TL 系统做近似，来进行足够准确的描述。

起始点是高度为 K 的阶跃输入信号下的系统阶跃响应。如图所示，确定系统传输系数 K_S 、延迟时间 T_u 和平衡时间 T_g 等所需参数。

其中，有必要测量最高为静态上限值 ($K \cdot K_S$) 范围内的传递函数，这样才能确定计算所需的系统传输系数 K_S 。

这种方法的重要优势在于，即使受控系统无法用解析方法来进行描述，仍可使用近似法。



图： T_u - T_g 近似法

4.8.1 按照齐格勒-尼科尔斯方法 (Ziegler-Nichols) 设置 PI 控制器

在研究 P-T₁-T_L 系统的过程中, 齐格勒和尼科尔斯发现了以下针对定值控制的最佳控制器设置:

$$K_{PR} = 0.9 \frac{T_g}{K_S T_u}$$

$$T_N = 3.33 T_u$$

使用这些设置值一般都得到非常好的抗干扰性。

4.8.2 按照 Chien-Hrones-Reswick 方法设置 PI 控制器

对于这种方法, 为了获得最有效的控制器参数, 需要对命令性能以及抗干扰性进行研究。在两种情况中会得到不同的值。此外, 指定的两种不同的设置方式分别可以满足不同的控制质量需求。

得出的设置结果如下:

- 针对抗干扰性:

非周期性瞬态过程, 持续时间最短

20% 过调, 振荡持续时间最短

$$K_{PR} = 0.6 \frac{T_g}{K_S T_u}$$

$$K_{PR} = 0.7 \frac{T_g}{K_S T_u}$$

$$T_N = 4 T_u$$

$$T_N = 2.3 T_u$$

- 针对命令性能:

非周期性瞬态过程, 持续时间最短

20% 过调, 振荡持续时间最短

$$K_{PR} = 0.35 \frac{T_g}{K_S T_u}$$

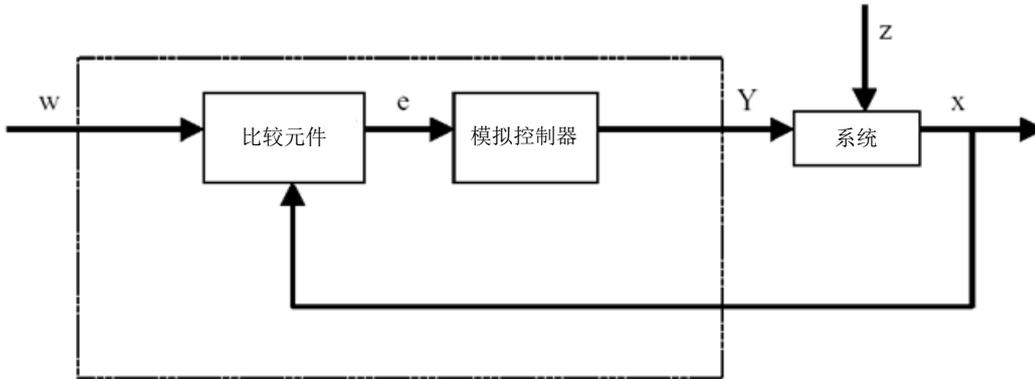
$$K_{PR} = 0.6 \frac{T_g}{K_S T_u}$$

$$T_N = 1.2 T_g$$

$$T_N = T_g$$

4.9 数字控制器

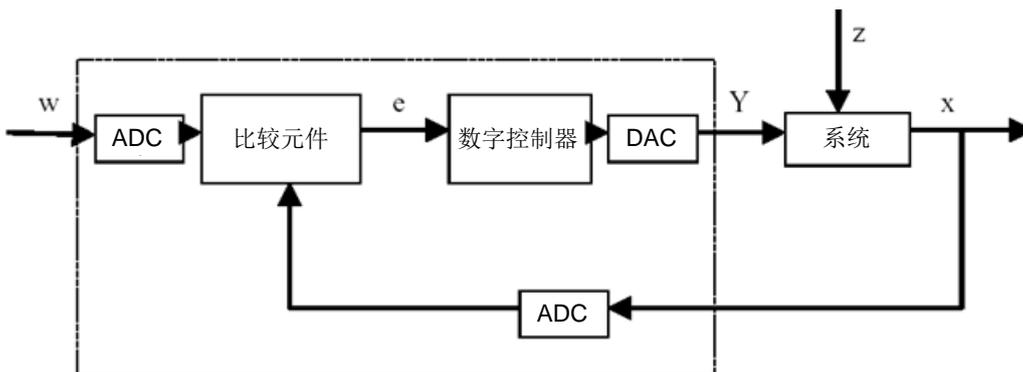
到目前为止，我们所讨论的主要是模拟控制器，即根据模拟值形式的控制偏差，利用模拟方法来得出控制器输出变量的控制器。这样一种闭环控制回路的示意图如下：



但是，以数字方式评估控制偏差其实更具优势。一方面，如果控制偏差与控制器输出变量间的关系通过可用于计算机编程的算法或公式来定义，而不是以模拟电路的形式来实现，那么这种关系将更容易确定。另一方面，数字技术能够实现集成度更高的电路，从而可以在狭小的空间中容纳多个控制器。最后一点是，如果计算能力足够大，通过分配计算时间，即使仅仅一个控制器也可以用作多个闭环回路的控制元件。

为了能够对变量进行数字化处理，首先必须在模数转换器 (ADC) 中将参考变量以及反馈变量转换成数字变量。数字比较元件会对这两个变量作减法运算，其差值会传递到数字控制元件当中。然后，再将控制器输出变量通过数模转换器 (DAC) 转换成模拟变量。从外部看来，这个由转换器、比较元件和控制元件组成的单元就像是一个模拟控制器。

数字控制器的结构如下图所示：



控制器数字转换在具有上述优势的同时, 也具有一定的问题。例如为数字控制器选取的某些变量必须足够大, 这样才能确保在数字化过程中, 控制精度不会受到过分影响。

数字计算机的性能标准:

— 数模转换器的量化分辨率

它是指连续值范围内数值栅格化的精细程度。选择的分辨率必须足够大, 确保控制所需的重要细节不会丢失。

— 模数转换器的采样速率

这是指对转换器中有待处理的模拟值进行测量以及数字化的频率。采样率必须足够快, 确保控制器在受控变量突然发生改变时也能及时反应。

— 周期时间

与模拟计算机不同, 每个数字计算机都按时钟周期工作。所用计算机的速度必须非常快, 以便受控变量在一个时钟周期 (即对输出值进行计算, 而不对输入值进行轮询的时间段) 内不会发生显著变化。

数字控制器的性能必须足够高, 使其从外部看来其响应速度和准确率均与模拟控制器相当。

5 任务要求

在本章节中，应为“SCE_ZH_032-500 模拟值”一章中的程序扩展出用于转速控制的 PID 控制器。为此必须删除对“MOTOR_SPEEDCONTROL”[FC10] 功能的调用。

6 规划

针对闭环控制技术，在 TIA Portal 中有工艺对象 PID_Compact 可用。

该工艺对象替代“MOTOR_SPEEDCONTROL”[FC10] 块，对电机转速进行闭环控制。

这是对“032-500_Analog_Values”项目的扩展。该项目必须事先取回。

必须在组织块“Main”[OB1] 中删除对“MOTOR_SPEEDCONTROL”[FC10] 功能的调用，之后才能在一个周期性中断块中调用工艺对象并互连。

然后必须对工艺对象 PID_Compact 进行配置和调试。

6.1 PID_Compact 控制块

工艺对象 PID_Compact 为按比例作用的执行元件提供一个集成有整定功能的 PID 控制器。

可实现以下运行方式：

- 未激活
- 预整定
- 精整定
- 自动模式
- 手动模式
- 具有故障监控的备用输出值

这里，应针对自动模式对此控制器进行互连、参数设置和调试。

在调试过程中，我们会借助集成式整定算法，并绘制出受控系统的控制行为。

始终从一个周期性中断块中调用工艺对象 PID_Compact，其周期时间在此固定设为 50 ms。

转速额定值在工艺对象的输入端“Setpoint”上预设为一个常数，单位：转/分钟（范围： ± 50 rpm）。此处的数据类型为 32 位浮点数 (Real)。

转速实际值 -B8（电机的转速实际值传感器 $\pm 10V$ 相当于 ± 50 rpm）被录入到输入端“Input_PER”上。

控制器的输出端“Output_PER”直接与信号 -U1（电机在两个方向上的转速调节变量 $\pm 10V$ 相当于 ± 50 rpm）互连。

控制器只在输出端 -Q3（输送带电机 -M1 可变转速）受控时激活。若输出端未受控，则通过接入输入端“Reset”来使控制器静止。

6.2 技术示意图

在此处可查看有关任务要求的技术示意图。

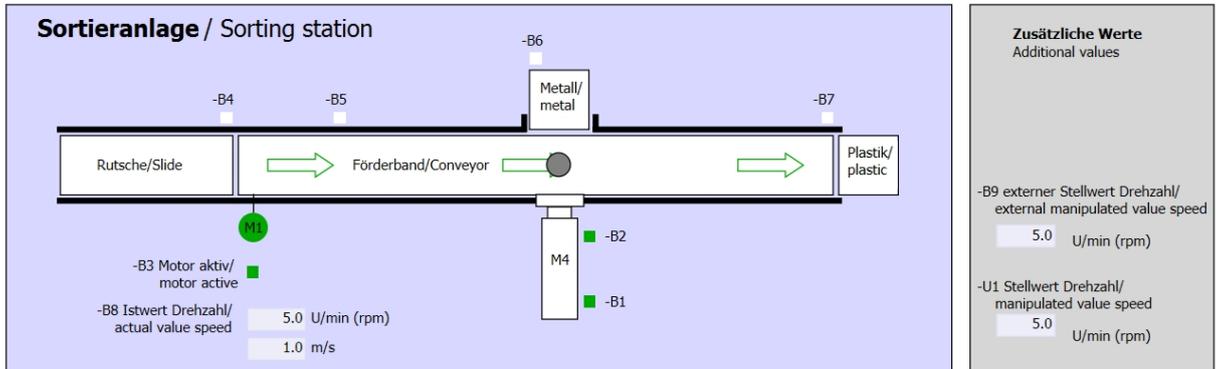


图 1: 技术示意图

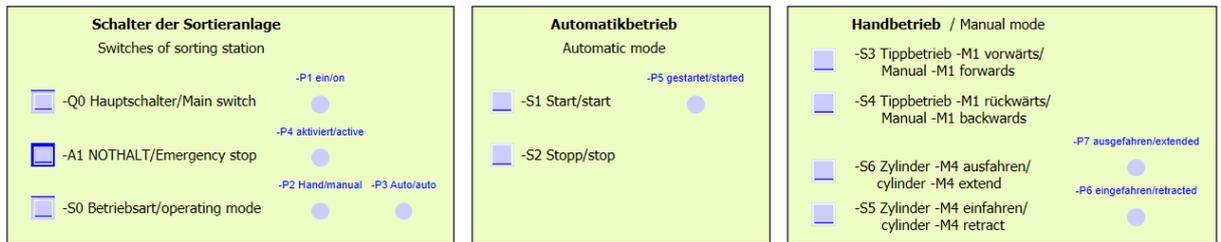


图 2: 控制面板

6.3 分配表

在该任务中需要使用以下信号作为全局操作数。

DI	类型	标号	功能	NC/NO
I 0.0	BOOL	-A1	发出“紧急停机 ok”报警	NC
I 0.1	BOOL	-K0	机组“接通”	NO
I 0.2	BOOL	-S0	手动 (0)/自动 (1) 模式选择开关	手动 = 0 自动 = 1
I 0.3	BOOL	-S1	自动模式启动按钮	NO
I 0.4	BOOL	-S2	自动模式停止按钮	NC
I 0.5	BOOL	-B1	“气缸 -M4 已驶入”传感器	NO
I 1.0	BOOL	-B4	“滑道已占用”传感器	NO
I 1.3	BOOL	-B7	“部件位于输送带末端”传感器	NO
IW64	BOOL	-B8	电机转速实际值传感器 +/-10V 相当于 +/- 50 rpm	

DO	类型	标号	功能	
Q 0.2	BOOL	-Q3	输送带电机 -M1 可变转速	
QW 64	BOOL	-U1	电机在两个方向上的转速调节值 +/-10V 相当于 +/- 50 rpm	

分配表的缩写说明

DI	数字输入	DO	数字输出
AI	模拟输入	AO	模拟输出
I	输入	Q	输出
NC	Normally Closed (常闭触点)		
NO	Normally Open (常开触点)		

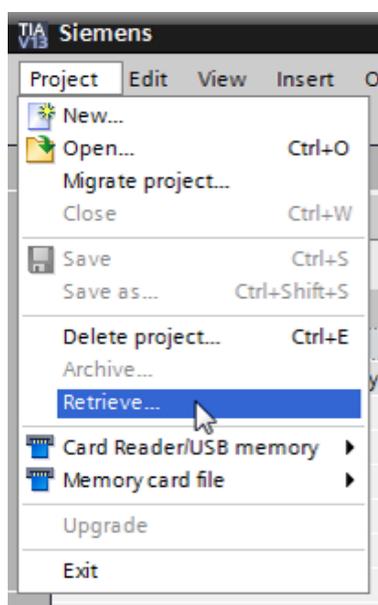
7 结构化的逐步式引导指南

以下是帮助您实现规划的引导指南。如果您已经充分了解，只需要使用带标号的步骤标题作为参考。否则，则需要遵从引导指南以下步骤中的详细说明。

7.1 取回一个现有项目

→ 在扩展章节“SCE_ZH_032-500 模拟值”中的“SCE_ZH_032-500_Analog_Values_R1508.zap13”项目之前，需要先将其取回。为了取回现有项目，必须在项目视图中通过 → 项目 → (Project) 搜索相应的压缩包。然后用“打开”(Open) 确认您的选择。

(→ 项目 → 取回 → 选择一个 .zap 文件包 → 打开)

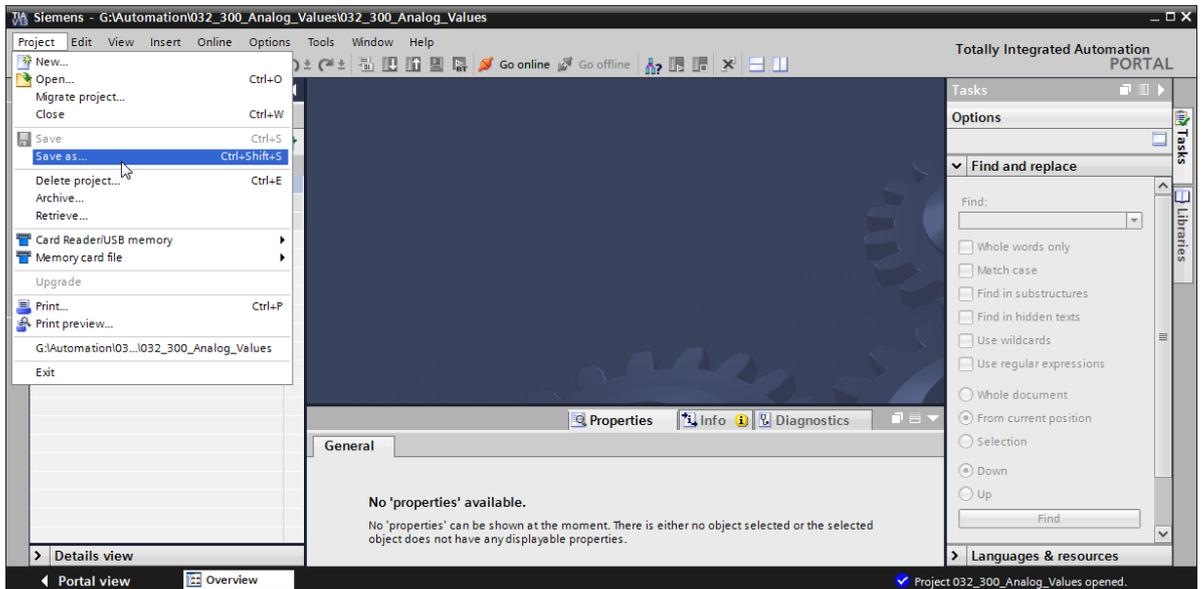


→ 接下来可以选择用于保存取回项目的目标目录。用“确定”(OK) 按钮确认您的选择。

(→ 目标目录 → 确定)

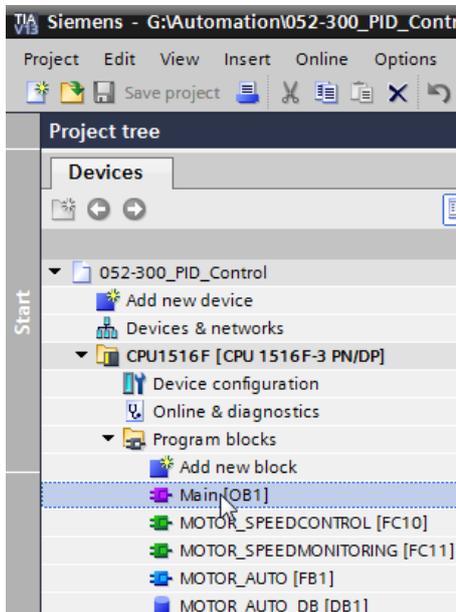
→ 将打开的项目另存为 052-300_PID_Controller。

(→ 项目 → 另存为 ... → 052-300_PID_Controller → 保存)



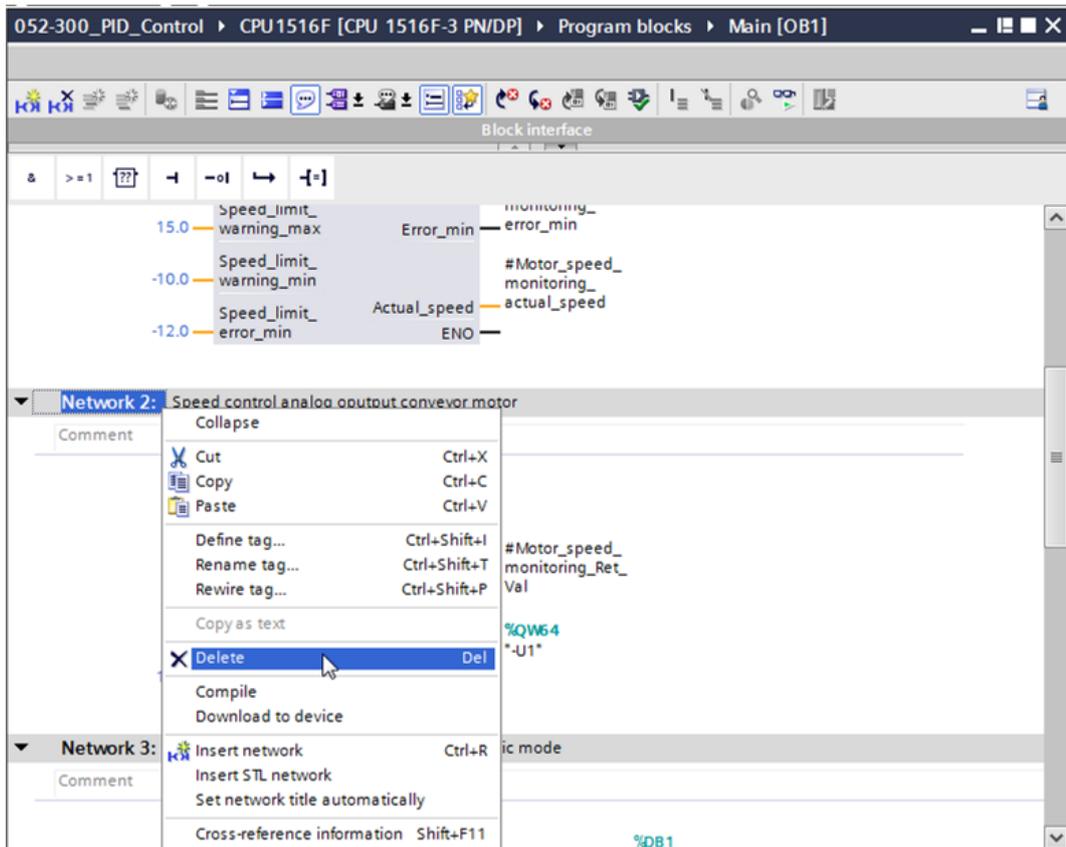
7.2 在周期性中断块中调用控制器 PID_Compact

→ 双击打开组织块“Main”[OB1]。



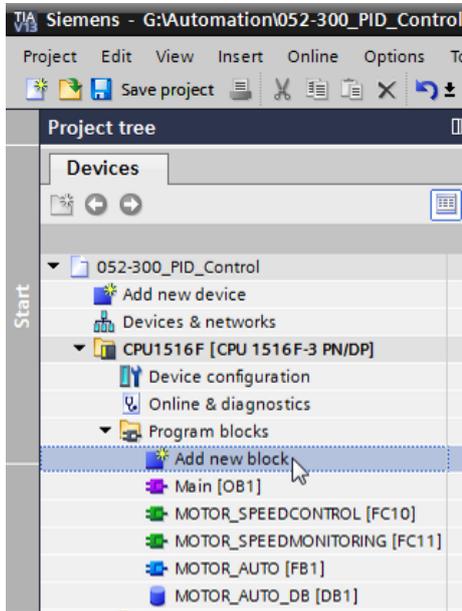
→ 删除 (delete) 网络 2 (Network 2) 和不再需要的功能调用“MOTOR_SPEEDCONTROL”[FC10]。

(→ 网络 2 → 删除)



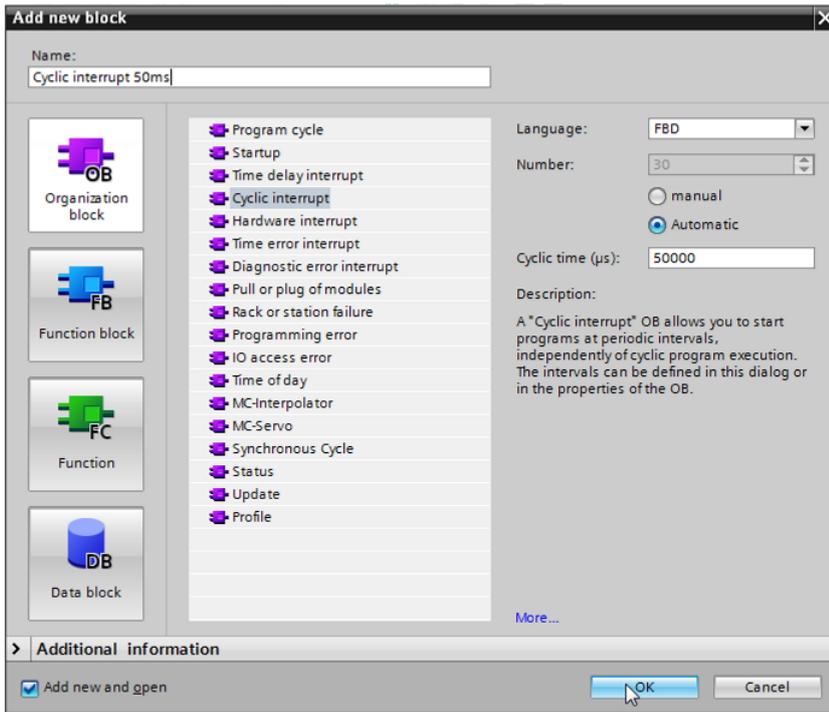
→ 我们需要一个周期性中断块来调用控制器 PID_Compact。为此在程序块 (Program blocks) 文件夹中选择“添加新块”(Add new block)。

(→ 程序块 → 添加新块)



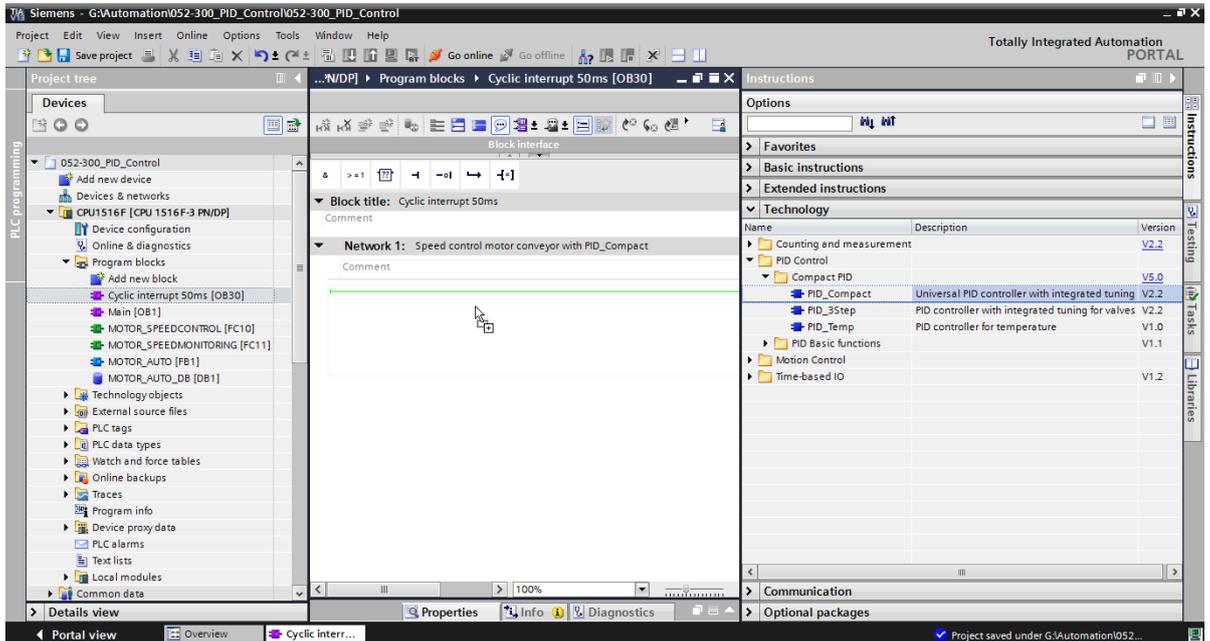
→ 在接下来的对话框中选择  并用以下名称命名周期性中断块: “Cyclic interrupt 50ms”。将语言 (Language) 设置为 FBD 并将时钟脉冲 (Cyclic time) 设置为 50000 μs。勾选“新建并打开”(Add new and open)。现在, 单击“确定”(OK)。

(→  → 名称: Cyclic interrupt 50 ms → 语言: FBD → 时钟脉冲 (μs): 50000 →  新建并打开 → 确定)



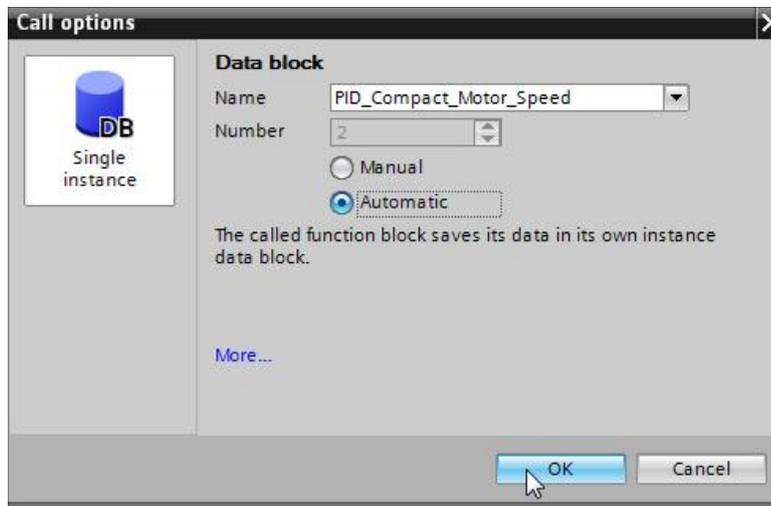
→ 块直接打开。现在输入注释并将工艺对象“PID_Compact”拖进网络 1 (Network 1)。

(→ 工艺 → PID 控制 → Compact PID → PID_Compact)



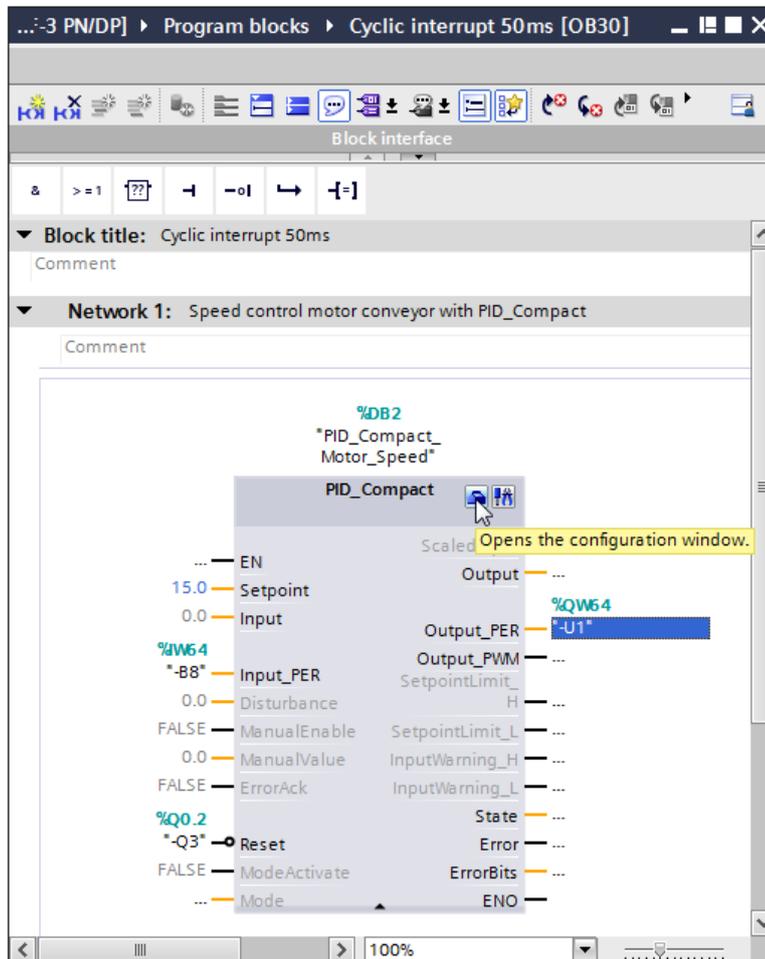
→ 为背景数据块分配一个名称并单击“确定”(OK) 确认接受此命名。

(→ PID_Compact_Motor_Speed → 确定)



→ 通过点击箭头  来扩展模块视图。如图所示, 将该块与额定值 (常数: 15.0)、实际值 (全局变量“-B8”)、调节变量 (全局变量“-U1”) 和用于禁用控制器的反馈输入 (全局变量“-Q3”) 互连。取消输入端 “Reset”。这样便可以打开控制器的配置界面 。

(→  → 15.0 → “-B8” → “-U1” → -Q3 →  → )



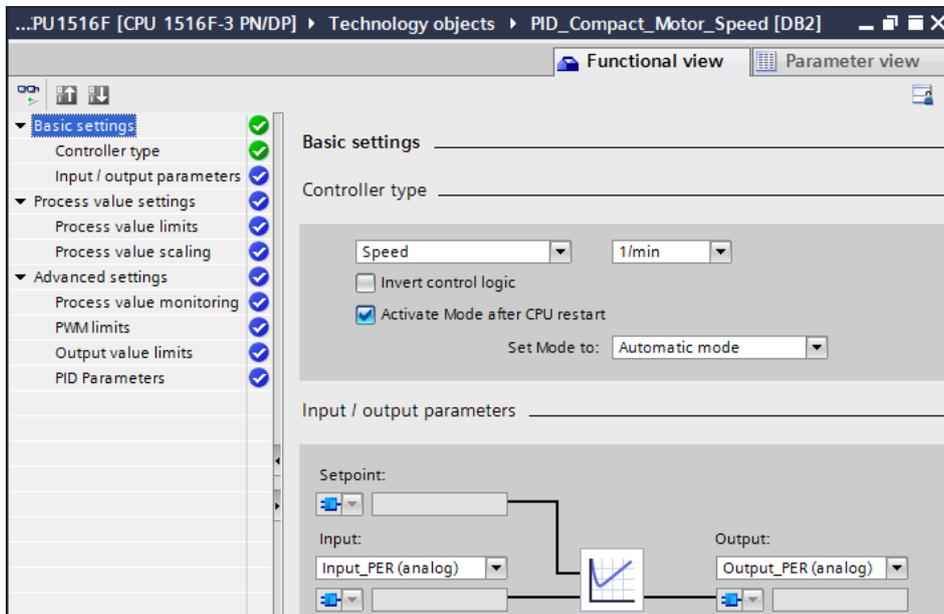
→ 配置控制器时有两种视图：参数视图和功能视图。我们在此使用易于理解的“功能视图”(Functional view)。

(→ 功能视图)

	Name in functional view	Name in DB	Start value project	Minimum value	Maximum value	Comment
Physical quantity	PhysicalQuantity	PhysicalQuantity	Speed	17		Selection of physical quantity.
	Unit of measurement	PhysicalUnit	1/min			Selection of unit of measureme...
PhysicalUnit	PhysicalUnit	PhysicalUnit	0			Selection of unit of measureme...
	..InvertControl	..InvertControl	FALSE			Enables inversion of control logic
RunModeByStartup	RunModeByStartup	RunModeByStartup	TRUE			Activates the operating mode s...
Mode	Mode	Mode	Automatic mode	0	4	Selection of operating mode.
Mode	Mode	Mode	3			Selection of operating mode.

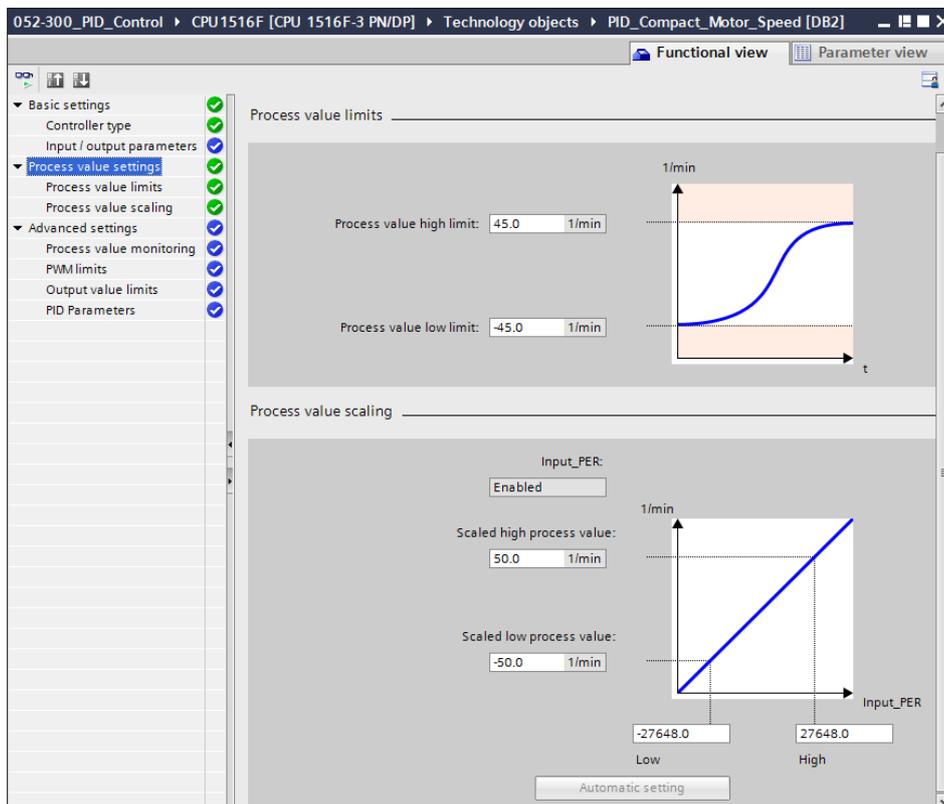
→ 在“基础设置”(Basic settings) 中, 首先确定“控制类型”(Controller type) 和“输入/输出参数”(Input/output parameters) 的互连。如图所示设置值。

(→ 基础设置 → 控制类型 → 输入/输出参数)



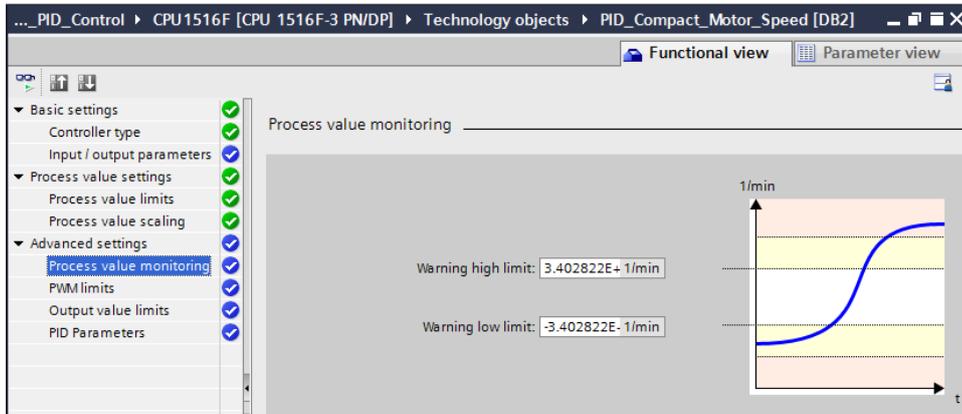
→ 在“实际值设置”(Process value settings) 中将范围缩放为 ± 50 rpm 并将“实际值极限”(Process value limits) 设置为 ± 45 rpm。

(→ 实际值设置 → 实际值极限 → 实际值缩放)



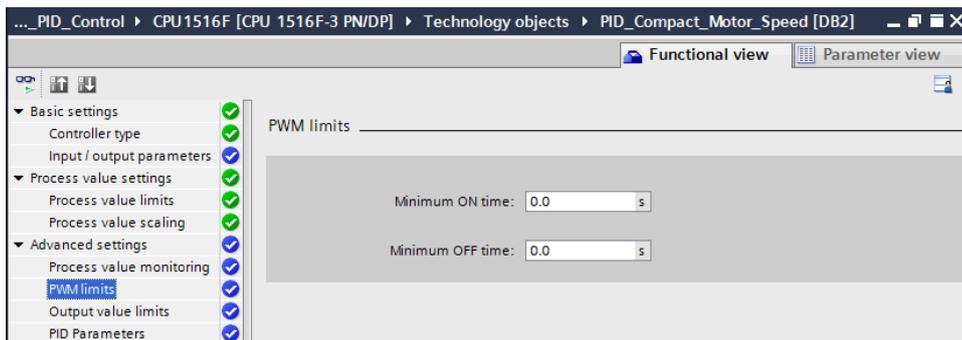
→ 在“高级设置”(Advanced settings) 中可以进行“实际值监控”(process value monitoring), 但这里并不需要。

(→ 高级设置 → 实际值监控)

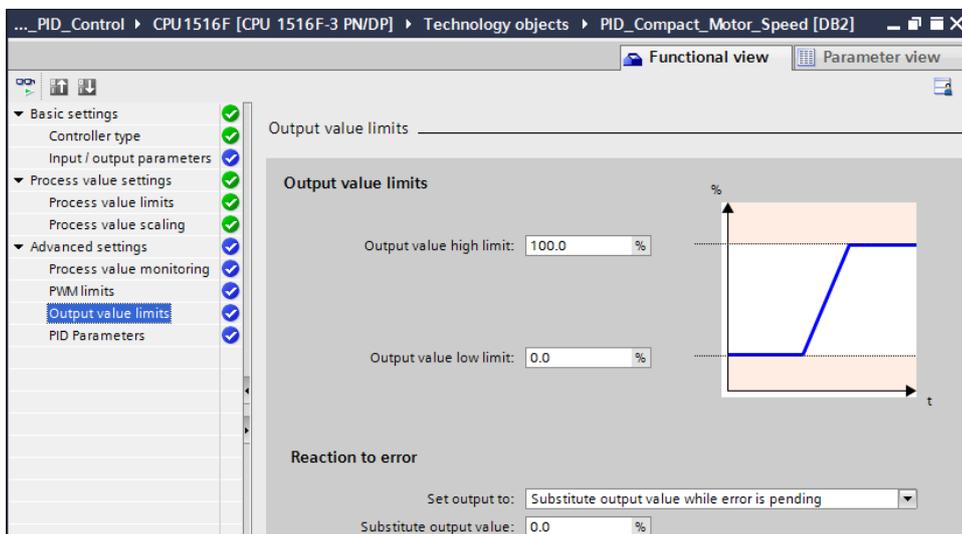


→ 在针对“PWM”（脉冲宽度调制）的“高级设置”中保留默认值，因为我们在该项目中不需要此项输出。

(→ 高级设置 → PWM)

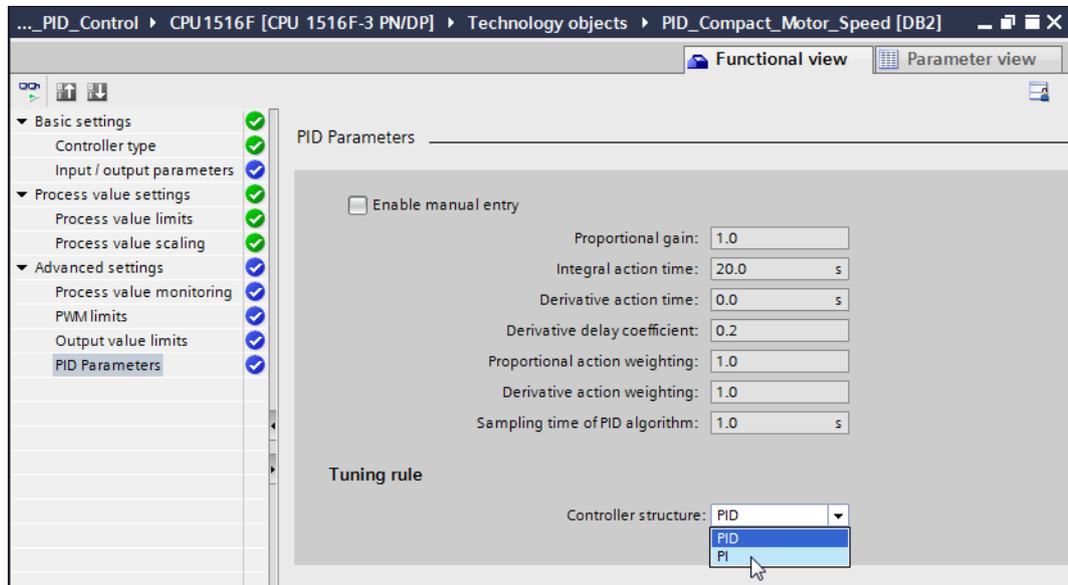


→ 在“高级设置”中在 0.0 % 至 100.0 % 之间定义“输出值极限”(Output value limits)。(→ 高级设置 → 输出值极限)

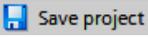


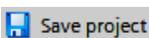
→ “高级设置”中还有针对“PID 参数”(PID parameters) 的手动设置项。我们在此将控制器结构切换为 'PI' 后, 点击  关闭配置窗口并得到一个具有高效 PID 控制器的完备程序。但这依然需要线上运行并进行整定。

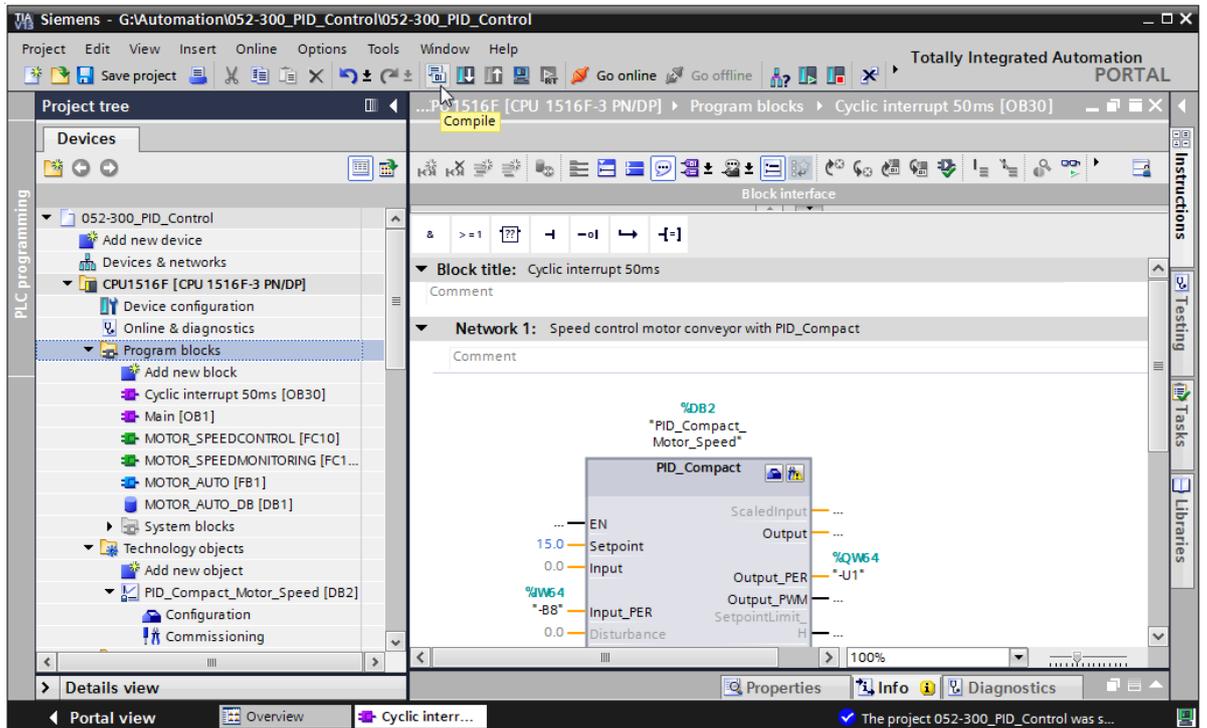
(→ 高级设置 → PID 参数 → 控制器结构: PI → )



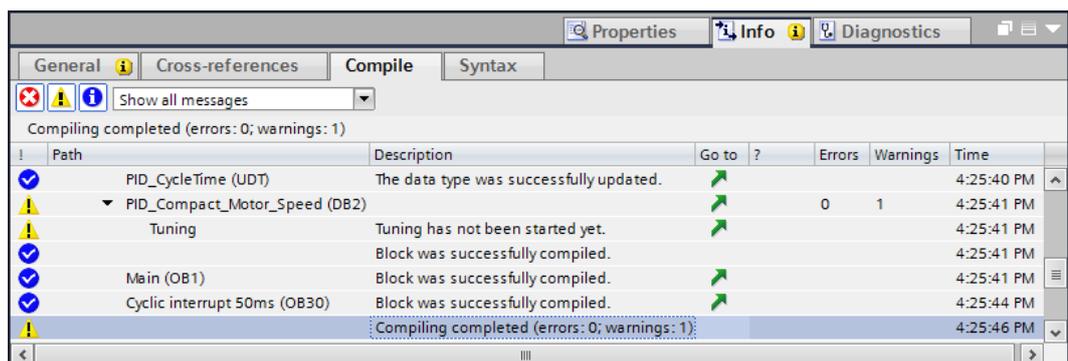
7.3 保存程序并编译

→ 如需保存项目请点击菜单里的  Save project 按钮。如需编译全部块，则需点击文件夹“程序块”(Program blocks) 并选择菜单里的编译符号 。

(→  Save project → 程序块 → )

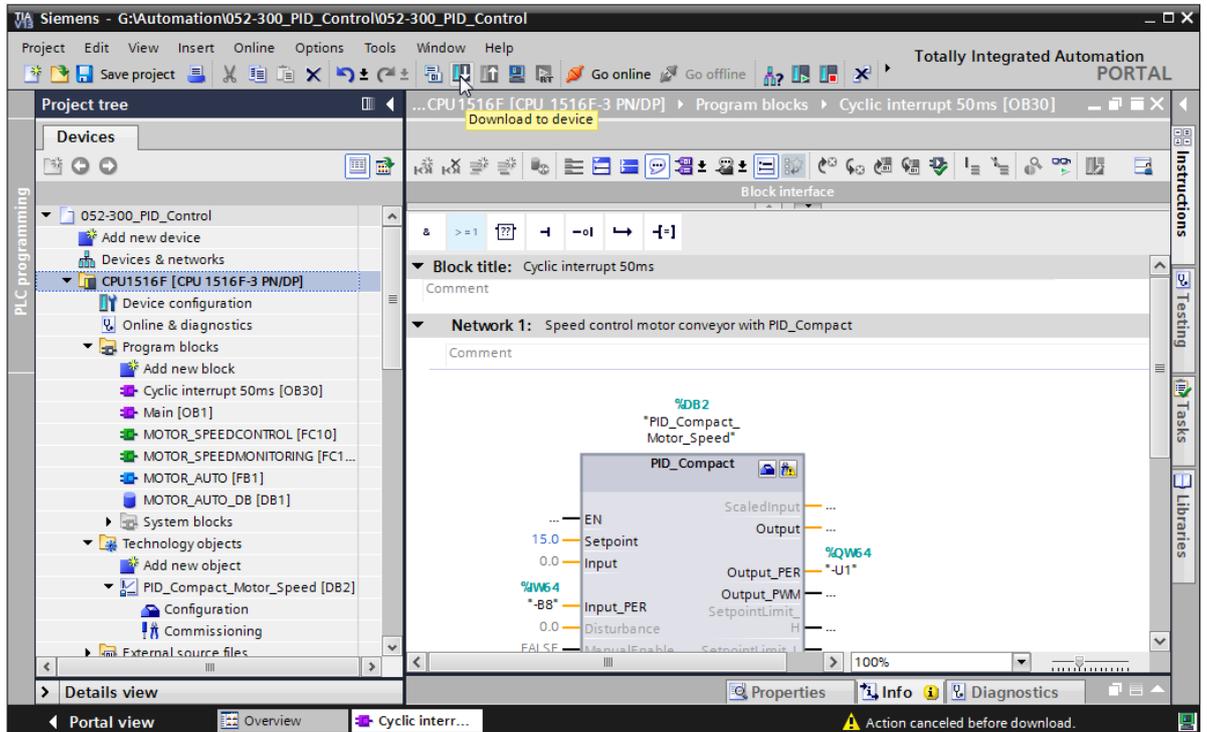


→ 随后会在“信息”(Info) -“编译”(Compile) 区域中显示已成功完成编译的块。



7.4 加载程序

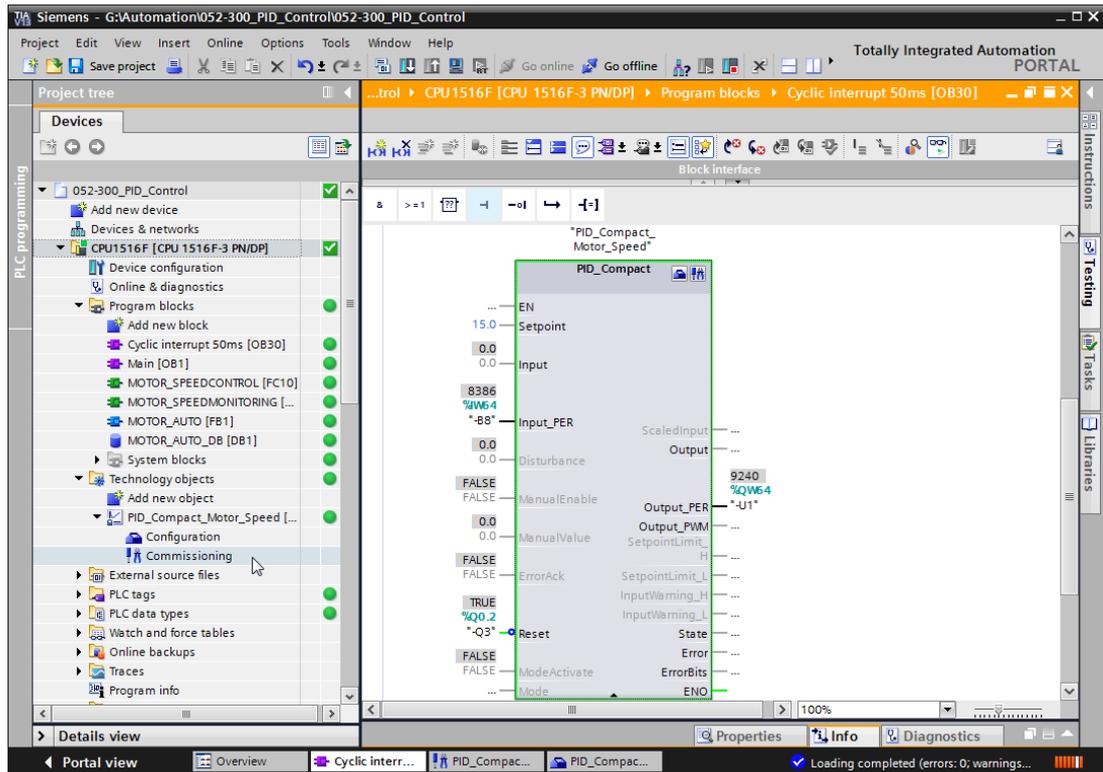
→ 成功完成编译后, 整个控制器将加载所创建的程序及硬件配置, 如前面的课程单元所述。



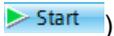
7.5 观测 PID_Compact

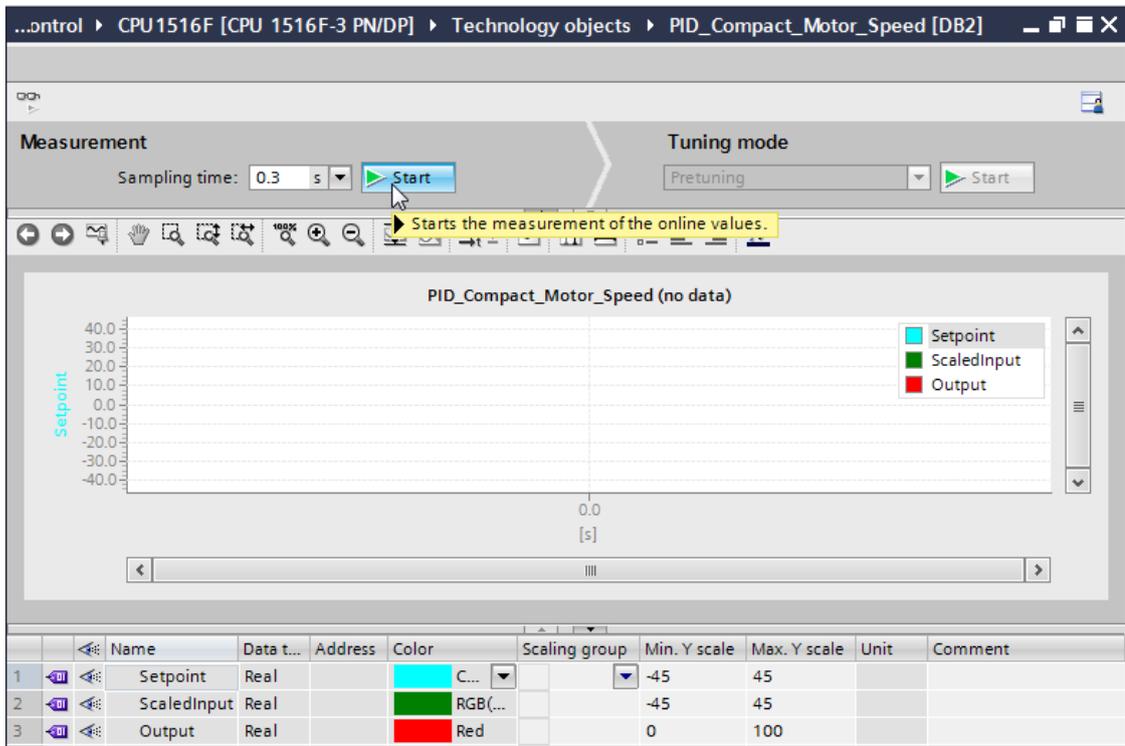
→ 用鼠标点击开启/关闭观测的符号 ，即可在测试程序时观测块和变量的状态。第一次启动 CPU 时，控制器“PID_Compact”尚未整定。为此必须使用鼠标点击符号  Commissioning 来启动整定。

(→ Cyclic interrupt 50ms [OB30] →  → PID_Compact →  Commissioning)

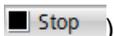


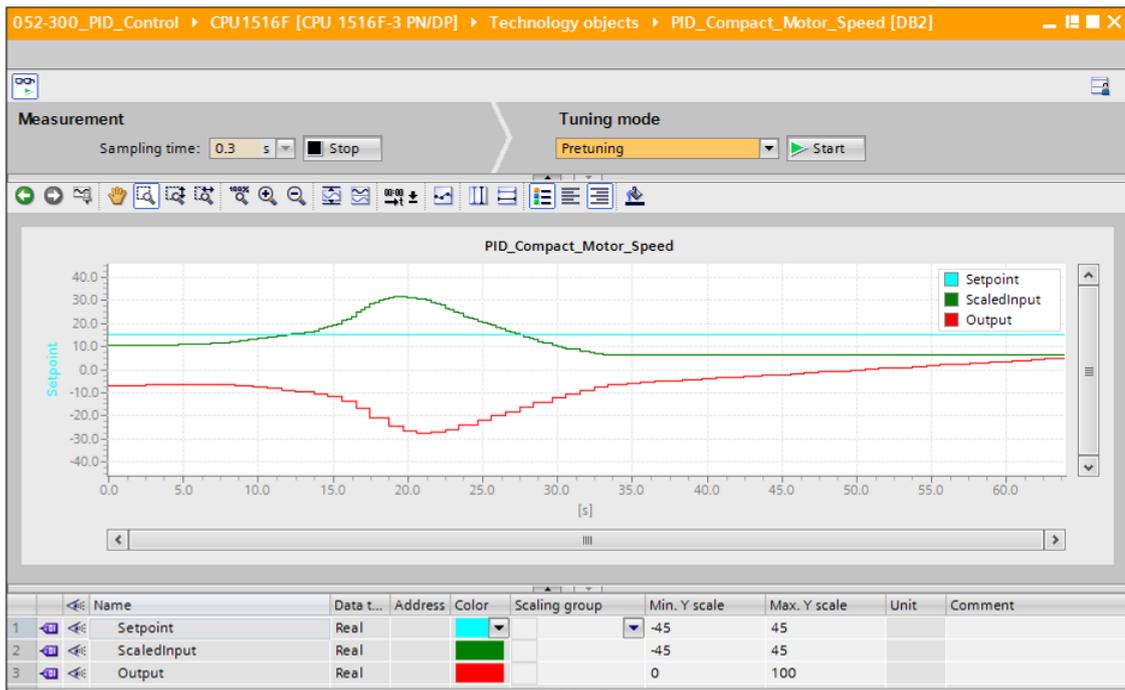
→ 在“测量”(Measurement) 中点击 ，即可在图表中显示额定值 (Setpoint)、实际值 (Scaledinput) 和调节变量 (Output) 并进行观测。

(→ )



→ 单击  中止测量。

(→ )



7.6 PID_Compact 预整定

预整定测得对输出值变化的过程响应并寻找拐点。从受控系统的最大坡度和死区时间中计算出 PID 参数。当执行了预整定和精整定后, 就可得到最佳的 PID 参数。

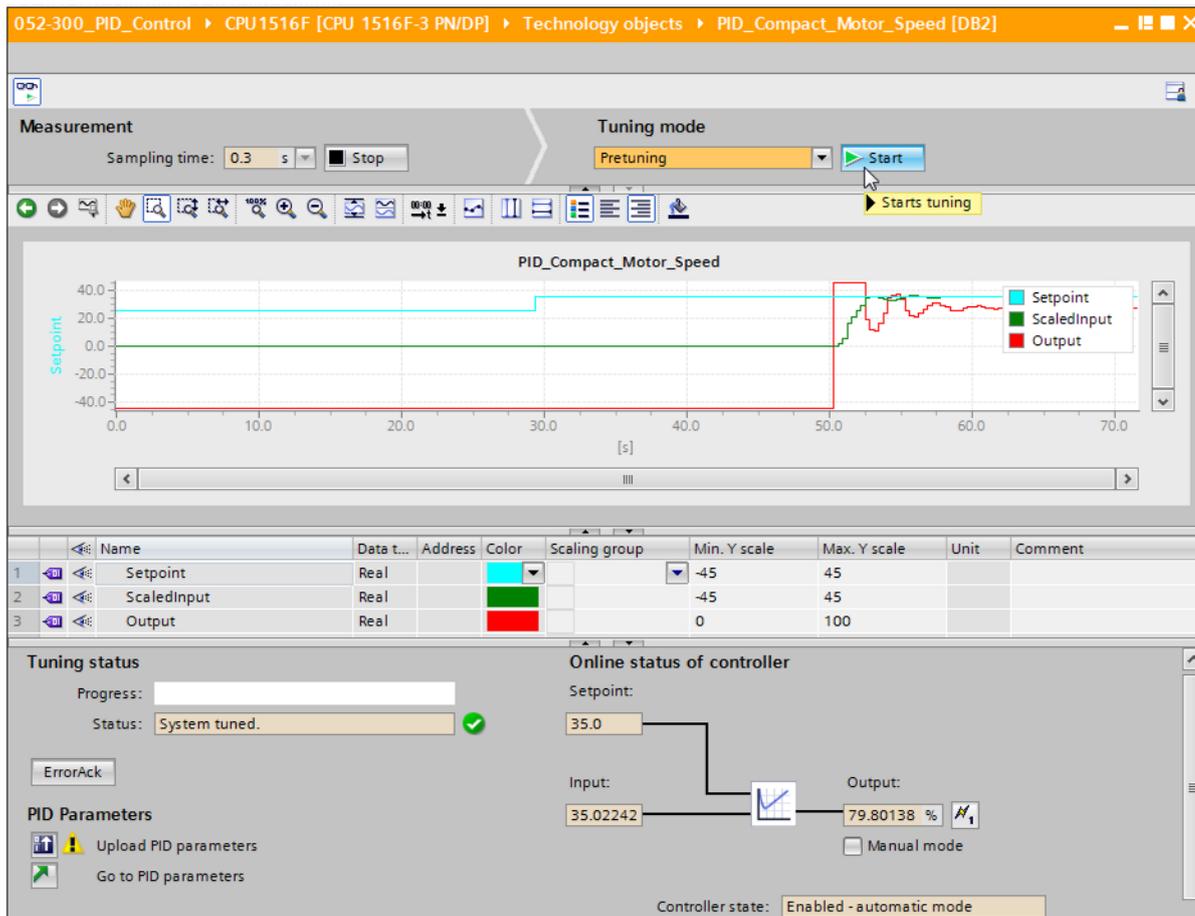
实际值越稳定, 就越易于获取准确的 PID 参数。只要实际值的上升幅度明显大于干扰量, 这种干扰量就是可以接受的。这种现象最容易出现在“未激活”或“手动模式”中。在重新计算 PID 参数前要先进行备份。

必须满足以下前提条件:

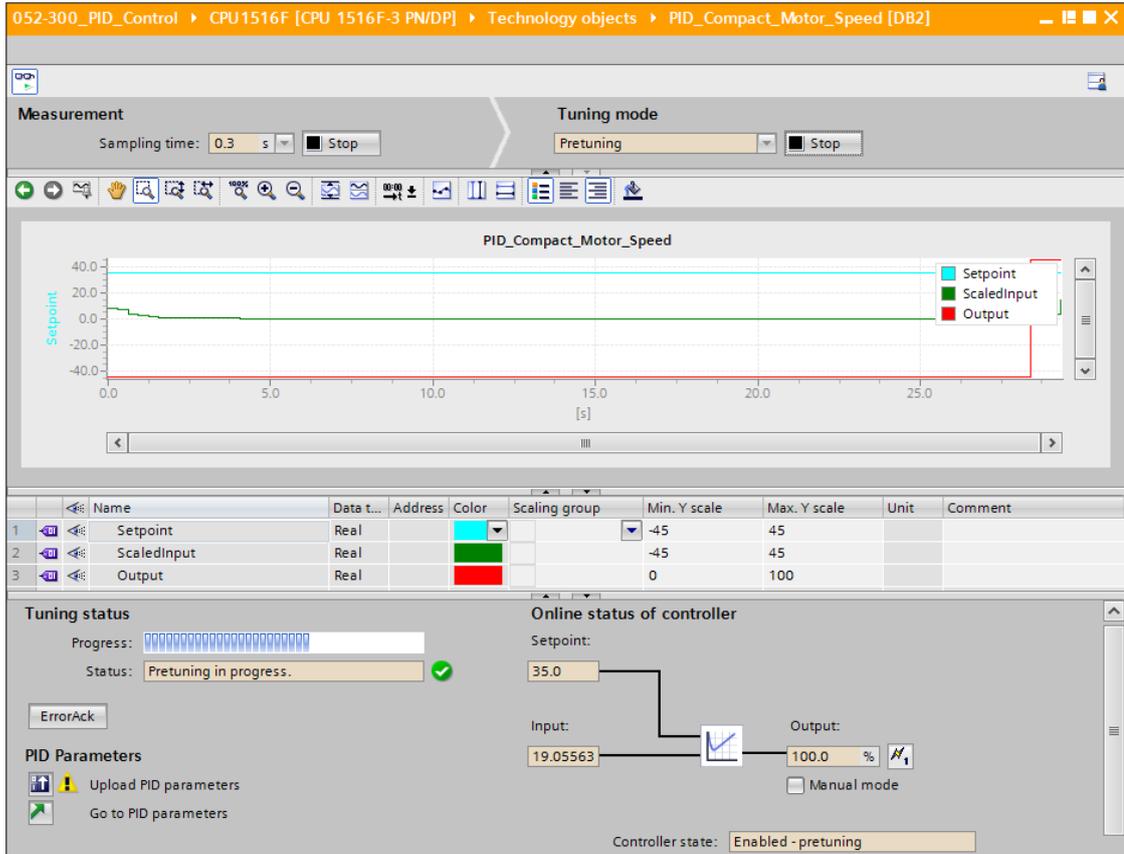
- 在周期性中断块中调用“PID_Compact”指令。
- ManualEnable = FALSE
- Reset = FALSE
- PID_Compact 处于“手动模式”、“未激活”或“自动模式”中。
- 额定值和实际值未超出配置的极限 (参见“实际值监控”(process value monitoring) 配置)。
- 额定值和实际值之间的差值大于实际值上下限差值的 30 %。
- 额定值和实际值之间的间距大于额定值的 50 %。

→ 在“整定方式”(Tuning mode) 中选择“预整定”(Pretuning) 并启动 (Start)。

(→ 整定方式 → 预整定 → )



→ 预整定现在开始。在“整定状态”(Tuning status) 区中显示当前的工作步骤和出现的错误。进度条表示当前工作步骤的进度。



7.7 PID_Compact 精整定

精整定对实际值产生一种恒定的、有限的振荡。通过这种振荡的振幅和频率,针对工作点对 PID 参数进行整定。所有 PID 参数根据结果重新计算。精整定得出的 PID 参数通常要比预整定得出的 PID 参数具有更好的命令性能和抗干扰性。当执行了预整定和精整定后,就可得到最佳的 PID 参数。

PID_Compact 自动尝试产生一种比实际值干扰量更大的振荡。精整定几乎不受实际值稳定性的影响。在重新计算 PID 参数前要先进行备份。

必须满足以下前提条件:

- 在周期性中断块中调用“PID_Compact”指令。
- ManualEnable = FALSE
- Reset = FALSE
- 额定值和实际值未超出配置的极限。
- 闭环控制回路在工作点上取得稳定。当实际值与额定值相同时,已达到工作点。
- 按照预期将不会出现故障。
- PID_Compact 处于“手动模式”、“未激活”或“自动模式”中。

在自动模式中启动时,精整定如下进行:

若要通过整定改善现有 PID 参数,在自动模式下启动精整定。

在闭环控制回路稳定且满足精整定的前提条件之前,PID_Compact 一直使用现有 PID 参数进行控制。然后开始精整定。

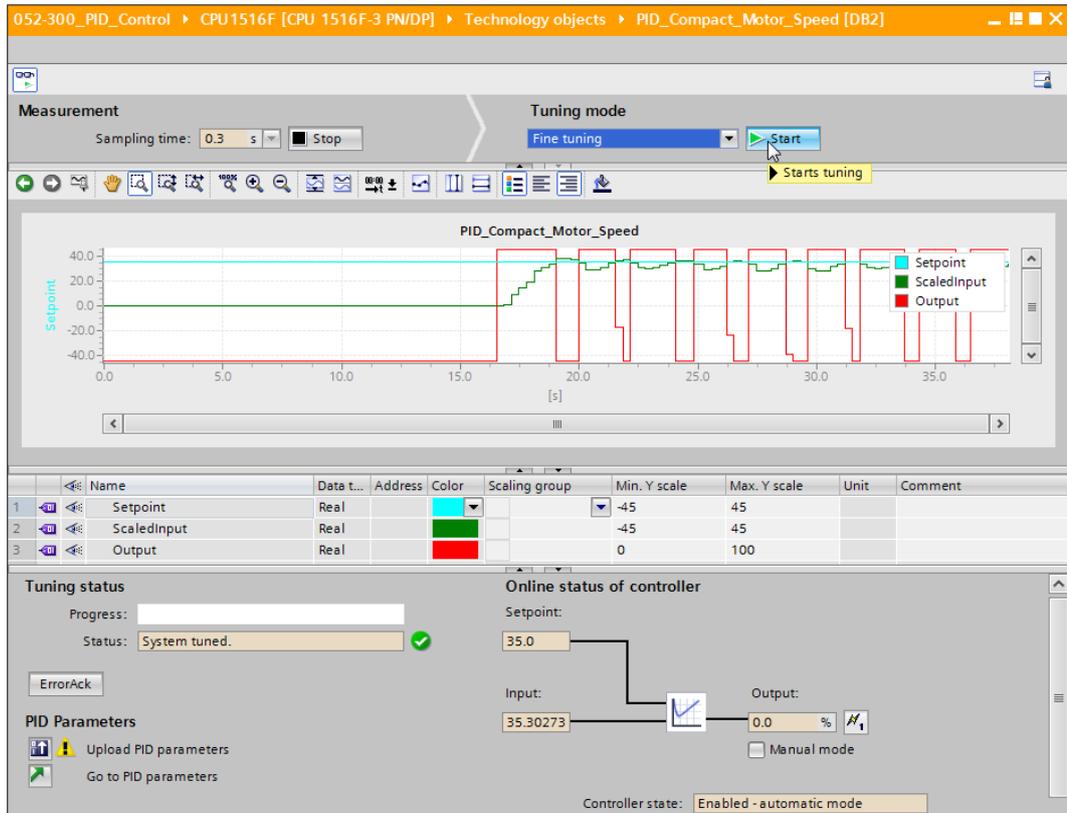
在未激活或手动模式中启动时,精整定如下进行:

满足预整定的前提条件后将开始预整定。在闭环控制回路稳定并满足精整定的前提条件之前,一直使用获取的 PID 参数进行控制。然后开始精整定。如果不能进行预整定,PID_Compact 的行为与故障情况下配置的行为相同。

如果实际值对于预整定来说过于接近额定值,则尝试用最小或最大输出值来达到额定值。这会造成过调升高。

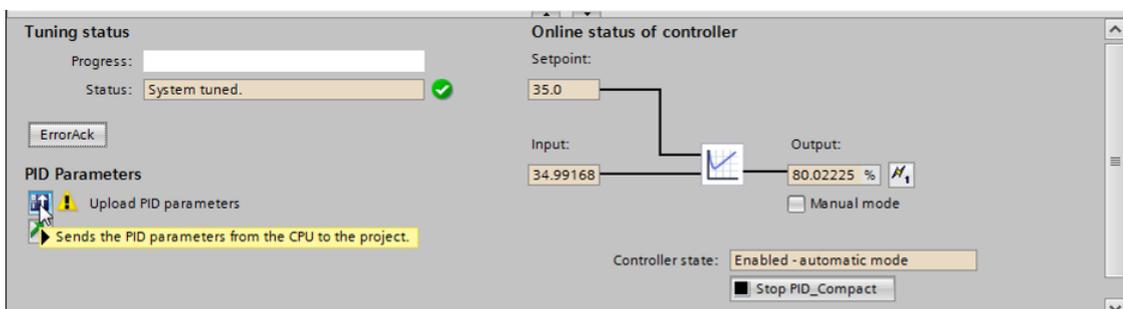
→ 在“整定方式”(Tuning mode) 中选择“精整定”(Fine Tuning) 并启动 (Start)。

(→ 整定方式 → 精整定 → )



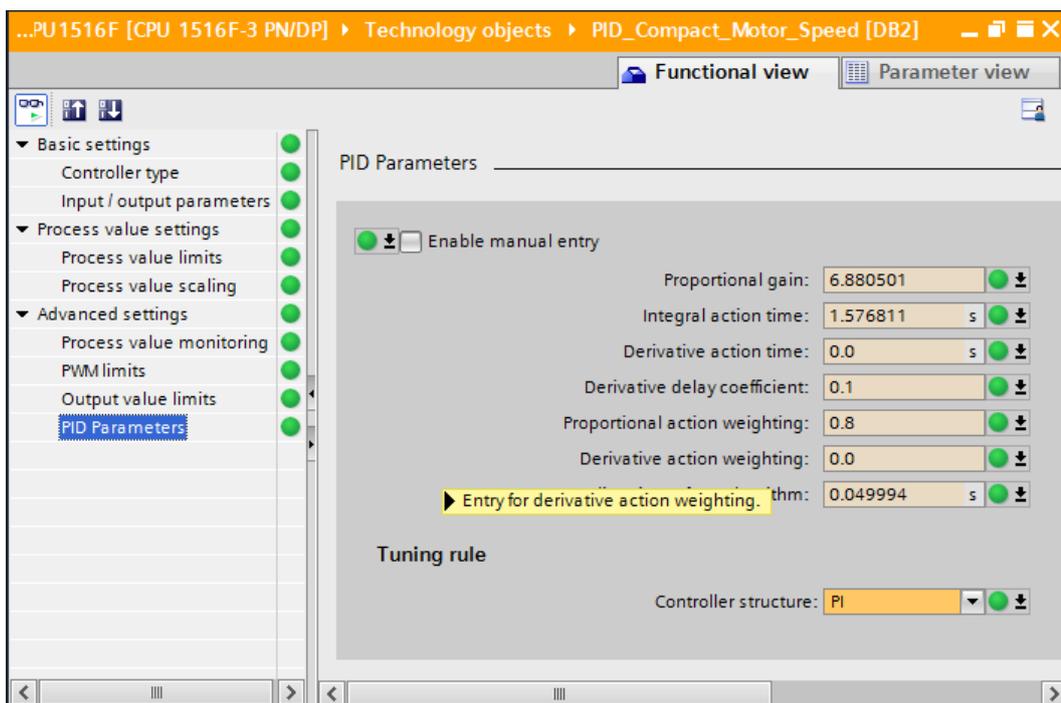
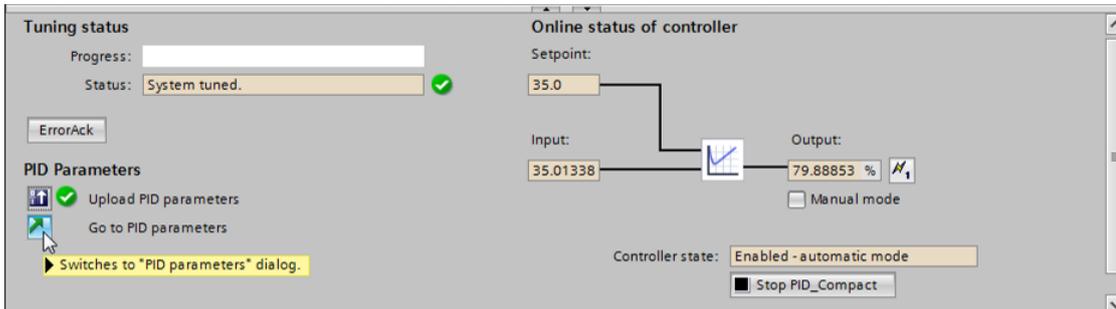
→ 精整定现在开始。在“整定状态”(Tuning status) 区中显示当前的工作步骤和出现的错误。如果自整定成功完成，没有发出错误报警，则说明 PID 参数整定完毕。PID 控制器切换到自动模式并使用整定后的参数。开机和重启 CPU 时，整定后的 PID 参数保留。使用  按钮可以将 CPU 的 PID 参数加载到您的项目中。

(→ )



→ 点击 , 可查看配置中的 PID 参数。

(→ )

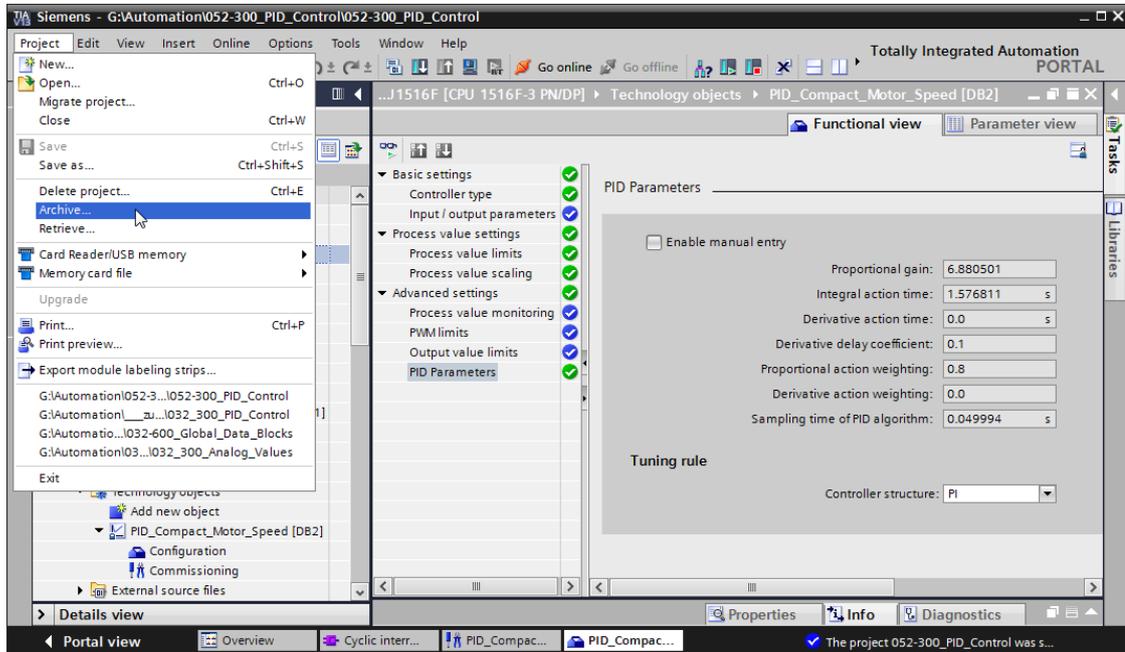


→ 结束时离线 (Go offline) 并备份整个项目 (Save project)。

(→  Go offline →  Save project)

7.8 项目归档

- 最后我们需要将整个项目归档。请选择菜单项 →“项目”(Project) 下的→“归档 ...”(Archive ...)。选择项目归档的文件夹，并以“TIA Portal 项目压缩文件包”的文件类型来保存项目。
(→ 项目 → 归档 → TIA Portal 项目压缩文件包 → 052-300_PID_Controller.... → 保存)



8 检查清单

编号	说明	已检查
1	已成功创建周期性中断块 Cyclic interrupt 50ms [OB30]。	
2	已在周期性中断块 Cyclic interrupt 50ms [OB30] 中调用 PID_Compact 控制器并互连。	
3	已对 PID_Compact 控制器进行了配置。	
4	编译成功完成且没有出现错误报警	
5	加载成功完成且没有出现错误报警	
6	预整定成功完成且没有出现错误报警	
7	精整定成功完成且没有出现错误报警	
8	接通机组 (-K0 = 1) 气缸已驶入/反馈已激活 (-B1 = 1) 紧急停机 (-A1 = 1) 未激活 自动模式 (-S0 = 1) 未按下自动模式停止按钮 (-S2 = 1) 短暂按下自动模式启动按钮 (-S1 = 1) “滑道已占用”传感器已激活 (-B4 = 1) 之后接通输送带电机 M1 可变转速 (-Q3 = 1) 并保持激活状态。 转速在转速额定值 +/- 50 rpm 的范围内	
9	输送带末端传感器已激活 (-B7 = 1) → -Q3 = 0 (2 秒后)	
10	短暂按下自动模式停止按钮 (-S2 = 0) → -Q3 = 0	
11	激活紧急停机 (-A1 = 0) → -Q3 = 0	
12	手动模式 (-S0 = 0) → -Q3 = 0	
13	关闭机组 (-K0 = 0) → -Q3 = 0	
14	柱体未驶入 (-B1 = 0) → -Q3 = 0	
15	转速 > 故障转速最大极限值 → -Q3 = 0	
16	转速 < 故障转速最小极限值 → -Q3 = 0	
17	项目成功归档	

9 更多相关信息

为帮助您进行入门学习或深化学习, 您可以找到更多指导信息作为辅助学习手段, 例如: 入门指南、视频、辅导材料、APP、手册、编程指南及试用版软件/固件, 请单击链接获取相关资料:

www.siemens.com/sce/s7-1500