SIEMENS

SIMATIC

S7-1200, S7-1500

PID 控制

功能手册

版本 09/2016

siemens.com

SIEMENS

SIMATIC

S7-1200, S7-1500
PID 控制

功能手册

09/2016

A5E35300232-AB

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

⚠危险
表示如果不采取相应的小心措施，**将会**导致死亡或者严重的人身伤害。

⚠警告
表示如果不采取相应的小心措施，**可能**导致死亡或者严重的人身伤害。

⚠小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

注意
表示如果不采取相应的小心措施，可能导致财产损失。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。 由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

⚠警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号 ® 的都是西门子股份有限公司的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
德国

A5E35300232-AB
℗ 09/2016 本公司保留更改的权利

前言

本文档用途

本文档可为用户组态和编程 S7-1200 和 S7-1500 自动化系统的控制任务提供支持。

所需基本知识

理解本文档中的内容，需要具备以下知识：

- 自动化技术的基本知识
- SIMATIC 工业自动化系统知识
- 熟练使用 STEP 7 (TIA Portal)

文档的有效性

本文档涉及的软件控制器适用于自动化系统 S7-1200 和 S7-1500 的 CPU 与 STEP 7 (TIA Portal) 搭配使用的情况。 本文档中未涉及的其它 SW 控制器适用于 S7-300 和 S7-400 与 STEP 7 (TIA Portal) 搭配使用的情况。 软件控制器概述 (页 41) 部分完整概述了 STEP 7 (TIA Portal) 中的所有软件控制器及其可能应用。

约定

请遵循下面所标注的注意事项：

说明

这些注意事项包含有关本文档所述的产品、使用该产品或应特别关注的文档部分的重要信息。

其它帮助

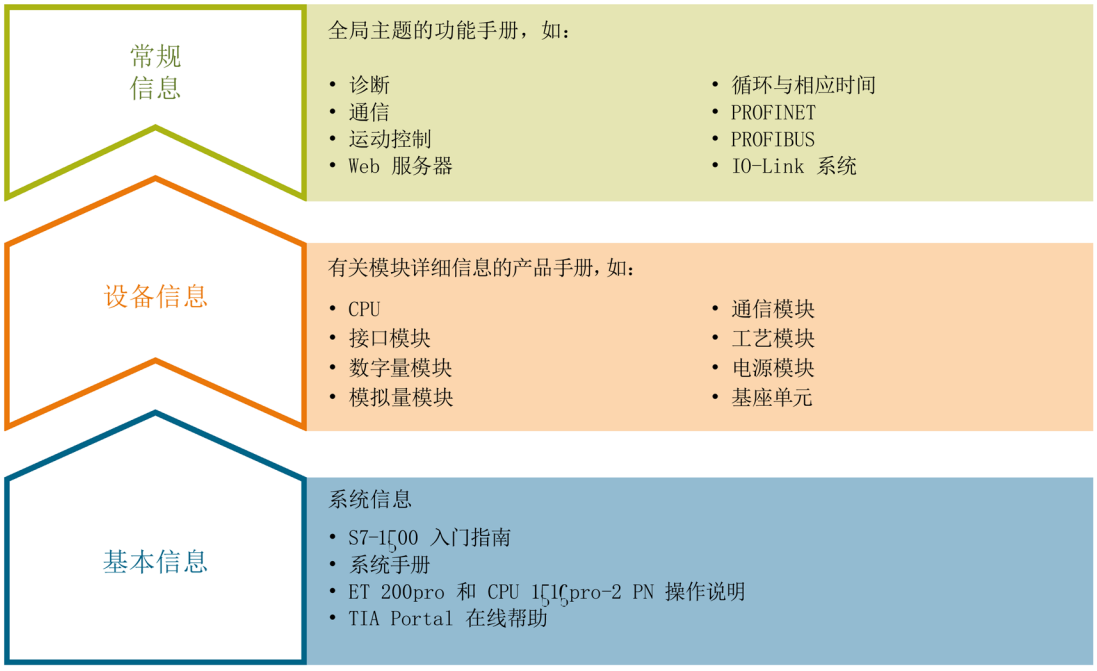
- 有关西门子技术支持方面的信息，清参见附录“服务与支持 (页 573)”。
- 关于各种 SIMATIC 产品与自动化系统的技术文档范围，请访问 Internet (http://www.siemens.com/simatic-tech-doku-portal)。
- Internet (http://mall.automation.siemens.com) 上还提供了在线目录和在线订购系统。

目录

文档指南

1

SIMATIC S7-1500 自动化系统、基于 SIMATIC S7-1500 的 CPU 1516pro-2 PN 和分布式 I/O 系统 SIMATIC ET 200MP、ET 200SP 与 ET 200AL 的文档分为 3 个部分。
这样，用户可以根据具体需求快速访问自己所需的特定信息。



基本信息

在系统手册和入门指南中，对 SIMATIC S7-1500、ET 200MP、ET 200SP 和 ET 200AL 系统的组态、安装、接线和调试进行了详细介绍。对于 CPU 1516pro-2 PN，可参见相应的操作说明。STEP 7 在线帮助则为用户提供有关组态和编程方面的技术支持。

设备信息

产品手册中包含模块特定信息的简洁描述，如特性、端子图、功能特性、技术数据。

常规信息

功能手册中包含有关常规主题的详细介绍，如诊断、通信、运动控制、Web 服务器、OPC UA 等等。

相关文档，可从 Internet (http://w3.siemens.com/mcms/industrial-automation-systems-simatic/en/manual-overview/Pages/Default.aspx) 免费下载。

产品信息数据表中记录了对这些手册的更改和补充。

有关产品信息，敬请访问 Internet：

- S7-1500/ET 200MP (https://support.industry.siemens.com/cs/cn/zh/view/68052815)
- ET 200SP (https://support.industry.siemens.com/cs/cn/zh/view/73021864)
- ET 200AL (https://support.industry.siemens.com/cs/cn/zh/view/99494757)

手册集

手册集中包含系统的完整文档，这些文档收集在一个文件中。

可以在 Internet 上找到手册集：

- S7-1500/ET 200MP (https://support.industry.siemens.com/cs/cn/zh/view/86140384)
- ET 200SP (https://support.industry.siemens.com/cs/cn/zh/view/84133942)
- ET 200AL (https://support.industry.siemens.com/cs/cn/zh/view/95242965)

“我的技术支持”

通过“我的技术支持”（我的个人工作区），“工业在线技术支持”的应用将更为方便快捷。

在“我的技术支持”中，用户可以保存过滤器、收藏夹和标签，请求 CAx 数据以及编译“文档”区内的个人数据库。此外，支持申请页面还支持用户资料自动填写。用户可随时查看当前的所申请的支持请求。

要使用“我的技术支持”中的所有功能，必须先进行注册。

有关“我的技术支持”，敬请访问 Internet (https://support.industry.siemens.com/My/ww/zh)。

“我的技术支持” - 文档

在“我的技术支持”中的“文档”区域，用户可以使用整个手册或部分手册生成自己的手册。也可以将手册导出为 PDF 文件或后期可编辑的其它格式。

有关“我的技术支持” - 文档，敬请访问 Internet (http://support.industry.siemens.com/My/ww/zh/documentation)。

“我的技术支持” - CAx 数据

在“我的技术支持”中的 CAx 数据区域，可以访问 CAx 或 CAe 系统的最新产品数据。

仅需轻击几次，用户即可组态自己的下载包。

在此，用户可选择：

- 产品图片、二维码、3D 模型、内部电路图、EPLAN 宏文件
- 手册、功能特性、操作手册、证书
- 产品主数据

有关“我的技术支持” - CAx 数据，敬请访问 Internet (http://support.industry.siemens.com/my/ww/zh/CAxOnline)。

应用示例

应用示例中包含有各种工具的技术支持和各种自动化任务应用示例。自动化系统中的多个组件完美协作，可组合成各种不同的解决方案，用户无需再关注各个单独的产品。

有关应用示例，敬请访问 Internet (https://support.industry.siemens.com/sc/ww/zh/sc/2054)。

TIA Selection Tool

通过 TIA Selection Tool，用户可选择、组态和订购全集成自动化 (TIA) 中所需设备。
该工具作为 SIMATIC Selection Tool 的新一代产品，在一个工具中完美集成了自动化技术的各种组态程序。
通过 TIA Selection Tool，用户可以根据产品选择或产品组态生成一个完整的订购列表。

TIA Selection Tool 可从 Internet (http://w3.siemens.com/mcms/topics/en/simatic/tia-selection-tool) 上下载。

SIMATIC Automation Tool

通过 SIMATIC Automation Tool，可同时对不同的 SIMATIC S7 站进行系统调试和维护操作，而无需打开 TIA Portal 系统。

SIMATIC Automation Tool 支持以下各种功能：

- 扫描 PROFINET/以太网工厂网络，识别所有连接的 CPU
- 为 CPU 分配地址（IP、子网、网关）和站名称（PROFINET 设备）
- 将日期和已转换为 UTC 时间的 PG/PC 时间传送到模块中
- 将程序下载到 CPU 中
- 切换操作模式 RUN/STOP
- 通过 LED 指示灯闪烁确定 CPU 状态
- 读取 CPU 错误信息
- 读取 CPU 诊断缓冲区
- 复位为出厂设置
- 更新 CPU 和所连模块的固件版本

SIMATIC Automation Tool 可从 Internet (https://support.industry.siemens.com/cs/cn/zh/view/98161300) 上下载。

PRONETA

SIEMENS PRONETA（PROFINET 网络分析服务）用于在调试过程中快速分析工厂网络的具体状况。PRONETA 具有以下两个核心功能：

- 拓扑总览功能，分别扫描 PROFINET 和连接的所有组件。
- IO 检查，快速测试工厂接线和模块组态。

SIEMENS PRONETA 可从 Internet (https://support.industry.siemens.com/cs/cn/zh/view/67460624) 上下载。

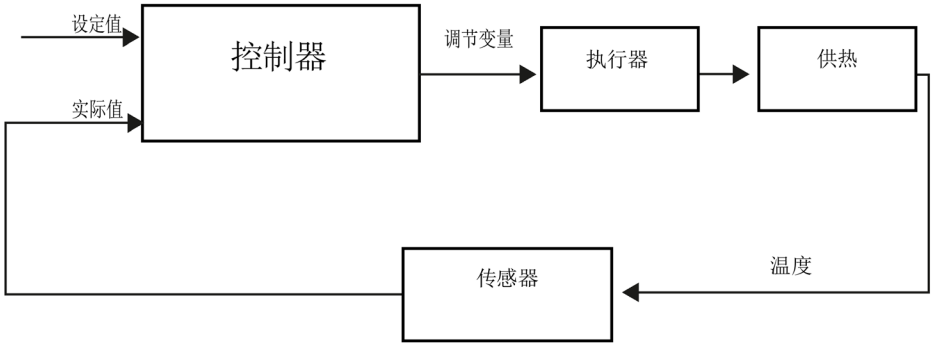
控制原理

2

2.1 受控系统和执行器

受控系统

通过加热系统控制室温是受控系统的一个简单示例。 传感器测量室温并将温度值传送给控制器。 控制器将当前室温与设定值进行比较，并计算加热控制的输出值（调节变量）。



如果 PID 控制器的设置正确，则会尽快达到此设定值，然后使其保持为常数值。 输出值更改后，过程值通常仅随时间延迟而变化。 控制器必须针对此响应进行补偿。

执行器

执行器是受控系统元件，受控制器影响。 其功能是修改质量和能量流。

下表概述了执行器的应用。

应用	执行器
液体或气体质量流	阀门、遮板、闸门阀
固体质量流，如大块材料	铰链式挡板、传送带、振动器通道
电流	开关触点、接触器、继电器、可控硅
	可变电阻、可调变压器、晶体管

执行器分为以下几种：

- 带有恒定起动信号的比例执行器

  这些元件用于设置开启角度、角位置，或与输出值成比例的位置。 输出值在控制范围内会对过程产生模拟量作用。

  此组中的执行器包括弹簧支撑的气动驱动器，以及构成位置控制系统的带位置反馈的电动驱动器。

  连续控制器（如 PID_Compact）会生成输出值。

- 带脉冲宽度调制信号的比例执行器

  这些执行器用于在采样时间间隔内生成长度与输出值成比例的脉冲输出。 执行器（如加热电阻或制冷装置）在等时模式下接通，持续时间根据输出值的不同而有所不同。

  起动信号可呈现单极“打开”或“关闭”状态，或表示双极状态，如“打开/关闭”、“向前/向后”、“加速/制动”。

  输出值由两位控制器（如具有脉宽调制的 PID_Compact）生成。

- 具有积分作用和三位起动信号的执行器

  执行器经常由电机操作，操作周期与阻塞元件的执行器进给成比例。 包括阀门、遮板和闸门阀等元件。 尽管所有这些执行器的设计有所不同，但它们都受到受控系统输入端的积分作用的影响。

  步进控制器（如 PID_3Step）会生成输出值。

2.2 受控系统

受控系统的属性几乎不受到影响，因为这些属性是由过程和机械的技术要求决定的。 只能通过为特定受控系统选择合适的控制器类型以及调整控制器以适应受控系统的时间响应，来实现可接受的控制结果。 因此，要对控制器的比例、积分和微分作用进行组态，很有必要详细了解受控系统的类型和参数。

受控系统类型

根据受控系统对输出值阶跃变化的时间响应来对受控系统进行分类。

受控系统有以下分类：

- 自调节受控系统
  - 比例作用受控系统
  - PT1 受控系统
  - PT2 受控系统
- 非自调节受控系统
- 具有/不具有时间的受控系统

自调节受控系统

比例作用受控系统

在比例作用受控系统中，过程值几乎会立即随输出值而变化。 过程值与输出值之间的比率由受控系统的比例 Gain 定义。

示例：

- 管道系统中的闸门阀
- 分压器
- 液压系统中的降压功能

PT1 受控系统

在 PT1 受控系统中，过程值的变化最初与输出值的变化成比例。 过程值的变化率随时间减小，直至达到最终值，即被延迟。

示例：

- 弹簧减震系统
- RC 元件的充电
- 由蒸汽加热的贮水器。

加热与制冷过程，或充电和放电特性的时间常量通常相同。 时间常量不同时，控制显然会更加复杂。

PT2 受控系统

在 PT2 受控系统中，过程值不会立即跟随输出值的阶跃变化，即，过程值的增加与正向上升率成正比，然后随着上升率的下降而逼近设定值。 受控系统通过二阶延迟元件显示比例响应特性。

示例：

- 压力控制
- 流速控制
- 温度控制

非自调节受控系统

非自调节受控系统具有积分响应。 过程值趋于无限大的值。

示例：

- 流入容器的液体

具有死时间的受控系统

死时间总是表示在系统输出测量系统输入的变化之前到期的运行时间或传输时间。

在具有死时间的受控系统中，过程值的变化将发生延迟，延迟时间等于死时间量。

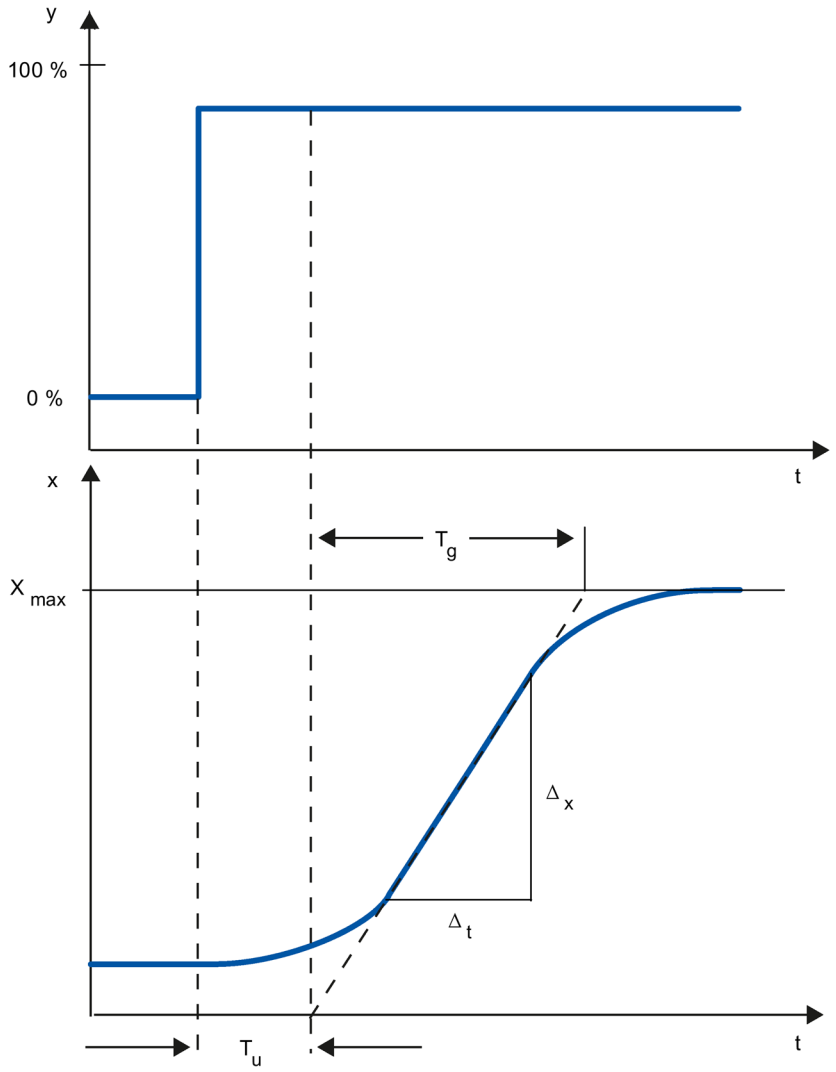
示例：

传送带

2.3 控制部分的特征值

根据阶跃响应确定时间响应

受控系统的时间响应可根据输出值 y 发生阶跃变化之后的过程值 x 的时间特性来确定。大多数受控系统为自调节受控系统。



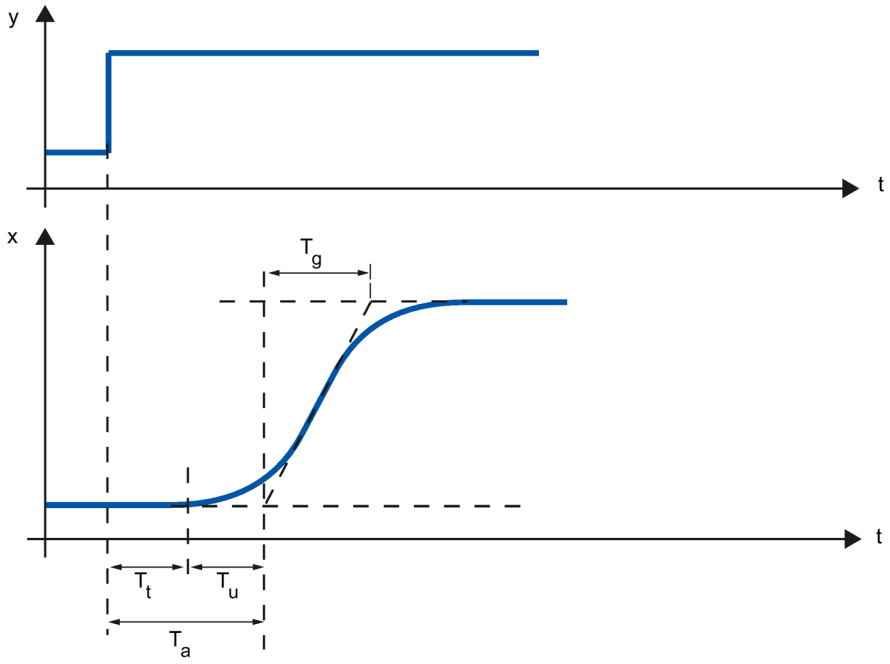
时间响应可由使用变量延迟时间 T_u、恢复时间 T_g 和最大值 X_{max} 来大致确定。 这些变量可通过最大值的切点和阶跃响应的转折点来确定。 在很多情况下，无法记录达到最大值的响应特性，因为过程值不能超过特定值。 在这种情况下，上升率 v_{max} 用于确定受控系统 ($v_{max} = \Delta_x/\Delta_t$)。

受控系统的可控性可根据比率 T_u/T_g 或 $T_u \times v_{max}/X_{max}$ 来估算。 规则：

过程类型	T_u / T_g	受控系统的适控性
I	< 0,1	可以很好地控制
II	0.1 到 0.3	仍可控制
III	> 0,3	难以控制

死时间对受控系统可控性的影响

具有死时间和恢复功能的受控系统对输出值跳变的响应如下所述。



T_t 死时间
T_u 延迟时间
T_g 恢复时间
y 输出值
x 过程值

具有死时间的自调节受控系统的可控性由 T_t 与 T_g 的比率确定。 T_t 必须小于 T_g。 规则：

$T_t/T_g \leq 1$

受控系统的响应率

可以根据以下值来判断受控系统：

T_u < 0.5 min、T_g < 5 min = 快速受控系统

T_u > 0.5 min、T_g > 5 min = 慢速受控系统

特定受控系统的参数

物理量	受控系统	延迟时间 T_u	恢复时间 T_g	上升率 v_{max}
温度	小型电热炉	0.5 到 1 min	5 到 15 min	一直到 60 K/min。
	大型电热退火炉	1 到 5 min	10 分钟到 20 分钟	一直到 20 K/min。
	大型燃气加热退火炉	0.2 到 5 min	3 分钟到 60 分钟	1 到 30 K/min
	蒸馏塔	1 到 7 min	40 到 60 min	0.1 到 0.5° C/s
	高压锅 (2.5 m^3)	0.5 分钟到 0.7 分钟	10 分钟到 20 分钟	未指定
	高压锅	12 分钟到 15 分钟	200 分钟到 300 分钟	未指定
	蒸气过热器	30 s 到 2.5 min	1 到 4 min	2°C/s
	注塑机	0.5 分钟到 3 分钟	3 分钟到 30 分钟	5 到 20 K/min
	挤压机	1 分钟到 6 分钟	5 分钟到 60 分钟	
	包装机	0.5 分钟到 4 分钟	3 分钟到 40 分钟	2 到 35 K/min
	暖气	1 到 5 min	10 到 60 min	1° C/min
流速	气体管道	0 到 5 s	0.2 到 10 s	不相关
	液体管道	无	无	

物理量	受控系统	延迟时间 T_u	恢复时间 T_g	上升率 v_{max}
压力	气体管道	无	0.1 s	不相关
	用煤气或石油作燃料的汽包锅炉	无	150 s	不相关
	采用撞击粉碎机的汽包锅炉	1 到 2 min	2 到 5 min	不相关
容器级别	汽包锅炉	0.6 到 1 min	未指定	0.1 到 0.3 cm/s
速度	小型电动机	无	0.2 到 10 s	不相关
	大型电动机	无	5 到 40 s	不相关
	蒸汽轮机	无	未指定	50 min^{-1}
电压	小型生成器	无	1 到 5 s	不相关
	大型生成器	无	5 到 10 s	不相关

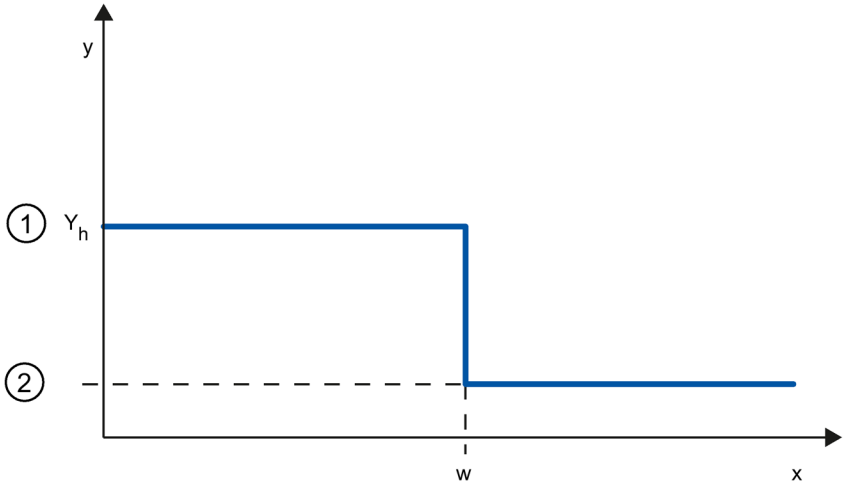
2.4 脉冲控制器

无反馈两位控制器

两位控制器将状态“ON”和“OFF”作为切换函数。 这与 100% 或 0% 输出相对应。 该特性会使过程值 x 在设定值 w 周围持续振荡。

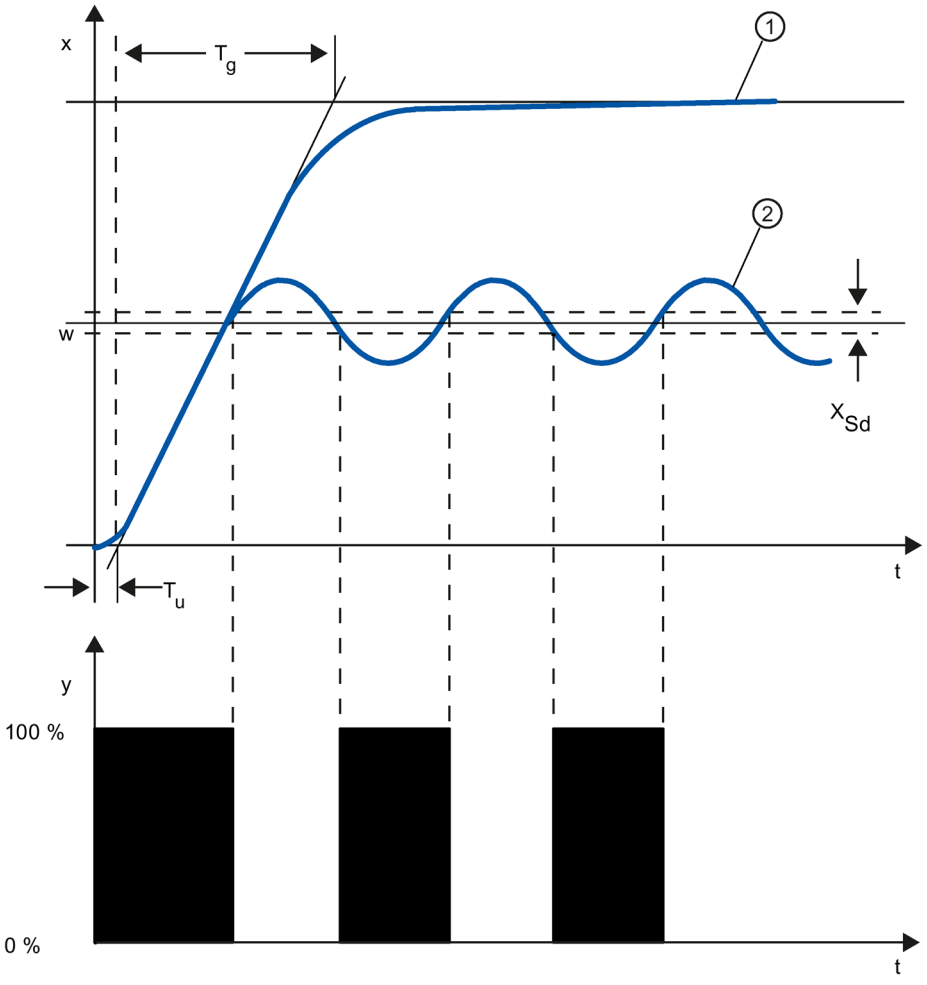
振幅和波动持续时间随受控系统的延迟时间 T_u 与恢复时间 T_g 之间的比例而增加。 这些控制器主要用于简单的温度控制系统（例如直接用电加热的炉子），或用作限值报警设备。

下图显示了两位控制器的特性



① ON
② OFF
Y_h 控制范围
w 设定值

下图显示了两位控制器的控制函数



① 不带控制器时的响应特性
② 带两位控制器时的响应特性
T_u 延迟时间
T_g 恢复时间
X_{Sd} 切换差异

有反馈两位控制器

在受控系统具有较长延迟时间的情况下（例如功能空间与加热空间分离的炉），可通过使用电力反馈改善两位控制器的特性。

反馈用于增加控制器的开关频率，但这会减小过程值的振幅。 此外，在动态操作中可充分改进控制作用结果。 切换频率限制由输出级别决定。 在机械起动器（例如继电器和触点）上，每分钟不得超过 1 到 5 次切换。 如果是下游可控硅或三端双向可控硅控制器的电压和电流输出，则可选择超过受控系统目前限制频率的高切换频率。

因为切换脉冲无法再通过受控系统的输出来确定，所以会得到与连续控制器结果类似的结果。

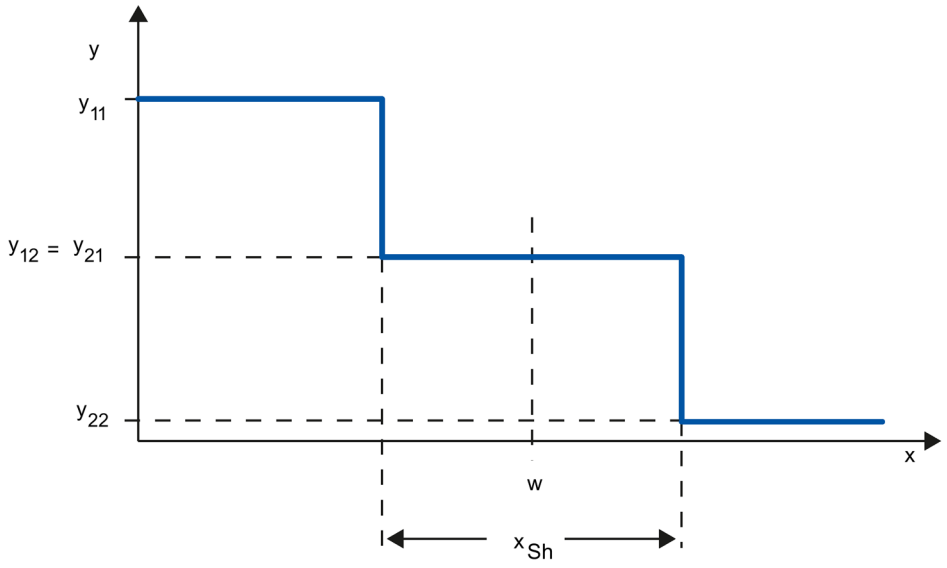
通过对连续控制器的输出值进行脉宽调制来生成输出值。

反馈两位控制器可用于炉子的温度控制，用于塑料、纺织品、纸张、橡胶和食品中使用的加工机，以及用于加热和冷却设备。

三位控制器

三位控制器用于加热/冷却。 这些控制器使用两个切换点作为输出。 控制作用结果可通过电子反馈结构进行优化。 此类控制器的应用领域包括供暖、低温试验箱、气候试验箱和塑料加工机的工具加热设备。

下图显示了三位控制器的特性

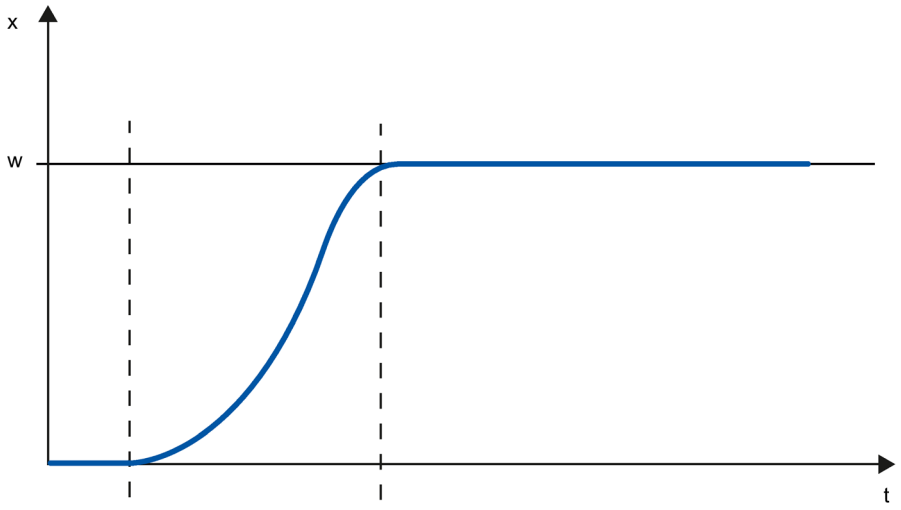


y	输出值，例如 y11 = 100% 加热 y12 = 0% 加热 y21 = 0% 制冷 y22 = 100% 制冷
x	过程值的物理量，例如，以 ° C 为单位的温度
w	设定值
x_{Sh}	切换点 1 与切换点 2 之间的距离

2.5 对设定值变化和干扰的响应

对设定值变化的响应

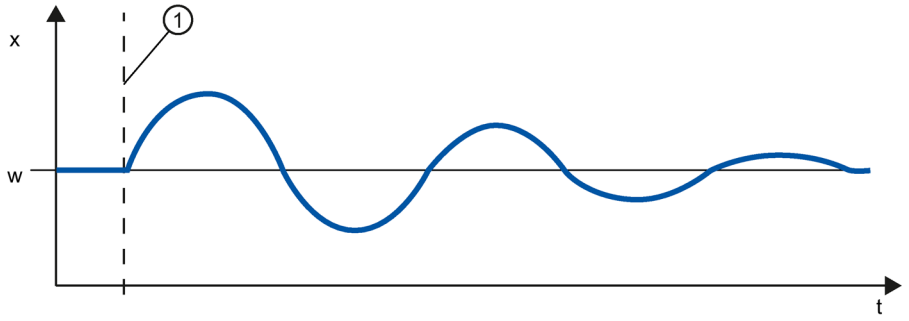
过程值应尽快随设定值而变化。 可通过最大限度地减小过程值的波动以及达到新设定值所需的时间来改进对设定值变化的响应。



x	过程值
w	设定值

对干扰的响应

设定值受干扰变量影响。 控制器必须在尽可能最短的时间内消除所生成的控制偏差。 可通过最大限度地减小过程值的波动以及达到新设定值所需的时间来改进对干扰的响应。



x	过程值
w	设定值
①	影响干扰变量

干扰变量由具有积分作用的控制器进行校正。 持久不变的干扰变量不会降低控制质量，因为控制偏差相对较稳定。 由于控制偏差波动，动态干扰变量会对控制质量产生较显著的影响。 只能通过缓动的积分作用来再次消除控制偏差。

可将可测量的干扰变量包含在受控系统中。 这样会显著提高控制器的响应速度。

2.6 不同反馈结构中的控制响应

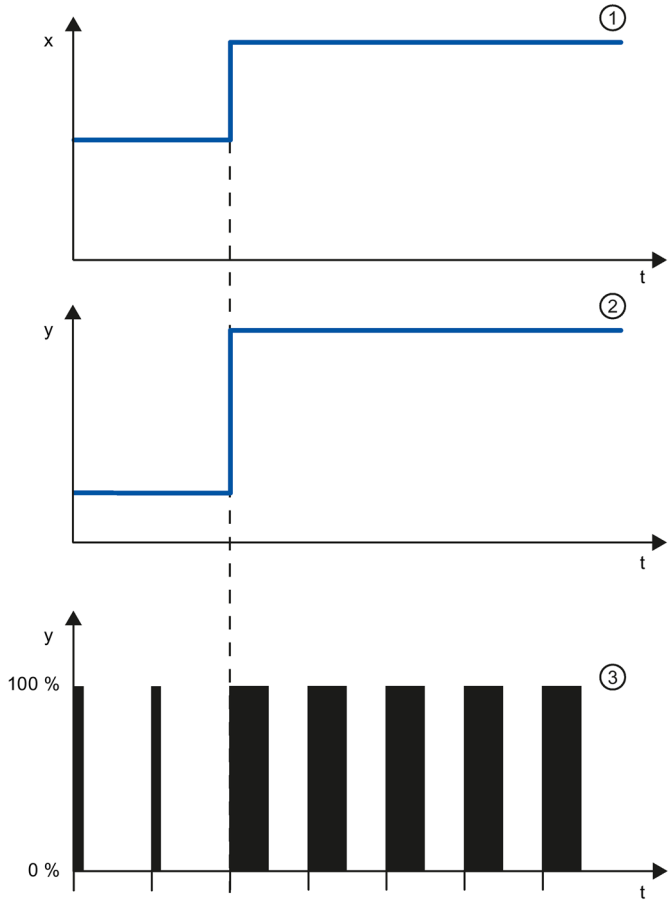
控制器的控制特性

控制器能否准确适应受控系统的时间相应，对于控制器准确稳定在设定值以及对干扰量做出最佳响应起着决定性的作用。

反馈电路可具有比例作用 (P)、比例微分作用 (PD)、比例积分作用 (PI) 或比例积分微分作用 (PID)。

如果阶跃函数由控制偏差触发，则控制器的阶跃响应会因控制器类型而异。

比例作用控制器的阶跃响应



① 控制偏差
② 连续控制器的输出值
③ 脉冲控制器的输出值

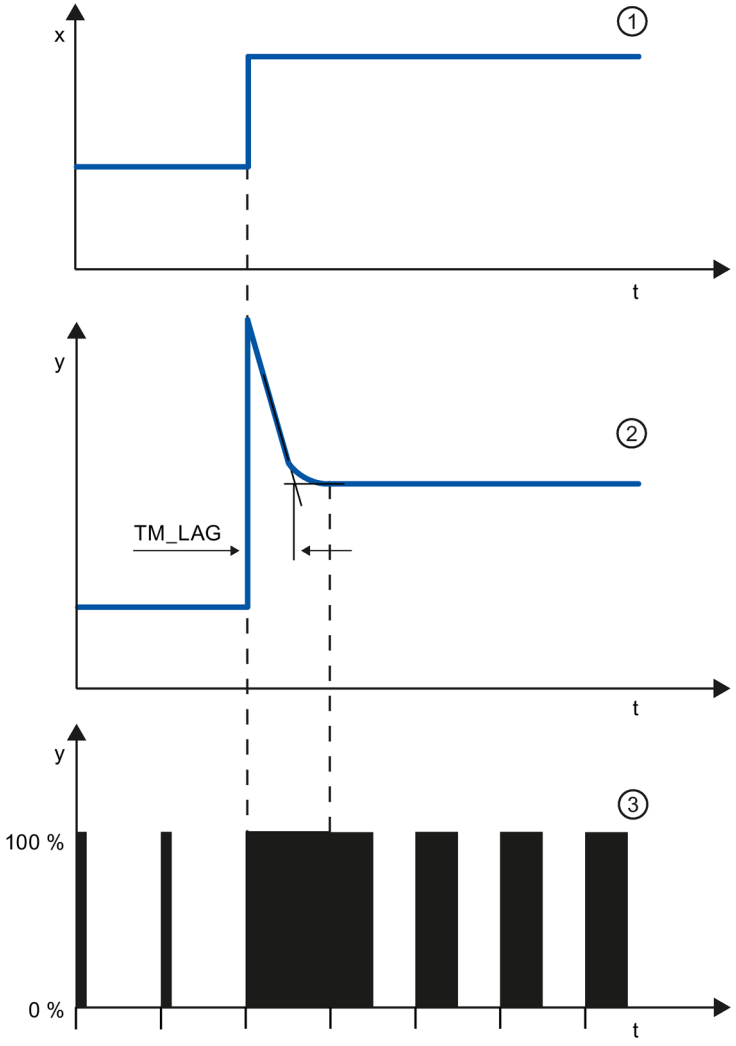
比例作用控制器的等式

输出值和控制偏差成正比，即：

输出值 = 比例增益 × 控制偏差

y = GAIN × x

PD 作用控制器的阶跃响应



① 控制偏差
② 连续控制器的输出值
③ 脉冲控制器的输出值
TM_LAG 微分作用的延迟

PD 作用控制器的等式

以下等式适用于在一定时间范围内 PD 作用控制器的阶跃响应：

$$y = GAIN \cdot X_W \cdot \left(1 + \frac{TD}{TM\_LAG} \cdot e^{\frac{-t}{TM\_LAG}} \right)$$

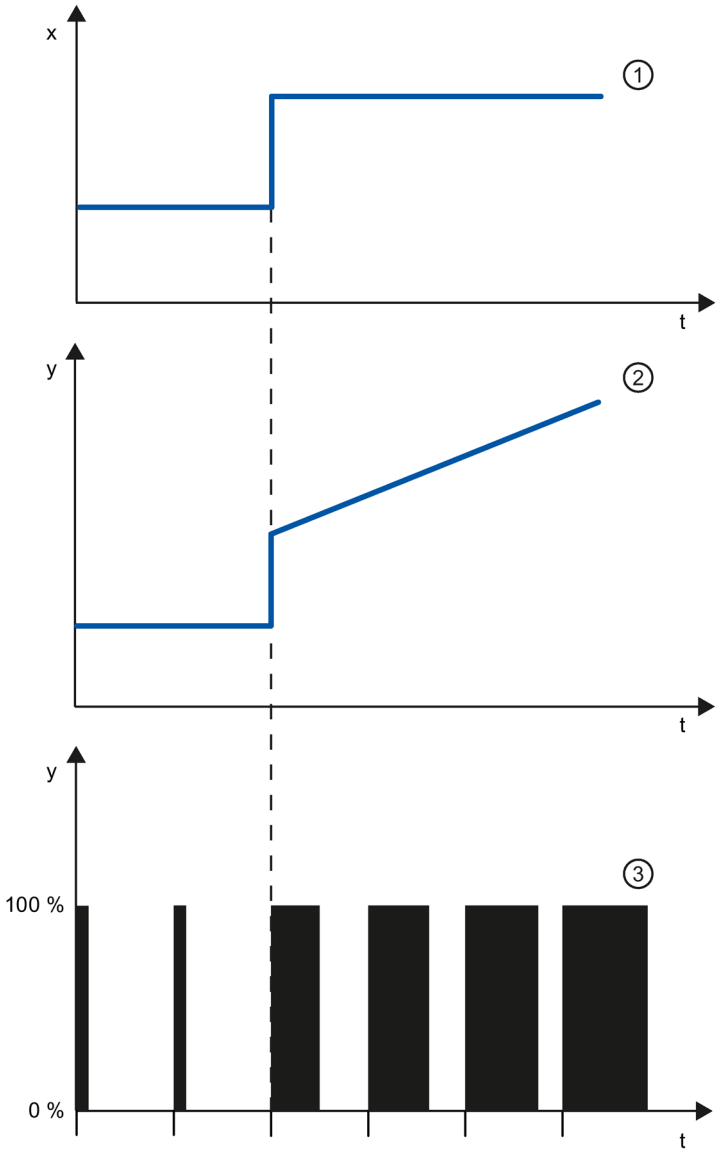
t = 自控制偏差阶跃后的时间间隔

微分作用根据过程值的变化率生成输出值。 微分作用本身不适合进行控制，因为输出值仅随过程值的阶跃而发生变化。 只要过程值保持恒定，输出值就不会再发生变化。

通过与比例作用相结合，可以改进对微分作用干扰的响应。 但无法完全校正干扰。 好的动态响应是有好处的。 在逼近和设定值改变期间可实现获得有效衰减的非波动响应。

如果受控系统具有脉冲测量的量（例如，在压力或流量控制系统中），则具有微分作用的控制器不适用。

PI 作用控制器的阶跃响应



① 控制偏差
② 连续控制器的输出值
③ 脉冲控制器的输出值

控制器中的积分作用会随时间而使控制偏差增大。 这意味着控制器会一直对系统进行校正，直到控制偏差消除为止。 持续控制偏差只会在具有比例作用的控制器中生成。 这种影响可通过控制器中的积分作用来消除。

根据对控制响应的要求，在实际操作中最好将比例、积分和微分作用结合使用。 各个分量的时间响应可通过控制器参数比例增益 GAIN、积分作用时间 TI （积分作用）和微分作用时间 TD（微分作用）来描述。

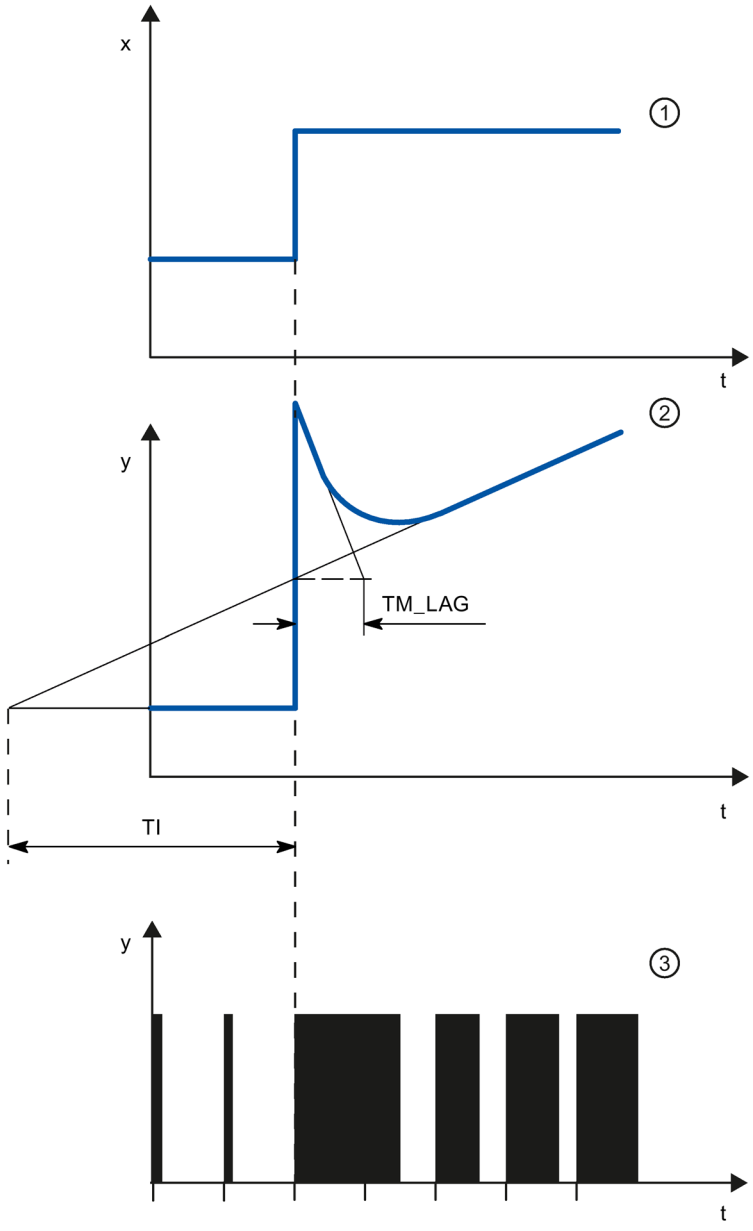
PI 作用控制器的等式

以下等式适用于在一定时间范围内 PI 作用控制器的阶跃响应：

$$y = GAIN \cdot X_W \cdot \left(1 + \frac{1}{TI \cdot t}\right)$$

t = 自控制偏差阶跃后的时间间隔

PID 控制器的阶跃响应



①	控制偏差
②	连续控制器的输出值
③	脉冲控制器的输出值
TM_LAG	微分作用的延迟
T_i	积分作用时间

PID 控制器的等式

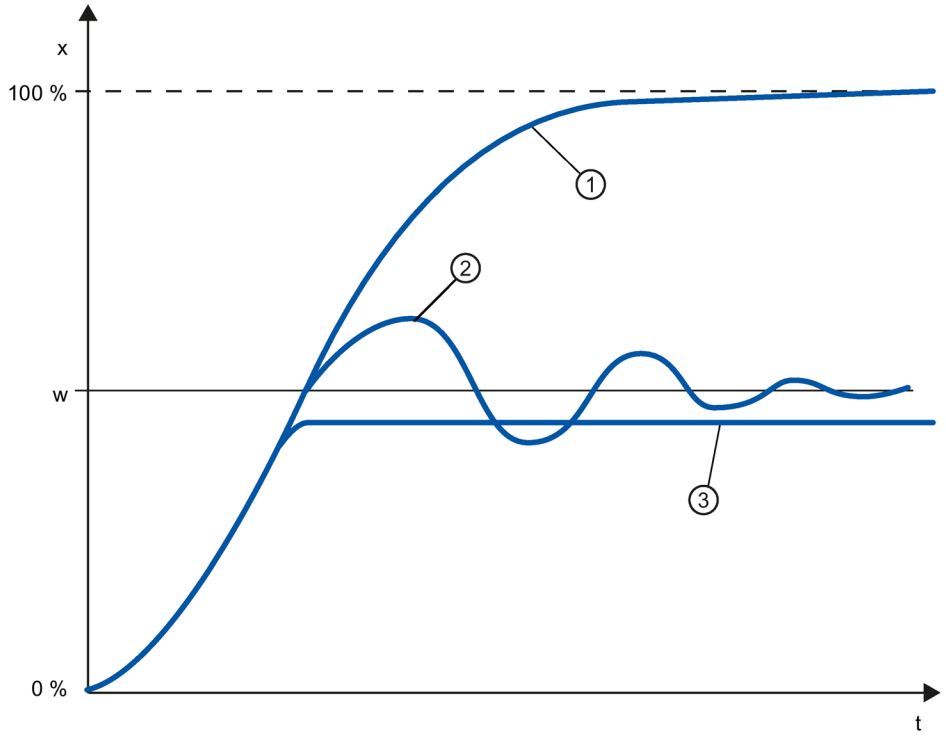
以下等式适用于在一定时间范围内 PID 控制器的阶跃响应：

$$y = GAIN \cdot X_w \cdot \left(1 + \frac{1}{TI \cdot t} + \frac{TD}{TM\_LAG} \cdot e^{\frac{-t}{TM\_LAG}} \right)$$

t = 自控制偏差阶跃后的时间间隔

具有不同控制器结构的受控系统的响应

过程工程中的大多数控制器系统都可以通过具有 PI 作用响应的控制器进行控制。 在具有较长空载时间的慢速控制系统情况中（例如，温度控制系统），可通过具有 PID 作用的控制器提高控制结果。



①	无控制器
②	PID 控制器
③	PD 作用控制器
w	设定值
x	过程值

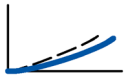
具有 PI 和 PID 作用的控制器的优势在于，过程值在稳定后不会与设定值之间存在任何偏差。 过程值在逼近过程中会在设定值周围振荡。

2.7 为指定受控系统选择控制器结构

对适用的控制器结构的选择

为实现最佳控制结果，应选择适合于受控系统且可在特定限值范围内适应受控系统的控制器结构。

下表概述了控制器结构与受控系统的适当组合。

受控系统		控制器结构			
		P	PD	PI	PID
	仅具有死时间	不适用	不适用	适用	不适用
	具有死时间的 PT1	不适用	不适用	非常适用	非常适用
	具有死时间的 PT2	不适用	有条件适用	非常适用	非常适用
	高阶	不适用	不适用	有条件适用	非常适用
	非自调节	非常适用	非常适用	非常适用	非常适用

下表概述了控制器结构与物理量的各种组合的适宜性。

物理量	控制器结构			
	P	PD	PI	PID
	持续控制偏差		无持续控制偏差	
温度	适用于对性能要求较低的系统以及 $T_u/Tg < 0,1$ 的比例作用受控系统	非常适用	最适用于高性能要求的控制器结构（除了经过特殊调整的特殊控制器）	
压力	适用，如果不考虑延迟时间	不适用	最适用于高性能要求的控制器结构（除了经过特殊调整的特殊控制器）	
流速	不适用，因为所需的 GAIN 范围通常过大	不适用	适合，但是单独使用积分作用控制器通常会更好	几乎不需要

2.8 PID 参数设置

参数设置的经验

控制器结构	设置
P	GAIN ≈ v_{max} × T_u [°C]
PI	GAIN ≈ 1.2 × v_{max} × T_u [°C] TI ≈ 4 × T_u [min]
PD	GAIN ≈ 0.83 × v_{max} × T_u [°C] TD ≈ 0.25 × v_{max} × T_u [min] TM_LAG ≈ 0.5 × TD[min]
PID	GAIN ≈ 0.83 × v_{max} × T_u [°C] TI ≈ 2 × T_u [min] TD ≈ 0.4 × T_u [min] TM_LAG ≈ 0.5 × TD[min]
PD/PID	GAIN ≈ 0.4 × v_{max} × T_u [°C] TI ≈ 2 × T_u [min] TD ≈ 0.4 × T_u [min] TM_LAG ≈ 0.5 × TD[min]

除了 $v_{max} = \Delta_x / \Delta_t$，还可以使用 X_{max} / T_g。

如果控制器具有 PID 结构，则积分作用时间的设置和微分作用时间的设置通常会相互结合。

比率 TI/TD 介于 4 和 5 之间，这对于大多数受控系统都是最优的。

在 PD 控制器中，不遵守微分作用时间 TD 并不重要。

对于 PI 和 PID 控制器，如果大部分情况下选择的积分作用时间 TI 过短，则会发生控制振荡。

如果积分作用时间过长，则会降低干扰的稳定速度。 不要希望进行第一次参数设置后，控制回路工作状态就能达到“最优”状态。 经验表明，当系统处于 $T_u / T_g > 0.3$ “难以控制”状态时，进行调整是很必要的。

组态软件控制器

3

3.1 软件控制器概述

要组态软件控制器，需要使用包含控制算法的指令和工艺对象。软件控制器的工艺对象相当于指令的背景 DB。控制器的组态数据保存在工艺对象中。与其它指令的背景数据块不同，工艺对象并非存储在程序资源中，而是存储在 CPU > 工艺对象下。

工艺对象和指令

CPU	库	指令	工艺对象	说明
S7-1200	Compact PID	PID_Compact V1.x	PID_Compact V1.x	具有集成调节功能的通用 PID 控制器
S7-1200		PID_3Step V1.x	PID_3Step V1.x	对阀门进行集成调节的 PID 控制器
S7-1500 S7-1200 V4.x		PID_Compact V2.x	PID_Compact V2.x	具有集成调节功能的通用 PID 控制器
S7-1500 S7-1200 V4.x		PID_3Step V2.x	PID_3Step V2.x	对阀门进行集成调节的 PID 控制器
S7-1500 ≥ V1.7 S7-1200 ≥ V4.1		PID_Temp V1.x	PID_Temp V1.x	具有集成调节功能的通用 PID 温度控制器
S7-1500/300/400	PID 基本功能	CONT_C	CONT_C	连续控制器
S7-1500/300/400		CONT_S	CONT_S	适合具有积分行为的执行器的步进控制器
S7-1500/300/400		PULSEGEN	-	适合具有比例行为的执行器的脉冲发生器
S7-1500/300/400		TCONT_CP	TCONT_CP	具有脉冲发生器的连续温度控制器
S7-1500/300/400		TCONT_S	TCONT_S	适合具有积分行为的执行器的温度控制器
S7-300/400	PID Self Tuner	TUN_EC	TUN_EC	连续控制器的优化
S7-300/400		TUN_ES	TUN_ES	步进控制器的优化

CPU	库	指令	工艺对象	说明
S7-300/400	Standard PID Control（PID Profes-sional 可选包）	PID_CP	PID_CP	具有脉冲发生器的连续控制器
S7-300/400		PID_ES	PID_ES	适合具有积分行为的执行器的步进控制器
S7-300/400		LP_SCHED	-	分配控制器调用
S7-300/400	Modular PID Control（PID Profes-sional 可选包）	A_DEAD_B	-	过滤控制偏差中的干扰信号
S7-300/400		CRP_IN	-	标定模拟量输入信号
S7-300/400		CRP_OUT	-	标定模拟量输出信号
S7-300/400		DEAD_T	-	延时输出输入信号
S7-300/400		DEADBAND	-	抑制过程值的微小波动
S7-300/400		DIF	-	将输入信号对时间差分
S7-300/400		ERR_MON		监视控制偏差
S7-300/400		INTEG	-	将输入信号对时间积分
S7-300/400		LAG1ST	-	一阶延迟元件
S7-300/400		LAG2ND	-	二阶延迟元件
S7-300/400		LIMALARM	-	报告限值
S7-300/400		LIMITER	-	限制调节变量
S7-300/400		LMNGEN_C	-	确定连续控制器的调节变量
S7-300/400		LMNGEN_S	-	确定步进控制器的调节变量
S7-300/400		NONLIN	-	线性化编码器信号
S7-300/400		NORM	-	物理标定过程值
S7-300/400		OVERRIDE	-	将调节变量从 2 个 PID 控制器切换至 1 个执行器
S7-300/400		PARA_CTL	-	切换参数集
S7-300/400		PID	-	PID 算法
S7-300/400		PUSLEGEN_M	-	为比例执行器生成脉冲
S7-300/400		RMP_SOAK	-	根据斜坡/保持函数指定设定值
S7-300/400		ROC_LIM	-	限制变化率
S7-300/400		SCALE_M	-	标定过程值

CPU	库	指令	工艺对象	说明
S7-300/400		SP_GEN	-	手动指定设定值
S7-300/400		SPLT_RAN	-	拆分调节变量范围
S7-300/400		SWITCH	-	切换模拟值
S7-300/400		LP_SCHED_M	-	分配控制器调用

3.2 组态软件控制器的步骤

所有软件控制器都按照相同的方案进行组态：

步骤	说明
1	添加工艺对象 (页 44)
2	组态工艺对象 (页 48)
3	在用户程序中调用指令 (页 50)
4	将工艺对象下载到设备 (页 76)
5	调试软件控制器 (页 78)
6	将优化的 PID 参数保存到项目中 (页 78)
7	比较值 (页 46)
8	显示工艺对象的背景 (页 79)

3.3 添加工艺对象

在项目浏览器中添加工艺对象

添加工艺对象时，会为该工艺对象的指令创建一个背景 DB。 工艺对象的组态存储在该背景数据块中。

要求

已创建具有 CPU 的项目。

步骤

要添加工艺对象，请按以下步骤操作：

1. 在项目树中打开 CPU 文件夹。
2. 打开“工艺对象”(Technology objects) 文件夹。
3. 双击“添加新对象”(Add new object)。
   将打开“添加新对象”(Add new object) 对话框。
4. 单击“PID”按钮。
   将显示所有可用于该 CPU 的 PID 控制器。
5. 选择该工艺对象的指令，例如，PID_Compact。
6. 在“名称”(Name) 输入域中输入该工艺对象的专用名称。
7. 如果要更改背景数据块的推荐数据块编号，请选择“手动”(Manual) 选项。
8. 如果想要为该工艺对象添加用户信息，请单击“更多信息”(Further information)。
9. 单击“确定”(OK) 进行确认。

结果

新工艺对象已创建，并存储在项目树的“工艺对象”(Technology objects) 文件夹中。 如果在循环中断 OB 中调用该工艺对象的指令，则将使用该对象。

说明

可以选中该对话框底部的“添加并打开新对象”(Add new and open) 复选框。 这将在添加操作完成后打开工艺对象的组态。

3.4 比较值

3.4.1 比较显示和约束条件

“比较值”功能提供了以下选项：

- 将项目中组态的起始值与 CPU 中的起始值和实际值进行比较
- 直接编辑实际值和项目的起始值
- 立即检测并显示输入错误和建议的更正措施
- 备份项目中的实际值
- 将项目的起始值作为实际值传送至 CPU

图标和操作员控件

提供以下图标和操作员控件：

图标	功能
	PLC 起始值与已组态的项目起始值相匹配。
	PLC 起始值与已组态的项目起始值不匹配。
	无法将 PLC 起始值与已组态的项目起始值进行比较
	两个比较值中至少有一个具有过程相关错误或语法错误。
	将实际值传送至离线项目
	将项目中已更新的起始值传送至 CPU（初始化设置值）
	打开“比较值”(Compare values) 对话框

约束条件

“比较值”功能适用于 S7-1200 和 S7-1500，不受限制。

以下限制适用于 S7-300 和 S7-400：

在监视模式下，S7-300/S7-400 无法向 CPU 传送起始值。 这些值无法通过“比较值”(Compare values) 在线显示。

可显示工艺对象的实际值并可直接对其进行修改。

3.4.2 比较值

下面以“PID 参数”为例说明具体步骤。

要求

- 组态具有软件控制器的项目。
- 将项目下载到 CPU 中。
- 在项目浏览器中打开组态对话框。

步骤

1. 在项目导航中打开所需软件控制器。
2. 双击“组态”(Configuration) 对象。
3. 在组态窗口中导航至“PID 参数”(PID Parameters) 对话框。
4. 单击  图标，激活监视模式。

   “比较值”功能的图标和操作员控件 (页 45)将在参数后显示。
5. 在输入框中单击所需参数并通过直接输入的方式手动更改参数值。
   - 如果输入框的背景为灰色，则表示该值为只读值，无法修改。
   - 要在“PID 参数”(PID Parameters) 对话框中更改值，需事先选中“启用手动输入”(Enable manual entry) 复选框来启用手动输入。
6. 单击  图标，打开起始值对话框。

   该对话框指示两个参数值：
   - CPU 中的起始值： CPU 中的起始值显示在顶部。
   - 项目中的起始值： 项目中组态的起始值显示在底部。
7. 在项目的输入框中输入所需值。

错误检测

检测到输入的值不正确。 这种情况下会提供建议的更正措施。

如果输入的值含有错误的语法，则参数下将打开包含相应错误消息的弹出窗口。 不会应用该错误值。

如果输入的值不适合过程，则会打开一个对话框，其中包含错误消息和建议的更正措施：

- 单击“否”(No) 接受建议的更正措施并修改输入。
- 单击“确定”(OK) 应用错误值。

注意
控制器故障 不适合过程的值会导致控制器发生故障。

备份实际值

单击 图标，将控制器的实际值传送到所组态项目的起始值。

将项目值传送到 CPU

单击 图标，将项目中组态的值传送到 CPU。

⚠小心
防止人身伤害和财产损失！ 在设备运行时下载和复位用户程序，可能会在发生故障或程序错误的情况下导致重大财产损失和严重人身伤害。 在下载和复位用户程序前，确保不会出现危险情况。

3.5 组态工艺对象

S7-1200 CPU 中的工艺对象的属性可以两种方式组态。

- 在程序编辑器的巡视窗口中
- 在组态窗口中

S7-300/400 CPU 中的工艺对象的属性只能在组态编辑器中组态。

程序编辑器的巡视窗口

在程序编辑器的巡视窗口中，只能组态所需的运行参数。

在线模式下还显示参数的离线值。只能在调试窗口中更改在线值。

要打开工艺对象的巡视窗口，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。
2. 双击要打开软件控制器的指令的块（循环中断 OB）。
   该块将在工作区中打开。
3. 单击软件控制器的指令。
4. 在巡视窗口中，依次选择“属性”(Properties) 和“组态”(Configuration) 选项卡。

组态窗口

对于各工艺对象，有特定的组态窗口用于组态所有属性。

要打开工艺对象的组态窗口，请按以下步骤操作：

1. 在项目树中打开“工艺对象”(Technology objects) 文件夹。
2. 在项目树中打开该工艺对象。
3. 双击“组态”(Configuration) 对象。

符号

组态的区域导航以及巡视窗口中的图标显示有关组态完成情况的详细信息：

	组态包含默认值且已完成。 组态仅包含默认值。通过这些默认值即可使用工艺对象，而无需进一步更改。
	组态包含用户定义或自动调整的值且已完成。 组态的所有输入字段中均包含有效值，而且至少更改了一个默认设置。
	组态不完整或有缺陷。 至少一个输入字段或可折叠列表不包含任何值或者包含一个无效值。相应域或下拉列表框的背景为红色。单击时，弹出的错误消息会指示错误原因。

工艺对象部分中详细介绍了工艺对象的属性。

3.6 在用户程序中调用指令

必须在循环中断 OB 中调用软件控制器的指令。软件控制器的采样时间由循环中断 OB 中两次调用的时间间隔决定。

要求

已创建循环中断 OB 并且循环中断 OB 的循环时间组态正确。

步骤

要在用户程序中调用指令，请按以下步骤操作：

1. 在项目树中打开 CPU 文件夹。
2. 打开“程序块”(Program blocks) 文件夹。
3. 双击循环中断 OB。
   该块将在工作区中打开。
4. 在“指令”(Instructions) 窗口和“PID 控制”(PID Control) 文件夹中打开“工艺”(Technology) 组。
   该文件夹包含可在 CPU 中组态的软件控制器的所有指令。
5. 选择指令，并将其拖动到循环中断 OB 中。
   “调用选项”(Call options) 对话框随之打开。
6. 从“名称”(Name) 列表中选择一个工艺对象或为新工艺对象输入名称。

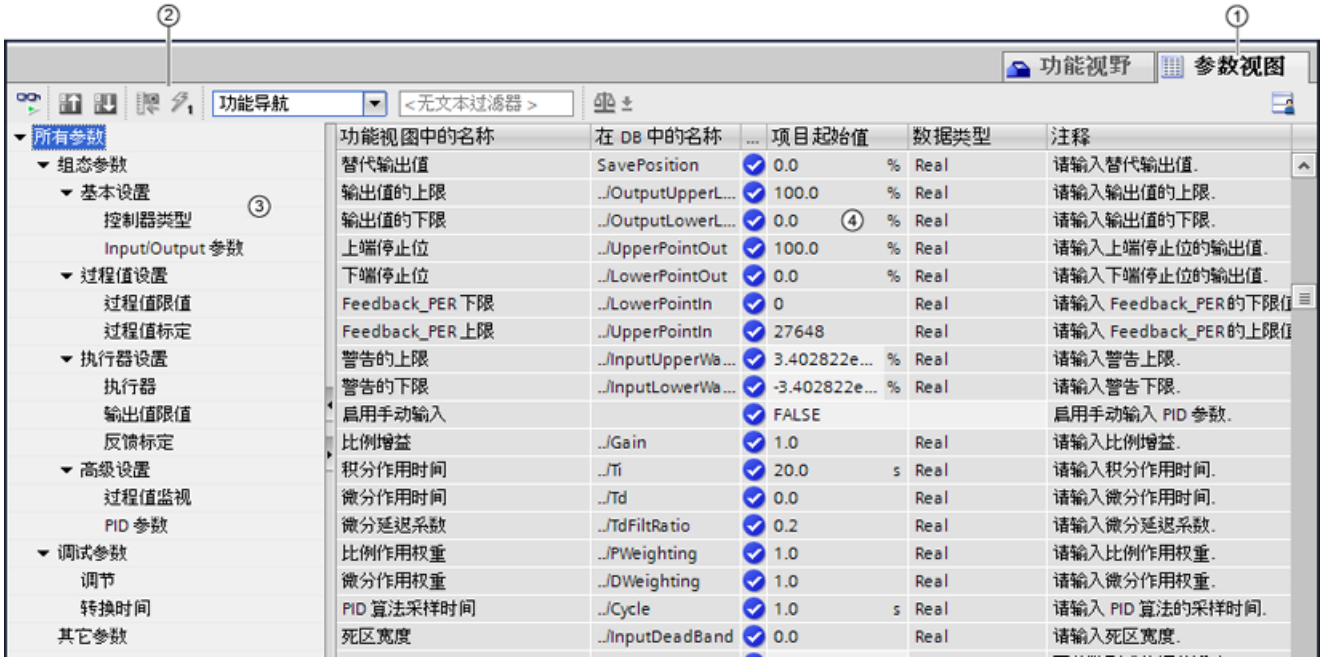
结果

如果工艺对象尚不存在，则会添加工艺对象。 该指令添加到循环中断 OB。该工艺对象分配给该指令的此调用。

3.7 参数视图

3.7.1 参数视图简介

参数视图提供了工艺对象中所有相关参数的一般概述。 可获得参数设置的概述，并可在离线和在线模式下轻松地对其进行更改。



① “参数视图”(Parameter view) 选项卡

② 工具栏 (页 54)

③ 导航 (页 55)

④ 参数表 (页 55)

功能范围

提供以下可用于分析工艺对象参数和启用目标性监视与修改的功能。

显示功能：

- 在离线和在线模式下显示参数值
- 显示参数的状态信息
- 显示值偏差和直接连接选项
- 显示组态错误
- 显示由参数引起的值更改
- 显示某参数所有的存储值： PLC 起始值、项目起始值、监视值
- 显示参数存储值的参数比较

操作员控制功能：

- 为在参数之间和参数结构之间进行快速更改而导航。
- 用于更快搜索具体参数的文本过滤器。
- 用于按需自定义参数和参数组顺序的排序功能。
- 用于备份参数视图的结构设置的存储功能。
- 在线监视和修改参数值。
- 为捕获并响应瞬时情况而保存 CPU 参数值快照的功能。
- 用于将参数值快照应用为起始值的功能。
- 将已修改的起始值下载至 CPU。
- 用于比较两个参数值的比较功能。

有效性

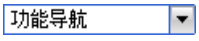
此处所述的“参数视图”(Parameter view) 适用于以下工艺对象：

- PID_Compact
- PID_3Step
- PID_Temp
- CONT_C（仅适用于 S7-1500）
- CONT_S（仅适用于 S7-1500）
- TCONT_CP（仅适用于 S7-1500）
- TCONT_S（仅适用于 S7-1500）
- TO_Axis_PTO（S7-1200 运动控制）
- TO_Positioning_Axis（S7-1200 运动控制）
- TO_CommandTable_PTO（S7-1200 运动控制）
- TO_CommandTable（S7-1200 运动控制）

3.7.2 参数视图结构

3.7.2.1 工具栏

可在参数视图的工具栏中选择以下功能。

图标	功能	说明
	监视全部	在活动的参数视图中启动可见参数监视（在线模式）。
	创建监视值的快照并将该快照的设定值接受为起始值	将当前的监视值应用到“快照”(Snapshot) 列，并更新项目中的起始值。 仅可在 PID_Compact、PID_3Step 和 PID_Temp 的在线模式下执行。
	初始化设定值	把在项目中更新过的起始值传送至 CPU。 仅可在 PID_Compact、PID_3Step 和 PID_Temp 的在线模式下执行。
	创建监视值的快照	将当前的监视值应用到“快照”(Snapshot) 列。 仅可在在线模式下执行。
	请立即一次性修改全部选定参数	该命令尽快执行一次，而不参考用户程序中的任何特定点。 仅可在在线模式下执行。
	选择导航结构	在功能导航和数据导航之间进行切换。
	文本过滤器...	在输入字符串之后：显示某一当前可见列中所有包括指定字符串在内的参数。
	选择比较值	在在线模式下，选择要与另一个参数值进行比较的参数值（项目起始值、PLC 起始值、快照） 仅可在在线模式下执行。
	保存窗口设置	为参数视图保存显示设置（例如选择的导航结构和激活的表中的列等）

3.7.2.2 导航

在“参数视图”(Parameter view) 选项卡中，有以下替代导航结构可供选择。

导航		说明
功能导航		在功能导航中，参数结构以组态对话框（“功能视图”(Functional view) 选项卡）、调试对话框和诊断对话框中的结构为基础。 最后一个组“其它参数”(Other parameters) 包括工艺对象的所有其它参数。
数据导航		在数据导航中，参数结构以背景数据块/工艺数据块中的结构为基础。 最后一个组“其它参数”(Other parameters) 包括背景数据块/工艺数据库中不包括的参数。

可以使用“选择导航结构”(Select navigation structure) 下拉列表来切换导航结构。

3.7.2.3 参数表

下表给出了参数表各列的含义。 可以根据需要显示或隐藏列。

- “离线”列 = X： 该列在离线模式下可见。
- “在线”列 = X： 该列在在线模式下可见（在线连接到 CPU）。

列	说明	离线	在线
功能视图中的名称	功能视图中的参数名称。 如果参数未通过工艺对象组态，该显示字段留空。	X	X
DB 中的全称	背景数据块/工艺数据块中参数的完整路径。 如果参数未包含在背景数据块/工艺数据块中，该显示字段留空。	X	X
在 DB 中的名称	背景数据块/工艺数据块中的参数名称。 如果参数是某结构或 UDT 的一部分，则应添加前缀“. ./”。 如果参数未包含在背景数据块/工艺数据块中，该显示字段留空。	X	X
组态的状态	使用状态符号显示组态完整性。 请参见组态的状态（离线） (页 67)	X	

列	说明	离线	在线
比较结果	“比较值”功能的结果。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮，则会显示该列。		X
项目起始值	在项目中组态起始值。 如果输入的值具有语法错误或过程相关错误，则会出现错误指示。	X	X
默认值	为该参数预分配的值。 如果参数未包含在背景数据块/工艺数据块中，该显示字段留空。	X	X
快照	CPU 中当前值的快照（监视值）。 如果值具有过程相关错误，则会出现错误指示。	X	X
PLC 起始值	CPU 中的起始值。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮，则会显示该列。 如果值具有过程相关错误，则会出现错误指示。		X
监视值	CPU 中的当前值。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮，则会显示该列。 如果值具有过程相关错误，则会出现错误指示。		X
修改值	用于更改监视值的值。 如果已在线连接并选择了“监视所有”(Monitor all) 按钮，则会显示该列。 如果输入的值具有语法错误或过程相关错误，则会出现错误指示。		X
选择用于传输	使用“请立即一次性修改全部选定参数”(Modify all selected parameters immediately and once) 按钮选择要传输的修改值。 该列与“修改值”(Modify value) 列一起显示。		X
最小值	参数的最小过程相关值。 如果最小值取决于其它参数，则将其定义为： • 离线： 由项目起始值决定。 • 在线： 由监视值决定。	X	X

列	说明	离线	在线
最大值	参数的最大过程相关值。 如果最大值取决于其它参数，则将其定义为： • 离线： 由项目起始值决定。 • 在线： 由监视值决定。	X	X
设定值	将参数指定为设定值。 可在线初始化这些参数。	X	X
数据类型	变量的数据类型。 如果参数未包含在背景数据块/工艺数据块中，该显示字段留空。	X	X
保持性	将值指定为保持值。 保持性参数的值将保留，即使在电源关闭后也是如此。	X	X
可从 HMI 访问	指示运行期间 HMI 是否可以访问此参数。	X	X
HMI 中可见	指示 HMI 选择列表中的参数是否默认可见。	X	X
注释	参数的简要描述。	X	X

参见

比较值 (页 45)

3.7.3 打开参数视图

要求

工艺对象已添加到项目树中，即已创建指令的相关背景数据块/工艺数据块。

步骤

1. 在项目树中打开“工艺对象”(Technology objects) 文件夹。
2. 在项目树中打开该工艺对象。
3. 双击“组态”(Configuration) 对象。
4. 选择右上角的“参数视图”(Parameter view) 选项卡。

结果

参数视图将打开。 所显示的每个参数都会以参数表中的一行表示。

可显示的参数属性（表的列）根据所使用的参数视图模式（离线或在线）而有所不同。

此外，可以有选择地显示和隐藏表的各个列。

参见

参数视图默认设置 (页 59)

3.7.4 参数视图默认设置

默认设置

为了让您高效使用参数视图，可以自定义参数显示并保存设置。

可以执行并保存下列自定义项：

- 显示和隐藏列
- 改变列宽
- 改变列的顺序
- 切换导航
- 在导航中选择参数组
- 选择比较值

显示和隐藏列

要显示或隐藏参数表中的列，请按以下步骤操作：

1. 将光标放在参数表的标题上。
2. 在快捷菜单中选择“显示/隐藏”(Show/Hide) 命令。
   将显示可选的列。
3. 要显示列，请选中该列的复选框。
4. 要隐藏列，请清除该列的复选框。

或

1. 将光标放在参数表的标题上。
2. 如果要显示所有离线或在线模式的列，请在快捷菜单中选择“显示所有列”(Show all columns) 命令。

一些列只能在在线模式下显示： 请参见参数表 (页 55)。

改变列宽

要自定义列宽以便阅读所有行中的文本，请按以下步骤操作：

1. 将光标放在参数表标题中要进行自定义的列的右侧，直到光标的形状变为十字。
2. 然后双击此位置。

或

1. 打开参数表标题上的快捷菜单。
2. 点击
   - ""优化列宽"(Optimize column width) 或
   - ""优化所有列的宽度"(Optimize width of all columns)。

如果列宽设置得过窄，可将光标短暂悬停在相关字段上，即可出现各个字段的完整内容。

改变列的顺序

参数表的列可按照任意方式排列。

要更改列的顺序，请按以下步骤操作：

1. 单击列标题并使用拖放操作将其移动至所需位置。

   当松开鼠标时，对象会锚定在新位置上。

切换导航

要切换参数的显示格式，请执行以下步骤：

1. 在"选择导航结构"(Select navigation structure) 下拉列表中选择所需的导航。
   - 数据导航
   - 功能导航

另请参见导航 (页 55)。

在导航中选择参数组

在所选导航内，可在显示"所有参数"(All parameters) 或显示所选的下级参数组二者中选择其一。

1. 在导航中单击所需的参数组。

   参数表仅显示该参数组的参数。

选择比较值（在线）

要为“比较值”(Compare values) 功能设置比较值，请按下列步骤操作：

1. 在“选择比较值”(Selection of compare values) 下拉列表中选择所需的比较值。
   - 项目起始值/PLC 起始值
   - 项目起始值/快照
   - PLC 起始值/快照

默认情况下，会设置“项目起始值/PLC 起始值”(Start value project / Start value PLC) 选项。

保存“参数视图”(Parameter view) 的默认设置

要保存参数视图的以上自定义项，请按下列步骤操作：

1. 根据需要自定义参数视图。
2. 单击参数视图右上角的“保存窗口设置”(Save window settings) 按钮。

3.7.5 使用参数视图

3.7.5.1 概述

如下文所述，下表提供了在线和离线状态下的参数视图功能总览。

- “离线”列 = X： 只能在离线模式下使用本功能。
- “在线”列 = X： 只能在在线模式下使用本功能。

功能/操作	离线	在线
过滤参数表 (页 63)	X	X
将参数表排序 (页 64)	X	X
将参数数据传送给其它编辑器 (页 64)	X	X
指示错误 (页 65)	X	X
在项目中编辑起始值 (页 65)	X	X
组态的状态（离线） (页 67)	X	
参数视图中的在线监视值 (页 68)		X
创建监视值的快照 (页 69)		X
修改值 (页 70)		X
比较值 (页 72)		X
将来自在线程序的值应用为起始值 (页 74)		X
初始化在线程序中的设定值 (页 75)		X

3.7.5.2 过滤参数表

可通过下列方式过滤参数表中的参数：

- 使用文本过滤器
- 使用导航子组

两种过滤方法可同时使用。

使用文本过滤器

可过滤参数表中可见的文本。 这表示只可以过滤显示的参数行和参数列中的文本。

1. 在“文本过滤器...”(Text filter...) 输入框中输入所需字符串以进行过滤。

   参数表仅显示包含该字符串的参数。

文本过滤器会在下列情况下复位。

- 当选择了导航中的另一个参数组时。
- 当导航从数据导航更改为功能导航，或从功能导航更改为数据导航时。

使用导航子组

1. 在导航中单击所需的参数组，例如“静态”(Static)。

   参数表仅会显示静态参数。 可以为导航的一些组选择其它子组。

2. 如果要再次显示所有参数，请在导航中单击“所有参数”(All parameters)。

3.7.5.3 将参数表排序

参数值按行排列。 参数表可以按照任意显示的列进行排序。

- 包含数字值的列会按照数字值的峰值进行排序。
- 文本列会按照字母顺序进行排序。

按列排序

1. 将光标放在所需列的标题单元格中。

   该单元格的背景会变为蓝色。

2. 单击列标题。

结果

整个参数表会按照所选的列进行排序。 列标题中会出现一个尖向上的三角形。

再次单击列标题会按以下情况更改排序方式：

- 符号“▲”： 参数表按升序排序。
- 符号“▼”： 参数表按降序排序。
- 无符号： 再次移除排序。 参数表采用默认显示。

排序时，忽略“在 DB 中的名称”列的“../”前缀。

3.7.5.4 将参数数据传送给其它编辑器

在选择了参数表的整个参数行之后，可以使用以下操作：

- 拖放
- <Ctrl+C>/<Ctrl+V>
- 通过快捷菜单中的复制/粘贴

将参数传送给 TIA Portal 的以下编辑器：

- 程序编辑器
- 监视表
- 用于跟踪功能的信号表

参数会以全称插入： 请参见“DB 中的全称”(Full name in DB) 列中的信息。

3.7.5.5 指示错误

错误指示

会导致编译错误（例如越限）的参数分配错误将表示在参数视图中。

每当在参数视图中输入值，就会检查是否有过程相关错误和语法错误，并将结果表示出来。

错误的值会由以下方法表示：

- “组态的状态”（离线模式）或“比较结果”（在线模式，取决于所选的比较类型）列中的红色错误符号

和/或

- 背景为红色的表字段

  如果单击错误的字段，会出现弹出错误消息，其中包含有关允许的值范围或所需语法（格式）的信息

编译错误

如果参数未显示在组态对话框中，可以从编译器的错误消息处直接打开包含出错参数的参数视图（功能导航）。

3.7.5.6 在项目中编辑起始值

可使用参数视图在离线模式和在线模式下编辑项目中的起始值。

- 可在参数表的“项目起始值”(Start value project) 列中更改值。
- 在参数表的“组态状态”(Status of configuration) 列中，会通过与工艺对象组态对话框中相似的状态符号来表示组态进度。

约束条件

- 如果其它参数取决于那些起始值发生更改的参数，那么这些相关参数的起始值也会发生调整。
- 如果工艺对象的参数不可编辑，则也无法在参数视图中对其进行编辑。 参数是否可以编辑还取决于其它参数值。

定义新起始值

要在参数视图中定义参数的起始值，请按下列步骤操作：

1. 打开工艺对象的参数视图。
2. 在“项目起始值”(Start value project) 列中输入所需的起始值。 该值必须与参数的数据类型相匹配，不能超过参数的值范围。
   “最大值”(Maximum value) 和“最小值”(Minimum value) 列中给出了值范围的限值。

“组态的状态”(Status of configuration) 列用彩色符号表示组态进度。

另请参见 组态的状态（离线） (页 67)

在调整了起始值并将工艺对象下载到 CPU 的情况下，如果未将参数声明为保持（“保留”(Retain) 列），则参数会在启动时采用定义值。

错误指示

当输入了起始值时，会检查是否有过程相关错误和语法错误，并将结果表示出来。

错误的起始值会由以下方法表示：

- “组态的状态”（离线模式）或“比较结果”（在线模式，取决于所选的比较类型）列中的红色错误符号

和/或

- “项目起始值”(Start value project) 字段中的红色背景
  如果单击错误的字段，会出现弹出错误消息，其中包含有关允许的值范围或必要的语法（格式）的信息。

更正错误起始值

1. 使用来自弹出错误消息的信息更正错误的起始值。

   将不再显示红色错误信息、红色字段背景和弹出错误消息。

除非起始值无误，否则项目将无法成功编译。

3.7.5.7 组态的状态（离线）

表示组态状态的图标位于：

- 参数表中的“组态的状态”(Status of configuration) 列中
- 功能导航和数据导航的导航结构中

“组态的状态”列中的符号

符号	含义
	参数的起始值对应于默认值且有效。用户尚未定义起始值。
	参数的起始值中包含用户定义或自动调整的值。起始值与默认值不同。该起始值无误且有效。
	参数的起始值无效（语法或过程相关错误）。 输入框的背景为红色。单击弹出错误消息，会指出错误原因。
	仅限 S7-1200 运动控制： 参数的起始值有效但包含警告。 输入框的背景为黄色。
	在当前组态中，该参数不相关。

导航中的符号

导航中的符号指示组态过程的方式与工艺对象组态对话框中的方式相同。

参见

组态工艺对象 (页 48)

3.7.5.8 参数视图中的在线监视值

可直接在参数视图中监视 CPU 中工艺对象参数当前采用的值（监视值）。

要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。

步骤

1. 单击 启动监视。

   一旦参数视图在线，将额外显示以下各列：

   – 比较结果
   – PLC 起始值
   – 监视值
   – 修改值
   – 选择用于传输

   “监视值”(Monitor value) 列显示了 CPU 上的当前参数值。

   各附加列的含义： 请参见参数表 (页 55)
2. 再次单击 停止监视。

显示

所有仅在线时可用的列以橙色背景显示：

- 浅橙色单元格 中的值可以更改。
- 背景为深橙色 的单元格中的值无法更改。

3.7.5.9 创建监视值的快照

可在 CPU 上备份工艺对象的当前值（监视值）并将其显示在参数视图中。

要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮 已选择。

步骤

要显示当前参数值，请按以下步骤操作：

1. 在参数视图中，单击“创建监视值的快照”(Create snapshot of monitor values) 图标 。

结果

会向参数表的“快照”(Snapshot) 列传送一次当前监视值。

监视值会在“监视值”(Monitor values) 列中继续更新，此时可分析以此方式“冻结”的值。

3.7.5.10 修改值

可通过参数视图修改 CPU 中工艺对象的值。

可以向参数分配一次值（修改值）并立即对其进行修改。 修改请求会尽快执行，而不参考用户程序中的任何特定点。

⚠危险
修改时存在的危险： 在发生故障或程序错误的情况下，如果在设备运行时更改参数值，则可能会导致严重财产损失和人员重伤。 在使用“修改”功能之前，请确保不会发生危险。

要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮 已选择。
- 参数需可修改（“修改值”(Modify value) 列中的相关字段背景为浅橙色）。

步骤

要立即修改参数，请按以下步骤操作：

1. 在参数表的“修改值”(Modify value) 列中输入所需修改值。
2. 检查“选择用于传输”(Select for transmission) 列中用于修改的复选框是否已选中。

   修改值和相关参数的相关复选框会同时自动调整。
3. 单击“请立即一次性修改全部选定参数”(Modify all selected parameters immediately and once) 图标 。

选定参数会由指定值立即一次性修改，并可以在“修改值”(Modify values) 列中进行监视。“选择用于传输”(Selection for transmission) 列中用于修改的复选框会在修改请求完成之后自动清除。

错误指示

当输入了起始值时，会立即检查是否有过程相关错误和语法错误，并将结果表示出来。

错误的起始值会由以下方法表示：

- “修改值”(Modify value) 字段中的红色背景

并且

- 如果单击错误的字段，会出现弹出错误消息，其中包含有关允许的值范围或必要的语法（格式）的信息

错误修改值

- 具有过程相关错误的修改值可以进行传输。
- 具有语法错误的修改值**无法**进行传输。

3.7.5.11 比较值

可以使用比较功能来比较参数的以下存储值：

- 项目起始值
- PLC 起始值
- 快照

要求

- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮 已选择。

步骤

要比较不同目标系统中的起始值，请按下列步骤操作：

1. 单击“选择比较值”(Selection of compare values) 图标 。

   将打开一个包含比较选项的选择列表：

   – 项目起始值 - PLC 起始值（默认设置）
   – 项目起始值 - 快照
   – PLC 起始值 - 快照

2. 选择所需的比较选项。

   所选比较选项的执行方式如下：

   – 在选择进行比较的两个列的标题单元格中会出现刻度符号。
   – “比较结果”(Compare result) 列中使用的符号用来指示所选列的比较结果。

“比较结果”列中的符号

符号	含义
	比较值相等且无误。
	比较值不相等但无误。
	两个比较值中至少有一个具有过程相关错误或语法错误。
	无法进行比较。两个比较值中，至少一个不可用（如，快照）。
	由于该值与组态无关，不适用于进行比较。

导航中的符号

如果所显示的导航结构下方的参数中至少有一个应用了比较结果，则导航中的符号会以相同方式显示。

3.7.5.12 将来自在线程序的值应用为起始值

为了将来自 CPU 的优化值应用为项目的起始值，应创建监视值的快照。 标记为设定值的快照值将应用为项目的起始值。

要求

- 工艺对象的类型为“PID_Compact”或“PID_3Step”。
- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮 已选择。

步骤

要应用来自 CPU 的优化值，请按下列步骤操作：

1. 单击“创建监视值的快照并将该快照的设定值接受为起始值”(Create snapshot of monitor values and accept setpoints of this snapshot as start values) 图标 。

结果

应用到“快照”(Snapshot) 列及其设定值的当前监视值将作为新的起始值复制到“项目起始值”(Start value project) 列。

说明

应用各个参数的值

还可以将未标记为设定值的各个参数值从“快照”(Snapshot) 列应用到“项目起始值”(Start value project) 列。 为此，请使用快捷菜单中的“复制”(Copy) 和“粘贴”(Paste) 命令复制这些值并将其插入“项目起始值”(Start value project) 列中。

3.7.5.13 初始化在线程序中的设定值

在 CPU 中，仅通过一个步骤即可将所有在参数视图中标记为“设定值”(Setpoint) 的参数初始化为新值。 为此，请将起始值从项目下载到 CPU 中。 CPU 将保持为“RUN”模式。

为避免在冷启动或暖启动期间丢失 CPU 上的数据，请务必将工艺对象也下载到 CPU 中。

⚠危险
更改参数值时存在的危险 在发生故障或程序错误的情况下，如果在设备运行时更改参数值，则可能会导致严重财产损失和人员重伤。 在重新初始化设定值之前，请确保不会发生危险。

要求

- 工艺对象的类型为“PID_Compact”或“PID_3Step”。
- 需要有在线连接。
- 工艺对象已下载到 CPU 中。
- 程序执行处于激活状态（CPU 处于“RUN”模式）。
- 工艺对象的参数视图已打开。
- “监视所有”(Monitor all) 按钮 已选择。
- 标记为“设定值”(Setpoint) 的参数具有无过程相关错误和语法错误的“项目起始值”(Start value project)。

步骤

要初始化所有设定值，请按以下步骤操作：

1. 在“项目起始值”(Start value project) 列中输入所需的值。

   确保起始值没有过程相关错误和语法错误。

2. 单击“初始化设定值”（Initialize setpoints） 图标 。

结果

CPU 中的设定值初始化为项目的起始值。

3.8 将工艺对象下载到设备

必须将新的或修改的工艺对象组态下载到在线模式的 CPU。 下载保持性数据时下列特性适用：

- **软件（仅限更改）**
  - S7-1200、S7-1500：
    保留保持性数据。
  - S7-300/400：
    立即更新保持性数据。 CPU 不更改为 Stop 模式。
- **将 PLC 程序下载到设备并复位**
  - S7-1200、S7-1500：
    下次从 Stop 更改为 RUN 时更新保持性数据。 PLC 程序只能完全下载。
  - S7-300/400：
    下次从 Stop 更改为 RUN 时更新保持性数据。

将保持性数据下载到 S7-1200 或 S7-1500 CPU

说明

如果在执行系统操作期间下载和复位 PLC 程序时出现误操作或程序错误，则会造成严重的人员伤害或设备损坏。

在下载和复位 PLC 程序前，确保不会出现危险情况。

请按如下步骤下载保持性数据：

1. 在项目树中选择 CPU 条目。
2. 从“在线”(Online) 菜单中选择“下载和复位 PLC 程序”(Download and reset PLC program) 命令。
   - 如果尚未建立在线连接，则会打开“扩展的下载”(Extended download) 对话框。 这种情况下，设置连接所需的所有参数，然后单击“下载”(Download)。
   - 如果已定义在线连接，则可根据需要编译项目数据并打开“装载预览”(Load preview) 对话框。 此对话框会显示消息并建议下载必需的操作。
3. 检查这些消息。

   只要可进行下载，“下载”(Download) 按钮就会变为激活状态。

4. 单击“下载”(Download)。

   将下载完整的 PLC 程序并打开“装载结果”(Load results) 对话框。 此对话框会显示下载后的状态和操作。

5. 要在下载完成后立即重启模块，请选中“全部启动”(Start all) 复选框。

6. 单击“完成”(Finish) 关闭“下载结果”(Download results) 对话框。

结果

将完整的 PLC 程序下载到设备。 仅会删除设备中在线存在的块。 通过下载所有受影响的块并删除设备中不需要的所有块，可避免用户程序中的两个块之间出现不一致。

巡视窗口的“信息 > 常规”(Info > General) 下的消息将指示下载是否成功。

3.9 调试软件控制器

步骤

要打开工艺对象的“调试”(Commissioning) 工作区，请按以下步骤操作：

1. 在项目树中打开“工艺对象”(Technology objects) 文件夹。
2. 在项目树中打开该工艺对象。
3. 双击“调试”(Commissioning) 对象。

每个控制器都有特定的调试功能并对其进行了描述。

3.10 保存项目中优化的 PID 参数

软件控制器在 CPU 中进行优化。 这样，CPU 中的背景 DB 中的值与项目中对应的值不再一致。

要使用优化的 PID 参数更新项目中的 PID 参数，请按以下步骤操作：

要求

- 与 CPU 建立了在线连接，并且 CPU 处于“RUN”模式。
- 已通过“启动”(Start) 按钮启用了调试窗口的功能。

步骤

1. 在项目树中打开 CPU 文件夹。
2. 打开“工艺对象”(Technology objects) 文件夹。
3. 打开工艺对象。
4. 双击“调试”(Commissioning)。
5. 单击  图标“上传 PID 参数”(Upload PID parameters)。
6. 保存项目。

结果

当前激活的 PID 参数存储在项目数据中。 重新在 CPU 中加载项目数据时，将使用优化的参数。

3.11 显示工艺对象的背景 DB。

将为各工艺对象创建保存参数和静态变量的背景 DB。

步骤

要显示工艺对象的背景 DB，请按以下步骤操作：

1. 在项目树中打开 CPU 文件夹。
2. 打开“工艺对象”(Technology objects) 文件夹。
3. 突出显示工艺对象。
4. 在快捷菜单中，选择命令“打开 DB 编辑器”(Open DB editor)。

使用 PID_Compact

4

4.1 PID_Compact V2

4.1.1 组态 PID_Compact V2

4.1.1.1 基本设置 V2

简介 V2

在巡视窗口或组态窗口的“基本设置”(Basic settings) 下，组态工艺对象“PID_Compact”的以下属性：

- 物理量
- 控制逻辑
- 复位后的启动行为
- 设定值（仅在巡视窗口中）
- 过程值（仅在巡视窗口中）
- 输出值（仅在巡视窗口中）

设定值、过程值和输出值

只能在程序编辑器的巡视窗口中组态设定值、过程值和输出值。 为每个值选择一个源：

- 背景 DB

  使用背景数据块中保存的值。

  必须通过用户程序在背景 DB 中更新值。

  指令中不应有值。

  可通过 HMI 进行更改。

- 指令

  使用与指令相连的值。
  每次调用指令时都会将值写入背景数据块。

  无法通过 HMI 进行更改。

控制模式 V2

物理量

在“控制器类型”(Controller type) 组中，为设定值、过程值和扰动变量选择物理量和测量单位。 设定值、过程值和扰动变量以该测量单位显示。

控制逻辑

通常，可通过增大输出值来增大过程值。 这种做法称为常规控制逻辑。

PID_Compact 不使用负比例增益。 要在输出值增大时使过程值减小，请选中复选框“反转控制逻辑”(Invert control logic)。

示例

- 打开排泄阀将使容器盛装物的液位降低。
- 增加冷却能力将使温度降低。

启动特性

1. 要在 CPU 重启后切换到“未激活”模式，请清除“在 CPU 重启后激活模式”(Activate Mode after CPU restart) 复选框。

   要在 CPU 重启后切换到“模式”(Mode) 参数中保存的工作模式，请选中“在 CPU 重启后激活模式”(Activate Mode after CPU restart) 复选框。

2. 在“将模式设置为”(Set Mode to) 下拉列表中，选择要在完整下载到设备后启用的模式。

   完整下载到设备后，PID_Compact 以所选工作模式启动。 以后每次重启时，PID_Compact 都以上次保存在“模式”(Mode) 中的模式启动。

示例

您已选中“在 CPU 重启后激活模式”(Activate Mode after CPU restart) 复选框和“将模式设置为”(Set Mode to) 列表中的“预调节”(Pretuning) 条目。 完整下载到设备后，PID_Compact 以“预调节”(Pretuning) 模式启动。 如果预调节仍处于激活状态，则 PID_Compact 在 CPU 重启后再次以“预调节”(Pretuning) 模式启动。 如果预调节已成功完成并且自动模式处于激活状态，则 PID_Compact 在 CPU 重启后以“自动模式”(Automatic mode) 启动。

设定值 V2

步骤

要定义固定设定值，请按以下步骤操作：

1. 选择“背景 DB”(Instance DB)。
2. 输入一个设定值，例如 80° C。
3. 删除指令中的任何条目。

要定义可变设定值，请按以下步骤操作：

1. 选择“指令”(Instruction)。
2. 输入保存设定值的 REAL 变量的名称。

   可通过程序控制的方式来为该 REAL 变量分配变量值，例如，采用时间控制的方式来更改设定值。

过程值 V2

如果直接使用模拟量输入值，则 PID_Compact 会将该模拟量输入值标定为物理量。

如果要预先处理一下该模拟量输入值，则需要编写一个处理程序。 例如，过程值与模拟量输入值并不成正比。 经过处理的过程值必须为浮点格式。

步骤

要使用未经处理的模拟量输入值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input_PER”。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址。

要使用经过处理的浮点格式的过程值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input”。
2. 选择“指令”(Instruction) 作为源。
3. 输入变量的名称，用来保存经过处理的过程值。

输出值 V2

PID_Compact 提供三个输出值。 执行器将决定要使用的输出值。

- Output_PER

  通过模拟量输出触发执行器，使用连续信号（如 0...10V、4...20mA）进行控制。

- Output

  例如，由于执行器响应是非线性的，因而需要通过用户程序来处理输出值。

- Output_PWM

  通过数字量输出控制执行器。 脉宽调制可产生最短 ON 时间和最短 OFF 时间。

步骤

要使用模拟量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output_PER（模拟量）”(Output_PER (analog))。
2. 选择“指令”(Instruction)。
3. 输入模拟量输出的地址。

要使用用户程序来处理输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output”。
2. 选择“背景数据块”(Instance DB)。

   计算的输出值保存在背景数据块中。
3. 使用输出参数 Output 准备输出值。
4. 通过数字量或模拟量 CPU 输出将经过处理的输出值传送到执行器。

要使用数字量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output_PWM”。
2. 选择“指令”(Instruction)。
3. 输入数字量输出的地址。

4.1.1.2 过程值设置 V2

过程值标定 V2

如果已在基本设置中对 Input_PER 的使用进行了组态，则必须将模拟量输入值转换为过程值的物理量。 当前组态将显示在 Input_PER 画面中。

如果过程值与模拟量输入值成正比，则将使用上下限值对来标定 Input_PER。

步骤

要标定过程值，请按下列步骤操作：

1. 在“标定的过程值的下限”(Scaled low process value) 和“下限”(Low) 输入字段中输入一对下限值。
2. 在“标定的过程值的上限”(Scaled high process value) 和“上限”(High) 输入框中输入一对上限值。

这些值对的默认设置存储在硬件配置中。 要使用硬件配置中的值对，请按下列步骤操作：

1. 在程序编辑器中选择 PID_Compact 指令。
2. 在基本设置中将 Input_PER 与模拟量输入互连。
3. 在过程值设置中单击“自动设置”(Automatic setting) 按钮。

现有值将被硬件配置中的值覆盖。

过程值的限值 V2

必须为过程值指定正确的绝对上限和绝对下限，作为受控系统的限值。只要过程值超出这些限值，就会出现错误 (ErrorBits = 0001h)。 如果超出过程值的限值，则取消调节操作。可在输出值设置中组态 PID_Compact 如何在自动模式下对错误进行响应。

4.1.1.3 高级设置 V2

过程值监视 V2

在“过程值监视”(Process value monitoring) 组态窗口中，组态过程值的警告上限和下限。如果在运行期间超出或低于某一警告限值，则将在 PID_Compact 指令的以下参数中显示一条警告：

- 输出参数 InputWarning_H，前提是超出警告上限
- 输出参数 InputWarning_L，前提是低于警告下限

警告限值必须处于过程值的限值范围内。

如果未输入警告限值，将使用过程值的上限和下限。

示例

过程值上限 = 98 °C；警告上限 = 90 °C

警告下限 = 10 °C；过程值下限 = 0 °C

PID_Compact 将按如下方式响应：

过程值	InputWarning_H	InputWarning_L	Error Bits	工作模式
> 98 °C	TRUE	FALSE	0001 h	未激活或 带错误监视的替代输出值
≤ 98 °C 且 > 90 °C	TRUE	FALSE	0000 h	自动模式
≤ 90 °C 且 ≥ 10 °C	FALSE	FALSE	0000 h	自动模式
< 10 °C 且 ≥ 0 °C	FALSE	TRUE	0000 h	自动模式
< 0 °C	FALSE	TRUE	0001 h	未激活或 带错误监视的替代输出值

在输出值设置中，可以指定超出过程值上限或下限时 PID_Compact 的响应。

参见

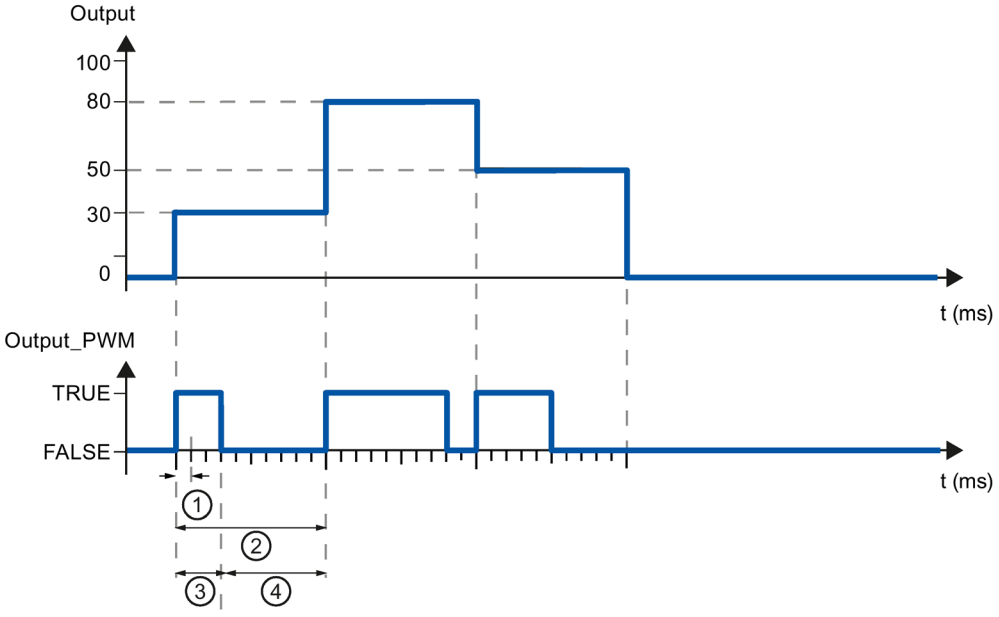
模式 V2 的参数状态 (页 296)

PWM 限值 V2

输出参数 Output 中的值被转换为一个脉冲序列，该序列通过脉宽调制在输出参数 Output_PWM 中输出。在 PID 算法采样时间内计算 Output，在采样时间 PID_Compact 内输出 Output_PWM。

在预调节或精确调节期间确定 PID 算法采样时间。 如果手动设置 PID 参数，则还需要组态 PID 算法采样时间。 PID_Compact 采样时间等于调用 OB 的周期时间。

脉冲宽度与 Output 中的值成比例并始终为 PID_Compact 采样时间的整数倍。



① PID_Compact 采样时间
② PID 算法采样时间
③ 脉冲持续时间
④ 中断时间

“最短开启时间”或“最短关闭时间”舍入为采样时间 PID_Compact 的整数倍。

脉冲或中断时间永远不会小于最短开关时间。 在下一个周期中累加和补偿由此引起的误差。

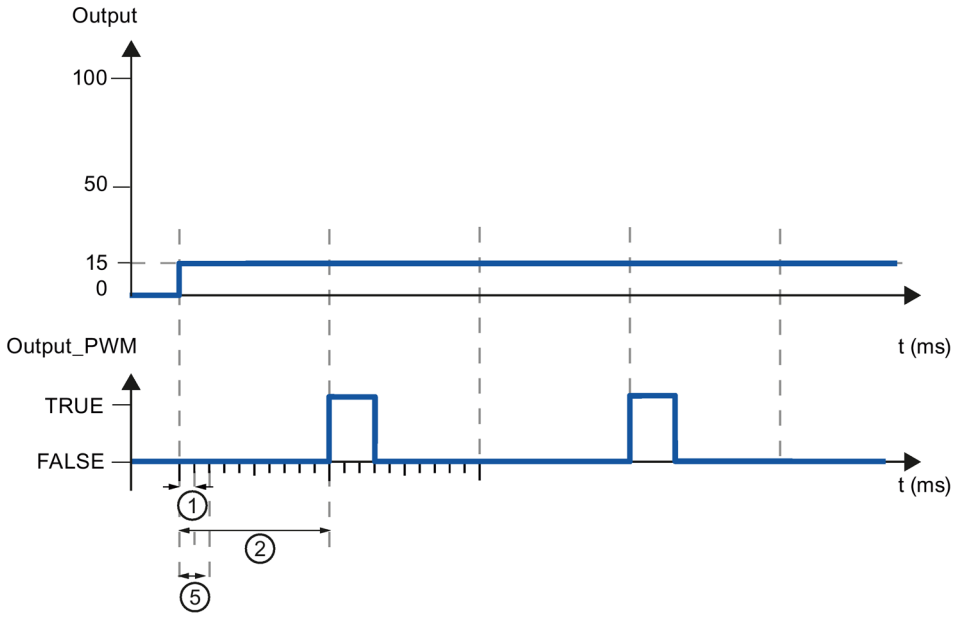
示例

PID_Compact 采样时间 = 100 ms

PID 算法采样时间 = 1000 ms

最短开启时间 = 200 ms

输出恒定为 15%。 可输出的最小脉冲为 PID_Compact 20%。 在第一个周期内不输出脉冲。 在第二个周期内，将第一个周期内未输出的脉冲累加到第二个周期的脉冲。



① PID_Compact 采样时间
② PID 算法采样时间
⑤ 最短 ON 时间

为最大程度地减小工作频率并节省执行器，可延长最短开关时间。

如果要使用“Output”或“Output_PER”，则必须分别为最短开关时间组态值 0.0。

说明

最短开关时间只影响输出参数 Output_PWM，不用于 CPU 中集成的任何脉冲发生器。

输出值 V2

输出值的限值

在“输出值的限值”组态窗口中，以百分比形式组态输出值的绝对限值。 无论是在手动模式还是自动模式下，都不要超过输出值的绝对限值。 如果在手动模式下指定了一个超出限值范围的输出值，则 CPU 会将有效值限制为组态的限值。

输出值限值必须与控制逻辑相匹配。

有效的输出值限值取决于所用的 Output。

Output	-100.0 至 100.0%
Output_PER	-100.0 至 100.0%
Output_PWM	0.0 至 100.0%

对错误的响应

注意
您的系统可能已损坏。 如果在出现错误时输出“错误未决时的当前值”或“错误未决时的替代输出值”，PID_Compact 将保持自动模式。 这可能导致超出过程值的限值并损坏系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

PID_Compact 需要预设置，以便在发生错误时，控制器在大多数情况下均可保持激活状态。 如果在控制器模式下频繁发生错误，则该默认响应会对控制响应产生负面影响。 这种情况下，检查 Errorbits 参数并消除错误原因。

PID_Compact 会生成可设定的输出值来对错误做出响应：

- 零（未激活）

  PID_Compact 针对所有错误都输出 0.0 作为输出值，然后切换到“未激活”模式。 只能通过 Reset 的下降沿或 ModeActivate 的上升沿重新激活控制器。

- 错误未决时的当前值

  如果在**自动模式**下发生以下错误，则只要这些错误不再处于未决状态，PID_Compact 便会返回自动模式。

  如果发生一个或多个下列错误，则 PID_Compact 停留在
  自动模式下：

  – 0001h： 参数“Input”超出了过程值限值的范围。

  – 0800h： 采样时间错误

  – 40000h：Disturbance 参数的值无效。

  如果在**自动模式**下发生一个或多个下列错误，PID_Compact 将切换到“带错误监视的替代输出值”模式并输出最后一个有效输出值：

  – 0002h： Input_PER 参数的值无效。

  – 0200h： Input 参数的值无效。

  – 0400h： 输出值计算失败。

  – 1000h： Setpoint 参数的值无效。

  如果在**手动模式**下发生错误，PID_Compact 将继续使用手动值作为输出值。 如果手动值无效，则使用替代输出值。 如果手动值无效和替代输出值都无效，则使用输出值下限。

  如果在**预调节或精确调节**期间出现下列错误，PID_Compact 将保持激活模式。

  – 0020h： 精确调节期间不允许预调节。

  出现其它错误时，PID_Compact 将取消调节并切换到调节开始时的模式。

  只要错误不再处于未决状态，PID_Compact 就会返回自动模式。

- 错误未决时的替代输出值

  PID_Compact 将输出替代输出值。

  如果发生下列错误，PID_Compact 将保持“带错误监视的替代输出值”模式，并输出输出值下限：

  – 20000h： 变量 SubstituteOutput 的值无效。

  对于所有其它错误，PID_Compact 按照“错误未决时的当前值”中的描述进行响应。

参见

模式 V2 的参数状态 (页 296)

PID 参数 V2

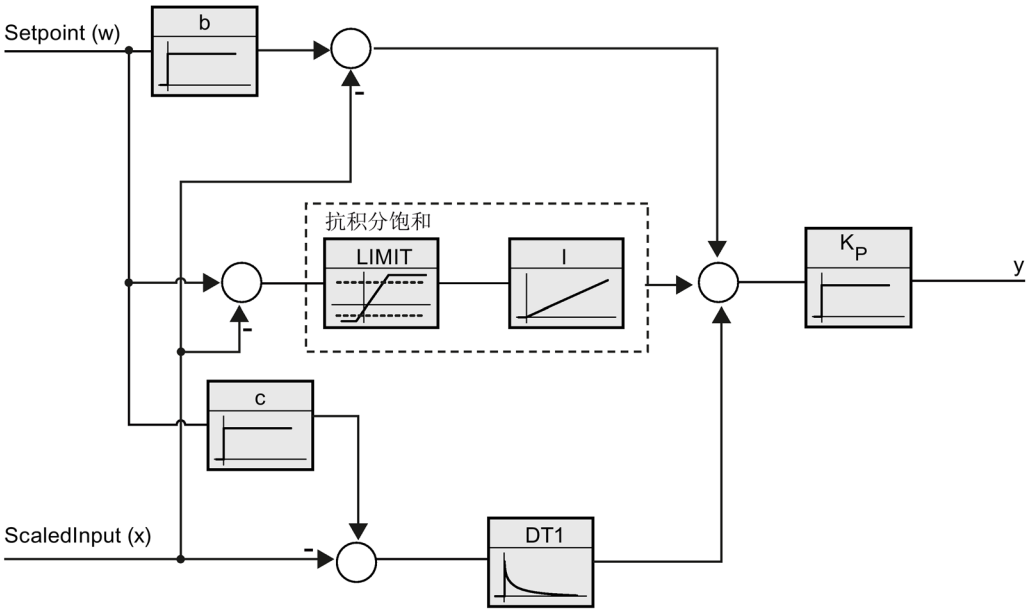
PID 参数显示在“PID 参数”(PID Parameters) 组态窗口中。 在控制器调节期间将调整 PID 参数以适应受控系统。 用户不必手动输入 PID 参数。

PID 算法根据以下等式工作：

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
y	PID 算法的输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分作用时间
a	微分延迟系数（微分延迟 T1 = a × T_D）
T_D	微分作用时间
c	微分作用权重

下图说明了集成到 PID 算法中的参数：



所有 PID 参数均具有保持性。 如果手动输入 PID 参数，则必须完整下载 PID_Compact。

将工艺对象下载到设备 (页 76)

比例增益

该值用于指定控制器的比例增益。 PID_Compact 不使用负比例增益。 在“基本设置 > 控制器类型”下，控制逻辑会反转。

积分作用时间

积分作用时间用于确定积分作用的时间特性。 积分作用时间 = 0.0 时，将禁用积分作用。

微分作用时间

微分作用时间用于确定微分作用的时间特性。 微分作用时间 = 0.0 时，将禁用微分作用。

微分延迟系数

微分延迟系数用于延迟微分作用的生效。

微分延迟 = 微分作用时间 × 微分延迟系数

- 0.0: 微分作用仅在一个周期内有效，因此几乎不产生影响。
- 0.5: 此值经实践证明对于具有一个优先时间常量的受控系统非常有用。
- > 1.0: 系数越大，微分作用的生效时间延迟越久。

比例作用权重

比例作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0: 应对设定值变化的比例作用完全有效
- 0.0: 应对设定值变化的比例作用无效

当过程值变化时，比例作用始终完全有效。

微分作用权重

微分作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0: 设定值变化时微分作用完全有效
- 0.0: 设定值变化时微分作用不生效

当过程值变化时，微分作用始终完全有效。

PID 算法采样时间

受控系统需要一定的时间来对输出值的变化做出响应。 因此，建议不要在每次循环中都计算输出值。 PID 算法的采样时间是两次计算输出值之间的时间。 该时间在调节期间进行计算，并舍入为循环时间的倍数。 PID_Compact 的所有其它功能会在每次调用时执行。

如果使用 Output_PWM，则输出信号的精度由 PID 算法采样时间与 OB 的周期时间的比率确定。PID 算法采样时间对应于脉宽调制的时间周期。 该周期时间至少应为 PID 算法采样时间的 10 倍。

调节的规则

在“控制器结构”(Controller structure) 下拉列表中选择要计算 PI 还是 PID 参数。

- **PID**

  预调节和精确调节期间计算 PID 参数。

- **PI**

  预调节和精确调节期间计算 PI 参数。

- **用户自定义**

  如果通过用户程序为预调节和精确调节组态了不同的控制器结构，则下拉列表会显示“用户自定义”(User-defined)。

4.1.2 调试 PID_Compact V2

4.1.2.1 预调节 V2

预调节功能可确定对输出值跳变的过程响应，并搜索拐点。 根据受控系统的最大上升速率与死时间计算 PID 参数。 可在执行预调节和精确调节时获得最佳 PID 参数。

过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。最可能的情况是处于工作模式“未激活”和“手动模式”下。 重新计算前会备份 PID 参数。

要求

- 已在循环中断 OB 中调用“PID_Compact”指令。
- ManualEnable = FALSE
- Reset = FALSE
- PID_Compact 处于下列模式之一：“未激活”、“手动模式”或“自动模式”。
- 设定值和过程值均处于组态的限值范围内（请参见“过程值监视”组态）。
- 设定值与过程值的差值大于过程值上限与过程值下限之差的 30%。
- 设定值与过程值的差值大于设定值的 50%。

步骤

要执行预调节，请按下列步骤操作：

1. 在项目树中双击“PID_Compact > 调试”(PID_Compact > Commissioning) 条目。
2. 在“调节模式”(Tuning mode) 下拉列表中选择条目“预调节”(Pretuning)。
3. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动预调节功能。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。 进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及控制器调节功能看似受阻时，请单击“Stop”图标。检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果执行预调节时未产生错误消息，则 PID 参数已调节完毕。PID_Compact 将切换到自动模式并使用已调节的参数。 在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果无法实现预调节，PID_Compact 将根据已组态的响应对错误作出反应。

参见

模式 V2 的参数状态 (页 296)

4.1.2.2 精确调节 V2

精确调节将使过程值出现恒定受限的振荡。 将根据此振荡的幅度和频率为操作点调节 PID 参数。 所有 PID 参数都根据结果重新计算。精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。 可在执行预调节和精确调节时获得最佳 PID 参数。

PID_Compact 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。 重新计算前会备份 PID 参数。

要求

- 已在循环中断 OB 中调用 PID_Compact 指令。
- ManualEnable = FALSE
- Reset = FALSE
- 设定值和过程值均在组态的限值范围内。
- 在操作点处，控制回路已稳定。 过程值与设定值一致时，表明到达了操作点。
- 不能被干扰。
- PID_Compact 处于下列工作模式之一： 未激活、自动模式或手动模式。

过程取决于初始情况

可在以下工作模式下启动精确调节： “未激活”、“自动模式”或“手动模式”。 在以下模式下启动精确调节时，具体情况如下所述：

- 自动模式

  如果希望通过调节来改进现有 PID 参数，请在自动模式下启动精确调节。

  PID_Compact 将使用现有的 PID 参数控制系统，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。

- 未激活模式或手动模式

  如果满足预调节的要求，则启动预调节。 已确定的 PID 参数将用于控制，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。如果无法实现预调节，PID_Compact 将根据已组态的响应对错误作出反应。

  如果预调节的过程值已经十分接近设定值，则将尝试利用最小或最大输出值来达到设定值。 这可能会增加超调量。

步骤

要执行精确调节，请按下列步骤操作：

1. 在“调节模式”(Tuning mode) 下拉列表中选择条目“精确调节”(Fine tuning)。
2. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动精确调节过程。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。 进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及调节功能看似受阻时，请单击“调节模式”(Tuning mode) 组中的“Stop”图标。检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果在精确调节期间未发生错误，则 PID 参数已调节完毕。PID_Compact 将切换到自动模式并使用已调节的参数。 在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果在“精确调节”期间出现错误，PID_Compact 将根据已组态的响应对错误作出反应。

参见

模式 V2 的参数状态 (页 296)

4.1.2.3 “手动”模式 V1

以下部分将说明如何在“PID_Compact”工艺对象的调试窗口中使用“手动模式”工作模式。错误未决时也可使用手动模式。

要求

- 已在循环中断 OB 中调用“PID_Compact”指令。
- 与 CPU 建立了在线连接，并且 CPU 处于“RUN”模式。

步骤

如果要通过指定手动值来测试受控系统，请使用调试窗口中的“手动模式”。 要定义手动值，请按以下步骤操作：

1. 单击“Start”图标。
2. 在“控制器的在线状态”(Online status of the controller) 区域中，选中复选框“手动模式”(Manual mode)。

   PID_Compact 将在手动模式下运行。 最新的当前输出值仍然有效。
3. 在“输出”(Output) 字段中，输入 % 形式的手动值。
4. 单击 图标。

结果

手动值被写入 CPU 并立即生效。

如果希望 PID 控制器重新指定输出值，请清除“手动模式”(Manual mode) 复选框。 到自动模式的切换是无扰动的。

参见

模式 V2 的参数状态 (页 296)

4.1.3 通过 PID_Compact V2 进行超驰控制

超驰控制

超驰控制时，两个或多个控制器共享一个执行器。只有一个控制器可以随时访问执行器并影响过程。

由逻辑运算决定可以访问执行器的控制器。通常根据所有控制器的输出值比较结果做出此决定（例如，进行最大选择时），具有最大输出值的控制器将获得对执行器的访问权限。

基于输出值的选择要求所有控制器均在自动模式下工作。对不影响执行器的控制器进行更新。为防止饱和效应及其对控制响应和控制器之间的切换产生负面影响，这很有必要。

自版本 V2.3 起，PID_Compact 通过提供一个用于更新未激活控制器的简单过程，支持超驰控制：通过使用变量 OverwriteInitialOutputValue 和 PIDCtrl.PIDInit，可以预分配自动模式下控制器的积分作用，好像在上一周期中 PID 算法已计算输出值的 Output = OverwriteInititalOutputValue。为此，OverwriteInitialOutputValue 与当前可以访问执行器的控制器的输出值互连。通过设置位 PIDCtrl.PIDInit，触发积分作用的预分配以及控制器循环和 PWM 周期的重启。根据预分配的（并针对所有控制器同步的）积分作用，以及当前控制偏差的比例作用与积分作用，在当前循环中进行输出值的后续计算。通过 PIDCtrl.PIDInit = TRUE 调用期间，微分作用未激活，因此对输出值不起作用。

此过程可以确保仅根据当前的过程状态和 PI 参数对当前输出值进行计算，并从而决定可以访问执行器的控制器。可防止未激活控制器的饱和效应，并因此防止切换逻辑的错误决定。

要求

- PIDCtrl.PIDInit 仅在积分作用激活（Retain.CtrlParams.Ti 变量 > 0.0）时有效。
- 您必须在用户程序中自行分配 PIDCtrl.PIDInit 和 OverwriteInitialOutputValue（请参见下面的示例）。PID_Compact 不会自动更改这些变量。
- 仅当 PID_Compact 处于自动模式（参数 State = 3）时，PIDCtrl.PIDInit 才有效。
- 如果可能，选择 PID 算法的采样时间（Retain.CtrlParams.Cycle 变量）时，应使其对所有控制器均相同，并在同一循环中断 OB 中调用所有控制器。这样，可以确保在一个控制器循环或 PWM 周期内不发生切换。

说明

不断调整输出值限制

也可以通过在其它控制器系统中不断调整输出值限制实现这一操作，而不是如此处所述对没有执行器访问权的控制器进行主动更新。

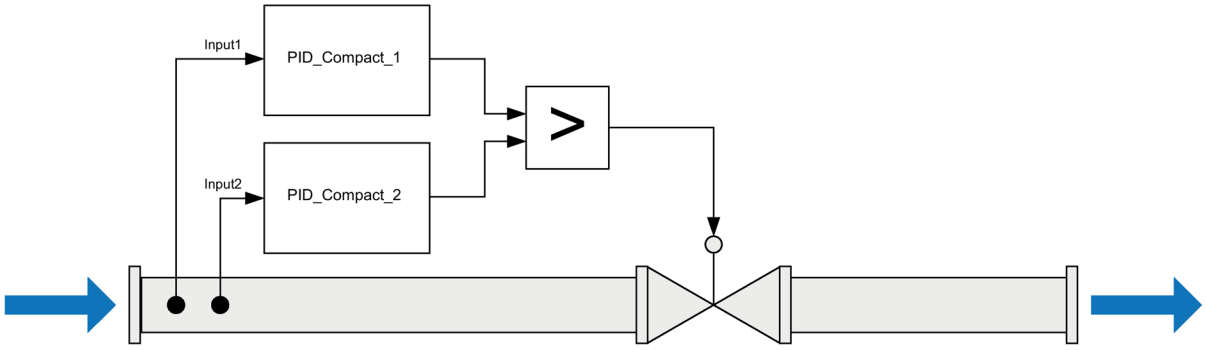
无法使用 PID_Compact 实现这一操作，因为在自动模式下不支持更改输出值限制。

示例：煤气管道的控制

PID_Compact 用于控制煤气管道。

主要目标是控制流速 Input1。为此使用控制器 PID_Compact_1。此外，使用限制控制器 PID_Compact_2 将压力 Input2（在阀前方沿流动方向测量）保持在上限以下。

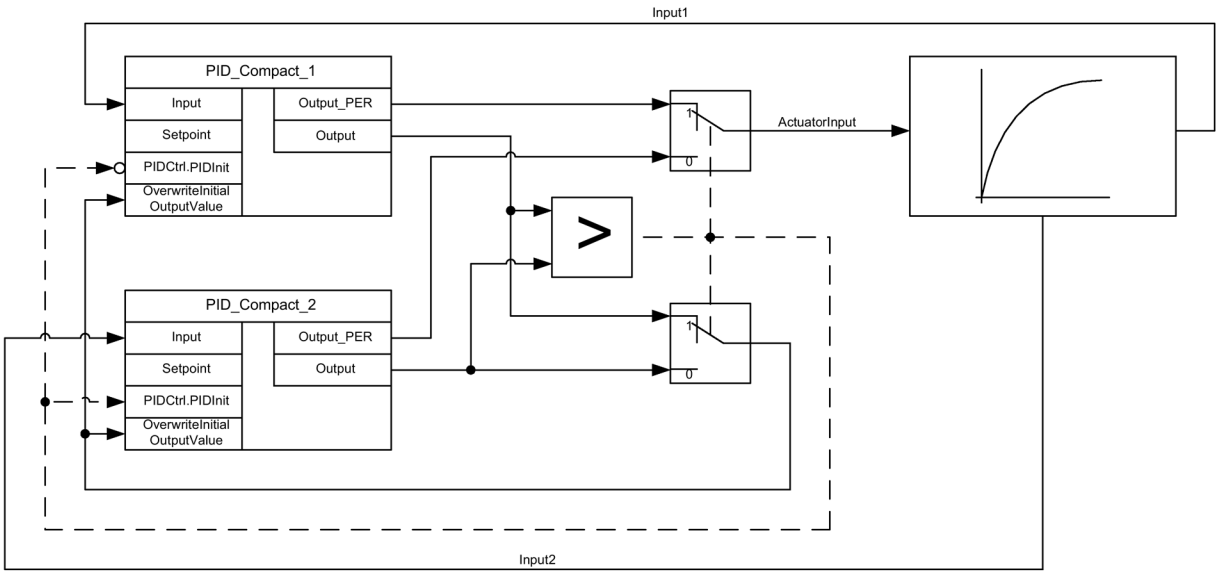
通过一个电磁阀控制流速和压力。控制器的输出值与阀门的打开相对应：输出值增加时阀门打开。这意味着压力下降（反转控制逻辑）时，流速增大（正常控制逻辑）。



通过编写程序变量 ActuatorInput，借助 I/O 格式的 PID_Compact 的输出值（参数 Output_PER）控制阀门。

在 PID_Compact_1.Setpoint 参数中指定流速的设定值。

在 PID_Compact_2.Setpoint 参数中将压力上限指定为设定值。



两个控制器必须共享一个阀门作为共享的执行器。在这种情况下，通过输出值（采用实数格式，参数 Output）的最大选择实现逻辑，该逻辑决定哪个控制器获得执行器的访问权。因为输出值与阀门的打开程度相对应，所以需要阀门打开较大程度的控制器将获得控制权。

说明

激活控制逻辑的反转

输出值增加（阀门打开）时，需要通过压力调节器 PID_Compact_2 来实现实际值（压力）的降低，因此必须激活控制逻辑的反转：PID_Compact_2.Config.InvertControl = TRUE。

设备正常运行时，流速的实际值与设定值相对应。流量控制器 PID_Compact_1 已稳定在固定的输出值 PID_Compact_1.Output。正常操作过程中，压力的实际值显著低于指定为 PID_Compact_2 设定值的上限。因此，压力调节器要进一步关闭阀门以增加压力，即它将计算一个输出值 PID_Compact_2.Output，该输出值小于流量控制器 PID_Compact_1.Output 的输出值。切换逻辑的最大选择从而使得流量控制器 PID_Compact_1 可以继续访问执行器。此外，确保通过赋值 PID_Compact_2.OverwriteInitialOutputValue = PID_Compact_1.Output 以及 PID_Compact_2.PIDCtrl.PIDInit = TRUE 来更新 PID_Compact_2。

如果由于故障等原因压力现在接近或超过上限，压力调节器 PID_Compact_2 将计算一个更大的输出值以进一步打开阀门，从而降低压力。如果 PID_Compact_2.Output 大于 PID_Compact_1.Output，则压力调节器 PID_Compact_2 通过最大选择获得执行器访问权并将其打开。确保通过赋值 PID_Compact_1.OverwriteInitialOutputValue = PID_Compact_2.Output 以及 PID_Compact_1.PIDCtrl.PIDInit = TRUE 来更新 PID_Compact_1。

流速增加时压力降低，且压力不再保持在设定值。

解决故障后，压力将继续下降，并通过压力调节器降低阀门的打开程度。如果流量控制器计算更大的打开程度作为输出值，则设备将恢复正常操作，使流量控制器 PID_Compact_1 再次获得对执行器的访问权限。

可以通过以下 SCL 程序代码实现此示例：

```
"PID_Compact_1"(Input := "Input1");
"PID_Compact_2"(Input := "Input2");
IF "PID_Compact_1".Output >= "PID_Compact_2".Output THEN
  "ActuatorInput" := "PID_Compact_1".Output_PER;
  "PID_Compact_1".PIDCtrl.PIDInit := FALSE;
  "PID_Compact_2".PIDCtrl.PIDInit := TRUE;
  "PID_Compact_2".OverwriteInitialOutputValue := "PID_Compact_1".Output;
ELSE
  "ActuatorInput" := "PID_Compact_2".Output_PER;
  "PID_Compact_1".PIDCtrl.PIDInit := TRUE;
  "PID_Compact_2".PIDCtrl.PIDInit := FALSE;
  "PID_Compact_1".OverwriteInitialOutputValue := "PID_Compact_2".Output;
END_IF;
```

4.1.4 使用 PLCSIM 仿真 PID_Compact V2

说明

使用 PLCSIM 进行仿真

不支持使用 PLCSIM 仿真 PID_Compact V2.x 后将其用于 CPU S7-1200。

PID_Compact V2.x 只能通过 PLCSIM 仿真后用于 CPU S7-1500。

对于使用 PLCSIM 进行的仿真，仿真 PLC 的时间特性与“真实”PLC 并不完全相同。仿真 PLC 循环中断 OB 的实际周期时钟波动比“真实”PLC 的波动大。

在标准组态中，PID_Compact 会自动确定调用之间的时间，并监视波动情况。

因此，使用 PLCSIM 仿真 PID_Compact 时，可能检测到采样时间错误 (ErrorBits = DW#16#00000800)。

这会导致进行中的调节中止。

自动模式下的响应取决于 ActivateRecoverMode 变量的值。

为防止此类情况发生，应按下列方式为使用 PLCSIM 进行的仿真组态 PID_Compact：

- CycleTime.EnEstimation = FALSE
- CycleTime.EnMonitoring = FALSE
- CycleTime.Value：以秒为单位为此变量分配调用循环中断 OB 的周期时钟。

4.2 PID_Compact V1

4.2.1 组态 PID_Compact V1

4.2.1.1 基本设置 V1

简介 V1

在巡视窗口或组态窗口的“基本设置”(Basic settings) 下，组态工艺对象“PID_Compact”的以下属性：

- 物理量
- 控制逻辑
- 复位后的启动行为
- 设定值（仅在巡视窗口中）
- 过程值（仅在巡视窗口中）
- 输出值（仅在巡视窗口中）

设定值、过程值和输出值

只能在程序编辑器的巡视窗口中组态设定值、过程值和输出值。 为每个值选择一个源：

- 背景 DB

  使用背景数据块中保存的值。

  必须通过用户程序在背景 DB 中更新值。

  指令中不应有值。

  可通过 HMI 进行更改。

- 指令

  使用与指令相连的值。
  每次调用指令时都会将值写入背景数据块。

  无法通过 HMI 进行更改。

控制模式 V1

物理量

在“控制器类型”(Controller type) 组中，为设定值和过程值选择测量单位和物理量。 设定值和过程值将以该测量单位显示。

控制逻辑

通常，可通过增大输出值来增大过程值。 这种做法称为常规控制逻辑。

PID_Compact 不使用负比例增益。 要在输出值增大时使过程值减小，请选中复选框“反转控制逻辑”(Invert control logic)。

示例

- 打开排泄阀将使容器盛装物的液位降低。
- 增加冷却能力将使温度降低。

复位后的启动行为

要在重启 CPU 后直接切换到上次激活的模式，请选中“CPU 重启后启用上一模式”(Enable last mode after CPU restart) 复选框。

如果清除该复选框，PID_Compact 将保持在“未激活”模式。

设定值 V1

步骤

要定义固定设定值，请按以下步骤操作：

1. 选择“背景 DB”(Instance DB)。
2. 输入一个设定值，例如 80° C。
3. 删除指令中的任何条目。

要定义可变设定值，请按以下步骤操作：

1. 选择“指令”(Instruction)。
2. 输入保存设定值的 REAL 变量的名称。

   可通过程序控制的方式来为该 REAL 变量分配变量值，例如，采用时间控制的方式来更改设定值。

过程值 V1

如果直接使用模拟量输入值，则 PID_Compact 会将该模拟量输入值标定为物理量。

如果要预先处理一下该模拟量输入值，则需要编写一个处理程序。 例如，过程值与模拟量输入值并不成正比。 经过处理的过程值必须为浮点格式。

步骤

要使用未经处理的模拟量输入值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input_PER”。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址。

要使用经过处理的浮点格式的过程值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input”。
2. 选择“指令”(Instruction) 作为源。
3. 输入变量的名称，用来保存经过处理的过程值。

输出值 V1

PID_Compact 提供三个输出值。 执行器将决定要使用的输出值。

- Output_PER

  通过模拟量输出触发执行器，使用连续信号（如 0...10V、4...20mA）进行控制。

- Output

  例如，由于执行器响应是非线性的，因而需要通过用户程序来处理输出值。

- Output_PWM

  通过数字量输出控制执行器。 脉宽调制可产生最短 ON 时间和最短 OFF 时间。

步骤

要使用模拟量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output_PER（模拟量）”(Output_PER (analog))。
2. 选择“指令”(Instruction)。
3. 输入模拟量输出的地址。

要使用用户程序来处理输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output”。
2. 选择“背景数据块”(Instance DB)。

   计算的输出值保存在背景数据块中。
3. 使用输出参数 Output 准备输出值。
4. 通过数字量或模拟量 CPU 输出将经过处理的输出值传送到执行器。

要使用数字量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output_PWM”。
2. 选择“指令”(Instruction)。
3. 输入数字量输出的地址。

4.2.1.2 过程值设置 V1

在“过程值设置”(Process value settings) 组态窗口中，组态过程值的标定并指定过程值的绝对限值。

标定过程值

如果已在基本设置中对 Input_PER 的使用进行了组态，则需要将模拟量输入值转换为过程值的物理量。 当前组态将显示在 Input_PER 画面中。

如果过程值与模拟量输入值成正比，则将使用上下限值对来标定 Input_PER。

1. 在“标定的过程值的下限”(Scaled low process value) 和“下限”(Low) 输入字段中输入一对下限值。
2. 在“标定的过程值的上限”(Scaled high process value) 和“上限”(High) 输入框中输入一对上限值。

这些值对的默认设置保存在硬件配置中。 要使用硬件配置中的值对，请按以下步骤操作：

1. 在程序编辑器中选择指令 PID_Compact。
2. 在基本设置中连接 Input_PER 与模拟量输入。
3. 在过程值设置中单击“自动设置”(Automatic setting) 按钮。

现有值将被硬件配置中的值覆盖。

监视过程值

指定过程值的绝对上限和下限。 只要在运行期间超出这些限值，控制器就会关闭，同时输出值设置为 0%。 必须为受控系统输入合理的限值。 合理的限值在获取最优 PID 参数的优化过程中是重要的。

“过程值的上限”的默认值是 120 %。在 I/O 输入中，过程值最大可超出标准范围 18%（过范围）。 如果超出“过程值的上限”，将不再报告错误。 仅识别断线和短路，然后 PID_Compact 切换到“未激活”模式。

⚠警告
如果将过程值的限值范围设置得非常大（例如 $-3.4*10^{38}...+3.4*10^{38}$），则将禁用过程值监视功能。 如果发生错误，则可能损坏系统。

参见

过程值监视 V1 (页 110)

PWM 限值 V1 (页 111)

输出值的限值 V1 (页 113)

PID 参数 V1 (页 114)

4.2.1.3 高级设置 V1

过程值监视 V1

在“过程值监视”(Process value monitoring) 组态窗口中，组态过程值的警告上限和下限。如果在运行期间超出或低于某一警告限值，则将在 PID_Compact 指令的以下参数中显示一条警告：

- 输出参数 InputWarning_H，前提是超出警告上限
- 输出参数 InputWarning_L，前提是低于警告下限

警告限值必须处于过程值的限值范围内。

如果未输入警告限值，将使用过程值的上限和下限。

示例

过程值上限 = 98° C；警告上限 = 90° C

警告下限 = 10° C；过程值下限 = 0° C

PID_Compact 将按如下方式响应：

过程值	InputWarning_H	InputWarning_L	工作模式
> 98° C	TRUE	FALSE	未激活
≤ 98° C 且 > 90° C	TRUE	FALSE	自动模式
≤ 90° C 且 ≥ 10° C	FALSE	FALSE	自动模式
< 10° C 且 ≥ 0° C	FALSE	TRUE	自动模式
< 0° C	FALSE	TRUE	未激活

参见

过程值设置 V1 (页 108)

PWM 限值 V1 (页 111)

输出值的限值 V1 (页 113)

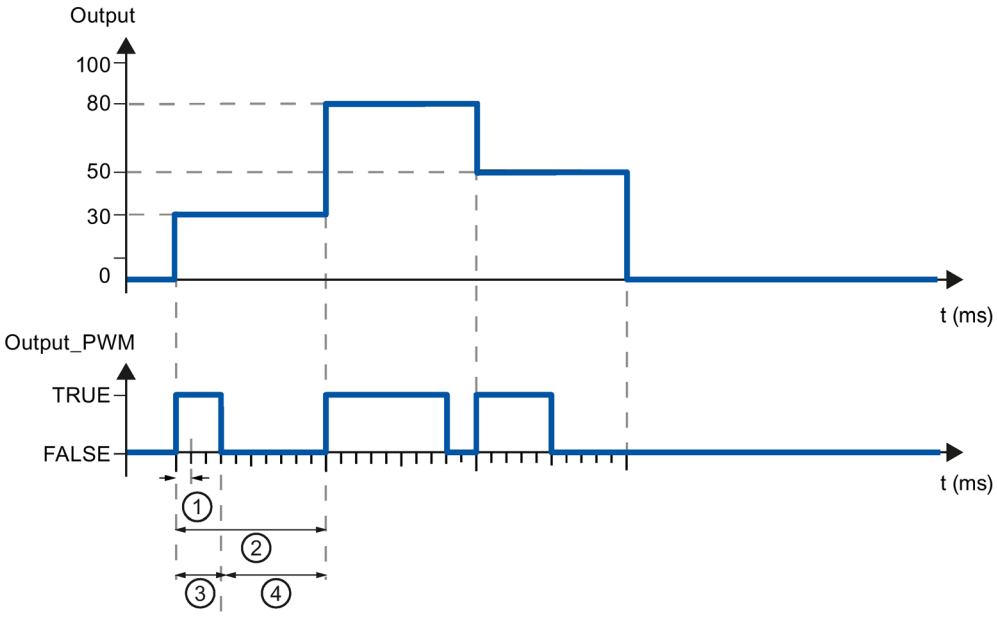
PID 参数 V1 (页 114)

PWM 限值 V1

输出参数 Output 中的值被转换为一个脉冲序列，该序列通过脉宽调制在输出参数 Output_PWM 中输出。在 PID 算法采样时间内计算 Output，在采样时间 PID_Compact 内输出 Output_PWM。

在预调节或精确调节期间确定 PID 算法采样时间。 如果手动设置 PID 参数，则还需要组态 PID 算法采样时间。 PID_Compact 采样时间等于调用 OB 的周期时间。

脉冲宽度与 Output 中的值成比例并始终为 PID_Compact 采样时间的整数倍。



① PID_Compact 采样时间
② PID 算法采样时间
③ 脉冲持续时间
④ 中断时间

“最短开启时间”或“最短关闭时间”舍入为采样时间 PID_Compact 的整数倍。

脉冲或中断时间永远不会小于最短开关时间。 在下一个周期中累加和补偿由此引起的误差。

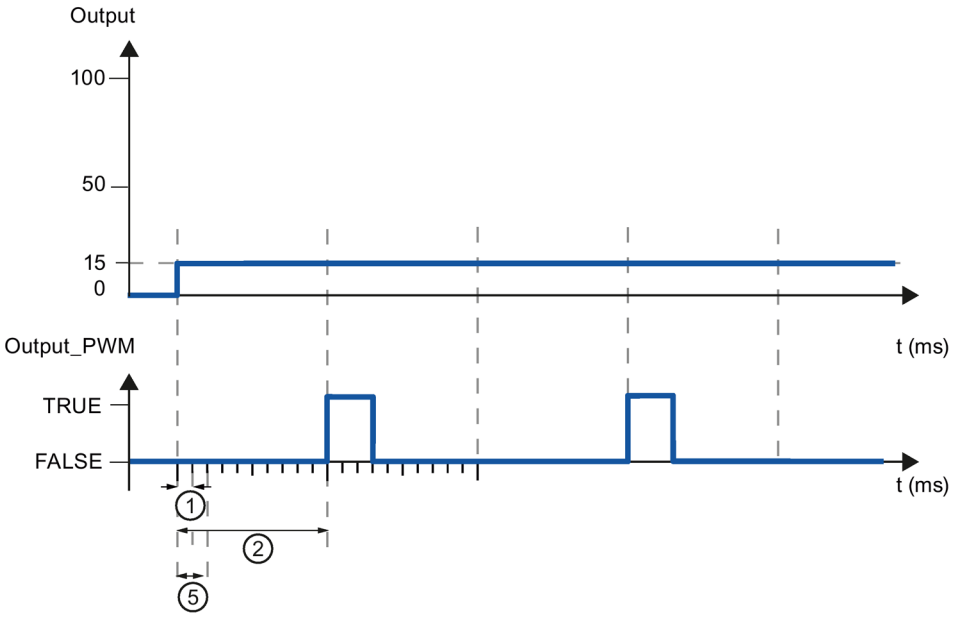
示例

PID_Compact 采样时间 = 100 ms

PID 算法采样时间 = 1000 ms

最短开启时间 = 200 ms

输出恒定为 15%。 可输出的最小脉冲为 PID_Compact 20%。 在第一个周期内不输出脉冲。 在第二个周期内，将第一个周期内未输出的脉冲累加到第二个周期的脉冲。



① PID_Compact 采样时间
② PID 算法采样时间
⑤ 最短 ON 时间

为最大程度地减小工作频率并节省执行器，可延长最短开关时间。

如果要使用“Output”或“Output_PER”，则必须分别为最短开关时间组态值 0.0。

说明

最短开关时间只影响输出参数 Output_PWM，不用于 CPU 中集成的任何脉冲发生器。

参见

过程值设置 V1 (页 108)

过程值监视 V1 (页 110)

输出值的限值 V1 (页 113)

PID 参数 V1 (页 114)

输出值的限值 V1

在“输出值的限值”组态窗口中，以百分比形式组态输出值的绝对限值。 无论是在手动模式还是自动模式下，都不要超过输出值的绝对限值。 如果在手动模式下指定了一个超出限值范围的输出值，则 CPU 会将有效值限制为组态的限值。

有效的输出值限值取决于所用的 Output。

Output	-100.0 到 100.0
Output_PER	-100.0 到 100.0
Output_PWM	0.0 到 100.0

如果发生错误，则 PID_Compact 会将输出值设置为 0.0。 因此，0.0 必须始终处于输出值的限值范围内。 如果要使输出值下限大于 0.0，则需要在用户程序中为 Output 和 Output_PER 增加一个偏移量。

参见

过程值设置 V1 (页 108)

过程值监视 V1 (页 110)

PWM 限值 V1 (页 111)

PID 参数 V1 (页 114)

PID 参数 V1

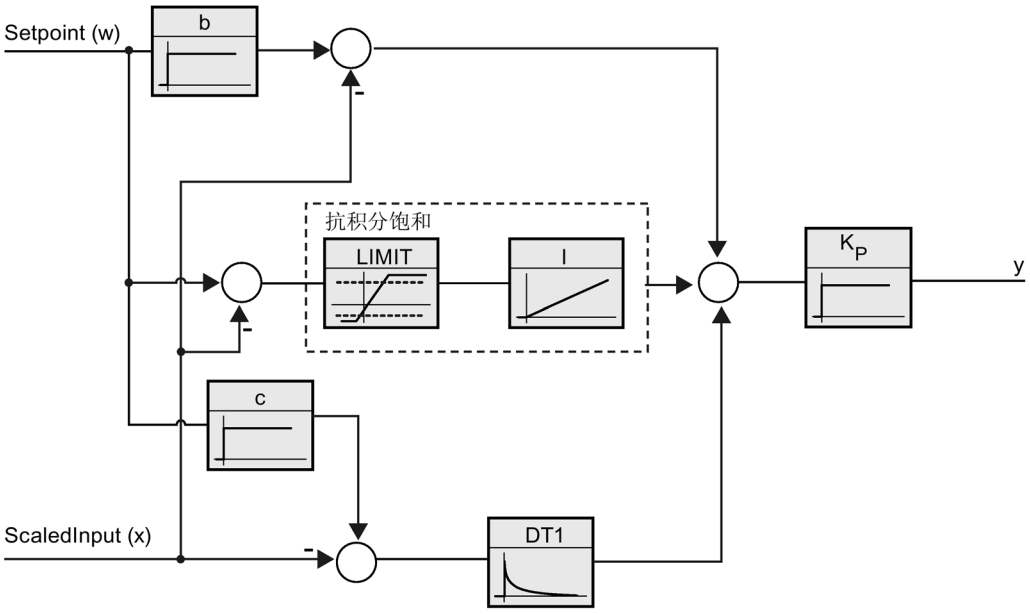
PID 参数显示在“PID 参数”(PID Parameters) 组态窗口中。 在控制器调节期间将调整 PID 参数以适应受控系统。 用户不必手动输入 PID 参数。

PID 算法根据以下等式工作：

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
y	PID 算法的输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分作用时间
a	微分延迟系数（微分延迟 T1 = a × T_D）
T_D	微分作用时间
c	微分作用权重

下图说明了集成到 PID 算法中的参数：



所有 PID 参数均具有保持性。 如果手动输入 PID 参数，则必须完整下载 PID_Compact。

Auto-Hotspot

比例增益

该值用于指定控制器的比例增益。 PID_Compact 不使用负比例增益。 在“基本设置 > 控制器类型”下，控制逻辑会反转。

积分作用时间

积分作用时间用于确定积分作用的时间特性。 积分作用时间 = 0.0 时，将禁用积分作用。

微分作用时间

微分作用时间用于确定微分作用的时间特性。 微分作用时间 = 0.0 时，将禁用微分作用。

微分延迟系数

微分延迟系数用于延迟微分作用的生效。

微分延迟 = 微分作用时间 × 微分延迟系数

- 0.0: 微分作用仅在一个周期内有效，因此几乎不产生影响。
- 0.5: 此值经实践证明对于具有一个优先时间常量的受控系统非常有用。
- > 1.0: 系数越大，微分作用的生效时间延迟越久。

比例作用权重

比例作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0: 应对设定值变化的比例作用完全有效
- 0.0: 应对设定值变化的比例作用无效

当过程值变化时，比例作用始终完全有效。

微分作用权重

微分作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0: 设定值变化时微分作用完全有效
- 0.0: 设定值变化时微分作用不生效

当过程值变化时，微分作用始终完全有效。

PID 算法采样时间

受控系统需要一定的时间来对输出值的变化做出响应。 因此，建议不要在每次循环中都计算输出值。 PID 算法的采样时间是两次计算输出值之间的时间。 该时间在调节期间进行计算，并舍入为循环时间的倍数。 PID_Compact 的所有其它功能会在每次调用时执行。

如果使用 Output_PWM，则输出信号的精度由 PID 算法采样时间与 OB 的周期时间的比率确定。PID 算法采样时间对应于脉宽调制的时间周期。 该周期时间至少应为 PID 算法采样时间的 10 倍。

调节的规则

在“控制器结构”(Controller structure) 下拉列表中选择要计算 PI 还是 PID 参数。

- **PID**

  预调节和精确调节期间计算 PID 参数。

- **PI**

  预调节和精确调节期间计算 PI 参数。

- **用户自定义**

  如果通过用户程序为预调节和精确调节组态了不同的控制器结构，则下拉列表会显示“用户自定义”(User-defined)。

参见

将工艺对象下载到设备 (页 76)

4.2.2 调试 PID_Compact V1

4.2.2.1 调试 V1

调试窗口有助于您调试 PID 控制器。 可以在趋势视图中监视设定值、过程值以及输出值随时间轴的变化。 调试窗口支持以下功能：

- 控制器预调节
- 控制器精确调节

  使用精确调节对 PID 参数进行精确调节。
- 在趋势视图中监视当前闭环控制
- 通过指定手动输出值测试受控系统

所有功能均要求已与 CPU 建立在线连接。

基本处理操作

- 在“采样时间”(Sampling time) 下拉列表中，选择所需的采样时间。

  调试窗口中的所有值将以所选的更新时间进行更新。
- 如果要使用调试功能，请单击测量组中的“启动”(Start) 图标。

  将启动值记录操作。 设定值、过程值以及输出值的当前值将输入到趋势视图中。 可以对调试窗口进行操作。
- 如果要结束调试功能，请单击“停止”(Stop) 图标。

  可以继续对趋势视图中记录的值进行分析。

关闭调试窗口将终止趋势视图中的记录操作并删除所记录的值。

参见

预调节 V1 (页 119)

精确调节 V1 (页 121)

“手动”模式 V1 (页 123)

4.2.2.2 预调节 V1

预调节功能可确定对输出值跳变的过程响应，并搜索拐点。 根据受控系统的最大斜率与死时间计算已调节的 PID 参数。

过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。 只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。 重新计算前会备份 PID 参数。

要求

- 已在循环中断 OB 中调用“PID_Compact”指令。
- ManualEnable = FALSE
- PID_Compact 处于“未激活”或“手动”模式。
- 控制器调节期间不能更改设定值。 否则将禁用 PID_Compact。
- 设定值和过程值均处于组态的限值范围内（请参见“过程值监视”组态）。
- 设定值与过程值的差值大于过程值上限与过程值下限之差的 30%。
- 设定值与过程值的差值大于设定值的 50%。

步骤

要执行预调节，请按下列步骤操作：

1. 在项目树中双击“PID_Compact > 调试”(PID_Compact > Commissioning) 条目。
2. 在“调节模式”(Tuning mode) 下拉列表中选择条目“预调节”(Pretuning)。
3. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动预调节功能。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。 进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及控制器调节功能看似受阻时，请单击“Stop”图标。 检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果执行预调节时未产生错误消息，则 PID 参数已调节完毕。PID_Compact 将切换到自动模式并使用已调节的参数。在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果无法实现预调节，PID_Compact 将切换到“未激活”模式。

参见

参数 State 和 sRet.i_Mode V1 (页 331)

调试 V1 (页 118)

精确调节 V1 (页 121)

“手动”模式 V1 (页 123)

4.2.2.3 精确调节 V1

精确调节将使过程值出现恒定受限的振荡。 将根据此振荡的幅度和频率为操作点优化 PID 参数。 所有 PID 参数都将根据相应结果进行重新计算。 精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。

PID_Compact 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。重新计算前会备份 PID 参数。

要求

- 已在循环中断 OB 中调用 PID_Compact 指令。
- ManualEnable = FALSE
- 设定值和过程值均处于组态的限值范围内（请参见“过程值监视”组态）。
- 在操作点处，控制回路已稳定。 过程值与设定值一致时，表明到达了操作点。
- 不能被干扰。
- 控制器调节期间不能更改设定值。
- PID_Compact 处于未激活模式、自动模式或手动模式。

过程取决于初始情况

可以在“未激活”、“自动”或“手动”模式下启动精确调节。 在以下模式下启动精确调节时，具体情况如下所述：

- 自动模式

  如果希望通过控制器调节来改进现有 PID 参数，请在自动模式下启动精确调节。

  PID_Comact 将使用现有的 PID 参数进行调节，直到控制回路已稳定并且精确调节的要求得到满足为止。之后才会启动精确调节。

- 未激活模式或手动模式

  如果满足预调节的要求，则启动预调节。 建立的 PID 参数将用于进行调节，直到控制回路已稳定并且精确调节的要求得到满足为止。之后才会启动精确调节。如果无法实现预调节，PID_Compact 将切换到“未激活”模式。

  如果预调节的过程值已经十分接近设定值，则将尝试利用最小或最大输出值来达到设定值。 这可能会增加超调量。

步骤

要执行“精确调节”，请按以下步骤操作：

1. 在“调节模式”(Tuning mode) 下拉列表中选择条目“精确调节”(Fine tuning)。
2. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动精确调节过程。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。 进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及控制器调节功能看似受阻时，请单击“调节模式”(Tuning mode) 组中的“Stop”图标。 检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果已执行精确调节且没有错误，则 PID 参数已得到优化。 PID_Compact 切换到自动模式，并使用优化的参数。 在电源关闭以及重启 CPU 期间，优化的 PID 参数保持不变。

如果“精确调节”期间出错，PID_Compact 将切换到“未激活”模式。

参见

参数 State 和 sRet.i_Mode V1 (页 331)

调试 V1 (页 118)

预调节 V1 (页 119)

“手动”模式 V1 (页 123)

4.2.2.4 “手动”模式 V1

下面说明如何在工艺对象“PID Compact”的调试窗口中使用“手动”工作模式。

要求

- “PID_Compact”指令在循环中断 OB 中调用。
- 与 CPU 建立了在线连接，并且 CPU 处于“RUN”模式。
- 已通过“启动”(Start) 图标启用了调试窗口的功能。

步骤

如果要通过指定手动值来测试过程，请使用调试窗口中的“手动模式”。 要定义手动值，请按以下步骤操作：

1. 在“控制器的在线状态”(Online status of the controller) 区域中，选中复选框“手动模式”(Manual mode)。

   PID_Compact 将在手动模式下运行。 最新的当前输出值仍然有效。

2. 在“输出”(Output) 字段中，输入 % 形式的手动值。
3. 单击控制图标 。

结果

手动值被写入 CPU 并立即生效。

说明

PID_Compact 继续监视过程值。 如果超出过程值的限值，则将禁用 PID_Compact。

如果希望 PID 控制器重新指定输出值，请清除“手动模式”(Manual mode) 复选框。 到自动模式的切换是无扰动的。

参见

参数 State 和 sRet.i_Mode V1 (页 331)

调试 V1 (页 118)

预调节 V1 (页 119)

精确调节 V1 (页 121)

4.2.3 使用 PLCSIM 仿真 PID_Compact V1

说明

使用 PLCSIM 进行仿真

对于使用 PLCSIM 进行的仿真，仿真 PLC 的时间特性与“真实”PLC 并不完全相同。仿真 PLC 循环中断 OB 的实际周期时钟波动比“真实”PLC 的波动大。

在标准组态中，PID_Compact 会自动确定调用之间的时间，并监视波动情况。

因此，使用 PLCSIM 仿真 PID_Compact 时，可能检测到采样时间错误 (ErrorBits = DW#16#00000800)。

在这种情况下，PID_Compact 切换到“未激活”模式 (State = 0)。

为防止此类情况发生，应按下列方式为使用 PLCSIM 进行的仿真组态 PID_Compact：

- sb_EnCyclEstimation = FALSE
- sb_EnCyclMonitoring = FALSE
- sPid_Calc.r_Cycle：以秒为单位为此变量分配调用循环中断 OB 的周期时钟。

4.3 工艺对象 PID_Compact

PID_Compact 工艺对象可实现一个集成优化功能的连续 PID 控制器。还可以组态脉冲控制器。手动和自动模式均可。

PID-Compact 连续采集在控制回路内测量的过程值，并将其与所需的设定值进行比较。指令 PID_Compact 根据所生成的控制偏差来计算输出值，通过该输出值，可以尽可能快速且稳定地将过程值调整为设定值。PID 控制器的输出值由三种作用构成：

- **比例**作用

  输出值的比例作用与控制偏差成比例增加。

- **I** 作用

  输出值的积分作用一直增加，直到控制偏差达到平衡状态。

- **D** 作用

  微分作用随控制偏差的变化率而增加。过程值会尽快校正到设定值。如果控制偏差的变化率下降，则微分作用将再次减弱。

指令 PID_Compact 在预调节期间计算受控系统的比例、积分和微分参数。精确调节可用于进一步调节这些参数。用户不必手动确定这些参数。

附加信息

- 软件控制器概述 (页 41)
- 添加工艺对象 (页 44)
- 组态工艺对象 (页 48)
- 组态 PID_Compact V2 (页 80)
- 组态 PID_Compact V1 (页 104)

常见问题解答

有关详细信息，请参见西门子工业在线支持中的以下常见问题解答。

- 条目 ID 79047707 (https://support.industry.siemens.com/cs/ww/en/view/79047707)

使用 PID_3Step

5

5.1 工艺对象 PID_3Step

工艺对象 PID_3Step 提供一个 PID 控制器，可通过积分响应对阀门或执行器进行调节。

可组态以下控制器：

- 带位置反馈的三点步进控制器
- 不带位置反馈的三点步进控制器
- 具有模拟量输出值的阀门控制器

PID_3Step 连续采集在控制回路内测量的过程值并将其与设定值进行比较。PID_3Step 根据所生成的控制偏差来计算输出值，通过该输出值，过程值可以尽可能快速且稳定地到达设定值。PID 控制器的输出值由三种作用构成：

- **比例**作用

  输出值的比例作用与控制偏差成比例增加。

- **I** 作用

  输出值的积分作用一直增加，直到控制偏差达到平衡状态。

- **D** 作用

  微分作用随控制偏差的变化率而增加。过程值会尽快校正到设定值。如果控制偏差的变化率下降，则微分作用将再次减弱。

指令 PID_3Step 在预调节期间计算受控系统的比例、积分和微分参数。精确调节可用于进一步调节这些参数。用户不必手动确定这些参数。

附加信息

- 软件控制器概述 (页 41)
- 添加工艺对象 (页 44)
- 组态工艺对象 (页 48)
- 组态 PID_3Step V2 (页 127)
- 组态 PID_3Step V1 (页 150)

原理

有关详细信息，请参见西门子工业在线支持中的以下常见问题解答。

- 条目 ID 68011827 (https://support.industry.siemens.com/cs/ww/en/view/68011827)

5.2 PID_3Step V2

5.2.1 组态 PID_3Step V2

5.2.1.1 基本设置 V2

简介 V2

在巡视窗口或组态窗口的“基本设置”(Basic settings) 下，组态工艺对象“PID_3Step”的以下属性：

- 物理量
- 控制逻辑
- 复位后的启动行为
- 设定值（仅在巡视窗口中）
- 过程值（仅在巡视窗口中）
- 输出值（仅在巡视窗口中）
- 位置反馈（仅在巡视窗口中）

设定值、过程值、输出值和位置反馈

只能在程序编辑器的巡视窗口中组态设定值、过程值、输出值和位置反馈。 为每个值选择一个源：

- 背景 DB

  使用背景数据块中保存的值。

  必须通过用户程序在背景 DB 中更新值。

  指令中不应有值。

  可通过 HMI 进行更改。

- 指令

  使用与指令相连的值。
  每次调用指令时都会将值写入背景数据块。

  无法通过 HMI 进行更改。

控制模式 V2

物理量

在“控制器类型”(Controller type) 组中，为设定值、过程值和扰动变量选择物理量和测量单位。 设定值、过程值和扰动变量以该测量单位显示。

控制逻辑

通常，可通过增大输出值来增大过程值。 这种做法称为常规控制逻辑。

PID_3Step 不使用负比例增益。 要在输出值增大时使过程值减小，请选中复选框“反转控制逻辑”(Invert control logic)。

示例

- 打开排泄阀将使容器盛装物的液位降低。
- 增加冷却能力将使温度降低。

启动特性

1. 要在 CPU 重启后切换到“未激活”模式，请清除“在 CPU 重启后激活模式”(Activate Mode after CPU restart) 复选框。

   要在 CPU 重启后切换到“模式”(Mode) 参数中保存的工作模式，请选中“在 CPU 重启后激活模式”(Activate Mode after CPU restart) 复选框。

2. 在“将模式设置为”(Set Mode to) 下拉列表中，选择要在完整下载到设备后启用的模式。

   完整下载到设备后，PID_3Step 以所选工作模式启动。 以后每次重启时，PID_3Step 都以上次保存在“模式”(Mode) 中的模式启动。

示例

您已选中“在 CPU 重启后激活模式”(Activate Mode after CPU restart) 复选框和“将模式设置为”(Set Mode to) 列表中的“预调节”(Pretuning) 条目。 完整下载到设备后，PID_3Step 以“预调节”(Pretuning) 模式启动。 如果预调节仍处于激活状态，则 PID_3Step 在 CPU 重启后再次以“预调节”(Pretuning) 模式启动。 如果预调节已成功完成并且自动模式处于激活状态，则 PID_3Step 在 CPU 重启后以“自动模式”(Automatic mode) 启动。

设定值 V2

步骤

要定义固定设定值，请按以下步骤操作：

1. 选择“背景 DB”(Instance DB)。
2. 输入一个设定值，例如 80° C。
3. 删除指令中的任何条目。

要定义可变设定值，请按以下步骤操作：

1. 选择“指令”(Instruction)。
2. 输入保存设定值的 REAL 变量的名称。

   可通过程序控制的方式来为该 REAL 变量分配变量值，例如，采用时间控制的方式来更改设定值。

过程值 V2

如果直接使用模拟量输入值，则 PID_3Step 会将该模拟量输入值标定为物理量。

如果要预先处理一下该模拟量输入值，则需要编写一个处理程序。 例如，过程值与模拟量输入值并不成正比。 经过处理的过程值必须为浮点格式。

步骤

要使用未经处理的模拟量输入值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input_PER”。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址。

要使用经过处理的浮点格式的过程值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input”。
2. 选择“指令”(Instruction) 作为源。
3. 输入变量的名称，用来保存经过处理的过程值。

位置反馈 V2

位置反馈组态取决于所用的执行器。

- 不提供位置反馈的执行器
- 提供数字停止位信号的执行器
- 提供模拟位置反馈的执行器
- 提供模拟位置反馈和停止位信号的执行器

不提供位置反馈的执行器

要为不提供位置反馈的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“无 Feedback”(No feedback)。

提供数字停止位信号的执行器

要为提供停止位信号的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“无 Feedback”(No feedback)。
2. 激活“执行器停止位信号”(Actuator endstop signals) 复选框。
3. 选择“指令”(Instruction) 作为 Actuator_H 和 Actuator_L 的源。
4. 分别为 Actuator_H 和 Actuator_L 输入数字量输入地址。

提供模拟位置反馈的执行器

要为提供模拟位置反馈的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“Feedback”或“Feedback_PER”。
   - 使用 Feedback_PER 的模拟量输入值。 在执行器设置中组态 Feedback_PER 标定。
   - 使用用户程序处理 Feedback 的模拟量输入值。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址或者用户程序的变量。

提供模拟位置反馈和停止位信号的执行器

要为提供模拟位置反馈和停止位信号的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“Feedback”或“Feedback_PER”。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址或者用户程序的变量。
4. 激活“执行器停止位信号”(Actuator endstop signals) 复选框。
5. 选择“指令”(Instruction) 作为 Actuator_H 和 Actuator_L 的源。
6. 分别为 Actuator_H 和 Actuator_L 输入数字量输入地址。

输出值 V2

PID_3Step 提供模拟量输出值 (Output_PER) 和数字量输出值（Output_UP、Output_DN）。执行器将决定要使用的输出值。

- Output_PER

  执行器具有相关的电机转换时间，可通过模拟量输出触发该执行器，并通过连续信号（如 0...10 V 或 4...20 mA）控制该执行器。Output_PER 的值与阀门的目标位置相对应，例如，当阀门打开 50% 时 Output_PER = 13824。

  对于自动调节和抗饱和行为，例如，PID_3Step 会将因电机转换时间所致的模拟量输出值对过程的延迟影响考虑在内。如果相关电机转换时间并未影响过程（如使用电磁阀），因此输出值直接且完全影响过程，则使用 PID_Compact。

- Output_UP、Output_DN

  执行器具有相关电机转换时间，通过两个数字量输出控制执行器。
  Output_UP 沿打开状态方向移动阀门。
  Output_DN 沿关闭状态方向移动阀门。

在计算模拟量输出值和数字量输出值时，会将电机转换时间考虑在内。自动调节和抗饱和行为期间，需要该时间来确保正常运行。因此，应在“执行器设置”下组态电机转换时间，其值为电机将执行器从关闭状态转为开启状态所需的时间。

步骤

要使用模拟量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output（模拟量）”(Output_PER (analog))。
2. 选择“指令”(Instruction)。
3. 输入模拟量输出的地址。

要使用数字量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output（数字量）”(Output (digital))。
2. 为 Output_UP 和 Output_DN 选择“指令”(Instruction)。
3. 输入数字量输出的地址。

要使用用户程序来处理输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择与该执行器对应的条目。
2. 选择“指令”(Instruction)。
3. 输入用于处理输出值的变量的名称。
4. 通过模拟量或数字量 CPU 输出将经过处理的输出值传送到执行器。

5.2.1.2 过程值设置 V2

过程值标定 V2

如果已在基本设置中对 Input_PER 的使用进行了组态，则必须将模拟量输入值转换为过程值的物理量。 当前组态将显示在 Input_PER 画面中。

如果过程值与模拟量输入值成正比，则将使用上下限值对来标定 Input_PER。

步骤

要标定过程值，请按下列步骤操作：

1. 在“标定的过程值的下限”(Scaled low process value) 和“下限”(Low) 文本框中输入一对下限值。
2. 在“标定的过程值的上限”(Scaled high process value) 和“上限”(High) 输入框中输入一对上限值。

这些值对的默认设置存储在硬件配置中。 要使用硬件配置中的值对，请按下列步骤操作：

1. 在程序编辑器中选择 PID_3Step 指令。
2. 在基本设置中将 Input_PER 与模拟量输入互连。
3. 在过程值设置中单击“自动设置”(Automatic setting) 按钮。

现有值将被硬件配置中的值覆盖。

过程值的限值 V2

必须为过程值指定正确的绝对上限和绝对下限，作为受控系统的限值。只要过程值超出这些限值，就会出现错误 (ErrorBits = 0001h)。 如果超出过程值的限值，则取消调节操作。可以在执行器设置中指定 PID_3Step 在自动模式下对错误的响应方式。

5.2.1.3 最终控制元件设置 V2

最终控制元件 V2

执行器特定的时间

组态电机转换时间和最短开关时间，以防止执行器被损坏。可以在执行器数据表中找到相应规范。

电机转换时间指的是电机将执行器从关闭状态转为开启状态所需的时间（以秒为单位）。可以在调试期间测量电机转换时间。

在计算模拟量输出值和数字量输出值时，会将电机转换时间考虑在内。自动调节和抗饱和行为期间，需要该时间来确保正常运行。

如果相关电机转换时间并未影响过程（如使用电磁阀），因此输出值直接且完全影响过程，则使用 PID_Compact。

电机转换时间具有保持性。如果手动输入电机转换时间，则必须完整下载 PID_3Step。

将工艺对象下载到设备 (页 76)

如果正在使用“Output_UP”或“Output_DN”，则可通过最短开启时间和最短关闭时间来降低开关频率。

在自动模式下，计算出的开启或关闭时间会进行累加，并且仅当累加总和大于或等于最短开启或关闭时间时，计算出的开启或关闭时间才生效。

手动模式下，Manual_UP = TRUE 或 Manual_DN = TRUE 操作执行器并至少持续最短开启时间或最短关闭时间。

如果已选择模拟量输出值 Output_PER，将不评估最短开启时间和最短关闭时间，并且也无法更改这两个时间。

对错误的响应

PID_3Step 需要预设置，以便在发生错误时，控制器在大多数情况下均可保持激活状态。如果在控制器模式下频繁发生错误，则该默认响应会对控制响应产生负面影响。这种情况下，检查 Errorbits 参数并消除错误原因。

注意
您的系统可能已损坏。 如果在出现错误时输出“错误未决时的当前值”或“错误未决时的替代输出值”，则 PID_3Step 将保持自动模式，即使已超出过程值的限值。这可能损坏您的系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

PID_3Step 会在出现错误时生成可编程的输出值：

- 当前值

  PID_3Step 关闭，且不再修改执行器位置。

- 发生错误时（错误未决时）的当前值

  PID_3Step 的控制器功能被关闭，并且执行器的位置不再发生变化。

  如果在自动模式下发生以下错误，则只要这些错误不再处于未决状态，PID_3Step 便会返回到自动模式。

  - 0002h：Input_PER 参数的值无效。
  - 0200h：Input 参数的值无效。
  - 0400h：输出值计算失败。
  - 1000h：Setpoint 参数的值无效。
  - 2000h：Feedback_PER 参数的值无效。
  - 4000h：Feedback 参数的值无效。
  - 8000h：数字位置反馈期间出错。
  - 20000h：变量 SavePosition 的值无效。

如果发生一个或多个下列错误，则 PID_3Step 停留在
自动模式下：

- 0001h：参数 Input 超出了过程值限值的范围。
- 0800h：采样时间错误
- 40000h：Disturbance 参数的值无效。

如果在手动模式下发生错误，则 PID_3Step 仍保持在手动模式下。

如果在调节或转换时间测量期间发生错误，PID_3Step 将切换到启动调节或转换时间测量时的模式。只有发生以下错误时不会中止调节：

- 0020h：精确调节期间不允许预调节。

- 替代输出值

  PID_3Step 将执行器移动到替代输出值位置，然后关闭。

- 错误未决时的替代输出值

  PID_3Step 将执行器移动到替代输出值位置。达到替代输出值时，PID_3Step 的响应与处理“错误未决时的当前值”的方式相同。

输入百分数形式的替代输出值。

对于不提供模拟位置反馈的执行器，只能精确逼近替代输出值 0% 和 100%。可通过内部仿真的位置反馈逼近不等于 0% 或 100% 的替代输出值。但是，此过程不能精确地逼近替代输出值。

对于提供模拟位置反馈的执行器，可以精确逼近所有替代输出值。

标定 V2 位置反馈

标定位置反馈

如果已在基本设置中组态使用 Feedback_PER，则需要将模拟量输入值转换为百分数形式。 当前组态将显示在“Feedback”画面中。

使用上下限值对来标定 Feedback_PER。

1. 在“下端停止位”和“下限”输入框中输入一对下限值。

2. 在“上端停止位”和“上限”输入框中输入一对上限值。

“下端停止位”必须小于“上端停止位”；“下限”必须小于“上限”。

“上端停止位”和“下端停止位”的有效值取决于：

- 无 Feedback、Feedback、Feedback_PER
- Output（模拟量）、Output（数字量）

Output	Feedback	下端停止位	上端停止位
Output（数字量）	无 Feedback	无法设置 (0.0%)	无法设置 (100.0%)
Output（数字量）	Feedback	-100.0% 或 0.0%	0.0% 或 +100.0%
Output（数字量）	Feedback_PER	-100.0% 或 0.0%	0.0% 或 +100.0%
Output（模拟量）	无 Feedback	无法设置 (0.0%)	无法设置 (100.0%)
Output（模拟量）	Feedback	-100.0% 或 0.0%	0.0% 或 +100.0%
Output（模拟量）	Feedback_PER	-100.0% 或 0.0%	0.0% 或 +100.0%

输出值的限值 V2

限制输出值

在测量转换时间期间且模式 = 10 时，输出值才能高于上限或低于下限。在所有其它模式下输出值都会被限制为这些值。

在“输出值上限”和“输出值下限”输入框中，键入输出值的绝对限值。输出值的限值必须位于“下端停止位”和“上端停止位”范围内。

如果无 Feedback 可用并且置位了 Output（数字量），则不可限制输出值。Output_UP 和 Output_DN 将在 Actuator_H = TRUE 或 Actuator_L = TRUE 时复位。如果不存在停止位信号，则 Output_UP 和 Output_DN 将在 150% 的电机执行时间的行程时间过后复位。

默认值 150% 可以通过变量 Config.VirtualActuatorLimit 进行调整。自 PID_3Step 版本 2.3 起，可以通过 Config.VirtualActuatorLimit = 0.0 取消激活行程时间的监视和限制。

5.2.1.4 高级设置 V2

实际值监视 V2

在“过程值监视”(Process value monitoring) 组态窗口中，组态过程值的警告上限和下限。如果在运行期间超出或低于某一警告限值，则将在 PID_3Step 指令的以下参数中显示一条警告：

- 输出参数 InputWarning_H，前提是超出警告上限
- 输出参数 InputWarning_L，前提是低于警告下限

警告限值必须处于过程值的限值范围内。

如果未输入警告限值，将使用过程值的上限和下限。

示例

过程值上限 = 98° C；警告上限 = 90° C

警告下限 = 10° C；过程值下限 = 0° C

PID_3Step 将按如下方式响应：

过程值	InputWarning_H	InputWarning_L	Error Bits	工作模式
> 98° C	TRUE	FALSE	0001 h	按照组态
≤ 98° C 且 > 90° C	TRUE	FALSE	0000 h	自动模式
≤ 90° C 且 ≥ 10° C	FALSE	FALSE	0000 h	自动模式
< 10° C 且 ≥ 0° C	FALSE	TRUE	0000 h	自动模式
< 0° C	FALSE	TRUE	0001 h	按照组态

在执行器设置中，您可以组态超出过程值上限或下限时 PID_3Step 的响应。

PID 参数 V2

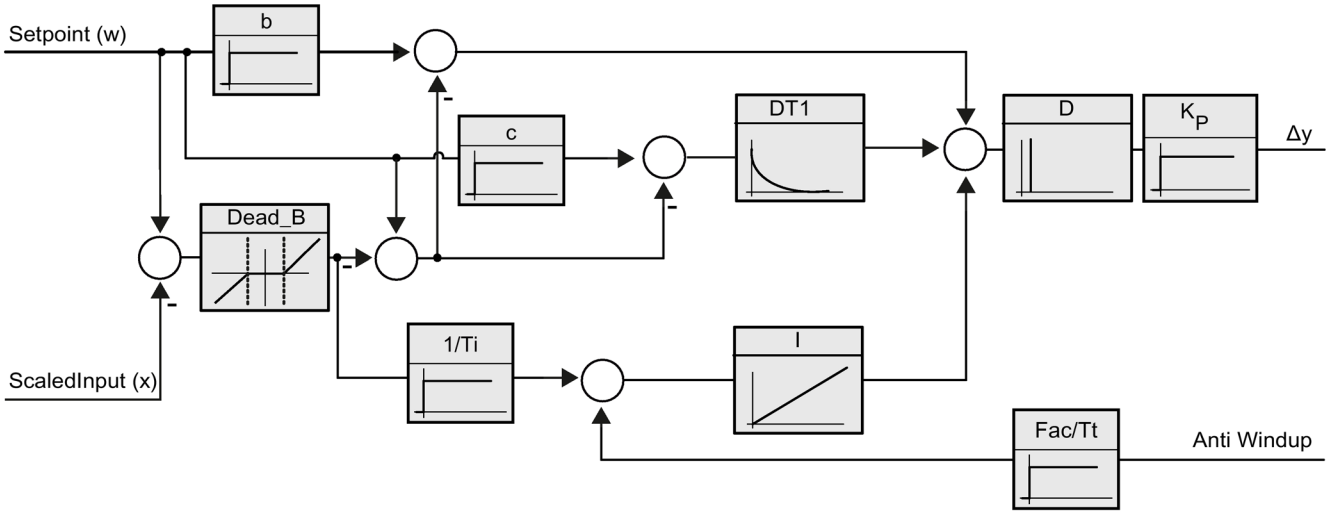
PID 参数显示在“PID 参数”(PID Parameters) 组态窗口中。在控制器调节期间将调整 PID 参数以适应受控系统。用户不必手动输入 PID 参数。

PID 算法根据以下等式工作：

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
Δy	PID 算法的输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分时间
a	微分延迟系数（微分延迟 T1 = a × T_D）
T_D	微分作用时间
c	微分作用权重

下图说明了集成到 PID 算法中的参数：



所有 PID 参数均具有保持性。如果手动输入 PID 参数，则必须完整下载 PID_3Step。

将工艺对象下载到设备 (页 76)

比例增益

该值用于指定控制器的比例增益。PID_3Step 不使用负比例增益。在“基本设置 > 控制器类型”下，控制逻辑会反转。

积分时间

积分时间用于确定积分作用的时间特性。积分时间 = 0.0 时，将禁用积分作用。

微分作用时间

微分作用时间用于确定微分作用的时间特性。微分作用时间 = 0.0 时，将禁用微分作用。

微分延迟系数

微分延迟系数用于延迟微分作用的生效。

微分延迟 = 微分作用时间 × 微分延迟系数

- 0.0：微分作用仅在一个周期内有效，因此几乎不产生影响。
- 0.5：此值经实践证明对于具有一个优先时间常量的受控系统非常有用。
- > 1.0：系数越大，微分作用的生效时间延迟越久。

比例作用权重

比例作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0：应对设定值变化的比例作用完全有效
- 0.0：应对设定值变化的比例作用无效

当过程值变化时，比例作用始终完全有效。

微分作用权重

微分作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0：设定值变化时微分作用完全有效
- 0.0：设定值变化时微分作用不生效

当过程值变化时，微分作用始终完全有效。

PID 算法采样时间

受控系统需要一定的时间来对输出值的变化做出响应。因此，建议不要在每次循环中都计算输出值。PID 算法的采样时间是两次计算输出值之间的时间。该时间在调节期间进行计算，并舍入为 PID_3Step 采样时间的倍数。PID_3Step 的所有其它功能在每次调用时均执行。

死区宽度

死区可抑制控制器处于稳态的噪声分量。死区宽度指定死区的大小。如果死区宽度为 0.0，则死区关闭。

如果将不等于 1.0 的值组态为比例作用权重或微分作用权重，则即使在死区内，设定值的变化也会影响输出值。
无论权重如何，死区内的过程值变化都不会影响输出值。

5.2.2 调试 PID_3Step V2

5.2.2.1 预调节 V2

预调节可确定对输出值脉冲的过程响应，并搜索拐点。 根据受控系统的最大斜率与死时间计算已调节的 PID 参数。 可在执行预调节和精确调节时获得最佳 PID 参数。

过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。 只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。 最可能的情况是处于工作模式“未激活”和“手动模式”下。 重新计算前会备份 PID 参数。

预调节期间冻结设定值。

要求

- 已在循环中断 OB 中调用 PID_3Step 指令。
- ManualEnable = FALSE
- Reset = FALSE
- 已对电机转换时间进行了组态或测量。
- PID_3Step 处于下列模式之一： “未激活”、“手动模式”或“自动模式”。
- 设定值和过程值均处于组态的限值范围内（请参见“过程值设置”组态）。

步骤

要执行预调节，请按下列步骤操作：

1. 在项目树中双击“PID_3Step > 调试”(PID_3Step > Commissioning) 条目。
2. 在“调节”(Tuning) 工作区的“调节模式”(Tuning mode) 下拉列表中选择条目“预调节”(Pretuning)。
3. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动预调节功能。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及控制器调节功能看似受阻时，请单击“Stop”图标。检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果执行预调节时未产生错误消息，则 PID 参数已调节完毕。PID_3Step 将切换到自动模式并使用已调节的参数。在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果无法实现预调节，PID_3Step 将根据已组态的响应对错误作出反应。

5.2.2.2 精确调节 V2

精确调节将使过程值出现恒定受限的振荡。将根据此振荡的幅度和频率为操作点调节 PID 参数。所有 PID 参数都根据结果重新计算。 精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。 可在执行预调节和精确调节时获得最佳 PID 参数。

PID_3Step 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。 重新计算前会备份 PID 参数。

精确调节期间冻结设定值。

要求

- 已在循环中断 OB 中调用 PID_3Step 指令。
- ManualEnable = FALSE
- Reset = FALSE
- 已对电机转换时间进行了组态或测量。
- 设定值和过程值均处于组态的限值范围内（请参见“过程值设置”组态）。
- 在操作点处，控制回路已稳定。 过程值与设定值一致时，表明到达了操作点。
- 不能被干扰。
- PID_3Step 处于未激活模式、自动模式或手动模式。

过程取决于初始情况

在以下模式下启动精确调节时，具体情况如下所述：

- 自动模式

  如果希望通过调节来改进现有 PID 参数，请在自动模式下启动精确调节。

  PID_3Step 将使用现有的 PID 参数控制系统，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。

- 未激活模式或手动模式

  总是先启动预调节。已确定的 PID 参数将用于控制，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。

步骤

要执行精确调节，请按下列步骤操作：

1. 在“调节模式”(Tuning mode) 下拉列表中选择条目“精确调节”(Fine tuning)。
2. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动精确调节过程。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。 进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及控制器调节功能看似受阻时，请单击“调节模式”(Tuning mode) 组中的“Stop”图标。 检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果在精确调节期间未产生错误，则 PID 参数已调节完毕。PID_3Step 将切换到自动模式并使用已调节的参数。在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果精确调节期间出现错误，PID_3Step 将根据已组态的响应对错误作出反应。

5.2.2.3 使用手动 PID 参数 V2 进行调试

要求

- 已在循环中断 OB 中调用 PID_3Step 指令。
- ManualEnable = FALSE
- Reset = FALSE
- 已对电机转换时间进行了组态或测量。
- PID_3Step 处于“未激活”模式。
- 设定值和过程值均处于组态的限值范围内（请参见“过程值设置”组态）。

步骤

要使用手动 PID 参数调试 PID_3Step，请按以下步骤操作：

1. 在项目树中双击“PID_3Step > 组态”(PID_3Step > Configuration)。
2. 在组态窗口中单击“高级设置 > PID 参数”(Advanced settings > PID Parameters)。
3. 选中复选框“启用直接输入”(Enable direct input)。
4. 输入 PID 参数。
5. 在项目树中双击“PID_3Step > 调试”(PID_3Step > Commissioning) 条目。
6. 与 CPU 之间建立在线连接。
7. 将 PID 参数装载到 CPU。
8. 单击“Start PID_3Step”图标。

结果

PID_3Step 切换到自动模式，并使用当前 PID 参数进行控制。

参见

PID 参数 V2 (页 140)

5.2.2.4 测量电机转换时间 V2

简介

PID_3Step 要求电机转换时间尽可能准确，以便获得良好的控制器结果。 执行器文档中的数据包含此类执行器的平均值。针对特定执行器的值可能不同。

如果使用提供位置反馈或停止位信号的执行器，则可在调试期间测量电机转换时间。 测量电机转换时间期间，不考虑输出值的限值。 执行器可行进至上端停止位或下端停止位。

如果位置反馈或停止位信号均不可用，则无法测量电机转换时间。

提供模拟位置反馈的执行器

要使用位置反馈测量电机转换时间，请按以下步骤操作：

要求

- 已在基本设置中选择 Feedback 或 Feedback_PER 并且已连接信号。
- 已与 CPU 建立在线连接。

1. 选中“使用位置反馈”(Use position feedback) 复选框。
2. 在“目标位置”(Target position) 输入字段中输入执行器要移动到的位置。

   将显示当前位置反馈（起始位置）。 “目标位置”(Target position) 与“位置反馈”(Position feedback) 之间的差值必须至少为有效输出值范围的 50%。
3. 单击“Start”图标。

结果

将执行器从起始位置移动到目标位置。 立即开始时间测量并在执行器到达目标位置时结束。 根据以下等式计算电机转换时间：

电机转换时间 = (输出值上限 - 输出值下限) × 测量时间/总量 (目标位置 - 起始位置)。

将显示转换时间测量的进度和状态。 测得的转换时间保存在 CPU 的背景数据块中，并显示在“测量的转换时间”(Measured transition time) 字段中。 转换时间测量结束且 ActivateRecoverMode = TRUE 时，PID_3Step 切换到转换时间测量开始时的工作模式。转换时间测量结束且 ActivateRecoverMode = FALSE 时，PID_3Step 切换到“未激活”(Inactive) 模式。

说明

单击图标  “上传所测量的转换时间”(Load measured transition time)，将所测量的电机转换时间装载到项目中。

提供停止位信号的执行器

要测量提供停止位信号的执行器的转换时间，请按以下步骤操作：

要求

- 已在基本设置中选中“停止位信号”(Endstop signals) 复选框并且已连接 Actuator_H 和 Actuator_L。
- 已与 CPU 建立在线连接。

要使用停止位信号测量电机转换时间，请按以下步骤操作：

1. 选中“使用执行器停止位信号”(Use actuator endstop signals) 复选框。
2. 选择要在哪个方向上移动执行器。
   - 打开 - 关闭 - 打开

     执行器首先会移动到上端停止位，接着移动到下端停止位，然后返回到上端停止位。
   - 关闭 - 打开 - 关闭

     执行器首先会移动到下端停止位，接着移动到上端停止位，然后返回到下端停止位。
3. 单击“Start”图标。

结果

沿所选方向移动执行器。 时间测量将在执行器到达第一个停止位时启动，而在执行器第二次到达该停止位时结束。 电机转换时间等于所测得的时间除以二。

将显示转换时间测量的进度和状态。 测得的转换时间保存在 CPU 的背景数据块中，并显示在“测量的转换时间”(Measured transition time) 字段中。 转换时间测量结束且 ActivateRecoverMode = TRUE 时，PID_3Step 切换到转换时间测量开始时的工作模式。转换时间测量结束且 ActivateRecoverMode = FALSE 时，PID_3Step 切换到“未激活”(Inactive) 模式。

取消转换时间测量

如果您通过按下 Stop 按钮取消转换时间测量，则 PID_3Step 会切换到“未激活”模式。

5.2.3 使用 PLCSIM 仿真 PID_3Step V2

说明

使用 PLCSIM 进行仿真

不支持使用 PLCSIM 仿真 PID_3Step V2.x 后将其用于 CPU S7-1200。

只能通过 PLCSIM 针对 CPU S7-1500 仿真 PID_3Step V2.x。

对于使用 PLCSIM 进行的仿真，仿真 PLC 的时间特性与“真实”PLC 并不完全相同。仿真 PLC 循环中断 OB 的实际周期时钟波动比“真实”PLC 的波动大。

在标准组态中，PID_3Step 会自动确定调用之间的时间，并监视波动情况。

因此，使用 PLCSIM 仿真 PID_3Step 时，可能检测到采样时间错误 ((ErrorBits = DW#16#00000800)。

这会导致进行中的调节中止。

自动模式下的响应取决于 ActivateRecoverMode 变量的值。

为防止此类情况发生，应按下列方式为使用 PLCSIM 进行的仿真组态 PID_3Step：

- CycleTime.EnEstimation = FALSE
- CycleTime.EnMonitoring = FALSE
- CycleTime.Value：以秒为单位为此变量分配调用循环中断 OB 的周期时钟。

5.3 PID_3Step V1

5.3.1 组态 PID_3Step V1

5.3.1.1 基本设置 V1

简介 V1

在巡视窗口或组态窗口的“基本设置”(Basic settings) 下，组态工艺对象“PID_3Step”的以下属性：

- 物理量
- 控制逻辑
- 复位后的启动行为
- 设定值（仅在巡视窗口中）
- 过程值（仅在巡视窗口中）
- 输出值（仅在巡视窗口中）
- 位置反馈（仅在巡视窗口中）

设定值、过程值、输出值和位置反馈

只能在程序编辑器的巡视窗口中组态设定值、过程值、输出值和位置反馈。 为每个值选择一个源：

- 背景 DB

  使用背景数据块中保存的值。

  必须通过用户程序在背景 DB 中更新值。

  指令中不应有值。

  可通过 HMI 进行更改。

- 指令

  使用与指令相连的值。
  每次调用指令时都会将值写入背景数据块。

  无法通过 HMI 进行更改。

控制模式 V1

物理量

在“控制器类型”(Controller type) 组中，为设定值和过程值选择测量单位和物理量。 设定值和过程值将以该测量单位显示。

控制逻辑

通常，可通过增大输出值来增大过程值。 这种做法称为常规控制逻辑。

PID_3Step 不使用负比例增益。 要在输出值增大时使过程值减小，请选中复选框“反转控制逻辑”(Invert control logic)。

示例

- 打开排泄阀将使容器盛装物的液位降低。
- 增加冷却能力将使温度降低。

复位后的启动行为

要在重启 CPU 后直接切换到上次激活的模式，请选中“CPU 重启后启用上一模式”(Enable last mode after CPU restart) 复选框。

如果清除该复选框，PID_3Step 将保持在“未激活”模式。

设定值 V1

步骤

要定义固定设定值，请按以下步骤操作：

1. 选择“背景 DB”(Instance DB)。
2. 输入一个设定值，例如 80° C。
3. 删除指令中的任何条目。

要定义可变设定值，请按以下步骤操作：

1. 选择“指令”(Instruction)。
2. 输入保存设定值的 REAL 变量的名称。

   可通过程序控制的方式来为该 REAL 变量分配变量值，例如，采用时间控制的方式来更改设定值。

过程值 V1

如果直接使用模拟量输入值，则 PID_3Step 会将该模拟量输入值标定为物理量。

如果要预先处理一下该模拟量输入值，则需要编写一个处理程序。 例如，过程值与模拟量输入值并不成正比。 经过处理的过程值必须为浮点格式。

步骤

要使用未经处理的模拟量输入值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input_PER”。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址。

要使用经过处理的浮点格式的过程值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input”。
2. 选择“指令”(Instruction) 作为源。
3. 输入变量的名称，用来保存经过处理的过程值。

位置反馈 V1

位置反馈组态取决于所用的执行器。

- 不提供位置反馈的执行器
- 提供数字停止位信号的执行器
- 提供模拟位置反馈的执行器
- 提供模拟位置反馈和停止位信号的执行器

不提供位置反馈的执行器

要为不提供位置反馈的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“无 Feedback”(No feedback)。

提供数字停止位信号的执行器

要为提供停止位信号的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“无 Feedback”(No feedback)。
2. 激活“执行器停止位信号”(Actuator endstop signals) 复选框。
3. 选择“指令”(Instruction) 作为 Actuator_H 和 Actuator_L 的源。
4. 分别为 Actuator_H 和 Actuator_L 输入数字量输入地址。

提供模拟位置反馈的执行器

要为提供模拟位置反馈的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“Feedback”或“Feedback_PER”。
   - 使用 Feedback_PER 的模拟量输入值。 在执行器设置中组态 Feedback_PER 标定。
   - 使用用户程序处理 Feedback 的模拟量输入值。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址或者用户程序的变量。

提供模拟位置反馈和停止位信号的执行器

要为提供模拟位置反馈和停止位信号的执行器组态 PID_3Step，请按以下步骤操作：

1. 在下拉列表“Feedback”中选择条目“Feedback”或“Feedback_PER”。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址或者用户程序的变量。
4. 激活“执行器停止位信号”(Actuator endstop signals) 复选框。
5. 选择“指令”(Instruction) 作为 Actuator_H 和 Actuator_L 的源。
6. 分别为 Actuator_H 和 Actuator_L 输入数字量输入地址。

输出值 V1

PID_3Step 提供模拟量输出值 (Output_PER) 和数字量输出值（Output_UP、Output_DN）。执行器将决定要使用的输出值。

- Output_PER

  执行器具有相关的电机转换时间，可通过模拟量输出触发该执行器，并通过连续信号（如 0...10 V 或 4...20 mA）控制该执行器。Output_PER 的值与阀门的目标位置相对应，例如，当阀门打开 50% 时 Output_PER = 13824。

  对于自动调节和抗饱和行为，例如，PID_3Step 会将因电机转换时间所致的模拟量输出值对过程的延迟影响考虑在内。如果相关电机转换时间并未影响过程（如使用电磁阀），因此输出值直接且完全影响过程，则使用 PID_Compact。

- Output_UP、Output_DN

  执行器具有相关电机转换时间，通过两个数字量输出控制执行器。
  Output_UP 沿打开状态方向移动阀门。
  Output_DN 沿关闭状态方向移动阀门。

在计算模拟量输出值和数字量输出值时，会将电机转换时间考虑在内。自动调节和抗饱和行为期间，需要该时间来确保正常运行。因此，应在“执行器设置”下组态电机转换时间，其值为电机将执行器从关闭状态转为开启状态所需的时间。

步骤

要使用模拟量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output（模拟量）”(Output_PER (analog))。
2. 选择“指令”(Instruction)。
3. 输入模拟量输出的地址。

要使用数字量输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择条目“Output（数字量）”(Output (digital))。
2. 为 Output_UP 和 Output_DN 选择“指令”(Instruction)。
3. 输入数字量输出的地址。

要使用用户程序来处理输出值，请按以下步骤操作：

1. 在下拉列表“Output”中选择与该执行器对应的条目。
2. 选择“指令”(Instruction)。
3. 输入用于处理输出值的变量的名称。
4. 通过模拟量或数字量 CPU 输出将经过处理的输出值传送到执行器。

5.3.1.2 过程值设置 V1

在“过程值设置”(Process value settings) 组态窗口中，组态过程值的标定并指定过程值的绝对限值。

标定过程值

如果已在基本设置中对 Input_PER 的使用进行了组态，则需要将模拟量输入值转换为过程值的物理量。 当前组态将显示在 Input_PER 画面中。

如果过程值与模拟量输入值成正比，则将使用上下限值对来标定 Input_PER。

1. 在“标定的过程值的下限”(Scaled low process value) 和“下限”(Low) 输入字段中输入一对下限值。
2. 在“标定的过程值的上限”(Scaled high process value) 和“上限”(High) 输入框中输入一对上限值。

这些值对的默认设置保存在硬件配置中。 要使用硬件配置中的值对，请按以下步骤操作：

1. 在程序编辑器中选择指令 PID_3Step。
2. 在基本设置中连接 Input_PER 与模拟量输入。
3. 在过程值设置中单击“自动设置”(Automatic setting) 按钮。

   现有值将被硬件配置中的值覆盖。

监视过程值

指定过程值的绝对上限和下限。 必须为受控系统输入合理的限值。 合理的限值在获取最优 PID 参数的优化过程中是重要的。 “过程值的上限”的默认值是 120 %。 在 I/O 输入中，过程值最大可超出标准范围 18%（过范围）。 此设置可确保因超出“过程值上限”而不再报告错误。仅识别断线和短路，然后 PID_3Step 将根据已组态的错误响应方式进行响应。

注意
您的系统可能已损坏。 如果将过程值的限值范围设置得非常大（例如 $-3.4*10^{38}...+3.4*10^{38}$），则将禁用过程值监视功能。 如果发生错误，则可能损坏系统。 您需要为受控系统组态有效的过程值。

5.3.1.3 V1 最终控制元件设置

执行器特定的时间

组态电机转换时间和最短开关时间，以防止执行器被损坏。可以在执行器数据表中找到相应规范。

电机转换时间指的是电机将执行器从关闭状态转为开启状态所需的时间（以秒为单位）。执行器在一个方向上移动的最长时间是电机转换时间的 110%。可以在调试期间测量电机转换时间。

在计算模拟量输出值和数字量输出值时，会将电机转换时间考虑在内。自动调节和抗饱和行为期间，需要该时间来确保正常运行。

如果相关电机转换时间并未影响过程（如使用电磁阀），因此输出值直接且完全影响过程，则使用 PID_Compact。

如果正在使用“Output_UP”或“Output_DN”，则可通过最短开启时间和最短关闭时间来降低开关频率。

在自动模式下，计算出的开启或关闭时间会进行累加，并且仅当累加总和大于或等于最短开启或关闭时间时，计算出的开启或关闭时间才生效。

在手动模式下，Manual_UP 或 Manual_DN 的上升沿将会使执行器运行，运行时间至少为最短开启或关闭时间。

如果已选择模拟量输出值 Output_PER，将不评估最短开启时间和最短关闭时间，并且也无法更改这两个时间。

对错误的响应

PID_3Step 需要预设置，以便在发生错误时，控制器在大多数情况下均可保持激活状态。如果在控制器模式下频繁发生错误，则该默认响应会对控制响应产生负面影响。这种情况下，检查 Errorbits 参数并消除错误原因。

PID_3Step 会生成可设定的输出值来对错误做出响应：

- 当前值

  PID_3Step 关闭，且不再修改执行器位置。

- 发生错误时（错误未决时）的当前值

  PID_3Step 的控制器功能被关闭，并且执行器的位置不再发生变化。

  如果在自动模式下发生以下错误，则只要这些错误不再处于未决状态，PID_3Step 便会返回到自动模式。

  - 0002h：Input_PER 参数的值无效。
  - 0200h：Input 参数的值无效。
  - 0800h：采样时间错误
  - 1000h：Setpoint 参数的值无效。
  - 2000h：Feedback_PER 参数的值无效。
  - 4000h：Feedback 参数的值无效。
  - 8000h：数字位置反馈期间出错。

  如果在手动模式下发生上述错误之一，则 PID_3Step 将保持在手动模式下。

  如果在调节期间或转换时间测量期间发生错误，则 PID_3Step 将关闭。

- 替代输出值

  PID_3Step 将执行器移动到替代输出值位置，然后关闭。

- 错误未决时的替代输出值

  PID_3Step 将执行器移动到替代输出值位置。达到替代输出值时，PID_3Step 的响应与处理“错误未决时的当前值”的方式相同。

输入百分数形式的替代输出值。

对于不提供模拟位置反馈的执行器，只能精确逼近替代输出值 0% 和 100%。执行器将在一个方向上持续移动 110% 的电机转换时间，以确保达到上端或下端停止位。此处的停止位信号具有优先权。可通过内部仿真的位置反馈逼近不等于 0% 或 100% 的替代输出值。但是，此过程不能精确地逼近替代输出值。

对于提供模拟位置反馈的执行器，可以精确逼近所有替代输出值。

标定位置反馈

如果已在基本设置中组态使用 Feedback_PER，则需要将模拟量输入值转换为百分数形式。当前组态将显示在“Feedback”画面中。

使用上下限值对来标定 Feedback_PER。

1. 在“下端停止位”和“下限”输入框中输入一对下限值。

2. 在“上端停止位”和“上限”输入框中输入一对上限值。

“下端停止位”必须小于“上端停止位”；“下限”必须小于“上限”。

“上端停止位”和“下端停止位”的有效值取决于：

- 无 Feedback、Feedback、Feedback_PER
- Output（模拟量）、Output（数字量）

Output	Feedback	下端停止位	上端停止位
Output（数字量）	无 Feedback	无法设置 (0.0%)	无法设置 (100.0%)
Output（数字量）	Feedback	-100.0% 或 0.0%	0.0% 或 +100.0%
Output（数字量）	Feedback_PER	-100.0% 或 0.0%	0.0% 或 +100.0%
Output（模拟量）	无 Feedback	无法设置 (0.0%)	无法设置 (100.0%)
Output（模拟量）	Feedback	-100.0% 或 0.0%	0.0% 或 +100.0%
Output（模拟量）	Feedback_PER	-100.0% 或 0.0%	0.0% 或 +100.0%

限制输出值

只有在测量转换时间期间，输出值才能高于上限或低于下限。在所有其它模式下输出值都会被限制为这些值。

在“输出值上限”和“输出值下限”输入框中，键入输出值的绝对限值。输出值的限值必须位于“下端停止位”和“上端停止位”范围内。

如果无 Feedback 可用并且置位了 Output（数字量），则不能限制输出值。当 Actuator_H = TRUE 或 Actuator_L = TRUE 时，或者在行进时间达到电机转换时间的 110% 后，数字量输出将复位。

5.3.1.4 高级设置 V1

实际值监视 V1

在“过程值监视”(Process value monitoring) 组态窗口中，组态过程值的警告上限和下限。如果在运行期间超出或低于某一警告限值，则将在 PID_3Step 指令的以下参数中显示一条警告：

- 输出参数 InputWarning_H，前提是超出警告上限
- 输出参数 InputWarning_L，前提是低于警告下限

警告限值必须处于过程值的限值范围内。

如果未输入警告限值，将使用过程值的上限和下限。

示例

过程值上限 = 98° C；警告上限 = 90° C

警告下限 = 10° C；过程值下限 = 0° C

PID_3Step 将按如下方式响应：

过程值	InputWarning_H	InputWarning_L	工作模式
> 98° C	TRUE	FALSE	未激活
≤ 98° C 且 > 90° C	TRUE	FALSE	自动模式
≤ 90° C 且 ≥ 10° C	FALSE	FALSE	自动模式
< 10° C 且 ≥ 0° C	FALSE	TRUE	自动模式
< 0° C	FALSE	TRUE	未激活

PID 参数 V1

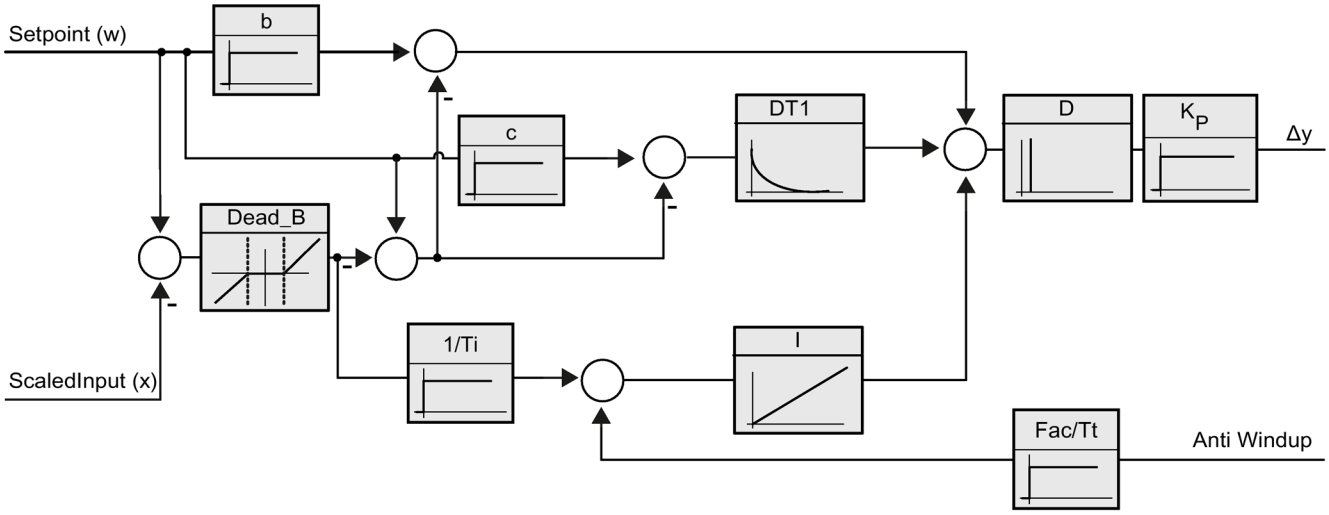
PID 参数显示在“PID 参数”(PID Parameters) 组态窗口中。在控制器调节期间将调整 PID 参数以适应受控系统。用户不必手动输入 PID 参数。

PID 算法根据以下等式工作：

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
Δy	PID 算法的输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分时间
a	微分延迟系数（微分延迟 T1 = a × T_D）
T_D	微分作用时间
c	微分作用权重

下图说明了集成到 PID 算法中的参数：



所有 PID 参数均具有保持性。如果手动输入 PID 参数，则必须完整下载 PID_3Step。

将工艺对象下载到设备 (页 76)

比例增益

该值用于指定控制器的比例增益。PID_3Step 不使用负比例增益。在“基本设置 > 控制器类型”下，控制逻辑会反转。

积分时间

积分时间用于确定积分作用的时间特性。积分时间 = 0.0 时，将禁用积分作用。

微分作用时间

微分作用时间用于确定微分作用的时间特性。微分作用时间 = 0.0 时，将禁用微分作用。

微分延迟系数

微分延迟系数用于延迟微分作用的生效。

微分延迟 = 微分作用时间 × 微分延迟系数

- 0.0：微分作用仅在一个周期内有效，因此几乎不产生影响。
- 0.5：此值经实践证明对于具有一个优先时间常量的受控系统非常有用。
- > 1.0：系数越大，微分作用的生效时间延迟越久。

比例作用权重

比例作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0：应对设定值变化的比例作用完全有效
- 0.0：应对设定值变化的比例作用无效

当过程值变化时，比例作用始终完全有效。

微分作用权重

微分作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0：设定值变化时微分作用完全有效
- 0.0：设定值变化时微分作用不生效

当过程值变化时，微分作用始终完全有效。

PID 算法采样时间

受控系统需要一定的时间来对输出值的变化做出响应。因此，建议不要在每次循环中都计算输出值。PID 算法的采样时间是两次计算输出值之间的时间。该时间在调节期间进行计算，并舍入为 PID_3Step 采样时间的倍数。PID_3Step 的所有其它功能在每次调用时均执行。

死区宽度

死区可抑制控制器处于稳态的噪声分量。死区宽度指定死区的大小。如果死区宽度为 0.0，则死区关闭。

如果将不等于 1.0 的值组态为比例作用权重或微分作用权重，则即使在死区内，设定值的变化也会影响输出值。
无论权重如何，死区内的过程值变化都不会影响输出值。

5.3.2 调试 PID_3Step V1

5.3.2.1 调试 V1

可在“调节”(Tuning) 工作区中监视设定值、过程值和输出值随时间的变化。 曲线绘图仪支持以下调试功能：

- 控制器预调节
- 控制器精确调节
- 在趋势视图中监视当前闭环控制

所有功能均要求已与 CPU 建立在线连接。

基本处理操作

- 在“采样时间”(Sampling time) 下拉列表中，选择所需的采样时间。

  调节工作区中的所有值将以所选更新时间进行更新。
- 如果要使用调试功能，请单击测量组中的“启动”(Start) 图标。

  将启动值记录操作。 设定值、过程值以及输出值的当前值将输入到趋势视图中。 可以对调试窗口进行操作。
- 如果要结束调试功能，请单击“停止”(Stop) 图标。

  可以继续对趋势视图中记录的值进行分析。
- 关闭调试窗口将终止趋势视图中的记录操作并删除所记录的值。

5.3.2.2 预调节 V1

预调节可确定对输出值脉冲的过程响应，并搜索拐点。 根据受控系统的最大斜率与死时间计算已调节的 PID 参数。

过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。 只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。 重新计算前会备份 PID 参数。

预调节期间冻结设定值。

要求

- 已在循环中断 OB 中调用 PID_3Step 指令。
- ManualEnable = FALSE
- PID_3Step 处于“未激活”或“手动”模式。
- 设定值和过程值均处于组态的限值范围内（请参见“过程值设置”组态）。

步骤

要执行预调节，请按下列步骤操作：

1. 在项目树中双击“PID_3Step > 调试”(PID_3Step > Commissioning) 条目。
2. 在“调节”(Tuning) 工作区的“调节模式”(Tuning mode) 下拉列表中选择条目“预调节”(Pretuning)。
3. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动预调节功能。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及控制器调节功能看似受阻时，请单击“Stop”图标。检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果执行预调节时未产生错误消息，则 PID 参数已调节完毕。PID_3Step 将切换到自动模式并使用已调节的参数。在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果无法实现预调节，PID_3Step 将切换到“未激活”模式。

5.3.2.3 精确调节 V1

精确调节将使过程值出现恒定受限的振荡。 将根据此振荡的幅度和频率为操作点优化 PID 参数。所有 PID 参数都将根据相应结果进行重新计算。精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。

PID_3Step 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。 重新计算前会备份 PID 参数。

精确调节期间冻结设定值。

要求

- 已在循环中断 OB 中调用 PID_3Step 指令。
- ManualEnable = FALSE
- 已对电机转换时间进行了组态或测量。
- 设定值和过程值均处于组态的限值范围内（请参见“过程值设置”组态）。
- 在操作点处，控制回路已稳定。 过程值与设定值一致时，表明到达了操作点。
- 不能被干扰。
- PID_3Step 处于未激活模式、自动模式或手动模式。

过程取决于初始情况

在以下模式下启动精确调节时，具体情况如下所述：

- 自动模式

  如果希望通过控制器调节来改进现有 PID 参数，请在自动模式下启动精确调节。

  PID_3Step 将使用现有的 PID 参数进行调节，直到控制回路已稳定并且精确调节的要求得到满足为止。之后才会启动精确调节。

- 未激活模式或手动模式

  总是先启动预调节。 建立的 PID 参数将用于进行调节，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。

步骤

要执行“精确调节”，请按以下步骤操作：

1. 在“调节模式”(Tuning mode) 下拉列表中选择条目“精确调节”(Fine tuning)。
2. 单击“Start”图标。
   - 将建立在线连接。
   - 将启动值记录操作。
   - 将启动精确调节过程。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。进度条指示当前步骤的进度。

说明

当进度条达到 100% 以及控制器调节功能看似受阻时，请单击“调节模式”(Tuning mode) 组中的“Stop”图标。 检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果已执行精确调节且没有错误，则 PID 参数已得到优化。 PID_3Step 切换到自动模式，并使用优化的参数。 在电源关闭以及重启 CPU 期间，优化的 PID 参数保持不变。

如果精确调节期间出错，PID_3Step 将切换到“未激活”模式。

5.3.2.4 使用手动 PID 参数 V1 进行调试

步骤

要使用手动 PID 参数调试 PID_3Step，请按以下步骤操作：

1. 在项目树中双击“PID_3Step > 组态”(PID_3Step > Configuration)。
2. 在组态窗口中单击“高级设置 > PID 参数”(Advanced settings > PID Parameters)。
3. 选中复选框“启用直接输入”(Enable direct input)。
4. 输入 PID 参数。
5. 在项目树中双击“PID_3Step > 调试”(PID_3Step > Commissioning)。
6. 与 CPU 之间建立在线连接。
7. 将 PID 参数装载到 CPU。
8. 单击“激活控制器”(Activate controller) 图标。

结果

PID_3Step 切换到自动模式，并使用当前 PID 参数进行控制。

5.3.2.5 测量电机转换时间 V1

简介

PID_3Step 要求电机转换时间尽可能准确，以便获得良好的控制器结果。 执行器文档中的数据包含此类执行器的平均值。针对特定执行器的值可能不同。

如果使用提供位置反馈或停止位信号的执行器，则可在调试期间测量电机转换时间。 测量电机转换时间期间，不考虑输出值的限值。 执行器可行进至上端停止位或下端停止位。

如果位置反馈或停止位信号均不可用，则无法测量电机转换时间。

提供模拟位置反馈的执行器

要使用位置反馈测量电机转换时间，请按以下步骤操作：

要求

- 已在基本设置中选择 Feedback 或 Feedback_PER 并且已连接信号。
- 已与 CPU 建立在线连接。

1. 选中“使用位置反馈”(Use position feedback) 复选框。
2. 在“目标位置”(Target position) 输入字段中输入执行器要移动到的位置。

   将显示当前位置反馈（起始位置）。 “目标位置”(Target position) 与“位置反馈”(Position feedback) 之间的差值必须至少为有效输出值范围的 50%。
3. 单击 “启动转换时间测量”(Start transition time measurement) 图标。

结果

将执行器从起始位置移动到目标位置。 立即开始时间测量并在执行器到达目标位置时结束。 根据以下等式计算电机转换时间：

电机转换时间 = (输出值上限 - 输出值下限) × 测量时间/总量 (目标位置 - 起始位置)。

将显示转换时间测量的进度和状态。测得的转换时间保存在 CPU 的背景数据块中，并显示在“测量的转换时间”(Measured transition time) 字段中。转换时间测量完成后，PID_3Step 将切换到“未激活”模式。

说明

单击图标 “上传所测量的转换时间”(Load measured transition time)，将所测量的电机转换时间装载到项目中。

提供停止位信号的执行器

要测量提供停止位信号的执行器的转换时间，请按以下步骤操作：

要求

- 已在基本设置中选中“停止位信号”(Endstop signals) 复选框并且已连接 Actuator_H 和 Actuator_L。
- 已与 CPU 建立在线连接。

要使用停止位信号测量电机转换时间，请按以下步骤操作：

1. 选中“使用执行器停止位信号”(Use actuator endstop signals) 复选框。
2. 选择要在哪个方向上移动执行器。
   - 打开 - 关闭 - 打开

     执行器首先会移动到上端停止位，接着移动到下端停止位，然后返回到上端停止位。
   - 关闭 - 打开 - 关闭

     执行器首先会移动到下端停止位，接着移动到上端停止位，然后返回到下端停止位。
3. 单击 “启动转换时间测量”(Start transition time measurement) 图标。

结果

沿所选方向移动执行器。 时间测量将在执行器到达第一个停止位时启动，而在执行器第二次到达该停止位时结束。 电机转换时间等于所测得的时间除以二。

将显示转换时间测量的进度和状态。测得的转换时间保存在 CPU 的背景数据块中，并显示在“测量的转换时间”(Measured transition time) 字段中。转换时间测量完成后，PID_3Step 将切换到“未激活”模式。

取消转换时间测量

如果取消转换时间测量，PID_3Step 将立即切换到“未激活”模式。 将停止移动执行器。可以在曲线绘图仪中重新激活 PID-3Step。

5.3.3 使用 PLCSIM 仿真 PID_3Step V1

说明

使用 PLCSIM 进行仿真

对于使用 PLCSIM 进行的仿真，仿真 PLC 的时间特性与“真实”PLC 并不完全相同。仿真 PLC 循环中断 OB 的实际周期时钟波动比“真实”PLC 的波动大。

在标准组态中，PID_3Step 会自动确定调用之间的时间，并监视波动情况。

因此，使用 PLCSIM 仿真 PID_3Step 时，可能检测到采样时间错误 (ErrorBits = DW#16#00000800)。

这会导致进行中的调节中止。

自动模式下的响应取决于 ActivateRecoverMode 变量的值。

为防止此类情况发生，应按下列方式为使用 PLCSIM 进行的仿真组态 PID_3Step：

- CycleTime.EnEstimation = FALSE
- CycleTime.EnMonitoring = FALSE
- CycleTime.Value：以秒为单位为此变量分配调用循环中断 OB 的周期时钟。

使用 PID_Temp

6

6.1 工艺对象 PID_Temp

PID_Temp 工艺对象提供具有集成调节功能的连续 PID 控制器。PID_Temp 专为温度控制而设计，适用于加热或加热/制冷应用。 为此提供了两路输出，分别用于加热和制冷。PID_Temp 还可以用于其它控制任务。PID_Temp 可以级联，可以在手动或自动模式下使用。

PID_Temp 可连续采集在控制回路内测量的过程值并将其与所设置的设定值进行比较。指令 PID_Temp 将根据生成的控制偏差计算加热和/或制冷的输出值，而该值用于将过程值调整到设定值。 PID 控制器的输出值由三种作用构成：

- 比例作用

  输出值的比例作用与控制偏差成比例增加。

- 积分作用

  输出值的积分作用一直增加，直到控制偏差达到平衡状态。

- 微分作用

  微分作用随控制偏差的变化率而增加。 过程值会尽快校正到设定值。 如果控制偏差的变化率下降，则微分作用将再次减弱。

指令 PID_Temp 在“预调节”期间计算受控系统的比例、积分和微分参数。 “精确调节”可用于进一步调节这些参数。 用户不必手动确定这些参数。

可以为加热和制冷应用使用一个固定的制冷系数或两个 PID 参数集。

更多信息

- 软件控制器概述 (页 41)
- 添加工艺对象 (页 44)
- 组态工艺对象 (页 48)
- 组态 PID_Temp (页 172)

6.2 组态 PID_Temp

6.2.1 基本设置

6.2.1.1 简介

在巡视窗口或组态窗口的“基本设置”(Basic settings) 下，组态工艺对象“PID_Temp”的以下属性：

- 物理量
- 复位后的启动行为
- 设定值的来源和输入（仅在巡视窗口中）
- 过程值的选择
- 过程值的来源和输入（仅在巡视窗口中）
- 加热输出值的选择
- 加热输出值的来源和输入（仅在巡视窗口中）
- 制冷输出值的激活和选择
- 制冷输出值的来源和输入（仅在巡视窗口中）
- PID_Temp 激活为级联的主控制器或从控制器
- 从控制器的数量
- 主控制器的选择（仅在巡视窗口中）

设定值、过程值、加热输出值和制冷输出值

可以在程序编辑器的巡视窗口中为设定值、过程值、加热输出值和制冷输出值选择来源或为其输入值或变量。

为每个值选择一个源：

- 背景数据块：

  使用背景数据块中保存的值。 必须通过用户程序在背景 DB 中更新值。 指令中不应有值。 可以使用 HMI 进行更改。

- 指令：

  使用与指令相连的值。 每次调用指令时都会将值写入背景数据块。 无法使用 HMI 进行更改。

6.2.1.2 控制器类型

物理量

在“控制器类型”(Controller type) 组中，为设定值和过程值选择测量单位和物理量。 设定值和过程值将以该测量单位显示。

启动特性

1. 要在 CPU 重启后切换到“未激活”模式，请清除“CPU 重启后激活 Mode”(Activate Mode after CPU restart) 复选框。

   要在 CPU 重启后切换到“模式”(Mode) 参数中保存的工作模式，请选中“CPU 重启后激活 Mode”(Activate Mode after CPU restart) 复选框。

2. 在“将 Mode 设置为”(Set Mode to) 下拉列表中，选择在执行完整下载到设备后要启用的模式。

   执行完整下载到设备后，PID_Temp 将以所选工作模式启动。 以后每次重启时，PID_Temp 都以上次保存在“模式”(Mode) 中的模式启动。

   选择预调节或精确调节时，还必须设置或复位 Heat.EnableTuning 与 Cool.EnableTuning 变量，以便在加热调节和制冷调节之间选择。

示例：

您已选中“CPU 重启后激活 Mode”(Activate Mode after CPU restart) 复选框以及“将 Mode 设置为”(Set Mode to) 列表中的“预调节”(Pretuning) 条目。 在执行了完整“下载到设备”后，PID_Temp 将以“预调节”模式启动。 如果预调节仍处于激活状态，则 PID_Temp 在 CPU 重启后再次以“预调节”模式启动（加热/制冷取决于变量 Heat.EnableTuning 和 Cool.EnableCooling）。 如果预调节已成功完成并且自动模式处于激活状态，则 PID_Temp 在 CPU 重启后将以“自动模式”启动。

6.2.1.3 设定值

步骤

要定义固定设定值，请按以下步骤操作：

1. 选择“背景 DB”(Instance DB)。
2. 输入一个设定值，例如 80° C。
3. 删除指令中的任何条目。

要定义可变设定值，请按以下步骤操作：

1. 选择“指令”(Instruction)。
2. 输入保存设定值的 REAL 变量的名称。

   可通过程序控制的方式为该 REAL 变量分配各种值，例如，采用时间控制的方式来更改设定值。

6.2.1.4 过程值

如果直接使用模拟量输入值，则 PID_Temp 会将该模拟量输入值标定为物理量。

如果要预先处理一下该模拟量输入值，则需要编写一个处理程序。 例如，过程值与模拟量输入值并不成正比。 经过处理的过程值必须为浮点格式。

步骤

要使用未经处理的模拟量输入值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input_PER”。
2. 选择“指令”(Instruction) 作为源。
3. 输入模拟量输入的地址。

要使用经过处理的浮点格式的过程值，请按以下步骤操作：

1. 在下拉列表“Input”中选择条目“Input”。
2. 选择“指令”(Instruction) 作为源。
3. 输入变量的名称，用来保存经过处理的过程值。

6.2.1.5 加热和制冷输出值

PID_Temp 指令提供了一种可对温度过程进行集成调节的 PID 控制器。 PID_Temp 适用于加热或加热和制冷应用。

PID_Temp 提供以下输出值。 执行器将决定要使用的输出值。

- OutputHeat

  加热输出值（浮点格式）： 由于执行器响应为非线性等原因，需要通过用户程序来处理加热的输出值。

- OutputHeat_PER

  模拟量加热输出值： 通过模拟量输出触发加热执行器，并使用连续信号（如 0...10 V、4...20 mA）控制加热执行器。

- OutputHeat_PWM

  脉宽调制加热输出值： 通过数字量输出控制加热执行器。 脉宽调制可生成不同的 ON 和 OFF 时间。

- OutputCool

  制冷输出值（浮点格式）： 例如，由于执行器响应是非线性的，因而需要通过用户程序来处理制冷的输出值。

- OutputCool_PER

  模拟量制冷输出值： 通过模拟量输出触发制冷执行器，并使用连续信号（如 0...10 V、4...20 mA）控制制冷执行器。

- OutputCool_PWM

  脉宽调制制冷输出值： 通过数字量输出控制制冷执行器。 脉宽调制可生成不同的 ON 和 OFF 时间。

制冷输出仅在通过“激活制冷”(Activate cooling) 复选框激活后可用。

- 如果清除该复选框，PID 算法的输出值 (PidOutputSum) 将在标定后在加热输出中输出。
- 如果选中该复选框，PID 算法的正输出值 (PidOutputSum) 将在标定后在加热输出中输出。 PID 算法的负输出值则在标定后在制冷输出中输出。 还可以在输出设置中从两种输出值计算方法中选择。

说明

注意：

- 只有从下拉列表中选择 OutputHeat_PWM、OutputHeat_PER、OutputCool_PWM、OutputCool_PER 输出后，才会相应计算这些输出。
- 始终会计算 OutputHeat 输出。
- 如果选中控制制冷的复选框，将计算 OutputCool 输出。
- 只有控制器不是组态成级联中的主控制器时，“激活制冷”(Activate cooling) 复选框才可用。

步骤

要使用模拟量输出值，请按以下步骤操作：

1. 在“OutputHeat”或“OutputCool”下拉列表中选择“OutputHeat_PER”或“OutputCool_PER”条目。
2. 选择“指令”(Instruction)。
3. 输入模拟量输出的地址。

要使用脉宽调制输出值，请按以下步骤操作：

1. 在“OutputHeat”或“OutputCool”下拉列表中选择“OutputHeat_PWM”或“OutputCool_PWM”条目。
2. 选择“指令”(Instruction)。
3. 输入数字量输出的地址。

要使用用户程序来处理输出值，请按以下步骤操作：

1. 在“OutputHeat”或“OutpuCool”下拉列表中选择“OutputHeat”或“OutputCool”条目。
2. 选择“指令”(Instruction)。
3. 输入用于处理输出值的变量的名称。
4. 通过模拟量或数字量 CPU 输出将经过处理的输出值传送到执行器。

6.2.1.6 级联

如果 PID_Temp 实例从上级主控制器接收设定值，并转而将其输出值输出到从属从控制器，则此 PID_Temp 实例既为主控制器又为从控制器。对于此类 PID_Temp 实例，必须执行下文列出的两种组态。例如，具有三个级联连接测量变量和三个 PID_Temp 实例的级联控制系统中，中间的 PID_Temp 实例便属于此种情况。

将控制器组态为级联中的主控制器

主控制器通过其输出定义从控制器的设定值。

要将 PID_Temp 用作级联中的主控制器，必须在基本设置中禁用制冷。要将此 PID_Temp 实例组态成级联中的主控制器，请激活“控制器为主控制器”(Controller is master) 复选框。加热输出值的选择将自动设置为 OutputHeat。

无法在级联的主控制器上使用 OutputHeat_PWM 和 OutputHeat_PER。

随后，指定从该主控制器接收设定值的直接从属从控制器的数目。

将主控制器的 OutputHeat 参数分配给从控制器的 Setpoint 参数时，如果未使用用户自己的标定功能，则可能需要根据从控制器的设定值/过程值范围调整主控制器的输出值限值和输出标定。可以在主控制器输出设置的“OutputHeat / OutputCool”部分执行调整。

将控制器组态为级联中的从控制器

从控制器从其主控制器的输出（OutputHeat 参数）中接收其设定值（Setpoint 参数）。

要将此 PID_Temp 实例组态成级联中的从控制器，请在基本设置中激活“控制器为从控制器”(Controller is slave) 复选框。

随后，在编程编辑器的巡视窗口中，为该从控制器选择选择要用作其主控制器的 PID_Temp 实例。从控制器的 Master 和 Setpoint 参数随即与所选主控制器互连（将覆盖这些参数的既有互连）。在主控制器与从控制器之间便通过此互连交换信息和指定设定值。如有必要，以后可以在从控制器的 Setpoint 参数中更改此互连，例如，另外插入一个滤波器。之后不可更改 Master 参数处的互连。

对于所选主控制器，必须选中“控制器为主控制器”(Controller is master) 复选框，且必须正确组态从控制器的数量。在同一循环中断 OB 中，必须先调用主控制器，再调用从控制器。

更多信息

更多有关在级联控制系统中使用 PID_Temp 时的程序创建、组态和调试信息，请参见使用 PID_Temp 的级联控制 (页 208)。

6.2.2 过程值设置

6.2.2.1 过程值的限值

必须为过程值指定正确的绝对上限和绝对下限，作为受控系统的限值。 只要过程值超出这些限值，就会出现错误 (ErrorBits = 0001h)。 如果超出过程值的限值，则取消调节操作。 可以在输出设置中指定 PID_Temp 在自动模式下对错误的响应方式。

6.2.2.2 过程值标定

如果已在基本设置中对 Input_PER 的使用进行了组态，则需要将模拟量输入值转换为过程值的物理量。当前组态将显示在 Input_PER 画面中。

如果过程值与模拟量输入值成正比，则使用上下限值对来标定 Input_PER。

步骤

要标定过程值，请按下列步骤操作：

1. 在“标定的过程值的下限”(Scaled low process value) 和“下限”(Low) 输入字段中输入一对下限值。
2. 在“标定的过程值的上限”(Scaled high process value) 和“上限”(High) 输入字段中输入一对上限值。

这些值对的默认设置保存在硬件配置中。要使用硬件配置中的值对，请按以下步骤操作：

1. 在程序编辑器中选择指令 PID_Temp。
2. 在基本设置中将 Input_PER 与模拟量输入互连。
3. 在过程值设置中单击“自动设置”(Automatic setting) 按钮。

硬件配置中的值将覆盖现有值。

6.2.3 输出设置

6.2.3.1 输出的基本设置

加热和制冷的方法

如果在基本设置中激活制冷，则有两种方法可用于计算 PID 输出值：

- PID 参数切换 (Config.AdvancedCooling = TRUE)：

  通过单独的 PID 参数集来计算制冷的输出值。PID 算法将根据计算出的输出值和控制偏差确定使用加热过程还是制冷过程的 PID 参数。此方法适用于加热执行器和制冷执行器的时间响应和增益都不同的情况。

  仅在选择该方法后才可对制冷进行预调节和精确调节。

- 制冷系数 (Config.AdvancedCooling = FALSE)：

  通过加热过程的 PID 参数并考虑可组态的制冷系数 Config.CoolFactor 来执行制冷输出值计算。此方法适用于加热执行器和制冷执行器的时间响应相似但增益不同的情况。选择该方法时，无法对制冷进行预调节和精确调节并且控制制冷的 PID 参数集不可用。只能执行加热调节。

制冷系数

如果选择制冷系数作为加热/制冷方法，则在制冷的输出值计算中将使用此系数。因此，可以考虑加热执行器与制冷执行器增益不同的情况。

制冷系数既不会自动进行设置，也不会在调节期间进行调整。必须通过“加热执行器增益/制冷执行器增益”的比值手动组态正确的制冷系数。

示例：制冷系数为 2.0 表示加热执行器增益是制冷执行器增益的两倍。

只有选择“制冷系数”(Cooling factor) 作为加热/制冷方法时，制冷系数才有效并且才可以更改。

对错误的响应

注意
您的系统可能已损坏。 如果在出现错误时输出“错误未决时的当前值”或“错误未决时的替代输出值”，PID_Temp 将保持自动模式或手动模式。这可能导致超出过程值限值并损坏系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

PID_Temp 需要预设置，以便在发生错误时，控制器在大多数情况下均可保持激活状态。

如果在控制器模式下频繁发生错误，则该默认响应会对控制响应产生负面影响。这种情况下，检查 ErrorBits 参数并消除错误原因。

PID_Temp 会生成可设定的输出值来对错误做出响应：

- 零（未激活）

  在所有错误情况下，PID_Temp 都切换到“未激活”工作模式并输出以下值：

  – 输出 0.0 作为 PID 输出值 (PidOutputSum)
  – 输出 0.0 作为加热输出值 (OutputHeat) 和制冷输出值 (OutputCool)
  – 输出 0 作为加热的模拟量输出值 (OutputHeat_PER) 和制冷的模拟量输出值 (OutputCool_PER)
  – 输出 FALSE 作为加热的 PWM 输出值 (OutputHeat_PWM) 和制冷的 PWM 输出值 (OutputCool_PWM)

  这与输出值限值和标定的组态无关。只能通过 Reset 的下降沿或 ModeActivate 的上升沿重新激活控制器。

- 错误未决时的当前值

  错误响应取决于发生的错误和工作模式。

  如果发生一个或多个下列错误，则 PID_Temp 停留在自动模式下：

  – 0000001h：参数 Input 超出了过程值限值的范围。
  – 0000800h：采样时间错误
  – 0040000h：Disturbance 参数的值无效。
  – 8000000h：计算 PID 参数期间出错。

如果在自动模式下发生一个或多个下列错误，PID_Temp 将切换到“含错误监视功能的替代输出值”模式并输出上一个有效 PID 输出值 (PidOutputSum)：

– 0000002h：Input_PER 参数的值无效。

– 0000200h：Input 参数的值无效。

– 0000400h：输出值计算失败。

– 0001000h：Setpoint 或 SubstituteSetpoint 参数的值无效。

在应用 PID 输出值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况。

当错误不再处于未决状态时，PID_Temp 切换回自动模式。

如果在手动模式下发生错误，PID_Temp 保持手动模式并继续使用手动值作为 PID 输出值。

如果手动值无效，则使用组态的替代输出值。

如果手动值和替代输出值都无效，则使用加热过程的 PID 输出值下限 (Config.Output.Heat.PidLowerLimit)。

如果在预调节或精确调节期间出现下列错误，PID_Temp 将保持激活模式：

– 0000020h：精确调节期间不允许预调节。

出现其它错误时，PID_Temp 将取消调节并切换到启动调节时的模式。

- 错误未决时的替代输出值

  在“含错误监视功能的替代输出值”工作模式下，PID_Temp 按照“错误未决时的当前值”中的描述操作，但输出组态的替代输出值 (SubstituteOutput) 作为 PID 输出值 (PidOutputSum)。

  在应用 PID 输出值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况。

  对于激活了制冷输出 (Config.ActivateCooling = TRUE) 的控制器，请输入：

  – 正的替换输出值以在加热输出上输出该值。

  – 负的替换输出值以在制冷输出上输出该值。

  如果发生下列错误，PID_Temp 将保持“含错误监视功能的替代输出值”模式，并输出加热过程的 PID 输出值下限 (Config.Output.Heat.PidLowerLimit)：

  – 0020000h：变量 SubstituteOutput 的值无效。

6.2.3.2 输出值限值和标定

根据具体的工作模式，PID 输出值 (PidOutputSum) 或是通过 PID 算法自动计算，或是使用手动值 (ManualValue)或已组态的替代输出值 (SubstituteOutput) 来计算。

根据组态限制 PID 输出值：

- 如果在基本设置中禁用制冷 (Config.ActivateCooling = FALSE)，该值将限制在 PID 输出值的上限（加热）(Config.Output.Heat.PidUpperLimit) 和 PID 输出值的下限（加热）(Config.Output.Heat.PidLowerLimit) 之间。

  标定特征线的水平轴上的两个限值都可以在“OutputHeat/OutputCool”部分进行组态。它们将显示在“OutputHeat_PWM/OutputCool_PWM”和“OutputHeat_PER/OutputCool_PER”部分，但无法更改。

- 如果在基本设置中激活制冷 (Config.ActivateCooling = TRUE)，该值将限制在 PID 输出值上限 (Config.Output.Heat.PidUpperLimit) 和 PID 输出值下限（制冷）(Config.Output.Cool.PidLowerLimit) 之间。

  标定特征线的水平轴上的两个限值都可以在“OutputHeat/OutputCool”部分进行组态。它们将显示在“OutputHeat_PWM/OutputCool_PWM”和“OutputHeat_PER/OutputCool_PER”部分，但无法更改。

  PID 输出值下限（加热）(Config.Output.Heat.PidLowerLimit) 和 PID 输出值上限（制冷）(Config.Output.Cool.PidUpperLimit) 无法更改，且必须分配为值 0.0。

PID 输出值经过标定在加热和制冷输出中输出。可以为每个输出单独指定标定，并且使用 2 个值对来指定，每个值对都由一个 PID 输出值限值和一个标定值组成。

输出	值对	参数
OutputHeat	值对 1	PID 输出值上限（加热） Config.Output.Heat.PidUpperLimit， 标定的输出上限值（加热） Config.Output.Heat.UpperScaling
	值对 2	PID 输出值下限（加热） Config.Output.Heat.PidLowerLimit， 标定的输出下限值（加热） Config.Output.Heat.LowerScaling
OutputHeat_PWM	值对 1	PID 输出值上限（加热） Config.Output.Heat.PidUpperLimit， 标定的 PWM 输出上限值（加热） Config.Output.Heat.PwmUpperScaling

输出	值对	参数
	值对 2	PID 输出值下限（加热） Config.Output.Heat.PidLowerLimit， 标定的 PWM 输出下限值（加热） Config.Output.Heat.PwmLowerScaling
OutputHeat_PER	值对 1	PID 输出值上限（加热） Config.Output.Heat.PidUpperLimit， 标定的模拟量输出上限值（加热） Config.Output.Heat.PerUpperScaling
	值对 2	PID 输出值下限（加热） Config.Output.Heat.PidLowerLimit， 标定的模拟量输出下限值（加热） Config.Output.Heat.PerLowerScaling
OutputCool	值对 1	PID 输出值下限（制冷） Config.Output.Cool.PidLowerLimit， 标定的输出上限值（制冷） Config.Output.Cool.UpperScaling
	值对 2	PID 输出值上限（制冷） Config.Output.Cool.PidUpperLimit， 标定的输出下限值（制冷） Config.Output.Cool.LowerScaling
OutputCool_PWM	值对 1	PID 输出值下限（制冷） Config.Output.Cool.PidLowerLimit， 标定的 PWM 输出上限值（制冷） Config.Output.Cool.PwmUpperScaling
	值对 2	PID 输出值上限（制冷） Config.Output.Cool.PidUpperLimit， 标定的 PWM 输出下限值（制冷） Config.Output.Cool.PwmLowerScaling

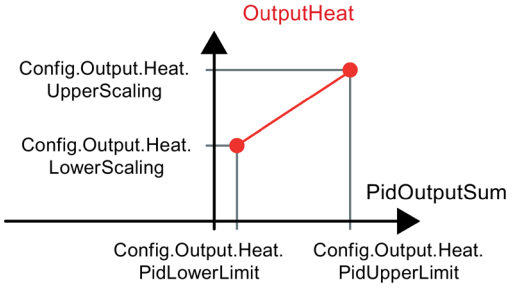
输出	值对	参数
OutputCool_PER	值对 1	PID 输出值下限（制冷） Config.Output.Cool.PidLowerLimit， 标定的模拟量输出上限值（制冷） Config.Output.Cool.PerUpperScaling
	值对 2	PID 输出值上限（制冷） Config.Output.Cool.PidUpperLimit， 标定的模拟量输出下限值（制冷） Config.Output.Cool.PerLowerScaling

如果已激活制冷 (Config.ActivateCooling = TRUE)，则 PID 输出值下限（加热）(Config.Output.Heat.PidLowerLimit) 必须为 0.0。

PID 输出值上限（制冷）Config.Output.Cool.PidUpperLimit) 必须始终为 0.0。

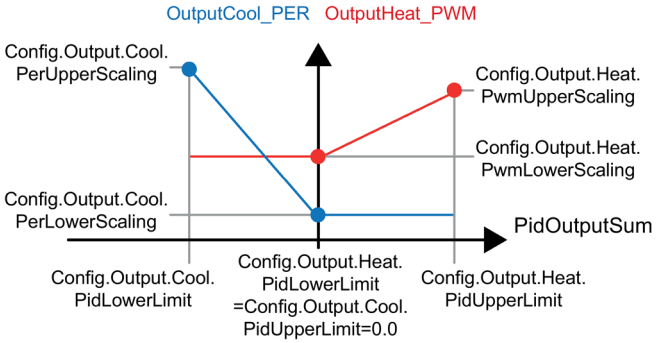
示例：

使用 OutputHeat 输出时的输出标定。前提是制冷已禁用，PID 输出值的下限（加热）(Config.Output.Heat.PidLowerLimit) 可以不等于 0.0。



示例：

使用 OutputHeat_PWM 和 OutputCool_PER 输出时的输出标定。前提是制冷已激活，PID 输出值的下限（加热）(Config.Output.Heat.PidLowerLimit) 必须为 0.0。



除在“未激活”工作模式下外，输出处的值始终介于标定的输出上限值与标定的输出下限值之间，例如 OutputHeat 始终介于标定的输出上限值（加热）(Config.Output.Heat.UpperScaling) 和标定的输出下限值（加热）(Config.Output.Heat.LowerScaling) 之间。

如果要限制相关输出中的值，因此还必须调整这些标定值。

您可以组态标定特征线纵轴上的输出标定值。每个输出都有两个单独的标定值。只能更改 OutputHeat_PWM, OutputCool_PWM、OutputHeat_PER 和 OutputCool_PER 输出的标定值，而且前提是在基本设置中选择了相应输出。另外，还必须在基本设置中为所有制冷输出激活制冷。

不管基本设置中的所选输出为何，调试对话框中的趋势视图都只记录 OutputHeat 和 OutputCool 的值。因此，如果使用 OutputHeat、OutputCool、OutputHeat_PWM 或 OutputHeat_PER 并且想要在调试对话框中使用趋势视图，则可根据需要调整 OutputCool_PWM 或 OutputCool_PER 的标定值。

6.2.4 高级设置

6.2.4.1 过程值监视

在“过程值监视”(Process value monitoring) 组态窗口中，组态过程值的警告上限和下限。如果在运行期间超出或低于其中一个警告限值，则将在 PID_Temp 指令的以下参数中显示一条警告：

- 输出参数 InputWarning_H，前提是超出警告上限
- 输出参数 InputWarning_L，前提是低于警告下限

警告限值必须处于过程值的限值范围内。

如果未输入警告限值，则使用过程值的上限和下限。

示例

过程值上限 = 98° C；警告上限 = 90° C

警告下限 = 10° C；过程值下限 = 0° C

PID_Temp 将按如下方式响应：

过程值	InputWarning_H	InputWarning_L	ErrorBits
> 98 °C	TRUE	FALSE	0001h
≤ 98° C 且 > 90° C	TRUE	FALSE	0000h
≤ 90° C 且 ≥ 10° C	FALSE	FALSE	0000h
< 10° C 且 ≥ 0° C	FALSE	TRUE	0000h
< 0° C	FALSE	TRUE	0001h

可以在输出设置中组态超出过程值上限或下限时 PID_Temp 的响应。

6.2.4.2 PWM 限值

PID 输出值 PidOutputSum 在标定后通过脉宽调制转换成脉冲串在 OutputHeat_PWM 或 OutputCool_PWM 输出参数中输出。“PID 算法的采样时间”是两次计算 PID 输出值之间的时间。该采样时间用作脉宽调制的时间。

加热期间，在“加热的 PID 算法采样时间”内始终会计算 PID 输出值。

制冷期间的 PID 输出值计算取决于在“基本设置 > 输出”(Basic settings > Output) 中选择的制冷类型：

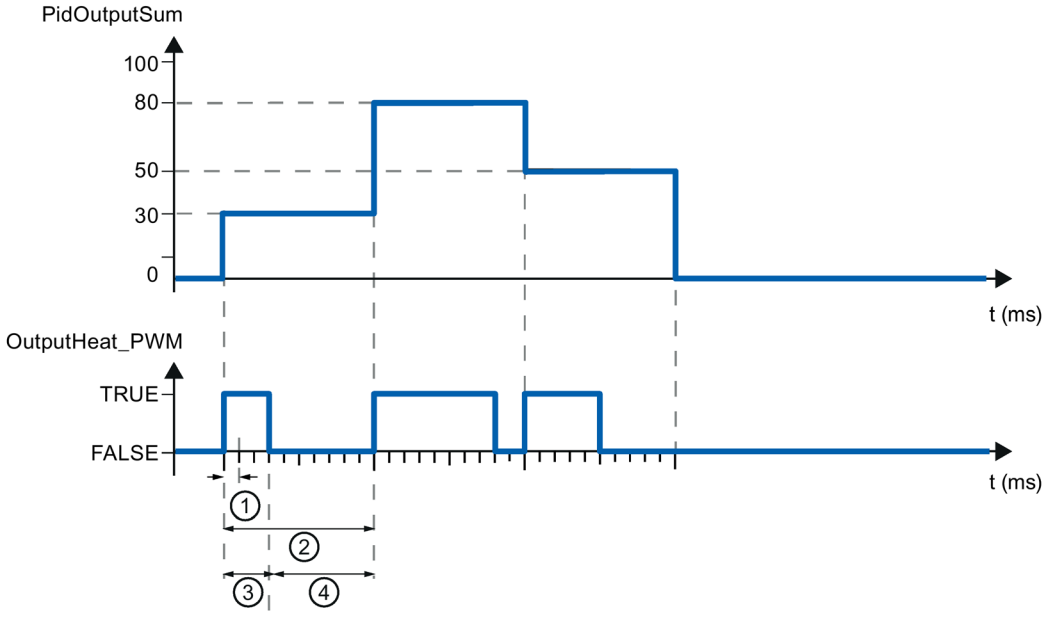
- 如果使用制冷系数，则“加热的 PID 算法采样时间”适用。
- 如果使用 PID 参数切换，则“制冷的 PID 算法采样时间”适用。

OutputHeat_PWM 和 OutputCool_PWM 在 PID_Temp 采样时间（等于调用 OB 的周期时间）内输出。

在预调节或精确调节期间确定加热或制冷的 PID 算法采样时间。如果手动设置 PID 参数，则还需要组态加热或制冷的 PID 算法采样时间。PID_Temp 采样时间等于调用 OB 的周期时间。

脉冲宽度与 PID 输出值成比例并始终为 PID_Temp 采样时间的整数倍。

OutputHeat_PWM 的示例



① PID_Temp 采样时间
② 加热的 PID 算法采样时间
③ 脉冲持续时间
④ 中断时间

可以分别为加热和制冷设置“最短开启时间”和“最短关闭时间”，这两个时间将舍入为 PID_Temp 采样时间的整数倍。

脉冲或中断时间永远不会小于最短开关时间。在下一个周期中累加和补偿由此引起的误差。

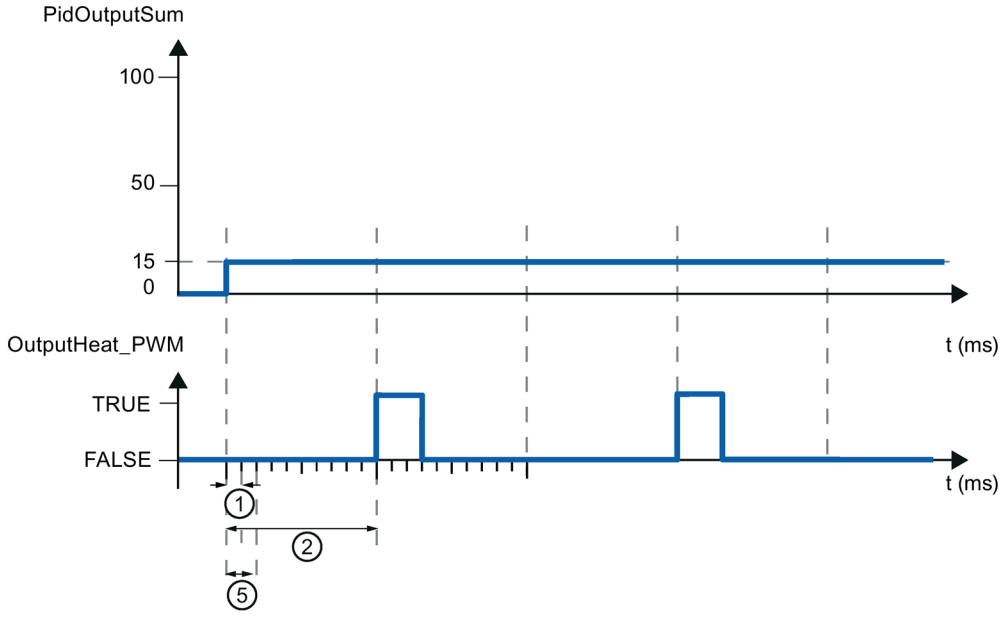
OutputHeat_PWM 的示例

PID_Temp 采样时间 = 100 ms

PID 算法采样时间 = 1000 ms

最短开启时间 = 200 ms

PID 输出值 PidOutputSum 总计为 15% 并保持不变。PID_Temp 可输出的最短脉冲为 20%。在第一个周期内不输出脉冲。在第二个周期内，将第一个周期内未输出的脉冲累加到第二个周期的脉冲。



① PID_Temp 采样时间
② 加热的 PID 算法采样时间
⑤ 最短 ON 时间

为最大程度地减小工作频率并节省执行器，可延长最短开关时间。

如果已在基本设置中选择 OutputHeat/OutputCool 或 OutputHeat_PER/OutputCool_PER 作为输出，将不评估最短开启时间和最短关闭时间，并且也无法更改这两个时间。

使用 OutputHeat_PWM 或 OutputCool_PWM 时，如果“PID 算法采样时间”(Retain.CtrlParams.Cool.Cycle 或 Retain.CtrlParams.Cool.Cycle) 和脉宽调制的持续时间过长，则可在 Config.Output.Heat.PwmPeriode 或 Config.Output.Cool.PwmPeriode 参数中指定一个存在偏差的较短的持续时间，以改善过程值的平滑度（另请参见 PwmPeriode 变量 (页 503)）。

说明

最短开关时间只影响输出参数 OutputHeat_PWM 或 OutputCool_PWM，不用于 CPU 中集成的任何脉冲发生器。

6.2.4.3 PID 参数

PID 参数显示在“PID 参数”(PID Parameters) 组态窗口中。

如果在基本设置中已激活制冷，并且在输出设置中将 PID 参数切换选作加热/制冷方法，则可使用两个参数集：一个用于加热，另一个用于制冷。

这种情况下，PID 算法将根据计算出的输出值和控制偏差确定使用用于加热的 PID 参数还是用于制冷的 PID 参数。

如果禁用制冷，或将制冷系数选作加热/制冷方法，则始终使用用于加热的参数集。

在调节过程中，除了死区宽度必须手动组态以外，其余 PID 参数会根据受控系统进行调整。

PID_Temp 是一种具有抗积分饱和功能并且能够对比例作用和微分作用进行加权的 PIDT1 控制器。

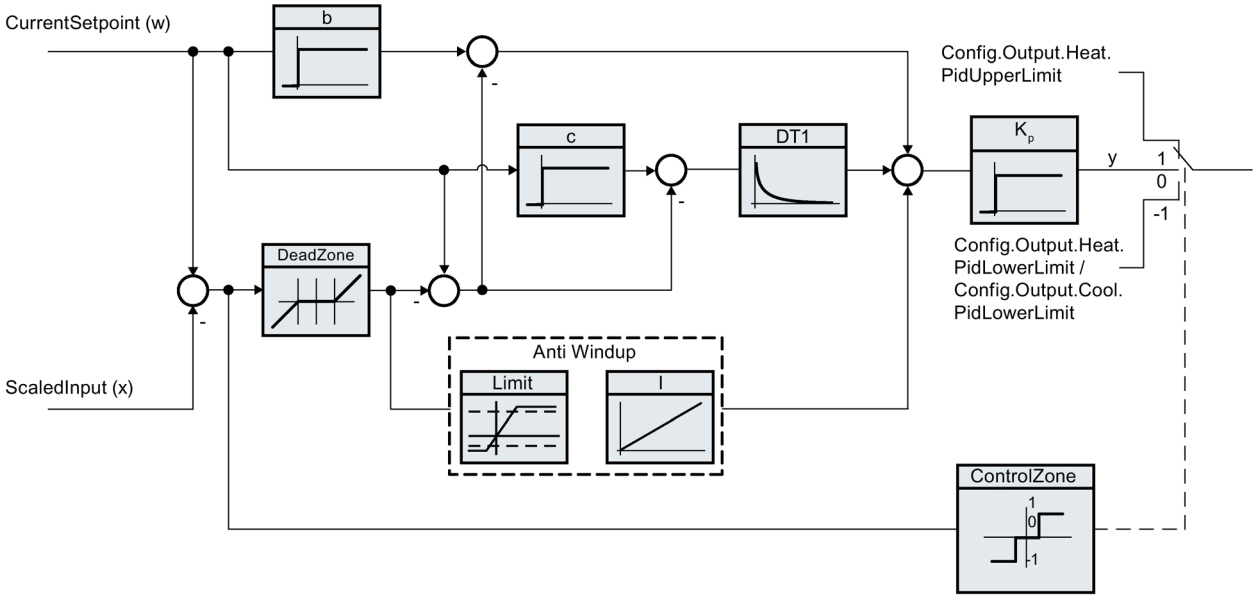
PID 算法根据以下等式工作（控制区和死区已禁用）：

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明	PID_Temp 指令的关联参数
y	PID 算法的输出值	-
K_p	比例增益	Retain.CtrlParams.Heat.Gain Retain.CtrlParams.Cool.Gain CoolFactor
s	拉普拉斯运算符	-
b	比例作用权重	Retain.CtrlParams.Heat.PWeighting Retain.CtrlParams.Cool.PWeighting
w	设定值	CurrentSetpoint
x	过程值	ScaledInput
T_I	积分作用时间	Retain.CtrlParams.Heat.Ti Retain.CtrlParams.Cool.Ti
T_D	微分作用时间	Retain.CtrlParams.Heat.Td Retain.CtrlParams.Cool.Td
a	微分作用延迟系数 （微分延迟 T1 = a × T_D）	Retain.CtrlParams.Heat.TdFiltRatio Retain.CtrlParams.Cool.TdFiltRatio

符号	说明	PID_Temp 指令的关联参数
c	微分作用权重	Retain.CtrlParams.Heat.DWeighting Retain.CtrlParams.Cool.DWeighting
DeadZone	死区宽度	Retain.CtrlParams.Heat.DeadZone Retain.CtrlParams.Cool.DeadZone
ControlZone	控制区宽度	Retain.CtrlParams.Heat.ControlZone Retain.CtrlParams.Cool.ControlZone

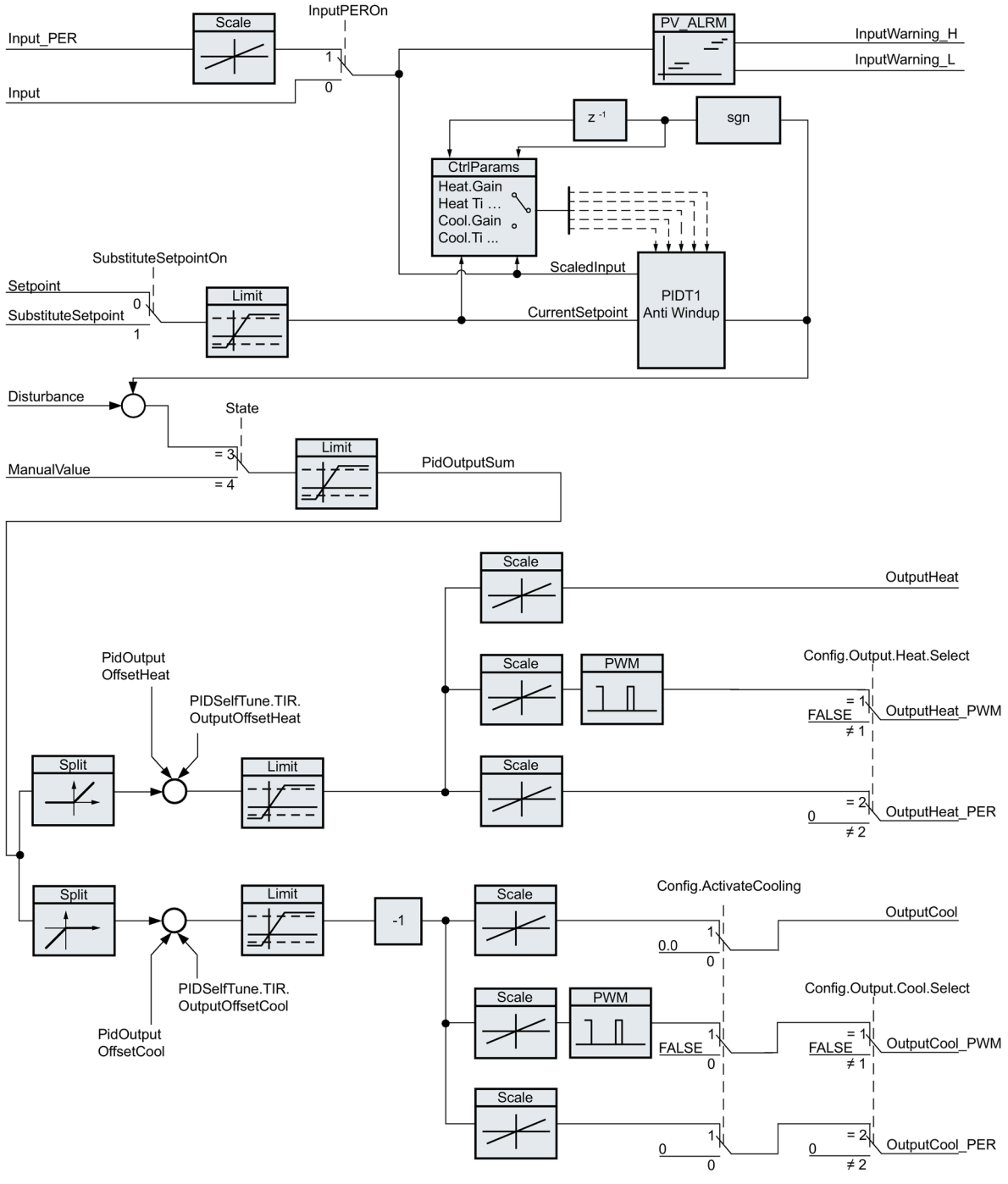
下图说明了集成到 PID 算法中的参数：



所有 PID 参数均具有保持性。如果手动输入 PID 参数，则必须完整下载 PID_Temp (将工艺对象下载到设备 (页 76))。

PID_Temp 方框图

以下方框图说明如何在 PID_Temp 中集成 PID 算法。



比例增益

该值用于指定控制器的比例增益。PID_Temp 运行时不使用负比例增益，且只支持常规控制方向，也就是说过程值会随着 PID 输出值 (PidOutputSum) 的增加而增加。

积分作用时间

积分作用时间用于确定积分作用的时间特性。积分作用时间 = 0.0 时，将禁用积分作用。

微分作用时间

微分作用时间用于确定微分作用的时间特性。微分作用时间 = 0.0 时，将禁用微分作用。

微分延迟系数

微分延迟系数用于延迟微分作用的生效。

微分延迟 = 微分作用时间 × 微分延迟系数

- 0.0：微分作用仅在一个周期内有效，因此几乎不产生影响。
- 0.5：实践证明，该值对具有一个主时间常数的受控系统很有效。
- > 1.0：系数越大，微分作用的生效时间延迟越久。

比例作用权重

改变设定值有可能削弱比例作用。

允许使用 0.0 到 1.0 之间的值。

- 1.0：应对设定值变化的比例作用完全有效
- 0.0：应对设定值变化的比例作用无效

当过程值变化时，比例作用始终完全有效。

微分作用权重

微分作用随着设定值的变化而减弱。

允许使用 0.0 到 1.0 之间的值。

- 1.0：设定值变化时微分作用完全有效
- 0.0：设定值变化时微分作用不生效

当过程值变化时，微分作用始终完全有效。

PID 算法采样时间

受控系统需要一定的时间来对输出值的变化做出响应。因此，建议不要在每次循环中都计算输出值。“PID 算法”的采样时间是两次计算 PID 输出值之间的时间。该时间在调节期间进行计算，并舍入为 PID_Temp 采样时间的倍数（循环中断 OB 的循环时间）。PID_Temp 的所有其它功能会在每次调用时执行。

如果使用 OutputHeat_PWM 或 OutputCool_PWM，PID 算法的采样时间将用作脉宽调制的持续时间。输出信号的精度由 PID 算法采样时间与 OB 的周期时间之比来确定。该周期时间不应超出 PID 算法采样时间的十分之一。

用作 OutputCool_PWM 脉宽调制持续时间的 PID 算法采样时间取决于在“输出基本设置”中选择的加热/制冷方法：

- 如果使用制冷系数，则“加热的 PID 算法采样时间”同样适用于 OutputCool_PWM。
- 如果使用 PID 参数切换，则“制冷的 PID 算法采样时间”可用作 OutputCool_PWM 的持续时间。

如果使用 OutputHeat_PWM 或 OutputCool_PWM 时 PID 算法采样时间和脉宽调制的持续时间过长，则可在 Config.Output.Heat.PwmPeriode 或 Config.Output.Cool.PwmPeriode 参数中指定一个存在偏差的较短的持续时间，以改善过程值的平滑度。

死区宽度

如果过程值受到噪声影响，则噪声也会对输出值产生影响。当控制器增益较高并且激活微分作用时，输出值会出现明显的波动。如果过程值位于设定值附近的死区内，则控制偏差会受到抑制，这样 PID 算法就不会做出响应并且会减少输出值不必要的波动。

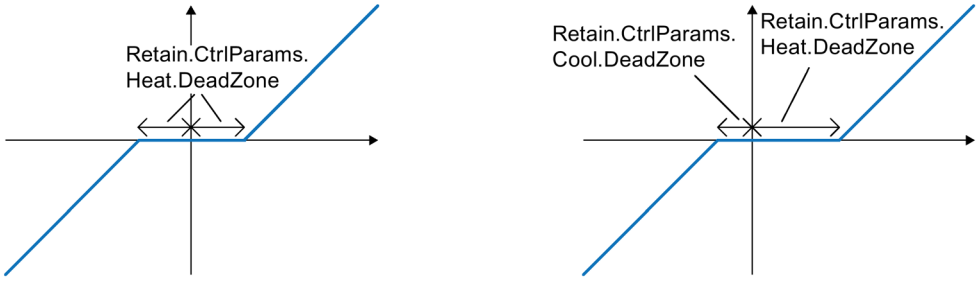
在调节过程中，加热或制冷过程的死区宽度不会自动设置。必须手动对死区宽度进行正确组态。如果将死区宽度设置为 0.0，会禁用死区。

启用死区时，结果可能是永久控制偏差（设定值与实际值之间的偏差）。这可能对精确调节产生负面影响。

如果已在基本设置中激活了制冷，并且在输出设置中将 PID 参数切换选作加热/制冷方法，则死区位于“设定值 - 死区宽度（加热）”和“设定值 + 死区宽度（制冷）”之间。

如果已在基本设置中禁用了制冷，或使用了制冷系数，则死区对称地位于“设定值 - 死区宽度（加热）”和“设定值 + 死区宽度（加热）”之间。

如果将不等于 1.0 的值组态为比例作用权重或微分作用权重，则即使在死区内，设定值的变化也会影响输出值。
无论权重如何，死区内的过程值变化都不会影响输出值。



禁用制冷或使用制冷系数时的死区（左），或激活制冷并采用 PID 参数切换时的死区（右）。x/水平轴表示控制偏差 = 设定值 - 过程值。y/垂直轴表示传送到 PID 算法的死区输出信号。

控制区宽度

如果过程值不处于设定值附近的控制区，控制器将输出最小输出值或最大输出值。这意味着，过程值会更快地到达设定值。

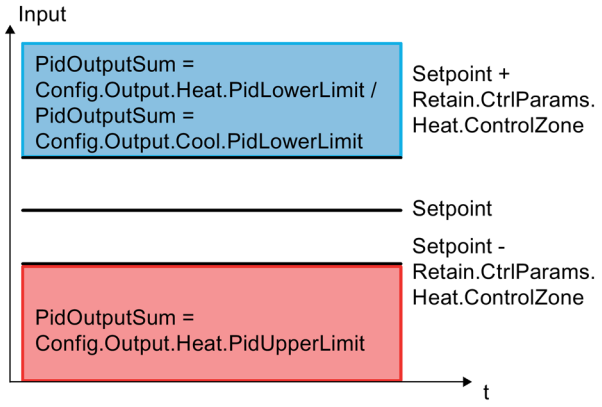
如果过程值位于设定值附近的控制区内，则输出值通过 PID 算法进行计算。

只有将“PID（温度）”选作制冷或加热过程的控制器结构时，才会在预调节过程中自动设置加热或制冷的控制区宽度。

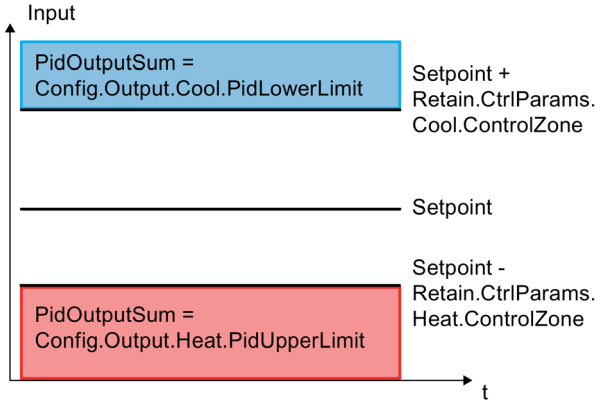
如果将控制区宽度设置为 3.402822e+38，会禁用控制区。

如果已在基本设置中禁用了制冷，或使用了制冷系数，则控制区对称地位于“设定值 - 控制区宽度（加热）”和“设定值 + 控制区宽度（加热）”之间。

如果已在基本设置中激活了制冷，并且在输出设置中将 PID 参数切换选作加热/制冷方法，则控制区位于“设定值 - 控制区宽度（加热）”和“设定值 + 控制区宽度（制冷）”之间。



禁用制冷或使用制冷系数时的控制区。



激活制冷并采用 PID 参数切换时的控制区。

调节的规则

从“控制器结构”(Controller structure) 下拉列表中选择要计算 PI 还是 PID 参数。可分别指定适用于加热和制冷的调节规则。

- PID（温度）

  在预调节和精确调节期间计算 PID 参数。

  预调节专门用于温度控制过程，可生成更慢、更为渐近的控制响应，与“PID”选项相比过调很少。精确调节与“PID”选项相同。

  只有选择此选项后，预调节期间才会自动确定控制区宽度。

- PID

  在预调节和精确调节期间计算 PID 参数。

- PI

  在预调节和精确调节期间计算 PI 参数。

- 用户自定义

  如果通过用户程序或参数视图为预调节和精确调节组态了不同的控制器结构，则下拉列表会显示“用户自定义”(User-defined)。

6.3 调试 PID_Temp

6.3.1 调试

调试窗口有助于您调试 PID 控制器。 可以在趋势视图中监视加热和制冷的设定值、过程值以及输出值随时间轴的变化。调试窗口支持以下功能：

- 控制器预调节
- 控制器精确调节

  使用精确调节对 PID 参数进行精确调节。
- 在趋势视图中监视当前闭环控制
- 通过指定手动 PID 输出值和替代设定值来测试受控系统
- 将 PID 参数的实际值保存到离线项目。

所有功能都需要与 CPU 建立在线连接。

如果尚未与 CPU 建立在线连接，应建立此连接，然后通过趋势视图的“全部监视”(Monitor all) 或“启动”(Start) 按钮使调试窗口运行。

趋势视图的操作

- 从“采样时间”(Sampling time) 下拉列表中，选择所需的采样时间。

  趋势视图的所有值按所选的采样时间进行更新。
- 如果要使用趋势视图，请单击测量组中的“启动”(Start) 图标。

  将启动值记录操作。 加热和制冷的设定值、过程值以及输出值的当前值将输入到趋势视图中。
- 如果要结束趋势视图，请单击“停止”(Stop) 图标。

  可以继续对趋势视图中记录的值进行分析。

关闭调试窗口将终止趋势视图中的记录操作并删除所记录的值。

6.3.2 预调节

预调节功能可确定对输出值跳变的过程响应，并搜索拐点。根据受控系统的最大斜率与死时间计算已调节的 PID 参数。可在执行预调节和精确调节时获得最佳 PID 参数。

过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。处于“未激活”或“手动模式”工作模式时就很可能出现这种情况。重新计算前会备份 PID 参数。

PID_Temp 可根据组态提供不同的预调节类型：

- 预调节加热

  加热输出值输出跳变，计算加热过程的 PID 参数，然后将设定值用作自动模式的控制变量。

- 预调节加热和制冷

  加热输出值输出跳变。

  只要过程值接近设定值，制冷输出值便输出跳变。

  计算加热（Retain.CtrlParams.Heat 结构）和制冷（Retain.CtrlParams.Cool 结构）过程的 PID 参数，然后将设定值用作自动模式的控制变量。

- 预调节制冷

  制冷输出值输出跳变。

  计算制冷的 PID 参数，然后将设定值用作自动模式的控制变量。

如果要调节加热和制冷过程的 PID 参数，先后使用“预调节加热”(Pretuning heating) 和“预调节制冷”(Pretuning cooling) 与单独使用“预调节加热和制冷”(Pretuning heating and cooling) 相比，可获得更好的控制响应。但是，分两个步骤进行预调节耗费的时间较长。

常规要求

- 已在循环中断 OB 中调用 PID_Temp 指令。
- ManualEnable = FALSE
- Reset = FALSE
- PID_Temp 处于下列模式之一：“未激活”、“手动模式”或“自动模式”。
- 设定值和过程值均在组态的限值范围内（请参见过程值监视 (页 186)组态）。

预调节加热的相关要求

- 设定值与过程值的差值大于过程值上限与过程值下限之差的 30%。
- 设定值与过程值的差值大于设定值的 50%。
- 设定值大于过程值。

预调节加热和制冷的相关要求

- 在“基本设置”中已激活制冷输出 (Config.ActivateCooling = TRUE)。
- 在“输出值的基本设置”中已激活 PID 参数切换 (Config.AdvancedCooling = TRUE)。
- 设定值与过程值的差值大于过程值上限与过程值下限之差的 30%。
- 设定值与过程值的差值大于设定值的 50%。
- 设定值大于过程值。

预调节制冷的相关要求

- 在“基本设置”中已激活制冷输出 (Config.ActivateCooling = TRUE)。
- 在“输出值的基本设置”中已激活 PID 参数切换 (Config.AdvancedCooling = TRUE)。
- 已成功执行“预调节加热”或“预调节加热和制冷”(PIDSelfTune.SUT.ProcParHeatOk = TRUE)。对于所有调节，应使用同一设定值。
- 设定值与过程值的差值小于过程值上限与过程值下限之差的 5%。

步骤

要执行预调节，请按下列步骤操作：

1. 在项目树中双击“PID_Temp > 调试”(PID_Temp > Commissioning) 条目。
2. 激活“全部监视”(Monitor all) 按钮或启动趋势视图。

   将建立在线连接。

3. 从“调节模式”(Tuning mode) 下拉列表中选择所需的预调节条目。
4. 单击“Start”图标。
   - 将启动预调节功能。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。进度条指示当前步骤的进度。

说明

如果进度条（“进度”变量）长时间无变化，猜测可能是调节功能受到限制时，请单击“Stop”图标。检查工艺对象的组态，必要时请重新启动控制器调节功能。

结果

如果执行预调节时未产生错误消息，则 PID 参数已调节完毕。PID_Temp 将切换到自动模式并使用已调节的参数。在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果无法实现预调节，PID_Temp 将根据已组态的响应对错误作出反应。

6.3.3 精确调节

精确调节将使过程值出现恒定受限的振荡。将根据此振荡的幅度和频率为工作点调节 PID 参数。PID 参数将根据结果重新计算。精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。可在执行预调节和精确调节时获得最佳 PID 参数。

PID_Temp 将自动尝试生成大于过程值噪声的振荡。过程值的稳定性对精确调节的影响非常小。重新计算前会备份 PID 参数。

PID_Temp 可根据组态提供不同的精确调节类型：

- 精确调节加热：

  PID_Temp 使过程值出现振荡，加热输出值发生周期性变化，并计算加热过程的 PID 参数。

- 精确调节制冷：

  PID_Temp 使过程值出现振荡，制冷输出值发生周期性变化，并计算制冷的 PID 参数。

加热/制冷控制器的临时调节偏移量

如果将 PID_Temp 用作加热/制冷控制器 (Config.ActivateCooling = TRUE)，则相应设定值对应的 PID 输出值 (PidOutputSum) 必须符合以下要求，这样才能使过程值出现振荡从而成功进行精确调节：

- 精确调节加热的 PID 输出值为正
- 精确调节制冷的 PID 输出值为负

如果不满足上述条件，则可以为精确调节指定一个临时偏移量，以在具有相反效果的输出上输出。

- 精确调节加热过程时的制冷输出偏移量 (PIDSelfTune.TIR.OutputOffsetCool)。

  启动调节前，输入负的制冷调节偏移量，该偏移量小于静止状态下相应设定值对应的 PID 输出值 (PidOutputSum)。

- 精确调节制冷时的加热输出偏移量 (PIDSelfTune.TIR.OutputOffsetHeat)

  启动调节前，输入正的加热调节偏移量，该偏移量大于静止状态下相应设定值对应的 PID 输出值 (PidOutputSum)。

随后，由 PID 算法抵消指定的偏移量，从而使过程值保持为设定值。偏移高度允许对 PID 输出值进行相应调整从而使其满足上述要求。

为避免在定义偏移量后过程值过调较大，还可以分多步增大偏移量。

如果 PID_Temp 退出精确调节模式，将重置调节偏移量。

示例：指定精确调节制冷的偏移量

- 不指定偏移量
  - Setpoint = 过程值 (ScaledInput) = 80 °C
  - PID 输出值 (PidOutputSum) = 30.0
  - 加热输出值 (OutputHeat) = 30.0
  - 制冷输出值 (OutputCool) = 0.0

    只有制冷输出无法使过程值围绕设定值振荡。此时无法执行精确调节。
- 加热输出的偏移量 (PIDSelfTune.TIR.OutputOffsetHeat) = 80.0
  - Setpoint = 过程值 (ScaledInput) = 80 °C
  - PID 输出值 (PidOutputSum) = -50.0
  - 加热输出值 (OutputHeat) = 80.0
  - 制冷输出值 (OutputCool) = -50.0

    由于指定了加热输出的偏移量，加热输出现在可以使设定值附近的过程值出现振荡。现在可以成功执行精确调节。

常规要求

- 已在循环中断 OB 中调用 PID_Temp 指令。
- ManualEnable = FALSE
- Reset = FALSE
- 设定值和过程值均处于组态的限值范围内（请参见“过程值设置”组态）。
- 控制回路已稳定在工作点。过程值与设定值一致时，表明到达了工作点。

  启用死区时，结果可能是永久控制偏差（设定值与实际值之间的偏差）。这可能对精确调节产生负面影响。
- 不能被干扰。
- PID_Temp 处于未激活模式、自动模式或手动模式。

精确调节加热的相关要求

- Heat.EnableTuning = TRUE
- Cool.EnableTuning = FALSE
- 如果将 PID_Temp 组态为加热和制冷控制器 (Config.ActivateCooling = TRUE)，则在达到要开始调节的工作点时必须激活加热输出。

  PidOutputSum > 0.0（请参见调节偏移量）

精确调节制冷的相关要求

- Heat.EnableTuning = FALSE
- Cool.EnableTuning = TRUE
- 已激活制冷输出 (Config.ActivateCooling = TRUE)。
- 已激活 PID 参数切换 (Config.AdvancedCooling = TRUE)。
- 在达到要开始调节的工作点时必须激活制冷输出。

  PidOutputSum < 0.0（请参见调节偏移量）

过程取决于初始情况

可在以下工作模式下启动精确调节：“未激活”、“自动模式”或“手动模式”。

在以下模式下启动精确调节时，具体情况如下所述：

- 自动模式，且 PIDSelfTune.TIR.RunIn = FALSE（默认）

  如果希望通过调节来改进现有 PID 参数，请在自动模式下启动精确调节。

  PID_Temp 将使用现有的 PID 参数控制系统，直到控制回路已稳定并且精确调节的要求得到满足为止。之后才会启动精确调节。

- 未激活，手动模式或自动模式，且 PIDSelfTune.TIR.RunIn = TRUE

  系统尝试利用最小或最大输出值达到设定值（两点控制）：

  – 在精确调节加热时，使用最小或最大加热输出值。

  – 在精确调节制冷时，使用最小或最大制冷输出值。

  这可能会增加超调量。精确调节将在达到设定值时启动。

  如果无法达到设定值，PID_Temp 不会自动中止调节过程。

步骤

要执行精确调节，请按下列步骤操作：

1. 在项目树中双击“PID_Temp > 调试”(PID_Temp > Commissioning) 条目。
2. 激活“全部监视”(Monitor all) 按钮或启动趋势视图。

   将建立在线连接。
3. 从“调节模式”(Tuning mode) 下拉列表中选择所需的精确调节条目。
4. 如有需要（请参见调节偏移量），可指定调节偏移量，然后等到再次达到静止状态。
5. 单击“Start”图标。
   - 将启动精确调节过程。
   - “状态”(Status) 字段显示当前步骤和所发生的所有错误。

     进度条指示当前步骤的进度。

说明

如果进度条（“进度”变量）长时间无变化，猜测可能是调节功能受到限制时，请单击“调节模式”(Tuning mode) 组中的“Stop”图标。检查工艺对象的组态，必要时请重新启动控制器调节功能。

尤其是在以下阶段，如果无法达到设定值，将不会自动中止调节过程。

- “尝试使用两点控制达到加热过程的设定值。”
- “尝试使用两点控制达到制冷过程的设定值。”

结果

如果执行精确调节时未出错，则 PID 参数已调节完毕。PID_Temp 将切换到自动模式并使用已调节的参数。在电源关闭以及重启 CPU 期间，已调节的 PID 参数保持不变。

如果精确调节期间出现错误，PID_Temp 将根据已组态的响应对错误作出反应。

6.3.4 “手动”模式

下面说明如何在工艺对象“PID_Temp”的调试窗口中使用“手动模式”。

错误未决时也可使用手动模式。

要求

- 已在循环中断 OB 中调用“PID_Temp”指令。
- 已与 CPU 建立在线连接。
- CPU 处于“RUN”模式。

步骤

如果要通过指定手动值来测试受控系统，请使用调试窗口中的“手动模式”。

要定义手动值，请按以下步骤操作：

1. 在项目树中双击“PID_Temp > 调试”(PID_Temp > Commissioning) 条目。
2. 激活“全部监视”(Monitor all) 按钮或启动趋势视图。

   将建立在线连接。
3. 在“控制器的在线状态”(Online status of the controller) 区域中，选中复选框“手动模式”(Manual mode)。

   PID_Temp 将在手动模式下运行。 最新的当前输出值仍然有效。
4. 在可编辑字段中，输入 % 形式的手动值。

   如果已在基本设置中激活制冷过程，请按下列方式输入手动值：

   – 输入正的手动值以输出加热输出的值。
   – 输入负的手动值以输出制冷输出的值。
5. 单击 图标。

结果

手动值被写入 CPU 并立即生效。

如果希望由 PID 控制器重新指定输出值，请清除“手动模式”(Manual mode) 复选框。

到自动模式的切换是无扰动的。

6.3.5 替代设定值

下面说明如何在工艺对象“PID_Temp”的调试窗口中使用替代设定值。

要求

- 已在循环中断 OB 中调用“PID_Temp”指令。
- 已与 CPU 建立在线连接。
- CPU 处于“RUN”模式。

步骤

如果要用作设定值的值不同于在“Setpoint”参数中指定的值（如调节级联结构中的从控制器），请在调试窗口中使用替代设定值。

要指定替代设定值，请按以下步骤操作：

1. 在项目树中双击“PID_Temp > 调试”(PID_Temp > Commissioning) 条目。
2. 激活“全部监视”(Monitor all) 按钮或启动趋势视图。

   将建立在线连接。
3. 在“控制器的在线状态”(Online status of the controller) 部分，选中复选框“Subst.Setpoint”。

   使用最近更新的设定值初始化替代设定值（SubstituteSetpoint 变量），并且立即使用替代设定值。
4. 在可编辑字段中输入替代设定值。
5. 单击 图标。

结果

替代设定值被写入 CPU 并立即生效。

如果希望将“Setpoint”参数的值重新用作设定值，则清除“Subst.Setpoint”复选框。

切换是无扰动的。

6.3.6 级联调试

有关使用 PID_Temp 级联调试的信息，请参见调试 (页 214)。

6.4 使用 PID_Temp 的级联控制

6.4.1 简介

在级联控制中，多个控制回路相互嵌套。在此过程中，从控制器会从相应的较高级的主控制器的输出值 (OutputHeat) 接收其设定值 (Setpoint)。

建立级联控制系统的先决条件是，受控系统可分为具有自身测量变量的各个子系统。

受控变量的设定值在最外层的主控制器指定。

最内层从控制器的输出值应用于执行器，因此作用于受控系统。

与单回路控制系统相比，使用级联控制系统的主要优势如下：

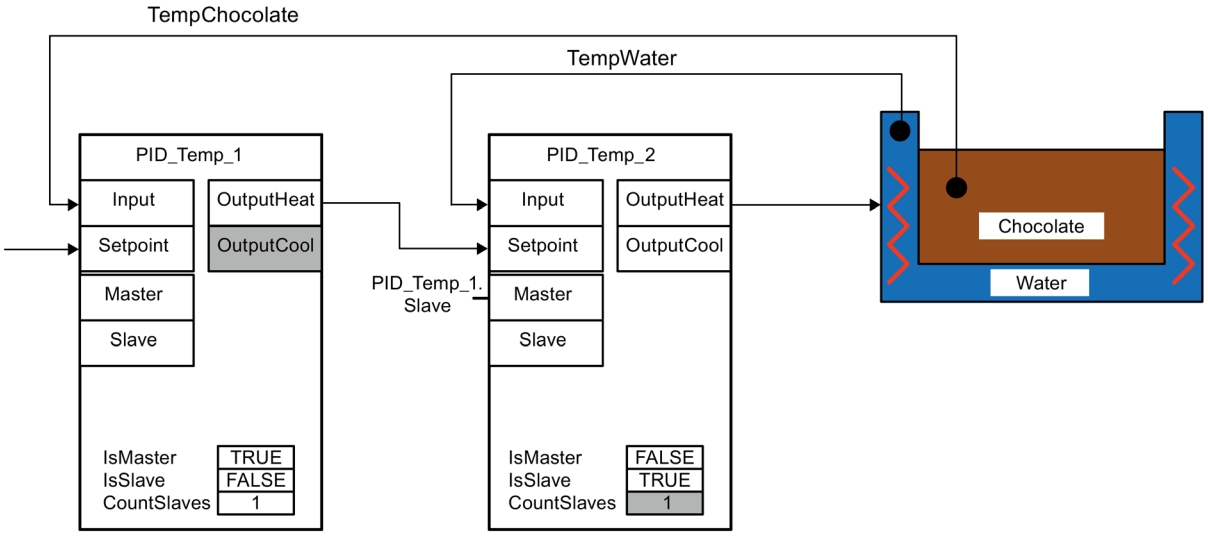
- 由于额外存在从属控制回路，可迅速纠正控制系统中发生的扰动。这会显著降低扰动对控制变量的影响。因此，可改善扰动行为。
- 从属控制回路以线性形式发挥作用。因此，这些非线性扰动对受控变量的负面影响可得到缓解。

PID_Temp 具有以下专用于级联控制系统的功能：

- 指定替代设定值
- 在主从控制器间交换状态信息（如当前操作模式）
- 不同的 Anti-Wind-Up 模式（主控制器对其从控制器限值的响应）

示例

以下框图以巧克力融化装置为例，显示使用 PID_Temp 的级联控制系统：



PID_Temp_1 主控制器将巧克力温度 (TempChocolate) 的过程值与用户在 Setpoint 参数中指定的设定值进行比较。其输出值 OutputHeat 构成从控制器 PID_Temp_2 的设定值。

PID_Temp_2 尝试将水浴温度 (TempWater) 的过程值调节到此设定值。PID_Temp_2 的输出值直接作用于受控系统（水浴加热）的执行器，因此可影响水浴温度。而水浴温度又会影响巧克力温度。

常见问题解答

有关详细信息，请参见西门子工业在线支持中的以下常见问题解答。

- 条目 ID 103526819 (https://support.industry.siemens.com/cs/ww/en/view/103526819)

参见

创建程序 (页 210)

6.4.2 创建程序

在创建程序的过程中应注意下列几点：

- PID_Temp 实例数量

  循环中断 OB 中调用的不同 PID_Temp 实例的数量必须与该过程中级联连接的测量变量的数量一致。

  在此例中共有两个级联连接的测量变量：TempChocolate 和 TempWater。因此需要两个 PID_Temp 实例。

- 调用顺序

  在同一循环中断 OB 中，必须先调用主控制器，再调用从控制器。

  首先调用指定用户设定值的最外层主控制器。

  随后调用设定值由最外层主控制器指定的从控制器，依此类推。

  通过输出值作用于该过程执行器的最内层从控制器最后调用。

  在此例中，先调用 PID_Temp_1 再调用 PID_Temp_2。

- 测量变量的互连

  最外层的主控制器与要被调节为用户设定值的最外层测量变量互连。

  最内层从控制器与受执行器直接影响的最内层测量变量互连。

  通过参数 Input 或 Input_PER 实现测量变量与 PID_Temp 的互连。

  在此例中，最外层的测量变量 TempChocolate 与 PID_Temp_1 互连，最内层的测量变量 TempWater 与 PID_Temp_2 互连。

- 主控制器输出值与从控制器设定值的互连

  必须将主控制器的输出值 (OutputHeat) 分配给从控制器的设定值 (Setpoint)。

  此互连可在编程编辑器中执行，或在从控制器巡视窗口的基本设置中通过选择主控制器来自动执行。

  如有需要，可插入您自己的滤波器或标定功能，例如，这样可以使主控制器的输出值范围根据从控制器的设定值/过程值范围进行调整。

  在此例中，将 PID_Temp_1 的 OutputHeat 分配给 PID_Temp_2 的 Setpoint。

- 用于在主从控制器间交换信息的接口的互连

  必须将主控制器的“Slave”参数分配给其所有直接从属从控制器（这些控制器从此主控制器接收设定值）的“Master”参数。可通过从控制器接口执行分配，从而将一个主控制器与多个从控制器互连，并在从控制器巡视窗口的基本设置中显示互连。

  此互连可在编程编辑器中执行，或在从控制器巡视窗口的基本设置中通过选择主控制器来自动执行。

  只有执行此互连后，Anti-Wind-Up 功能以及主控制器对从控制器工作模式的评估才能正常运行。

  在本例中，将 PID_Temp_1 的“Slave”参数分配给 PID_Temp_2 的“Master”参数。

使用 SCL 实现的程序代码示例（未将从控制器的输出值分配给执行器）：

```
"PID_Temp_1"(Input:="TempChocolate");

"PID_Temp_2"(Input:="TempWater", Master := "PID_Temp_1".Slave, Setpoint :=
"PID_Temp_1".OutputHeat);
```

参见

PID_Temp ActivateRecoverMode 变量 (页 500)

6.4.3 组态

您可以通过用户程序、组态编辑器或 PID_Temp 调用的巡视窗口执行组态。

在级联控制系统中使用 PID_Temp 时，应确保对以下指定的设置进行正确组态。

如果 PID_Temp 实例从上级主控制器接收设定值，并转而将其输出值输出到从属从控制器，则此 PID_Temp 实例既为主控制器又为从控制器。对于此类 PID_Temp 实例，必须执行下文列出的两种组态。例如，具有三个级联连接测量变量和三个 PID_Temp 实例的级联控制系统中，中间的 PID_Temp 实例便属于此种情况。

主控制器的组态

组态编辑器或巡视窗口中的设置	DB 参数	说明
基本设置 → 级联： 激活“控制器为主控制器”(Controller is master) 复选框	Config.Cascade.IsMaster = TRUE	将此控制器激活为级联中的主控制器
基本设置 → 级联： 从控制器的数量	Config.Cascade.CountSlaves	直接从此主控制器接收设定值的直接从属从控制器的数量
基本设置 → 输入/输出参数： 选择输出值（加热） = OutputHeat	Config.Output.Heat.Select = 0	主控制器仅使用输出参数 OutputHeat。 将禁用 OutputHeat_PWM 和 OutputHeat_PER。
基本设置 → 输入/输出参数： 清除“激活制冷”(Activate cooling) 复选框	Config.ActivateCooling = FALSE	必须在主控制器中禁用制冷。

组态编辑器或巡视窗口中的设置	DB 参数	说明
输出设置 → 输出限值和标定 → OutputHeat / OutputCool： PID 输出值下限（加热）， PID 输出值上限（加热）， 标定的输出下限值（加热） 标定的输出上限值（加热）	Config.Output.Heat.PidLowerLimit, Config.Output.Heat.PidUpperLimit, Config.Output.Heat.LowerScaling, Config.Output.Heat.UpperScaling	将主控制器的 OutputHeat 分配给从控制器的 Setpoint 时，如果未使用用户自己的标定功能，则可能需要根据从控制器的设定值/过程值范围调整主控制器的输出值限值和输出标定。
在巡视窗口或组态编辑器的功能视图中不存在该变量。 您可以通过组态编辑器的参数视图对其进行更改。	Config.Cascade.AntiWindUpMode	Anti-Wind-Up 模式确定当直接从属从控制器到达输出值限值时，如何处理此主控制器的积分作用。 选项有： • AntiWindUpMode = 0： 禁用 AntiWindUp 功能。主控制器不会对其从控制器的限值做出响应。 • AntiWindUpMode = 1（默认）： 主控制器的积分作用在关系“达到限值的从控制器/从控制器数量”中会减弱。这将减弱限值对控制行为的影响。 • AntiWindUpMode = 2： 从控制器达到限值后，主控制器的积分作用将立即暂停。

从控制器的组态

组态编辑器或巡视窗口中的设置	DB 参数	说明
基本设置 → 级联： 选中“控制器为从控制器”(Controller is slave) 复选框	Config.Cascade.IsSlave = TRUE	将此控制器激活为级联中的主控制器

6.4.4 调试

编译和加载程序后，可启动级联控制系统的调试过程。

在调试过程中（执行调节或使用现有 PID 参数更改为自动模式），从最内层的从控制器开始，然后逐步向外调试，直到达到最外层的主控制器。

在上述示例中，首先调试 PID_Temp_2，然后继续调试 PID_Temp_1。

调节从控制器

调节 PID_Temp 时要求设定值恒定。因此，激活从控制器的替代设定值（SubstituteSetpoint 和 SubstituteSetpointOn 变量）以调节从控制器，或通过相应的手动值将相关主控制器设置为手动模式。这样可以确保从控制器的设定值在调节过程中保持恒定。

调节主控制器

为使主控制器对该过程产生影响或执行调节，必须将所有下游从控制器置于自动模式，且必须禁用这些从控制器的替代设定值。主控制器会通过用于在主从控制器间（Master 参数和 Slave 参数）进行信息交换的接口对这些条件进行评估，并在 AllSlaveAutomaticState 和 NoSlaveSubstituteSetpoint 变量中显示当前状态。相应的状态消息会在调试编辑器中输出。

主控制器调试编辑器中的状态消息	主控制器的 DB 参数	纠正措施
一个或多个从控制器未处于自动模式。	AllSlaveAutomaticState = FALSE, NoSlaveSubstituteSetpoint = TRUE	首先，对所有下游从控制器执行调试。 执行调节或激活主控制器的手动模式或自动模式之前，确保已满足下列条件： • 所有下游从控制器都处于自动模式（状态 = 3）。 • 所有下游从控制器都已禁用替代设定值 (SubstituteSetpointOn = FALSE)。
一个或多个从控制器已激活替代设定值。	AllSlaveAutomaticState = TRUE, NoSlaveSubstituteSetpoint = FALSE	
一个或多个从控制器未处于自动模式，且已激活替代设定值。	AllSlaveAutomaticState = FALSE, NoSlaveSubstituteSetpoint = FALSE	

如果已启动主控制器的预调节或精确调节，PID_Temp 在以下情况会中止调节并通过 ErrorBits = DW#16#0200000 显示错误：

- 一个或多个从控制器未处于自动模式 (AllSlaveAutomaticState = FALSE)
- 一个或多个从控制器已激活替代设定值 (NoSlaveSubstituteSetpoint = FALSE)。

后续的工作模式切换取决于 ActivateRecoverMode。

6.4.5 替代设定值

为了指定设定值，除 Setpoint 参数外，PID_Temp 会通过 SubstituteSetpoint 变量提供替代设定值。此替代设定值可通过设置 SubstituteSetpointOn = TRUE 或在调试编辑器中选中相应的复选框来激活。

通过替代设定值，可在调试或调节等过程中直接在从控制器暂时指定设定值。

这种情况下，不必在程序中对主控制器输出值与从控制器设定值的互连（级联控制系统正常运行所必需的）进行更改。

为使主控制器对该过程产生影响或执行调节，必须禁用所有下游从控制器的替代设定值。

可以对当前有效的设定值进行监视，因为该设定值以 CurrentSetpoint 变量的形式被 PID 算法使用参与计算。

6.4.6 工作模式和故障响应

PID_Temp 实例的主控制器或从控制器不会更改此 PID_Temp 实例的工作模式。

如果其中一个从控制器发生故障，主控制器仍然保持当前工作模式。

如果主控制器发生故障，从控制器仍然保持当前工作模式。 但是，由于将主控制器的输出值用作从控制器的设定值，之后从控制器的进一步操作将取决于主控制器的故障和组态的故障响应：

- 如果对主控制器组态了 ActivateRecoverMode = TRUE，且故障不会阻止 OutputHeat 的计算过程，则故障不会对从控制器产生任何影响。
- 如果对主控制器组态了 ActivateRecoverMode = TRUE，且故障会阻止 OutputHeat 的计算过程，则主控制器会输出上一次的输出值或已组态的替代输出值 SubstituteOutput，具体取决于 SetSubstituteOutput。 然后，从控制器会将其用作设定值。

  由于已对 PID_Temp 进行预组态，在此情况下会输出替代输出值 0.0（ActivateRecoverMode = TRUE、SetSubstituteOutput = TRUE、SubstituteOutput = 0.0）。 为应用组态合适的替代输出值，或启用上一个有效 PID 输出值 (SetSubstituteOutput = FALSE)。
- 如果对主控制器组态了 ActivateRecoverMode = FALSE，则当发生故障或输出 OutputHeat = 0.0 时，主控制器会切换到“未激活”模式。 然后，从控制器会使用 0.0 作为设定值。

故障响应位于组态编辑器的输出设置中。

6.5 使用 PID_Temp 的多区域控制

简介

在多区域控制系统中，可同时控制工厂的多个部分（即所谓的多个区域），使其达到不同的温度。多区域控制系统的特点为各个温度区域会由于热耦合而相互影响，例如，某个区域的过程值会因热耦合而影响其它区域的过程值。这种影响的作用强度取决于工厂的结构和这些区域所选的工作点。

示例：例如，塑料加工行业的挤压厂。

必须对通过挤压机的混合物进行控制，使其达到不同的温度，以实现最优处理。例如，可能要求挤压机填料口的温度不同于排料口。各个温度区域会由于热耦合而相互影响。

在多区域控制系统中使用 PID_Temp 时，每个温度区域都由单独的 PID_Temp 实例进行控制。

在多区域控制系统中使用 PID_Temp 时，请遵照下列说明。

分别进行加热和制冷预调节

通常，对工厂进行初始调试时首先会执行预调节，以便对 PID 参数进行初始设置并对工作点进行控制。对多区域控制系统进行预调节时通常可对所有区域同时执行预调节。

对于已激活制冷过程且将 PID 参数切换作为加热/制冷方法（Config.ActivateCooling = TRUE，Config.AdvancedCooling = TRUE）的控制器，PID_Temp 可在一个步骤中实现加热和制冷的预调节（Mode = 1，Heat.EnableTuning = TRUE，Cool.EnableTuning = TRUE）。

但是，建议不要使用这种调节对多区域控制系统中的多个 PID_Temp 实例同时进行预调节。而应首先分别执行加热预调节（Mode = 1，Heat.EnableTuning = TRUE，Cool.EnableTuning = FALSE）和制冷预调节（Mode = 1，Heat.EnableTuning = FALSE，Cool.EnableTuning = TRUE）。

只有当所有区域都完成加热预调节且达到工作点时，才能启动制冷预调节。

这会降低调节过程中各区域间由于热耦合而产生的相互影响。

调整延迟时间

如果应用 PID_Temp 的多区域控制系统的各区域间存在较强的热耦合，应确保通过 PIDSelfTune.SUT.AdaptDelayTime = 0 禁止调整预调节延迟时间。否则，如果在调整延迟时间期间（此阶段加热被禁用），某个区域的制冷因其它区域的热效应而无法进行，则确定延迟时间时可能出错。

暂时禁用制冷

对于已激活制冷 (Config.ActivateCooling = TRUE) 的控制器，通过设置 DisableCooling = TRUE，PID_Temp 可在自动模式下暂时禁用制冷。

这可以确保当其它区域的控制器尚未完成加热调节时，此控制器在调试过程中不会以自动模式制冷。否则，调节可能会因各区域间的热耦合而受到负面影响。

步骤

对存在热耦合的多区域控制系统进行调试时，可按以下步骤进行操作：

1. 对于所有已激活制冷的控制器，设置 DisableCooling = TRUE。
2. 对于所有控制器，设置 PIDSelfTune.SUT.AdaptDelayTime = 0。
3. 指定所需设定值（Setpoint 参数）并对所有控制器同时启动加热预调节（Mode = 1，Heat.EnableTuning = TRUE，Cool.EnableTuning = FALSE）。
4. 耐心等待，直到所有控制器均完成加热预调节。
5. 对于所有已激活制冷的控制器，设置 DisableCooling = FALSE。
6. 耐心等待，直到所有区域的过程值均达到稳定状态，且接近相应的设定值。

   如果对于某个区域，经过很长时间都无法达到设定值，则说明加热或制冷执行器的作用太弱。
7. 对于所有已激活制冷的控制器，启动制冷预调节（Mode = 1，Heat.EnableTuning = FALSE，Cool.EnableTuning = TRUE）。

说明

过程值超出限值

如果在自动模式下通过 DisableCooling = TRUE 禁用了制冷，则可能导致当 DisableCooling = TRUE 时，过程值超出设定值或过程值限值。使用 DisableCooling 时请注意观察过程值，在适用的情况下可以进行干预。

说明

多区域控制系统

对于多区域控制系统，各区域间的热耦合在调试或运行期间可能导致过调次数增加、暂时或长时间超出限值或出现暂时或长时间的控制偏差。请注意观察过程值并准备好进行干预。根据系统不同，操作步骤可能会与上述步骤有所不同。

同步多个精确调节过程

如果在自动模式下启动精确调节且 PIDSelfTune.TIR.RunIn = FALSE，则 PID_Temp 会尝试通过 PID 控制和当前 PID 参数达到设定值。达到设定值后，才会启动实际调节过程。对于多区域控制系统，各个区域达到设定值所需的时间可能各不相同。

如果要对多个区域同时执行精确调节，PID_Temp 可以在达到设定值后，等待进一步的调节步骤，从而同步这些过程。

步骤

这可以确保当实际调节步骤启动时，所有控制器都已达到设定值。这会降低调节过程中各区域间由于热耦合而产生的相互影响。

对于相应区域要同时执行精确调节的各控制器，请执行以下步骤：

1. 对于所有控制器，设置 PIDSelfTune.TIR.WaitForControlIn = TRUE。

   这些控制器必须处于自动模式，且 PIDSelfTune.TIR.RunIn = FALSE。

2. 指定所需设定值（Setpoint 参数）并对所有控制器启动精确调节。
3. 耐心等待，直到所有控制器的 PIDSelfTune.TIR.ControlInReady = TRUE。
4. 对于所有控制器，设置 PIDSelfTune.TIR.FinishControlIn = TRUE。

然后，所有控制器会同时启动实际调节过程。

6.6 使用 PID_Temp 进行超驰控制

超驰控制

超驰控制时，两个或多个控制器共享一个执行器。只有一个控制器可以随时访问执行器并影响过程。

由逻辑运算决定可以访问执行器的控制器。通常根据所有控制器的输出值比较结果做出此决定（例如，进行最大选择时），具有最大输出值的控制器将获得对执行器的访问权限。

基于输出值的选择要求所有控制器均在自动模式下工作。对不影响执行器的控制器进行更新。为防止饱和效应及其对控制响应和控制器之间的切换产生负面影响，这很有必要。

自版本 V1.1 起，PID_Temp 通过提供一个用于更新未激活控制器的简单过程，支持超驰控制：通过使用变量 OverwriteInitialOutputValue 和 PIDCtrl.PIDInit，可以预分配自动模式下控制器的积分作用，好像在上一周期中 PID 算法已为 PID 输出值计算 PidOutputSum = OverwriteInititalOutputValue。为此，OverwriteInitialOutputValue 与当前可以访问执行器的控制器的输出值互连。通过设置位 PIDCtrl.PIDInit，触发积分作用的预分配以及控制器循环和 PWM 周期的重启。根据预分配的（并针对所有控制器同步的）积分作用，以及当前控制偏差的比例作用与积分作用，在当前循环中进行输出值的后续计算。通过 PIDCtrl.PIDInit = TRUE 调用期间，微分作用未激活，因此对输出值不起作用。

此过程可以确保仅根据当前的过程状态和 PI 参数对当前输出值进行计算，并从而决定可以访问执行器的控制器。可防止未激活控制器的饱和效应，并因此防止切换逻辑的错误决定。

要求

- 只有在激活了积分作用时（变量 Retain.CtrlParams.Heat.Ti 和 Retain.CtrlParams.Cool.Ti > 0.0），PIDCtrl.PIDInit 才有效。
- 您必须在用户程序中自行分配 PIDCtrl.PIDInit 和 OverwriteInitialOutputValue（请参见下面的示例）。PID_Temp 不会自动更改这些变量。
- 仅当 PID_Temp 处于自动模式（参数 State = 3）时，PIDCtrl.PIDInit 才有效。
- 如果可能，请选择 PID 算法的采样时间（ Retain.CtrlParams.Heat.Cycle 和 Retain.CtrlParams.Cool.Cycle 变量）以使所有控制器的采样时间均相同，并在同一个循环中断 OB 中调用所有控制器。这样，可以确保在一个控制器循环或 PWM 周期内不发生切换。

说明

不断调整输出值限制

也可以通过在其它控制器系统中不断调整输出值限制实现这一操作，而不是如此处所述对没有执行器访问权的控制器进行主动更新。

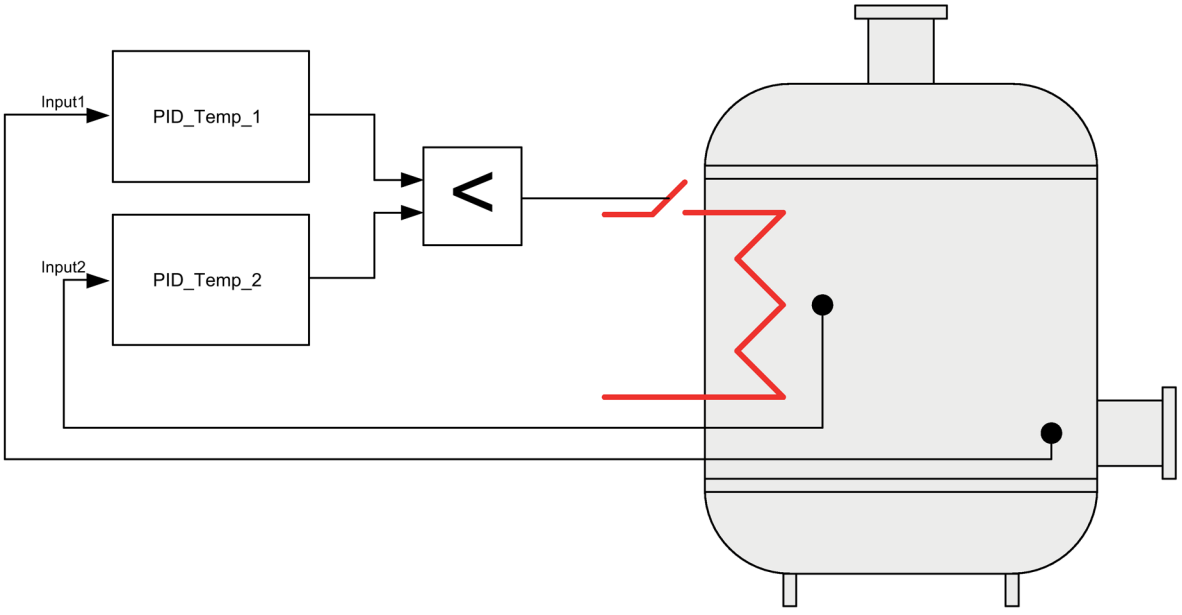
无法使用 PID_Temp 实现这一操作，因为在自动模式下不支持更改输出值限制。

示例：大型锅炉的控制

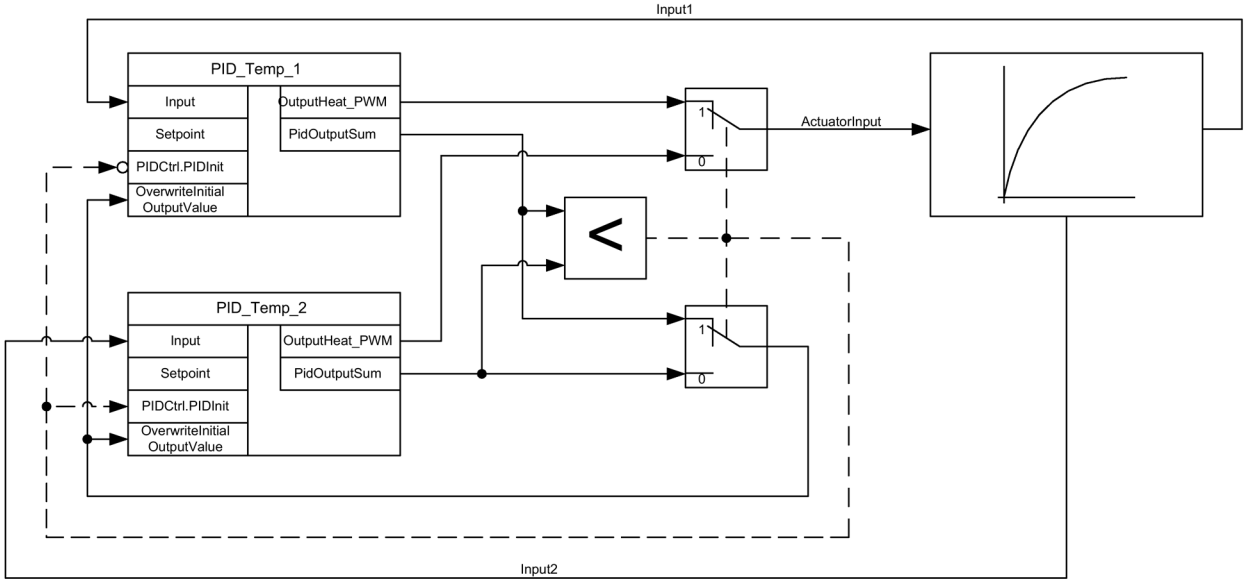
PID_Temp 用于控制大型锅炉。

主要目标是控制温度 Input1。为此使用控制器 PID_Temp_1。此外，通过限制控制器 PID_Temp_2 使温度 Input2 保持在附加测量点的上限值以下。

这两个温度仅受一个加热器的影响。控制器的输出值对应于加热功率。



通过编写程序变量 ActuatorInput 并借助 PID_Temp 的脉宽调制输出值（参数 OutputHeat_PWM）对该加热器进行控制。在参数 PID_Temp_1.Setpoint 处指定温度 Input1 的设定值。在参数 PID_Temp_2.Setpoint 处将附加测量点的温度上限值指定为设定值。



两个控制器必须共享一个加热器作为共享的执行器。在这种情况下，通过 PID 输出值（采用实数格式，参数 PidOutputSum）的最小选择实现逻辑，该逻辑决定哪个控制器获得执行器的访问权。由于 PID 输出值对应于加热功率，因此需要较低加热功率的控制器将获得控制权。

设备正常运行时，主受控变量的过程值对应于设定值。主控制器 PID_Temp_1 已稳定在固定的 PID 输出值 PID_Temp_1.PidOutputSum。正常操作过程中，限制控制器 Input2 的过程值显著低于指定为 für PID_Temp_2 设定值的上限。因此，限制控制器要增大加热功率以增大其过程值，即，它将计算一个大于主控制器 PID_Temp_1.PidOutputSum 的 PID 输出值 PID_Temp_2.PidOutputSum。切换逻辑的最小选择从而使得主控制器 PID_Temp_1 可以继续访问执行器。此外，确保通过赋值 PID_Temp_2.OverwriteInitialOutputValue = PID_Temp_1.PidOutputSum 以及 PID_Temp_2.PIDCtrl.PIDInit = TRUE 来更新 PID_Temp_2。

如果 Input2 现已接近于上限或超出上限（例如，因故障而导致），则限制控制器 PID_Temp_2 会计算一个较小的 PID 输出值，以限制加热功率并因此而减小 Input2。如果 PID_Temp_2.PidOutputSum 小于 PID_Temp_1.PidOutputSum，则限制控制器 PID_Temp_2 将通过最小选择获得执行器访问权，并减小加热功率。确保通过赋值 PID_Temp_1.OverwriteInitialOutputValue = PID_Temp_2.PidOutputSum 以及 PID_Temp_1.PIDCtrl.PIDInit = TRUE 来更新 PID_Temp_1。

附加测量点 Input2 处的温度下降。主受控变量 Input1 的温度也会下降，并且无法再保持在设定值。

解决故障后，Input2 会继续下降，并通过限制控制器进一步增大加热功率。只要主控制器计算出了一个较小的加热功率作为输出值，设备就会恢复正常操作，以使主控制器 PID_Temp_1 再次获得对执行器的访问权限。可以通过以下 SCL 程序代码实现此示例：

```
"PID_Temp_1"(Input := "Input1");

"PID_Temp_2"(Input := "Input2");

IF "PID_Temp_1".PidOutputSum <= "PID_Temp_2".PidOutputSum THEN

  "ActuatorInput" := "PID_Temp_1".OutputHeat_PWM;

  "PID_Temp_1".PIDCtrl.PIDInit := FALSE;

  "PID_Temp_2".PIDCtrl.PIDInit := TRUE;

  "PID_Temp_2".OverwriteInitialOutputValue := "PID_Temp_1".PidOutputSum;

ELSE

  "ActuatorInput" := "PID_Temp_2".OutputHeat_PWM;

  "PID_Temp_1".PIDCtrl.PIDInit := TRUE;

  "PID_Temp_2".PIDCtrl.PIDInit := FALSE;

  "PID_Temp_1".OverwriteInitialOutputValue := "PID_Temp_2".PidOutputSum;

END_IF;
```

6.7 使用 PLCSIM 仿真 PID_Temp

说明

使用 PLCSIM 进行仿真

不支持通过 PLCSIM 针对 CPU S7-1200 仿真 PID_Temp。

只能通过 PLCSIM 针对 CPU S7-1500 仿真 PID_TEMP。

对于使用 PLCSIM 进行的仿真，仿真 PLC 的时间特性与“真实”PLC 并不完全相同。仿真 PLC 循环中断 OB 的实际周期时钟波动比“真实”PLC 的波动大。

在标准组态中，PID_Temp 会自动确定调用之间的时间，并监视波动情况。

因此，使用 PLCSIM 仿真 PID_Temp 时，可能检测到采样时间错误 (ErrorBits = DW#16#00000800)。

这会导致进行中的调节中止。

自动模式下的响应取决于 ActivateRecoverMode 变量的值。

为防止此类情况发生，应按下列方式为使用 PLCSIM 进行的仿真组态 PID_Temp：

- CycleTime.EnEstimation = FALSE
- CycleTime.EnMonitoring = FALSE
- CycleTime.Value：以秒为单位为此变量分配调用循环中断 OB 的周期时钟。

使用 PID 的基本功能 7

7.1 CONT_C

7.1.1 工艺对象 CONT_C

工艺对象 CONT_C 提供一个自动和手动模式的连续 PID 控制器。 它与指令 CONT_C 的背景数据块相对应。 可以使用 PULSEGEN 指令来组态脉冲控制器。

比例、积分 (INT) 和微分分量 (DIF) 彼此之间并行切换，可以单独打开和关闭。 使用它，可以设置 P、I、PI、PD 和 PID 控制器。

S7-1500
工艺对象的所有参数和变量均具有保持性，在完整下载 CONT_C 的前提下，只能在下载到设备期间更改这些数据。

参见

软件控制器概述 (页 41)

添加工艺对象 (页 44)

组态工艺对象 (页 48)

CONT_C (页 509)

将工艺对象下载到设备 (页 76)

7.1.2 组态控制器误差 CONT_C

使用外设过程值

要在输入参数 PV_PER 中使用外设格式的过程值，请按以下步骤操作：

1. 选中“启用 I/O”(Enable I/O) 复选框。
2. 如果过程值是以实际大小提供的，请以百分比形式输入标定的因子和偏移量。
   系统随后会根据以下公式来确定过程值：
   PV = PV_PER × PV_FAC + PV_OFF

使用内部过程值

要在输入参数 PV_IN 中使用浮点格式的过程值，请按以下步骤操作：

1. 清除“启用 I/O”(Enable I/O) 复选框。

控制偏差

根据以下要求设置死区范围：

- 过程值信号有噪声。
- 控制器增益很高。
- 微分作用激活。

这种情况下，过程值的噪声分量会导致输出值出现巨大偏差。 死区可抑制控制器处于稳态的噪声分量。 死区范围指定死区的大小。 死区范围为 0.0 时，死区关闭。

参见

CONT_C 的工作原理 (页 510)

7.1.3 组态控制器算法 CONT_C

常规步骤

要确定激活控制算法的哪些分量，请执行以下操作：

1. 从“控制器结构”(Controller structure) 列表中选择一个条目。
   只能为所选控制器结构指定所需参数。

比例作用

1. 如果该控制器结构包含比例作用，请输入“比例增益”。

积分作用

1. 如果该控制器结构包含积分作用，请输入积分作用时间。
2. 要给积分作用赋予初始化值，请选中“初始化积分作用”(Initialize integral action) 复选框并输入初始化值。
3. 要将积分作用永久设置为此初始化值，请选中“积分作用保持”(Integral action hold) 复选框。

微分作用

1. 如果该控制器结构包含微分作用，请输入微分作用时间、微分作用权重和延迟时间。

参见

CONT_C 的工作原理 (页 510)

7.1.4 组态输出值 CONT_C

常规步骤

可以在手动或自动模式下设置 CONT_C。

1. 要设置手动调节值，请激活选项“激活手动模式”(Activate manual mode) 选项复选框。
   您可以在输入参数 MAN 中指定手动调节值。

调节值限制

调节值具有上限和下限，因此只能接受有效值。 您无法关闭限值。 超出限值时会通过输出参数 QLMN_HLM 和 QLMN_LLM 显示。

1. 输入调节值的上限和下限值。
   如果调节值是实际大小，则调节值上下限的单位必须一致。

标定

调节值可根据以下公式，通过因子和偏移量标定为作为浮点值和外设值输出。

标定调节值 = 调节值 x 因子 + 偏移量

默认值是因子等于 1.0，偏移量等于 0.0。

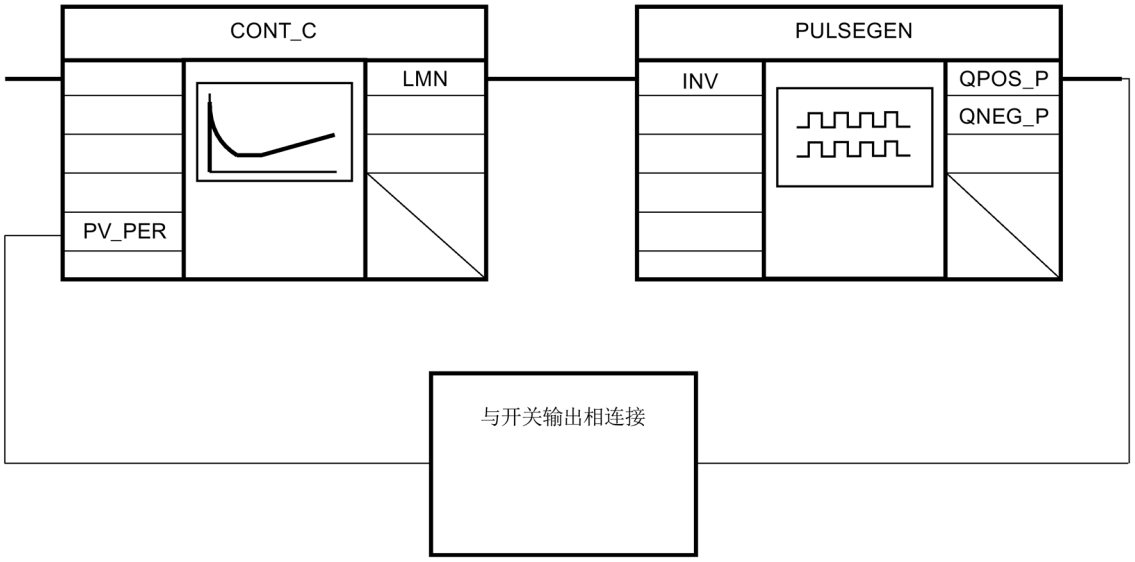
1. 输入因子和偏移量的值。

参见

CONT_C 的工作原理 (页 510)

7.1.5 对脉冲控制器进行编程

利用连续控制器 CONT_C 和脉冲整形器 PULSEGEN，可以实现一个设定值固定的控制器，使其具有比例执行器的开关输出。 下图显示了控制回路的信号流。



连续控制器 CONT_C 构成输出值 LMN，脉冲整形器 PULSEGEN 将该输出值转换为脉冲/中断信号 QPOS_P 或 QNEG_P。

参见

PULSEGEN (页 521)

7.1.6 调试 CONT_C

要求

- 已在 CPU 中装载指令和工艺对象。

步骤

要手动确定最佳 PID 参数，请按以下步骤操作：

1. 单击“Start”图标。

   如果不存在在线连接，则将建立在线连接。 系统会记录设定值、过程值和输出值的当前值。

2. 在“P”、“I”、“D”和“延迟时间”(Delay time) 字段中输入新的 PID 参数。
3. 在“调节”(Tuning) 组中单击图标 “将参数发送到 CPU”(Send parameter to CPU)。
4. 在“当前值”(Current values) 组中选中“更改设定值”(Change setpoint) 复选框。
5. 输入新设定值并在“当前值”(Current values) 组中单击图标 。
6. 清除“手动模式”(Manual mode) 复选框。

   此时控制器使用新 PID 参数工作并控制新设定值。

7. 检查 PID 参数的质量以检查曲线点。
8. 重复步骤 2 到 6，直至对控制器结果满意为止。

7.2 CONT_S

7.2.1 工艺对象 CONT_S

工艺对象 CONT_S 提供了一个用于控制具有积分行为的执行器的步进控制器，并且可用于通过二进制输出值输出信号控制工艺温度过程。 该工艺对象对应于 CONT_S 指令的背景数据块。 其工作原理基于采样控制器的 PI 控制算法。步进控制器在没有位置反馈信号的情况下运行。手动和自动模式均可。

S7-1500
工艺对象的所有参数和变量均具有保持性，在完整下载 CONT_S 的前提下，只能在下载到设备期间更改这些数据。

参见

软件控制器概述 (页 41)

添加工艺对象 (页 44)

组态工艺对象 (页 48)

CONT_S (页 516)

将工艺对象下载到设备 (页 76)

7.2.2 组态控制器误差 CONT_S

使用外设过程值

要在输入参数 PV_PER 中使用外设格式的过程值，请按以下步骤操作：

1. 选中“启用 I/O”(Enable I/O) 复选框。
2. 如果过程值以物理量形式提供，请以百分比形式输入标定的因子和偏移量。
   系统随后会根据以下公式来确定过程值：
   PV = PV_PER × PV_FAC + PV_OFF

使用内部过程值

要在输入参数 PV_IN 中使用浮点格式的过程值，请按以下步骤操作：

1. 清除“启用 I/O”(Enable I/O) 复选框。

控制偏差

根据以下要求设置死区范围：

- 过程值信号含有噪声。
- 控制器增益很高。
- 微分作用激活。

这种情况下，过程值的噪声分量会导致调节变量出现巨大偏差。 死区可抑制控制器处于稳态的噪声分量。 死区范围指定死区的大小。 死区范围为 0.0 时，死区关闭。

参见

CONT_S 工作模式 (页 517)

7.2.3 组态控制算法 CONT_S

PID 算法

1. 为 P 分量输入“比例放大倍数”。
2. 为 I 分量的时间行为输入积分时间。
   积分时间为 0.0 时，I 分量关闭。

参见

CONT_S 工作模式 (页 517)

7.2.4 组态调节值 CONT_S

常规步骤

可以在手动或自动模式下设置 CONT_S。

1. 要设置手动调节值，请激活“激活手动模式”(Activate manual mode) 选项复选框。
   为输入参数 LMNUP 和 LMNDN 输入手动调节值。

脉冲发生器

1. 输入最短脉冲持续时间和最短暂停持续时间。
   值必须大于等于输入参数 CYCLE 的周期时间。 因此，操作频率会降低。
2. 输入电机设定时间。
   值必须大于等于输入参数 CYCLE 的周期时间。

参见

CONT_S 工作模式 (页 517)

7.2.5 调试 CONT_S

要求

- 已将指令和工艺对象加载到 CPU。

步骤

要手动确定最优 PID 参数，请按以下步骤操作：

1. 单击“Start”图标。

   如果不存在在线连接，则将建立在线连接。 系统会记录设定值、过程值和输出值的当前值。

2. 在字段“P”和“I”中，输入新比例值和新积分值。
3. 在“调节”(Tuning) 组中单击图标 “将参数发送到 CPU”(Send parameter to CPU)。
4. 在“当前值”(Current values) 组中选中“更改设定值”(Change setpoint) 复选框。
5. 输入新设定值并在“当前值”(Current values) 组中单击图标 。
6. 清除“手动模式”(Manual mode) 复选框。

   这时控制器使用新参数工作并控制新设定值。

7. 检查 PID 参数的质量以检查曲线点。
8. 重复步骤 2 到 6，直至对控制器结果满意为止。

7.3 TCONT_CP

7.3.1 工艺对象 TCONT_CP

工艺对象 TCONT_CP 提供一个具有脉冲发生器的连续温度控制器。 它与指令 TCONT_CP 的背景数据块对应。 此操作基于采样控制器的 PID 控制算法。手动和自动模式均可。

指令 TCONT_CP 在预调节期间计算受控系统的比例、积分和微分参数。 “精确调节”可用于进一步调节这些参数。 用户还可以手动输入 PID 参数。

S7-1500
工艺对象的所有参数和变量均具有保持性，在完整下载 TCONT_CP 的前提下，只能在下载到设备期间更改这些数据。

参见

软件控制器概述 (页 41)

添加工艺对象 (页 44)

组态工艺对象 (页 48)

TCONT_CP (页 532)

将工艺对象下载到设备 (页 76)

7.3.2 组态 TCONT_CP

7.3.2.1 控制器误差

使用外设过程值

要使用输入参数 PV_PER，请执行以下步骤：

1. 从“源”(Source) 列表中选择条目“外设”(Periphery)。
2. 选择“传感器类型”。
   对于不同类型的传感器，过程值会根据不同的公式进行标定。
   - 标准
     热电偶；PT100/NI100

     PV = 0.1 × PV_PER × PV_FAC + PV_OFFS
   - 冷却；
     PT100/NI100

     PV = 0.01 × PV_PER × PV_FAC + PV_OFFS
   - 电流/电压

     PV = 100/27648 × PV_PER × PV_FAC + PV_OFFS
3. 输入用于标定外设过程值的因子和偏移量。

使用内部过程值

要使用输入参数 PV_IN，请执行以下步骤：

1. 从“源”(Source) 列表中选择条目“内部”(Internal)。

控制偏差

根据以下要求设置死区范围：

- 过程值信号含有噪声。
- 控制器增益很高。
- 微分作用激活。

这种情况下，过程值的噪声分量会导致调节变量出现巨大偏差。 死区可抑制控制器处于稳态的噪声分量。 死区范围指定死区的大小。 死区范围为 0.0 时，死区关闭。

参见

TCONT_CP 的工作模式 (页 533)

7.3.2.2 控制算法

常规步骤

1. 输入“PID 算法采样时间”。
   控制器采样时间不应超过确定的控制器积分作用时间 (TI) 的 10 %。
2. 如果该控制器结构包含比例作用，请输入“比例增益”。
   如果比例增益为负，则规则含义为相反的含义。

比例作用

如果设定值发生变化，可能会导致比例作用超调。 通过比例作用的权重，可选择设定值发生变化时比例作用的响应程度。 通过补偿积分作用可弱化比例作用。

1. 要弱化应对设定值变化的比例作用，可相应地输入“比例作用权重”。
   - 1.0: 应对设定值变化的比例作用完全有效
   - 0.0: 应对设定值变化的比例作用无效

积分作用

达到调节值的限制值时，积分作用停止。 如果控制偏差将积分作用朝内部设定范围的方向移动，则积分作用将再次释放。

1. 如果该控制器结构包含积分作用，请输入“积分作用时间”。
   积分作用时间为 0.0 时，积分作用关闭。
2. 要给积分作用赋予初始化值，请选中复选框“初始化积分作用”(Initialize integral action) 并输入“初始化值”。
   重新启动后或 COM_RST = TRUE 时，积分作用将设置为此值。

微分作用

1. 如果该控制器结构包含微分作用，请输入微分作用时间 (TD) 和系数 DT1 (D_F))。
   对于启用的微分作用，应保持以下的等式关系：
   TD = 0.5 × CYCLE× D_F.
   延迟时间根据以下公式进行计算：
   延迟时间 = TD/D_F

通过工作点设置 PD 控制器

1. 输入积分作用时间 0.0。
2. 激活“初始化积分作用”(Initialize integral action) 复选框。
3. 输入工作点作为初始化值。

通过工作点设置 P 控制器

1. 通过工作点设置 PD 控制器。
2. 输入微分作用时间 0.0。
   微分作用被禁用。

控制区

控制区限制控制偏差的值范围。 如果控制偏差超出此值范围，则使用调节值限制值。

使用控制区时，微分作用会导致调节变量迅速减小。 因此，控制区仅对启用的微分作用有意义。 如果不使用控制区，只有减小比例作用才能减小调节值。 如果从新工作点所需的调节值中移除输出的最小或最大调节值，控制区会导致无超调/欠调的快速振荡。

1. 在“控制区”(control zone) 组中激活“激活”(Activate) 复选框。
2. 在“宽度”(Width) 输入字段中输入设定值，过程值可能高于或低于该设定值。

参见

TCONT_CP 的工作模式 (页 533)

7.3.2.3 调节值连续控制器

调节值限制

调节值具有上限和下限，因此只能接受有效值。 您无法关闭限值。 超出限值时会通过输出参数 QLMN_HLM 和 QLMN_LLM 显示。

1. 输入调节值的上限和下限值。

标定

调节值可根据以下公式，通过因子和偏移量标定为作为浮点值和外设值输出。

标定调节值 = 调节值 x 因子 + 偏移量

默认值是因子等于 1.0，偏移量等于 0.0。

1. 输入因子和偏移量的值。

脉冲发生器

可以为连续控制器打开脉冲发生器。

1. 在“脉冲发生器”(Pulse generator) 组中禁用“激活”(Activate) 选项复选框。

参见

TCONT_CP 的工作模式 (页 533)

7.3.2.4 调节值脉冲控制器

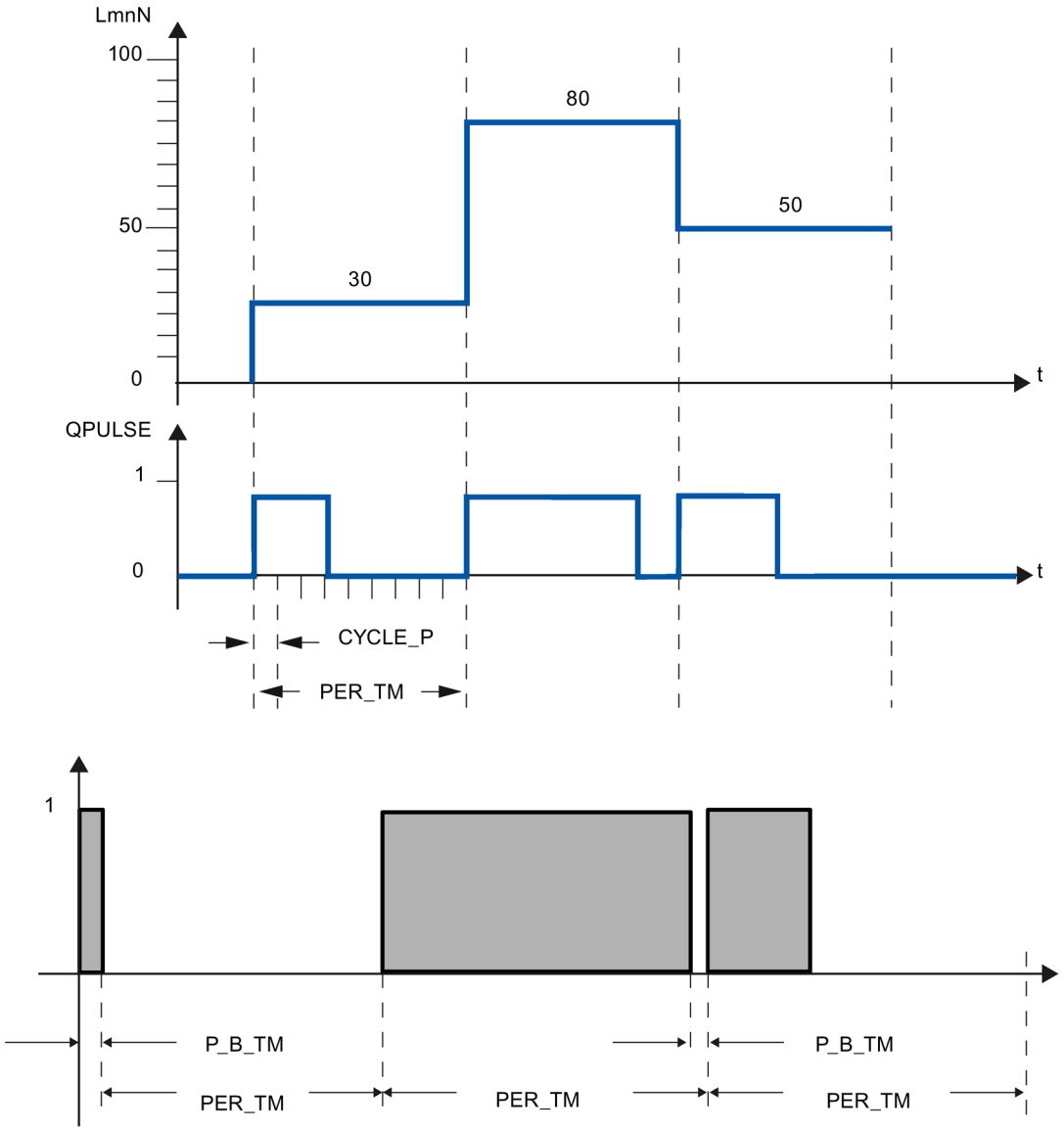
脉冲发生器

模拟调节值 (LmnN) 可通过作为脉冲序列的输出参数 QPULSE 上的脉冲持续时间调制输出。

要使用脉冲发生器，请执行以下步骤：

1. 在“脉冲发生器”(pulse generator) 组中激活“激活”(Activate) 选项复选框。
2. 输入“采样时间脉冲发生器”、“最短脉冲/中断持续时间”和“周期持续时间”。

下图阐明了“采样脉冲发生器”(CYCLE_P)、“最短脉冲/中断持续时间”(P_B_TM) 和“周期持续时间”(PER_TM) 之间的联系。



采样时间脉冲发生器

采样时间脉冲发生器必须适合所调用的循环中断 OB 的时间节拍。 所建立脉冲的持续时间始终是该值的整数倍。 要获得足够精确的调节值分辨率，应该应用以下关系：

CYCLE_P ≤ PER_TM/50

最短脉冲/中断持续时间

通过设定最短脉冲/中断持续时间，可避免执行器上的开或关时间过短。 小于 P_B_TM 的脉冲将被抑制。

建议值 P_B_TM ≤ 0.1 × PER_TM。

周期持续时间

周期持续时间不应超过确定的控制器积分时间 (TI) 的 20%：
PER_TM ≤ TI/5

参数 CYCLE_P、CYCLE 和 PER_TM 的作用示例：

周期持续时间 PER_TM = 10 s

采样时间 PID 算法 CYCLE = 1 s

采样时间脉冲发生器 CYCLE_P = 100 ms。

每秒钟出现一个新调节值，每 100 ms 将调节值与先前输出的脉冲长度和中断长度比较一次。

- 如果输出脉冲，则存在 2 种可能：
  - 计算的调节值大于先前的脉冲长度/PER_TM。 这时脉冲延长。
  - 计算的调节值小于等于先前的脉冲长度/PER_TM。 这时将不输出脉冲信号。
- 如果不输出脉冲，也存在 2 种可能：
  - 值（100 % - 计算的调节值）大于先前的中断长度/PER_TM。 这时中断延长。
  - 值（100 % - 计算的调节值）小于等于先前的中断长度/PER_TM。 这时将输出脉冲信号。

参见

TCONT_CP 的工作模式 (页 533)

脉冲发生器的工作原理 (页 543)

7.3.3 调试 TCONT_CP

7.3.3.1 TCONT_CP 优化

应用可能性

适用于过程类型 I 的加热或冷却过程的控制器优化。 但对于更高级别的过程，如过程类型 II 或 III，您可以使用块。

会自动确定并设置 PI/PID 参数。 该控制器的设计目的是达到最佳破坏行为。由此产生的“精确”参数导致设定值跳跃高度超调跳跃高度的 10% 到 40%。

控制器优化阶段

要进行控制器优化，会经历下列各个阶段，您可以在参数 PHASE 中读取这些阶段。

PHASE = 0

未执行任何调节。TCONT_CP 在自动模式或手动模式下工作。

PHASE = 0 期间，您可以确保受控系统满足优化要求。

优化结束时，TCONT_CP 重新更改为 PHASE = 0。

PHASE = 1

TCONT_CP 正准备优化。只有在满足优化要求时，才启动 PHASE = 1。

PHASE = 1 期间，会确定以下值：

- 过程值噪声 NOISE_PV
- 初始斜率 PVDT0
- 调节变量的平均值
- 采样时间 PID 算法 CYCLE
- 采样时间脉冲发生器 CYCLE_P

PHASE = 2

在阶段 2 中，过程值尝试通过常量调节变量检测拐点。 此方法将防止由于过程变量噪声而过早地找到拐点。

使用脉冲控制器时，通过 N 次脉冲循环均分过程变量，然后提供给控制器阶段。 过程变量在控制器阶段中会进一步均分： 最初，此均分未激活；换句话说，均分始终在经过 1 个循环后发生。 只要噪声超过某个特定级别，循环次数就会加倍。

将计算噪声的周期和振幅。 估计周期期间，仅当梯度总是小于最大上升时，才会取消搜索拐点并退出阶段 2。 而 TU 和 T_P_INF 在实际拐点处计算。

但仅在满足以下两个条件时调节才会结束：

1. 过程值与拐点相距超过 2*NOISE_PV。
2. 过程值已超过拐点 20%。

说明

使用设定值阶跃变化激发过程时，调节最迟在过程值超过设定值阶跃变化 (SP_INT-PV0) 的 75% 时结束（请参见下文）。

PHASE = 3, 4, 5

阶段 3、4 和 5 每个阶段持续 1 个周期。

在阶段 3 中，计算优化和过程参数之前会保存有效的 PI/PID 参数。

在阶段 4 中，会计算新 PI/PID 参数。

在阶段 5 中，计算新的调节变量并给出受控系统。

PHASE = 7

会在阶段 7 中检查过程类型，因为在优化完之后 TCONT_CP 会始终更改为自动模式。 当 LMN = LMN0 + 0.75*TUN_DLMN 作为调节变量时，自动模式启动。 过程类型的测试使用最近重新计算的控制器参数**在自动模式下**进行，并最晚在拐点之后的 0.35*TA（平衡时间）结束。 如果过程顺序严重偏离估计值，将重新计算控制器参数并使 STATUS_D 加 1；否则，控制器参数保持不变。

这时优化模式完成，TCONT_CP 返回到 PHASE = 0。 通过 STATUS_H 参数，可确认调节是否成功完成。

优化提前取消

在阶段 1、2 或 3 中，可通过重置 TUN_ON = FALSE 取消优化，无需计算新参数。 当 LMN = LMN0 + TUN_DLMN 时，控制器在自动模式下启动。 如果调节之前控制器处于手动模式，则将输出旧的手动调节变量。

如果通过设置 TUN_ON = FALSE，在阶段 4、5 或 7 取消调节，则在该阶段之前包含确定的受控参数。

7.3.3.2 优化要求

瞬态响应

该过程在发生时间延迟时必须具有稳定的渐近瞬态响应。

受控变量阶跃变化后，过程值必须保持为稳定状态。 因此，这样可排除已显示出没有控制的振荡响应的过程，以及没有进行恢复的过程（控制系统中的积分器）。

⚠警告

这可能导致人员死亡、严重受伤或造成重大财产损失。

调节期间，参数 MAN_ON 无效。 在此期间，输出值或过程值可能是非预期值，甚至是极值。

输出值通过调节来定义。 要取消调节，首先必须设置 TUN_ON = FALSE。 这会使 MAN_ON 再次有效。

保证稳定的初始状态（阶段 0）

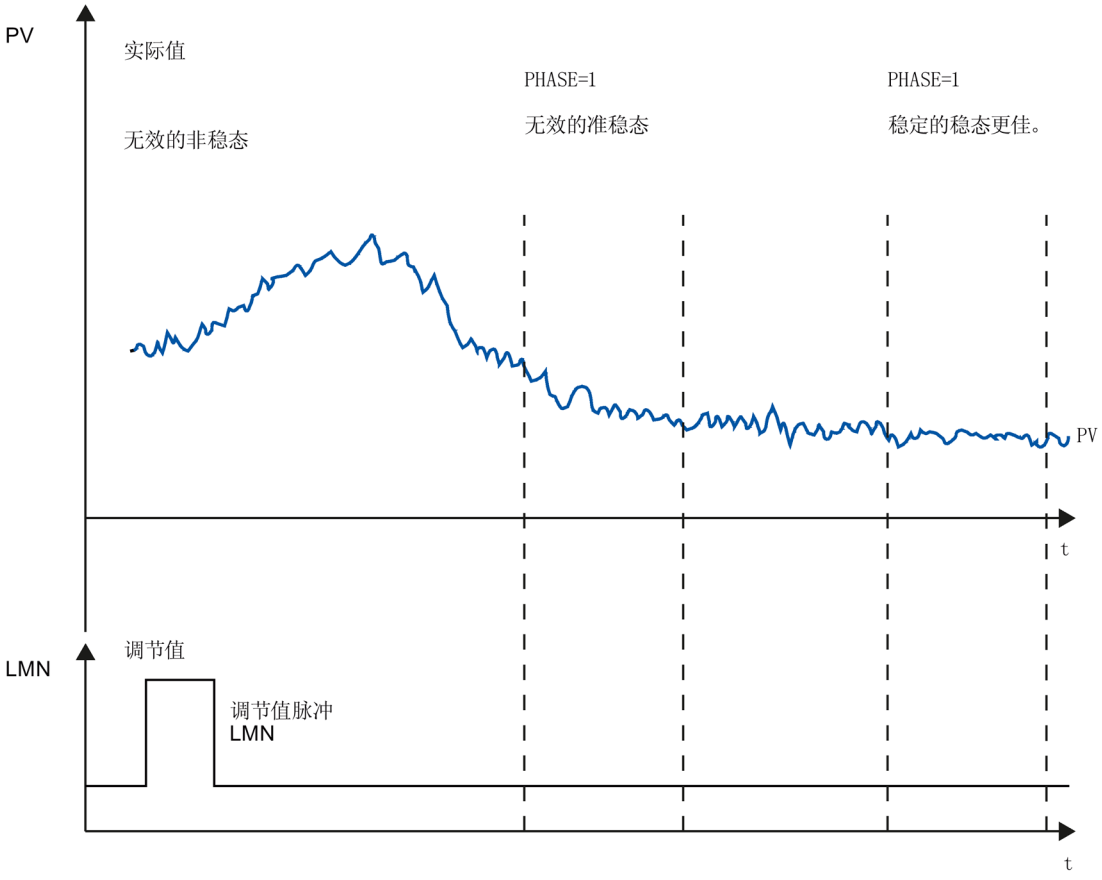
如果由于控制器参数不正确等原因导致过程值低频振荡，则在启动调节之前必须将控制器置于手动模式并等待振荡停止。 也可切换到“软”设置的 PI 控制器（小回路增益、长积分时间）。

现在，必须等到达到稳定状态，也就是等到过程值和输出值达到稳态。 还允许过程值有渐近瞬态振荡或慢速漂移（稳定状态，请参见下图）。 输出值必须为常量或上下波动一个恒定平均值。

说明

请避免在马上要启动调节之前更改调节变量。 建立测试条件（例如，关闭烤箱门）时，可能会在无意中更改调节变量！ 如果出现这种情况，则必须至少等到过程值再次具有处于稳定状态的渐近瞬态振荡。 如果等到瞬态效应完全消失，则可以得到更好的控制器参数。

下图解释了处于稳定状态的瞬态振荡：



线性和操作范围

该过程响应在整个操作范围中必须是线性的。 例如，聚集状态改变时，将发生非线性响应。 必须在操作范围的线性部分中进行调节。

也就是说，在调节和正常控制操作期间，在该操作范围内的非线性影响必须特别微小。但是，如果在新操作点附近重复进行调节，并且调节期间没有出现非线性影响，则可在操作点改变时重新调节该过程。

如果已知某个特定的静态非线性影响（例如，阀特性），始终建议使用折线对其进行补偿，从而线性化该过程响应。

温度过程中的干扰

诸如将热量传送到相邻区域的干扰必须不得过多影响整体温度过程。 例如，优化挤压机的区域时，必须同时加热所有区域。

7.3.3.3 优化可能性

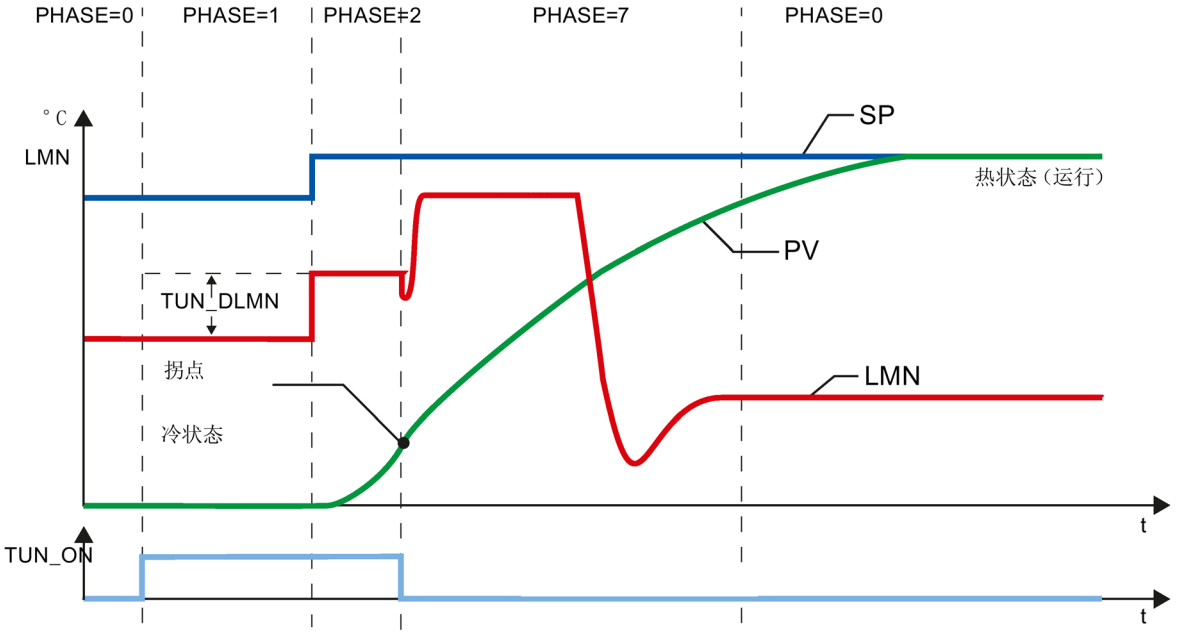
有以下几种调节方法：

- 预调节
- 精确调节
- 在控制模式下手动精确调节

预调节

在此调节过程中，通过设定值跳跃从冷态向工作点靠近。

TUN_ON = TRUE 时，可以建立调节准备状态。 控制器从 PHASE = 0 切换至 PHASE = 1。



通过设定值更改（跳转阶段 1 -> 2）激活调节受控变量 (LMN0 + TUN_DLMN)。 设定值在达到拐点之前不会生效（达到此点之前，无法启用自动模式）。

用户负责根据允许的过程值变化来定义输出激发增量 (TUN_DLMN)。 必须根据预期的过程值变化设置 TUN_DLMN 的符号（考虑控制操作时的方向）。

设定值阶跃变化和 TUN_DLMN 必须恰当地匹配。 如果 TUN_DLMN 的值过高，则存在设定值阶跃变化达到 75% 之前找不到拐点的风险。

尽管如此，TUN_DLMN 必须足够高，以确保过程值至少达到设定值阶跃变化的 22 %。否则，过程将保持为调节模式（阶段 2）。

解决方法： 在拐点搜索期间减小设定值。

说明

如果过程极慢，建议您在调节期间指定略微低于期望操作点的目标设定值，并密切监视状态位和 PV（超调风险）。

仅在线性范围内调节：

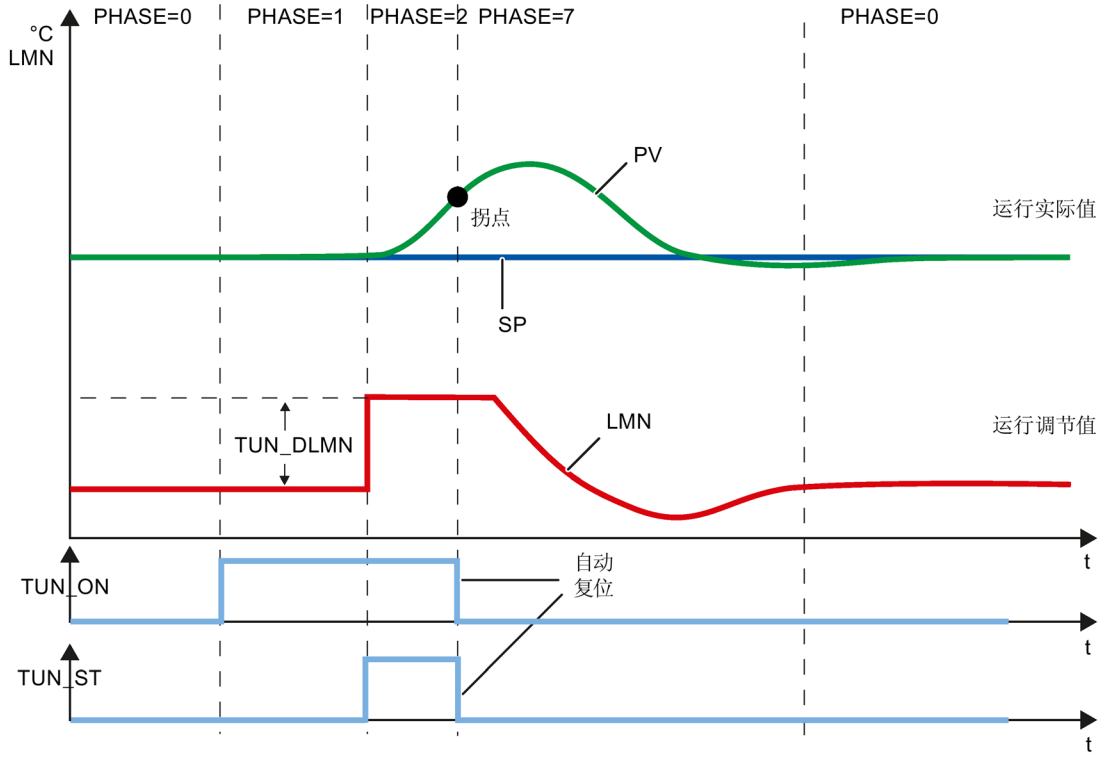
特定过程（例如，锌或镁冶炼炉）的信号将通过操作范围附近的非线性区域（聚集状态改变）。

通过选择适当的设定值阶跃变化，可将调节限制在线性范围之内。 当过程值超过设定值阶跃变化 (SP_INT-PV0) 的 75% 时，调节将结束。

同时，应将 TUN_DLMN 减小到可保证设定值阶跃变化达到 75% 之前能够发现拐点的范围。

精确调节

在此调节过程中，通过输出值跳跃激活设定值恒定不变的过程。



通过设置启动位 TUN_ST（从阶段 1 -> 2 的跳转）激活调节受控变量 (LMN0 + TUN_DLMN)。 修改设定值时，新值在达到拐点之前不会生效（达到此点之前无法启用自动模式）。

用户负责根据允许的过程值变化来定义输出激发增量 (TUN_DLMN)。 必须根据预期的过程值变化设置 TUN_DLMN 的符号（考虑控制操作时的方向）。

注意
通过 TUN_ST 激发过程时，安全性不会小于 75%。调节会在达到拐点时结束。 但是，在噪声过程中可能会显著超过拐点。

在控制模式下手动精确调节

可以采用以下措施以实现无超调的设定值响应：

- 调整控制区
- 优化命令操作
- 控制参数的衰减
- 修改控制参数

7.3.3.4 调谐结果

STATUS_H 的左侧数字显示调节状态

STATUS_H	结果
0	默认值，即（尚）未找到新的控制器参数。
10000	找到适合的控制参数。
2xxxx	已通过估计值找到控制参数；请检查控制响应或检查 STATUS_H 诊断消息并重复控制器调节。
3xxxx	发生一个操作员错误；请检查 STATUS_H 诊断消息并重复控制器调节。

CYCLE 和 CYCLE_P 采样时间已在阶段 1 中检查。

以下控制器参数在 TCONT_CP 中进行更新：

- P（比例 GAIN）
- I（积分时间 TI）
- D（微分时间 TD）
- 比例作用的权重 PFAC_SP
- 系数 DT1 (D_F)
- 控制区打开/关闭 CONZ_ON
- 控制区宽度 CON_ZONE

仅在过程类型适合（过程类型 I 和 II）并使用了 PID 控制器时才会激活控制区 (CONZ_ON = TRUE)。

根据 PID_ON，使用 PI 或 PID 控制器来执行控制。 旧的控制器参数已保存，并且可以使用 UNDO_PAR 恢复。另外还在 PI_CON 和 PID_CON 结构中保存了一个 PI 参数记录和一个 PID 参数记录。 也可以随后使用 LOAD_PID 并适当设置 PID_ON，在调节的 PI 或 PID 参数之间进行切换。

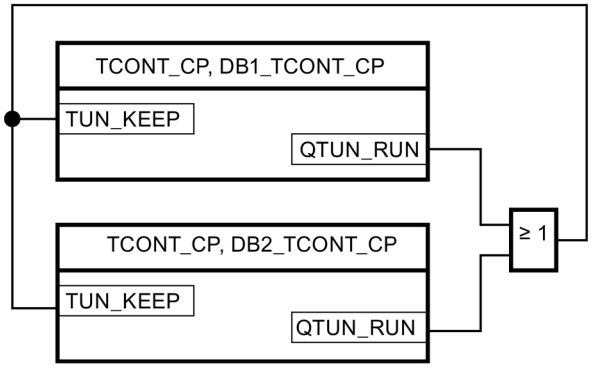
7.3.3.5 控制器通道的并行调谐

相邻区域（强热耦合）

如果两个或更多控制器正在费力地控制温度（换言之，存在采用强热耦合的两个加热器和两个测量过程值），请按以下步骤进行操作：

1. 以 OR 连接两个输出 QTUN_RUN。
2. 每个 TUN_KEEP 输入与 OR 元件的输出互连。
3. 通过同时指定设定值阶跃变化或同时设置 TUN_ST 来启动两个控制器。

以下示意图说明了控制器通道的并行调节。



优点：

两个控制器将输出 LMN0 + TUN_DLMN，直到它们同时离开阶段 2。这将防止首先完成调节的控制器由于受控变量中的更改而窜改另一个控制器的调节结果。

注意
设定值阶跃变化达到 75% 时会导致退出阶段 2 和重置输出 QTUN_RUN。 但是，自动模式直到 TUN_KEEP 也为 0 时才会启动。

相邻区域（弱热耦合）

一般来说，应执行调节以反映随后操作控制器的方式。 如果在生产期间各区域同时操作（以便保持区域之间的温度差），则调节期间相邻区域的温度应该相应地提高。

调节开始时温度中的差值不相关，因为它们会通过初始加热得到补偿（-> 初始上升 = 0）。

7.3.3.6 故障说明和更正措施

补偿操作员错误

操作员错误	STATUS 和操作	注释
同时设置了 TUN_ON 和设定值阶跃变化或 TUN_ST	跳转至阶段 1；但未启动调节。 • SP_INT = SP_{old} 或 • TUN_ST = FALSE	设定值更改被取消。 这将阻止控制器稳定到新设定值以及不必要地离开稳定操作点。
有效 TUN_DLMN < 5%（阶段 1 结束）	STATUS_H = 30002 • 跳转至阶段 0 • TUN_ON = FALSE • SP = SP_{old}	调节被取消。 设定值更改被取消。 这将阻止控制器稳定到新设定值以及不必要地离开稳定操作点。

未达到拐点（仅当通过设定值阶跃变化被激发时）

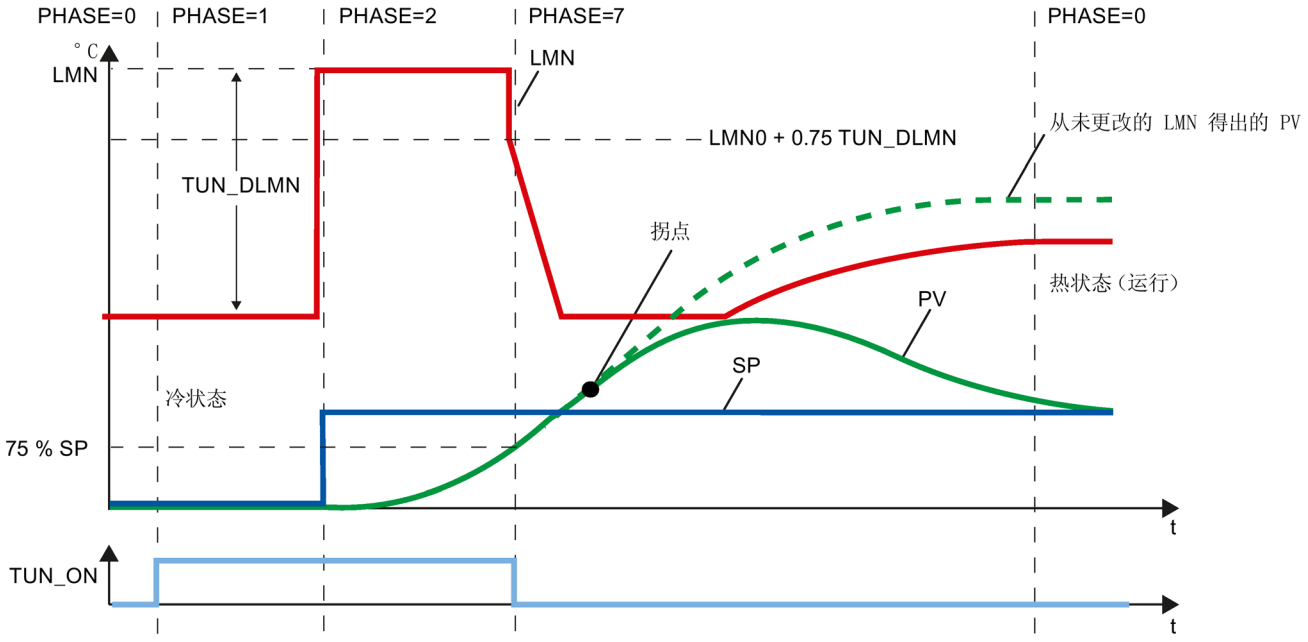
调节最迟在过程值超过设定值阶跃变化 (SP_INT-PV0) 的 75% 时结束。 将在 STATUS_H (2xx2x) 中发送“未达到拐点”信号。

始终应用当前有效的设定值。 通过减少设定值，可以尽快结束调节功能。

在典型温度过程中，在设定值阶跃变化的 75% 处取消调节通常足以防止超调。 但是会出现**警告**，尤其是在有较大延迟的过程中（TU/TA > 0.1，过程类型 III）。 如果受控变量激发与设定值阶跃变更相比过强，过程值可能严重超调（因子最高为 3）。

在更高阶的过程中，如果在达到设定值阶跃变化的 75% 之后仍远未达到拐点，则将会有显著的超调量。 此外，控制器参数过于严格。 在这种情况下，应减少控制器参数或重复尝试。

以下示意图说明了激发过强时过程变量的超调量（过程类型 III）：



在典型温度过程中，根据控制器参数在达到拐点之前快速取消无关紧要。

如果重复尝试，请减小 TUN_DLMN 或增加设定值阶跃变化。

原理： 用于调节的受控变量值必须适合设定值阶跃变化。

估计延迟时间或顺序时出错

未正确获取延迟时间（STATUS_H = 2x1xx 或 2x3xx）或顺序（STATUS_H = 21xxx 或 22xxx）。 将使用会产生非最佳控制器参数的估计值继续进行操作。

重复调节程序并确保过程值不受干扰。

说明

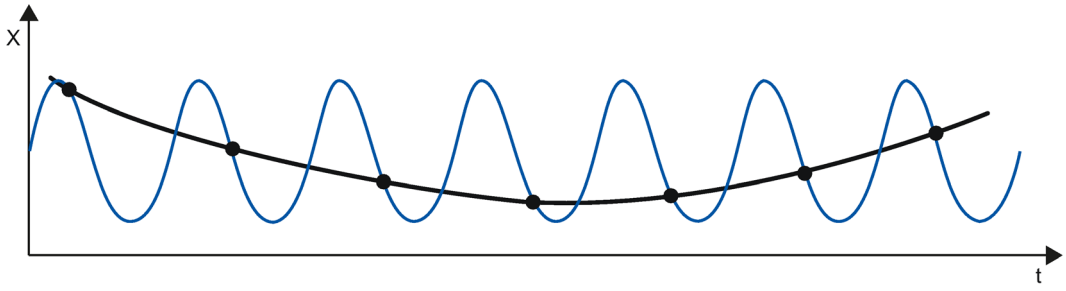
仅 PT1 过程的特殊情况也由 STATUS_H = 2x1xx (TU <= 3*CYCLE) 表示。 在这种情况下，不必再重复尝试。 如果控制发生振荡，则减少控制器参数。

测量信号的质量（测量噪声、低频干扰）

测量噪声或低频干扰可使调节结果失真。 请注意以下几点：

- 如果遇到测量噪声，请将采样频率设置得更高而不是更低。 在一个噪声周期期间，应至少采样两次过程值。 在脉冲模式下，积分平均值过滤会有帮助。 但是，这是假设过程变量 PV 在快速脉冲周期中传送给指令的情况。 噪声级别不应超过有用信号变化的 5%。
- 高频干扰不能通过 TCONT_CP 过滤掉。应尽早在测量传感器中过滤此干扰，以防止混淆效应。

  以下示意图说明了采样时间过长时的混淆效应：



- 对于低频干扰，确保足够高的采样率则相对简单。 但是，TCONT_CP 必须随后通过在平均值过滤中产生较大间隔来生成统一的测量信号。 平均值过滤必须至少扩展超过两个噪声周期。 在块内部，这会迅速导致更长的采样时间，因此会对调节的精度产生不利影响。 至少需要拐点的 40 个噪声周期才足以保证精度。

  重复尝试时可行的解决方法：

  增加 TUN_DLMN。

超调

以下情况可能发生超调：

情况	原因	解决方法
调节结束	• 与设定值阶跃变化相比过高的调节值变化引起激发（请参见上文）。 • 通过设置 PID_ON = FALSE 激活 PI 控制器。	• 增加设定值阶跃变化或减少调节值阶跃变化。 • 如果过程允许 PID 控制器，请通过 PID_ON = TRUE 启动调节。
阶段 7 中的调节	最初，会确定较平稳的控制器参数（过程类型 III）；这些参数会导致在阶段 7 中发生超调。	-
控制模式	适用于过程类型 I 的 PI 控制器 (FAC_SP = 1.0)。	如果过程允许 PID 控制器，请通过 PID_ON = TRUE 启动调节。

7.3.3.7 执行预调节

要求

- 已在 CPU 中装载指令和工艺对象。

步骤

要手动确定适用于初次调试的最优 PID 参数，请按以下步骤操作：

1. 单击“Start”图标。

   如果不存在在线连接，则将建立在线连接。 系统会记录设定值、过程值和输出值的当前值。

2. 从“模式”(Mode) 下拉列表中选择“预调节”(Pretuning)。

   TCONT_CP 准备好执行调节。

3. 在“输出值跳跃”(Output value jump) 字段中，指定输出值的增加量。

4. 在“设定值”(Setpoint) 字段中输入设定值。 输出值跳跃仅在输入另一设定值时才生效。

5. 单击 “启动调节”(Start tuning) 图标。

   预调节启动。 显示调节的状态。

7.3.3.8 执行精确调节

要求

- 已在 CPU 中装载指令和工艺对象。

步骤

要确定操作点处的最佳 PID 参数，请按以下步骤操作：

1. 单击“Start”图标。

   如果不存在在线连接，则将建立在线连接。 系统会记录设定值、过程值和输出值的当前值。

2. 从“模式”(Mode) 下拉列表中选择“精确调节”(Fine tuning)。

   TCONT_CP 准备好执行调节。

3. 在“输出值跳跃”(Output value jump) 字段中，指定输出值的增加量。

4. 单击 “启动调节”(Start tuning) 图标。

   精确调节启动。 显示调节的状态。

7.3.3.9 取消预调节或精确调节

要取消预调节或精确调节，请单击 图标“停止调节”(Stop tuning)。

如果尚未计算和存储 PID 参数，TCONT_CP 将在自动模式下启动 (LMN = LMN0 + TUN_DLMN)。 如果调节之前控制器处于手动模式，则将输出旧的手动调节变量。

如果已保存计算出的 PID 参数，TCONT_CP 将在自动模式下启动，并使用先前确定的 PID 参数。

7.3.3.10 在控制模式下手动精确调节

可以采用以下措施以实现无超调的设定值响应：

调整控制区

调节过程中，“TCONT_CP”确定控制区 CON_ZONE，如果过程类型适合（过程类型 I 和 II）并且使用 PID 控制器 (CONZ_ON = TRUE)，则会将其激活： 在控制模式下，您可以修改控制区或将其完全关闭（设置 CONZ_ON = FALSE）。

说明

使用更高阶过程（过程类型 III）激活控制区通常不会带来任何好处，因为控制区随即会大于使用 100% 受控变量可达到的控制范围。 激活 PI 控制器的控制区也没有任何优势。

手动开启控制区之前，请确保控制区不会过窄。 如果控制区设置过窄，受控变量和过程值将发生振荡。

使用 PFAC_SP 实现控制响应的连续衰减

控制响应可使用 PFAC_SP 参数进行衰减。 该参数可指定对设定值阶跃变化有效的比例分量的百分比。

无论何种过程类型，都会通过调节功能将 PFAC_SP 设置为默认值 0.8；如果需要，您可以稍后修改该值。 为了在设定值阶跃变化（使用其他正确的控制器参数）期间将超调量限制在 2% 左右，下列值适用于 PFAC_SP：

	过程类型 I	过程类型 II	过程类型 III
	典型温度过程	中间范围	更高阶温度过程
PI	0.8	0.82	0.8
PID	0.6	0.75	0.96

调整默认因子 (0.8)，特别是在下列情况下：

- 过程类型 I，其中 PID (0.8 →0.6)： 在 PFAC_SP = 0.8 的情况下，控制区内的设定点阶跃更改仍会导致 18% 左右的超调量。
- 过程类型 III，其中 PID (0.8 →0.96)： 在 PFAC_SP = 0.8 的情况下，设定值阶跃变化会非常剧烈地衰减。 这将严重减缓响应时间。

控制参数的衰减

当闭环控制电路发生振荡或设定值阶跃变化后出现超调量时，可以减少控制器的 GAIN（例如，减少到原始值的 80%）并增加积分时间（例如，增加到原始值的 150%）。 如果连续控制器的模拟量输出值经脉冲整形器转化为二进制动作信号，那么量化噪声可能会导致小幅永久振荡。 可以通过增加控制器死区 DEADB_W 来消除这种振荡。

修改控制参数

按照下列步骤修改控制参数：

1. 使用 SAVE_PAR 保存当前参数。
2. 修改参数。
3. 测试控制响应。

如果新参数设置比旧参数设置差，请使用 UNDO_PAR 恢复旧参数。

7.3.3.11 手动执行精确调节

要求

- 已将指令和工艺对象加载到 CPU。

步骤

要手动确定最优 PID 参数，请按以下步骤操作：

1. 单击“Start”图标。

   如果不存在在线连接，则将建立在线连接。 系统会记录设定值、过程值和输出值的当前值。

2. 从“模式”(Mode) 下拉列表中选择“手动”(Manual)。
3. 输入新的 PID 参数。
4. 在“调节”(Tuning) 组中单击图标 “将参数发送到 CPU”(Send parameter to CPU)。
5. 在“当前值”(Current values) 组中选中“更改设定值”(Change setpoint) 复选框。
6. 输入新设定值并在“当前值”(Current values) 组中单击图标 。
7. 清除“手动模式”(Manual mode) 复选框。

   此时控制器使用新 PID 参数工作并控制新设定值。

8. 检查 PID 参数的质量以检查曲线点。
9. 重复步骤 3 到 8，直至对控制器结果满意为止。

7.4 TCONT_S

7.4.1 工艺对象 TCONT_S

工艺对象 TCONT_S 提供了一个用于控制具有积分行为的执行器的步进控制器，并且可用于通过二进制输出值输出信号控制工艺温度过程。 该工艺对象对应于 TCONT_S 指令的背景数据块。 其工作原理基于采样控制器的 PI 控制算法。步进控制器在没有位置反馈信号的情况下运行。手动和自动模式均可。

S7-1500
工艺对象的所有参数和变量均具有保持性，在完整下载 TCONT_S 的前提下，只能在下载到设备期间更改这些数据。

参见

软件控制器概述 (页 41)

添加工艺对象 (页 44)

组态工艺对象 (页 48)

TCONT_S (页 559)

将工艺对象下载到设备 (页 76)

7.4.2 组态控制器误差 TCONT_S

使用外设过程值

要使用输入参数 PV_PER，请执行以下步骤：

1. 从“源”(Source) 列表中选择条目“外设”(Periphery)。
2. 选择“传感器类型”。
   对于不同类型的传感器，过程值会根据不同的公式进行标定。
   - 标准
     热电偶；PT100/NI100

     PV = 0.1 × PV_PER × PV_FAC + PV_OFFS
   - 冷却；
     PT100/NI100

     PV = 0.01 × PV_PER × PV_FAC + PV_OFFS
   - 电流/电压

     PV = 100/27648 × PV_PER × PV_FAC + PV_OFFS
3. 输入用于标定外设过程值的因子和偏移量。

使用内部过程值

要使用输入参数 PV_IN，请执行以下步骤：

1. 从“源”(Source) 列表中选择条目“内部”(Internal)。

控制偏差

根据以下要求设置死区范围：

- 过程值信号含有噪声。
- 控制器增益很高。
- 微分作用激活。

这种情况下，过程值的噪声分量会导致输出值出现巨大偏差。 死区可抑制控制器处于稳态的噪声分量。 死区范围指定死区的大小。 死区范围为 0.0 时，死区关闭。

参见

TCONT_S 的工作模式 (页 561)

7.4.3 组态控制器算法 TCONT_S

常规步骤

1. 输入“PID 算法采样时间”。
   控制器采样时间不应超过确定的控制器积分作用时间 (TI) 的 10 %。
2. 如果该控制器结构包含比例作用，请输入“比例增益”。
   如果比例增益为负，则规则含义为相反的含义。

比例作用

如果设定值发生变化，可能会导致比例作用超调。 通过比例作用的权重，可选择设定值发生变化时比例作用的响应程度。 通过补偿积分作用可弱化比例作用。

1. 要弱化应对设定值变化的比例作用，可相应地输入“比例作用权重”。
   - 1.0: 应对设定值变化的比例作用完全有效
   - 0.0: 应对设定值变化的比例作用无效

积分作用

1. 如果该控制器结构包含积分作用，请输入“积分作用时间”。
   积分作用时间为 0.0 时，积分作用关闭。

参见

TCONT_S 的工作模式 (页 561)

7.4.4 组态调节值 TCONT_S

脉冲发生器

1. 输入最短脉冲持续时间和最短暂停持续时间。
   值必须大于等于输入参数 CYCLE 的周期时间。 因此，操作频率会降低。
2. 输入电机设定时间。
   值必须大于等于输入参数 CYCLE 的周期时间。

参见

TCONT_S 的工作模式 (页 561)

7.4.5 调试 TCONT_S

要求

- 已将指令和工艺对象加载到 CPU。

步骤

要手动确定最优 PID 参数，请按以下步骤操作：

1. 单击“Start”图标。

   如果不存在在线连接，则将建立在线连接。 系统会记录设定值、过程值和输出值的当前值。
2. 在“P”、“I”和“加权比例作用”(weighting proportional action) 字段中输入新的 PID 参数。
3. 在“调节”(Tuning) 组中单击图标 “将参数发送到 CPU”(Send parameter to CPU)。
4. 在“当前值”(Current values) 组中选中“更改设定值”(Change setpoint) 复选框。
5. 输入新设定值并在“当前值”(Current values) 组中单击图标 。
6. 清除“手动模式”(Manual mode) 复选框。

   这时控制器使用新参数工作并控制新设定值。
7. 检查 PID 参数的质量以检查曲线点。
8. 重复步骤 2 到 6，直至对控制器结果满意为止。

指令

8

8.1 PID_Compact

8.1.1 PID_Compact 的新特性

PID_Compact V2.3

- **从“未激活”工作模式切换到“自动模式”时输出值的响应**

  添加了新选项 IntegralResetMode = 4 = 4，并将其定义为默认设置。如果 IntegralResetMode = 4，从“未激活”工作模式切换到“自动模式”时会自动预分配积分作用，以便控制偏差导致带有相同符号的输出值发生跳变。

- **自动模式下积分作用的初始化**

  可以通过变量 OverwriteInitialOutputValue 和 PIDCtrl.PIDInit 在自动模式下对积分作用进行初始化。这简化了使用 PID_Compact 进行超驰控制的过程。

PID_Compact V2.2

- **使用 S7-1200**

  自 PID_Compact V2.2 起，固件为 4.0 或更高版本的 S7-1200 上也可以使用具有 V2 功能的指令。

PID_Compact V2.0

- **对错误的响应**

  对错误的响应已经过全面改进。在默认设置下，PID_Compact 现在的响应方式具有更强的容错性。将 PID_Compact V1.X 从 S7-1200 CPU 复制到 S7-1500 CPU 时，设置此响应。

注意
您的系统可能已损坏。 如果使用默认设置，则超过过程值的限值时，PID_Compact 保持自动模式。这可能损坏您的系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

  Error 参数指示是否存在错误处于未决状态。当错误不再处于未决状态时，Error = FALSE。ErrorBits 参数显示发生的具体错误。使用 ErrorAck 在不重启控制器或清除积分作用的情况下确认错误和警告。切换工作模式不会清除处于非未决状态的错误。

  可使用 SetSubstituteOutput 和 ActivateRecoverMode 来组态对错误的响应。

- **替代输出值**

  可以组态出现错误时要输出的替代输出值。

- **切换工作模式**

  在 Mode 的输入/输出参数处指定工作模式，并通过 ModeActivate 的上升沿启动该工作模式。sRet.i_Mode 变量已被忽略。

- **多重背景功能**

  可将 PID_Compact 作为多重背景数据块进行调用。这种情况下，不会创建任何工艺对象，也没有任何参数分配接口或调试接口可用。必须直接在多重背景数据块中为 PID_Compact 分配参数，并通过监视表格进行调试。

- **启动特性**

  如果 RunModeByStartup = TRUE，则通过 Mode 参数指定的工作模式也将在 Reset 的下降沿和 CPU 冷启动期间启动。

- **ENO 特性**

  ENO 根据工作模式进行设置。

  如果 State = 0，那么 ENO = FALSE。

  如果 State ≠ 0，那么 ENO = TRUE。

- **在调节期间指定设定值**

  在 CancelTuningLevel 变量中进行调节期间，组态允许的设定值波动。

- **输出值限值的值范围**

  在输出值限值范围内，值 0.0 不再下降。

- **预分配积分作用**

  从“未激活”工作模式切换到“自动模式”时，可使用变量 IntegralResetMode 和 OverwriteInitialOutputValue 确定积分作用的预分配。

- **启用扰动变量**

  可在 Disturbance 参数中启用扰动变量。

- **PID 参数的默认值**

  下列默认设置已更改：

  – 比例作用权重 (PWeighting)，从 0.0 到 1.0

  – 微分作用权重 (DWeighting)，从 0.0 到 1.0

  – 微分延迟系数 (TdFiltRatio)，从 0.0 到 0.2

- **重命名变量**

  已为静态变量指定新名称，这些名称与 PID_3Step 兼容。

PID_Compact V1.2

- **CPU 启动时的手动模式**

  如果 CPU 启动时 ManualEnable = TRUE，则 PID_Compact 以手动模式启动。并非一定需要 ManualEnable 出现上升沿。

- **预调节**

  如果在预调节期间关闭 CPU，当重新开启 CPU 时预调节会再次启动。

PID_Compact V1.1

- **CPU 启动时的手动模式**

  CPU 启动时，仅在 ManualEnable 出现上升沿时，PID_Compact 才切换到手动模式。没有上升沿时，PID_Compact 在 ManualEnable 为 FALSE 的上一个工作模式下启动。

- **对复位的响应**

  Reset 出现上升沿时会复位错误和警告，并清除积分作用。复位时出现下降沿会触发切换到最近激活的工作模式。

- **过程值的默认上限**

  r_Pv_Hlm 的默认值已更改为 120.0。

- **监视采样时间**

  – 当前采样时间大于等于当前平均值的 1.5 倍，或者当前采样时间小于等于当前平均值的 0.5 倍时，不再输出错误。自动模式下的采样时间可能有很大偏离。

  – PID_Compact 与 V2.0 或更高版本的 FW 兼容。

- **访问变量**

  现在可以在用户程序中使用以下变量。

  – i_Event_SUT

  – i_Event_TIR

  – r_Ctrl_Ioutv

- **故障排除**

  当最短开启时间不等于最短关闭时间时，PID_Compact 现在会输出正确的脉冲。

8.1.2 与 CPU 和 FW 的兼容性

下表显示了 PID_Compact 的每个版本可用于哪种 CPU。

CPU	FW	PID_Compact
S7-1200	V4.2 或更高版本	V2.3 V2.2 V1.2
	V4.0 到 V4.1	V2.2 V1.2
	V3.x	V1.2 V1.1
	V2.x	V1.2 V1.1
	V1.x	V1.0
S7-1500	V2.0 或更高版本	V2.3 V2.2 V2.1 V2.0
	V1.5 到 V1.8	V2.2 V2.1 V2.0
	V1.1	V2.1 V2.0
	V1.0	V2.0

8.1.3 PID_Compact V2

8.1.3.1 PID_Compact V2 的说明

说明

PID_Compact 指令提供了一种可对具有比例作用的执行器进行集成调节的 PID 控制器。

存在下列工作模式：

- 未激活
- 预调节
- 精确调节
- 自动模式
- 手动模式
- 带错误监视的替代输出值

有关工作模式的详细信息，请参见 State 参数。

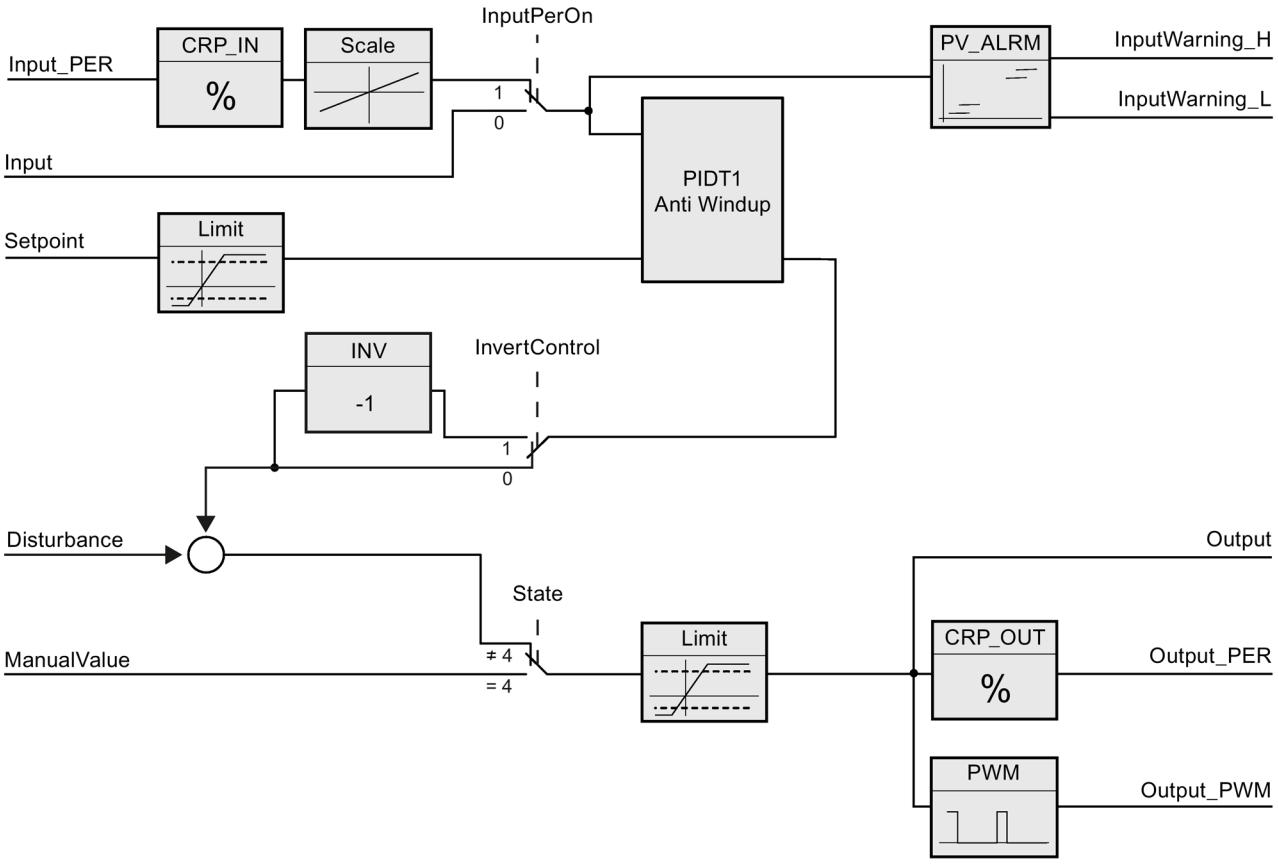
PID 算法

PID_Compact 是一种具有抗积分饱和功能并且能够对比例作用和微分作用进行加权的 PIDT1 控制器。 PID 算法根据以下等式工作：

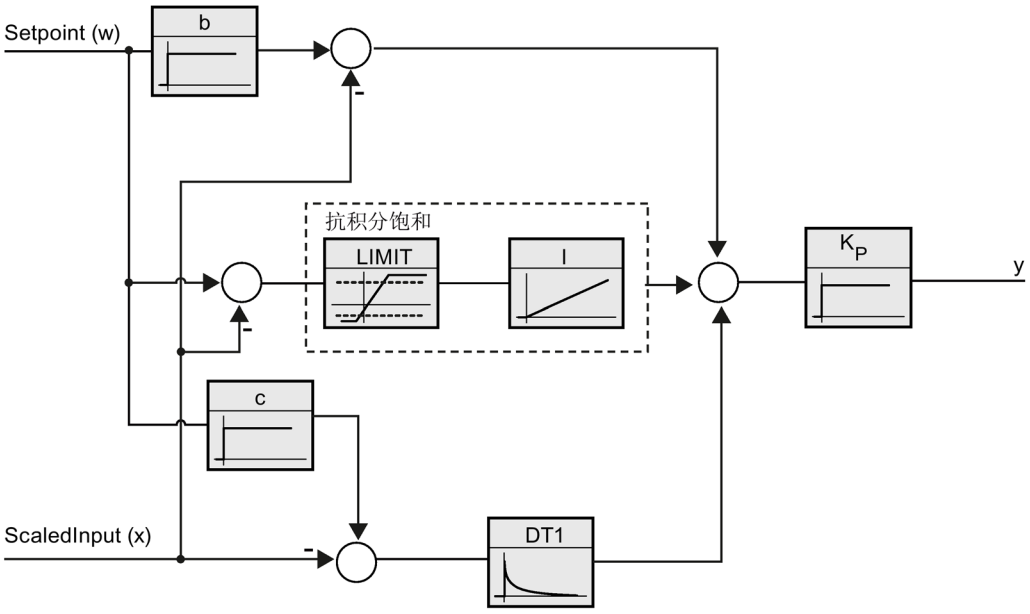
$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
y	PID 算法的输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分作用时间
T_D	微分作用时间
a	微分延迟系数（微分延迟 T1 = a × T_D）
c	微分作用权重

PID_Compact 方框图



带抗积分饱和的 PIDT1 的方框图



调用

在周期中断 OB 的恒定时间范围内调用 PID_Compact。

如果将 PID_Compact 作为多重背景数据块调用，将不会创建任何工艺对象。 没有参数分配接口或调试接口可用。 必须直接在多重背景数据块中为 PID_Compact 分配参数，并通过监视表格进行调试。

下载到设备

仅当完全下载 PID_Compact 后，才能更新保持性变量的实际值。

将工艺对象下载到设备 (页 76)

启动

CPU 启动时，PID_Compact 以保存在 Mode 输入/输出参数中的工作模式启动。 要在启动期间切换到“未激活”工作模式，应设置 RunModeByStartup = FALSE。

对错误的响应

在自动模式下和调试期间，对错误的响应取决于 SetSubstituteOutput 和 ActivateRecoverMode 变量。 在手动模式下，该响应与 SetSubstituteOutput 和 ActivateRecoverMode 变量无关。 如果 ActivateRecoverMode = TRUE 变量，则该响应还取决于所发生的错误。

SetSubstituteOutput	ActivateRecoverMode	组态编辑器 > 输出值 > 将 Output 设置为	响应
不相关	FALSE	零（未激活）	切换到“未激活”模式 (State = 0) 值 0.0 0 传送到执行器。
FALSE	TRUE	错误未决时的当前输出值	切换到“带错误监视的替代输出值”模式 (State = 5) 当错误未决时，当前输出值会传送到执行器。
TRUE	TRUE	错误未决时的替代输出值	切换到“带错误监视的替代输出值”模式 (State = 5) 当错误未决时，SubstituteOutput 中的值会传送到执行器。

在手动模式下，PID_Compact 使用 ManualValue 作为输出值，除非 ManualValue 无效。 如果 ManualValue 无效，将使用 SubstituteOutput。 如果 ManualValue 和 SubstituteOutput 无效，将使用 Config.OutputLowerLimit。

Error 参数指示是否存在错误处于未决状态。 当错误不再处于未决状态时，Error = FALSE。 ErrorBits 参数显示了已发生的错误。通过 Reset 或 ErrorAck 的上升沿来复位 ErrorBits。

8.1.3.2 PID_Compact V2 的工作模式

监视过程值的限值

在 Config.InputUpperLimit 和 Config.InputLowerLimit 变量中指定过程值的上限和下限。如果过程值超出这些限值，将出现错误 (ErrorBits = 0001h)。

在 Config.InputUpperWarning 和 Config.InputLowerWarning 变量中指定过程值的警告上限和警告下限。如果过程值超出这些警告限值，将发生警告 (Warning = 0040h)，并且 InputWarning_H 或 InputWarning_L 输出参数会更改为 TRUE。

限制设定值

可在 Config.SetpointUpperLimit 和 Config.SetpointLowerLimit 变量中指定设定值的上限和下限。PID_Compact 会自动将设定值限制在过程值的限值范围内。可以将设定值限制在更小的范围内。PID_Compact 会检查此范围是否处于过程值的限值范围内。如果设定值超出这些限值，上限和下限将用作设定值，并且输出参数 SetpointLimit_H 或 SetpointLimit_L 将设置为 TRUE。

在所有操作模式下均限制设定值。

限制输出值

在 Config.OutputUpperLimit 变量和 Config.OutputLowerLimit 变量中指定输出值的上限和下限。Output、ManualValue 和 SubstituteOutput 限制为这些值。输出值限值必须与控制逻辑相匹配。

有效的输出值限值取决于所用的 Output。

Output	-100.0 至 100.0%
Output_PER	-100.0 至 100.0%
Output_PWM	0.0 至 100.0%

规则：

OutputUpperLimit > OutputLowerLimit

说明

与两个或多个执行器结合使用

PID_Compact 不适合与两个或多个执行器结合使用（例如，在加热/制冷应用中），因为不同的执行器需要不同的 PID 参数以实现良好的控制响应。针对两个执行器在相反方向起作用的应用，使用 PID_Temp。

替代输出值

出现错误时，PID_Compact 可输出您在 SubstituteOutput 变量处定义的替代输出值。替代输出值必须处于输出值的限值范围内。

监视信号有效性

使用以下参数时，监视其有效性：

- Setpoint
- Input
- Input_PER
- Disturbance
- ManualValue
- SubstituteOutput
- Output
- Output_PER
- Output_PWM

PID_Compact 采样时间的监视

理想情况下，采样时间等于调用 OB 的周期时间。PID_Compact 指令测量两次调用之间的时间间隔。这就是当前采样时间。每次切换工作模式以及初始启动期间，平均值由前 10 个采样时间构成。当前采样时间与该平均值之间的差值过大时会触发错误 (Error = 0800h)。

如果存在以下情况，调节期间将发生错误：

- 新平均值 >= 1.1 x 原平均值
- 新平均值 <= 0.9 x 原平均值

如果存在以下情况，将在自动模式下发生错误：

- 新平均值 >= 1.5 x 原平均值
- 新平均值 <= 0.5 x 原平均值

如果禁用采样时间监视 (CycleTime.EnMonitoring = FALSE)，则也可在 OB1 中调用 PID_Compact。由于采样时间发生偏离，因此随后必须接受质量较低的控制。

PID 算法的采样时间

受控系统需要一定的时间来对输出值的变化做出响应。因此，建议不要在每次循环中都计算输出值。PID 算法的采样时间是两次计算输出值之间的时间。该时间在调节期间进行计算，并舍入为循环时间的倍数。PID_Compact 的所有其它功能会在每次调用时执行。

如果使用 Output_PWM，输出信号的精度将由 PID 算法采样时间与 OB 的周期时间之比来确定。该周期时间至少应为 PID 算法采样时间的 10 倍。

控制逻辑

通常，可通过增大输出值来增大过程值。这种做法称为常规控制逻辑。对于制冷和放电控制系统，可能需要反转控制逻辑。PID_Compact 不使用负比例增益。如果 InvertControl = TRUE，则不断增大的控制偏差将导致输出值减小。在预调节和精确调节期间还会考虑控制逻辑。

8.1.3.3 PID_Compact V2 的输入参数

表格 8- 1

参数	数据类型	默认值	说明
Setpoint	REAL	0.0	PID 控制器在自动模式下的设定值
Input	REAL	0.0	用户程序的变量用作过程值的源。 如果正在使用参数 Input，则必须设置 Config.InputPerOn = FALSE。
Input_PER	INT	0	模拟量输入用作过程值的源。 如果正在使用参数 Input_PER，则必须设置 Config.InputPerOn = TRUE。
Disturbance	REAL	0.0	扰动变量或预控制值
ManualEnable	BOOL	FALSE	• 出现 FALSE -> TRUE 沿时会激活“手动模式”，而 State = 4 和 Mode 保持不变。 只要 ManualEnable = TRUE，便无法通过 ModeActivate 的上升沿或使用调试对话框来更改工作模式。 • 出现 TRUE -> FALSE 沿时会激活由 Mode 指定的工作模式。 建议只使用 ModeActivate 更改工作模式。
ManualValue	REAL	0.0	手动值 该值用作手动模式下的输出值。 允许介于 Config.OutputLowerLimit 与 Config.OutputUpperLimit 之间的值。
ErrorAck	BOOL	FALSE	• FALSE -> TRUE 沿 将复位 ErrorBits 和 Warning。

参数	数据类型	默认值	说明
Reset	BOOL	FALSE	重新启动控制器。 • FALSE -> TRUE 沿 – 切换到“未激活”模式 – 将复位 ErrorBits 和 Warnings。 • 只要 Reset = TRUE， – PID_Compact 将保持在“未激活”模式下 (State = 0)。 – 无法通过 Mode 和 ModeActivate 或 ManualEnable 更改工作模式。 – 无法使用调试对话框。 • TRUE -> FALSE 沿 – 如果 ManualEnable = FALSE，则 PID_Compact 会切换到保存在 Mode 中的工作模式。 – 如果 Mode = 3，会将积分作用视为已通过变量 IntegralResetMode 进行组态。
ModeActivate	BOOL	FALSE	• FALSE -> TRUE 沿 PID_Compact 将切换到保存在 Mode 参数中的工作模式。

8.1.3.4 PID_Compact V2 的输出参数

表格 8- 2

Parameter	数据类型	默认值	说明
ScaledInput	REAL	0.0	标定的过程值
可同时使用“Output”、“Output_PER”和“Output_PWM”输出。			
Output	REAL	0.0	REAL 形式的输出值
Output_PER	INT	0	模拟量输出值
Output_PWM	BOOL	FALSE	脉宽调制输出值 输出值由变量开关时间形成。
SetpointLimit_H	BOOL	FALSE	如果 SetpointLimit_H = TRUE，则说明达到了设定值的绝对上限 (Setpoint ≥ Config.SetpointUpperLimit)。 此设定值将限制为 Config.SetpointUpperLimit 。
SetpointLimit_L	BOOL	FALSE	如果 SetpointLimit_L = TRUE，则说明已达到设定值的绝对下限 (Setpoint ≤ Config.SetpointLowerLimit)。 此设定值将限制为 Config.SetpointLowerLimit 。
InputWarning_H	BOOL	FALSE	如果 InputWarning_H = TRUE，则说明过程值已达到或超出警告上限。
InputWarning_L	BOOL	FALSE	如果 InputWarning_L = TRUE，则说明过程值已经达到或低于警告下限。
State	INT	0	State 参数 (页 296)显示了 PID 控制器的当前工作模式。 可使用输入参数 Mode 和 ModeActivate 处的上升沿更改工作模式。 • State = 0：未激活 • State = 1：预调节 • State = 2：精确调节 • State = 3：自动模式 • State = 4：手动模式 • State = 5：带错误监视的替代输出值

Parameter	数据类型	默认值	说明
Error	BOOL	FALSE	如果 Error = TRUE，则此周期内至少有一条错误消息处于未决状态。
ErrorBits	DWORD	DW#16#0	ErrorBits 参数 (页 301)显示了处于未决状态的错误消息。通过 Reset 或 ErrorAck 的上升沿来保持并复位 ErrorBits。

8.1.3.5 PID_Compact V2 的输入/输出参数

表格 8- 3

Parameter	数据类型	默认值	说明
Mode	INT	4	在 Mode 上，指定 PID_Compact 将转换到的工作模式。 选项包括： • Mode = 0：未激活 • Mode = 1：预调节 • Mode = 2：精确调节 • Mode = 3：自动模式 • Mode = 4：手动模式 工作模式由以下沿激活： • ModeActivate 的上升沿 • Reset 的下降沿 • ManualEnable 的下降沿 • 如果 RunModeByStartup = TRUE，则冷启动 CPU。 保持 Mode。 有关工作模式的详细说明，请参见模式 V2 的参数状态 (页 296)。

参见

模式 V2 的参数状态 (页 296)

8.1.3.6 PID_Compact V2 的静态变量

不得更改未列出的变量。这些变量仅供内部使用。

变量	数据类型	默认值	说明
IntegralResetMode	INT	V2.2 及之前的版本：1， V2.3 或更高版本：4	IntegralResetMode V2 变量 (页 307) 用于确定从“未激活”工作模式切换到“自动模式”时如何预分配积分作用 PIDCtrl.IntegralSum。此设置仅在一个周期内有效。 选项包括： • IntegralResetMode = 0：平滑 • IntegralResetMode = 1：删除 • IntegralResetMode = 2：保持 • IntegralResetMode = 3：预分配 • IntegralResetMode = 4：类似于设定值更改（仅适用于版本 2.3 及更高版本的 PID_Compact）
OverwriteInitialOutputValue	REAL	0.0	如果满足以下条件之一，则会自动预分配 PIDCtrl.IntegralSum 的积分作用，如同在上一周期中 Output = OverwriteInitialOutputValue： • 从“未激活”工作模式切换到“自动模式”时 IntegralResetMode = 3。 • 参数 Reset 的 TRUE -> FALSE 沿并且参数 Mode = 3 • 在“自动模式”下 PIDCtrl.PIDInit = TRUE（自 PID_Compact 版本 2.3 起可用）
RunModeByStartup	BOOL	TRUE	CPU 重启后，激活 Mode 参数中的工作模式。 如果 RunModeByStartup = TRUE，PID_Compact 将在 CPU 启动后以保存在模式参数中的工作模式启动。 如果 RunModeByStartup = FALSE，PID_Compact 在 CPU 启动后仍保持“未激活”模式下。

变量	数据类型	默认值	说明
LoadBackUp	BOOL	FALSE	如果 LoadBackUp = TRUE，则重新加载上一个 PID 参数集。该设置在最后一次调节前保存。LoadBackUp 自动设置回 FALSE。
PhysicalUnit	INT	0	过程值和设定值的测量单位，例如 ºC 或 ºF。
PhysicalQuantity	INT	0	过程值和设定值的物理量，如温度。
ActivateRecoverMode	BOOL	TRUE	变量 ActivateRecoverMode V2 (页 304) 确定对错误的响应方式。
Warning	DWORD	0	自 Reset = TRUE 或 ErrorAck =TRUE 起，变量 Warning V2 (页 306)会显示警告。保持 Warning。
Progress	REAL	0.0	百分数形式的调节进度 (0.0 - 100.0)
CurrentSetpoint	REAL	0.0	CurrentSetpoint 始终显示当前设定值。调节期间该值处于冻结状态。
CancelTuningLevel	REAL	10.0	调节期间允许的设定值拐点。出现以下情况之前，不会取消调节： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel
SubstituteOutput	REAL	0.0	替代输出值 满足以下条件时，使用替代输出值： • 错误发生在自动模式下。 • SetSubstituteOutput = TRUE • ActivateRecoverMode = TRUE

变量	数据类型	默认值	说明
SetSubstituteOutput	BOOL	TRUE	如果 SetSubstituteOutput = TRUE 且 ActivateRecoverMode = TRUE，则只要错误未决，便会输出已组态的替代输出值。 如果 SetSubstituteOutput = FALSE 且 ActivateRecoverMode = TRUE，则只要错误未决，执行器便会仍保持为当前输出值。 如果 ActivateRecoverMode = FALSE，则 SetSubstituteOutput 无效。 如果 SubstituteOutput 无效 (ErrorBits = 20000h)，则不能输出替代输出值。
Config.InputPerOn	BOOL	TRUE	如果 InputPerOn = TRUE，则使用参数 Input_PER。如果 InputPerOn = FALSE，则使用参数 Input。
Config.InvertControl	BOOL	FALSE	反转控制逻辑 如果 InvertControl = TRUE，则不断增大的控制偏差将导致输出值减小。
Config.InputUpperLimit	REAL	120.0	过程值的上限 监控 Input 和 Input_PER，以确保符合此限制。 在 I/O 输入中，过程值最大可超出标准范围 18%（过范围）。此预分配可确保因超出“过程值上限”而不再报告错误。仅识别断线和短路，然后 PID_Compact 将根据已组态的错误响应方式进行响应。 InputUpperLimit > InputLowerLimit
Config.InputLowerLimit	REAL	0.0	过程值的下限 监控 Input 和 Input_PER，以确保符合此限制。 InputLowerLimit < InputUpperLimit

变量	数据类型	默认值	说明
Config.InputUpperWarning	REAL	3.402822e+38	过程值的警告上限 如果设置的 InputUpperWarning 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作警告上限。 如果组态的 InputUpperWarning 值位于过程值的限值范围内，则该值将用作警告上限。 InputUpperWarning > InputLowerWarning InputUpperWarning ≤ InputUpperLimit
Config.InputLowerWarning	REAL	-3.402822e+38	过程值的警告下限 如果设置的 InputLowerWarning 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作警告下限。 如果组态的 InputLowerWarning 值位于过程值的限值范围内，则该值将用作警告下限。 InputLowerWarning < InputUpperWarning InputLowerWarning ≥ InputLowerLimit
Config.OutputUpperLimit	REAL	100.0	输出值的上限 有关详细信息，请参见 OutputLowerLimit OutputUpperLimit > OutputLowerLimit
Config.OutputLowerLimit	REAL	0.0	输出值的下限 对于 Output 和 Output_PER，-100.0 到 +100.0 的值范围有效（包括零）。-100.0 时，Output_PER = -27648；+100.0 时，Output_PER = 27648。 对于 Output_PWM，则值范围 0.0 到 +100.0 适用。 输出值限值必须与控制逻辑相匹配。 OutputLowerLimit < OutputUpperLimit

变量	数据类型	默认值	说明
Config.SetpointUpperLimit	REAL	3.402822e+38	设定值的上限 如果组态的 SetpointUpperLimit 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作设定值的上限。 如果组态的 SetpointUpperLimit 值位于过程值的限值范围内，则该值将用作设定值的上限。
Config.SetpointLowerLimit	REAL	-3.402822e+38	设定值的下限 如果设置的 SetpointLowerLimit 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作设定值的下限。 如果设置的 SetpointLowerLimit 值位于过程值的限值范围内，则该值将用作设定值的下限。
Config.MinimumOnTime	REAL	0.0	脉宽调制的最小 ON 时间（秒）舍入为 MinimumOnTime = n×CycleTime.Value
Config.MinimumOffTime	REAL	0.0	脉宽调制的最小 OFF 时间（秒）舍入为 MinimumOffTime = n×CycleTime.Value
Config.InputScaling.UpperPointIn	REAL	27648.0	标定的 Input_PER 上限 根据以下两个值对将 Input_PER 转换为百分数：UpperPointOut 和 UpperPointIn；LowerPointOut 和 LowerPointIn。
Config.InputScaling.LowerPointIn	REAL	0.0	标定的 Input_PER 下限 根据以下两个值对将 Input_PER 转换为百分数：UpperPointOut 和 UpperPointIn；LowerPointOut 和 LowerPointIn。
Config.InputScaling.UpperPointOut	REAL	100.0	标定的过程值的上限 根据以下两个值对将 Input_PER 转换为百分数：UpperPointOut 和 UpperPointIn；LowerPointOut 和 LowerPointIn。

变量	数据类型	默认值	说明
Config.InputScaling.LowerPointOut	REAL	0.0	标定的过程值的下限 根据以下两个值对将 Input_PER 转换为百分数：UpperPointOut 和 UpperPointIn；LowerPointOut 和 LowerPointIn。
CycleTime.StartEstimation	BOOL	TRUE	如果 CycleTime.StartEstimation = TRUE，将开始自动确定循环时间。完成测量后，CycleTime.StartEstimation = FALSE。
CycleTime.EnEstimation	BOOL	TRUE	如果 CycleTime.EnEstimation = TRUE，则计算 PID_Compact 采样时间。 如果 CycleTime.EnEstimation = FALSE，则不计算 PID_Compact 采样时间，并且您需要手动更正 CycleTime.Value 的组态。
CycleTime.EnMonitoring	BOOL	TRUE	如果 CycleTime.EnMonitoring = FALSE，则不会监视 PID_Compact 采样时间。如果不能在采样时间内执行 PID_Compact，则不会输出错误 (ErrorBits=0800h)，PID_Compact 也不会切换到“未激活”模式。
CycleTime.Value	REAL	0.1	PID_Compact 采样时间（以秒为单位） CycleTime.Value 会自动确定，通常等于调用 OB 的循环时间。
CtrlParamsBackUp.Gain	REAL	1.0	保存的比例增益 LoadBackUp = TRUE 时，可以从 CtrlParamsBackUp 结构中重新加载值。
CtrlParamsBackUp.Ti	REAL	20.0	保存的积分作时间 [s]
CtrlParamsBackUp.Td	REAL	0.0	保存的微分作时间 [s]
CtrlParamsBackUp.TdFiltRatio	REAL	0.2	保存的微分延时系数
CtrlParamsBackUp.PWeighting	REAL	1.0	保存的比例作用权重因子
CtrlParamsBackUp.DWeighting	REAL	1.0	保存的微分作用权重因子
CtrlParamsBackUp.Cycle	REAL	1.0	保存的 PID 算法的采样时间

变量	数据类型	默认值	说明
PIDSelfTune.SUT.CalculateParams	BOOL	FALSE	受控系统的属性在调节期间保存。如果 SUT.CalculateParams = TRUE，则根据这些属性重新计算预调节的参数。这样无需重复进行控制器调节，就可以更改参数计算方法。 计算后，SUT.CalculateParams 将设置为 FALSE。
PIDSelfTune.SUT.TuneRule	INT	0	预调节期间用于计算参数的方法： • SUT.TuneRule = 0：根据 Chien、Hrones 和 Reswick 计算 PID • SUT.TuneRule = 1：根据 Chien、Hrones 和 Reswick 计算 PI
PIDSelfTune.SUT.State	INT	0	SUT.State 变量指示当前的预调节阶段： • State = 0：初始化预调节 • State = 100：计算标准偏差 • State = 200：查找拐点 • State = 9900：预调节成功 • State = 1：预调节未成功
PIDSelfTune.TIR.RunIn	BOOL	FALSE	利用 RunIn 变量，您可以指定无需预调节也可执行精确调节。 • RunIn = FALSE 在未激活模式或手动模式下启动精确调节时，将启动预调节。如果不满足预调节的要求，则 PID_Compact 的响应将类似于 RunIn = TRUE 时的响应。 如果精确调节在自动模式下启动，系统将使用现有的 PID 参数来控制设定值。 之后才会启动精确调节。如果无法实现预调节，PID_Compact 将切换到调节开始时的模式。 • RunIn = TRUE 将跳过预调节。PID_Compact 将尝试利用最小或最大输出值来达到设定值。这可能会增加超调量。随后将自动启动精确调节。 精确调节后，RunIn 将设置为 FALSE。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.CalculateParams	BOOL	FALSE	受控系统的属性在调节期间保存。如果 TIR.CalculateParams = TRUE，则根据这些属性重新计算精确调节的参数。这样无需重复进行控制器调节，就可以更改参数计算方法。 计算后，TIR.CalculateParams 将设置为 FALSE。
PIDSelfTune.TIR.TuneRule	INT	0	精确调节期间用于计算参数的方法： • TIR.TuneRule = 0：PID 自动 • TIR.TuneRule = 1：PID 快速 • TIR.TuneRule = 2：PID 慢速 • TIR.TuneRule = 3：Ziegler-Nichols PID • TIR.TuneRule = 4：Ziegler-Nichols PI • TIR.TuneRule = 5：Ziegler-Nichols P
PIDSelfTune.TIR.State	INT	0	TIR.State 变量指示当前的精确调节阶段： • State = -100：无法进行精确调节。将首先执行预调节。 • State = 0：初始化精确调节 • State = 200：计算标准偏差 • State = 300：尝试达到设定值 • State = 400：尝试使用现有 PID 参数达到设定值 （如果预调节已成功） • State = 500：确定波动并计算参数 • State = 9900：精确调节已成功 • State = 1：精确调节未成功
PIDCtrl.IntegralSum	REAL	0.0	当前积分作用

变量	数据类型	默认值	说明
PIDCtrl.PIDInit	BOOL	FALSE	自 PID_Compact 版本 2.3 起 PIDCtrl.PIDInit 可用。 如果在“自动模式”下 PIDCtrl.PIDInit = TRUE，则会自动预分配 PIDCtrl.IntegralSum 的积分作用，如同在上一周期中 Output = OverwriteInitialOutputValue。这可用于通过 PID_Compact V2 进行超驰控制 (页 99)。
Retain.CtrlParams.Gain	REAL	1.0	有效的比例增益 要反转控制逻辑，使用 Config.InvertControl 变量。Gain 上的负值也会反转控制逻辑。我们建议您仅使用 InvertControl 设置控制逻辑。如果 InvertControl = TRUE 且 Gain < 0.0，则控制逻辑也会反转。 保持 Gain。
Retain.CtrlParams.Ti	REAL	20.0	• CtrlParams.Ti > 0.0：有效积分作用时间 • CtrlParams.Ti = 0.0：积分作用取消激活 保持 Ti。
Retain.CtrlParams.Td	REAL	0.0	• CtrlParams.Td > 0.0：有效的微分作用时间 • CtrlParams.Td = 0.0：微分作用取消激活 保持 Td。

变量	数据类型	默认值	说明
Retain.CtrlParams.TdFiltRatio	REAL	0.2	有效的微分延时系数 微分延迟系数用于延迟微分作用的生效。 微分延迟 = 微分作用时间 × 微分延迟系数 • 0.0：微分作用仅在一个周期内有效，因此几乎不产生影响。 • 0.5：此值经实践证明对于具有一个优先时间常量的受控系统非常有用。 • > 1.0：系数越大，微分作用的生效时间延迟越久。 保持 TdFiltRatio。
Retain.CtrlParams.PWeighting	REAL	1.0	有效的比例作用权重 比例作用随着设定值的变化而减弱。 允许使用 0.0 到 1.0 之间的值。 • 1.0：应对设定值变化的比例作用完全有效 • 0.0：应对设定值变化的比例作用无效 当过程值变化时，比例作用始终完全有效。 保持 PWeighting。
Retain.CtrlParams.DWeighting	REAL	1.0	有效的微分作用权重 微分作用随着设定值的变化而减弱。 允许使用 0.0 到 1.0 之间的值。 • 1.0：设定值变化时微分作用完全有效 • 0.0：设定值变化时微分作用不生效 当过程值变化时，微分作用始终完全有效。 保持 DWeighting。
Retain.CtrlParams.Cycle	REAL	1.0	有效的 PID 算法采样时间 在调节期间计算 CtrlParams.Cycle，并将其舍入为 CycleTime.Value 的整数倍。 保持 Cycle。

说明

请在“未激活”模式下更改本表列出的变量，以防 PID 控制器出现故障。

参见

变量 ActivateRecoverMode V2 (页 304)

变量 Warning V2 (页 306)

将工艺对象下载到设备 (页 76)

8.1.3.7 更改 PID_Compact V2 接口

下表显示了 PID_Compact 指令接口中的一些变化。

PID_Compact V1	PID_Compact V2	更改
Input_PER	Input_PER	数据类型由字改为整数
	Disturbance	新增
	ErrorAck	新增
	ModeActivate	新增
Output_PER	Output_PER	数据类型由字改为整数
Error	ErrorBits	重命名
	Error	新增
	Mode	新增
sb_RunModeByStartup	RunModeByStartup	功能
	IntegralResetMode	
	OverwriteInitialOutputValue	新增
	SetSubstituteOutput	新增
	CancelTuningLevel	新增
	SubstituteOutput	新增

下表显示了已重命名的变量。

PID_Compact V1.x	PID_Compact V2
sb_GetCycleTime	CycleTime.StartEstimation
sb_EnCyclEstimation	CycleTime.EnEstimation
sb_EnCyclMonitoring	CycleTime.EnMonitoring
sb_RunModeByStartup	RunModeByStartup
si_Unit	PhysicalUnit
si_Type	PhysicalQuantity
sd_Warning	Warning
sBackUp.r_Gain	CtrlParamsBackUp.Gain
sBackUp.r_Ti	CtrlParamsBackUp.Ti

PID_Compact V1.x	PID_Compact V2
sBackUp.r_Td	CtrlParamsBackUp.Td
sBackUp.r_A	CtrlParamsBackUp.TdFiltRatio
sBackUp.r_B	CtrlParamsBackUp.PWeighting
sBackUp.r_C	CtrlParamsBackUp.DWeighting
sBackUp.r_Cycle	CtrlParamsBackUp.Cycle
sPid_Calc.r_Cycle	CycleTime.Value
sPid_Calc.b_RunIn	PIDSelfTune.TIR.RunIn
sPid_Calc.b_CalcParamSUT	PIDSelfTune.SUT.CalculateParams
sPid_Calc.b_CalcParamTIR	PIDSelfTune.TIR.CalculateParams
sPid_Calc.i_CtrlTypeSUT	PIDSelfTune.SUT.TuneRule
sPid_Calc.i_CtrlTypeTIR	PIDSelfTune.TIR.TuneRule
sPid_Calc.r_Progress	Progress
sPid_Cmpt.r_Sp_Hlm	Config.SetpointUpperLimit
sPid_Cmpt.r_Sp_Llm	Config.SetpointLowerLimit
sPid_Cmpt.r_Pv_Norm_IN_1	Config.InputScaling.LowerPointIn
sPid_Cmpt.r_Pv_Norm_IN_2	Config.InputScaling.UpperPointIn
sPid_Cmpt.r_Pv_Norm_OUT_1	Config.InputScaling.LowerPointOut
sPid_Cmpt.r_Pv_Norm_OUT_2	Config.InputScaling.UpperPointOut
sPid_Cmpt.r_Lmn_Hlm	Config.OutputUpperLimit
sPid_Cmpt.r_Lmn_Llm	Config.OutputLowerLimit
sPid_Cmpt.b_Input_PER_On	Config.InputPerOn
sPid_Cmpt.b_LoadBackUp	LoadBackUp
sPid_Cmpt.b_InvCtrl	Config.InvertControl
sPid_Cmpt.r_Lmn_Pwm_PPTm	Config.MinimumOnTime
sPid_Cmpt.r_Lmn_Pwm_PBTm	Config.MinimumOffTime
sPid_Cmpt.r_Pv_Hlm	Config.InputUpperLimit
sPid_Cmpt.r_Pv_Llm	Config.InputLowerLimit
sPid_Cmpt.r_Pv_HWrn	Config.InputUpperWarning
sPid_Cmpt.r_Pv_LWrn	Config.InputLowerWarning

PID_Compact V1.x	PID_Compact V2
sParamCalc.i_Event_SUT	PIDSelfTune.SUT.State
sParamCalc.i_Event_TIR	PIDSelfTune.TIR.State
sRet.i_Mode	sRet.i_Mode 已删除。 使用 Mode 和 ModeActivate 更改工作模式。
sRet.r_Ctrl_Gain	Retain.CtrlParams.Gain
sRet.r_Ctrl_Ti	Retain.CtrlParams.Ti
sRet.r_Ctrl_Td	Retain.CtrlParams.Td
sRet.r_Ctrl_A	Retain.CtrlParams.TdFiltRatio
sRet.r_Ctrl_B	Retain.CtrlParams.PWeighting
sRet.r_Ctrl_C	Retain.CtrlParams.DWeighting
sRet.r_Ctrl_Cycle	Retain.CtrlParams.Cycle

8.1.3.8 模式 V2 的参数状态

参数的相关性

State 参数显示了 PID 控制器的当前工作模式。 您无法更改 State 参数。

当 ModeActivate 出现上升沿时，PID_Compact 将切换到保存在 Mode 输入/输出参数中的工作模式。

CPU 启动或从 Stop 切换为 RUN 模式时，PID_Compact 将以保存在 Mode 参数中的工作模式启动。 要使 PID_Compact 保持在“未激活”模式下，应设置 RunModeByStartup = FALSE。

值的含义

State / Mode	工作模式说明
0	未激活 在“未激活”工作模式下，将始终输出输出值 0.0，无论 Config.OutputUpperLimit 以及 Config.OutputLowerLimit 如何。 脉宽调制关闭。
1	预调节 预调节功能可确定对输出值跳变的过程响应，并搜索拐点。 根据受控系统的最大上升速率与死时间计算 PID 参数。 可在执行预调节和精确调节时获得最佳 PID 参数。 预调节的要求： • 未激活 (State = 0)、手动模式 (State = 4) 或自动模式 (State = 3) • ManualEnable = FALSE • Reset = FALSE • 过程值不能过于接近设定值。 |Setpoint - Input| > 0.3 * | Config.InputUpperLimit - Config.InputLowerLimit| 和 |Setpoint - Input| > 0.5 * |Setpoint| • 设定值和过程值均在组态的限值范围内。 过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。 设定值在变量 CurrentSetpoint 中冻结。 出现以下情况时，调节将取消： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel 重新计算 PID 参数之前将对其进行备份并且可使用 LoadBackUp 重新激活这些参数。 预调节成功后，控制器将切换到自动模式。 如果预调节未成功，则工作模式的切换取决于 ActivateRecoverMode。 预调节阶段由 PIDSelfTune.SUT.State 来指示。

State / Mode	工作模式说明
2	精确调节 精确调节将使过程值出现恒定受限的振荡。 根据该振荡的幅度和频率重新计算 PID 参数。精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。 可在执行预调节和精确调节时获得最佳 PID 参数。 PID_Compact 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。 设定值在变量 CurrentSetpoint 中冻结。 出现以下情况时，调节将取消： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel 重新计算 PID 参数之前将对其进行备份并且可使用 LoadBackUp 重新激活这些参数。 精确调节的要求： • 不能被干扰。 • 设定值和过程值均在组态的限值范围内。 • ManualEnable = FALSE • Reset = FALSE • 自动模式 (State = 3)、未激活模式 (State = 0) 或手动模式 (State = 4) 在以下模式下启动精确调节时，具体情况如下所述： • 自动模式 (State = 3) 如果希望通过调节来改进现有 PID 参数，请在自动模式下启动精确调节。 PID_Compact 将使用现有的 PID 参数控制系统，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。 • 未激活模式 (State = 0) 或手动模式 (State = 4) 如果满足预调节的要求，则启动预调节。 已确定的 PID 参数将用于控制，直到控制回路已稳定并且精确调节的要求得到满足为止。 如果预调节的过程值已经十分接近设定值或 PIDSelfTune.TIR.RunIn = TRUE，则将尝试利用最小或最大输出值来达到设定值。 这可能会增加超调量。 之后才会启动精确调节。 精确调节成功后，控制器将切换到自动模式。 如果精确调节未成功，则工作模式的切换取决于 ActivateRecoverMode。 “精确调节”阶段由 PIDSelfTune.TIR.State 来指示。

State / Mode	工作模式说明
3	自动模式 在自动模式下，PID_Compact 会按照指定的参数来更正受控系统。 如果满足下列要求之一，则控制器将切换到自动模式： • 预调节成功完成 • 精确调节成功完成 • Mode 输入/输出参数更改为值 3 并且 ModeActivate 出现上升沿。 从自动模式到手动模式的切换只有在调试编辑器中执行时，才是无扰动的。 自动模式下会考虑 ActivateRecoverMode 变量。
4	手动模式 在手动模式下，在 ManualValue 参数中指定手动输出值。 还可以使用 ManualEnable = TRUE 来激活该工作模式。 建议只使用 Mode 和 ModeActivate 更改工作模式。 从手动模式到自动模式的切换是无扰动的。 错误未决时也可使用手动模式。
5	带错误监视的替代输出值 控制算法取消激活。 SetSubstituteOutput 变量确定此工作模式中输出哪个输出值。 • SetSubstituteOutput = FALSE：上一个有效输出值 • SetSubstituteOutput = TRUE：替代输出值 无法使用 Mode = 5 激活该工作模式。 如果满足以下所有条件，出现错误时会激活该工作模式而不激活“未激活”工作模式。 • 自动模式 (Mode = 3) • ActivateRecoverMode = TRUE • 已出现一个或多个错误，并且 ActivateRecoverMode 生效。 当错误不再处于未决状态时，PID_Compact 切换回自动模式。

ENO 特性

如果 State = 0，那么 ENO = FALSE。

如果 State ≠ 0，那么 ENO = TRUE。

在调试期间自动切换工作模式

预调节或精确调节成功后，将激活自动模式。 下表显示了成功预调节期间 Mode 和 State 的更改方式。

周期编号	Mode	State	操作
0	4	4	设置 Mode = 1
1	1	4	设置 ModeActivate = TRUE
1	4	1	State 的值保存在模式参数中 启动预调节功能
n	4	1	预调节成功完成
n	3	3	启动自动模式

PID_Compact 将在出现错误时自动切换工作模式。 下表显示了出现错误的预调节期间 Mode 和 State 的更改方式。

周期编号	Mode	State	操作
0	4	4	设置 Mode = 1
1	1	4	设置 ModeActivate = TRUE
1	4	1	State 的值保存在模式参数中 启动预调节功能
n	4	1	取消预调节
n	4	4	启动手动模式

如果 ActivateRecoverMode = TRUE，将激活保存在 Mode 参数中的工作模式。 开始预调节和精确调节时，PID_Compact 已将 State 的值保存在 Mode 输入/输出参数中。因此 PID_Compact 会切换到调节开始时工作模式。

如果 ActivateRecoverMode = FALSE，系统将切换到“未激活”工作模式。

参见

PID_Compact V2 的输出参数 (页 280)

8.1.3.9 参数 ErrorBits V2

如果多个错误同时处于待决状态，将通过二进制加法显示 ErrorBits 的值。例如，显示 ErrorBits = 0003h 表示错误 0001h 和 0002h 同时处于待决状态。

在手动模式下，PID_Compact 使用 ManualValue 作为输出值。Errorbits = 10000h 除外。

ErrorBits (DW#16#...)	说明
0000	没有任何错误。
0001	参数“Input”超出了过程值限值的范围。 • Input > Config.InputUpperLimit 或 • Input < Config.InputLowerLimit 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 保持自动模式。 如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。
0002	参数“Input_PER”的值无效。请检查模拟量输入是否有处于未决状态的错误。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 输出组态的替换输出值。当错误不再处于未决状态时，PID_Compact 切换回自动模式。 如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。
0004	精确调节期间出错。过程值无法保持振荡状态。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 取消调节并切换到 Mode 参数中保存的工作模式。
0008	预调节启动时出错。过程值过于接近设定值。启动精确调节。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 取消调节并切换到 Mode 参数中保存的工作模式。
0010	调节期间设定值发生更改。 可在 CancelTuningLevel 变量中设置允许的设定值波动。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 取消调节并切换到 Mode 参数中保存的工作模式。
0020	精确调节期间不允许预调节。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 保持在精确调节模式。

ErrorBits (DW#16#...)	说明
0080	预调节期间出错。输出值限值的组态不正确。 检查输出值的限值是否已正确组态及其是否匹配控制逻辑。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 取消调节并切换到 Mode 参数中保存的工作模式。
0100	精确调节期间的错误导致生成无效参数。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 取消调节并切换到 Mode 参数中保存的工作模式。
0200	参数“Input”的值无效：值的数字格式无效。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 输出组态的替换输出值。当错误不再处于未决状态时，PID_Compact 切换回自动模式。 如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。
0400	输出值计算失败。请检查 PID 参数。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 输出组态的替换输出值。当错误不再处于未决状态时，PID_Compact 切换回自动模式。 如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。
0800	采样时间错误：在循环中断 OB 的采样时间内没有调用 PID_Compact。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 保持自动模式。 如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。 如果在使用 PLCSIM 进行仿真期间出现该错误，请参见使用 PLCSIM 仿真 PID_Compact V2 (页 103)下的说明。
1000	参数“Setpoint”的值无效：值的数字格式无效。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 输出组态的替换输出值。当错误不再处于未决状态时，PID_Compact 切换回自动模式。 如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。

ErrorBits (DW#16#...)	说明
10000	ManualValue 参数的值无效。值的数字格式无效。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 会将 SubstituteOutput 用作输出值。在 ManualValue 中指定有效值后，PID_Compact 便会将其作为输出值。
20000	变量 SubstituteOutput 的值无效。值的数字格式无效。 PID_Compact 使用输出值下限作为输出值。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_Compact 切换回自动模式。
40000	Disturbance 参数的值无效。值的数字格式无效。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 Disturbance 将设置为零。PID_Compact 保持自动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。如果当前阶段中的 Disturbance 对输出值无影响，则不会取消调节。

8.1.3.10 变量 ActivateRecoverMode V2

ActivateRecoverMode 变量确定错误响应方式。 Error 参数指示是否存在错误处于未决状态。 当错误不再处于未决状态时，Error = FALSE。 ErrorBits 参数显示发生的具体错误。

自动模式

注意
您的系统可能已损坏。 如果 ActivateRecoverMode = TRUE，则 PID_Compact 保持自动模式，即使出现错误或超过过程限值。 这可能损坏您的系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

ActivateRecoverMode	说明
FALSE	PID_Compact 将在出现错误时自动切换到“未激活”模式。 只能通过 Reset 的下降沿或 ModeActivate 的上升沿激活控制器。
TRUE	如果在自动模式下频繁出现错误，则该设置会对控制响应产生负面影响，这是由于发生每个错误时，PID_Compact 在计算的输出值和替代输出值之间切换导致。 这种情况下，检查 ErrorBits 参数并消除错误原因。 如果发生一个或多个下列错误，则 PID_Compact 停留在自动模式下： • 0001h： 参数“Input”超出了过程值限值的范围。 • 0800h： 采样时间错误 • 40000h：参数 Disturbance 的值无效。 如果发生一个或多个下列错误，PID_Compact 切换到“带错误监视的替代输出值”模式： • 0002h： Input_PER 参数的值无效。 • 0200h： Input 参数的值无效。 • 0400h： 输出值计算失败。 • 1000h： Setpoint 参数的值无效。 如果发生下列错误，PID_Compact 将切换到“带错误监视的替代输出值”模式，并将执行器移至 Config.OutputLowerLimit： • 20000h： 变量 SubstituteOutput 的值无效。 值的数字格式无效。 此特性与 SetSubstituteOutput 无关。 当错误不再处于未决状态时，PID_Compact 切换回自动模式。

预调节和精确调节

ActivateRecoverMode	说明
FALSE	PID_Compact 将在出现错误时自动切换到“未激活”模式。 只能通过 Reset 的下降沿或 ModeActivate 的上升沿激活控制器。
TRUE	如果发生下列错误，PID_Compact 将保持在激活模式： • 0020h： 精确调节期间不允许预调节。 以下错误将被忽略： • 10000h： ManualValue 参数的值无效。 • 20000h：变量 SubstituteOutput 的值无效。 出现其它错误时，PID_Compact 将取消调节并切换到调节开始时的模式。

手动模式

手动模式下 ActivateRecoverMode 无效。

8.1.3.11 变量 Warning V2

如果多个警告同时处于待决状态，将通过二进制加法显示 Warning 变量值。 例如，显示警告 0003h 表示警告 0001h 和 0002h 同时处于待决状态。

Warning (DW#16#....)	说明
0000	无警告处于待决状态。
0001	预调节期间未发现拐点。
0004	设定值被限制为组态的限值。
0008	在所选计算方法中未定义所有必要的受控系统属性。 而是使用 TIR.TuneRule = 3 方法计算 PID 参数。
0010	由于 Reset = TRUE 或 ManualEnable = TRUE，无法更改工作模式。
0020	调用 OB 的循环时间会限制 PID 算法的采样时间。 通过缩短 OB 循环时间来改进结果。
0040	过程值超出其警告限值之一。
0080	Mode 的值无效。 工作模式不变。
0100	手动值被限制为控制器输出的限值。
0200	不支持指定的调节规则。 不计算任何 PID 参数。
1000	无法达到替代输出值，因为它超出了输出值限值。

以下警告在消除问题的原因后即被删除：

- 0001h
- 0004h
- 0008h
- 0040h
- 0100h

所有其它警告均在 Reset 或 ErrorAck 出现上升沿时清除。

8.1.3.12 IntegralResetMode V2 变量

IntegralResetMode 变量用于确定如何预分配积分作用 PIDCtrl.IntegralSum：

- 从“未激活”工作模式切换到“自动模式”时
- 参数 Reset 出现 TRUE -> FALSE 沿并且参数 Mode = 3 时

只有在激活了积分作用时，该设置才会在一个周期内有效（Retain.CtrlParams.Ti > 0.0 变量）。

IntegralResetMode	说明
0	平滑 已经预分配了 PIDCtrl.IntegralSum 的值，因此可以实现无扰动切换，即通过输出值 = 0.0（参数 Output）启动“自动模式”，并且无论是否存在控制偏差（设定值 – 实际值），输出值都不会发生跳变。
1	删除 如果使用该选项，我们建议将比例作用的权重 (Retain.CtrlParams.PWeighting) 设为 1.0。 PIDCtrl.IntegralSum 的值已删除。任何控制偏差都将导致输出值跳变。输出值的跳变方向取决于组态的比例作用权重（Retain.CtrlParams.PWeighting 变量）以及控制偏差： • 比例作用权重 = 1.0： 输出值跳变与控制偏差的符号相同。 示例：如果实际值小于设定值（正控制偏差），则输出值会跳变至正值。 • 比例作用权重 < 1.0： 对于较大的控制偏差，输出值跳变与控制偏差的符号相同。 示例：如果实际值远远小于设定值（正控制偏差），则输出值会跳变至正值。 对于较小的控制偏差，输出值跳变与控制偏差的符号不同。 示例：如果实际值略小于设定值（正控制偏差），则输出值会跳变至负值。通常不希望出现这种情况，因为这会导致控制偏差暂时增大。 组态的比例作用权重越小，控制偏差就越大，以便接收具有相同符号的输出值跳变。 如果使用该选项，我们建议将比例作用的权重 (Retain.CtrlParams.PWeighting) 设为 1.0。否则，可能会出现针对小控制偏差所说明的不良行为。您还可以使用 IntegralResetMode = 4。该选项确保输出值跳变与控制偏差的符号相同，无论组态的比例作用权重和控制偏差为何值。
2	保持 PIDCtrl.IntegralSum 的值未更改。您可以使用用户程序定义一个新值。

IntegralResetMode	说明
3	预分配 自动预分配 PIDCtrl.IntegralSum 的值，如同在上一周期中 Output = OverwriteInitialOutputValue。
4	类似于设定值更改（仅适用于版本 2.3 及更高版本的 PID_Compact） 自动预分配 PIDCtrl.IntegralSum 的值，以便使输出值跳变与自动模式下设定值从当前实际值更改为当前设定值时的 PI 控制器的行为类似。 任何控制偏差都会导致输出值跳变。输出值跳变与控制偏差的符号相同。 示例：如果实际值小于设定值（正控制偏差），则输出值会跳变至正值。这与组态的比例作用权重和控制偏差无关。

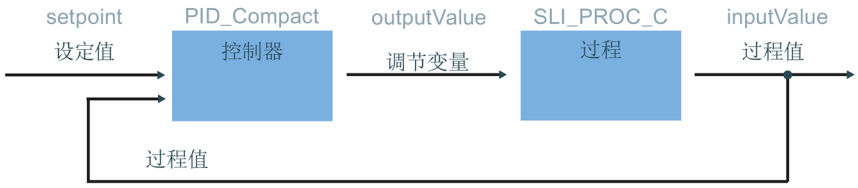
如果为 IntegralResetMode 分配的值不在有效值范围内，PID_Compact 的行为将与 IntegralResetMode 预分配时的情况相同：

- PID_Compact V2.2 及之前的版本：IntegralResetMode = 1
- PID_Compact V2.3 和更高版本：IntegralResetMode = 4

上述与输出值跳变的符号相关的所有说明均基于正常控制逻辑（Config.InvertControl = FALSE 变量）。对于反转的控制逻辑 (Config.InvertControl = TRUE)，输出值跳变的符号将相反。

8.1.3.13 PID_Compact 的示例程序

在以下示例中，通过指令“PID_Compact”的工艺对象来控制温度值。基于一个用于仿真三阶延时元件（PT3 元件）的块对温度值进行仿真。工艺对象的 PID 参数可通过预调节自动设置。



数据存储

在全局数据块中创建 7 个变量，以便存储互连数据。

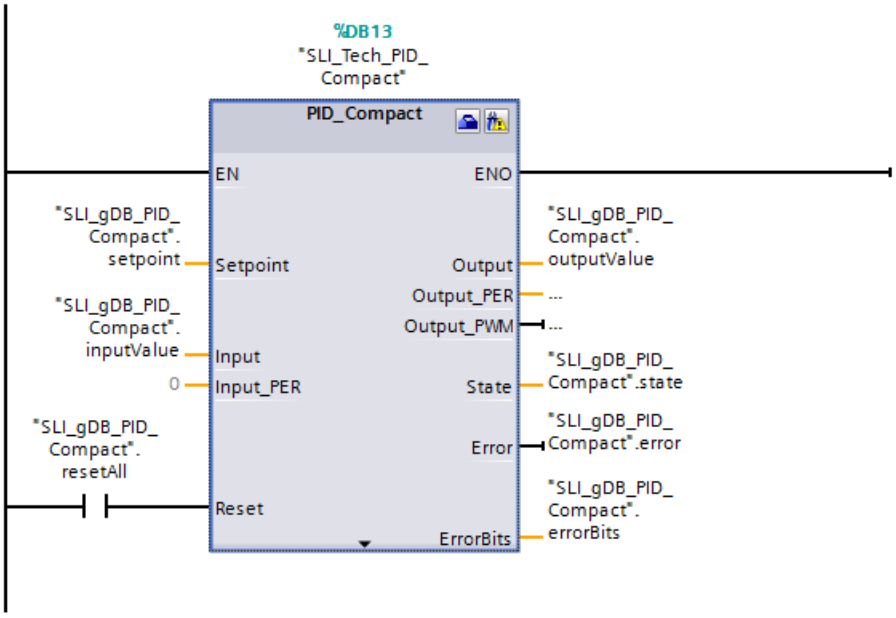
SLI_gDB_PID_Compact

	Name	Data type	Start value
1	Static		
2	setpoint	Real	75.0
3	inputValue	Real	0.0
4	outputValue	Real	0.0
5	state	Int	0
6	error	Bool	false
7	errorBits	DWord	16#0
8	resetAll	Bool	false

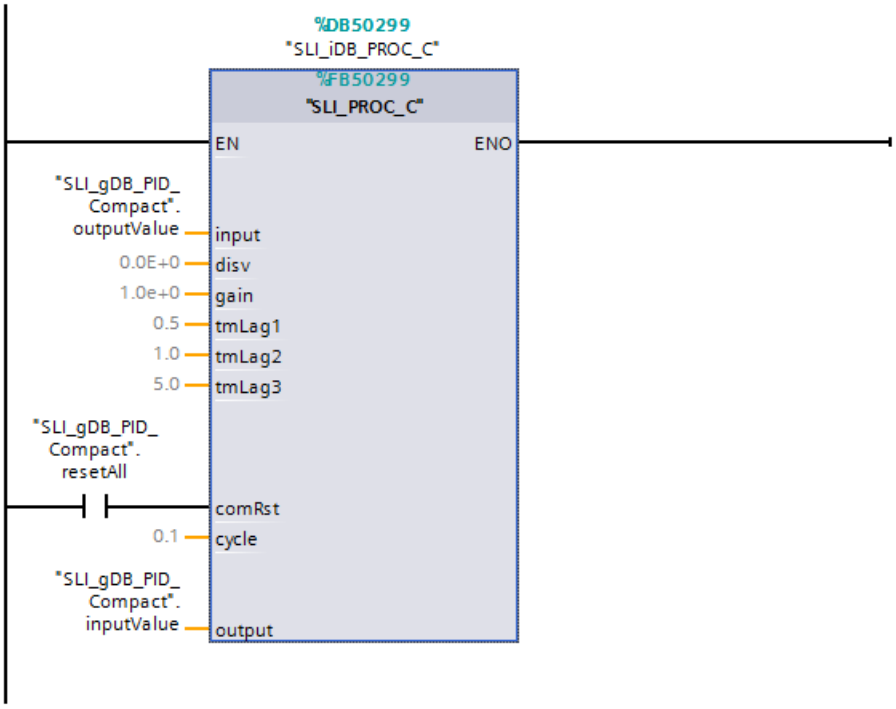
参数的互连

在循环中断 OB 中调用以下互连。

网络 1：按照如下所示，对指令“PID_Compact”的参数进行互连。

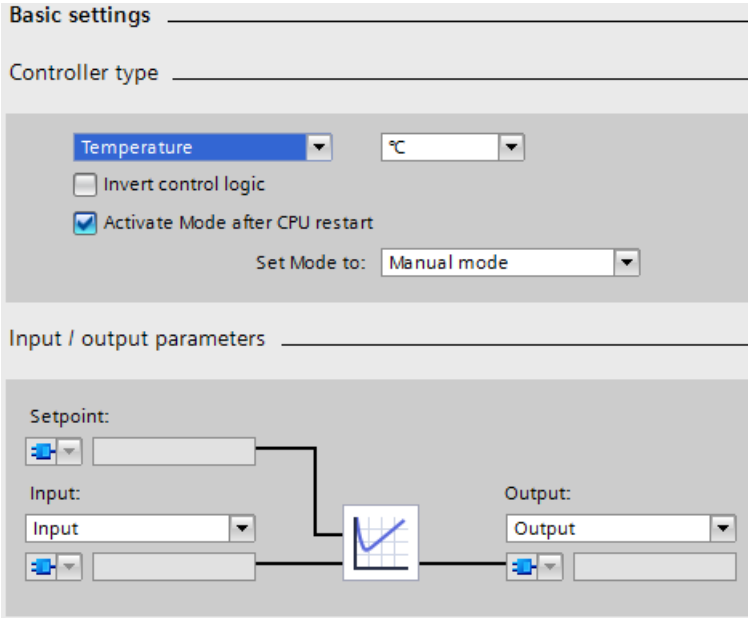


程序段 2：按照如下所示，对用于仿真温度值“SLI_PROC_C”的块参数进行互连。



工艺对象

通过指令“PID_Compact”的属性或使用路径“工艺对象 > 组态”(Technology object > Configuration) 对工艺对象进行组态。对于本示例，控制器类型和输入/输出参数至关重要。借助控制器类型，您可以预先选择要控制值的单位。在本示例中，将单位为“°C”的“温度”(Temperature) 用作控制器类型。“PID_Compact”的参数已经与全局变量互连。因此，有关参数 Input 和 Output 的使用信息十分充分。



启动控制的步骤

下载到 CPU 后，PID_Compact 处于手动模式，手动值为 0.0。要启动控制，请按照以下步骤进行操作：

1. 打开工艺对象“SLI_Tech_PID_Compact”的“调试”(Commissioning) 对话框。
2. 单击“测量”(Measurement) 区域中的“启动”(Start) 按钮。

测量过程将启动并且可激活 PID_Compact。

3. 选择预调节。

   单击“调节模式”(Tuning mode) 区域中的“启动”(Start) 按钮。

   执行了预调节。PID 参数根据过程自动调整。完成预调节后，PID_Compact 切换到自动模式。

说明

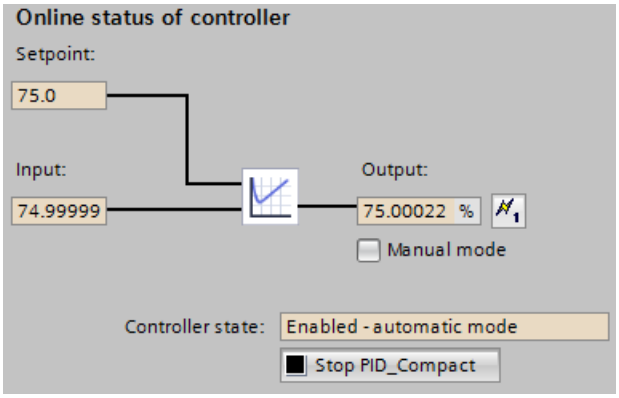
选择启动 PID_Compact

还可以不进行预调节，在“控制器的在线状态”(Online status of controller) 区域中通过“停止 PID_Compact”(Stop PID_Compact)/“启动 PID_Compact”(Start PID_Compact) 来将 PID_Compact 切换到自动模式。这种情况下，控制器使用 PID 参数的默认值并显示了该应用情况下不良的控制器表现。

停止控制的步骤

要停止并退出 PID_Compact 和程序，请按照以下步骤进行操作：

1. 单击工艺对象“SLI_Tech_PID_Compact”中“控制器的在线状态”(Online status of controller) 区域中的“Stop PID_Compact”按钮。



   指令“PID_Compact”将退出控制并将值“0.0”作为调节变量输出。

2. 单击“测量”(Measurement) 区域中的“停止”(Stop) 按钮。
3. 要将过程值立即设为“0.0”，请按下列步骤操作：

   在“SLI_OB_PID_Compact”块中，将“resetAll”变量设为值“TRUE”，然后再设为值“FALSE”。

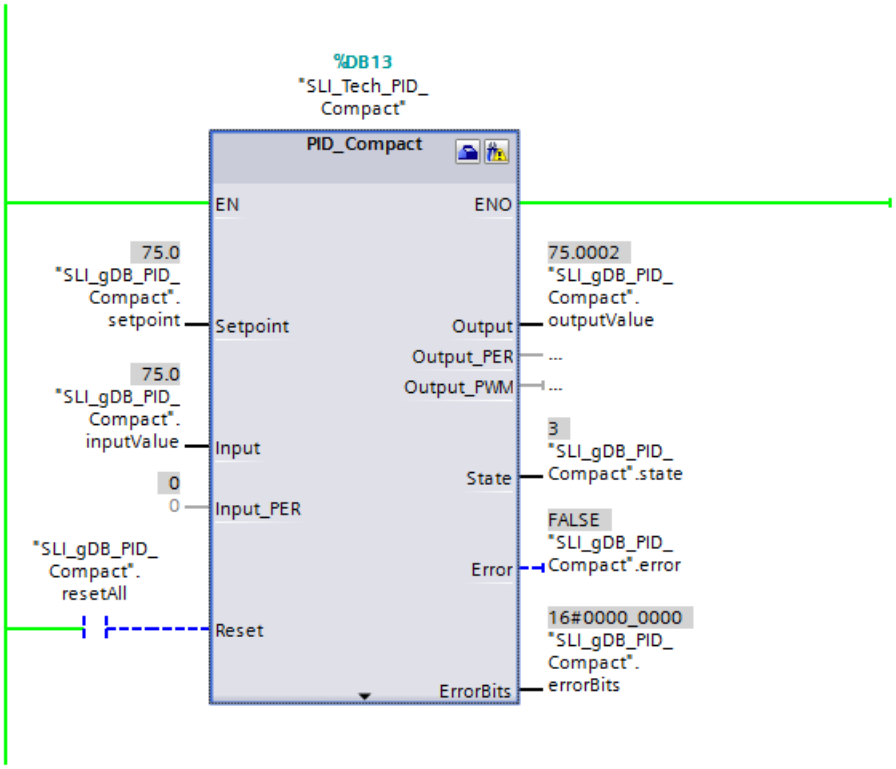
“PID_Compact”指令

在参数 Setpoint（“setpoint”）处指定要控制的温度的设定值。通过工艺对象启动指令“PID_Compact”时启动控制。指令“PID_Compact”将在输出参数 Output（“outputValue”）处输出一个调节变量。通过输入参数 Input（“inputValue”）将温度的过程值传送给指令“PID_Compact”。

指令“PID_Compact”可根据设定值（“setpoint”）和过程值（“inputValue”）之间的历史偏差调整操作变量（“outputValue”）。重复进行该过程，通过操作变量（“outputValue”）使过程值（“inputValue”）接近设定值（“setpoint”）。

输出参数 State（“state”）处显示了指令“PID_Compact”的当前工作模式。在完成预调节后（“state”的值为“1”），PID_Compact 将切换至自动模式（值为“3”）。

当前的输出参数 Error（“error”）显示无错误待决。发生错误时，输出参数 ErrorBits（“errorBits”）提供关于错误类型的信息。如果发生错误，可在工艺对象中的优化状态区域通过“ErrorAck”按钮确认。

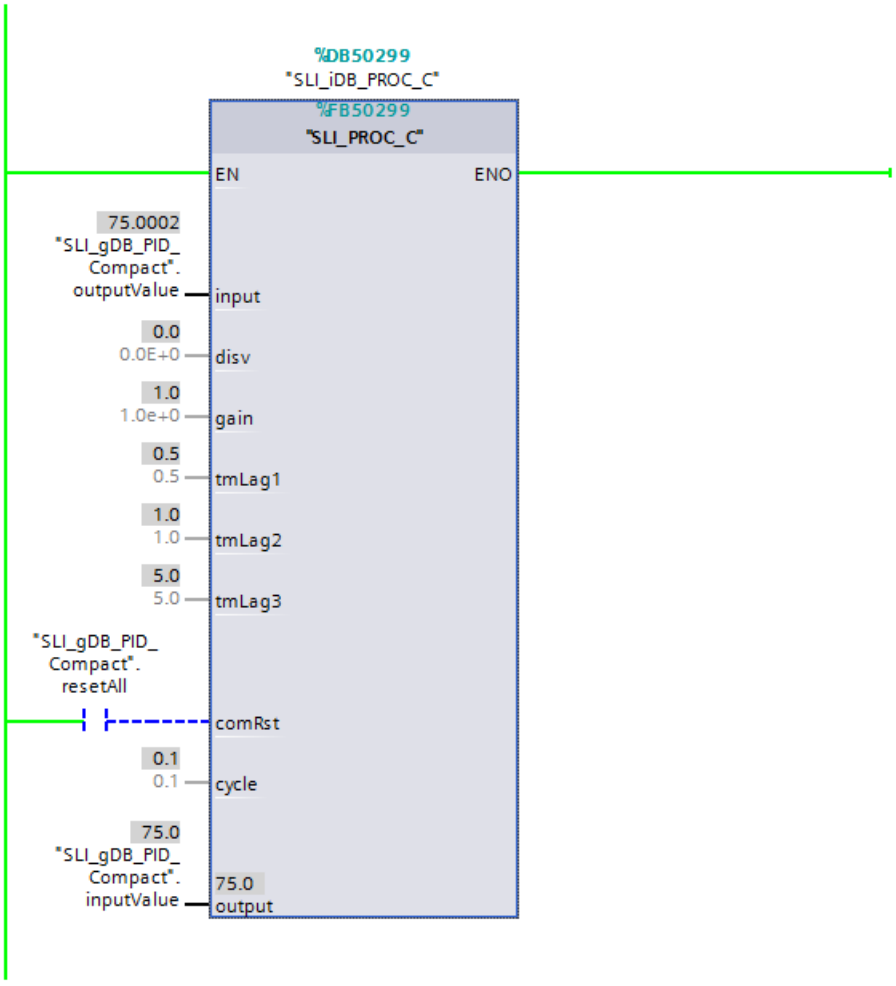


“SLI_PROC_C”块

“SLI_PROC_C”块用于对不断上升的设备温度的过程值（“inputValue”）进行仿真。“SLI_PROC_C”块包含控制器的操作变量（“outputValue”），可对过程的温度特性进行仿真。该温度作为过程值（“inputValue”）反馈到控制器中。

“resetAll”变量（comRst 参数）值的变化 () 会产生以下影响：

参数 comRst（“resetAll”）	指令“PID_Compact”正在运行	指令“PID_Compact”已停止运行
comRst（“resetAll”）仍设置为值“FALSE”	“SLI_PROC_C”块基于操作变量（“outputValue”）输出一个新的过程值（“inputValue”）。	“SLI_PROC_C”块不会接收大于“0.0”的操作变量，但仍会输出一个新的过程值 >“0.0”。
comRst（“resetAll”）：从“FALSE”变为值“TRUE”	操作变量（“outputValue”）和输出过程值（“inputValue”）均复位为“0.0”。	“SLI_PROC_C”块的过程值（“inputValue”）/温度复位为“0.0”。
comRst（“resetAll”）：从“TRUE”变为值“FALSE”	再次启动温度控制。	输出过程值/温度（“inputValue”）保持“0.0”。



程序代码

有关上述示例中程序代码的更多信息，请搜索关键词“指令示例库”。

8.1.4 PID_Compact V2.x 的 CPU 处理时间和存储器要求

CPU 处理时间

自版本 V2.0 起的 PID_Compact 工艺对象典型 CPU 处理时间（取决于 CPU 类型）。

CPU	典型 CPU 处理时间 (PID_Compact V2.x)
CPU 1211C ≥ V4.0	300 µs
CPU 1215C ≥ V4.0	300 µs
CPU 1217C ≥ V4.0	300 µs
CPU 1505S ≥ V1.0	45 µs
CPU 1510SP-1 PN ≥ V1.6	85 µs
CPU 1511-1 PN ≥ V1.5	85 µs
CPU 1512SP-1 PN ≥ V1.6	85 µs
CPU 1516-3 PN/DP ≥ V1.5	50 µs
CPU 1518-4 PN/DP ≥ V1.5	4 µs

存储器要求

自版本 V2.0 起的 PID_Compact 工艺对象背景数据块的存储器要求。

	PID_Compact V2.x 背景数据块的 存储器要求
装载存储器要求	约 12000 个字节
总工作存储器要求	788 个字节
保持性工作存储器要求	44 个字节

8.1.5 PID_Compact V1

8.1.5.1 PID_Compact V1 说明

说明

PID_Compact 指令提供了一种可在自动和手动模式下进行调节的 PID 控制器。

调用

以调用 OB 的循环时间的恒定间隔（最好在循环中断 OB 中）调用 PID_Compact。

下载到设备

仅当完全下载 PID_Compact 后，才能更新保持性变量的实际值。

将工艺对象下载到设备 (页 76)

启动

CPU 启动时，PID_Compact 以上次激活的操作模式启动。 要将 PID_ Compact 保留在“未激活”模式下，应设置 sb_RunModeByStartup = FALSE。

PID_Compact 采样时间的监视

理想情况下，采样时间等于调用 OB 的循环时间。PID_Compact 指令可测量两次调用之间的时间间隔。 这就是当前采样时间。 每次切换工作模式以及初始启动期间，平均值由前 10 个采样时间构成。 如果当前采样时间严重偏离该平均值，则将出现 Error = 0800 hex，并且 PID_Compact 将切换到“未激活”模式。

在下列条件下，版本 1.1 或更高版本的 PID_Compact 在控制器调节期间将设置为“未激活”模式：

- 新平均值 >= 1.1 x 原平均值
- 新平均值 <= 0.9 x 原平均值

在自动模式下，版本 1.1 或更高版本的 PID_Compact 在下列条件下将设置为“未激活”模式：

- 新平均值 >= 1.5 x 原平均值
- 新平均值 <= 0.5 x 原平均值

在控制器调节期间和自动模式下，PID_Compact 1.0 在下列条件下将设置为“未激活”操作模式：

- 新平均值 >= 1.1 x 原平均值
- 新平均值 <= 0.9 x 原平均值
- 当前采样时间 >= 1.5 x 当前平均值
- 当前采样时间 <= 0.5 x 当前平均值

PID 算法的采样时间

受控系统需要一定的时间来对输出值的变化做出响应。 因此，建议不要在每次循环中都计算输出值。 PID 算法的采样时间是两次计算输出值之间的时间。 该时间在调节期间进行计算，并舍入为循环时间的倍数。 PID_Compact 的所有其它功能会在每次调用时执行。

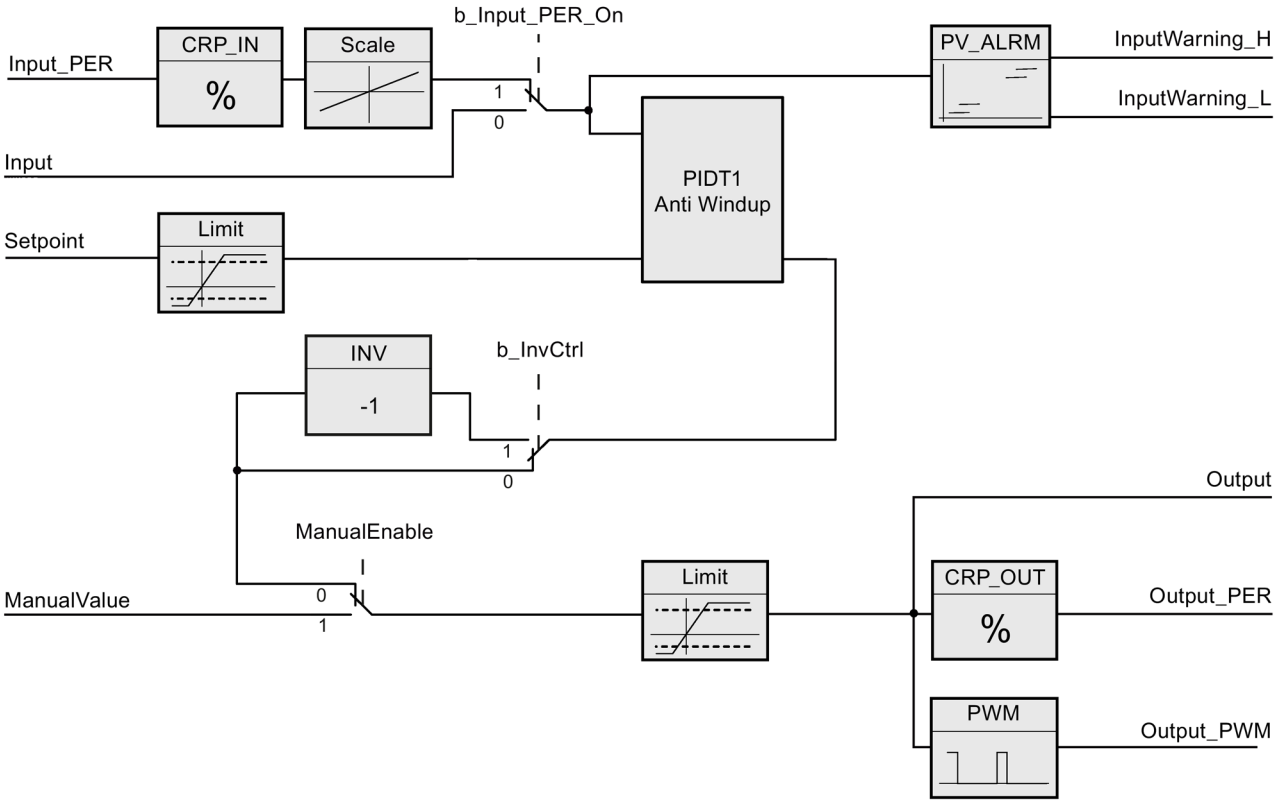
PID 算法

PID_Compact 是一种具有抗积分饱和功能并且能够对比例作用和微分作用进行加权的 PIDT1 控制器。 采用以下方程来计算输出值。

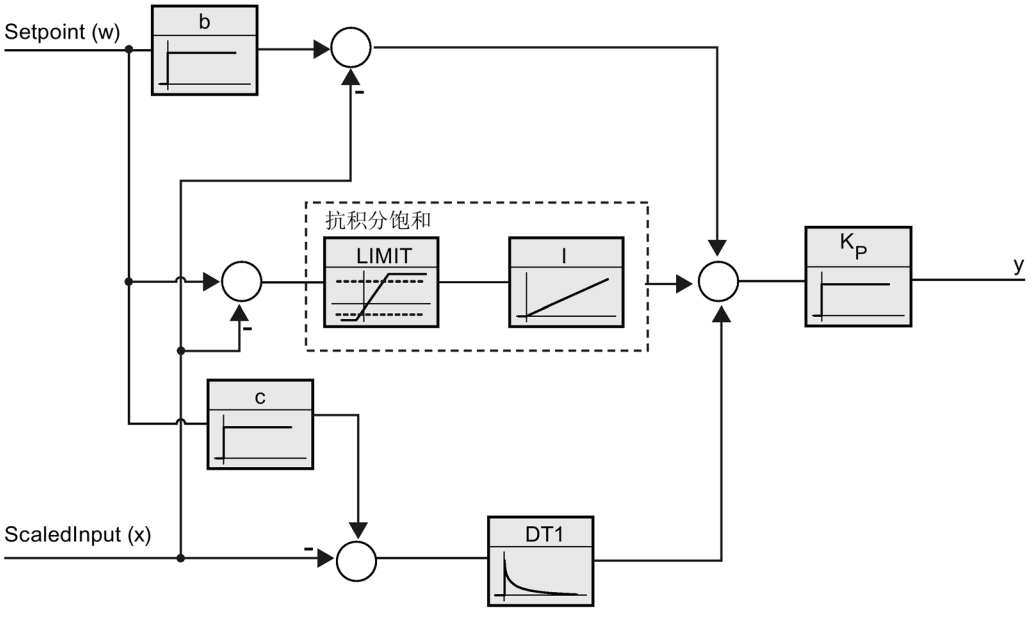
$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
y	输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分作用时间
a	微分延迟系数 (T1 = a × T_D)
	微分作用时间
c	微分作用权重

PID_Compact 方框图



带抗积分饱和的 PIDT1 的方框图



对错误的响应

如果出现错误，会在参数 Error 中输出，且 PID_Compact 会切换到“未激活”模式。 使用 Reset 参数复位错误。

控制逻辑

通常，可通过增大输出值来增大过程值。 这种做法称为常规控制逻辑。对于制冷和放电控制系统，可能需要反转控制逻辑。PID_Compact 不使用负比例增益。 如果 InvertControl = TRUE，则不断增大的控制偏差将导致输出值减小。 在预调节和精确调节期间还会考虑控制逻辑。

参见

控制模式 V1 (页 105)

8.1.5.2 PID_Compact V1 的输入参数

表格 8- 4

参数	数据类型	默认值	说明
Setpoint	REAL	0.0	PID 控制器在自动模式下的设定值
Input	REAL	0.0	用户程序的变量用作过程值的源。 如果正在使用参数 Input，则必须设置 sPid_Cmpt.b_Input_PER_On = FALSE。
Input_PER	WORD	W#16#0	模拟量输入用作过程值的源 如果正在使用参数 Input_PER，则必须设置 sPid_Cmpt.b_Input_PER_On = TRUE。
ManualEnable	BOOL	FALSE	• 出现 FALSE -> TRUE 沿时会选择“手动模式”，而 State = 4，sRet.i_Mode 保持不变。 • 出现 TRUE -> FALSE 沿时会选择最近激活的操作模式，State =sRet.i_Mode sRet.i_Mode 的变化在 ManualEnable = TRUE 期间不会生效。 仅在 ManualEnable 处出现 TRUE -> FALSE 沿时 sRet.i_Mode 的变化才会生效。 **PID_Compact V1.2 und PID_Compact V1.0** 如果 CPU 启动时 ManualEnable = TRUE，PID_Compact 将在手动模式下启动。 并非一定需要 ManualEnable 出现上升沿 (FALSE -> TRUE) 时，才会执行上述操作。 **PID_Compact V1.1** CPU 启动时，PID_Compact 仅在 ManualEnable 出现上升沿 (FALSE->TRUE) 时才切换到手动模式。 没有上升沿时，PID_Compact 在 ManualEnable 为 FALSE 的上一个工作模式下启动。
ManualValue	REAL	0.0	手动值 该值用作手动模式下的输出值。
Reset	BOOL	FALSE	Reset 参数 (页 336)可重启控制器。

8.1.5.3 PID_Compact V1 的输出参数

表格 8- 5

Parameter	数据类型	默认值	说明
ScaledInput	REAL	0.0	标定的过程值的输出
可同时使用输出“Output”、“Output_PER”和“Output_PWM”。			
Output	REAL	0.0	REAL 形式的输出值
Output_PER	WORD	W#16#0	模拟量输出值
Output_PWM	BOOL	FALSE	脉宽调制输出值 输出值由最短开关时间形成。
SetpointLimit_H	BOOL	FALSE	如果 SetpointLimit_H = TRUE，则说明达到了设定值的绝对上限。 在 CPU 中，该设定值被限制为所组态的设定值的绝对上限。 设定值的上限默认设置为所组态的过程值的绝对上限。 如果设置的 sPid_Cmpt.r_Sp_Hlm 值位于过程值的限值范围内，则该值将用作设定值的上限。
SetpointLimit_L	BOOL	FALSE	如果 SetpointLimit_L = TRUE，则说明已达到设定值的绝对下限。 在 CPU 中，该设定值被限制为所组态的设定值的绝对下限。 设定值的下限将默认设置为所组态的过程值的绝对下限。 如果设置的 sPid_Cmpt.r_Sp_Llm 值位于过程值的限值范围内，则该值将用作设定值的下限。
InputWarning_H	BOOL	FALSE	如果 InputWarning_H = TRUE，则说明过程值已达到或超出警告上限。
InputWarning_L	BOOL	FALSE	如果 InputWarning_L = TRUE，则说明过程值已经达到或低于警告下限。
State	INT	0	State 参数 (页 331)显示 PID 控制器的当前操作模式。 要更改操作模式，请使用变量 sRet.i_Mode。 • State = 0：未激活 • State = 1：预调节 • State = 2：精确调节 • State = 3：自动模式 • State = 4：手动模式
Error	DWORD	W#16#0	Error 参数 (页 335)指示错误消息。 Error = 0000：没有待决的错误。

8.1.5.4 PID_Compact V1 的静态变量

不得更改未列出的变量。这些变量仅供内部使用。

表格 8- 6

变量	数据类型	默认值	说明
sb_GetCycleTime	BOOL	TRUE	如果 sb_GetCycleTime = TRUE，则开始自动确定循环时间。一旦测量完成，CycleTime.StartEstimation = FALSE。
sb_EnCyclEstimation	BOOL	TRUE	如果 sb_EnCyclEstimation = TRUE，则不会监视 PID_Compact 采样时间。
sb_EnCyclMonitoring	BOOL	TRUE	如果 sb_EnCyclMonitoring = FALSE，则不会监视 PID_Compact 采样时间。如果不能在采样时间内执行 PID_Compact，则不会输出 0800 错误，PID_Compact 也不会切换到“未激活”模式。
sb_RunModeByStartup	BOOL	TRUE	在 CPU 重启后激活模式 如果 sb_RunModeByStartup = FALSE，则控制器在 CPU 启动后扔保持未激活状态。 CPU 启动后，如果 sb_RunModeByStartup = TRUE，控制器将返回到最近激活的操作模式。
si_Unit	INT	0	过程值和设定值的测量单位，例如 ºC 或 ºF。
si_Type	INT	0	过程值和设定值的物理量，如温度。
sd_Warning	DWORD	DW#16#0	变量 sd_warning (页 338) 显示自复位或上一次更改操作模式以来所生成的警告。
sBackUp.r_Gain	REAL	1.0	保存的比例增益 sPid_Cmpt.b_LoadBackUp = TRUE 时，可以从 sBackUp 结构中重新加载值。
sBackUp.r_Ti	REAL	20.0	保存的积分作时间 [s]
sBackUp.r_Td	REAL	0.0	保存的微分作时间 [s]
sBackUp.r_A	REAL	0.0	保存的微分延时系数
sBackUp.r_B	REAL	0.0	保存的比例作用权重因子

变量	数据类型	默认值	说明
sBackUp.r_C	REAL	0.0	保存的微分作用权重因子
sBackUp.r_Cycle	REAL	1.0	保存的 PID 算法的采样时间
sPid_Calc.r_Cycle	REAL	0.1	PID_Compact 指令采样时间 r_Cycle 会自动确定，通常等于调用 OB 的周期时间。
sPid_Calc.b_RunIn	BOOL	FALSE	• b_RunIn = FALSE 在未激活模式或手动模式下启动精确调节时，将启动预调节。如果不满足预调节的要求，则 PID_Compact 的响应将类似于 b_RunIn = TRUE 时的响应。 如果精确调节在自动模式下启动，系统将使用现有的 PID 参数来控制设定值。 之后才会启动精确调节。如果无法实现预调节，PID_Compact 将切换到“未激活”模式。 • b_RunIn = TRUE 将跳过预调节。PID_3Compact 将尝试利用最小或最大输出值来达到设定值。这可能会增加超调量。随后将自动启动精确调节。 精确调节后，b_RunIn 将被设置为 FALSE。
sPid_Calc.b_CalcParamSUT	BOOL	FALSE	如果 b_CalcParamSUT = TRUE，将重新计算用于预调节的参数。这样无需重复进行控制器调节，就可以更改参数计算方法。 计算后，b_CalcParamSUT 将设置为 FALSE。
sPid_Calc.b_CalcParamTIR	BOOL	FALSE	如果 b_CalcParamTIR = TRUE，将重新计算用于精确调节的参数。这样无需重复进行控制器调节，就可以更改参数计算方法。 计算后，b_CalcParamTIR 将被设置为 FALSE。

变量	数据类型	默认值	说明
sPid_Calc.i_CtrlTypeSUT	INT	0	预调节期间用于计算参数的方法： • i_CtrlTypeSUT = 0：根据 Chien、Hrones 和 Reswick 计算 PID • i_CtrlTypeSUT = 1：根据 Chien、Hrones 和 Reswick 计算 PI
sPid_Calc.i_CtrlTypeTIR	INT	0	精确调节期间用于计算参数的方法： • i_CtrlTypeTIR = 0：PID 自动 • i_CtrlTypeTIR = 1：PID 快速 • i_CtrlTypeTIR = 2：PID 慢速 • i_CtrlTypeTIR = 3：Ziegler-Nichols PID • i_CtrlTypeTIR = 4：Ziegler-Nichols PI • i_CtrlTypeTIR = 5：Ziegler-Nichols P
sPid_Calc.r_Progress	REAL	0.0	百分数形式的调节进度 (0.0 - 100.0)
sPid_Cmpt.r_Sp_Hlm	REAL	+3.402822e+38	设定值的上限 如果组态的 sPid_Cmpt.r_Sp_Hlm 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作设定值的上限。 如果设置的 sPid_Cmpt.r_Sp_Hlm 值位于过程值的限值范围内，则该值将用作设定值的上限。
sPid_Cmpt.r_Sp_Llm	REAL	-3.402822e+38	设定值的下限 如果设置的 sPid_Cmpt.r_Sp_Llm 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作设定值的下限。 如果设置的 sPid_Cmpt.r_Sp_Llm 值位于过程值的限值范围内，则该值将用作设定值的下限。

变量	数据类型	默认值	说明
sPid_Cmpt.r_Pv_Norm_IN_1	REAL	0.0	标定的 Input_PER 下限 根据 sPid_Cmpt 结构中的以下两个值对将 Input_PER 转换为百分数：r_Pv_Norm_OUT_1、r_Pv_Norm_IN_1 以及 r_Pv_Norm_OUT_2、r_Pv_Norm_IN_2。
sPid_Cmpt.r_Pv_Norm_IN_2	REAL	27648.0	标定的 Input_PER 上限 根据 sPid_Cmpt 结构中的以下两个值对将 Input_PER 转换为百分数：r_Pv_Norm_OUT_1、r_Pv_Norm_IN_1 以及 r_Pv_Norm_OUT_2、r_Pv_Norm_IN_2。
sPid_Cmpt.r_Pv_Norm_OUT_1	REAL	0.0	标定的过程值的下限 根据 sPid_Cmpt 结构中的以下两个值对将 Input_PER 转换为百分数：r_Pv_Norm_OUT_1、r_Pv_Norm_IN_1 以及 r_Pv_Norm_OUT_2、r_Pv_Norm_IN_2。
sPid_Cmpt.r_Pv_Norm_OUT_2	REAL	100.0	标定的过程值的上限 根据 sPid_Cmpt 结构中的以下两个值对将 Input_PER 转换为百分数：r_Pv_Norm_OUT_1、r_Pv_Norm_IN_1 以及 r_Pv_Norm_OUT_2、r_Pv_Norm_IN_2。
sPid_Cmpt.r_Lmn_Hlm	REAL	100.0	输出参数“Output”的输出值上限
sPid_Cmpt.r_Lmn_Llm	REAL	0.0	输出参数“Output”的输出值下限
sPid_Cmpt.b_Input_PER_On	BOOL	TRUE	如果 b_Input_PER_On = TRUE，则使用参数 Input_PER。如果 b_Input_PER_On = FALSE，则使用参数 Input。
sPid_Cmpt.b_LoadBackUp	BOOL	FALSE	激活备份参数集。如果优化失败，可通过置位该位重新激活先前的 PID 参数。

变量	数据类型	默认值	说明
sPid_Cmpt.b_InvCtrl	BOOL	FALSE	反转控制逻辑 如果 b_InvCtrl = TRUE，则不断增大的控制偏差将导致输出值减小。
sPid_Cmpt.r_Lmn_Pwm_PPTm	REAL	0.0	脉宽调制的最小 ON 时间（秒）舍入为 r_Lmn_Pwm_PPTm = r_Cycle 或 r_Lmn_Pwm_PPTm = n*r_Cycle
sPid_Cmpt.r_Lmn_Pwm_PBTm	REAL	0.0	脉宽调制的最小 OFF 时间（秒）舍入为 r_Lmn_Pwm_PBTm = r_Cycle 或 r_Lmn_Pwm_PBTm = n*r_Cycle
sPid_Cmpt.r_Pv_Hlm	REAL	120.0	过程值的上限 在 I/O 输入中，过程值最大可超出标准范围 18%（过范围）。如果超出“过程值的上限”，将不再报告错误。仅识别断线和短路，然后 PID_Compact 切换到“未激活”模式。 r_Pv_Hlm > r_Pv_Llm
sPid_Cmpt.r_Pv_Llm	REAL	0.0	过程值的下限 r_Pv_Llm < r_Pv_Hlm
sPid_Cmpt.r_Pv_HWrn	REAL	+3.402822e+38	过程值的警告上限 如果设置的 r_Pv_HWrn 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作警告上限。 如果组态的 r_Pv_HWrn 值位于过程值的限值范围内，则该值将用作警告上限。 r_Pv_HWrn > r_Pv_LWrn r_Pv_HWrn ≤ r_Pv_Hlm

变量	数据类型	默认值	说明
sPid_Cmpt.r_Pv_LWrn	REAL	-3.402822e+38	过程值的警告下限 如果设置的 r_Pv_LWrn 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作警告下限。 如果组态的 r_Pv_LWrn 值位于过程值的限值范围内，则该值将用作警告下限。 r_Pv_LWrn < r_Pv_HWrn r_Pv_LWrn ≥ r_Pv_LWrn
sPidCalc.i_Ctrl_IOutv	REAL	0.0	当前积分作用
sParamCalc.i_Event_SUT	INT	0	变量 i_Event_SUT (页 339) 指示当前的“预调节”阶段：
sParamCalc.i_Event_TIR	INT	0	变量 i_Event_TIR (页 339) 指示当前的“精确调节”阶段：
sRet.i_Mode	INT	0	操作模式的更改由沿触发。 变量值发生变化时，将相应启用以下操作模式 • i_Mode = 0：“未激活”模式（控制器停止） • i_Mode = 1：“预调节”模式 • i_Mode = 2：“精确调节”模式 • i_Mode = 3：“自动”模式 • i_Mode = 4：“手动”模式 保持 i_Mode。
sRet.r_Ctrl_Gain	REAL	1.0	有效的比例增益 保持 Gain。
sRet.r_Ctrl_Ti	REAL	20.0	• r_Ctrl_Ti > 0.0：有效积分作用时间 • r_Ctrl_Ti = 0.0：积分作用取消激活 保持 r_Ctrl_Ti。
sRet.r_Ctrl_Td	REAL	0.0	• r_Ctrl_Td > 0.0：有效的微分作用时间 • r_Ctrl_Td = 0.0：微分作用取消激活 保持 r_Ctrl_Td。

变量	数据类型	默认值	说明
sRet.r_Ctrl_A	REAL	0.0	有效的微分延时系数 保持 r_Ctrl_A。
sRet.r_Ctrl_B	REAL	0.0	有效的比例作用权重 保持 r_Ctrl_B。
sRet.r_Ctrl_C	REAL	0.0	有效的微分作用权重 保持 r_Ctrl_C。
sRet.r_Ctrl_Cycle	REAL	1.0	有效的 PID 算法采样时间 在调节期间计算 r_Ctrl_Cycle，并将其舍入为 r_Cycle 的整数倍。 保持 r_Ctrl_Cycle。

说明

请在“未激活”模式下更改本表列出的变量，以防 PID 控制器出现故障。通过将“sRet.i_Mode”变量设置为“0”来强制切换为“未激活”模式。

参见

将工艺对象下载到设备 (页 76)

8.1.5.5 参数 State 和 sRet.i_Mode V1

参数的相关性

State 参数指示 PID 控制器的当前操作模式。 您无法修改 State 参数。

需要修改 sRet.i_Mode 变量来更改操作模式。 当新操作模式的值已处于 sRet.i_Mode 中时，这同样适用。 首先设置 sRet.i_Mode = 0 ，然后设置 sRet.i_Mode = 3。 如果控制器的当前操作模式支持此更改，则会将 State 设置为 sRet.i_Mode 的值。

PID_Compact 自动切换操作模式时，State != sRet.i_Mode。

示例：

- 成功预调节
  State = 3 并且 sRet.i_Mode = 1
- 出错
  State = 0 并且 sRet.i_Mode 的值保持不变，例如 sRet.i_Mode = 3
- ManualEnalbe = TRUE
  State = 4，并且 sRet.i_Mode 保持之前的值不变，例如 sRet.i_Mode = 3

说明

如果希望重复成功的精确调节而不退出自动模式，则设置 i_Mode = 0。

在一个周期内将 sRet.i_Mode 设置为无效值（例如 9999）对 State 没有影响。 在下一个周期中设置 Mode = 2。 这样可以对 sRet.i_Mode 进行更改而无需首先切换到“未激活”模式。

值的含义

State / sRet.i_Mode	操作模式说明
0	未激活 控制器关闭。 执行预调节前控制器处于“未激活”模式。 如果发生错误或者在调试窗口中单击“禁用控制器”(Deactivate controller) 图标，则运行中的 PID 控制器将切换为“未激活”模式。
1	预调节 预调节功能可确定对输出值跳变的过程响应，并搜索拐点。 根据受控系统的最大上升速率与死时间的函数计算最佳的 PID 参数。 预调节的要求： • 控制器处于未激活模式或手动模式 • ManualEnable = FALSE • 过程值不能过于接近设定值。 |Setpoint - Input| > 0.3 * |sPid_Cmpt.r_Pv_Hlm - sPid_Cmpt.r_Pv_Llm| 且 |Setpoint - Input| > 0.5 * |Setpoint| • 预调节期间不能更改设定值。 过程值的稳定性越高，就越容易计算 PID 参数以及提高结果的精度。 只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。 重新计算 PID 参数之前将对其进行备份并且可使用 sPid_Cmpt.b_LoadBackUp 重新激活这些参数。 预调节成功后自动模式将发生变化，预调节失败后“未激活”模式将发生变化。 预调节阶段由变量 i_Event_SUT V1 (页 339)来指示。

State / sRet.i_Mode	操作模式说明
2	精确调节 精确调节将使过程值出现恒定受限的振荡。 根据该振荡的幅度和频率对 PID 参数进行优化。 对预调节期间与精确调节期间的过程响应之间的差异进行分析。 所有 PID 参数都将根据相应结果进行重新计算。 精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。 PID_Compact 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。 重新计算 PID 参数之前将对其进行备份并且可使用 sPid_Cmpt.b_LoadBackUp 重新激活这些参数。 精确调节的要求： • 不能被干扰。 • 设定值和过程值均在组态的限值范围内。 • 精确调节期间不能更改设定值。 • ManualEnable = FALSE • 自动模式 (State = 3)、未激活模式 (State = 0) 或手动模式 (State = 4) 在以下模式下启动精确调节时，具体情况如下所述： • 自动模式 (State = 3) 如果希望通过控制器调节来改进现有 PID 参数，请在自动模式下启动精确调节。 PID_Comact 将使用现有的 PID 参数进行调节，直到控制回路已稳定并且精确调节的要求得到满足为止。之后才会启动精确调节。 • 未激活模式 (State = 0) 或手动模式 (State = 4) 如果满足预调节的要求，则启动预调节。 建立的 PID 参数将用于进行调节，直到控制回路已稳定并且精确调节的要求得到满足为止。之后才会启动精确调节。如果无法实现预调节，PID_Compact 将切换到“未激活”模式。 如果预调节的过程值已经十分接近设定值或者 sPid_Calc.b_RunIn = TRUE，则将尝试利用最小或最大输出值来达到设定值。这可能会增加超调量。 控制器将在成功完成“精确调节”后切换为“自动模式”，如果未成功完成“精确调节”，则切换为“未激活”模式。 “精确调节”阶段由变量 i_Event_TIR V1 (页 339)来指示。

State / sRet.i_Mode	操作模式说明
3	自动模式 在自动模式下，PID_Compact 会按照指定的参数来更正受控系统。 如果满足下列条件之一，控制器将切换到自动模式： • 预调节成功完成 • 精确调节成功完成 • 变量 sRet.i_Mode 的值变为 3。 CPU 启动或从 Stop 模式切换到 RUN 模式后，PID_Compact 会以最近激活的操作模式启动。要使 PID_Compact 保持在“未激活”模式下，应设置 sb_RunModeByStartup = FALSE。
4	手动模式 在手动模式下，在 ManualValue 参数中指定手动输出值。 如果 sRet.i_Mode = 4 或 ManualEnable 处于上升沿，将启用此操作模式。 如果 ManualEnable 更改为 TRUE，则只有 State 将发生改变。sRet.i_Mode 将保留其当前值。PID_Compact 将在 ManualEnable 处出现下降沿时返回到前一个操作模式。 到自动模式的切换是无扰动的。

参见

PID_Compact V1 的输出参数 (页 323)

预调节 V1 (页 119)

精确调节 V1 (页 121)

“手动”模式 V1 (页 123)

变量 i_Event_SUT V1 (页 339)

变量 i_Event_TIR V1 (页 339)

8.1.5.6 参数 Error V1

如果多个错误同时处于待决状态，将通过二进制加法显示错误代码值。例如，显示错误代码 0003 表示错误 0001 和 0002 同时处于待决状态。

Error (DW#16#...)	说明
0000	没有任何错误。
0001	参数“Input”超出了过程值限值的范围。 • Input > sPid_Cmpt.r_Pv_Hlm 或 • Input < sPid_Cmpt.r_Pv_Llm 在消除错误之前不能再次移动执行器。
0002	参数“Input_PER”的值无效。请检查模拟量输入是否有处于未决状态的错误。
0004	精确调节期间出错。过程值无法保持振荡状态。
0008	预调节启动时出错。过程值过于接近设定值。启动精确调节。
0010	调节期间设定值发生更改。
0020	在自动模式下或调节期间不允许进行预调节。
0080	输出值限值的组态不正确。 检查输出值的限值是否已正确组态及其是否匹配控制逻辑。
0100	调节期间的错误导致生成无效参数。
0200	参数“Input”的值无效：值的数字格式无效。
0400	输出值计算失败。请检查 PID 参数。
0800	采样时间错误：在循环中断 OB 的采样时间内没有调用 PID_Compact。 如果在使用 PLCSIM 进行仿真期间出现该错误，请参见使用 PLCSIM 仿真 PID_Compact V1 (页 124)下的说明。
1000	参数“Setpoint”的值无效：值的数字格式无效。

参见

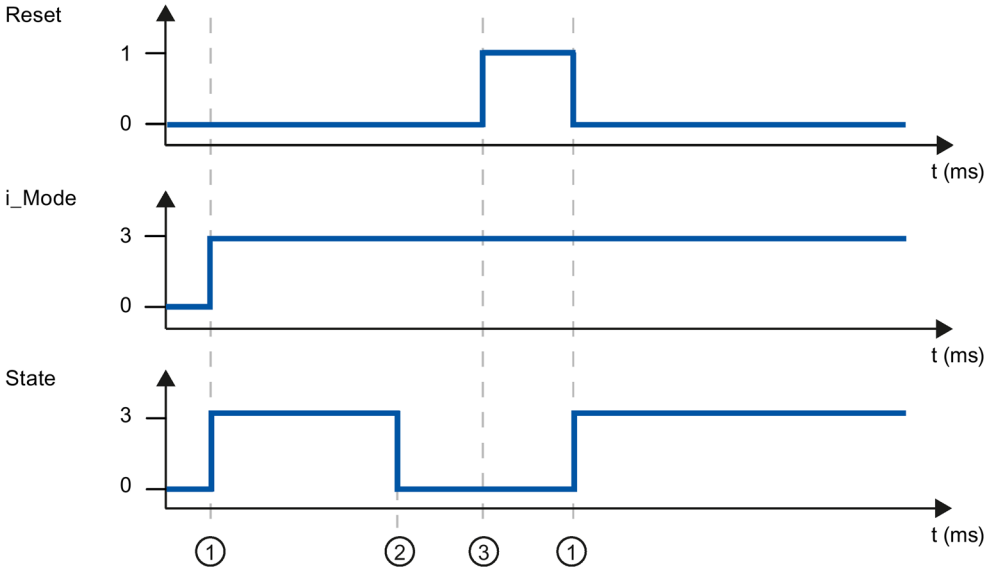
PID_Compact V1 的输出参数 (页 323)

8.1.5.7 参数 Reset V1

对 Reset = TRUE 的响应取决于 PID_Compact 指令的版本。

复位响应 PID_Compact V.1.1 或更高版本

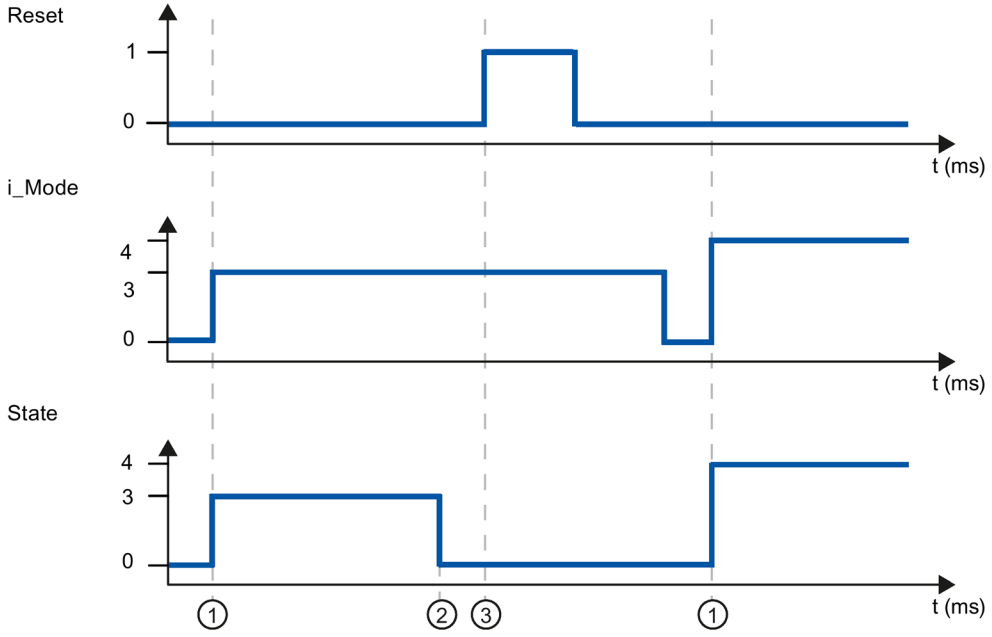
Reset 出现上升沿时会触发切换到“未激活”模式；复位错误和警告并删除积分作用。
Reset 出现下降沿时会触发切换到最近激活的操作模式。如果之前自动模式已激活，则会以无扰动切换的方式对积分作用进行预分配。



① 激活
② 错误
③ 复位

复位响应 PID_Compact V.1.0

Reset 出现上升沿时会触发切换到“未激活”模式；复位错误和警告并删除积分作用。在 i_Mode 出现下一沿之前，不会重新激活控制器。



① 激活
② 错误
③ 复位

8.1.5.8 变量 sd_warning V1

如果多个警告处于待决状态，将通过二进制加法显示变量 sd_warning 的值。 例如，显示警告 0003 表示警告 0001 和 0002 也处于待决状态。

sd_warning (DW#16#....)	说明
0000	无警告处于待决状态。
0001	预调节期间未发现拐点。
0002	精确调节期间振荡增加。
0004	设定值超出设置的限值。
0008	在所选计算方法中未定义所有必要的受控系统属性。 因此，使用“i_CtrlTypeTIR = 3”方法计算 PID 参数。
0010	由于 ManualEnable = TRUE，无法更改操作模式。
0020	调用 OB 的循环时间会限制 PID 算法的采样时间。 通过缩短 OB 循环时间来改进结果。
0040	过程值超出其警告限值之一。

以下警告在处理掉问题的原因后即被删除：

- 0004
- 0020
- 0040

所有其它警告均在 Reset 出现上升沿时清除。

8.1.5.9 变量 i_Event_SUT V1

i_Event_SUT	名称	说明
0	SUT_INIT	初始化预调节
100	SUT_STDABW	计算标准偏差
200	SUT_GET_POI	查找拐点
9900	SUT_IO	预调节成功
1	SUT_NIO	预调节未成功

参见

PID_Compact V1 的静态变量 (页 324)

参数 State 和 sRet.i_Mode V1 (页 331)

8.1.5.10 变量 i_Event_TIR V1

i_Event_TIR	名称	说明
-100	TIR_FIRST_SUT	无法进行精确调节。将首先执行预调节。
0	TIR_INIT	初始化精确调节
200	TIR_STDABW	计算标准偏差
300	TIR_RUN_IN	尝试达到设定值
400	TIR_CTRLN	尝试使用现有 PID 参数达到设定值 （如果预调节已成功）
500	TIR_OSZIL	确定波动并计算参数
9900	TIR_IO	精确调节成功
1	TIR_NIO	精确调节未成功

参见

PID_Compact V1 的静态变量 (页 324)

参数 State 和 sRet.i_Mode V1 (页 331)

8.2 PID_3Step

8.2.1 PID_3Step 的新特性

PID_3Step V2.3

- 自 PID_3Step 版本 2.3 起，可以通过 Config.VirtualActuatorLimit = 0.0 取消激活行程时间的监视和限制。

PID_3Step V2.2

- **与 S7-1200 结合使用**

  自 PID_3Step V2.2 起，固件为 4.0 或更高版本的 S7-1200 上也可以使用具有 V2 功能的指令。

PID_3Step V2.0

- **对错误的响应**

  对 ActivateRecoverMode = TRUE 的响应已经过全面改进。在默认设置下，PID_3Step 的响应方式具有更强的容错性。

注意
您的系统可能已损坏。 如果使用默认设置，则即使超过过程值的限值，PID_3Step 也将保持自动模式。这可能损坏您的系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

  可使用 ErrorAck 输入参数在不重启控制器或清除积分作用的情况下确认错误和警告。

  切换工作模式不会确认处于非未决状态的错误。

- **切换工作模式**

  在 Mode 输入/输出参数处指定工作模式，并在 ModeActivate 的上升沿启动该工作模式。Retain.Mode 变量已被忽略。

  转换时间测量不能再通过 GetTransitTime.Start 进行启动，只能通过 Mode = 6 并在 ModeActivate 的上升沿进行启动。

- **多重背景功能**

  可将 PID_3Step 作为多重背景数据块进行调用。这种情况下，不会创建任何工艺对象，也没有任何参数分配接口或调试接口可用。必须直接在多重背景数据块中为 PID_3Step 分配参数，并通过监视表格进行调试。

- **启动特性**

  如果 RunModeByStartup = TRUE，则在 Mode 参数处指定的工作模式也将在 Reset 的下降沿并在 CPU 冷启动期间启动。

- **ENO 特性**

  ENO 根据工作模式进行设置。

  如果 State = 0，那么 ENO = FALSE。

  如果 State ≠ 0，那么 ENO = TRUE。

- **手动模式**

  Manual_UP 和 Manual_DN 输入参数不再用作沿触发参数。使用 ManualUpInternal 和 ManualDnInternal 变量仍可实现沿触发手动模式。

  在“无停止位信号的手动模式”(Mode = 10) 下，停止位信号 Actuator_H 和 Actuator_L 即使激活也被忽略。

- **PID 参数的默认值**

  下列默认设置已更改：

  – 比例作用权重 (PWeighting)，从 0.0 到 1.0

  – 微分作用权重 (DWeighting)，从 0.0 到 1.0

  – 微分延迟系数 (TdFiltRatio)，从 0.0 到 0.2

- **电机转换时间限制**

  在 Config.VirtualActuatorLimit 变量中组态执行器在一个方向上行进的时间占电机转换时间的最大百分比。

- **在调节期间指定设定值**

  在 CancelTuningLevel 变量中进行调节期间，组态允许的设定值波动。

- **启用扰动变量**

  可在 Disturbance 参数中启用扰动变量。

- **故障排除**

  如果未激活停止位信号 (ActuatorEndStopOn = FALSE)，将在无 Actuator_H 或 Actuator_L 的情况下确定 ScaledFeedback。

PID_3Step V1.1

- **CPU 启动时的手动模式**

  如果 CPU 启动时 ManualEnable = TRUE，则 PID_3Step 以手动模式启动。无需在 ManualEnable 上升沿启动。

- **对错误的响应**

  ActivateRecoverMode 变量在手动模式下不再有效。

- **故障排除**

  成功调节或转换时间测量后复位 Progress 变量。

8.2.2 与 CPU 和 FW 的兼容性

下表显示了 PID_3Step 的每个版本可用于哪种 CPU。

CPU	FW	PID_3Step
S7-1200	V4.2 或更高版本	V2.3 V2.2 V1.1
	V4.0 到 V4.1	V2.2 V1.1
	V3.x	V1.1 V1.0
	V2.x	V1.1 V1.0
	V1.x	-
S7-1500	V2.0 或更高版本	V2.3 V2.2 V2.1 V2.0
	V1.5 到 V1.8	V2.2 V2.1 V2.0
	V1.1	V2.1 V2.0
	V1.0	V2.0

8.2.3 PID_3Step V2.x 的 CPU 处理时间和存储器要求

CPU 处理时间

自版本 V2.0 起的 PID_3Step 工艺对象典型 CPU 处理时间（取决于 CPU 类型）。

CPU	典型 CPU 处理时间 (PID_3Step V2.x)
CPU 1211C ≥ V4.0	410 µs
CPU 1215C ≥ V4.0	410 µs
CPU 1217C ≥ V4.0	410 µs
CPU 1505S ≥ V1.0	50 µs
CPU 1510SP-1 PN ≥ V1.6	120 µs
CPU 1511-1 PN ≥ V1.5	120 µs
CPU 1512SP-1 PN ≥ V1.6	120 µs
CPU 1516-3 PN/DP ≥ V1.5	65 µs
CPU 1518-4 PN/DP ≥ V1.5	5 µs

存储器要求

自版本 V2.0 起的 PID_3Step 工艺对象背景数据块的存储器要求。

	PID_3Step V2.x 背景数据块的 存储器要求
装载存储器要求	约 15000 个字节
总工作存储器要求	1040 个字节
保持性工作存储器要求	60 个字节

8.2.4 PID_3Step V2

8.2.4.1 PID_3Step V2 说明

说明

使用 PID_3Step 指令可对具有阀门自调节的 PID 控制器或具有积分行为的执行器进行组态。

存在下列工作模式：

- 未激活
- 预调节
- 精确调节
- 自动模式
- 手动模式
- 逼近替代输出值
- 转换时间测量
- 错误监视
- 在监视错误的同时逼近替代输出值
- 无停止位信号的手动模式

有关工作模式的详细信息，请参见 State 参数。

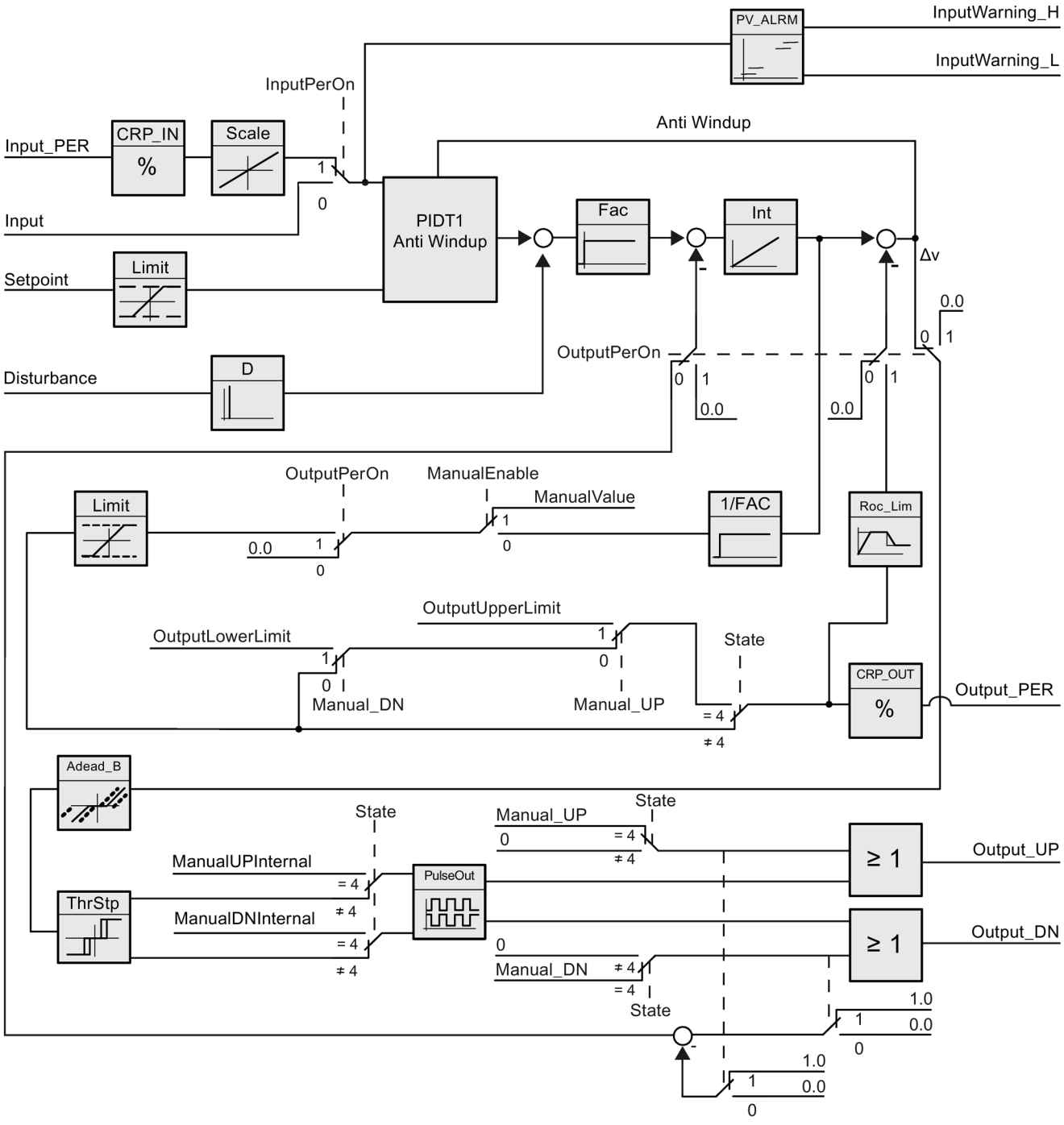
PID 算法

PID_3Step 是一种具有抗积分饱和功能并且能够对比例作用和微分作用进行加权的 PIDT1 控制器。 PID 算法根据以下等式工作：

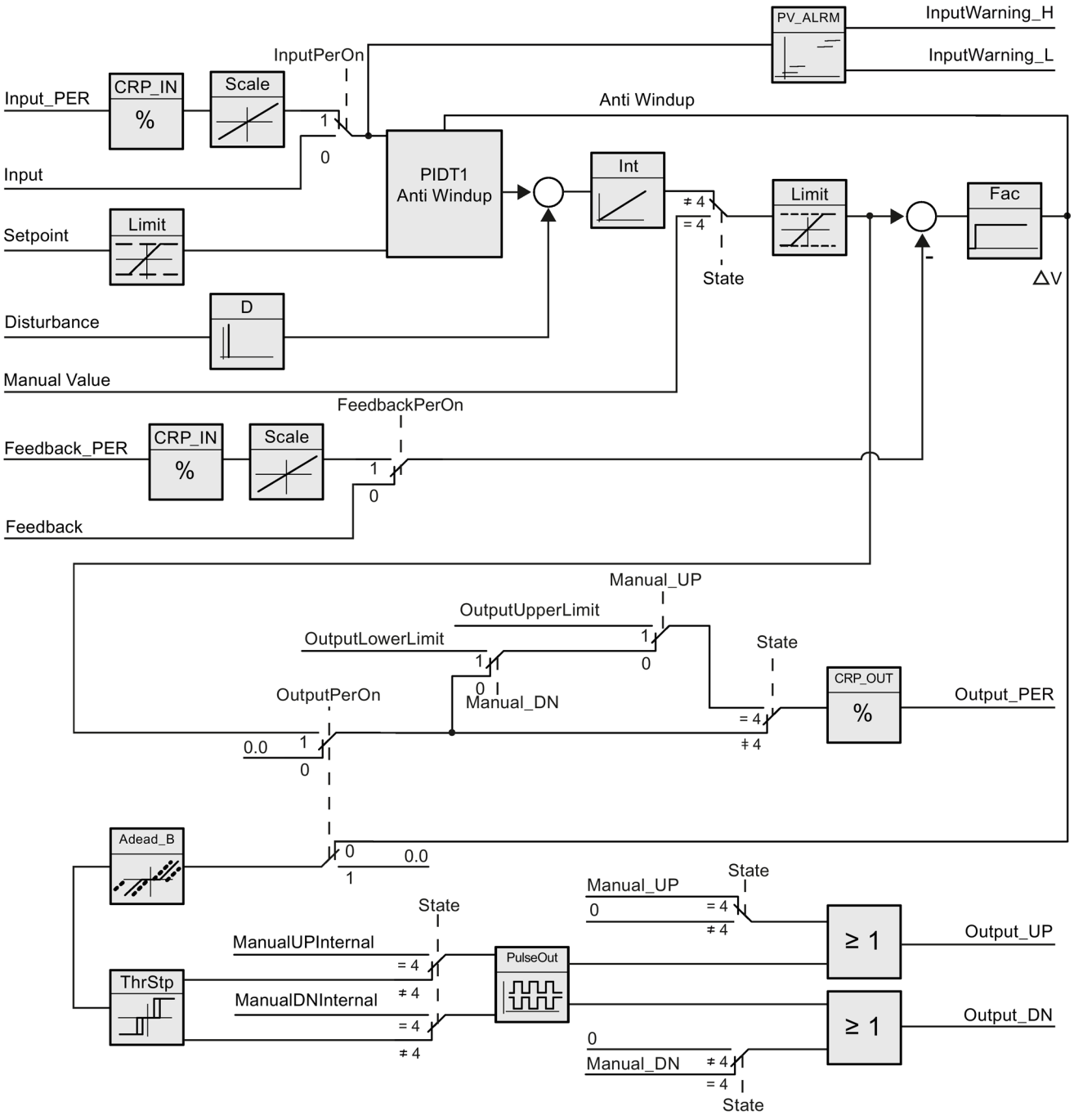
$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
Δy	PID 算法的输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分作用时间
T_D	微分作用时间
a	微分延迟系数（微分延迟 T1 = a × T_D）
c	微分作用权重

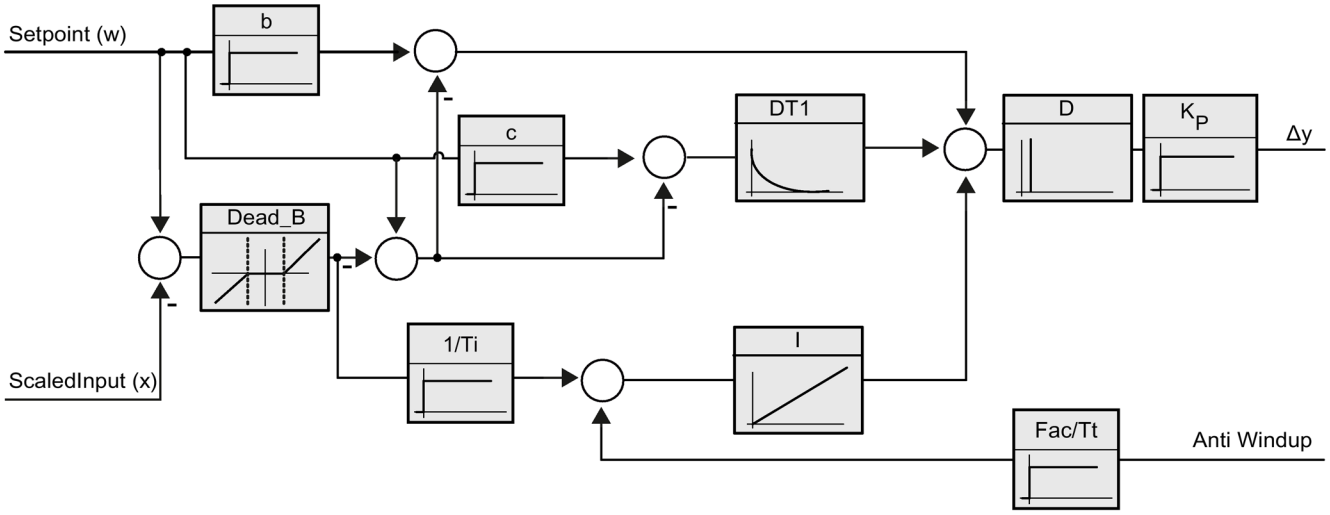
不带位置反馈的方框图


PV_ALRM
InputWarning_H
InputWarning_L
InputPerOn
Input_PER
CRP_IN
%
Scale
Anti Windup
Input
Setpoint
Limit
PIDT1
Anti Windup
Fac
Int
Δv
Disturbance
D
OutputPerOn
ManualEnable
ManualValue
1/FAC
Roc_Lim
OutputUpperLimit
OutputLowerLimit
State
Manual_DN
Manual_UP
CRP_OUT
Output_PER
Adead_B
ThrStp
ManualUPInternal
ManualDNInternal
PulseOut
≥ 1
Output_UP
Output_DN

带位置反馈的方框图


PV_ALRM
InputWarning_H
InputWarning_L
InputPerOn
Input_PER
CRP_IN
%
Scale
1
0
Anti Windup
Input
PIDT1
Anti Windup
Int
≠ 4
= 4
Limit
Fac
Setpoint
Limit
State
ΔV
Disturbance
D
Manual Value
FeedbackPerOn
Feedback_PER
CRP_IN
%
Scale
1
0
Feedback
Manual_UP
OutputUpperLimit
OutputLowerLimit
1
0
State
CRP_OUT
%
Output_PER
OutputPerOn
Manual_DN
0.0
1
0
= 4
≠ 4
Adead_B
0
1
0.0
State
Manual_UP
0
= 4
≠ 4
State
ManualUPInternal
= 4
≠ 4
ThrStp
ManualDNInternal
= 4
≠ 4
PulseOut
≥ 1
Output_UP
≥ 1
Output_DN
0
Manual_DN
≠ 4
= 4
State

带抗积分饱和的 PIDT1 的方框图



调用

在周期中断 OB 的恒定时间范围内调用 PID_3Step。

如果将 PID_3Step 作为多重背景数据块调用，将不会创建任何工艺对象。没有参数分配接口或调试接口可用。 必须直接在多重背景数据块中为 PID_3Step 分配参数，并通过监视表格进行调试。

下载到设备

仅当完全下载 PID_3Step 后，才能更新保持性变量的实际值。

将工艺对象下载到设备 (页 76)

启动

CPU 启动时，PID_3Step 以保存在 Mode 输入/输出参数中的工作模式启动。 要使 PID_3Step 保持在“未激活”模式下，应设置 RunModeByStartup = FALSE。

对错误的响应

在自动模式下和调试期间，对错误的响应取决于 ErrorBehaviour 和 ActivateRecoverMode 变量。 在手动模式下，该响应与 ErrorBehaviour 和 ActivateRecoverMode 变量无关。 如果 ActivateRecoverMode = TRUE 变量，则该响应还取决于所发生的错误。

ErrorBehaviour	ActivateRecoverMode	组态编辑器 > 执行器设置 > 将 Output 设置为	响应
FALSE	FALSE	当前输出值	切换到“未激活”模式 (State = 0) 执行器保持在当前位置。
FALSE	TRUE	错误未决时的当前输出值	切换到“错误监视”模式 (State = 7) 当错误未决时，执行器保持在当前位置。
TRUE	FALSE	替代输出值	切换到“逼近替代输出值”模式 (State = 5) 执行器移动到组态的替代输出值位置。 切换到“未激活”模式 (State = 0) 执行器保持在当前位置。
TRUE	TRUE	错误未决时的替代输出值	切换到“在监视错误的同时逼近替代输出值”模式 (State = 8) 执行器移动到组态的替代输出值位置。 切换到“错误监视”模式 (State = 7)

在手动模式下，PID_3Step 使用 ManualValue 作为输出值，除非出现以下错误：

- 2000h： Feedback_PER 参数的值无效。
- 4000h： Feedback 参数的值无效。
- 8000h： 数字位置反馈期间出错。

只能通过 Manual_UP 和 Manual_DN 更改执行器的位置，不能通过 ManualValue 更改。

Error 参数指示在此周期中是否已发生错误。 ErrorBits 参数显示了已发生的错误。通过 Reset 或 ErrorAck 的上升沿来复位 ErrorBits。

参见

模式 V2 的参数状态 (页 372)
参数 ErrorBits V2 (页 379)
组态 PID_3Step V2 (页 127)

8.2.4.2 PID_3Step V2 的工作模式

监视过程值的限值

在 Config.InputUpperLimit 和 Config.InputLowerLimit 变量中指定过程值的上限和下限。如果过程值超出这些限值，将出现错误 (ErrorBits = 0001h)。

在 Config.InputUpperWarning 和 Config.InputLowerWarning 变量中指定过程值的警告上限和警告下限。如果过程值超出这些警告限值，将发生警告 (Warning = 0040h)，并且 InputWarning_H 或 InputWarning_L 输出参数会更改为 TRUE。

限制设定值

在 Config.SetpointUpperLimit 和 Config.SetpointLowerLimit 变量中指定设定值的上限和下限。PID_3Step 会自动将设定值限制在过程值的限值范围内。可以将设定值限制在更小的范围内。PID_3Step 会检查此范围是否处于过程值的限值范围内。如果设定值超出这些限值，上限和下限将用作设定值，并且输出参数 SetpointLimit_H 或 SetpointLimit_L 将设置为 TRUE。

在所有操作模式下均限制设定值。

限制输出值

在 Config.OutputUpperLimit 变量和 Config.OutputLowerLimit 变量中指定输出值的上限和下限。输出值的限值必须位于“下端停止位”和“上端停止位”范围内。

- 上端停止位：Config.FeedbackScaling.UpperPointOut
- 下端停止位：Config.FeedbackScaling.LowerPointOut

规则：

UpperPointOut ≥ OutputUpperLimit > OutputLowerLimit ≥ LowerPointOut

“上端停止位”和“下端停止位”的有效值取决于：

- FeedbackOn
- FeedbackPerOn
- OutputPerOn

OutputPerOn	FeedbackOn	FeedbackPerOn	LowerPointOut	UpperPointOut
FALSE	FALSE	FALSE	无法设置 (0.0%)	无法设置 (100.0%)
FALSE	TRUE	FALSE	-100.0% 或 0.0%	0.0% 或 +100.0%
FALSE	TRUE	TRUE	-100.0% 或 0.0%	0.0% 或 +100.0%
TRUE	FALSE	FALSE	无法设置 (0.0%)	无法设置 (100.0%)
TRUE	TRUE	FALSE	-100.0% 或 0.0%	0.0% 或 +100.0%
TRUE	TRUE	TRUE	-100.0% 或 0.0%	0.0% 或 +100.0%

如果 OutputPerOn = FALSE 且 FeedbackOn = FALSE，则无法限制输出值。Output_UP 和 Output_DN 将在 Actuator_H = TRUE 或 Actuator_L = TRUE 时复位。若停止位信号也没出现，则 Output_UP 和 Output_DN 将在行程时间 Config.VirtualActuatorLimit × Retain.TransitTime/100 后复位。自 PID_3Step 版本 2.3 起，可以通过 Config.VirtualActuatorLimit = 0.0 取消激活行程时间的监视和限制。

输出值在 100% 时为 27648，在 -100% 时为 -27648。PID_3Step 必须能够完全关闭阀门。

说明

与两个或多个执行器结合使用

PID_3 Step 不适合与两个或多个执行器结合使用（例如，在加热/制冷应用中），因为不同的执行器需要不同的 PID 参数以实现良好的控制响应。

替代输出值

如果出现错误，PID_3Step 可输出一个替代输出值并将执行器移至变量 SavePosition 中指定的安全位置。替代输出值必须处于输出值的限值范围内。

监视信号有效性

使用以下参数时，监视其有效性：

- Setpoint
- Input
- Input_PER
- Input_PER
- Feedback
- Feedback_PER
- Disturbance
- ManualValue
- SavePosition
- Output_PER

监视 PID_3Step 采样时间

理想情况下，采样时间等于调用 OB 的周期时间。PID_3Step 指令测量两次调用之间的时间间隔。这就是当前采样时间。每次切换工作模式以及初始启动期间，平均值由前 10 个采样时间构成。当前采样时间与该平均值之间的差值过大时会触发错误 (ErrorBits = 0800h)。

如果存在以下情况，调节期间将发生错误：

- 新平均值 >= 1.1 x 原平均值
- 新平均值 <= 0.9 x 原平均值

如果存在以下情况，将在自动模式下发生错误：

- 新平均值 >= 1.5 x 原平均值
- 新平均值 <= 0.5 x 原平均值

如果禁用采样时间监视 (CycleTime.EnMonitoring = FALSE)，则也可在 OB1 中调用 PID_3Step。由于采样时间发生偏离，因此随后必须接受质量较低的控制。

PID 算法的采样时间

受控系统需要一定的时间来对输出值的变化做出响应。因此，建议不要在每次循环中都计算输出值。PID 算法的采样时间是两次计算输出值之间的时间。该时间在调节期间进行计算，并舍入为循环时间的倍数。PID_3Step 的所有其它功能在每次调用时均执行。

测量电机转换时间

电机转换时间指的是电机将执行器从关闭状态转为开启状态所需的时间（以秒为单位）。执行器朝一个方向移动的最长时间为 Config.VirtualActuatorLimit × Retain.TransitTime/100。PID_3Step 要求电机转换时间尽可能准确，以便获得良好的控制器结果。执行器文档中的数据包含此类执行器的平均值。针对特定执行器的值可能不同。可以在调试期间测量电机转换时间。测量电机转换时间期间，不考虑输出值的限值。执行器可行进至上端停止位或下端停止位。

在计算模拟量输出值和数字量输出值时，会将电机转换时间考虑在内。自动调节和抗饱和行为期间，需要该时间来确保正常运行。因此，应该将电机转换时间组态为电机将执行器从关闭状态转换至开启状态所需的值。

如果相关电机转换时间并未影响过程（如使用电磁阀），因此输出值直接且完全影响过程，则使用 PID_Compact。

控制逻辑

通常，可通过增大输出值来增大过程值。这种做法称为常规控制逻辑。对于制冷和放电控制系统，可能需要反转控制逻辑。PID_3Step 不使用负比例增益。如果 InvertControl = TRUE，则不断增大的控制偏差将导致输出值减小。在预调节和精确调节期间还会考虑控制逻辑。

参见

组态 PID_3Step V1 (页 150)

8.2.4.3 更改 PID_3Step V2 接口

下表显示了 PID_3Step 指令接口中的一些变化。

PID_3Step V1	PID_3Step V2	更改
Input_PER	Input_PER	数据类型由字改为整数
Feedback_PER	Feedback_PER	数据类型由字改为整数
	Disturbance	新增
Manual_UP	Manual_UP	功能
Manual_DN	Manual_DN	功能
	ErrorAck	新增
	ModeActivate	新增
Output_PER	Output_PER	数据类型由字改为整数
	ManualUPInternal	新增
	ManualDNInternal	新增
	CancelTuningLevel	新增
	VirtualActuatorLImit	新增
Config.Loadbackup	Loadbackup	重命名
Config.TransitTime	Retain.TransitTime	重命名并已添加保持性
GetTransitTime.Start		由 Mode 和 ModeActivate 代替
SUT.CalculateSUTParams	SUT.CalculateParams	重命名
SUT.TuneRuleSUT	SUT.TuneRule	重命名
TIR.CalculateTIRParams	TIR.CalculateParams	重命名
TIR.TuneRuleTIR	TIR.TuneRule	重命名
Retain.Mode	Mode	功能 声明静态输入/输出参数

8.2.4.4 PID_3Step V2 的输入参数

表格 8- 7

参数	数据类型	默认值	说明
Setpoint	REAL	0.0	PID 控制器在自动模式下的设定值
Input	REAL	0.0	用户程序的变量用作过程值的源。 如果正在使用参数 Input，则必须设置 Config.InputPerOn = FALSE。
Input_PER	INT	0	模拟量输入用作过程值的源。 如果正在使用参数 Input_PER，则必须设置 Config.InputPerOn = TRUE。
Actuator_H	BOOL	FALSE	阀门处于上端停止位时的数字位置反馈 如果 Actuator_H = TRUE，表明阀门处于上端停止位，并且不再向此方向移动。
Actuator_L	BOOL	FALSE	阀门处于下端停止位时的数字位置反馈 如果 Actuator_L = TRUE，表明阀门处于下端停止位，并且不再向此方向移动。
Feedback	REAL	0.0	阀门的位置反馈 如果正在使用参数 Feedback，则必须设置 Config.FeedbackPerOn = FALSE。
Feedback_PER	INT	0	阀门的模拟位置反馈 如果正在使用参数 Feedback_PER，则必须设置 Config.FeedbackPerOn = TRUE。 根据以下变量标定 Feedback_PER： • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut
Disturbance	REAL	0.0	扰动变量或预控制值

参数	数据类型	默认值	说明
ManualEnable	BOOL	FALSE	• 出现 FALSE -> TRUE 沿时会激活“手动模式”，而 State = 4 和 Mode 保持不变。 只要 ManualEnable = TRUE，便无法通过 ModeActivate 的上升沿或使用调试对话框来更改工作模式。 • 出现 TRUE -> FALSE 沿时会激活由 Mode 指定的工作模式。 建议只使用 ModeActivate 更改工作模式。
ManualValue	REAL	0.0	在手动模式下指定阀门的绝对位置。只有在使用 Output_PER，**或**位置反馈可用时，才对 ManualValue 进行评估。
Manual_UP	BOOL	FALSE	• Manual_UP = TRUE 即使正在使用 Output_PER 或位置反馈，阀门也打开。如果已达到上端停止位，则阀门将不再移动。 另参见 Config.VirtualActuatorLimit • Manual_UP = FALSE 如果正在使用 Output_PER 或位置反馈，则阀门移至 ManualValue。否则阀门不再移动。 如果 Manual_UP 和 Manual_DN 同时设置为 TRUE，则阀门不移动。
Manual_DN	BOOL	FALSE	• Manual_DN = TRUE 即使正在使用 Output_PER 或位置反馈，阀门也关闭。如果已达到下端停止位，则阀门将不再移动。 另参见 Config.VirtualActuatorLimit • Manual_DN = FALSE 如果正在使用 Output_PER 或位置反馈，则阀门移至 ManualValue。否则阀门不再移动。
ErrorAck	BOOL	FALSE	• FALSE -> TRUE 沿 将复位 ErrorBits 和 Warning。

参数	数据类型	默认值	说明
Reset	BOOL	FALSE	重新启动控制器。 • FALSE -> TRUE 沿 – 切换到“未激活”模式 – 将复位 ErrorBits 和 Warnings。 • 只要 Reset = TRUE， – PID_3Step 将保持在“未激活”模式下 (State = 0)。 – 无法通过 Mode 和 ModeActivate 或 ManualEnable 更改工作模式。 – 无法使用调试对话框。 • TRUE -> FALSE 沿 – 如果 ManualEnable = FALSE，则 PID_3Step 会切换到保存在 Mode 中的工作模式。 – 如果 Mode = 3，会将积分作用视为已通过变量 IntegralResetMode 进行组态。
ModeActivate	BOOL	FALSE	• FALSE -> TRUE 沿 PID_3Step 将切换到保存在 Mode 参数中的工作模式。

8.2.4.5 PID_3Step V2 的输出参数

表格 8- 8

参数	数据类型	默认值	说明
ScaledInput	REAL	0.0	标定的过程值
ScaledFeedback	REAL	0.0	标定的位置反馈 对于没有位置反馈的执行器，由 ScaledFeedback 指示的执行器位置非常不精确。这种情况下，ScaledFeedback 只可用于粗略估计当前位置。
Output_UP	BOOL	FALSE	用于打开阀门的数字量输出值 如果 Config.OutputPerOn = FALSE，则使用参数 Output_UP。
Output_DN	BOOL	FALSE	用于关闭阀门的数字量输出值 如果 Config.OutputPerOn = FALSE，则使用参数 Output_DN。
Output_PER	INT	0	模拟量输出值 如果 Config.OutputPerOn = TRUE，则使用 Output_PER。 如果将一个阀门用作通过模拟量输出进行触发并使用连续信号（例如，0...10 V 或 4...20 mA）进行控制的执行器，则使用 Output_PER。 Output_PER 的值与阀门的目标位置相对应，例如，当阀门打开 50% 时 Output_PER = 13824。
SetpointLimit_H	BOOL	FALSE	如果 SetpointLimit_H = TRUE，则说明达到了设定值的绝对上限 (Setpoint ≥ Config.SetpointUpperLimit)。 此设定值将限制为 Config.SetpointUpperLimit 。
SetpointLimit_L	BOOL	FALSE	如果 SetpointLimit_L = TRUE，则说明已达到设定值的绝对下限 (Setpoint ≤ Config.SetpointLowerLimit)。 此设定值将限制为 Config.SetpointLowerLimit 。
InputWarning_H	BOOL	FALSE	如果 InputWarning_H = TRUE，则说明过程值已达到或超出警告上限。

参数	数据类型	默认值	说明
InputWarning_L	BOOL	FALSE	如果 InputWarning_L = TRUE，则说明过程值已经达到或低于警告下限。
State	INT	0	State 参数 (页 372)显示了 PID 控制器的当前工作模式。可使用输入参数 Mode 和 ModeActivate 处的上升沿更改工作模式。 • State = 0：未激活 • State = 1：预调节 • State = 2：精确调节 • State = 3：自动模式 • State = 4：手动模式 • State = 5：逼近替代输出值 • State = 6：转换时间测量 • State = 7：错误监视 • State = 8：在监视错误的同时逼近替代输出值 • State = 10：无停止位信号的手动模式
Error	BOOL	FALSE	如果 Error = TRUE，则此周期内至少有一条错误消息处于未决状态。
ErrorBits	DWORD	DW#16#0	ErrorBits 参数 (页 379)显示了处于未决状态的错误消息。通过 Reset 或 ErrorAck 的上升沿来保持并复位 ErrorBits。

参见

模式 V2 的参数状态 (页 372)

参数 ErrorBits V2 (页 379)

8.2.4.6 PID-3Step V2 输入/输出参数

表格 8- 9

参数	数据类型	默认值	说明
Mode	INT	4	在模式参数中，指定 PID_3Step 将要切换到的工作模式。 选项包括： • Mode = 0：未激活 • Mode = 1：预调节 • Mode = 2：精确调节 • Mode = 3：自动模式 • Mode = 4：手动模式 • Mode = 6：转换时间测量 • Mode = 10：无停止位信号的手动模式 工作模式由以下沿激活： • ModeActivate 的上升沿 • Reset 的下降沿 • ManualEnable 的下降沿 • 如果 RunModeByStartup = TRUE，则冷启动 CPU。 保持 Mode。 有关工作模式的详细说明，请参见模式 V2 的参数状态 (页 372)。

8.2.4.7 PID_3Step V2 的静态变量

不得更改未列出的变量。这些变量仅供内部使用。

变量	数据类型	默认值	说明
ManualUpInternal	BOOL	FALSE	在手动模式下，每次出现上升沿时，阀门都将打开总控制范围的 5%，或者持续打开最短的电机转换时间。只有在未使用 Output_PER 和位置反馈时，才会对 ManualUpInternal 进行评估。此变量用在调试对话框。
ManualDnInternal	BOOL	FALSE	在手动模式下，每次出现上升沿时，阀门都将关闭总控制范围的 5%，或者持续关闭最短的电机转换时间。只有在未使用 Output_PER 和位置反馈时，才会对 ManualDnInternal 进行评估。此变量用在调试对话框。

变量	数据类型	默认值	说明
ActivateRecoverMode	BOOL	TRUE	ActivateRecoverMode V2 (页 383) 变量确定错误响应方式。
RunModeByStartup	BOOL	TRUE	CPU 重启后，激活 Mode 参数中的工作模式。 如果 RunModeByStartup = TRUE，PID_3Step 将在 CPU 启动后以保存在 Mode 参数中的工作模式启动。 如果 RunModeByStartup = FALSE，PID_3Step 在 CPU 启动后仍保持“未激活”模式下。
LoadBackUp	BOOL	FALSE	如果 LoadBackUp = TRUE，则重新加载上一个 PID 参数集。该设置在最后一次调节前保存。LoadBackUp 自动设置回 FALSE。
PhysicalUnit	INT	0	过程值和设定值的测量单位，例如 ºC 或 ºF。
PhysicalQuantity	INT	0	过程值和设定值的物理量，如温度
ErrorBehaviour	BOOL	FALSE	如果 ErrorBehaviour = FALSE 且发生错误，则阀门会停留在其当前位置，控制器会直接切换到“未激活”模式或“错误监视”模式。 如果 ErrorBehaviour = TRUE 且出现了错误，则执行器将移动到替代输出值对应的位置，并且仅在此时才切换到“未激活”模式或“错误监视”模式。 如果发生以下错误，将不再能将阀门移动到组态的替代输出值对应的位置。 • 2000h：Feedback_PER 参数的值无效。 • 4000h：Feedback 参数的值无效。 • 8000h：数字位置反馈期间出错。 • 20000h：变量 SavePosition 的值无效。
Warning	DWORD	DW#16#0	Warning 变量 (页 372)显示自 Reset = TRUE 或 ErrorAck =TRUE 以来的警告。Warning 具有保持性。 在删除警告原因前，会一直显示循环警告（如过程值警告）。一旦其产生原因消失，将自动删除这些警告。非循环警告（如未发现拐点）会保留且可以像错误一样被删除。
SavePosition	REAL	0.0	替代输出值 如果 ErrorBehaviour = TRUE，则在发生错误时执行器移至对工厂安全的位置。到达替代输出值后，PID_3Step 根据 ActivateRecoverMode 立即切换工作模式。

变量	数据类型	默认值	说明
CurrentSetpoint	REAL	0.0	当前活动的设定值。此值将在调节开始时冻结。
CancelTuningLevel	REAL	10.0	调节期间允许的设定值拐点。出现以下情况之前，不会取消调节： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel
Progress	REAL	0.0	百分数形式的调节进度 (0.0 - 100.0)
Config.InputPerOn	BOOL	TRUE	如果 InputPerOn = TRUE，则使用参数 Input_PER。如果 InputPerOn = FALSE，则使用参数 Input。
Config.OutputPerOn	BOOL	FALSE	如果 OutputPerOn = TRUE，则使用参数 Output_PER。如果 OutputPerOn = FALSE，则将使用 Ouput_UP 和 Output_DN 参数。
Config.InvertControl	BOOL	FALSE	反转控制逻辑 如果 InvertControl = TRUE，则不断增大的控制偏差将导致输出值减小。
Config.FeedbackOn	BOOL	FALSE	如果 FeedbackOn = FALSE，则会仿真位置反馈。 位置反馈通常在 FeedbackOn = TRUE 时激活。
Config.FeedbackPerOn	BOOL	FALSE	仅当 FeedbackOn = TRUE 时，FeedbackPerOn 才有效。 如果 FeedbackPerOn = TRUE，则将模拟量输入用于位置反馈（Feedback_PER 参数）。 如果 FeedbackPerOn = FALSE，则将 Feedback 参数用于位置反馈。
Config.ActuatorEndStopOn	BOOL	FALSE	如果 ActuatorEndStopOn = TRUE，则将考虑数字位置反馈 Actuator_L 和 Actuator_H 。

变量	数据类型	默认值	说明
Config.InputUpperLimit	REAL	120.0	过程值的上限 监控 Input 和 Input_PER，以确保符合此限制。 在 I/O 输入中，过程值最大可超出标准范围 18%（过范围）。因超出“过程值上限”，将不再报告错误。仅识别断线和短路，然后 PID_3Step 将根据已组态的错误响应方式进行响应。 InputUpperLimit > InputLowerLimit
Config.InputLowerLimit	REAL	0.0	过程值的下限 InputLowerLimit < InputUpperLimit
Config.InputUpperWarning	REAL	+3.402822e+38	过程值的警告上限 如果设置的 InputUpperWarning 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作警告上限。 如果组态的 InputUpperWarning 值位于过程值的限值范围内，则该值将用作警告上限。 InputUpperWarning > InputLowerWarning InputUpperWarning ≤ InputUpperLimit
Config.InputLowerWarning	REAL	-3.402822e+38	过程值的警告下限 如果设置的 InputLowerWarning 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作警告下限。 如果组态的 InputLowerWarning 值位于过程值的限值范围内，则该值将用作警告下限。 InputLowerWarning < InputUpperWarning InputLowerWarning ≥ InputLowerLimit
Config.OutputUpperLimit	REAL	100.0	输出值的上限 有关详细信息，请参见 OutputLowerLimit
Config.OutputLowerLimit	REAL	0.0	输出值的下限 如果 OutputPerOn = TRUE 或 FeedbackOn = TRUE，则 -100% 到 +100% 的值范围有效（包括零）。-100% 时，Output = -27648；+100% 时，Output = 27648 如果 OutputPerOn = FALSE，则 0% 到 100% 的值范围有效。阀门在 0% 时完全关闭，在 100% 时完全打开。

变量	数据类型	默认值	说明
Config.SetpointUpperLimit	REAL	+3.402822e+38	设定值的上限 如果设置的 SetpointUpperLimit 超出了过程值的限值范围，则所组态的绝对过程值上限将预分配为设定值的上限。 如果组态的 SetpointUpperLimit 值位于过程值的限值范围内，则该值将用作设定值的上限。
Config.SetpointLowerLimit	REAL	- 3.402822e+38	设定值的下限 如果设置的 SetpointLowerLimit 超出了过程值的限值范围，则所组态的绝对过程值下限将预分配为设定值的下限。 如果设置的 SetpointLowerLimit 值位于过程值的限值范围内，则该值将用作设定值的下限。
Config.MinimumOnTime	REAL	0.0	最短 ON 时间 伺服驱动器必须开启的最短时间（以秒为单位）。 只有在使用 Output_UP 和 Output_DN 的情况下 (Config.OutputPerOn = FALSE)，Config.MinimumOnTime 才有效。
Config.MinimumOffTime	REAL	0.0	最短 OFF 时间 伺服驱动器必须关闭的最短时间（以秒为单位）。 只有在使用 Output_UP 和 Output_DN 的情况下 (Config.OutputPerOn = FALSE)，Config.MinimumOffTime 才有效。

变量	数据类型	默认值	说明
Config.VirtualActuatorLimit	REAL	150.0	如果所有以下条件都已满足，则执行器朝一个方向移动的最长时间为 VirtualActuatorLimit × Retain.TransitTime/100 并且将输出警告 2000h： • Config.OutputPerOn = FALSE • Config.ActuatorEndStopOn = FALSE • Config.FeedbackOn = FALSE 如果 Config.OutputPerOn = FALSE 且 Config.ActuatorEndStopOn = TRUE 或者 Config.FeedbackOn = TRUE，则仅输出警告 2000h。 如果 Config.OutputPerOn = TRUE，则将不考虑 VirtualActuatorLimit。 自 PID_3Step 版本 2.3 起，可以通过 Config.VirtualActuatorLimit = 0.0 取消激活行程时间的监视和限制。
Config.InputScaling.UpperPointIn	REAL	27648.0	标定的 Input_PER 上限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.InputScaling.LowerPointIn	REAL	0.0	标定的 Input_PER 下限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.InputScaling.UpperPointOut	REAL	100.0	标定的过程值的上限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.InputScaling.LowerPointOut	REAL	0.0	标定的过程值的下限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。

变量	数据类型	默认值	说明
Config.FeedbackScaling.UpperPointIn	REAL	27648.0	标定的 Feedback_PER 上限 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.FeedbackScaling.LowerPointIn	REAL	0.0	标定的 Feedback_PER 下限 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.FeedbackScaling.UpperPointOut	REAL	100.0	上端停止位 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.FeedbackScaling.LowerPointOut	REAL	0.0	下端停止位 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
GetTransitTime.InvertDirection	BOOL	FALSE	如果 InvertDirection = FALSE，则阀门将完全打开、关闭，然后再重新打开，以确定阀门转换时间。 如果 InvertDirection = TRUE，阀门会完全关闭，打开，然后再次关闭。
GetTransitTime.SelectFeedback	BOOL	FALSE	如果 SelectFeedback = TRUE，则转换时间测量中将考虑 Feedback_PER 或 Feedback。 如果 SelectFeedback = FALSE，则转换时间测量中将考虑 Actuator_H 和 Actuator_L。
GetTransitTime.State	INT	0	转换时间测量的当前阶段 • State = 0：未激活 • State = 1：完全打开阀门 • State = 2：完全关闭阀门 • State = 3：将阀门移至目标位置 (NewOutput) • State = 4：成功完成转换时间测量 • State = 5：已取消转换时间测量

变量	数据类型	默认值	说明
GetTransitTime.NewOutput	REAL	0.0	使用位置反馈时转换时间测量的目标位置 目标位置必须介于“上端停止位”和“下端停止位”之间。NewOutput 与 ScaledFeedback 之间的差值必须至少是允许控制范围的 50%。
CycleTime.StartEstimation	BOOL	TRUE	如果 StartEstimation = TRUE，则开始测量 PID_3Step 采样时间。一旦测量完成，CycleTime.StartEstimation = FALSE。
CycleTime.EnEstimation	BOOL	TRUE	如果 EnEstimation = TRUE，则计算 PID_3Step 采样时间。 如果 CycleTime.EnEstimation = FALSE，则不计算 PID_3Step 采样时间，并且您需要手动更正 CycleTime.Value 的组态。
CycleTime.EnMonitoring	BOOL	TRUE	如果 EnMonitoring = TRUE，则监视 PID_3Step 采样时间。如果无法在采样时间内执行 PID_3Step，将输出错误 0800h 并且工作模式将发生更改。 ActivateRecoverMode 和 ErrorBehaviour 可确定切换为哪种工作模式。 如果 EnMonitoring = FALSE，则不会监视 PID_3Step 采样时间，不会输出错误 0800h，也不会切换工作模式。
CycleTime.Value	REAL	0.1	PID_3Step 采样时间（以秒为单位） CycleTime.Value 会自动确定，通常等于调用 OB 的循环时间。
CtrlParamsBackUp.SetByUser	BOOL	FALSE	保存的 Retain.CtrlParams.SetByUser 的值 LoadBackUp = TRUE 时，可以从 CtrlParamsBackUp 结构中重新加载值。
CtrlParamsBackUp.Gain	REAL	1.0	保存的比例增益
CtrlParamsBackUp.Ti	REAL	20.0	保存的积分时间（以秒为单位）
CtrlParamsBackUp.Td	REAL	0.0	保存的微分作用时间（以秒为单位）
CtrlParamsBackUp.TdFiltRatio	REAL	0.2	保存的微分延时系数
CtrlParamsBackUp.PWeighting	REAL	1.0	保存的比例作用权重
CtrlParamsBackUp.DWeighting	REAL	1.0	保存的微分作用权重
CtrlParamsBackUp.Cycle	REAL	1.0	保存的 PID 算法的采样时间（以秒为单位）
CtrlParamsBackUp.InputDeadBand	REAL	0.0	保存的控制偏差的死区宽度

变量	数据类型	默认值	说明
PIDSelfTune.SUT.CalculateParams	BOOL	FALSE	受控系统的属性在调节期间保存。如果 CalculateParams = TRUE，PID 参数都将根据这些属性进行重新计算。将使用 TuneRule 中设置的方法计算 PID 参数。计算后，CalculateParams 将设置为 FALSE。
PIDSelfTune.SUT.TuneRule	INT	1	预调节期间用于计算参数的方法： • SUT.TuneRule = 0：PID 快速 I • SUT.TuneRule = 1：PID 慢速 I • SUT.TuneRule = 2：Chien、Hrones 和 Reswick PID • SUT.TuneRule = 3：Chien、Hrones、Reswick PI • SUT.TuneRule = 4：PID 快速 II • SUT.TuneRule = 5：PID 慢速 II
PIDSelfTune.SUT.State	INT	0	SUT.State 变量指示当前的预调节阶段： • State = 0：初始化预调节 • State = 50：确定无位置反馈的起始位置 • State = 100：计算标准偏差 • State = 200：查找拐点 • State = 300：确定上升时间 • State = 9900：预调节成功 • State = 1：预调节未成功
PIDSelfTune.TIR.RunIn	BOOL	FALSE	利用 RunIn 变量，您可以指定无需预调节也可执行精确调节。 • RunIn = FALSE 在未激活模式或手动模式下启动精确调节时，将启动预调节。 如果精确调节在自动模式下启动，系统将使用现有的 PID 参数来控制设定值。 之后才会启动精确调节。如果无法实现预调节，PID_3Step 将切换到调节开始时的模式。 • RunIn = TRUE 将跳过预调节，PID_3Step 会尝试利用最小或最大输出值达到设定值。这可能会增加超调量。之后才会启动精确调节。 精确调节后，RunIn 将设置为 FALSE。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.CalculateParams	BOOL	FALSE	受控系统的属性在调节期间保存。如果 CalculateParams = TRUE，PID 参数都将根据这些属性进行重新计算。将使用 TuneRule 中设置的方法计算 PID 参数。计算后，CalculateParams 将设置为 FALSE。
PIDSelfTune.TIR.TuneRule	INT	0	精确调节期间用于计算参数的方法： • TIR.TuneRule = 0：PID 自动 • TIR.TuneRule = 1：PID 快速 • TIR.TuneRule = 2：PID 慢速 • TIR.TuneRule = 3：Ziegler-Nichols PID • TIR.TuneRule = 4：Ziegler-Nichols PI • TIR.TuneRule = 5：Ziegler-Nichols P
PIDSelfTune.TIR.State	INT	0	TIR.State 变量指示当前的精确调节阶段： • State = -100：无法进行精确调节。将首先执行预调节。 • State = 0：初始化精确调节 • State = 200：计算标准偏差 • State = 300：尝试利用最大或最小输出值达到设定值 • State = 400：尝试使用现有 PID 参数达到设定值（如果预调节成功） • State = 500：确定波动并计算参数 • State = 9900：精确调节已成功 • State = 1：精确调节未成功
Retain.TransitTime	REAL	30.0	电机转换时间（以秒为单位） 起动驱动器将阀门从关闭状态移至开启状态所需的时间（以秒为单位）。 保持 TransitTime。
Retain.CtrlParams.SetByUser	BOOL	FALSE	如果 SetByUser = FALSE，PID 参数将自动确定并且 PID_3Step 将在输出值中存在死区的情况下运行。死区宽度将在调节期间根据输出值的标准差计算得出并保存到 Retain.CtrlParams.OutputDeadBand 中。 如果 SetByUser = TRUE，PID 参数将手动输入并且 PID_3 Step 将在输出值中不存在死区的情况下运行。 Retain.CtrlParams.OutputDeadBand = 0.0 保持 SetByUser。

变量	数据类型	默认值	说明
Retain.CtrlParams.Gain	REAL	1.0	有效的比例增益 要反转控制逻辑，使用 Config.InvertControl 变量。Gain 上的负值也会反转控制逻辑。我们建议您仅使用 InvertControl 设置控制逻辑。如果 InvertControl = TRUE 且 Gain < 0.0，则控制逻辑也会反转。 保持 Gain。
Retain.CtrlParams.Ti	REAL	20.0	• Ti > 0.0：有效的积分时间（以秒为单位） • Ti = 0.0：积分作用取消激活 保持 Ti。
Retain.CtrlParams.Td	REAL	0.0	• Td > 0.0：有效的微分作用时间（以秒为单位） • Td = 0.0：微分作用取消激活 保持 Td。
Retain.CtrlParams.TdFiltRatio	REAL	0.2	有效的微分延时系数 微分延迟系数用于延迟微分作用的生效。 微分延迟 = 微分作用时间 × 微分延迟系数 • 0.0：微分作用仅在一个周期内有效，因此几乎不产生影响。 • 0.5：此值经实践证明对于具有一个优先时间常量的受控系统非常有用。 • > 1.0：系数越大，微分作用的生效时间延迟越久。 保持 TdFiltRatio。
Retain.CtrlParams.PWeighting	REAL	1.0	有效的比例作用权重 比例作用随着设定值的变化而减弱。 允许使用 0.0 到 1.0 之间的值。 • 1.0：应对设定值变化的比例作用完全有效 • 0.0：应对设定值变化的比例作用无效 当过程值变化时，比例作用始终完全有效。 保持 PWeighting。

变量	数据类型	默认值	说明
Retain.CtrlParams.DWeighting	REAL	1.0	有效的微分作用权重 微分作用随着设定值的变化而减弱。 允许使用 0.0 到 1.0 之间的值。 • 1.0：设定值变化时微分作用完全有效 • 0.0：设定值变化时微分作用不生效 当过程值变化时，微分作用始终完全有效。 保持 DWeighting。
Retain.CtrlParams.Cycle	REAL	1.0	PID 算法的有效采样时间（以秒为单位），舍入为调用 OB 的循环时间的整数倍。 保持 Cycle。
Retain.CtrlParams.InputDeadBand	REAL	0.0	控制偏差的死区宽度 保持 InputDeadBand。

说明

请在“未激活”模式下更改本表列出的变量，以防 PID 控制器出现故障。

参见

模式 V2 的参数状态 (页 372)

变量 ActivateRecoverMode V2 (页 383)

将工艺对象下载到设备 (页 76)

8.2.4.8 模式 V2 的参数状态

参数的相关性

State 参数显示了 PID 控制器的当前工作模式。 您无法更改 State 参数。

当 ModeActivate 出现上升沿时，PID_3Step 将切换到保存在 Mode 输入/输出参数中的工作模式。

CPU 启动或从 Stop 切换为 RUN 模式时，PID_3Step 将以保存在 Mode 参数中的工作模式启动。要使 PID_3Step 保持在“未激活”模式下，应设置 RunModeByStartup = FALSE。

值的含义

State	工作模式说明
0	未激活 控制器关闭，且不再更改阀门位置。
1	预调节 预调节可确定对输出值脉冲的过程响应，并搜索拐点。 根据受控系统的最大上升速率与死时间计算 PID 参数。 可在执行预调节和精确调节时获得最佳 PID 参数。
	预调节的要求： • 已对电机转换时间进行了组态或测量。 • 未激活 (State = 0)、手动模式 (State = 4) 或自动模式 (State = 3) • ManualEnable = FALSE • Reset = FALSE • 设定值和过程值均在组态的限值范围内。 过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。 只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。 最可能的情况是处于工作模式“未激活”和“手动模式”下。
1	设定值在变量 CurrentSetpoint 中冻结。 出现以下情况时，调节将取消： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel 重新计算 PID 参数之前将对其进行备份并且可使用 LoadBackUp 重新激活这些参数。 预调节成功后，控制器将切换到自动模式。 如果预调节未成功，则工作模式的切换取决于 ActivateRecoverMode 和 ErrorBehaviour。 预调节阶段通过 SUT.State 变量来指示。

State	工作模式说明
2	精确调节 精确调节将使过程值出现恒定受限的振荡。 根据该振荡的幅度和频率重新计算 PID 参数。 精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。 可在执行预调节和精确调节时获得最佳 PID 参数。 PID_3Step 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。 设定值在变量 CurrentSetpoint 中冻结。 出现以下情况时，调节将取消： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel
	精确调节前会备份 PID 参数。 可以使用 LoadBackUp 重新激活这些参数。 精确调节的要求： • 已对电机转换时间进行了组态或测量。 • 设定值和过程值均在组态的限值范围内。 • ManualEnable = FALSE • Reset = FALSE • 自动模式 (State = 3)、未激活模式 (State = 0) 或手动模式 (State = 4)
2	在以下模式下启动精确调节时，具体情况如下所述： • 自动模式 (State = 3) 如果希望通过调节来改进现有 PID 参数，请在自动模式下启动精确调节。 PID_3Step 将使用现有的 PID 参数控制系统，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。 • 未激活模式 (State = 0) 或手动模式 (State = 4) 如果满足预调节的要求，则启动预调节。 已确定的 PID 参数将用于控制，直到控制回路已稳定并且精确调节的要求得到满足为止。 如果 PIDSelfTune.TIR.RunIn = TRUE，则将跳过预调节，并将尝试利用最小或最大输出值来达到设定值。 这可能会增加超调量。 随后将自动启动精确调节。 精确调节成功后，控制器将切换到自动模式。 如果精确调节未成功，则工作模式的切换取决于 ActivateRecoverMode 和 ErrorBehaviour。 精确调节阶段使用 TIR.State 变量来指示。

State	工作模式说明
3	自动模式 在自动模式下，PID_3Step 会按照指定的参数来控制受控系统。 如果满足下列要求之一，则控制器将切换到自动模式： • 预调节成功完成 • 精确调节成功完成 • Mode 输入/输出参数更改为值 3 并且 ModeActivate 出现上升沿。 从自动模式到手动模式的切换只有在调试编辑器中执行时，才是无扰动的。 自动模式下会考虑 ActivateRecoverMode 变量。
4	手动模式 在手动模式下，在 Manual_UP 和 Manual_DN 参数或 ManualValue 参数中指定手动输出值。在发生错误时执行器是否可移动到输出值的情况将在 ErrorBits 参数中说明。 还可以使用 ManualEnable = TRUE 来激活该工作模式。 建议只使用 Mode 和 ModeActivate 更改工作模式。 从手动模式到自动模式的切换是无扰动的。 错误未决时也可使用手动模式。
5	逼近替代输出值 如果 Errorbehaviour = TRUE 且 ActivateRecoverMode = FALSE.，则出现错误时会激活该工作模式。 PID_3Step 将执行器移动到替代输出值位置，然后更改为“未激活”模式。
6	转换时间测量 电机将阀门从闭合状态完全打开的所需时间已确定。 当设置 Mode = 6 和 ModeActivate = TRUE 时，将激活此工作模式。 如果使用停止位信号测量转换时间，则阀门将从当前位置完全打开、完全关闭然后再次完全打开。 如果 GetTransitTime.InvertDirection = TRUE，将反转此行为。 如果使用位置反馈测量转换时间，那么会将执行器从其当前位置移至目标位置。 测量转换时间期间，不考虑输出值的限值。 执行器可行进至上端停止位或下端停止位。

State	工作模式说明
7	错误监视 控制算法关闭，并且不再更改阀门的位置。 出现错误时会激活该工作模式而不激活“未激活”模式。 必须满足以下所有条件： • 自动模式 (Mode = 3) • Errorbehaviour = FALSE • ActivateRecoverMode = TRUE • 已出现一个或多个错误，并且 ActivateRecoverMode (页 383) 生效。 当错误不再处于未决状态时，PID_3Step 切换回自动模式。
8	在监视错误的同时逼近替代输出值 出现错误时将激活该工作模式，而不是“逼近替代输出值”模式。PID_3Step 会将执行器移动到替代输出值，然后切换到“错误监视”模式。 必须满足以下所有条件： • 自动模式 (Mode = 3) • Errorbehaviour = TRUE • ActivateRecoverMode = TRUE • 已出现一个或多个错误，并且 ActivateRecoverMode (页 383) 生效。 当错误不再处于未决状态时，PID_3Step 切换回自动模式。
10	无停止位信号的手动模式 即使 Config.ActuatorEndStopOn = TRUE，也**不会**考虑停止位信号。 输出值的限值将不予考虑。 否则，PID_3Step 将与手动模式下的行为相同。

ENO 特性

如果 State = 0，那么 ENO = FALSE。

如果 State ≠ 0，那么 ENO = TRUE。

在调试期间自动切换工作模式

预调节或精确调节成功后，将激活自动模式。 下表显示了成功预调节期间 Mode 和 State 的更改方式。

周期编号	Mode	State	操作
0	4	4	设置 Mode = 1
1	1	4	设置 ModeActivate = TRUE
1	4	1	State 的值保存在模式参数中 启动预调节功能
n	4	1	预调节成功完成
n	3	3	启动自动模式

PID_3Step 将在出现错误时自动切换工作模式。 下表显示了出现错误的预调节期间 Mode 和 State 的更改方式。

周期编号	Mode	State	操作
0	4	4	设置 Mode = 1
1	1	4	设置 ModeActivate = TRUE
1	4	1	State 的值保存在模式参数中 启动预调节功能
n	4	1	取消预调节
n	4	4	启动手动模式

如果 ActivateRecoverMode = TRUE，将激活保存在 Mode 参数中的工作模式。 在开始转换时间测量、预调节或精确调节时，PID_3Step 已将 State 的值保存在 Mode 输入/输出参数中。因此 PID_3Step 会切换到转换时间测量开始时或调节开始时的工作模式。

如果 ActivateRecoverMode = FALSE，将激活“未激活”或“逼近替代输出值”模式。

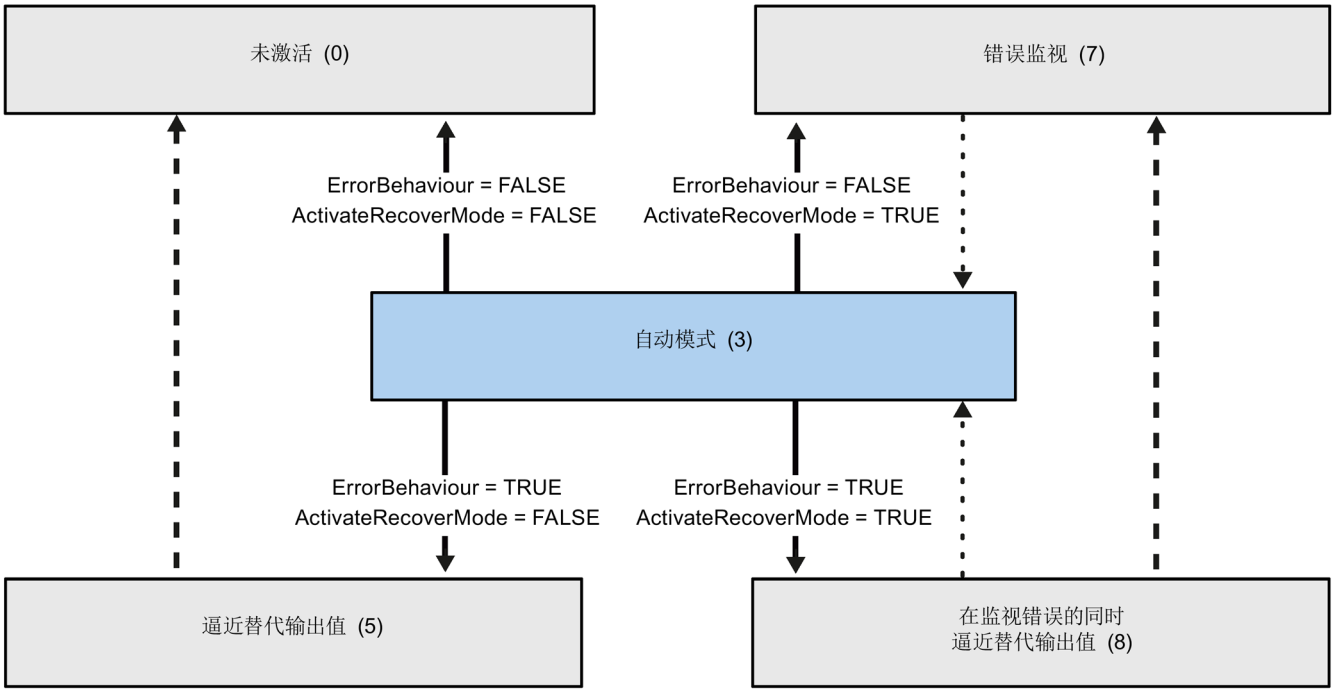
测量转换时间后自动切换工作模式

如果 ActivateRecoverMode = TRUE，在成功测量转换时间后，将激活保存在 Mode 参数中的工作模式。

如果 ActivateRecoverMode = FALSE，在成功测量转换时间后，系统将切换到“未激活”工作模式。

在自动模式中自动切换工作模式

PID_3Step 将在出现错误时自动切换工作模式。 下图说明了 ErrorBehaviour 和 ActivateRecoverMode 对工作模式切换的影响。



出现错误时自动切换工作模式。

完成当前操作后自动切换工作模式。

当错误不再处于未决状态时，自动切换工作模式。

参见

变量 ActivateRecoverMode V2 (页 383)

参数 ErrorBits V2 (页 379)

8.2.4.9 参数 ErrorBits V2

如果多个错误同时处于待决状态，将通过二进制加法显示 ErrorBits 的值。例如，显示 ErrorBits = 0003h 表示错误 0001h 和 0002h 同时处于待决状态。

如果存在位置反馈，则 PID_3Step 使用 ManualValue 作为手动模式下的输出值。Errorbits = 10000h 除外。

ErrorBits (DW#16#...)	说明
0000	没有任何错误。
0001	参数“Input”超出了过程值限值的范围。 • Input > Config.InputUpperLimit 或 • Input < Config.InputLowerLimit 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_3Step 保持自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。
0002	参数“Input_PER”的值无效。请检查模拟量输入是否有处于未决状态的错误。 如果在错误发生之前自动模式已激活并且 ActivateRecoverMode = TRUE，则 PID_3Step 将切换到“在监视错误的同时逼近替代输出值”或“错误监视”模式。当错误不再处于未决状态时，PID_3Step 切换回自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。
0004	精确调节期间出错。过程值无法保持振荡状态。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_3Step 取消调节并切换到 Mode 参数中保存的工作模式。
0010	调节期间设定值发生更改。 可在 CancelTuningLevel 变量中设置允许的设定值波动。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_3Step 取消调节并切换到 Mode 参数中保存的工作模式。
0020	精确调节期间不允许预调节。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_3Step 保持在精确调节模式。

ErrorBits (DW#16#...)	说明
0080	预调节期间出错。输出值限值的组态不正确。 检查输出值的限值是否已正确组态及其是否匹配控制逻辑。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_3Step 取消调节并切换到 Mode 参数中保存的工作模式。
0100	精确调节期间的错误导致生成无效参数。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_3Step 取消调节并切换到 Mode 参数中保存的工作模式。
0200	参数“Input”的值无效：值的数字格式无效。 如果在错误发生之前自动模式已激活并且 ActivateRecoverMode = TRUE，则 PID_3Step 将切换到“在监视错误的同时逼近替代输出值”或“错误监视”模式。当错误不再处于未决状态时，PID_3Step 切换回自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。
0400	输出值计算失败。请检查 PID 参数。 如果在错误发生之前自动模式已激活并且 ActivateRecoverMode = TRUE，则 PID_3Step 将切换到“在监视错误的同时逼近替代输出值”或“错误监视”模式。当错误不再处于未决状态时，PID_3Step 切换回自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。
0800	采样时间错误：在循环中断 OB 的采样时间内没有调用 PID_3Step。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_3Step 保持自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。 如果在使用 PLCSIM 进行仿真期间出现该错误，请参见使用 PLCSIM 仿真 PID_3Step V2 (页 149)下的说明。

ErrorBits (DW#16#...)	说明
1000	参数“Setpoint”的值无效：值的数字格式无效。 如果在错误发生之前自动模式已激活并且 ActivateRecoverMode = TRUE，则 PID_3Step 将切换到“在监视错误的同时逼近替代输出值”或“错误监视”模式。当错误不再处于未决状态时，PID_3Step 切换回自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。
2000	Feedback_PER 参数的值无效。 请检查模拟量输入是否有处于未决状态的错误。 执行器无法移动到替代输出值，并且将保持当前位置。在手动模式下，仅可通过 Manual_UP 和 Manual_DN 更改执行器的位置，而不可通过 ManualValue 更改。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。
4000	Feedback 参数的值无效。值的数字格式无效。 执行器无法移动到替代输出值，并且将保持当前位置。在手动模式下，仅可通过 Manual_UP 和 Manual_DN 更改执行器的位置，而不可通过 ManualValue 更改。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。
8000	数字位置反馈出现错误。Actuator_H = TRUE 和 Actuator_L = TRUE。 执行器无法移动到替代输出值，并且将保持当前位置。此状态下无法进入手动模式。 为了从此状态移动执行器，必须取消激活“执行器停止位”(Config.ActuatorEndStopOn = FALSE) 或者切换到无停止位信号的手动模式 (Mode = 10)。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。 如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE 已激活，则 PID_3Step 将切换到保存在 Mode 参数中的工作模式。

ErrorBits (DW#16#...)	说明
10000	ManualValue 参数的值无效。值的数字格式无效。 执行器无法移动到手动值，并且将保持当前位置。 在 ManualValue 中指定一个有效值或者在手动模式下通过 Manual_UP 和 Manual_DN 移动执行器。
20000	变量 SavePosition 的值无效。值的数字格式无效。 执行器无法移动到替代输出值，并且将保持当前位置。
40000	Disturbance 参数的值无效。值的数字格式无效。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 Disturbance 将设置为零。PID_3Step 保持自动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_3Step 切换到 Mode 参数中保存的工作模式。如果当前阶段中的 Disturbance 对输出值无影响，则不会取消调节。 转换时间测量期间错误没有影响。

8.2.4.10 变量 ActivateRecoverMode V2

ActivateRecoverMode 变量确定错误响应方式。 Error 参数指示是否存在错误处于未决状态。 当错误不再处于未决状态时，Error = FALSE。 ErrorBits 参数显示发生的具体错误。

注意
您的系统可能已损坏。 如果 ActivateRecoverMode = TRUE，则 PID_3Step 保持自动模式，即使超过过程值的限值。 这可能损坏您的系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

自动模式

ActivateRecoverMode	说明
FALSE	出现错误时，PID_3Step 将切换到“未激活”模式或“逼近替代输出值”模式。 只能通过 Reset 的下降沿或 ModeActivate 的上升沿激活控制器。
TRUE	如果在自动模式下频繁出现错误，则该设置会对控制响应产生负面影响，这是由于发生每个错误时，PID_3Step 在计算的输出值和替代输出值之间切换导致。 这种情况下，检查 ErrorBits 参数并消除错误原因。 如果发生一个或多个下列错误，则 PID_3Step 停留在自动模式下： • 0001h： 参数“Input”超出了过程值限值的范围。 • 0800h： 采样时间错误 • 40000h：Disturbance 参数的值无效。 如果发生一个或更多以下错误，PID_3Step 将切换到“在监视错误的同时逼近替代输出值”模式或“错误监视”模式： • 0002h： Input_PER 参数的值无效。 • 0200h： Input 参数的值无效。 • 0400h： 输出值计算失败。 • 1000h： Setpoint 参数的值无效。

ActivateRecoverMode	说明
TRUE	如果发生一个或多个下列错误，则 PID_3Step 将不再移动执行器： • 2000h： Feedback_PER 参数的值无效。 • 4000h： Feedback 参数的值无效。 • 8000h： 数字位置反馈期间出错。 • 20000h： 变量 SavePosition 的值无效。 值的数字格式无效。 该特性与 ErrorBehaviour 无关。 当错误不再处于未决状态时，PID_3Step 切换回自动模式。

预调节、精确调节和转换时间测量

ActivateRecoverMode	说明
FALSE	出现错误时，PID_3Step 将切换到“未激活”模式或“逼近替代输出值”模式。 只能通过 Reset 的下降沿或 ModeActivate 的上升沿激活控制器。 在成功测量转换时间后，控制器更改为“未激活”模式。
TRUE	如果发生下列错误，PID_3Step 将保持在激活模式： • 0020h： 精确调节期间不允许预调节。 以下错误将被忽略： • 10000h： ManualValue 参数的值无效。 • 20000h： 变量 SavePosition 的值无效。 出现其它错误时，PID_3Step 将取消调节并切换到调节开始时的模式。

手动模式

手动模式下 ActivateRecoverMode 无效。

参见

PID_3Step V2 的静态变量 (页 361)

模式 V2 的参数状态 (页 372)

8.2.4.11 变量 Warning V2

如果多个警告同时处于待决状态，将通过二进制加法显示它们的值。 例如，显示警告 0005h 表示警告 0001h 和 0004h 同时处于待决状态。

Warning (DW#16#...)	说明
0000	无警告处于待决状态。
0001	预调节期间未发现拐点。
0004	设定值被限制为组态的限值。
0008	在所选计算方法中未定义所有必要的受控系统属性。 而是使用 TIR.TuneRule = 3 方法计算 PID 参数。
0010	由于 Reset = TRUE 或 ManualEnable = TRUE，无法更改工作模式。
0020	调用 OB 的循环时间会限制 PID 算法的采样时间。 通过缩短 OB 循环时间来改进结果。
0040	过程值超出其警告限值之一。
0080	Mode 的值无效。 工作模式不变。
0100	手动值被限制为控制器输出的限值。
0200	不支持指定的调节规则。 不计算任何 PID 参数。
0400	由于执行器设置与所选的测量方法不匹配，无法测量转换时间。
0800	当前位置与新输出值之差太小，无法用于转换时间测量。 这可能产生错误结果。 当前输出值与新输出值之差必须至少是整个控制范围的 50%。
1000	无法达到替代输出值，因为它超出了输出值限值。
2000	执行器已在一个方向上移动超过 Config.VirtualActuatorLimit × Retain.TransitTime。 检查执行器是否已达到停止位信号。

以下警告在消除问题的原因后即被删除：

- 0001h
- 0004h
- 0008h
- 0040h
- 0100h
- 2000h

所有其它警告均在 Reset 或 ErrorAck 出现上升沿时清除。

8.2.5 PID_3Step V1

8.2.5.1 PID_3Step V1 说明

说明

使用 PID_3Step 指令可对具有阀门自调节的 PID 控制器或具有积分行为的执行器进行组态。

存在下列工作模式：

- 未激活
- 预调节
- 精确调节
- 自动模式
- 手动模式
- 逼近替代输出值
- 转换时间测量
- 在监视错误的同时逼近替代输出值
- 错误监视

有关工作模式的详细信息，请参见 State 参数。

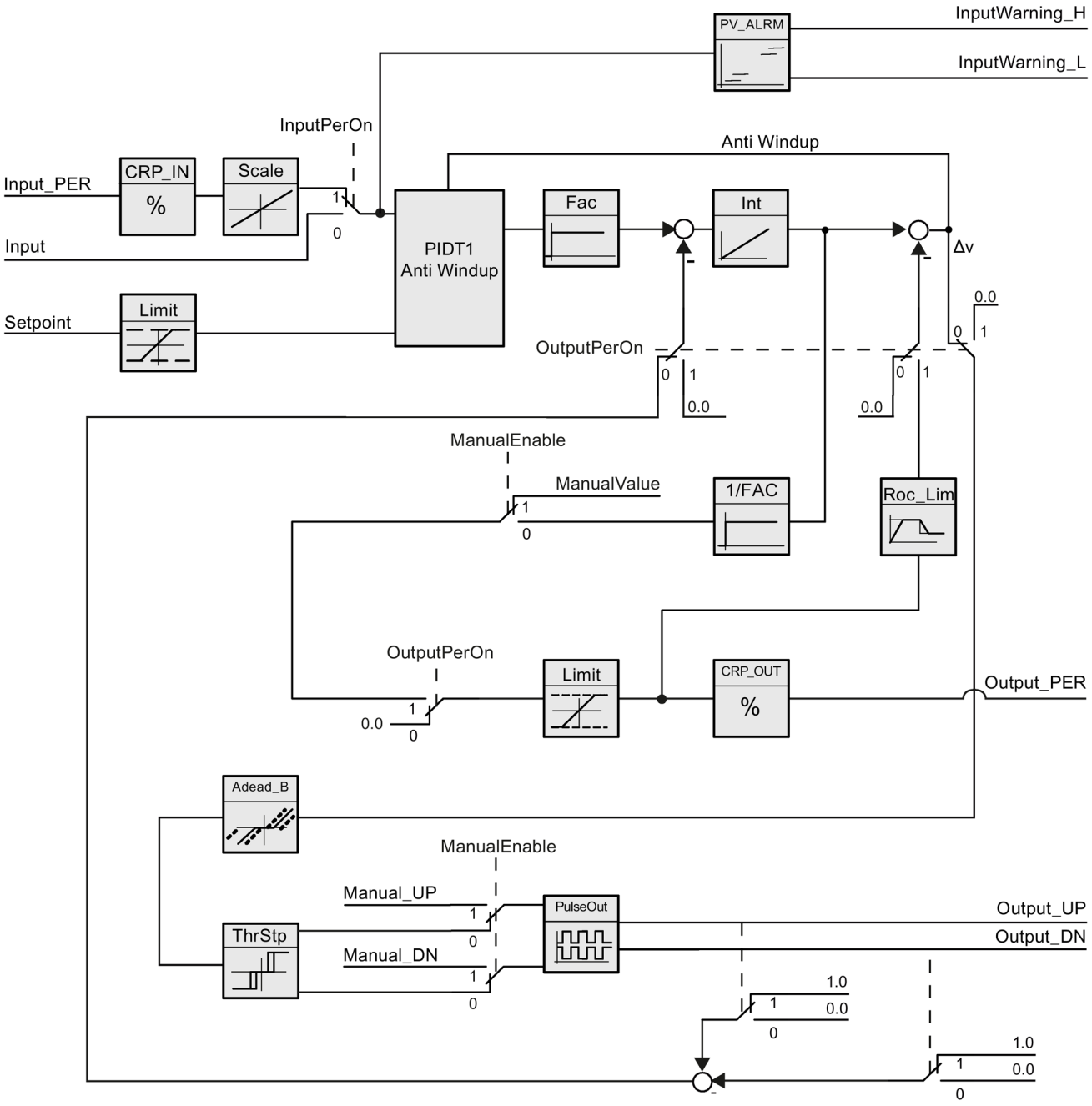
PID 算法

PID_3Step 是一种具有抗积分饱和功能并且能够对比例作用和微分作用进行加权的 PIDT1 控制器。 采用以下方程来计算输出值。

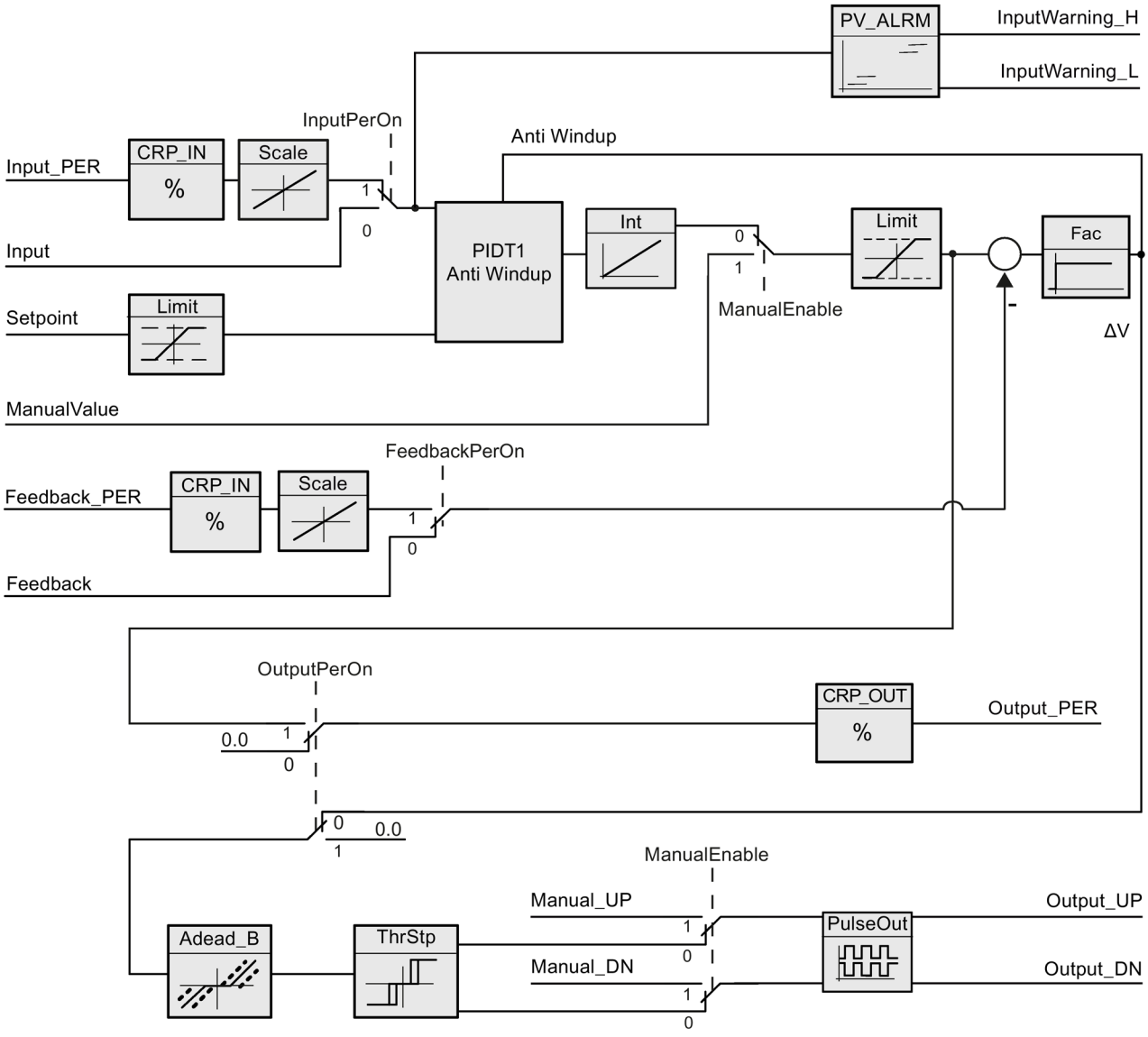
$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

符号	说明
y	输出值
K_p	比例增益
s	拉普拉斯运算符
b	比例作用权重
w	设定值
x	过程值
T_I	积分作用时间
a	微分延迟系数 ($T1 = a \times T_D$)
T_D	微分作用时间
c	微分作用权重

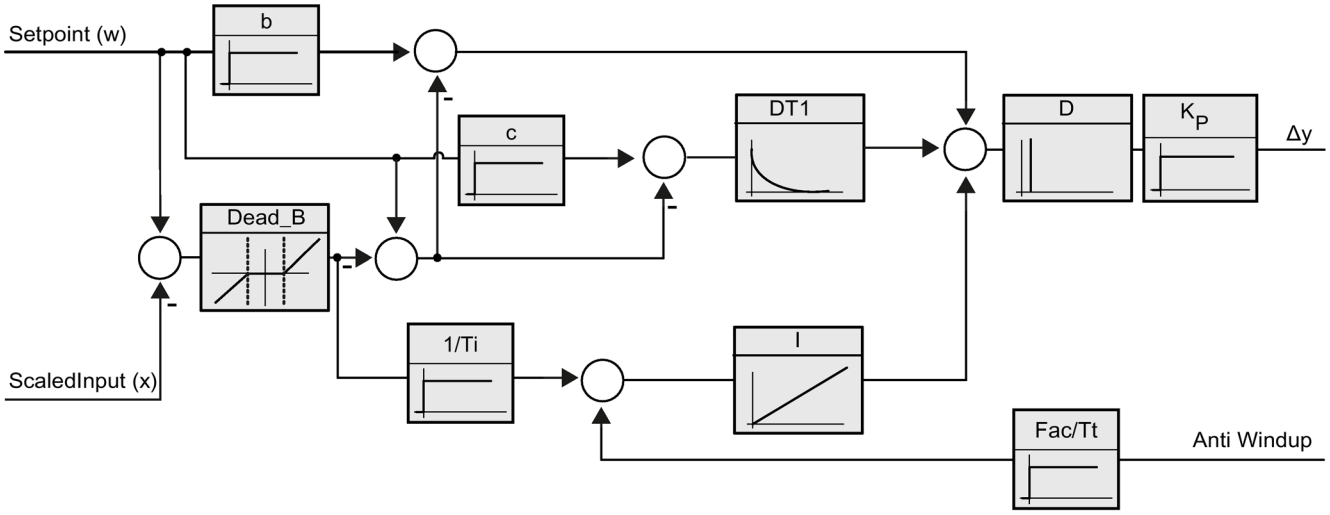
不带位置反馈的方框图


PV_ALRM
InputWarning_H
InputWarning_L
InputPerOn
Anti Windup
Input_PER
CRP_IN
%
Scale
Input
PIDT1
Anti Windup
Fac
Int
Δv
Setpoint
Limit
OutputPerOn
ManualEnable
ManualValue
1/FAC
Roc_Lim
OutputPerOn
Limit
CRP_OUT
%
Output_PER
Adead_B
ManualEnable
Manual_UP
Manual_DN
ThrStp
PulseOut
Output_UP
Output_DN

带位置反馈的方框图


PV_ALRM
InputWarning_H
InputWarning_L
InputPerOn
Anti Windup
Input_PER
CRP_IN
%
Scale
1
0
Input
PIDT1
Anti Windup
Int
0
1
Limit
Fac
Setpoint
Limit
ManualEnable
ΔV
ManualValue
FeedbackPerOn
Feedback_PER
CRP_IN
%
Scale
1
0
Feedback
OutputPerOn
CRP_OUT
%
Output_PER
0.0
1
0
0
0.0
1
ManualEnable
Manual_UP
Output_UP
Adead_B
ThrStp
PulseOut
Manual_DN
Output_DN
1
0
1
0

带抗积分饱和的 PIDT1 的方框图



调用

以调用 OB 的循环时间的恒定时间间隔（最好在循环中断 OB 中）调用 PID_3Step。

下载到设备

仅当完全下载 PID_3Step 后，才能更新保持性变量的实际值。

将工艺对象下载到设备 (页 76)

启动

CPU 启动时，PID_3Step 以上次激活的操作模式启动。 要使 PID_3Step 保持在“未激活”模式下，应设置 RunModeByStartup = FALSE。

对错误的响应

如果出现错误，将在 Error 参数中输出。 使用 ErrorBehaviour 和 ActivateRecoverMode 变量组态 PID_3Step 的响应。

ErrorBehaviour	ActivateRecoverMode	执行器设置组态 将 Output 设置为	响应
0	FALSE	当前输出值	切换到“未激活”模式 (Mode = 0)
0	TRUE	错误未决时的当前输出值	切换到“错误监视”模式 (Mode = 7)
1	FALSE	替代输出值	切换到“逼近替代输出值”模式 (Mode = 5) 切换到“未激活”模式 (Mode = 0)
1	TRUE	错误未决时的替代输出值	切换到“在监视错误的同时逼近替代输出值”模式 (Mode = 8) 切换到“错误监视”模式 (Mode = 7)

ErrorBits 参数显示发生的具体错误。

参见

参数 State 和 Retain.Mode V1 (页 411)

参数 ErrorBits V1 (页 419)

组态 PID_3Step V1 (页 150)

8.2.5.2 PID_3Step V1 工作原理

监视过程值的限值

在 Config.InputUpperLimit 和 Config.InputLowerLimit 变量中指定过程值的上限和下限。如果过程值超出这些限值，将出现错误 (ErrorBits = 0001hex)。

在 Config.InputUpperWarning 和 Config.InputLowerWarning 变量中指定过程值的警告上限和警告下限。如果过程值超出这些警告限值，将发生警告 (Warnings = 0040hex)，并且 InputWarning_H 或 InputWarning_L 输出参数会更改为 TRUE。

限制设定值

可在 Config.SetpointUpperLimit 和 Config.SetpointLowerLimit 变量中指定设定值的上限和下限。PID_3Step 会自动将设定值限制在过程值的限值范围内。可以将设定值限制在更小的范围内。PID_3Step 会检查此范围是否处于过程值的限值范围内。如果设定值超出这些限值，上限和下限将用作设定值，并且输出参数 SetpointLimit_H 或 SetpointLimit_L 将设置为 TRUE。

在所有操作模式下均限制设定值。

限制输出值

在 Config.OutputUpperLimit 变量和 Config.OutputLowerLimit 变量中指定输出值的上限和下限。输出值的限值必须位于“下端停止位”和“上端停止位”范围内。

- 上端停止位：Config.FeedbackScaling.UpperPointOut
- 下端停止位：Config.FeedbackScaling.LowerPointOut

规则：

UpperPointOut ≥ OutputUpperLimit > OutputLowerLimit ≥ LowerPointOut

“上端停止位”和“下端停止位”的有效值取决于：

- FeedbackOn
- FeedbackPerOn
- OutputPerOn

OutputPerOn	FeedbackOn	FeedbackPerOn	LowerPointOut	UpperPointOut
FALSE	FALSE	FALSE	无法设置 (0.0%)	无法设置 (100.0%)
FALSE	TRUE	FALSE	-100.0% 或 0.0%	0.0% 或 +100.0%
FALSE	TRUE	TRUE	-100.0% 或 0.0%	0.0% 或 +100.0%
TRUE	FALSE	FALSE	无法设置 (100.0%)	无法设置 (100.0%)
TRUE	TRUE	FALSE	-100.0% 或 0.0%	0.0% 或 +100.0%
TRUE	TRUE	TRUE	-100.0% 或 0.0%	0.0% 或 +100.0%

如果 OutputPerOn = FALSE 且 FeedbackOn = FALSE，则无法限制输出值。当 Actuator_H = TRUE 或 Actuator_L = TRUE 时，或者在行进时间达到电机转换时间的 110% 后，数字量输出将复位。

输出值在 100% 时为 27648，在 -100% 时为 -27648。PID_3Step 必须能够完全关闭阀门。因此，输出值的限值范围内必须包括零。

说明

与两个或多个执行器结合使用

PID_3 Step 不适合与两个或多个执行器结合使用（例如，在加热/制冷应用中），因为不同的执行器需要不同的 PID 参数以实现良好的控制响应。

替代输出值

如果出现错误，PID_3Step 可输出一个替代输出值并将执行器移至变量 SavePosition 中指定的安全位置。替代输出值必须处于输出值的限值范围内。

监视信号有效性

监视以下参数值的有效性：

- Setpoint
- Input
- Input_PER
- Feedback
- Feedback_PER
- Output

监视 PID_3Step 采样时间

理想情况下，采样时间等于调用 OB 的周期时间。PID_3Step 指令测量两次调用之间的时间间隔。这就是当前采样时间。每次切换工作模式以及初始启动期间，平均值由前 10 个采样时间构成。当前采样时间与该平均值之间的差值过大时会触发错误 (ErrorBits = 0800 hex)。

在下列条件下，PID_3Step 在调节期间将设置为“未激活”模式：

- 新平均值 >= 1.1 x 原平均值
- 新平均值 <= 0.9 x 原平均值

在自动模式下，PID_3Step 在下列条件下将设置为“未激活”模式：

- 新平均值 >= 1.5 x 原平均值
- 新平均值 <= 0.5 x 原平均值

PID 算法的采样时间

受控系统需要一定的时间来对输出值的变化做出响应。因此，建议不要在每次循环中都计算输出值。PID 算法的采样时间是两次计算输出值之间的时间。该时间在调节期间进行计算，并舍入为循环时间的倍数。PID_3Step 的所有其它功能在每次调用时均执行。

测量电机转换时间

电机转换时间指的是电机将执行器从关闭状态转为开启状态所需的时间（以秒为单位）。执行器在一个方向上移动的最长时间是电机转换时间的 110%。PID_3Step 要求电机转换时间尽可能准确，以便获得良好的控制器结果。执行器文档中的数据包含此类执行器的平均值。针对特定执行器的值可能不同。可以在调试期间测量电机转换时间。测量电机转换时间期间，不考虑输出值的限值。执行器可行进至上端停止位或下端停止位。

在计算模拟量输出值和数字量输出值时，会将电机转换时间考虑在内。自动调节和抗饱和行为期间，需要该时间来确保正常运行。因此，应该将电机转换时间组态为电机将执行器从关闭状态转换至开启状态所需的值。

如果相关电机转换时间并未影响过程（如使用电磁阀），因此输出值直接且完全影响过程，则使用 PID_Compact。

控制逻辑

通常，可通过增大输出值来增大过程值。这种做法称为常规控制逻辑。对于制冷和放电控制系统，可能需要反转控制逻辑。PID_3Step 不使用负比例增益。如果 InvertControl = TRUE，则不断增大的控制偏差将导致输出值减小。在预调节和精确调节期间还会考虑控制逻辑。

参见

组态 PID_3Step V1 (页 150)

8.2.5.3 PID_3Step V1 输入参数

表格 8- 10

参数	数据类型	默认值	说明
Setpoint	REAL	0.0	PID 控制器在自动模式下的设定值
Input	REAL	0.0	用户程序的变量用作过程值的源。 如果正在使用参数 Input，则必须设置 Config.InputPerOn = FALSE。
Input_PER	WORD	W#16#0	模拟量输入用作过程值的源。 如果正在使用参数 Input_PER，则必须设置 Config.InputPerOn = TRUE。
Actuator_H	BOOL	FALSE	阀门处于上端停止位时的数字位置反馈 如果 Actuator_H = TRUE，表明阀门处于上端停止位，并且不再向此方向移动。
Actuator_L	BOOL	FALSE	阀门处于下端停止位时的数字位置反馈 如果 Actuator_L = TRUE，表明阀门处于下端停止位，并且不再向此方向移动。
Feedback	REAL	0.0	阀门的位置反馈 如果正在使用参数 Feedback，则必须设置 Config.FeedbackPerOn = FALSE。
Feedback_PER	WORD	W#16#0	阀门的模拟位置反馈 如果正在使用参数 Feedback_PER，则必须设置 Config.FeedbackPerOn = TRUE。 根据以下变量标定 Feedback_PER： • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut

参数	数据类型	默认值	说明
ManualEnable	BOOL	FALSE	• 出现 FALSE -> TRUE 沿时会选择“手动模式”，而 State = 4，Retain.Mode 保持不变。 • 出现 TRUE -> FALSE 沿时会选择最近激活的操作模式 Retain.Mode 的变化在 ManualEnable = TRUE 期间不会生效。仅在 ManualEnable 处出现 TRUE -> FALSE 沿时 Retain.Mode 的变化才会生效。 **PID_3Step V1.1** 如果 CPU 启动时 ManualEnable = TRUE，则 PID_3Step 以手动模式启动。并非一定需要 ManualEnable 出现上升沿 (FALSE -> TRUE) 时，才会执行上述操作。 **PID_3Step V1.0** CPU 启动时，PID_3Step 仅在 ManualEnable 出现上升沿 (FALSE->TRUE) 时才切换到手动模式。没有上升沿时，PID_3Step 在 ManualEnable 为 FALSE 的上一个工作模式下启动。
ManualValue	REAL	0.0	在手动模式下指定阀门的绝对位置。仅当使用 OutputPer **或**位置反馈可用时，才会对 ManualValue 进行评估。
Manual_UP	BOOL	FALSE	在手动模式下，每次出现上升沿时，阀门都将打开总控制范围的 5%，或者持续打开最短的电机转换时间。仅当未使用 Output_PER 且没有位置反馈可用时，才会对 Manual_UP 进行评估。

参数	数据类型	默认值	说明
Manual_DN	BOOL	FALSE	在手动模式下，每次出现上升沿时，阀门都将关闭总控制范围的 5%，或者持续关闭最短的电机转换时间。仅当未使用 Output_PER 且没有位置反馈可用时，才会对 Manual_DN 进行评估。
Reset	BOOL	FALSE	重新启动控制器。 • FALSE -> TRUE 沿 – 切换到“未激活”模式 – 复位 ErrorBits 和警告 – 复位中间控制器值 （保留 PID 参数） • TRUE -> FALSE 沿 – 切换到最近激活的模式 – 如果之前已激活自动模式，则会以无扰动的方式切换至自动模式。

8.2.5.4 PID_3Step V1 输出参数

表格 8- 11

参数	数据类型	默认值	说明
ScaledInput	REAL	0.0	标定的过程值
ScaledFeedback	REAL	0.0	标定的位置反馈 对于没有位置反馈的执行器，由 ScaledFeedback 指示的执行器位置非常不精确。这种情况下，ScaledFeedback 只可用于粗略估计当前位置。
Output_UP	BOOL	FALSE	用于打开阀门的数字量输出值 如果 Config.OutputPerOn = FALSE，则使用参数 Output_UP。
Output_DN	BOOL	FALSE	用于关闭阀门的数字量输出值 如果 Config.OutputPerOn = FALSE，则使用参数 Output_DN。
Output_PER	WORD	W#16#0	模拟量输出值 如果 Config.OutputPerOn = TRUE，则使用 Output_PER。 如果将一个阀门用作通过模拟量输出进行触发并使用连续信号（例如，0...10 V 或 4...20 mA）进行控制的执行器，则使用 Output_PER。 Output_PER 的值与阀门的目标位置相对应，例如，当阀门打开 50% 时 Output_PER = 13824。
SetpointLimit_H	BOOL	FALSE	如果 SetpointLimit_H = TRUE，则说明达到了设定值的绝对上限。在 CPU 中，该设定值被限制为所组态的设定值的绝对上限。设定值的上限默认设置为所组态的过程值的绝对上限。 如果组态的 Config.SetpointUpperLimit 值位于过程值的限值范围内，则该值将用作设定值的上限。
SetpointLimit_L	BOOL	FALSE	如果 SetpointLimit_L = TRUE，则说明已达到设定值的绝对下限。在 CPU 中，该设定值被限制为所组态的设定值的绝对下限。设定值的下限将默认设置为所组态的过程值的绝对下限。 如果组态的 Config.SetpointLowerLimit 值位于过程值的限值范围内，则该值将用作设定值的下限。

参数	数据类型	默认值	说明
InputWarning_H	BOOL	FALSE	如果 InputWarning_H = TRUE，则说明过程值已达到或超出警告上限。
InputWarning_L	BOOL	FALSE	如果 InputWarning_L = TRUE，则说明过程值已经达到或低于警告下限。
State	INT	0	State 参数 (页 411)显示 PID 控制器的当前操作模式。使用 Retain.Mode 变量更改工作模式。 • State = 0：未激活 • State = 1：预调节 • State = 2：精确调节 • State = 3：自动模式 • State = 4：手动模式 • State = 5：逼近替代输出值 • State = 6：转换时间测量 • State = 7：错误监视 • State = 8：在监视错误的同时逼近替代输出值
Error	BOOL	FALSE	如果 Error = TRUE，则至少一条错误消息处于待决状态。
ErrorBits	DWORD	DW#16#0	ErrorBits 参数 (页 419)指示错误消息。

参见

参数 State 和 Retain.Mode V1 (页 411)

参数 ErrorBits V1 (页 419)

8.2.5.5 PID_3Step V1 静态变量

不得更改未列出的变量。这些变量仅供内部使用。

表格 8- 12

变量	数据类型	默认值	说明
ActivateRecoverMode	BOOL	TRUE	ActivateRecoverMode 变量 (页 422)确定错误响应方式。
RunModeByStartup	BOOL	TRUE	在 CPU 重启后激活模式 如果 RunModeByStartup = TRUE，则控制器将返回到 CPU 重启后的上一个活动工作模式。 如果 RunModeByStartup = FALSE，则控制器在 CPU 重启后仍保持未激活状态。
PhysicalUnit	INT	0	过程值和设定值的测量单位，例如 ºC 或 ºF。
PhysicalQuantity	INT	0	过程值和设定值的物理量，如温度。
ErrorBehaviour	INT	0	如果 ErrorBehaviour = 0 且发生错误，则阀门会停留在其当前位置，控制器会直接切换到“未激活”模式或“错误监视”模式。 如果 ErrorBehaviour = 1 且出现了错误，则执行器将移动到替代输出值对应的位置，并且仅在此时才切换到“未激活”模式或“错误监视”模式。 如果发生以下错误，将不再能将阀门移动到组态的替代输出值对应的位置。 • 2000h：Feedback_PER 参数的值无效。 • 4000h：Feedback 参数的值无效。 • 8000h：数字位置反馈期间出错。
Warning	DWORD	DW#16#0	Warnings 变量 (页 411)显示自复位或上一次切换工作模式以来所生成的警告。 在删除警告原因前，会一直显示循环警告（如过程值警告）。一旦其产生原因消失，将自动删除这些警告。非循环警告（如未发现拐点）会保留且可以像错误一样被删除。

变量	数据类型	默认值	说明
SavePosition	REAL	0.0	替代输出值 如果 ErrorBehaviour = 1 且出现了错误，则执行器将移动到设备的安全位置，并且仅在此时才切换到“未激活”模式。
CurrentSetpoint	REAL	0.0	当前活动的设定值。此值将在调节开始时冻结。
Progress	REAL	0.0	百分数形式的调节进度 (0.0 - 100.0)
Config.InputPerOn	BOOL	TRUE	如果 InputPerOn = TRUE，则使用参数 Input_PER。如果 InputPerOn = FALSE，则使用参数 Input。
Config.OutputPerOn	BOOL	FALSE	如果 OutputPerOn = TRUE，则使用参数 Output_PER。如果 OutputPerOn = FALSE，则将使用 Ouput_UP 和 Output_DN 参数。
Config.LoadBackUp	BOOL	FALSE	如果 LoadBackUp = TRUE，则重新加载上一个 PID 参数集。该设置在最后一次调节前保存。LoadBackUp 自动设置回 FALSE。
Config.InvertControl	BOOL	FALSE	反转控制逻辑 如果 InvertControl = TRUE，则不断增大的控制偏差将导致输出值减小。
Config.FeedbackOn	BOOL	FALSE	如果 FeedbackOn = FALSE，则会仿真位置反馈。 位置反馈通常在 FeedbackOn = TRUE 时激活。
Config.FeedbackPerOn	BOOL	FALSE	仅当 FeedbackOn = TRUE 时，FeedbackPerOn 才有效。 如果 FeedbackPerOn = TRUE，则将模拟量输入用于位置反馈（Feedback_PER 参数）。 如果 FeedbackPerOn = FALSE，则将 Feedback 参数用于位置反馈。
Config.ActuatorEndStopOn	BOOL	FALSE	如果 ActuatorEndStopOn = TRUE，则将考虑数字位置反馈 Actuator_L 和 Actuator_H 。

变量	数据类型	默认值	说明
Config.InputUpperLimit	REAL	120.0	过程值的上限 在 I/O 输入中，过程值最大可超出标准范围 18%（过范围）。因超出“过程值上限”，将不再报告错误。仅识别断线和短路，然后 PID_3Step 将根据已组态的错误响应方式进行响应。 InputUpperLimit > InputLowerLimit
Config.InputLowerLimit	REAL	0.0	过程值的下限 InputLowerLimit < InputUpperLimit
Config.InputUpperWarning	REAL	+3.402822e+38	过程值的警告上限 如果设置的 InputUpperWarning 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作警告上限。 如果组态的 InputUpperWarning 值位于过程值的限值范围内，则该值将用作警告上限。 InputUpperWarning > InputLowerWarning InputUpperWarning ≤ InputUpperLimit
Config.InputLowerWarning	REAL	-3.402822e+38	过程值的警告下限 如果设置的 InputLowerWarning 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作警告下限。 如果组态的 InputLowerWarning 值位于过程值的限值范围内，则该值将用作警告下限。 InputLowerWarning < InputUpperWarning InputLowerWarning ≥ InputLowerLimit
Config.OutputUpperLimit	REAL	100.0	输出值的上限 有关详细信息，请参见 OutputLowerLimit

变量	数据类型	默认值	说明
Config.OutputLowerLimit	REAL	0.0	输出值的下限 如果 OutputPerOn = TRUE 或 FeedbackOn = TRUE，则 -100% 到 +100% 的值范围有效（包括零）。-100% 时，Output = -27648；+100% 时，Output = 27648 如果 OutputPerOn = FALSE，则 0% 到 100% 的值范围有效。阀门在 0% 时完全关闭，在 100% 时完全打开。
Config.SetpointUpperLimit	REAL	+3.402822e+38	设定值的上限 如果设置的 SetpointUpperLimit 超出了过程值的限值范围，则所组态的绝对过程值上限将预分配为设定值的上限。 如果组态的 SetpointUpperLimit 值位于过程值的限值范围内，则该值将用作设定值的上限。
Config.SetpointLowerLimit	REAL	- 3.402822e+38	设定值的下限 如果设置的 SetpointLowerLimit 超出了过程值的限值范围，则所组态的绝对过程值下限将预分配为设定值的下限。 如果设置的 SetpointLowerLimit 值位于过程值的限值范围内，则该值将用作设定值的下限。
Config.MinimumOnTime	REAL	0.0	最短 ON 时间 伺服驱动器必须开启的最短时间（以秒为单位）。 只有在使用 Output_UP 和 Output_DN 的情况下 (Config.OutputPerOn = FALSE)，Config.MinimumOnTime 才有效。
Config.MinimumOffTime	REAL	0.0	最短 OFF 时间 伺服驱动器必须关闭的最短时间（以秒为单位）。 只有在使用 Output_UP 和 Output_DN 的情况下 (Config.OutputPerOn = FALSE)，Config.MinimumOffTime 才有效。

变量	数据类型	默认值	说明
Config.TransitTime	REAL	30.0	电机转换时间 起动驱动器将阀门从关闭状态移至开启状态所需的时间（以秒为单位）。
Config.InputScaling.UpperPointIn	REAL	27648.0	标定的 Input_PER 上限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.InputScaling.LowerPointIn	REAL	0.0	标定的 Input_PER 下限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.InputScaling.UpperPointOut	REAL	100.0	标定的过程值的上限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.InputScaling.LowerPointOut	REAL	0.0	标定的过程值的下限 根据以下两个值对将 Input_PER 转换为百分数：InputScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.FeedbackScaling.UpperPointIn	REAL	27648.0	标定的 Feedback_PER 上限 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.FeedbackScaling.LowerPointIn	REAL	0.0	标定的 Feedback_PER 下限 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。

变量	数据类型	默认值	说明
Config.FeedbackScaling.UpperPointOut	REAL	100.0	上端停止位 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
Config.FeedbackScaling.LowerPointOut	REAL	0.0	下端停止位 根据以下两个值对将 Feedback_PER 转换为百分数：FeedbackScaling 结构的 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn。
GetTransitTime.InvertDirection	BOOL	FALSE	如果 InvertDirection = FALSE，则阀门将完全打开、关闭，然后再重新打开，以确定阀门转换时间。 如果 InvertDirection = TRUE，阀门会完全关闭，打开，然后再次关闭。
GetTransitTime.SelectFeedback	BOOL	FALSE	如果 SelectFeedback = TRUE，则转换时间测量中将考虑 Feedback_PER 或 Feedback。 如果 SelectFeedback = FALSE，则转换时间测量中将考虑 Actuator_H 和 Actuator_L。
GetTransitTime.Start	BOOL	FALSE	如果 Start = TRUE，则开始转换时间测量。
GetTransitTime.State	INT	0	转换时间测量的当前阶段 • State = 0：未激活 • State = 1：完全打开阀门 • State = 2：完全关闭阀门 • State = 3：将阀门移至目标位置 (NewOutput) • State = 4：成功完成转换时间测量 • State = 5：已取消转换时间测量
GetTransitTime.NewOutput	REAL	0.0	使用位置反馈时转换时间测量的目标位置 目标位置必须介于“上端停止位”和“下端停止位”之间。NewOutput 与 ScaledFeedback 之间的差值必须至少是允许控制范围的 50%。

变量	数据类型	默认值	说明
CycleTime.StartEstimation	BOOL	TRUE	如果 StartEstimation = TRUE，则开始测量 PID_3Step 采样时间。一旦测量完成，CycleTime.StartEstimation = FALSE。
CycleTime.EnEstimation	BOOL	TRUE	如果 EnEstimation = TRUE，则计算 PID_3Step 采样时间。
CycleTime.EnMonitoring	BOOL	TRUE	如果 EnMonitoring = TRUE，则监视 PID_3Step 采样时间。如果无法在采样时间内执行 PID_3Step，将输出错误 0800h 并且工作模式将发生更改。 ActivateRecoverMode 和 ErrorBehaviour 可确定切换为哪种工作模式。 如果 EnMonitoring = FALSE，则不会监视 PID_3Step 采样时间，不会输出错误 0800h，也不会切换工作模式。
CycleTime.Value	REAL	0.1	PID_3Step 采样时间（以秒为单位） CycleTime.Value 会自动确定，通常等于调用 OB 的循环时间。
CtrlParamsBackUp.SetByUser	BOOL	FALSE	保存的 Retain.CtrlParams.SetByUser 的值。 Config.LoadBackUp = TRUE 时，可以从 CtrlParamsBackUp 结构中重新加载值。
CtrlParamsBackUp.Gain	REAL	1.0	保存的比例增益
CtrlParamsBackUp.Ti	REAL	20.0	保存的积分作用时间
CtrlParamsBackUp.Td	REAL	0.0	保存的微分作用时间
CtrlParamsBackUp.TdFiltRatio	REAL	0.0	保存的微分延时系数
CtrlParamsBackUp.PWeighting	REAL	0.0	保存的比例作用权重
CtrlParamsBackUp.DWeighting	REAL	0.0	保存的微分作用权重
CtrlParamsBackUp.Cycle	REAL	1.0	保存的 PID 算法的采样时间
CtrlParamsBackUp.InputDeadBand	REAL	0.0	保存的控制偏差的死区宽度

变量	数据类型	默认值	说明
PIDSelfTune.SUT.CalculateSUTParams	BOOL	FALSE	受控系统的属性在调节期间保存。如果 CalculateSUTParams = TRUE，PID 参数都将根据这些属性进行重新计算。将使用 TuneRuleSUT 中设置的方法计算 PID 参数。计算后，CalculateSUTParams 将设置为 FALSE。
PIDSelfTune.SUT.TuneRuleSUT	INT	1	预调节期间用于计算参数的方法： • TuneRuleSUT = 0：PID 快速 I • TuneRuleSUT = 1：PID 慢速 I • TuneRuleSUT = 2：Chien、Hrones 和 Reswick PID • TuneRuleSUT = 3：Chien、Hrones、Reswick PI • TuneRuleSUT = 4：PID 快速 II • TuneRuleSUT = 5：PID 慢速 II
PIDSelfTune.SUT.State	INT	0	SUT.State 变量指示当前的预调节阶段：
PIDSelfTune.TIR.RunIn	BOOL	FALSE	• RunIn = FALSE 在未激活模式或手动模式下启动精确调节时，将启动预调节。 如果精确调节在自动模式下启动，系统将使用现有的 PID 参数来控制设定值。 之后才会启动精确调节。如果无法实现预调节，PID_3Step 将切换到“未激活”模式。 • RunIn = TRUE 将跳过预调节，PID_3Step 会尝试利用最小或最大输出值达到设定值。这可能会增加超调量。之后才会启动精确调节。 精确调节后，RunIn 将设置为 FALSE。
PIDSelfTune.TIR.CalculateTIRParams	BOOL	FALSE	受控系统的属性在调节期间保存。如果 CalculateTIRParams = TRUE，PID 参数都将根据这些属性进行重新计算。将使用 TuneRuleTIR 中设置的方法计算 PID 参数。计算后，CalculateTIRParams 将设置为 FALSE。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.TuneRuleTIR	INT	0	精确调节期间用于计算参数的方法： • TuneRuleTIR = 0：PID 自动 • TuneRuleTIR = 1：PID 快速 • TuneRuleTIR = 2：PID 慢速 • TuneRuleTIR = 3：Ziegler-Nichols PID • TuneRuleTIR = 4：Ziegler-Nichols PI • TuneRuleTIR = 5：Ziegler-Nichols P
PIDSelfTune.TIR.State	INT	0	TIR.State 变量 指示当前的“精确调节”阶段：
Retain.Mode	INT	0	Retain.Mode 值的改变会使工作模式发生切换。 Mode 发生变化时，将相应启用以下操作模式： • Mode = 0：未激活 • Mode = 1：预调节 • Mode = 2：精确调节 • Mode = 3：自动模式 • Mode = 4：手动模式 • Mode = 5：逼近替代输出值 • Mode = 6：转换时间测量 • Mode = 7：错误监视 • Mode = 8：在监视错误的同时逼近替代输出值 保持 Mode。
Retain.CtrlParams.SetByUser	BOOL	FALSE	如果 SetByUser = FALSE，PID 参数将自动确定并且 PID_3Step 将在输出值中存在死区的情况下运行。死区宽度将在调节期间根据输出值的标准差计算得出并保存到 Retain.CtrlParams.OutputDeadBand 中。 如果 SetByUser = TRUE，PID 参数将手动输入并且 PID_3 Step 将在输出值中不存在死区的情况下运行。 Retain.CtrlParams.OutputDeadBand = 0.0 保持 SetByUser。

变量	数据类型	默认值	说明
Retain.CtrlParams.Gain	REAL	1.0	有效的比例增益 保持 Gain。
Retain.CtrlParams.Ti	REAL	20.0	• Ti > 0.0：有效积分作用时间 • Ti = 0.0：积分作用取消激活 保持 Ti。
Retain.CtrlParams.Td	REAL	0.0	• Td > 0.0：有效的微分作用时间 • Td = 0.0：微分作用取消激活 保持 Td。
Retain.CtrlParams.TdFiltRatio	REAL	0.0	有效的微分延时系数 保持 TdFiltRatio。
Retain.CtrlParams.PWeighting	REAL	0.0	有效的比例作用权重 保持 PWeighting。
Retain.CtrlParams.DWeighting	REAL	0.0	有效的微分作用权重 保持 DWeighting。
Retain.CtrlParams.Cycle	REAL	1.0	PID 算法的有效采样时间（以秒为单位），舍入为调用 OB 的循环时间的整数倍。 保持 Cycle。
Retain.CtrlParams.InputDeadBand	REAL	0.0	控制偏差的死区宽度 保持 InputDeadBand。

说明

请在“未激活”模式下更改本表列出的变量，以防 PID 控制器出现故障。“Retain.Mode”变量设置为“0”将强制切换为“未激活”模式。

参见

参数 State 和 Retain.Mode V1 (页 411)

变量 ActivateRecoverMode V1 (页 422)

将工艺对象下载到设备 (页 76)

8.2.5.6 参数 State 和 Retain.Mode V1

参数的相关性

State 参数显示了 PID 控制器的当前工作模式。 您无法更改 State 参数。

要从一个工作模式切换为另一个，必须更改 Retain.Mode 变量。 当新工作模式的值已处于 Retain.Mode 中时，这同样适用。 例如，首先设置 Retain.Mode = 0，然后设置 Retain.Mode = 3。 如果控制器的当前工作模式允许此切换，则会将 State 设置为 Retain.Mode 的值。

PID_3Step 从一种工作模式自动切换为另一种时，State != Retain.Mode。

示例：

- 预调节成功完成后
  State = 3 且 Retain.Mode = 1
- 在发生错误时
  State = 0 且 Retain.Mode 将保持为先前的值，例如 Retain.Mode = 3
- ManualEnalbe = TRUE
  State = 4，并且 Retain.Mode 保持之前的值不变，例如 Retain.Mode = 3

说明

例如，如果希望重复成功的精确调节而不退出自动模式，则设 Mode = 0。

在一个周期内将 Retain.Mode 设置为无效值（例如 9999）对 State 没有影响。 在下一个周期中设置 Mode = 2。 通过这种方式，可以对 Retain.Mode 进行更改而无需先切换到“未激活”模式。

值的含义

State / Retain.Mode	说明
0	未激活 控制器关闭，且不再更改阀门位置。
1	预调节 预调节可确定对输出值脉冲的过程响应，并搜索拐点。 根据受控系统的最大上升速率与死时间的函数计算最佳的 PID 参数。 预调节的要求： • State = 0 或 State = 4 • ManualEnable = FALSE • 已对电机转换时间进行了组态或测量。 • 设定值和过程值均在组态的限值范围内。 过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。 只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。 重新计算 PID 参数之前将对其进行备份并且可使用 Config.LoadBackUp 重新激活这些参数。设定值在变量 CurrentSetpoint 中冻结。 预调节成功后控制器将切换到自动模式，预调节失败后将切换到“未激活”模式。 预调节阶段通过 SUT.State 变量来指示。
2	精确调节 精确调节将使过程值出现恒定受限的振荡。 根据该振荡的幅度和频率对 PID 参数进行调节。对预调节期间与精确调节期间的过程响应之间的差异进行分析。 所有 PID 参数都根据结果重新计算。精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。 PID_3Step 将自动尝试生成大于过程值噪声的振荡。 过程值的稳定性对精确调节的影响非常小。 精确调节前会备份 PID 参数。 可以使用 Config.LoadBackUp 重新激活这些参数。 设定值在变量 CurrentSetpoint 中冻结。

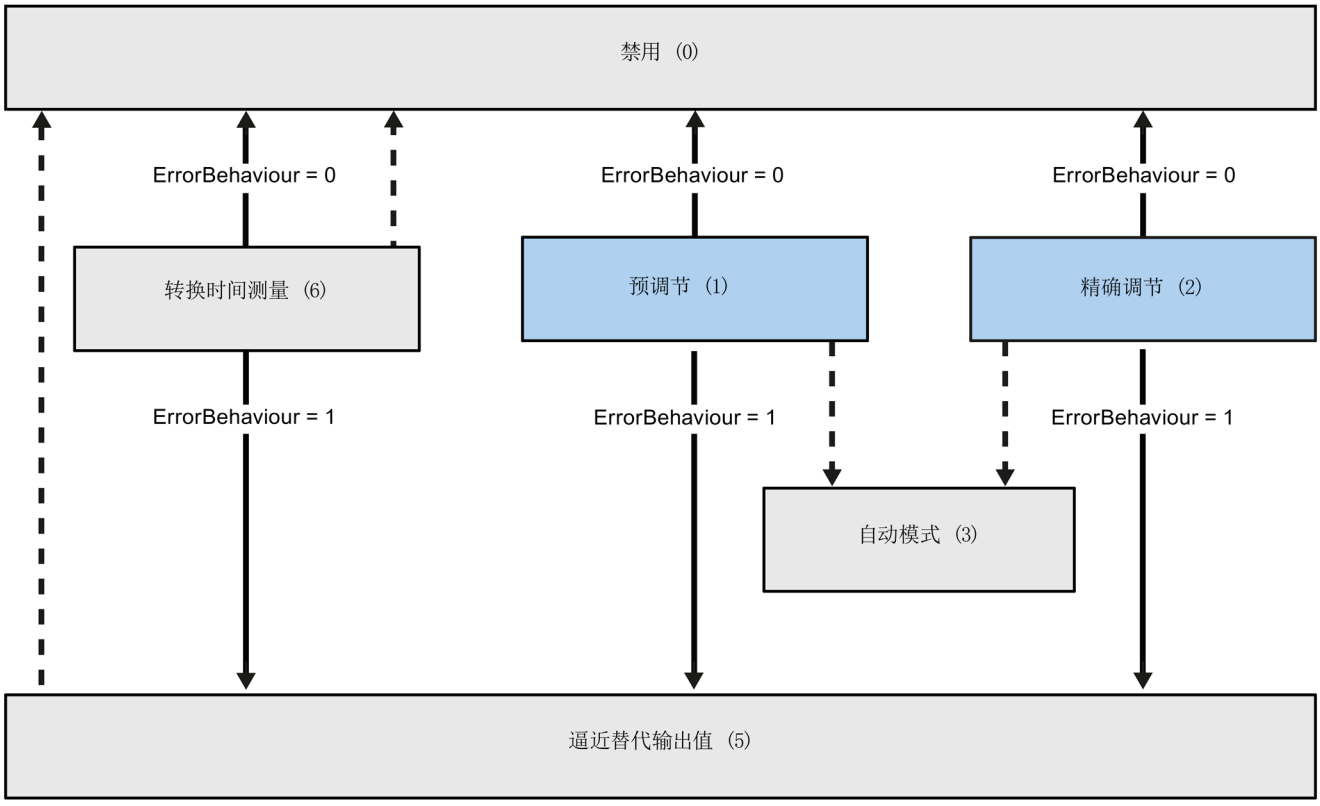
State / Retain.Mode	说明
2	精确调节的要求： • 已对电机转换时间进行了组态或测量。 • 设定值和过程值均在组态的限值范围内。 • ManualEnable = FALSE • 自动模式 (State = 3)、未激活模式 (State = 0) 或手动模式 (State = 4) 在以下模式下启动精确调节时，具体情况如下所述： • 自动模式 (State = 3) 如果希望通过调节来改进现有 PID 参数，请在自动模式下启动精确调节。 PID_3Step 将使用现有的 PID 参数控制系统，直到控制回路已稳定并且精确调节的要求得到满足为止。 之后才会启动精确调节。 • 未激活模式 (State = 0) 或手动模式 (State = 4) 总是先启动预调节。 已确定的 PID 参数将用于控制，直到控制回路已稳定并且精确调节的要求得到满足为止。 如果 PIDSelfTune.TIR.RunIn = TRUE，则将跳过预调节，并将尝试利用最小或最大输出值来达到设定值。 这可能会增加超调量。 随后将自动启动精确调节。 精确调节成功后，控制器将切换到自动模式。 如果精确调节未成功，则控制器将切换到“未激活”模式。 精确调节阶段使用 TIR.State 变量来指示。
3	自动模式 在自动模式下，PID_3Step 会按照指定的参数来控制受控系统。 如果满足下列要求之一，则控制器将切换到自动模式： • 预调节成功完成 • 精确调节成功完成 • 将 Retain.Mode 变量的值更改为 3。 当 CPU 启动或从 Stop 模式切换为 RUN 模式时，PID_3Step 会以最近激活的工作模式启动。 要使 PID_3Step 保持在“未激活”模式下，应设置 RunModeByStartup = FALSE。 自动模式下会考虑 ActivateRecoverMode 变量。

State / Retain.Mode	说明
4	手动模式 在手动模式下，在 Manual_UP 和 Manual_DN 参数或 ManualValue 参数中指定手动输出值。在发生错误时执行器是否可移动到输出值的情况将在 ErrorBits 参数中说明。 如果 Retain.Mode = 4 或 ManualEnable 处于上升沿，将启用此工作模式。 如果 ManualEnable 变为 TRUE，则只有 State 将发生更改。Retain.Mode 将保留其当前值。ManualEnable 处于下降沿时，PID_3Step 返回到前一个工作模式。 到自动模式的切换是无扰动的。 **PID_3Step V1.1** 在发生错误时手动模式始终可行。 **PID_3Step V1.0** 手动模式取决于发生错误时的 ActivateRecoverMode 变量。
5	逼近替代输出值 如果 Errorbehaviour = 1 且 ActivateRecoverMode = FALSE.，则出现错误或 Reset = TRUE 时会激活该工作模式。 PID_3Step 将执行器移动到替代输出值位置，然后更改为“未激活”模式。
6	转换时间测量 电机将阀门从闭合状态完全打开的所需时间已确定。 当设置 GetTransitTime.Start = TRUE 时，将激活此工作模式。 如果使用停止位信号测量转换时间，则阀门将从当前位置完全打开、完全关闭然后再次完全打开。 如果 GetTransitTime.InvertDirection = TRUE，将反转此行为。 如果使用位置反馈测量转换时间，那么会将执行器从其当前位置移至目标位置。 测量转换时间期间，不考虑输出值的限值。 执行器可行进至上端停止位或下端停止位。

State / Retain.Mode	说明
7	错误监视 控制算法关闭，并且不再更改阀门的位置。 出现错误时会激活该工作模式而不激活“未激活”模式。 必须满足以下所有条件： • Mode = 3 （自动模式） • Errorbehaviour = 0 • ActivateRecoverMode = TRUE • 已出现一个或多个错误，并且 ActivateRecoverMode (页 422) 生效。 当错误不再处于未决状态时，PID_3Step 切换回自动模式。
8	在监视错误的同时逼近替代输出值 出现错误时将激活该工作模式，而不是“逼近替代输出值”模式。PID_3Step 会将执行器移动到替代输出值，然后切换到“错误监视”模式。 必须满足以下所有条件： • Mode = 3 （自动模式） • Errorbehaviour = 1 • ActivateRecoverMode = TRUE • 已出现一个或多个错误，并且 ActivateRecoverMode (页 422) 生效。 当错误不再处于未决状态时，PID_3Step 切换回自动模式。

在调试期间自动切换工作模式

PID_3Step 将在出现错误时自动切换工作模式。 下图说明了 ErrorBehaviour 对始于转换时间测量、预调节和精确调节模式的工作模式切换的影响。

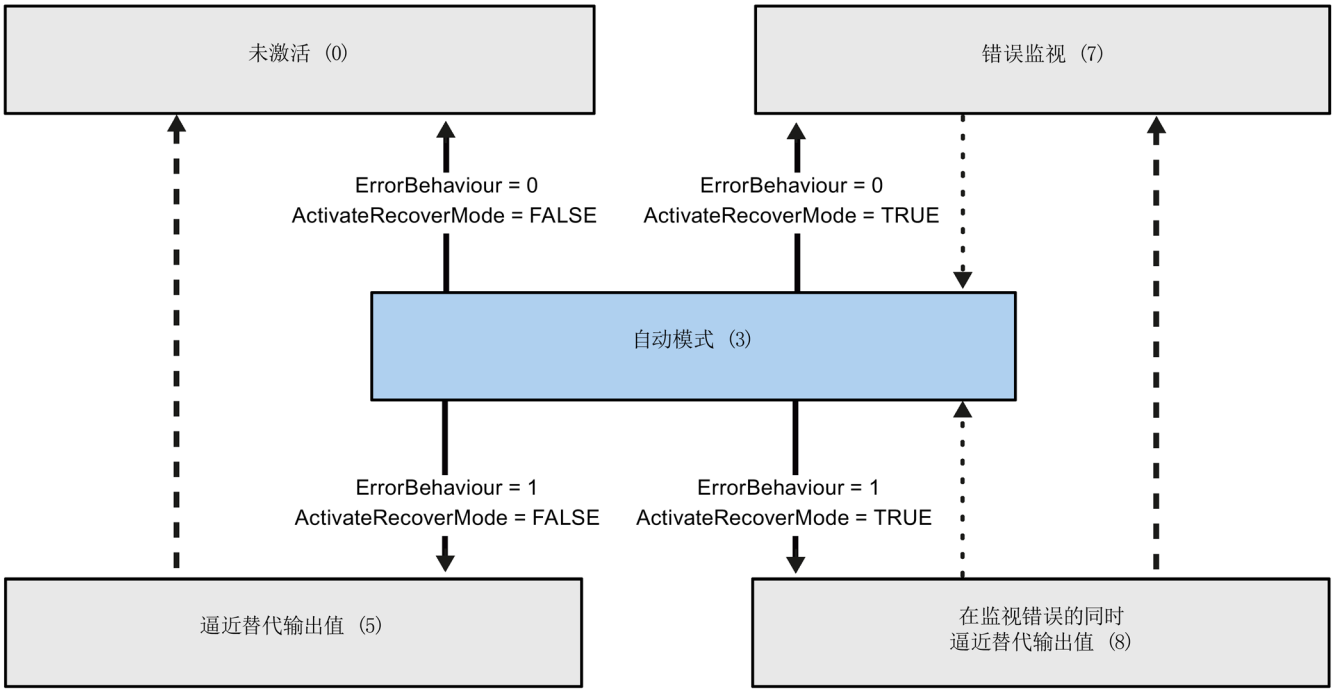


出现错误时自动切换工作模式。

完成当前操作后自动切换工作模式。

在自动模式下自动切换工作模式 (PID_3Step V1.1)

PID_3Step 将在出现错误时自动切换工作模式。 下图说明了 ErrorBehaviour 和 ActivateRecoverMode 对工作模式切换的影响。

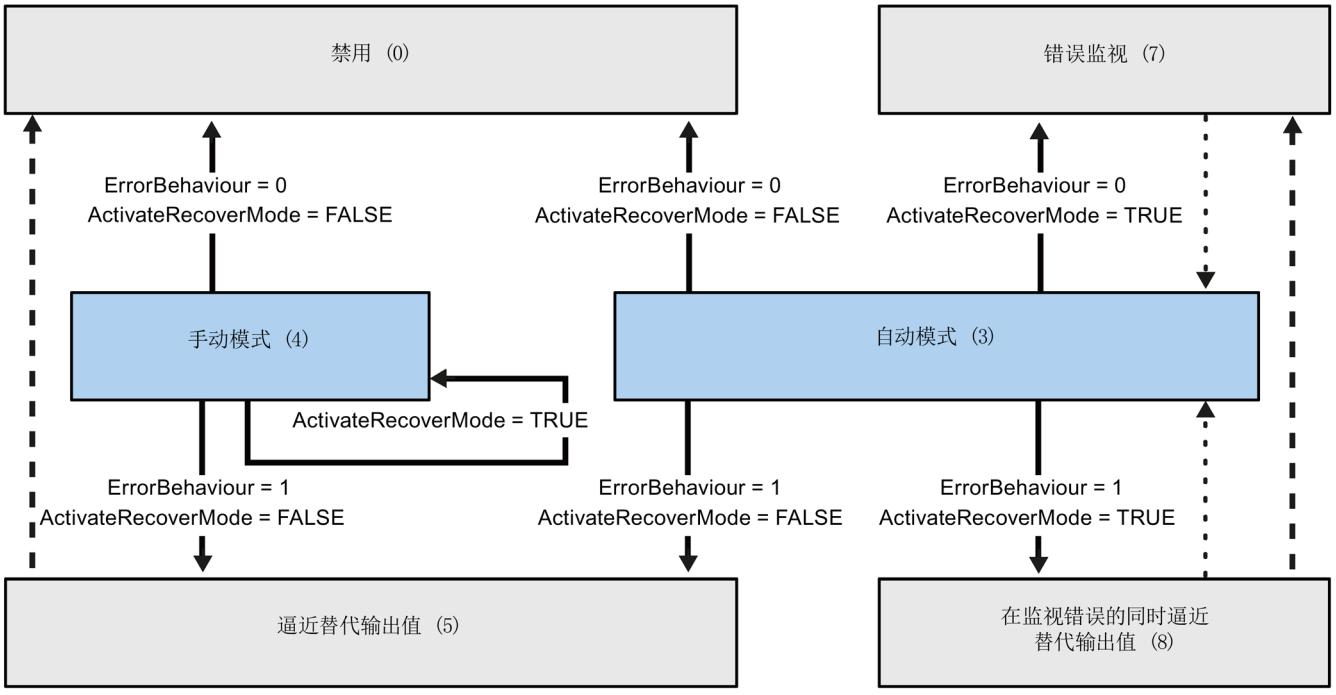


出现错误时自动切换工作模式。

完成当前操作后自动切换工作模式。

在自动和手动模式下自动切换工作模式 (PID_3Step V1.0)

PID_3Step 将在出现错误时自动切换工作模式。 下图说明了 ErrorBehaviour 和 ActivateRecoverMode 对工作模式切换的影响。



出现错误时自动切换工作模式。

完成当前操作后自动切换工作模式。

当错误不再处于未决状态时，自动切换工作模式。

参见

变量 ActivateRecoverMode V1 (页 422)

参数 ErrorBits V1 (页 419)

8.2.5.7 参数 ErrorBits V1

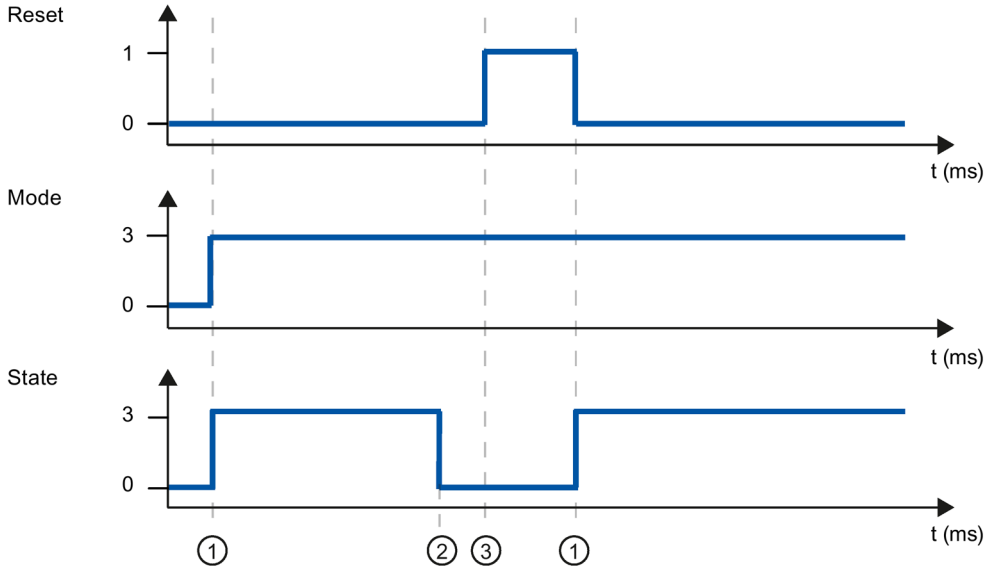
如果多个错误同时处于待决状态，将通过二进制加法显示错误代码值。例如，显示错误代码 0003 表示错误 0001 和 0002 同时处于待决状态。

ErrorBits (DW#16#...)	说明
0000	没有任何错误。
0001	参数“Input”超出了过程值限值的范围。 • Input > Config.InputUpperLimit 或 • Input < Config.InputLowerLimit 如果 ActivateRecoverMode = TRUE 并且 ErrorBehaviour = 1，则执行器将移动到替代输出值对应的位置。如果 ActivateRecoverMode = TRUE 并且 ErrorBehaviour = 0，则执行器停止在其当前位置。如果 ActivateRecoverMode = FALSE，则执行器停止在其当前位置。 PID_3Step V1.1 可以在手动模式下移动执行器。 PID_3Step V1.0 此状态下无法进入手动模式。在消除错误之前不能再次移动执行器。
0002	参数“Input_PER”的值无效。请检查模拟量输入是否有处于未决状态的错误。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。
0004	精确调节期间出错。过程值无法保持振荡状态。
0020	在自动模式下或调节期间不允许进行预调节。
0080	预调节期间出错。输出值限值的组态不正确。 检查输出值的限值是否已正确组态及其是否匹配控制逻辑。
0100	精确调节期间的错误导致生成无效参数。
0200	参数“Input”的值无效：值的数字格式无效。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。
0400	输出值计算失败。请检查 PID 参数。

ErrorBits (DW#16#...)	说明
0800	采样时间错误：在循环中断 OB 的采样时间内没有调用 PID_3Step。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。 如果在使用 PLCSIM 进行仿真期间出现该错误，请参见使用 PLCSIM 仿真 PID_3Step V1 (页 170)下的说明。
1000	参数“Setpoint”的值无效：值的数字格式无效。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。
2000	Feedback_PER 参数的值无效。 请检查模拟量输入是否有处于未决状态的错误。 执行器无法移动到替代输出值，并且将保持当前位置。此状态下无法进入手动模式。必须禁用位置反馈 (Config. FeedbackOn = FALSE) 才能使执行器退出此状态。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。
4000	Feedback 参数的值无效。值的数字格式无效。 执行器无法移动到替代输出值，并且将保持当前位置。此状态下无法进入手动模式。必须禁用位置反馈 (Config. FeedbackOn = FALSE) 才能使执行器退出此状态。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。
8000	数字位置反馈出现错误。Actuator_H = TRUE 和 Actuator_L = TRUE。 执行器无法移动到替代输出值，并且将保持当前位置。此状态下无法进入手动模式。 为将执行器移出此状态，必须禁用“执行器停止位”(Config.ActuatorEndStopOn = FALSE)。 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_3Step 切换回自动模式。

8.2.5.8 参数 Reset V1

Reset 出现上升沿时会触发切换到“未激活”模式，复位错误和警告。Reset 出现下降沿时会触发切换到最近激活的工作模式。如果之前已激活自动模式，则会以无扰动的方式切换至自动模式。



① 激活
② 错误
③ 复位

8.2.5.9 变量 ActivateRecoverMode V1

ActivateRecoverMode 变量的效果取决于 PID_3Step 的版本。

版的特性

ActivateRecoverMode 变量确定在自动模式下出现错误时的特性。ActivateRecoverMode 在预调节、精确调节和转换时间测量过程中不生效。

ActivateRecoverMode	说明
FALSE	出现错误时，PID_3Step 将切换到“未激活”工作模式或“逼近替代输出值”工作模式。 将通过复位或更改 Retain.Mode 值来激活控制器。
TRUE	如果在自动模式下频繁发生错误，则该设置会对控制响应产生负面影响。这种情况下，检查 ErrorBits 参数并消除错误原因。 如果发生一个或更多错误，PID_3Step 将切换到“在监视错误的同时逼近替代输出值”模式或“错误监视”模式： • 0002h：参数 Input_PER 的值无效。 • 0200h：参数 Input 的值无效。 • 0800h：采样时间错误 • 1000h：参数 Setpoint 的值无效。 • 2000h：参数 Feedback_PER 的值无效。 • 4000h：参数 Feedback 的值无效。 • 8000h: 数字位置反馈出现错误。 出现 2000h、4000h 和 8000h 错误时，PID_3Step 不能逼近组态的替代输出值。 当错误不再处于未决状态时，PID_3Step 切换回自动模式。

版的特性

ActivateRecoverMode 变量确定在自动模式和手动模式下发生错误时的特性。
ActivateRecoverMode 在预调节、精确调节和转换时间测量过程中不生效。

ActivateRecoverMode	说明
FALSE	出现错误时，PID_3Step 将切换到“未激活”工作模式或“逼近替代输出值”工作模式。 将通过复位或更改 Retain.Mode 值来激活控制器。
TRUE	**自动模式下的错误** 如果在自动模式下频繁发生错误，则该设置会对控制响应产生负面影响。这种情况下，检查 ErrorBits 参数并消除错误原因。 如果发生一个或更多错误，PID_3Step 将切换到“在监视错误的同时逼近替代输出值”模式或“错误监视”模式： • 0002h：参数 Input_PER 的值无效。 • 0200h：参数 Input 的值无效。 • 0800h：采样时间错误 • 1000h：参数 Setpoint 的值无效。 • 2000h：参数 Feedback_PER 的值无效。 • 4000h：参数 Feedback 的值无效。 • 8000h: 数字位置反馈出现错误。 出现 2000h、4000h 和 8000h 错误时，PID_3Step **不能**逼近组态的替代输出值。 当错误不再处于未决状态时，PID_3Step 切换回自动模式。 **手动模式下的错误** 如果发生一个或更多下列错误，则 PID_3Step 停留在手动模式下： • 0002h：参数 Input_PER 的值无效。 • 0200h：参数 Input 的值无效。 • 0800h：采样时间错误 • 1000h：参数 Setpoint 的值无效。 • 2000h：参数 Feedback_PER 的值无效。 • 4000h：参数 Feedback 的值无效。 • 8000h: 数字位置反馈出现错误。 出现 2000h、4000h 和 8000h 错误时，**不能**将阀门移动到合适的位置。

参见

PID_3Step V1 静态变量 (页 401)

参数 State 和 Retain.Mode V1 (页 411)

8.2.5.10 变量 Warning V1

如果多个警告同时处于待决状态，将通过二进制加法显示它们的值。 例如，显示警告 0003 表示警告 0001 和 0002 同时处于待决状态。

Warning (DW#16#...)	说明
0000	无警告处于待决状态。
0001	预调节期间未发现拐点。
0002	精确调节期间振荡增加。
0004	设定值被限制为组态的限值。
0008	在所选计算方法中未定义所有必要的受控系统属性。 因此，使用 TuneRuleTIR = 3 方法计算 PID 参数。
0010	由于 ManualEnable = TRUE，无法更改工作模式。
0020	调用 OB 的循环时间会限制 PID 算法的采样时间。 通过缩短 OB 循环时间来改进结果。
0040	过程值超出其警告限值之一。
0080	Retain.Mode 的值无效。 工作模式不变。
0100	手动值被限制为控制器输出的限值。
0200	用于调节的规则产生错误结果，或不受支持。
0400	为转换时间测量所选择的方法不适用于执行器。 由于执行器设置与所选的测量方法不匹配，无法测量转换时间。
0800	当前位置与新输出值之差太小，无法用于转换时间测量。 这可能产生错误结果。 当前输出值与新输出值之差必须至少是整个控制范围的 50%。
1000	无法达到替代输出值，因为它超出了输出值限值。

以下警告在消除问题的原因后即被删除：

- 0004
- 0020
- 0040
- 0100

所有其它警告均在 Reset 出现上升沿时清除。

8.2.5.11 变量 SUT.State V1

SUT.State	名称	说明
0	SUT_INIT	初始化预调节
50	SUT_TPDN	确定无位置反馈时的起始位置
100	SUT_STDABW	计算标准偏差
200	SUT_GET_POI	查找拐点
300	SUT_GET_RISETM	确定上升时间
9900	SUT_IO	预调节成功
1	SUT_NIO	预调节未成功

8.2.5.12 变量 TIR.State V1

TIR.State	名称	说明
-100	TIR_FIRST_SUT	无法进行精确调节。 将首先执行预调节。
0	TIR_INIT	初始化精确调节
200	TIR_STDABW	计算标准偏差
300	TIR_RUN_IN	尝试利用最大或最小输出值达到设定值
400	TIR_CTRLN	尝试使用现有 PID 参数达到设定值 （如果预调节已成功）
500	TIR_OSZIL	确定波动并计算参数
9900	TIR_IO	精确调节成功
1	TIR_NIO	精确调节未成功

8.3 PID_Temp

8.3.1 PID_Temp 的新特性

PID_Temp V1.1

- **从“未激活”工作模式切换到“自动模式”时输出值的响应**

  添加了新选项 IntegralResetMode = 4，并将其定义为默认设置。如果 IntegralResetMode = 4，从“未激活”工作模式切换到“自动模式”时会自动预分配积分作用，以便控制偏差导致带有相同符号的 PID 输出值发生跳变。

- **自动模式下积分作用的初始化**

  可以通过变量 OverwriteInitialOutputValue 和 PIDCtrl.PIDInit 在自动模式下对积分作用进行初始化。这简化了使用 PID_Temp 进行超驰控制的过程。

8.3.2 与 CPU 和 FW 的兼容性

下表显示了 PID_Temp 的每个版本可用于哪种 CPU。

CPU	FW	PID_Temp
S7-1200	V4.2 或更高版本	V1.1 V1.0
	V4.1	V1.0
S7-1500	V2.0 或更高版本	V1.1 V1.0
	V1.7 到 V1.8	V1.0

8.3.3 PID_Temp

8.3.3.1 PID_Temp 说明

说明

PID_Temp 指令提供了一种可对温度过程进行集成调节的 PID 控制器。PID_Temp 可用于纯加热或加热/制冷应用。

存在下列工作模式：

- 未激活
- 预调节
- 精确调节
- 自动模式
- 手动模式
- 含错误监视功能的替代输出值

有关工作模式的详细信息，请参见 State 参数。

PID 算法

PID_Temp 是一种具有抗积分饱和功能并且能够对比例作用和微分作用进行加权的 PIDT1 控制器。PID 算法根据以下等式工作（控制区和死区已禁用）：

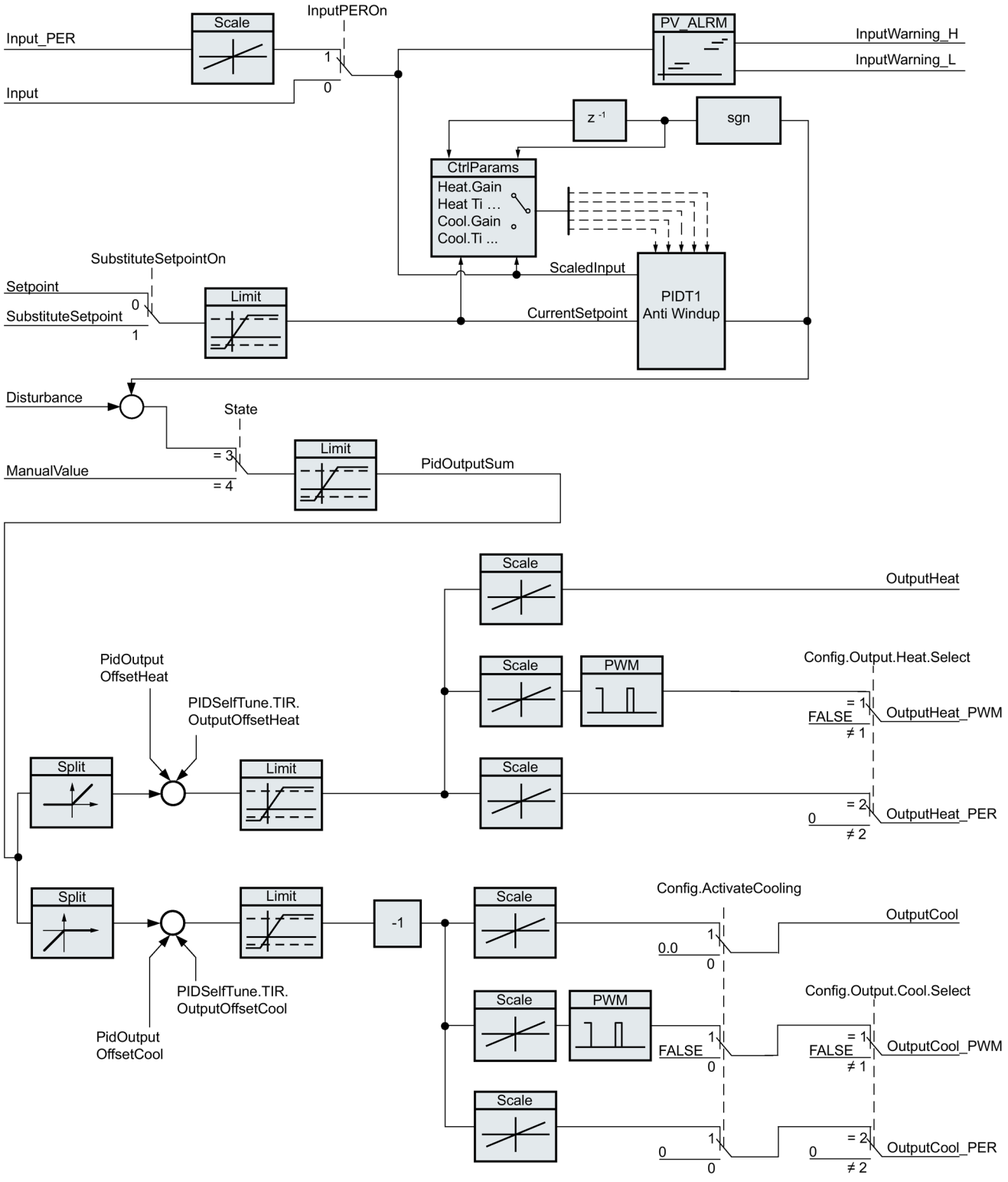
$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

下表给出了公式和后续图形中所使用图标的含义。

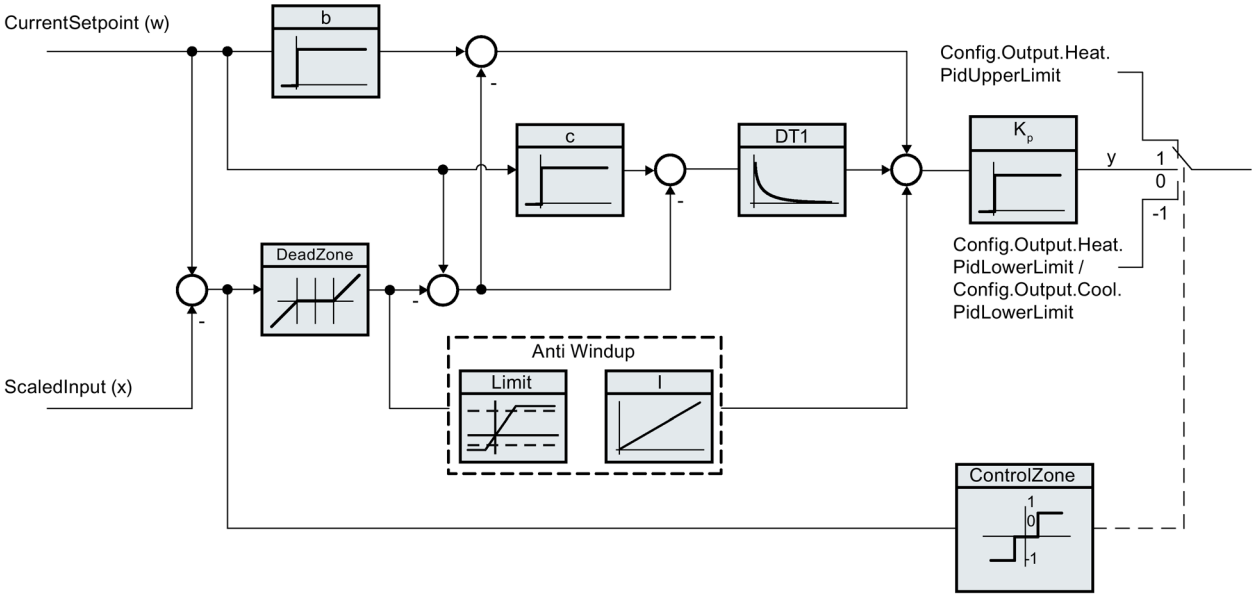
图标	说明	PID_Temp 指令的关联参数
y	PID 算法的输出值	-
K_p	比例增益	Retain.CtrlParams.Heat.Gain Retain.CtrlParams.Cool.Gain CoolFactor
s	拉普拉斯运算符	-

图标	说明	PID_Temp 指令的关联参数
b	比例作用权重	Re- tain.CtrlParams.Heat.PWeighting Re- tain.CtrlParams.Cool.PWeighting
w	设定值	CurrentSetpoint
x	过程值	ScaledInput
T_I	积分作用时间	Retain.CtrlParams.Heat.Ti Retain.CtrlParams.Cool.Ti
T_D	微分作用时间	Retain.CtrlParams.Heat.Td Retain.CtrlParams.Cool.Td
a	微分延迟系数（微分延迟 T1 = a × T_D）	Retain.CtrlParams.Heat.TdFiltRati o Retain.CtrlParams.Cool.TdFiltRati o
c	微分作用权重	Re- tain.CtrlParams.Heat.DWeighting Re- tain.CtrlParams.Cool.DWeighting
DeadZone	死区宽度	Retain.CtrlParams.Heat.DeadZon e Retain.CtrlParams.Cool.DeadZon e
ControlZone	控制区宽度	Retain.CtrlParams.Heat.ControlZo ne Retain.CtrlParams.Cool.ControlZo ne

PID_Temp 方框图


Input_PER
Scale
InputPEROn
1
0
Input
PV_ALRM
InputWarning_H
InputWarning_L
z -1
sgn
CtrlParams
Heat.Gain
Heat Ti ...
Cool.Gain
Cool.Ti ...
SubstituteSetpointOn
Setpoint
0
SubstituteSetpoint
1
Limit
ScaledInput
CurrentSetpoint
PIDT1
Anti Windup
Disturbance
State
= 3
ManualValue
= 4
Limit
PidOutputSum
Scale
OutputHeat
PidOutput
OffsetHeat
PIDSelfTune.TIR.
OutputOffsetHeat
Scale
PWM
Config.Output.Heat.Select
= 1
FALSE
≠ 1
OutputHeat_PWM
Split
Limit
Scale
= 2
0
≠ 2
OutputHeat_PER
Split
Limit
-1
Scale
Config.ActivateCooling
OutputCool
1
0.0
0
PIDSelfTune.TIR.
OutputOffsetCool
PidOutput
OffsetCool
Scale
PWM
Config.Output.Cool.Select
1
FALSE
0
= 1
FALSE
≠ 1
OutputCool_PWM
Scale
1
0
0
= 2
0
≠ 2
OutputCool_PER

带抗积分饱和的 PIDT1 的方框图



调用

在周期中断 OB 的恒定时间范围内调用 PID_Temp。

如果将 PID_Temp 作为多重实例 DB 调用，将不会创建任何工艺对象。没有参数分配接口或调试接口可用。必须直接在多重实例 DB 中为 PID_Temp 分配参数，并通过监视表格进行调试。

下载到设备

仅当完整下载 PID_Temp 后，才能更新保持性变量的过程值。

将工艺对象下载到设备 (页 76)

启动

CPU 启动时，PID_Temp 以保存在 Mode 输入/输出参数中的工作模式启动。要在启动期间切换到“未激活”工作模式，应设置 RunModeByStartup = FALSE。

对错误的响应

发生错误时的行为由变量 SetSubstituteOutput 和 ActivateRecoverMode 确定。如果 ActivateRecoverMode = TRUE，则行为还取决于所发生的错误。

SetSubstituteOutput	ActivateRecoverMode	组态编辑器 > 输出的基本设置 > 将 PidOutputSum 设置为	响应
不相关	FALSE	零（未激活）	切换到“未激活”(State = 0) 模式 PID 算法的输出值以及所有加热和制冷输出均设置为 0。加热和制冷输出的标定未激活。
FALSE	TRUE	发生错误时（错误未决时）的当前值	切换到“含错误监视功能的替代输出值”模式 (State = 5) 当错误未决时，当前输出值会传送到执行器。
TRUE	TRUE	错误未决时的替代输出值	切换到“含错误监视功能的替代输出值”模式 (State = 5) 当错误未决时，SubstituteOutput 中的值会传送到执行器。

在手动模式下，PID_Temp 使用 ManualValue 作为输出值，除非 ManualValue 无效。

- 如果 ManualValue 无效，将使用 SubstituteOutput。
- 如果 ManualValue 和 SubstituteOutput 无效，将使用 Config.Output.Heat.PidLowerLimit。

Error 参数指示是否存在错误处于未决状态。当错误不再处于未决状态时，Error = FALSE。ErrorBits 参数显示了已发生的错误。通过 Reset 或 ErrorAck 的上升沿来复位 ErrorBits。

8.3.3.2 PID_Temp 的工作模式

监视过程值的限值

在 Config.InputUpperLimit 和 Config.InputLowerLimit 变量中指定过程值的上限和下限。如果过程值超出这些限值，将出现错误 (ErrorBits = 0000001h)。

在 Config.InputUpperWarning 和 Config.InputLowerWarning 变量中指定过程值的警告上限和警告下限。如果过程值超出这些警告限值，将发生警告 (Warning = 0000040h)，并且 InputWarning_H 或 InputWarning_L 输出参数会更改为 TRUE。

限制设定值

在 Config.SetpointUpperLimit 变量和 Config.SetpointLowerLimit 变量中指定设定值的上限和下限。PID_Temp 会将设定值自动限制在过程值限值内。用户可以将设定值限制在更小范围内。PID_Temp 会检查该范围是否处于过程值限值内。如果设定值超出这些限值，上限和下限将用作设定值，并且输出参数 SetpointLimit_H 或 SetpointLimit_L 将设置为 TRUE。

在所有操作模式下均限制设定值。

替代设定值

用户可以在 SubstituteSetpoint 变量中指定替代设定值并通过 SubstituteSetpointOn = TRUE 将其激活。例如，通过这种方式可以直接为级联中的从控制器暂时指定设定值，而无需更改用户程序。为设定值设置的限值也适用于替代设定值。

加热和制冷

默认设置下，PID_Temp 仅使用加热输出（OutputHeat、OutputHeat_PWM、OutputHeat_PER）。PID 算法的输出值 (PidOutputSum) 经过标定在加热输出中输出。如果要计算 OutputHeat_PWM 或 OutputHeat_PER，需通过 Config.Output.Heat.Select 进行指定。始终会计算 OutputHeat。

还可以通过 Config.ActivateCooling = TRUE 激活制冷输出 (OutputCool, OutputCool_PWM, OutputCool_PER)。PID 算法的正输出值 (PidOutputSum) 将在标定后在加热输出中输出。PID 算法的负输出值则在标定后在制冷输出中输出。如果要计算 OutputCool_PWM 或 OutputCool_PER，需通过 Config.Output.Cool.Select 进行指定。始终会计算 OutputCool。

可通过两种方法计算已激活制冷的 PID 输出值：

- 制冷系数 (Config.AdvancedCooling = FALSE)：

  通过控制加热过程的 PID 参数并考虑可组态的制冷系数 Config.CoolFactor 来计算用于制冷的输出值。此方法适用于加热执行器和制冷执行器的时间响应相似但增益不同的情况。选择该方法时，无法对制冷进行预调节和精确调节并且控制制冷的 PID 参数集不可用。只能执行加热调节。

- PID 参数切换 (Config.AdvancedCooling = TRUE)：

  通过单独的 PID 参数集来计算制冷的输出值。PID 算法将根据计算出的输出值和控制偏差确定使用加热过程还是制冷过程的 PID 参数。此方法适用于加热执行器和制冷执行器的时间响应和增益都不同的情况。仅在选择该方法后才可对制冷进行预调节和精确调节。

输出值限值和标定

根据具体的工作模式，PID 输出值 (PidOutputSum) 将通过 PID 算法自动计算或者由手动值 (ManualValue)/已组态的替换输出值 (SubstituteOutput) 定义。

根据组态限制 PID 输出值：

- 如果禁用制冷 (Config.ActivateCooling = FALSE)，则 Config.Output.Heat.PidUpperLimit 作为上限值，Config.Output.Heat.PidLowerLimit 作为下限值。
- 如果激活制冷 (Config.ActivateCooling = TRUE)，则 Config.Output.Heat.PidUpperLimit 作为上限值，Config.Output.Cool.PidLowerLimit 作为下限值。

PID 输出值经过标定在加热和制冷输出中输出。可以单独为每个输出定义标定，并以带 2 个值对的 Config.Output.Heat 或 Config.Output.Cool 结构指定各个标定：

输出	值对	参数
OutputHeat	值对 1	PID 输出值上限（加热） Config.Output.Heat.PidUpperLimit， 标定的输出上限值（加热） Config.Output.Heat.UpperScaling
	值对 2	PID 输出值下限（加热） Config.Output.Heat.PidLowerLimit， 标定的输出下限值（加热） Config.Output.Heat.LowerScaling
OutputHeat_PWM	值对 1	PID 输出值上限（加热） Config.Output.Heat.PidUpperLimit， 标定的 PWM 输出上限值（加热） Config.Output.Heat.PwmUpperScaling
	值对 2	PID 输出值下限（加热） Config.Output.Heat.PidLowerLimit， 标定的 PWM 输出下限值（加热） Config.Output.Heat.PwmLowerScaling
OutputHeat_PER	值对 1	PID 输出值上限（加热） Config.Output.Heat.PidUpperLimit， 标定的模拟量输出上限值（加热） Config.Output.Heat.PerUpperScaling
	值对 2	PID 输出值下限（加热） Config.Output.Heat.PidLowerLimit， 标定的模拟量输出下限值（加热） Config.Output.Heat.PerLowerScaling
OutputCool	值对 1	PID 输出值下限（制冷） Config.Output.Cool.PidLowerLimit， 标定的输出上限值（制冷） Config.Output.Cool.UpperScaling

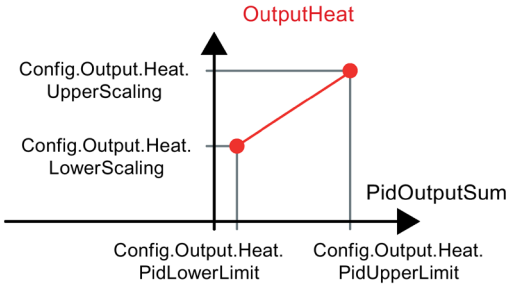
输出	值对	参数
	值对 2	PID 输出值上限（制冷） Config.Output.Cool.PidUpperLimit， 标定的输出下限值（制冷） Config.Output.Cool.LowerScaling
OutputCool_PWM	值对 1	PID 输出值下限（制冷） Config.Output.Cool.PidLowerLimit， 标定的 PWM 输出上限值（制冷） Config.Output.Cool.PwmUpperScaling
	值对 2	PID 输出值上限（制冷） Config.Output.Cool.PidUpperLimit， 标定的 PWM 输出下限值（制冷） Config.Output.Cool.PwmLowerScaling
OutputCool_PER	值对 1	PID 输出值下限（制冷） Config.Output.Cool.PidLowerLimit， 标定的模拟量输出上限值（制冷） Config.Output.Cool.PerUpperScaling
	值对 2	PID 输出值上限（制冷） Config.Output.Cool.PidUpperLimit， 标定的模拟量输出下限值（制冷） Config.Output.Cool.PerLowerScaling

如果激活制冷 (Config.ActivateCooling = TRUE)，则 Config.Output.Heat.PidLowerLimit 的值必须为 0.0。

Config.Output.Cool.PidUpperLimit 的值必须为 0.0。

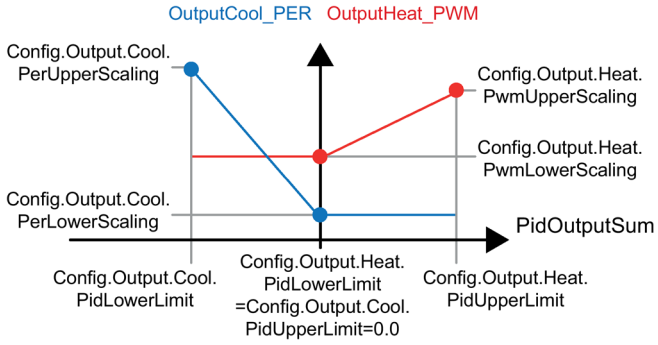
示例：

使用输出 OutputHeat 时的输出标定（禁用制冷；Config.Output.Heat.PidLowerLimit 可能不等于 0.0）：



示例：

使用输出 OutputHeat_PWM 和 OutputCool_PER 时的输出标定（激活制冷；Config.Output.Heat.PidLowerLimit 必须等于 0.0）：



除在“未激活”工作模式下外，输出的值始终介于其标定的输出上限值和标定的输出下限值之间，例如，OutputHeat 始终在 Config.Output.Heat.UpperScaling 和 Config.Output.Heat.LowerScaling 之间。

如果要限制相关输出的值，因此还必须调整这些标定值。

级联

PID_Temp 将在您使用级联控制时为您提供支持（请参见：创建程序 (页 210)）。

替代输出值

出现错误时，PID_Temp 可输出您在 SubstituteOutput 变量处定义的替代输出值。替换输出值必须处于 PID 输出值的限值范围内。在应用替代输出值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况。

监视信号有效性

使用以下参数时，监视其有效性：

- Setpoint
- SubstituteSetpoint
- Input
- Input_PER
- Disturbance
- ManualValue
- SubstituteOutput
- 具有 Retain.CtrlParams.Heat 和 Retain.CtrlParams.Cool. 结构的 PID 参数

监视采样时间 PID_Temp

理想情况下，采样时间等于循环中断 OB 的周期时间。PID_Temp 指令测量两次调用之间的时间间隔。这就是当前采样时间。每次切换工作模式以及初始启动期间，平均值由前 10 个采样时间构成。当前采样时间与该平均值之间的差值过大时会触发错误 (Error = 0000800h)。

如果存在以下情况，调节期间将发生错误：

- 新平均值 >= 1.1 x 原平均值
- 新平均值 <= 0.9 x 原平均值

如果存在以下情况，将在自动模式下发生错误：

- 新平均值 >= 1.5 x 原平均值
- 新平均值 <= 0.5 x 原平均值

如果禁用采样时间监视 (CycleTime.EnMonitoring = FALSE)，则也可在 OB1 中调用 PID_Temp。由于采样时间发生偏离，因此随后必须接受质量较低的控制。

PID 算法的采样时间

受控系统需要一定的时间来对输出值的变化做出响应。因此，建议不要在每次循环中都计算输出值。PID 算法的采样时间是两次计算输出值之间的时间。该时间在调节期间进行计算，并舍入为循环中断 OB 的循环时间（采样时间 PID_Temp）的倍数。PID_Temp 的所有其它功能会在每次调用时执行。

如果激活制冷和 PID 参数切换，则 PID_Temp 将为加热和制冷使用单独的 PID 算法采样时间。在其它所有组态中，仅使用加热的 PID 算法采样时间。

如果使用 OutputHeat_PWM 或 OutputCool_PWM，PID 算法的采样时间将用作脉宽调制的周期时间。输出信号的精度由 PID 算法采样时间与 OB 的周期时间之比来确定。该周期时间不应超出 PID 算法采样时间的十分之一。

如果使用 OutputHeat_PWM 或 OutputCool_PWM 时 PID 算法采样时间和脉宽调制的周期时间过长，则可在 Config.Output.Heat.PwmPeriode 或 Config.Output.Cool.PwmPeriode 参数中定义存在偏差的稍短周期时间来改善过程值的平滑度。

控制逻辑

PID_Temp 可用于加热或加热/制冷应用且始终使用常规控制逻辑。

PID 输出值 (PidOutputSum) 的增大用于增大过程值。在应用 PID 输出值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况。

不支持反转控制逻辑或负比例增益。

如果需要应用中的过程值随输出值的增大而减小（例如，放电控制），则可以使用具有反转控制逻辑的 PID_Compact。

8.3.3.3 PID_Temp 的输入参数

参数	数据类型	默认值	说明
Setpoint	REAL	0.0	PID 控制器在自动模式下的设定值 数值的有效范围： Config.SetpointUpperLimit ≥ Setpoint ≥ Config.SetpointLowerLimit Config.InputUpperLimit ≥ Setpoint ≥ Config.InputLowerLimit
Input	REAL	0.0	用户程序的变量用作过程值的源。 如果正在使用 Input 参数，则必须设置 Config.InputPerOn = FALSE。

参数	数据类型	默认值	说明
Input_PER	INT	0	模拟量输入用作过程值的源。 如果正在使用 Input_PER 参数，则必须设置 Config.InputPerOn = TRUE。
Disturbance	REAL	0.0	扰动变量或预控制值
ManualEnable	BOOL	FALSE	• 出现 FALSE -> TRUE 沿时会激活“手动模式”，而 State = 4 和 Mode 保持不变。 只要 ManualEnable = TRUE，便无法通过 ModeActivate 的上升沿或使用调试对话框来更改工作模式。 • 出现 TRUE -> FALSE 沿时会激活由 Mode 指定的工作模式。 建议只使用 Mode 和 ModeActivate 更改工作模式。
ManualValue	REAL	0.0	手动值 该值在手动模式下使用，用作 PID 输出值 (PidOutputSum)。 在应用此手动值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况（Config.Output.Heat 和 Config.Output.Cool 结构）。 对于具有已激活制冷输出的控制器 (Config.ActivateCooling = TRUE)，定义： • 正的手动值以输出加热输出中的值 • 负的手动值以输出制冷输出中的值 允许的取值范围由组态确定。 • 禁用制冷输出 (Config.ActivateCooling = FALSE)： Config.Output.Heat.PidUpperLimit ≥ ManualValue ≥ Config.Output.Heat.PidLowerLimit • 激活制冷输出 (Config.ActivateCooling = TRUE)： Config.Output.Heat.PidUpperLimit ≥ ManualValue ≥ Config.Output.Cool.PidLowerLimit
ErrorAck	BOOL	FALSE	• FALSE -> TRUE 沿 将复位 ErrorBits 和 Warning。

参数	数据类型	默认值	说明
Reset	BOOL	FALSE	重新启动控制器。 • FALSE -> TRUE 沿 – 切换到“未激活”模式 – 将复位 ErrorBits 和 Warning。 • 只要 Reset = TRUE， – PID_Temp 将保持在“未激活”模式下 (State = 0)。 – 无法通过 Mode 和 ModeActivate 或 ManualEnable 更改工作模式。 – 无法使用调试对话框。 • TRUE -> FALSE 沿 – 如果 ManualEnable = FALSE，则 PID_Temp 会切换到保存在 Mode 中的工作模式。 – 如果 Mode = 3（自动模式），会将积分作用视为已通过变量 IntegralResetMode 进行组态。
ModeActivate	BOOL	FALSE	• FALSE -> TRUE 沿 PID_Temp 切换到保存在 Mode 输入中的工作模式。

8.3.3.4 PID_Temp 的输出参数

参数	数据类型	默认值	说明
ScaledInput	REAL	0.0	标定的过程值
OutputHeat	REAL	0.0	REAL 形式的输出值（加热） PID 输出值 (PidOutputSum) 使用两个值对 Config.Output.Heat.PidUpperLimit、Config.Output.Heat.UpperScaling 和 Config.Output.Heat.PidLowerLimit、Config.Output.Heat.LowerScaling 进行标定，并以 REAL 形式在 OutputHeat 中输出。 始终计算 OutputHeat。
OutputCool	REAL	0.0	REAL 形式的输出值（制冷） PID 输出值 (PidOutputSum) 使用两个值对 Config.Output.Cool.PidUpperLimit、Config.Output.Cool.LowerScaling 和 Config.Output.Cool.PidLowerLimit、Config.Output.Cool.UpperScaling 进行标定，并以 REAL 形式在 OutputCool 中输出。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才会计算 OutputCool。
OutputHeat_PER	INT	0	模拟量输出值（加热） PID 输出值 (PidOutputSum) 使用两个值对 Config.Output.Heat.PidUpperLimit, Config.Output.Heat.PerUpperScaling 和 Config.Output.Heat.PidLowerLimit, Config.Output.Heat.PerLowerScaling 进行标定，并以模拟值形式在 OutputHeat_PER 中输出。 仅当 Config.Output.Heat.Select = 2 时才会计算 OutputHeat_PER。

参数	数据类型	默认值	说明
OutputCool_PER	INT	0	模拟量输出值（制冷） PID 输出值 (PidOutputSum) 使用两个值对 Config.Output.Cool.PidUpperLimit、Config.Output.Cool.PerLowerScaling 和 Config.Output.Cool.PidLowerLimit、Config.Output.Cool.PerUpperScaling 进行标定，并以模拟值形式在 OutputCool_PER 中输出。 仅在激活制冷输出 (Config.ActivateCooling = TRUE) 且 Config.Output.Cool.Select = 2 时才会计算 OutputCool_PER。
OutputHeat_PWM	BOOL	FALSE	脉宽调制输出值（加热） PID 输出值 (PidOutputSum) 使用两个值对 Config.Output.Heat.PidUpperLimit、Config.Output.Heat.PwmUpperScaling 和 Config.Output.Heat.PidLowerLimit、Config.Output.Heat.PwmLowerScaling 进行标定，并以脉宽调制值（变量开关时间）形式在 OutputHeat_PWM 中输出。 仅当 Config.Output.Heat.Select = 1 时才会计算 OutputHeat_PWM。
OutputCool_PWM	BOOL	FALSE	脉宽调制输出值（制冷） PID 输出值 (PidOutputSum) 使用两个值对 Config.Output.Cool.PidUpperLimit、Config.Output.Cool.PwmLowerScaling 和 Config.Output.Cool.PidLowerLimit、Config.Output.Cool.PwmUpperScaling 进行标定，并以脉宽调制值（变量开关时间）形式在 OutputCool_PWM 中输出。 仅在激活制冷输出 (Config.ActivateCooling = TRUE) 且 Config.Output.Cool.Select = 1 时才会计算 OutputCool_PWM。
SetpointLimit_H	BOOL	FALSE	如果 SetpointLimit_H = TRUE，则说明达到了设定值的绝对上限 (Setpoint ≥ Config.SetpointUpperLimit) 或者 Setpoint ≥ Config.InputUpperLimit。 设定值上限是 Config.SetpointUpperLimit 和 Config.InputUpperLimit 中的较小值。

参数	数据类型	默认值	说明
SetpointLimit_L	BOOL	FALSE	如果 SetpointLimit_L = TRUE，则说明达到了设定值的绝对下限 (Setpoint ≤ Config.SetpointLowerLimit) 或者 Setpoint ≤ Config.InputLowerLimit。 设定值下限是 Config.SetpointLowerLimit 和 Config.InputLowerLimit 中的较大值。
InputWarning_H	BOOL	FALSE	如果 InputWarning_H = TRUE，则说明过程值已达到或超出警告上限 (ScaledInput ≥ Config.InputUpperWarning)。
InputWarning_L	BOOL	FALSE	如果 InputWarning_L = TRUE，则说明过程值已经达到或低于警告下限 (ScaledInput ≤ Config.InputLowerWarning)。
State	INT	0	PID_Temp 状态和模式参数 (页 486)显示了 PID 控制器的当前工作模式。 可使用输入参数 Mode 和 ModeActivate 处的上升沿更改工作模式。对于预调节和精确调节，通过 Heat.EnableTuning 和 Cool.EnableTuning 指定针对加热还是制冷进行调节。 • State = 0：未激活 • State = 1：预调节 • State = 2：精确调节 • State = 3：自动模式 • State = 4：手动模式 • State = 5：含错误监视功能的替代输出值
Error	BOOL	FALSE	如果 Error = TRUE，则此周期内至少有一条错误消息处于未决状态。
ErrorBits	DWORD	DW#16#0	PID_Temp ErrorBits 参数 (页 496)显示了未决的错误消息。 ErrorBits 具有保持性，在 Reset 或 ErrorAck 出现上升沿时复位。

8.3.3.5 PID_Temp V2 的输入/输出参数

参数	数据类型	默认值	说明
Mode	INT	4	在 Mode 上，指定 PID_Temp 将转换到的工作模式。选项有： • Mode = 0：未激活 • Mode = 1：预调节 • Mode = 2：精确调节 • Mode = 3：自动模式 • Mode = 4：手动模式 工作模式由以下沿激活： • ModeActivate 的上升沿 • Reset 的下降沿 • ManualEnable 的下降沿 • 如果 RunModeByStartup = TRUE，则冷启动 CPU。 对于预调节和精确调节，通过 Heat.EnableTuning 和 Cool.EnableTuning 指定针对加热还是制冷进行调节。 保持 Mode。 有关工作模式的详细说明，请参见 State 和 Mode 参数 (页 486)。

参数	数据类型	默认值	说明
Master	DWORD	DW#16#0	级联控制的接口 如果该 PID_Temp 实例用作级联中的从控制器 (Config.Cascade.IsSlave = TRUE)，则在指令调用中通过主控制器的 Slave 参数分配 Master 参数。 示例： 在 SCL 中通过主控制器“PID_Temp_1”调用从控制器“PID_Temp_2”： --- `"PID_Temp_2"(Master := "PID_Temp_1".Slave, Setpoint := "PID_Temp_1".OutputHeat);` --- 使用此接口与主控制器交换关于从控制器的工作模式、限值和替代设定值的信息。 请记住，在同一个循环中断 OB 中，必须先调用主控制器，再调用从控制器。 分配： • 位 0 至 15：未分配 • 位 16 至 23 – 限值计数器： 输出值受限制的从控制器会使此计数器递增。 主控制器将根据已组态的从控制器数 (Config.Cascade.CountSlaves) 和抗积分饱和模式 (Config.Cascade.AntiWindUpMode) 作出相应反应。 • 位 24 – 从控制器的自动模式： 如果所有从控制器均处于自动模式，则为 TRUE • 位 25 – 从控制器的替代设定值： 如果从控制器已激活替代设定值 (SubstituteSetpointOn = TRUE)，则为 TRUE
Slave	DWORD	DW#16#0	级联控制的接口 使用此接口与主控制器交换关于从控制器的工作模式、限值和替代设定值的信息。 请参见 Master 参数的说明

参见

PID_Temp 状态和模式参数 (页 486)

创建程序 (页 210)

使用 PID_Temp 的级联控制 (页 208)

8.3.3.6 PID_Temp 静态变量

不得更改未列出的变量。这些变量仅供内部使用。

变量	数据类型	默认值	说明
IntegralResetMode	Int	V1.0：1， V1.1 或更高版本：4	IntegralResetMode 变量 (页 506)用于确定从“未激活”工作模式切换到“自动模式”时如何预分配积分作用 PIDCtrl.IOutputOld。 此设置仅在一个周期内有效。 • IntegralResetMode = 0：平滑 • IntegralResetMode = 1：删除 • IntegralResetMode = 2：保持 • IntegralResetMode = 3：预分配 • IntegralResetMode = 4：类似于设定值更改（仅适用于版本 1.1 及更高版本的 PID_Temp）
OverwriteInitialOutputValue	REAL	0.0	如果满足下列条件之一，则会自动预分配 PIDCtrl.IOutputOld 积分作用，就像上一周期中 PIDOutputSum = OverwriteInitialOutputValue 一样： • 从“未激活”工作模式切换到“自动模式”时 IntegralResetMode = 3 • 参数 Reset 的 TRUE -> FALSE 沿并且参数 Mode = 3 • 在“自动模式”下 PIDCtrl.PIDInit = TRUE（自 PID_Temp 版本 1.1 起可用）
RunModeByStartup	BOOL	TRUE	CPU 重启后，激活 Mode 参数中的工作模式。 • 如果 RunModeByStartup = TRUE，PID_Temp 将在 CPU 启动后以保存在模式参数中的工作模式启动。 • 如果 RunModeByStartup = FALSE，PID_Temp 在 CPU 启动后仍保持“未激活”模式。
LoadBackUp	BOOL	FALSE	如果 LoadBackUp = TRUE，则将从 CtrlParamsBackUp 结构中重新加载上一个 PID 参数集。该设置在最后一次调节前保存。LoadBackUp 自动设置回 FALSE。接受是无扰动的。

变量	数据类型	默认值	说明
SetSubstituteOutput	BOOL	TRUE	在错误未决时选择输出值 (State = 5)： • 如果 SetSubstituteOutput = TRUE 且 ActivateRecoverMode = TRUE，则只要错误未决，便会输出已组态的替代输出值 SubstituteOutput 作为 PID 输出值。 • 如果 SetSubstituteOutput = FALSE 且 ActivateRecoverMode = TRUE，则只要错误未决，执行器便会仍保持当前 PID 输出值。 • 如果 ActivateRecoverMode = FALSE，则 SetSubstituteOutput 无效。 • 如果 SubstituteOutput 无效 (ErrorBits = 0020000h)，则不能输出替代输出值。此时，会将加热的 PID 输出值下限 (Config.Output.Heat.PidLowerLimit) 用作 PID 输出值。
PhysicalUnit	INT	0	过程值和设定值的测量单位，例如 ºC 或 ºF。 该参数会在编辑器中显示，并且不影响控制算法。
PhysicalQuantity	INT	0	过程值和设定值的物理量，如温度。 该参数会在编辑器中显示，并且不影响控制算法。
ActivateRecoverMode	BOOL	TRUE	ActivateRecoverMode 变量决定错误响应方式。
Warning	DWORD	0	Warning 变量显示自 Reset = TRUE 或 ErrorAck =TRUE 以来的警告。警告具有保持性。
Progress	REAL	0.0	百分数形式的当前调节阶段进度 (0.0 - 100.0)
CurrentSetpoint	REAL	0.0	CurrentSetpoint 始终显示当前有效的设定值。调节期间该值处于冻结状态。
CancelTuningLevel	REAL	10.0	调节期间允许的设定值拐点。出现以下情况之前，不会取消调节： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel

变量	数据类型	默认值	说明
SubstituteOutput	REAL	0.0	只要满足以下条件，便会将替换输出值用作 PID 输出值： • 自动模式下有一个或多个错误未决，且 ActivateRecoverMode 有效 • SetSubstituteOutput = TRUE • ActivateRecoverMode= TRUE 在应用替代输出值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况（Config.Output.Heat 和 Config.Output.Cool 结构）。 对于具有已激活制冷输出的控制器 (Config.ActivateCooling = TRUE)，定义： • 正的替换输出值以在加热输出上输出该值 • 负的替换输出值以在制冷输出上输出该值 允许的取值范围由组态确定。 • 禁用制冷输出 (Config.ActivateCooling = FALSE)： Config.Output.Heat.PidUpperLimit ≥ SubstituteOutput ≥ Config.Output.Heat.PidLowerLimit • 激活制冷输出 (Config.ActivateCooling = TRUE)： Config.Output.Heat.PidUpperLimit ≥ SubstituteOutput ≥ Config.Output.Cool.PidLowerLimit

变量	数据类型	默认值	说明
PidOutputSum	REAL	0.0	PID 输出值 PidOutputSum 显示 PID 算法的输出值。根据具体的工作模式，将自动计算或通过手动值/已组态的替换输出值定义该值。 在应用 PID 输出值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况（Config.Output.Heat 和 Config.Output.Cool 结构）。 PidOutputSum 的限值在组态中进行定义。 • 禁用制冷输出 (Config.ActivateCooling = FALSE)： Config.Output.Heat.PidUpperLimit ≥ PidOutputSum ≥ Config.Output.Heat.PidLowerLimit • 激活制冷输出 (ConfigActivateCooling = TRUE)： Config.Output.Heat.PidUpperLimit ≥ PidOutputSum ≥ Config.Output.Cool.PidLowerLimit
PidOutputOffsetHeat	REAL	0.0	加热 PID 输出值的偏移量 PidOutputOffsetHeat 将添加到加热分支的 PidOutputSum 产生的值中。为 PidOutputOffsetHeat 输入正值以在加热输出上接收正偏移量。 加热输出中得到的值取决于输出标定的组态（Config.Output.Heat 结构）。 该偏移量可用于需要固定最小值的执行器，例如具有最小转速的风扇。
PidOutputOffsetCool	REAL	0.0	制冷 PID 输出值的偏移量 PidOutputOffsetCool 将添加到制冷分支的 PidOutputSum 产生的值中。为 PidOutputOffsetCool 输入负值以在制冷输出中接收正偏移量。 制冷输出中得到的值取决于输出标定的组态（ Config.Output.Cool 结构）。 该偏移量可用于需要固定最小值的执行器，例如具有最小转速的风扇。

变量	数据类型	默认值	说明
SubstituteSetpointOn	BOOL	FALSE	激活替代设定值作为控制器设定值。 • FALSE = 使用 Setpoint 参数。 • TRUE = 使用 SubstituteSetpoint 参数作为设定值 SubstituteSetpointOn 可用于直接指定级联中的从控制器的设定值，而无需更改用户程序。
SubstituteSetpoint	REAL	0.0	替代设定值 如果 SubstituteSetpointOn = TRUE，则 SubstituteSetpoint 参数用作设定值。 数值的有效范围： Config.SetpointUpperLimit ≥ SubstituteSetpoint ≥ Config.SetpointLowerLimit, Config.InputUpperLimit ≥ SubstituteSetpoint ≥ Config.InputLowerLimit
DisableCooling	BOOL	FALSE	DisableCooling = TRUE 通过将 PidOutputSum 设置为 0.0 作为下限来禁用自动模式下的加热/制冷控制器的制冷分支 (Config.ActivateCooling = TRUE)。 制冷输出的 PidOutputOffsetCool 和输出标定保持激活状态。 在所有控制器均完成调节前，可使用 DisableCooling 调节多区域应用以暂时禁用制冷分支。 该参数由用户手动设置/复位，而非通过 PID_Temp 指令自动复位。
AllSlaveAutomaticState	BOOL	FALSE	如果该 PID_Temp 实例用作级联中的主控制器 (Config.Cascade.IsMaster = TRUE)，则 AllSlaveAutomaticState = TRUE 表示所有从控制器均处于自动模式。 仅当所有从控制器均处于自动模式下时，才可精确执行主控制器的调节、手动模式或自动模式。 仅当主控制器和从控制器通过主从参数互连后才可确定 AllSlaveAutomaticState。 有关详细信息，请参见 Master 参数。

变量	数据类型	默认值	说明
NoSlaveSubstituteSetpoint	BOOL	FALSE	如果该 PID_Temp 实例用作级联中的主控制器 (Config.Cascade.IsMaster = TRUE)，则 NoSlaveSubstituteSetpoint = TRUE 表示所有从控制器均未激活其替代设定值。 仅当所有从控制器均未激活其替代设定值，才可精确执行主控制器的调节、手动模式或自动模式。 仅当主控制器和从控制器通过 Master 和 Slave 参数互连后才可确定 NoSlaveSubstituteSetpoint。 有关详细信息，请参见 Master 参数。
Heat.EnableTuning	BOOL	TRUE	启用加热调节 必须为以下调节设置 Heat.EnableTuning（以 Mode 或 ModeActivate 启动的同时或之前）： • 预调节加热 • 预调节加热和制冷 • 精确调节加热 该参数不会通过 PID_Temp 指令自动复位。
Cool.EnableTuning	BOOL	FALSE	启用制冷调节 必须为以下调节设置 Cool.EnableTuning（以 Mode 或 ModeActivate 启动的同时或之前）： • 预调节制冷 • 预调节加热和制冷 • 精确调节制冷 仅在激活制冷输出和 PID 参数切换时（“Config.ActivateCooling”= TRUE 且 “Config.AdvancedCooling”= TRUE）有效。 该参数不会通过 PID_Temp 指令自动复位。
Config.InputPerOn	BOOL	TRUE	如果 InputPerOn = TRUE，则将使用参数 Input_PER 检测过程值。如果 InputPerOn = FALSE，则使用参数 Input。

变量	数据类型	默认值	说明
Config.InputUpperLimit	REAL	120.0	过程值的上限 监控 Input 和 Input_PER，以确保符合此限值。如果超出限值，将输出错误并由 ActivateRecoverMode 确定响应方式。 在 I/O 输入中，过程值最大可超出额定范围 18%（过范围）。这意味着使用具有预设的上限和过程值标定的 I/O 输入时不会超出限值。 启动预调节后，将检查过程值上限和下限的差值以确定设定值和过程值之间的距离是否满足所需要求。 InputUpperLimit > InputLowerLimit
Config.InputLowerLimit	REAL	0.0	过程值的下限 监控 Input 和 Input_PER，以确保符合此限值。如果低于限值，将输出错误并由 ActivateRecoverMode 确定错误响应方式。 InputLowerLimit < InputUpperLimit
Config.InputUpperWarning	REAL	3.402822e+38	过程值的警告上限 监控 Input 和 Input_PER，以确保符合此限值。如果超出限值，将在 Warning 参数中输出警告。 • 如果设置的 InputUpperWarning 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作警告上限。 • 如果组态的 InputUpperWarning 值位于过程值的限值范围内，则该值将用作警告上限。 InputUpperWarning > InputLowerWarning
Config.InputLowerWarning	REAL	-3.402822e+38	过程值的警告下限 监控 Input 和 Input_PER，以确保符合此限值。如果低于限值，将在 Warning 参数中输出警告。 • 如果设置的 InputLowerWarning 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作警告下限。 • 如果组态的 InputLowerWarning 值位于过程值的限值范围内，则该值将用作警告下限。 InputLowerWarning < InputUpperWarning

变量	数据类型	默认值	说明
Config.SetpointUpperLimit	REAL	3.402822e+38	设定值的上限 监控 Setpoint 和 SubstituteSetpoint，以确保符合此限值。如果超出限值，将在 Warning 参数中输出警告。 • 如果组态的 SetpointUpperLimit 超出了过程值的限值范围，则所组态的过程值的绝对上限将用作设定值的上限。 • 如果组态的 SetpointUpperLimit 值位于过程值的限值范围内，则该值将用作设定值的上限。 SetpointUpperLimit > SetpointLowerLimit
Config.SetpointLowerLimit	REAL	-3.402822e+38	设定值的下限 监控 Setpoint 和 SubstituteSetpoint，以确保符合此限值。如果低于限值，将在 Warning 参数中输出警告。 • 如果设置的 SetpointLowerLimit 超出了过程值的限值范围，则所组态的过程值的绝对下限将用作设定值的下限。 • 如果组态的 SetpointLowerLimit 值位于过程值限值范围内，则该值将用作设定值下限。 SetpointLowerLimit < SetpointUpperLimit
Config.ActivateCooling	BOOL	FALSE	激活制冷输出 • Config.ActivateCooling = FALSE 仅使用加热输出。 • Config.ActivateCooling = TRUE 使用加热和制冷输出。 如果要使用制冷输出，则必须将控制器组态为主控制器（Config.Cascade.IsMaster 必须为 FALSE）。

变量	数据类型	默认值	说明
Config.AdvancedCooling	BOOL	TRUE	加热/制冷方法 • 制冷系数 (Config.AdvancedCooling = FALSE)： 通过控制加热过程的 PID 参数（以 Retain.CtrlParams.Heat 开头）并考虑可组态的制冷系数 Config.CoolFactor 来计算用于制冷的输出值。 此方法适用于加热执行器和制冷执行器的时间响应相似但增益不同的情况。 选择该方法时无法对制冷进行预调节和精确调节。只能执行加热调节。 • PID 参数切换 (Config.AdvancedCooling = TRUE) 通过单独的 PID 参数集（Retain.CtrlParams.Cool 结构）来计算制冷的输出值。 此方法适用于加热执行器和制冷执行器的时间响应和增益都不同的情况。 仅在选择该方法后才可对制冷进行预调节和精确调节（Mode = 1 或 2，Cool.EnableTuning = TRUE）。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才会计算 Config.AdvancedCooling。
Config.CoolFactor	REAL	1.0	制冷系数 如果 Config.AdvancedCooling = FALSE，则会将 Config.CoolFactor 视为制冷输出值计算中的系数。因此，可以考虑加热执行器与制冷执行器增益不同的情况。 Config.CoolFactor 既不会自动进行设置，也不会在调节期间进行调整。必须通过比值“加热执行器增益/制冷执行器增益”手动对 Config.CoolFactor 进行正确组态。 示例：Config.CoolFactor = 2.0 表示加热执行器增益是制冷执行器增益的两倍。 仅在激活制冷输出 (Config.ActivateCooling = TRUE) 且选择制冷系数作为加热/制冷方法时 (Config.AdvancedCooling = FALSE) Config.CoolFactor 才有效。 Config.CoolFactor > 0.0

变量	数据类型	默认值	说明
Config.InputScaling.UpperPointIn	REAL	27648.0	标定的 Input_PER 上限 根据两个值对 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn 对 Input_PER 进行标定。 仅当使用 Input_PER 进行过程值检测 (Config.InputPerOn = TRUE) 时才有效。 UpperPointIn > LowerPointIn
Config.InputScaling.LowerPointIn	REAL	0.0	标定的 Input_PER 下限 根据两个值对 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn 对 Input_PER 进行标定。 仅当使用 Input_PER 进行过程值检测 (Config.InputPerOn = TRUE) 时才有效。 LowerPointIn < UpperPointIn
Config.InputScaling.UpperPointOut	REAL	100.0	标定的过程上限值 根据两个值对 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn 对 Input_PER 进行标定。 仅当使用 Input_PER 进行过程值检测 (Config.InputPerOn = TRUE) 时才有效。 UpperPointOut > LowerPointOut
Config.InputScaling.LowerPointOut	REAL	0.0	标定的过程下限值 根据两个值对 UpperPointOut、UpperPointIn 和 LowerPointOut、LowerPointIn 对 Input_PER 进行标定。 仅当使用 Input_PER 进行过程值检测 (Config.InputPerOn = TRUE) 时才有效。 LowerPointOut < UpperPointOut
Config.Output.Heat.Select	INT	1	选择加热输出值 Config.Output.Heat.Select 指定用于加热的输出： • Heat.Select = 0 - 使用 OutputHeat • Heat.Select = 1 - 使用 OutputHeat 和 OutputHeat_PWM • Heat.Select = 2 - 使用 OutputHeat 和 OutputHeat_PER 未使用的输出不会计算，这些输出将保持其默认值。

变量	数据类型	默认值	说明
Config.Output.Heat.PwmPeriode	REAL	0.0	加热的脉宽调制 (PWM)（OutputHeat_PWM 输出）的周期时间（以秒为单位）： • Heat.PwmPeriode = 0.0 加热的 PID 算法的采样时间 (Retain.CtrlParams.Heat.Cycle) 用作 PWM 的周期时间。 • Heat.PwmPeriode > 0.0 该值将舍入为 PID_Temp 采样时间 (CycleTime.Value) 的整数倍并用作 PWM 的周期时间。 该设置可通过较长的 PID 算法采样时间来提高过程值的平滑度。 该值必须满足以下条件： – Heat.PwmPeriode ≤ Retain.CtrlParams.Heat.Cycle, – Heat.PwmPeriode > Config.Output.Heat.MinimumOnTime – Heat.PwmPeriode > Config.Output.Heat.MinimumOffTime
Config.Output.Heat.PidUpperLimit	REAL	100.0	加热的 PID 输出上限值 PID 输出值 (PidOutputSum) 限制为上限值。 Heat.PidUpperLimit 分别和以下参数构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热输出： • 用于 OutputHeat 的 Heat.UpperScaling • 用于 OutputHeat_PWM 的 Heat.PwmUpperScaling • 用于 OutputHeat_PER 的 Heat.PerUpperScaling 如果要限制相关输出中的值，还必须调整这些标定值。 Heat.PidUpperLimit > Heat.PidLowerLimit

变量	数据类型	默认值	说明
Config.Output.Heat.PidLowerLimit	REAL	0.0	加热的 PID 输出值下限 对于**已禁用**制冷输出的控制器 (Config.ActivateCooling = FALSE)，PID 输出值 (PidOutputSum) 限制为该下限值。 对于**已激活**制冷输出的控制器 (Config.ActivateCooling = TRUE),该值必须为 0.0。 Heat.PidLowerLimit 分别和以下参数构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热输出： • 用于 OutputHeat 的 Heat.LowerScaling • 用于 OutputHeat_PWM 的 Heat.PwmLowerScaling • 用于 OutputHeat_PER 的 Heat.PerLowerScaling 如果要限制相关输出中的值，还必须调整这些标定值。 允许的取值范围由组态确定。 • 禁用制冷输出 (Config.ActivateCooling = FALSE)： Heat.PidLowerLimit < Heat.PidUpperLimit • 激活制冷输出 (Config.ActivateCooling = TRUE)： Heat.PidLowerLimit = 0.0
Config.Output.Heat.UpperScaling	REAL	100.0	加热标定的输出上限值 Heat.UpperScaling 和 Heat.PidUpperLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热输出值 (OutputHeat)： OutputHeat 值始终位于 Heat.UpperScaling 和 Heat.LowerScaling 之间。 Heat.UpperScaling ≠ Heat.LowerScaling
Config.Output.Heat.LowerScaling	REAL	0.0	加热标定的输出下限值 Heat.LowerScaling 和 Heat.PidLowerLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热输出值 (OutputHeat)： OutputHeat 值始终位于 Heat.UpperScaling 和 Heat.LowerScaling 之间。 Heat.UpperScaling ≠ Heat.LowerScaling

变量	数据类型	默认值	说明
Config.Output.Heat.PwmUpperScaling	REAL	100.0	加热标定的 PWM 输出上限值 Heat.PwmUpperScaling 和 Heat.PidUpperLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热的脉宽调制输出值 (OutputHeat_PWM)。 OutputHeat_PWM 值始终位于 Heat.PwmUpperScaling 和 Heat.PWMLowerScaling 之间。 仅在选择 OutputHeat_PWM 作为加热输出时 (Heat.Select = 1) Heat.PwmUpperScaling 才有效。 100.0 ≥ Heat.PwmUpperScaling ≥ 0.0 Heat.PwmUpperScaling ≠ Heat.PwmLowerScaling
Config.Output.Heat.PwmLowerScaling	REAL	0.0	加热标定的 PWM 输出下限值 Heat.PwmLowerScaling 和 Heat.PidLowerLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热的脉宽调制输出值 (OutputHeat_PWM)。 OutputHeat_PWM 值始终位于 Heat.PwmUpperScaling 和 Heat.PwmLowerScaling 之间。 仅在选择 OutputHeat_PWM 作为加热输出时 (Heat.Select = 1) Heat.PwmLowerScaling 才有效。 100.0 ≥ Heat.PwmLowerScaling ≥ 0.0 Heat.PwmUpperScaling ≠ Heat.PwmLowerScaling

变量	数据类型	默认值	说明
Config.Output.Heat.PerUpperScaling	REAL	27648.0	加热标定的模拟量输出上限值 Heat.PerUpperScaling 和 Heat.PidUpperLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热模拟量输出值 (OutputHeat_PER)： OutputHeat_PER 值始终位于 Heat.PerUpperScaling 和 Heat.PerLowerScaling 之间。 仅在选择 OutputHeat_PER 作为加热输出时 (Heat.Select = 2) Heat.PerUpperScaling 才有效。 32511.0 ≥ Heat.PerUpperScaling ≥ -32512.0 Heat.PerUpperScaling ≠ Heat.PerLowerScaling
Config.Output.Heat.PerLowerScaling	REAL	0.0	加热标定的模拟量输出下限值 Heat.PerLowerScaling 和 Heat.PidLowerLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为加热模拟量输出值 (OutputHeat_PER)： OutputHeat_PER 值始终位于 Heat.PerUpperScaling 和 Heat.PerLowerScaling 之间。 仅在选择 OutputHeat_PER 作为加热输出时 (Heat.Select = 2) Heat.PerLowerScaling 才有效。 32511.0 ≥ Heat.PerLowerScaling ≥ -32512.0 Heat.PerUpperScaling ≠ Heat.PerLowerScaling
Config.Output.Heat.MinimumOnTime	REAL	0.0	加热的脉宽调制（OutputHeat_PWM 输出）的最短接通时间： PWM 脉冲绝不会短于该值。 该值将舍入为： Heat.MinimumOnTime = n × CycleTime.Value 仅在选择加热输出 OutputHeat_PWM (Heat.Select = 1) 时 Heat.MinimumOnTime 才有效。 100000.0 ≥ Heat.MinimumOnTime ≥ 0.0

变量	数据类型	默认值	说明
Config.Output.Heat.MinimumOffTime	REAL	0.0	加热的脉宽调制（OutputHeat_PWM 输出）的最短关断时间： PWM 暂停绝不会短于该值。 该值将舍入为： Heat.MinimumOffTime = n × CycleTime.Value 仅在选择加热输出 OutputHeat_PWM (Heat.Select = 1) 时 Heat.MinimumOffTime 才有效。 100000.0 ≥ Heat.MinimumOffTime ≥ 0.0
Config.Output.Cool.Select	INT	1	选择制冷输出值 Config.Output.Cool.Select 指定用于制冷的输出： • Cool.Select = 0 - 使用 OutputCool • Cool.Select = 1 - 使用 OutputCool 和 OutputCool_PWM • Cool.Select = 2 - 使用 OutputCool 和 OutputCool_PER 未使用的输出不会计算，这些输出将保持其默认值。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。

变量	数据类型	默认值	说明
Config.Output.Cool.PwmPeriode	REAL	0.0	制冷的脉宽调制（OutputCool_PWM 输出）的周期时间（以秒为单位）： • Cool.PwmPeriode = 0.0 且 Config.AdvancedCooling = FALSE： 加热的 PID 算法的采样时间 (Retain.CtrlParams.Heat.Cycle) 用作 PWM 的周期时间。 • Cool.PwmPeriode = 0.0 且 Config.AdvancedCooling = TRUE： 制冷的 PID 算法的采样时间 (Retain.CtrlParams.Cool.Cycle) 用作 PWM 的周期时间。 • Cool.PwmPeriode > 0.0： 该值将舍入为 PID_Temp 采样时间 (CycleTime.Value) 的整数倍并用作 PWM 的周期时间。 该设置可通过较长的 PID 算法采样时间来提高过程值的平滑度。 该值必须满足以下条件： – Cool.PwmPeriode ≤ Retain.CtrlParams.Cool.Cycle 或 Retain.CtrlParams.Heat.Cycle – Cool.PwmPeriode > Config.Output.Cool.MinimumOnTime – Cool.PwmPeriode > Config.Output.Cool.MinimumOffTime 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。

变量	数据类型	默认值	说明
Config.Output.Cool.PidUpperLimit	REAL	0.0	制冷的 PID 输出上限值 该值必须为 0.0。 Cool.PidUpperLimit 分别和以下参数构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷输出： • 用于 OutputCool 的 Cool.LowerScaling • 用于 OutputCool_PWM 的 Cool.PwmLowerScaling • 用于 OutputCool_PER 的 Cool.PerLowerScaling 如果要限制相关输出中的值，还必须调整这些标定值。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。 Cool.PidUpperLimit = 0.0
Config.Output.Cool.PidLowerLimit	REAL	-100.0	制冷的 PID 输出值下限 对于已激活制冷输出的控制器 (Config.ActivateCooling = TRUE)，PID 输出值 (PidOutputSum) 限制为该下限值。 Cool.PidLowerLimit 分别和以下参数构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷输出： • 用于 OutputCool 的 Cool.UpperScaling • 用于 OutputCool_PWM 的 Cool.PwmUpperScaling • 用于 OutputCool_PER 的 Cool.PerUpperScaling 如果要限制相关输出中的值，还必须调整这些标定值。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。 Cool.PidLowerLimit < Cool.PidUpperLimit

变量	数据类型	默认值	说明
Config.Output.Cool.UpperScaling	REAL	100.0	制冷标定的输出上限值 Cool.UpperScaling 和 Cool.PidLowerLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷输出值 (OutputCool)： OutputCool 值始终位于 Cool.UpperScaling 和 Cool.LowerScaling 之间。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。 Cool.UpperScaling ≠ Cool.LowerScaling
Config.Output.Cool.LowerScaling	REAL	0.0	制冷标定的输出下限值 Cool.LowerScaling 和 Cool.PidUpperLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷输出值 (OutputCool)： OutputCool 值始终位于 Cool.UpperScaling 和 Cool.LowerScaling 之间。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。 Cool.UpperScaling ≠ Cool.LowerScaling
Config.Output.Cool.PwmUpperScaling	REAL	100.0	制冷标定的 PWM 输出上限值 Cool.PwmUpperScaling 和 Cool.PidLowerLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷的脉宽调制输出值 (OutputCool_PWM)： OutputCool_PWM 值始终位于 Cool.PwmUpperScaling 和 Cool.PwmLowerScaling 之间。 仅在激活制冷输出 (Config.ActivateCooling = TRUE) 且选择 OutputCool_PWM 作为制冷输出时 (Cool.Select = 1) Cool.PwmUpperScaling 才有效。 100.0 ≥ Cool.PwmUpperScaling ≥ 0.0 Cool.PwmUpperScaling ≠ Cool.PwmLowerScaling

变量	数据类型	默认值	说明
Config.Output.Cool.PwmLowerScaling	REAL	0.0	制冷标定的 PWM 输出下限值 Cool.PwmLowerScaling 和 Cool.PidUpperLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷的脉宽调制输出值 (OutputCool_PWM)： OutputCool_PWM 值始终位于 Cool.PwmUpperScaling 和 CoolPwm.LowerScaling 之间。 仅在激活制冷输出 (Config.ActivateCooling = TRUE) 且选择 OutputCool_PWM 作为制冷输出时 (Cool.Select = 1) Cool.PwmLowerScaling 才有效。 100.0 ≥ Cool.PwmLowerScaling ≥ 0.0 Cool.PwmUpperScaling ≠ Cool.PwmLowerScaling
Config.Output.Cool.PerUpperScaling	REAL	27648.0	制冷标定的模拟量输出上限值 Cool.PerUpperScaling 和 Cool.PidLowerLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷模拟量输出值 (OutputCool_PER)。 OutputCool_PER 值始终位于 Cool.PerUpperScaling 和 Cool.PerLowerScaling 之间。 仅在激活制冷输出 (Config.ActivateCooling = TRUE) 且选择 OutputCool_PER 作为制冷输出时 (Cool.Select = 2) Cool.PerUpperScaling 才有效。 32511.0 ≥ Cool.PerUpperScaling ≥ -32512.0 Cool.PerUpperScaling ≠ Cool.PerLowerScaling

变量	数据类型	默认值	说明
Config.Output.Cool.PerLowerScaling	REAL	0.0	制冷标定的模拟量输出下限值 Cool.PerLowerScaling 和 Cool.PidUpperLimit 构成值对关系，用于将 PID 输出值 (PidOutputSum) 标定为制冷模拟量输出值 (OutputCool_PER)。 OutputCool_PER 值始终位于 Cool.PerUpperScaling 和 Cool.PerLowerScaling 之间。 仅在激活制冷输出 (Config.ActivateCooling = TRUE) 且选择 OutputCool_PER 作为制冷输出时 (Cool.Select = 2) Cool.PerLowerScaling 才有效。 32511.0 ≥ Cool.PerLowerScaling ≥ -32512.0 Cool.PerUpperScaling ≠ Cool.PerLowerScaling
Config.Output.Cool.MinimumOnTime	REAL	0.0	制冷的脉宽调制（OutputCool_PWM 输出）的最短接通时间 PWM 脉冲绝不会短于该值。 该值将舍入为： Cool.MinimumOnTime = n × CycleTime.Value 仅在选择制冷输出 OutputCool_PWM (Cool.Select = 1) 时 Cool.MinimumOnTime 才有效。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。 100000.0 ≥ Cool.MinimumOnTime ≥ 0.0
Config.Output.Cool.MinimumOffTime	REAL	0.0	制冷的脉宽调制（OutputCool_PWM 输出）的最短关断时间 PWM 暂停绝不会短于该值。 该值将舍入为： Cool.MinimumOffTime = n × CycleTime.Value 仅在选择制冷输出 OutputCool_PWM (Cool.Select = 1) 时 Cool.MinimumOffTime 才有效。 仅在激活制冷输出时 (Config.ActivateCooling = TRUE) 才有效。 100000.0 ≥ Cool.MinimumOffTime ≥ 0.0

变量	数据类型	默认值	说明
如果在级联中使用 PID_Temp，则主控制器和从控制器通过 Master 和 Slave 参数交换信息。 需要进行互连。有关详细信息，请参见 Master 参数。			
Config.Cascade.IsMaster	BOOL	FALSE	该控制器为级联中的主控制器，提供从控制器设定值。 如果要将该 PID_Temp 实例用作级联控制中的主控制器，则设置 IsMaster = TRUE。 主控制器通过其输出定义从控制器的设定值。PID_Temp 实例可以同时用作主控制器和从控制器。 如果该控制器用作主控制器，则必须禁用制冷输出 (Config.ActivateCooling = FALSE)。
Config.Cascade.IsSlave	BOOL	FALSE	该控制器在级联中为从控制器，并从主控制器中接收其设定值。 如果要将该 PID_Temp 实例用作级联中的从控制器，则设置 IsSlave = TRUE。 从控制器从其主控制器的输出（OutputHeat 参数）中接收其设定值（Setpoint 参数）。PID_Temp 实例可以同时用作主控制器和从控制器。
Config.Cascade.AntiWindUpMode	INT	1	级联中的抗积分饱和行为 选项有： • Anti-windup = 0 禁用 AntiWindUp 功能。主控制器不响应其从控制器的限值。 • Anti-windup = 1 主控制器的积分作用在比值“达到限值的从控制器/从控制器数量”（“CountSlaves”参数）中会减弱。这将减弱限值对控制行为的影响。 • Anti-windup = 2 从控制器达到限值后，主控制器的积分作用将立即暂停。 仅当控制器组态为主控制器时 (Config.Cascade.IsMaster = TRUE) 才有效。
Config.Cascade.CountSlaves	INT	1	从属从控制器的数量 在此处输入从该主控制器接收设定值的直接从属从控制器的数量。 仅当控制器组态为主控制器时 (Config.Cascade.IsMaster = TRUE) 才有效。 255 ≥ CountSlaves ≥ 1

变量	数据类型	默认值	说明
CycleTime.StartEstimation	BOOL	TRUE	如果 CycleTime.EnEstimation = TRUE，则 CycleTime.StartEstimation = TRUE 将开始自动确定 PID_Temp 采样时间（调用 OB 的循环时间）。 测量完成后，将设置 CycleTime.StartEstimation = FALSE。
CycleTime.EnEstimation	BOOL	TRUE	如果 CycleTime.EnEstimation = TRUE，将自动确定 PID_Temp 采样时间。 如果 CycleTime.EnEstimation = FALSE，则不会自动确定采样时间 PID_Temp，而是必须通过 CycleTime.Value 手动对该时间进行正确组态。
CycleTime.EnMonitoring	BOOL	TRUE	如果 CycleTime.EnMonitoring = FALSE，则不会监视 PID_Temp 采样时间。如果无法在采样时间内执行 PID_Temp，则既不会输出错误 (ErrorBits=0000800h)，PID_Temp 也不会按照 ActivateRecoverMode 的组态进行响应。
CycleTime.Value	REAL	0.1	PID_Temp 采样时间（调用 OB 的循环时间），以秒为单位 CycleTime.Value 会自动确定，通常等于调用 OB 的循环时间。
LoadBackUp = TRUE 时，可以从 CtrlParamsBackUp 结构中重新加载值。			
CtrlParamsBackUp.SetByUser	BOOL	FALSE	保存的 Retain.CtrlParams.SetByUser 的值
CtrlParamsBackUp.Heat.Gain	REAL	1.0	保存的加热比例增益
CtrlParamsBackUp.Heat.Ti	REAL	20.0	保存的加热积分作用时间（以秒为单位）
CtrlParamsBackUp.Heat.Td	REAL	0.0	保存的加热微分作用时间（以秒为单位）
CtrlParamsBackUp.Heat.TdFiltRatio	REAL	0.2	保存的加热微分延时系数
CtrlParamsBackUp.Heat.PWeighting	REAL	1.0	保存的加热比例作用的权重
CtrlParamsBackUp.Heat.DWeighting	REAL	1.0	保存的加热微分作用的权重
CtrlParamsBackUp.Heat.Cycle	REAL	1.0	保存的加热 PID 算法的采样时间（以秒为单位）
CtrlParamsBackUp.Heat.ControlZone	REAL	3.402822e+38	保存的加热控制区宽度

变量	数据类型	默认值	说明
CtrlParamsBackUp.Heat.DeadZone	REAL	0.0	保存的加热死区宽度
CtrlParamsBackUp.Cool.Gain	REAL	1.0	保存的制冷比例增益
CtrlParamsBackUp.Cool.Ti	REAL	20.0	保存的制冷积分作用时间（以秒为单位）
CtrlParamsBackUp.Cool.Td	REAL	0.0	保存的制冷微分作用时间（以秒为单位）
CtrlParamsBackUp.Cool.TdFiltRatio	REAL	0.2	保存的制冷微分延时系数
CtrlParamsBackUp.Cool.PWeighting	REAL	1.0	保存的制冷比例作用权重因子
CtrlParamsBackUp.Cool.DWeighting	REAL	1.0	保存的制冷微分作用权重因子
CtrlParamsBackUp.Cool.Cycle	REAL	1.0	保存的制冷 PID 算法的采样时间（以秒为单位）
CtrlParamsBackUp.Cool.ControlZone	REAL	3.402822e+38	保存的制冷控制区宽度
CtrlParamsBackUp.Cool.DeadZone	REAL	0.0	保存的制冷死区宽度
PIDSelfTune.SUT.CalculateParamsHeat	BOOL	FALSE	受控系统的加热分支属性在加热预调节期间保存。如果 SUT.CalculateParamsHeat = TRUE，将根据这些属性重新计算加热过程（Retain.CtrlParams.Heat 结构）的 PID 参数。这样无需重复进行调节，即可更改参数计算方法（PIDSelfTune.SUT.TuneRuleHeat 参数）。 计算后，SUT.CalculateParamsHeat 将设置为 FALSE。 仅当预调节成功时 (SUT.ProcParHeatOk = TRUE) 才能实现。

变量	数据类型	默认值	说明
PIDSelfTune.SUT.CalculateParamsCool	BOOL	FALSE	受控系统的制冷分支属性在制冷调节期间保存。如果 SUT.CalculateParamsCool = TRUE，将根据这些属性重新计算制冷过程（Retain.CtrlParams.Cool 结构）的 PID 参数。这样无需重复进行调节，即可更改参数计算方法（PIDSelfTune.SUT.TuneRuleCool 参数）。 计算后，SUT.CalculateParamsCool 将设置为 FALSE。 仅当预调节成功时 (SUT.ProcParCoolOk = TRUE) 才能实现。 仅当 Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE 时才有效。
PIDSelfTune.SUT.TuneRuleHeat	INT	2	通过加热预调节实现 PID 参数计算的方法 选项有： • SUT.TuneRuleHeat = 0：根据 CHR 计算 PID • SUT.TuneRuleHeat = 1：根据 CHR 计算 PI • SUT.TuneRuleHeat = 2：根据 CHR 计算温度过程的 PID（与 SUT.TuneRuleHeat = 0 相比，可生成更慢以及更接近的控制响应且过调很小） （CHR = Chien、Hrones 和 Reswick） 仅当 SUT.TuneRuleHeat = 2 时，控制区 Retain.CtrlParams.Heat.ControlZone 才会在加热预调节期间自动设置。

变量	数据类型	默认值	说明
PIDSelfTune.SUT.TuneRuleCool	INT	2	通过制冷预调节实现 PID 参数计算的方法 选项有： • SUT.TuneRuleCool = 0：根据 CHR 计算 PID • SUT.TuneRuleCool = 1：根据 CHR 计算 PI • SUT.TuneRuleCool = 2：根据 CHR 计算温度过程的 PID（与 SUT.TuneRuleCool = 0 相比，可生成更慢以及更接近的控制响应且过调很小） （CHR = Chien、Hrones 和 Reswick） 仅当 SUT.TuneRuleCool = 2 时，控制区 Retain.CtrlParams.Cool.ControlZone 才会在制冷预调节期间自动设置。 仅在激活制冷输出和 PID 参数切换时 (Config.ActivateCooling = TRUE, Config.AdvancedCooling = TRUE) SUT.TuneRuleCool 才有效。
PIDSelfTune.SUT.State	INT	0	SUT.State 变量指示当前的预调节阶段： • State = 0：初始化预调节 • State = 100：计算加热的标准偏差 • State = 200：计算制冷的标准偏差 • State = 300：确定加热拐点 • State = 400：确定制冷拐点 • State = 500：在达到拐点后将加热设置为设定值 • State = 600：在达到拐点后将制冷设置为设定值 • State = 700：比较加热执行器和制冷执行器的效率 • State = 800：加热和制冷已激活 • State = 900：制冷已激活 • State = 1000：确定停止加热后的延迟时间 • State = 9900：预调节成功 • State = 1：预调节未成功
PIDSelfTune.SUT.ProcParHeatOk	BOOL	FALSE	TRUE：预调节加热的过程参数计算成功。 该变量在调节期间进行设置。 计算加热 PID 参数时必须将其设置为 TRUE。

变量	数据类型	默认值	说明
PIDSelfTune.SUT.ProcParCoolOk	BOOL	FALSE	TRUE：预调节制冷的过程参数计算成功。 该变量在调节期间进行设置。 计算制冷 PID 参数时必须将其设置为 TRUE。
PIDSelfTune.SUT.AdaptDelayTime	INT	0	AdaptDelayTime 变量确定达到工作点时是否调整加热延迟时间（用于“预调节加热”和“预调节加热和制冷”）。 选项有： • SUT.AdaptDelayTime = 0： 不调整延迟时间。跳过 SUT.State = 1000 阶段。与 SUT.AdaptDelayTime = 1 相比，该选项可缩短调节时间。 • SUT.AdaptDelayTime = 1： 通过暂时停止加热将延迟时间调整为 SUT.State = 1000 阶段中的设定值。 与 SUT.AdaptDelayTime = 0 相比，该选项可延长调节时间。如果过程行为主要取决于工作点（非线性），则该选项可改善控制响应。该选项不适用于具有较强的热力连接的多区域应用。

变量	数据类型	默认值	说明
PIDSelfTune.SUT.CoolingMode	INT	0	CoolingMode 变量确定调节变量输出以确定制冷参数（用于预调节加热和制冷）。 选项有： • SUT.CoolingMode = 0： 达到设定值后停止加热并接通制冷。 跳过 SUT.State = 700 阶段。 阶段 SUT.State = 500 后跟阶段 SUT.State = 900。 如果制冷执行器的增益小于加热执行器的增益，则该选项可以改善控制响应。与 SUT.CoolingMode = 1 或 2 相比，该选项可缩短调节时间。 • SUT.CoolingMode = 1： 达到设定值后接通制冷并保持加热 跳过 SUT.State = 700 阶段。 阶段 SUT.State = 500 后跟阶段 SUT.State = 800。 如果制冷执行器的增益大于加热执行器的增益，则该选项可以改善控制响应。 • SUT.CoolingMode = 2： 加热到设定值后，阶段 SUT.State = 700 中将自动决定是否停止加热。阶段 SUT.State = 500 后跟阶段 SUT.State = 700，然后是 SUT.State = 800 或 SUT.State = 900。 与选项 0 或 1 相比，该选项将需要更长时间。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.RunIn	BOOL	FALSE	使用 RunIn 变量指定从自动模式启动时精确调节的顺序。 • RunIn = FALSE 如果精确调节在自动模式下启动，系统将使用现有的 PID 参数来控制设定值（TIR.State = 500 或 600）。之后才会启动精确调节。 • RunIn = TRUE PID_Temp 尝试利用最大或最小输出值达到设定值（TIR.State = 300 或 400）。这可能会增加超调量。随后将自动启动精确调节。 精确调节后，RunIn 将设置为 FALSE。 如果在未激活模式或手动模式下启动精确调节，PID_Temp 将按照 RunIn = TRUE 时所述的情况进行响应。
PIDSelfTune.TIR.CalculateParamsHeat	BOOL	FALSE	受控系统的加热分支属性在加热精确调节期间保存。如果 TIR.CalculateParamsHeat= TRUE，将根据这些属性重新计算加热过程（Retain.CtrlParams.Heat 结构）的 PID 参数。这样无需重复进行调节，即可更改参数计算方法（PIDSelfTune.TIR.TuneRuleHeat 参数）。 计算后，TIR.CalculateParamsHeat 将设置为 FALSE。 精确调节加热成功后 (TIR.ProcParHeatOk = TRUE) 才可实现。
PIDSelfTune.TIR.CalculateParamsCool	BOOL	FALSE	受控系统的制冷分支属性在制冷精确调节期间保存。如果 TIR.CalculateParamsCool= TRUE，将根据这些属性重新计算制冷过程（Retain.CtrlParams.Cool 结构）的 PID 参数。这样无需重复进行调节，即可更改参数计算方法（PIDSelfTune.TIR.TuneRuleCool 参数）。 计算后，TIR.CalculateParamsCool 将设置为 FALSE。 精确调节制冷成功后 (TIR.ProcParCoolOk = TRUE) 才可实现。 仅当 Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE 时才有效。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.TuneRuleHeat	INT	0	加热精确调节期间的参数计算方法 选项有： • TIR.TuneRuleHeat = 0：PID 自动 • TIR.TuneRuleHeat = 1：PID 快速（与 TIR.TuneRuleHeat = 2 相比，控制响应速度更快，输出值的幅度更大） • TIR.TuneRuleHeat = 2：PID 慢速（与 TIR.TuneRuleHeat = 1 相比，控制响应速度较慢，输出值的幅度较小） • TIR.TuneRuleHeat = 3：ZN PID • TIR.TuneRuleHeat = 4：ZN PI • TIR.TuneRuleHeat = 5：ZN P (ZN=Ziegler-Nichols) 要通过 TIR.CalculateParamsHeat 和 TIR.TuneRuleHeat = 0、1 或 2 重复计算加热过程的 PID 参数，也必须通过 TIR.TuneRuleHeat = 0、1 或 2 执行了先前的精确调节。否则，将使用 TIR.TuneRuleHeat = 3。 始终可以通过 TIR.CalculateParamsHeat 和 TIR.TuneRuleHeat = 3、4 或 5 重新计算加热 PID 参数。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.TuneRuleCool	INT	0	制冷精确调节期间的参数计算方法 选项有： • TIR.TuneRuleCool = 0：PID 自动 • TIR.TuneRuleCool = 1：PID 快速（与 TIR.TuneRuleCool = 2 相比，控制响应速度更快，输出值的幅度更大） • TIR.TuneRuleCool = 2：PID 慢速（与 TIR.TuneRuleCool = 1 相比，控制响应速度较慢，输出值的幅度较小） • TIR.TuneRuleCool = 3：ZN PID • TIR.TuneRuleCool = 4：ZN PI • TIR.TuneRuleCool = 5：ZN P (ZN=Ziegler-Nichols) 要通过 TIR.CalculateParamsCool 和 TIR.TuneRuleCool = 0、1 或 2 重复计算制冷过程的 PID 参数，也必须通过 TIR.TuneRuleCool = 0、1 或 2 执行了先前的精确调节。否则，将使用 TIR.TuneRuleCool = 3。 始终可以通过 TIR.CalculateParamsCool 和 TIR.TuneRuleCool = 3、4 或 5 重新计算制冷 PID 参数。 仅在激活制冷输出和 PID 参数切换时（ConfigActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.State	INT	0	TIR.State 变量指示当前的“精确调节”阶段： • State = 0：初始化精确调节 • State = 100：计算加热的标准偏差 • State = 200：计算制冷的标准偏差 • State = 300：正在尝试通过两步加热控制来达到加热过程的设定值 • State = 400：正在尝试通过两步制冷控制达到制冷过程的设定值 • State = 500：正在尝试通过 PID 控制达到加热过程的设定值 • State = 600：正在尝试通过 PID 控制达到制冷过程的设定值 • State = 700：计算加热的标准偏差 • State = 800：计算制冷的标准偏差 • State = 900：针对加热过程确定波动并计算参数 • State = 1000：针对制冷过程确定波动并计算参数 • State = 9900：精确调节已成功 • State = 1：精确调节未成功
PIDSelfTune.TIR.ProcParHeatOk	BOOL	FALSE	TRUE：精确调节加热的过程参数计算成功。 该变量在调节期间进行设置。 计算加热 PID 参数时必须将其设置为 TRUE。
PIDSelfTune.TIR.ProcParCoolOk	BOOL	FALSE	TRUE：精确调节制冷的过程参数计算成功。 该变量在调节期间进行设置。 计算制冷 PID 参数时必须将其设置为 TRUE。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.OutputOffsetHeat	REAL	0.0	PID 输出值的加热调节偏移量 TIR.OutputOffsetHeat 将添加到加热分支的 PidOutputSum 产生的值中。 要在加热输出上接收正偏移量，请为 TIR.OutputOffsetHeat 定义一个正值。 加热输出中得到的值取决于输出标定的组态 (Struktur Config.Output.Heat)。 已激活制冷输出和 PID 参数切换的控制器（Config.ActivateCooling = TRUE，Config.AdvancedCooling = TRUE）可使用该调节偏移量实现制冷精确调节。如果在达到要开始调节的设定值时制冷输出未激活 (PidOutputSum > 0.0)，则无法实现制冷精确调节。此时，定义一个正加热调节偏移量，且必须大于启动调节前相应设定值对应的稳态 PID 输出值 (PidOutputSum)。该步骤可增大加热输出中的值并激活制冷输出 (PidOutputSum < 0.0)。此时可以实现制冷精确调节。 精确调节完成后，TIR.OutputOffsetHeat 复位为 0.0。 TIR.OutputOffsetHeat 在一个步骤中发生较大更改可导致临时过调。 Config.Output.Heat.PidUpperLimit ≥ PIDSelfTune.TIR.OutputOffsetHeat ≥ Config.Output.Heat.PidLowerLimit

变量	数据类型	默认值	说明
PIDSelfTune.TIR.OutputOffsetCool	REAL	0.0	PID 输出值的制冷调节偏移量 TIR.OutputOffsetCool 将添加到制冷分支的 PidOutputSum 产生的值中。 要在制冷输出上接收正偏移量，请为 TIR.OutputOffsetCool 定义一个负值。 制冷输出中得到的值取决于输出标定的组态 (Struktur Config.Output.Coool)。 已激活制冷输出的控制器 (Config.ActivateCooling = TRUE) 可使用该调节偏移量实现加热精确调节。如果在达到要开始调节的设定值时加热输出未激活 (PidOutputSum < 0.0)，则无法实现加热精确调节。此时，定义一个负制冷调节偏移量，且必须小于启动调节前相应设定值对应的稳态 PID 输出值 (PidOutputSum)。该步骤可增大制冷输出中的值并激活加热输出 (PidOutputSum > 0.0)。此时可以实现加热精确调节。 精确调节完成后，TIR.OutputOffsetCool 复位为 0.0。 TIR.OutputOffsetCool 在一个步骤中发生较大更改可导致临时过调。 Config.Output.Cool.PidUpperLimit ≥ PIDSelfTune.TIR.OutputOffsetCool ≥ Config.Output.Cool.PidLowerLimit
PIDSelfTune.TIR.WaitForControlIn	BOOL	FALSE	达到设定值后在精确调节期间等待 如果 TIR.WaitForControlIn = TRUE，则在达到设定值（TIR.State = 500 或 600）后、计算标准偏差（TIR.State = 700 或 800）前的这段时间内，精确调节将一直等待，直到 TIR.FinishControlIn 出现 FALSE -> TRUE 沿为止。 TIR.WaitForControlIn 可用于多区域应用中多个控制器的同步精确调节以同步调节各个区域。这可确保在实际调节开始前，所有区域均已达到各自的设定值。利用这种方式，可减少各区域间的热力连接对调节的影响。 仅当通过 PIDSelfTune.TIR.RunIn = FALSE 从自动模式启动调节时 TIR.WaitForControlIn 才有效。

变量	数据类型	默认值	说明
PIDSelfTune.TIR.ControlInReady	BOOL	FALSE	如果 TIR.WaitForControlIn = TRUE，则达到设定值后 PID_Temp 会立即设置 TIR.ControlInReady = TRUE 并一直等待，直到 TIR.FinishControlIn 出现 FALSE -> TRUE 沿后再继续进行其它调节步骤。
PIDSelfTune.TIR.FinishControlIn	BOOL	FALSE	如果 TIR.ControlInReady = TRUE，则 TIR.FinishControlIn 的 FALSE -> TRUE 沿将停止等待并恢复精确调节。
PIDCtrl.IOutputOld	REAL	0.0	上一循环中的积分作用
PIDCtrl.PIDInit	BOOL	FALSE	自 PID_Temp 版本 1.1 起 PIDCtrl.PIDInit 可用。 如果在“自动模式”下 PIDCtrl.PIDInit = TRUE，则会自动预分配 PIDCtrl.IOutputOld 积分作用，就像上一周期中 PidOutputSum = OverwriteInitialOutputValue 一样。这可用于使用 PID_Temp 进行超驰控制 (页 220)。
Retain.CtrlParams.SetByUser	BOOL	FALSE	如果手动在组态编辑器中输入 PID 参数，则 SetByUser = TRUE。 该参数会在编辑器中显示，并且不影响控制算法。 SetByUser 具有保持性。
Retain.CtrlParams.Heat.Gain	REAL	1.0	有效的加热比例增益 Heat.Gain 具有保持性。 Heat.Gain ≥ 0.0
Retain..CtrlParams.Heat.Ti	REAL	20.0	有效的加热积分作用时间（以秒为单位） Heat.CtrlParams.Ti = 0.0 时，加热过程将关闭积分作用。 Heat.Ti 具有保持性。 100000.0 ≥ Heat.Ti ≥ 0.0
Retain.CtrlParams.Heat.Td	REAL	0.0	有效的加热微分作用时间（以秒为单位） Heat.CtrlParams.Td = 0.0 时，加热过程将关闭微分作用。 Heat.Td 具有保持性。 100000.0 ≥ Heat.Td ≥ 0.0

变量	数据类型	默认值	说明
Retain.CtrlParams.Heat.TdFiltRatio	REAL	0.2	有效的加热微分延时系数 微分延迟系数用于延迟微分作用的生效。 微分延迟 = 微分作用时间 × 微分延迟系数 • 0.0：微分作用仅在一个周期内有效，因此几乎不产生影响。 • 0.5：实践证明，该值对具有一个主时间常数的受控系统很有效。 • > 1.0：系数越大，微分作用的生效时间延迟越久。 Heat.TdFiltRatio 具有保持性。 Heat.TdFiltRatio ≥ 0.0
Retain.CtrlParams.Heat.PWeighting	REAL	1.0	有效的加热比例作用的权重 改变设定值有可能削弱比例作用。 允许使用 0.0 到 1.0 之间的值。 • 1.0：应对设定值变化的比例作用完全有效 • 0.0：应对设定值变化的比例作用无效 当过程值变化时，比例作用始终完全有效。 Heat.PWeighting 具有保持性。 1.0 ≥ Heat.PWeighting ≥ 0.0
Retain.CtrlParams.Heat.DWeighting	REAL	1.0	有效的加热微分作用的权重 微分作用随着设定值的变化而减弱。 允许使用 0.0 到 1.0 之间的值。 • 1.0：设定值变化时微分作用完全有效 • 0.0：设定值变化时微分作用不生效 当过程值变化时，微分作用始终完全有效。 Heat.DWeighting 具有保持性。 1.0 ≥ Heat.DWeighting ≥ 0.0

变量	数据类型	默认值	说明
Retain.CtrlParams.Heat.Cycle	REAL	1.0	有效的加热 PID 算法的采样时间（以秒为单位） 在调节期间计算 CtrlParams.Heat.Cycle，并将其舍入为 CycleTime.Value 的整数倍。 如果 Config.Output.Heat.PwmPeriode = 0.0，则 Heat.Cycle 用作加热脉宽调制的周期时间。 如果 Config.Output.Cool.PwmPeriode = 0.0 且 Config.AdvancedCooling = FALSE，则 Heat.Cycle 用作制冷脉宽调制的周期时间。 Heat.Cycle 具有保持性。 100000.0 ≥ Heat.Cycle > 0.0
Retain.CtrlParams.Heat.ControlZone	REAL	3.402822e+38	有效的加热控制区宽度 Heat.ControlZone = 3.402822e+38 时，加热过程将关闭控制区。 仅在选择 PIDSelfTune.SUT.TuneRuleHeat = 2 作为参数计算方法时，才会在预调节加热或预调节加热和制冷期间自动设置 Heat.ControlZone。 对于已禁用制冷输出的控制器 (Config.ActivateCooling = FALSE) 或已激活制冷输出和制冷系数的控制器 (Config.AdvancedCooling = FALSE)，控制区介于 Setpoint – Heat.ControlZone 和 Setpoint + Heat.ControlZone 之间呈对称分布。 对于已激活制冷输出和 PID 参数切换的控制器（Config.ActivateCooling = TRUE，Config.AdvancedCooling = TRUE），控制区介于 Setpoint – Heat.ControlZone 和 Setpoint + Cool.ControlZone 之间。 Heat.ControlZone 具有保持性。 Heat.ControlZone > 0.0

变量	数据类型	默认值	说明
Retain.CtrlParams.Heat.DeadZone	REAL	0.0	有效的加热死区宽度（请参见 PID 参数 (页 190)） Heat.DeadZone = 0.0 时，将关闭加热过程的死区。 Heat.DeadZone 既不会自动进行设置，也不会在调节期间进行调整。必须手动对 Heat.DeadZone 进行正确组态。 启用死区时，结果可能是永久控制偏差（设定值与过程值之间的偏差）。这可能对精确调节产生负面影响。 对于已禁用制冷输出的控制器 (Config.ActivateCooling = FALSE) 或已激活制冷输出和制冷系数的控制器 (Config.AdvancedCooling = FALSE)，死区介于 Setpoint – Heat.DeadZone 和 Setpoint + Heat.DeadZone 之间并且呈对称分布。 对于已激活制冷输出和 PID 参数切换的控制器（Config.ActivateCooling = TRUE，Config.AdvancedCooling = TRUE），死区介于 Setpoint – Heat.DeadZone 和 Setpoint + Cool.DeadZone 之间。 Heat.DeadZone 具有保持性。 Heat.DeadZone ≥ 0.0
Retain.CtrlParams.Cool.Gain	REAL	1.0	有效的制冷比例增益 Cool.Gain 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 Cool.Gain ≥ 0.0
Retain.CtrlParams.Cool.Ti	REAL	20.0	有效的制冷积分作用时间（以秒为单位） Cool.CtrlParams.Ti = 0.0 时，制冷过程将关闭积分作用。 Cool.Ti 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 100000.0 ≥ Cool.Ti ≥ 0.0

变量	数据类型	默认值	说明
Retain.CtrlParams.Cool.Td	REAL	0.0	有效的制冷微分作用时间（以秒为单位） Cool.CtrlParams.Td = 0.0 时，制冷过程将关闭微分作用。 Cool.Td 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 100000.0 ≥ Cool.Td ≥ 0.0
Retain.CtrlParams.Cool.TdFiltRatio	REAL	0.2	有效的制冷微分延时系数 微分延迟系数用于延迟微分作用的生效。 微分延迟 = 微分作用时间 × 微分延迟系数 • 0.0：微分作用仅在一个周期内有效，因此几乎不产生影响。 • 0.5：实践证明，该值对具有一个主时间常数的受控系统很有效。 • > 1.0：系数越大，微分作用的生效时间延迟越久。 Cool.TdFiltRatio 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 Cool.TdFiltRatio ≥ 0.0
Retain.CtrlParams.Cool.PWeighting	REAL	1.0	有效的制冷比例作用的权重 改变设定值有可能削弱比例作用。 允许使用 0.0 到 1.0 之间的值。 • 1.0：应对设定值变化的比例作用完全有效 • 0.0：应对设定值变化的比例作用无效 当过程值变化时，比例作用始终完全有效。 Cool.PWeighting 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 1.0 ≥ Cool.PWeighting ≥ 0.0

变量	数据类型	默认值	说明
Retain.CtrlParams.Cool.DWeighting	REAL	1.0	有效的制冷微分作用的权重 微分作用随着设定值的变化而减弱。 允许使用 0.0 到 1.0 之间的值。 • 1.0：设定值变化时微分作用完全有效 • 0.0：设定值变化时微分作用不生效 当过程值变化时，微分作用始终完全有效。 Cool.DWeighting 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 1.0 ≥ Cool.DWeighting ≥ 0.0
Retain.CtrlParams.Cool.Cycle	REAL	1.0	有效的制冷 PID 算法的采样时间（以秒为单位） 在调节期间计算 CtrlParams.Cool.Cycle 并将其舍入为 CycleTime. 的整数倍。 如果 Config.Output.Cool.PwmPeriode = 0.0 且 Config.AdvancedCooling = TRUE，则 Cool.Cycle 用作制冷脉宽调制的周期时间。 如果 Config.Output.Cool.PwmPeriode = 0.0 且 Config.AdvancedCooling = FALSE，则 Heat.Cycle 用作制冷脉宽调制的周期时间。 Cool.Cycle 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 100000.0 ≥ Cool.Cycle > 0.0

变量	数据类型	默认值	说明
Retain.CtrlParams.Cool.ControlZone	REAL	3.402822e+38	有效的制冷控制区宽度 Cool.ControlZone = 3.402822e+38 时，制冷过程将关闭控制区。 仅在选择 PIDSelfTune.SUT.TuneRuleCool = 2 作为参数计算方法时，才会在预调节制冷或预调节加热和制冷期间自动设置 Cool.ControlZone。 Cool.ControlZone 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 Cool.ControlZone > 0.0
Retain.CtrlParams.Cool.DeadZone	REAL	0.0	有效的制冷死区宽度（请参见 PID 参数 (页 190)） Cool.DeadZone = 0.0 时，将关闭制冷过程的死区。 Cool.DeadZone 既不会自动进行设置，也不会在调节期间进行调整。必须手动对 Cool.DeadZone 进行正确组态。 启用死区时，结果可能是永久控制偏差（设定值与过程值之间的偏差）。这可能对精确调节产生负面影响。 Cool.DeadZone 具有保持性。 仅在激活制冷输出和 PID 参数切换时（Config.ActivateCooling = TRUE 且 Config.AdvancedCooling = TRUE）有效。 Cool.DeadZone ≥ 0.0

说明

请在“未激活”模式下更改本表列出的变量，以防 PID 控制器出现故障。

参见

PID_Temp ActivateRecoverMode 变量 (页 500)

PID_Temp 警告变量 (页 502)

使用 PID_Temp 的多区域控制 (页 217)

8.3.3.7 PID_Temp 状态和模式参数

参数的相关性

State 参数显示了 PID 控制器的当前工作模式。您无法更改 State 参数。

当 ModeActivate 出现上升沿时，PID_Temp 将切换到保存在 Mode 输入/输出参数中的工作模式。

如果针对加热或制冷进行调节，则通过 Heat.EnableTuning 和 Cool.EnableTuning 指定预调节和精确调节。

CPU 通电或从 Stop 切换到 RUN 模式时，PID_Temp 将以保存在 Mode 参数中的工作模式启动。要使 PID_Temp 保持在“未激活”模式下，应设置 RunModeByStartup = FALSE。

值的含义

State / Mode	工作模式说明
0	**未激活** 在“未激活”模式下输出下列输出值： • 输出 0.0 作为 PID 输出值 (PidOutputSum) • 输出 0.0 作为加热输出值 (OutputHeat) 和制冷输出值 (OutputCool) • 输出 0 作为加热的模拟量输出值 (OutputHeat_PER) 和制冷的模拟量输出值 (OutputCool_PER) • 输出 FALSE 作为加热的 PWM 输出值 (OutputHeat_PWM) 和制冷的 PWM 输出值 (OutputCool_PWM) 这与 Config.Output.Heat 和 Config.Output.Cool 结构中的输出值限值和标定组态无关。
1	**预调节** 预调节功能可确定对输出值跳变的过程响应，并搜索拐点。根据受控系统的最大上升速率与死时间计算 PID 参数。可在执行预调节和精确调节时获得最佳 PID 参数。 PID_Temp 可根据组态提供不同的预调节类型： • 预调节加热： 加热输出值输出跳变，计算加热过程的 PID 参数（Retain.CtrlParams.Heat 结构），然后在自动模式下控制到设定值。 如果过程行为很大程度上取决于工作点，则可使用 PIDSelfTune.SUT.AdaptDelayTime 激活在达到设定值时调整延迟时间功能。 • 预调节加热和制冷： 加热输出值输出跳变。只要过程值接近设定值，制冷输出值便输出跳变。同时计算加热（Retain.CtrlParams.Heat 结构）和制冷（Retain.CtrlParams.Cool 结构）的 PID 参数。然后，在自动模式下控制到设定值。 如果过程行为很大程度上取决于工作点，则可使用 PIDSelfTune.SUT.AdaptDelayTime 激活在达到设定值时调整延迟时间功能。 由于冷执行器与加热执行器存在效果差异，调节过程中是否同时运行加热输出和制冷输出可能影响调节质量。这可以通过 PIDSelfTune.SUT.CoolingMode 来指定。 • 预调节制冷： 制冷输出值输出跳变，计算制冷的 PID 参数 (Struktur Retain.CtrlParams.Cool)。然后，在自动模式下控制到设定值。 如果要调节加热和制冷过程的 PID 参数，先后使用“预调节加热”(Pretuning heating) 和“预调节制冷”(Pretuning cooling) 与单独使用“预调节加热和制冷”(Pretuning heating and cooling) 相比，可获得更好的控制响应。但是，分两个步骤进行预调节耗费的时间较长。

State / Mode	工作模式说明
1	**预调节的常规要求：** • 已在循环中断 OB 中调用 PID_Temp 指令。 • 未激活 (State = 0)、手动模式 (State = 4) 或自动模式 (State = 3) • ManualEnable = FALSE • Reset = FALSE • 设定值和过程值均在组态的限值范围内。 **预调节加热的相关要求：** • Heat.EnableTuning = TRUE • Cool.EnableTuning = FALSE • 过程值不能过于接近设定值。 |Setpoint - Input| > 0.3 * |Config.InputUpperLimit - Config.InputLowerLimit| 且 |Setpoint - Input| > 0.5 * |Setpoint| • 设定值大于过程值。 Setpoint > Input **预调节加热和制冷的相关要求：** • Heat.EnableTuning = TRUE • Cool.EnableTuning = TRUE • 已激活制冷输出 (Config.ActivateCooling = TRUE)。 • 已激活 PID 参数切换 (Config.AdvancedCooling = TRUE)。 • 过程值不能过于接近设定值。 |Setpoint - Input| > 0.3 * |Config.InputUpperLimit - Config.InputLowerLimit| 且 |Setpoint - Input| > 0.5 * |Setpoint| • 设定值大于过程值。 Setpoint > Input

State / Mode	工作模式说明
1	**预调节制冷的相关要求：** • Heat.EnableTuning = FALSE· • Cool.EnableTuning = TRUE· • 已激活制冷输出 (Config.ActivateCooling = TRUE)。 • 已激活 PID 参数切换 (Config.AdvancedCooling = TRUE)。 • 已成功执行“预调节加热”或“预调节加热和制冷”(PIDSelfTune.SUT.ProcParHeatOk = TRUE)，在可能情况下请使用同一设定值。 • 过程值必须接近设定值。 |Setpoint - Input| < 0.05 * |Config.InputUpperLimit - Config.InputLowerLimit| 过程值越稳定，PID 参数就越容易计算，结果的精度也会越高。只要过程值的上升速率明显高于噪声，就可以容忍过程值的噪声。处于“未激活”或“手动模式”工作模式时就很可能出现这种情况。 设定值在变量 CurrentSetpoint 中冻结。出现以下情况时，调节将取消： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel 可通过 PIDSelfTune.SUT.TuneRuleHeat 和 PIDSelfTune.SUT.TuneRuleCool 分别为加热和制冷指定 PID 参数的计算方法。 重新计算 PID 参数之前，这些参数将以 CtrlParamsBackUp 结构备份，并且可使用 LoadBackUp 重新激活。 预调节成功后，将切换到自动模式。 如果预调节未成功，则根据 ActivateRecoverMode 确定切换到哪种模式。 预调节阶段由 PIDSelfTune.SUT.State 来指示。

State / Mode	工作模式说明
2	**精确调节** 精确调节将使过程值出现恒定受限的振荡。将根据此振荡的幅度和频率为工作点调节 PID 参数。精确调节得出的 PID 参数通常比预调节得出的 PID 参数具有更好的主控和扰动特性。可在执行预调节和精确调节时获得最佳 PID 参数。 PID_Temp 将自动尝试生成大于过程值噪声的振荡。过程值的稳定性对精确调节的影响非常小。 PID_Temp 可根据组态提供不同的精确调节类型： • 精确调节加热： PID_Temp 使过程值出现振荡，加热输出值发生周期性变化，并计算加热过程的 PID 参数 (Struktur Retain.CtrlParams.Heat)。 • 精确调节制冷： PID_Temp 使过程值出现振荡，制冷输出值发生周期性变化，并计算制冷的 PID 参数 (Struktur Retain.CtrlParams.Cool)。
	加热/制冷控制器的临时调节偏移量 如果将 PID_Temp 用作加热/制冷控制器 (Config.ActivateCooling = TRUE)，则相应设定值对应的 PID 输出值 (PidOutputSum) 必须符合以下要求，这样才能使过程值出现振荡从而成功进行精确调节： • 精确调节加热的 PID 输出值为正 • 精确调节制冷的 PID 输出值为负 如果不满足上述要求，则可以为精确调节定义一个临时偏移量，以在具有相反效果的输出上输出： • 精确调节加热时的制冷输出偏移量 (PIDSelfTune.TIR.OutputOffsetCool)。 定义一个负制冷调节偏移量，且必须小于启动调节前相应设定值对应的稳态 PID 输出值 (PidOutputSum)。 • 精确调节制冷时的加热输出偏移量 (PIDSelfTune.TIR.OutputOffsetHeat)。 定义一个正加热调节偏移量，且必须大于启动调节前相应设定值对应的稳态 PID 输出值 (PidOutputSum)。 随后，由 PID 算法抵消指定的偏移量，从而使过程值保持为设定值。偏移高度允许对 PID 输出值进行相应调整从而使其满足上述要求。 为避免在定义偏移量后过程值过调较大，还可以分多步增大偏移量。 如果 PID_Temp 退出精确调节模式，将重置调节偏移量。

State / Mode	工作模式说明
2	精确调节制冷的偏移量定义示例： • 不指定偏移量： – 设定值 = 过程值 (ScaledInput) = 80°C – PID 输出值 (PidOutputSum) = 30.0 – 加热输出值 (OutputHeat) = 30.0 – 制冷输出值 (OutputCool) = 0.0 只通过制冷输出无法使过程值围绕设定值振荡。 此时无法执行精确调节。 • 指定加热输出的偏移量 (PIDSelfTune.TIR.OutputOffsetHeat) = 80.0 – Setpoint = 过程值 (ScaledInput) = 80°C – PID 输出值 (PidOutputSum) = -50.0 – 加热输出值 (OutputHeat) = 80.0 – 制冷输出值 (OutputCool) = -50.0 通过指定加热输出的偏移量，现在可以使用制冷输出使过程值围绕设定值振荡。 现在可以成功执行精确调节。 **精确调节的一般要求：** • 已在循环中断 OB 中调用 PID_Temp 指令。 • 不能被干扰。 • 设定值和过程值均在组态的限值范围内。 • 控制回路已稳定在工作点。过程值与设定值一致时，表明到达了工作点。 启用死区时，结果可能是永久控制偏差（设定值与实际值之间的偏差）。这可能对精确调节产生负面影响。 • ManualEnable = FALSE • Reset = FALSE • 自动模式 (State = 3)、未激活模式 (State = 0) 或手动模式 (State = 4)

State / Mode	工作模式说明
2	**精确调节加热的相关要求：** • Heat.EnableTuning = TRUE • Cool.EnableTuning = FALSE • 如果将 PID_Temp 组态为加热/制冷控制器 (Config.ActivateCooling = TRUE)，则在达到要开始调节 (PidOutputSum > 0.0)（请参见调节偏移量）的工作点时必须激活加热输出。 **精确调节制冷的相关要求：** • Heat.EnableTuning = FALSE • Cool.EnableTuning = TRUE • 已激活制冷输出 (Config.ActivateCooling = TRUE)。 • 已激活 PID 参数切换 (Config.AdvancedCooling = TRUE) • 在达到要开始调节 (PidOutputSum < 0.0)（请参见调节偏移）的工作点时必须激活制冷输出。
	精确调节过程由启动模式决定： • 自动模式 (State = 3) 且 PIDSelfTune.TIR.RunIn = FALSE（默认） 如果希望通过调节来改进现有 PID 参数，请在自动模式下启动精确调节。 PID_Temp 将使用现有的 PID 参数控制系统，直到控制回路已稳定并且精确调节的要求得到满足为止。之后才会启动精确调节。 • 未激活 (State = 0)、手动模式 (State = 4) 或 PIDSelfTune.TIR.RunIn = TRUE 的自动模式 (State = 3) 系统将尝试利用最小或最大输出值达到设定值： – 在精确调节加热时，使用最小或最大加热输出值。 – 在精确调节制冷时，使用最小或最大制冷输出值。 这可能会增加超调量。精确调节将在达到设定值时启动。 如果无法达到设定值，PID_Temp 不会自动中止调节。

State / Mode	工作模式说明
2	设定值在变量 CurrentSetpoint 中冻结。出现以下情况时，调节将取消： • Setpoint > CurrentSetpoint + CancelTuningLevel 或 • Setpoint < CurrentSetpoint - CancelTuningLevel 可通过 PIDSelfTune.TIR.TuneRuleHeat 和 PIDSelfTune.TIR.TuneRuleCool 分别为加热和制冷指定 PID 参数的计算方法。 重新计算 PID 参数之前，这些参数将以 CtrlParamsBackUp 结构备份，并且可使用 LoadBackUp 重新激活。 精确调节成功后，控制器将切换到自动模式。 如果精确调节未成功，则根据 ActivateRecoverMode 确定切换到哪种模式。 “精确调节”阶段由 PIDSelfTune.TIR.State 来指示。
3	**自动模式** 在自动模式下，PID_Temp 会按照指定的参数来更正受控系统。 如果满足下列要求之一，则控制器将切换到自动模式： • 预调节已成功完成 • 精确调节已成功完成 • Mode 输入/输出参数更改为值 3 并且 ModeActivate 出现上升沿。 从自动模式到手动模式的切换只有在调试编辑器中执行时，才是无扰动的。 自动模式下会考虑 ActivateRecoverMode 变量。
4	**手动模式** 在手动模式下，在 ManualValue 参数中指定手动 PID 输出值。在应用此手动值后，相关输出上输出的加热或制冷值取决于输出标定的组态情况。 还可以使用 ManualEnable = TRUE 来激活该工作模式。建议只使用 Mode 和 ModeActivate 更改工作模式。 从手动模式到自动模式的切换是无扰动的。 手动模式下会考虑 ActivateRecoverMode 变量。

State / Mode	工作模式说明
5	**含错误监视功能的替代输出值** 控制算法取消激活。SetSubstituteOutput 变量决定此工作模式中输出哪个 PID 输出值 (PidOutputSum)。 • SetSubstituteOutput = FALSE：上一个有效 PID 输出值 • SetSubstituteOutput = TRUE：替代输出值 (SubstituteOutput) 无法使用 Mode = 5 激活该工作模式。 如果满足以下所有条件，出现错误时会激活该工作模式而不激活“未激活”工作模式。 • 自动模式 (State = 3) • ActivateRecoverMode = TRUE • 已出现一个或多个错误，并且 ActivateRecoverMode 生效。 当错误不再处于未决状态时，PID_Temp 切换回自动模式。

ENO 特性

如果 State = 0，那么 ENO = FALSE。

如果 State ≠ 0，那么 ENO = TRUE。

在调试期间自动切换工作模式

预调节或精确调节成功后，将激活自动模式。下表显示了成功预调节期间 Mode 和 State 的更改方式。

周期编号	Mode	State	操作
0	4	4	设置 Mode = 1
1	1	4	设置 ModeActivate = TRUE
1	4	1	State 的值保存在模式参数中 启动预调节功能
n	4	1	预调节已成功完成
n	3	3	启动自动模式

PID_Temp 将在出现错误时自动切换工作模式。

下表显示了出现错误的预调节期间 Mode 和 State 的更改方式。

周期编号	Mode	State	操作
0	4	4	设置 Mode = 1
1	1	4	设置 ModeActivate = TRUE
1	4	1	State 的值保存在模式参数中 启动预调节功能
n	4	1	取消预调节
n	4	4	启动手动模式

如果 ActivateRecoverMode = TRUE，将激活保存在 Mode 参数中的工作模式。启动预调节或精确调节时，PID_Temp 已将 State 的值保存到 Mode 输入/输出参数中。也就是说，PID_Temp 将切换到启动调节时的模式。

如果 ActivateRecoverMode = FALSE，系统将切换到“未激活”工作模式。

参见

PID_Temp 的输出参数 (页 441)

PID_Temp V2 的输入/输出参数 (页 444)

8.3.3.8 PID_Temp ErrorBits 参数

如果多个错误同时处于待决状态，将通过二进制加法显示 ErrorBits 的值。例如，显示 ErrorBits = 0000003h 表示错误 0000001h 和 0000002h 同时处于待决状态。

ErrorBits (DW#16#...)	说明
0000000	没有任何错误。
0000001	参数“Input”超出了过程值限值的范围。 • Input > Config.InputUpperLimit 或 • Input < Config.InputLowerLimit 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持自动模式。 如果在错误发生前手动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持手动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。
0000002	参数“Input_PER”的值无效。请检查模拟量输入是否有处于未决状态的错误。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 输出组态的替代输出值。当错误不再处于未决状态时，PID_Temp 切换回自动模式。 如果在错误发生前手动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持手动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。
0000004	精确调节期间出错。过程值无法保持振荡状态。 如果将 PID_Temp 用作加热-制冷控制器 (Config.ActivateCooling = TRUE)，为产生实际值振荡，设定值对应的 PID 输出值 (PidOutputSum) 必须： • 为正值才能进行加热过程的精确调节， • 为负值才能进行制冷过程的精确调节 如果未满足此要求，可使用调节偏移量（ PIDSelfTune.TIR.OutputOffsetCool 和 PIDSelfTune.TIR.OutputOffsetHeat 变量），请参见精确调节 (页 202)。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。

ErrorBits (DW#16#...)	说明
0000008	预调节启动时出错。过程值过于接近设定值或大于设定值。启动精确调节。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
0000010	调节期间设定值发生更改。 可在 CancelTuningLevel 变量中设置允许的设定值波动。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
0000020	精确调节期间不允许预调节。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 保持精确调节模式。
0000040	预调节期间出错。制冷无法减小过程值。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
0000100	精确调节期间的错误导致生成无效参数。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
0000200	参数“Input”的值无效：值的数字格式无效。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 输出组态的替代输出值。当错误不再处于未决状态时，PID_Temp 切换回自动模式。 如果在错误发生前手动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持手动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。
0000400	输出值计算失败。请检查 PID 参数。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 输出组态的替代输出值。当错误不再处于未决状态时，PID_Temp 切换回自动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。

ErrorBits (DW#16#...)	说明
0000800	采样时间错误：在循环中断 OB 的采样时间内没有调用 PID_Temp。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持自动模式。 如果在错误发生前手动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持手动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。 如果在使用 PLCSIM 进行仿真期间出现该错误，请参见使用 PLCSIM 仿真 PID_Temp (页 225)下的说明。
0001000	“Setpoint”参数或“SubstituteSetpoint”的值无效：值的数字格式无效。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 输出组态的替代输出值。当错误不再处于未决状态时，PID_Temp 切换回自动模式。 如果在错误发生前手动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持手动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。
0010000	ManualValue 参数的值无效。值的数字格式无效。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 保持手动模式并使用 SubstituteOutput 作为 PID 输出值。在 ManualValue 中指定一个有效值后，PID_Temp 会立即将其用作 PID 输出值。
0020000	变量 SubstituteOutput 的值无效。值的数字格式无效。 PID_Temp 保持“含错误监视功能的替代输出值”模式或手动模式，将加热 PID 输出值的下限 (Config.Output.Heat.PidLowerLimit) 用作 PID 输出值。 在 SubstituteOutput 中指定一个有效值后，PID_Temp 会立即将其用作 PID 输出值。
0040000	Disturbance 参数的值无效。值的数字格式无效。 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 Disturbance 将设置为零。PID_Temp 保持自动模式。 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。如果当前阶段中的 Disturbance 对输出值无影响，则不会取消调节。

ErrorBits (DW#16#...)	说明
0200000	级联中的主控制器出错：Slaves 未处于自动模式，或已激活替代设定值，妨碍了主控制器的调节。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
0400000	在制冷过程处于激活状态时不允许对加热过程进行预调节。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
0800000	过程值必须接近设定值才能启动预调节制冷。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
1000000	调节启动时出错：Heat.EnableTuning 和 Cool.EnableTuning 未设置或与组态不匹配。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
2000000	预调节制冷要求成功完成了预调节加热。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
4000000	启动精确调节时出错：Heat.EnableTuning 和 Cool.EnableTuning 不能同时设置。 如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 取消调节并切换到 Mode 参数中保存的工作模式。
8000000	计算 PID 参数时出错，导致生成无效的参数。 无效参数被丢弃，原始 PID 参数保持不变。 我们可区别以下情况： • 如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持自动模式。 • 如果在错误发生前手动模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 保持手动模式。 • 如果在错误发生前预调节或精确调节模式已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。

8.3.3.9 PID_Temp ActivateRecoverMode 变量

ActivateRecoverMode 变量决定错误响应方式。 Error 参数指示是否存在错误处于未决状态。 当错误不再处于未决状态时，Error = FALSE。 ErrorBits 参数显示发生的具体错误。

自动模式和手动模式

注意
您的系统可能已损坏。 如果 ActivateRecoverMode = TRUE，则 PID_Temp 保持自动模式或手动模式，即使出现错误或超过过程限值也是如此。 这可能损坏您的系统。 必须组态受控系统在出现错误时如何作出响应以避免系统损坏。

ActivateRecoverMode	说明
FALSE	PID_Temp 将在出现错误时切换到“未激活”模式。只能通过 Reset 的下降沿或 ModeActivate 的上升沿激活控制器。
TRUE	**自动模式** 如果在自动模式下频繁出现错误，则该设置会对控制响应产生负面影响，因为每次出错时，PID_Temp 都会在计算的 PID 输出值和替代输出值之间切换。 这种情况下，检查 ErrorBits 参数并消除错误原因。 如果发生下列一个或多个错误且在发生错误前自动模式已激活，则 PID_Temp 保持自动模式： • 0000001h： 参数“Input”超出了过程值限值的范围。 • 0000800h：采样时间错误 • 0040000h：Disturbance 参数的值无效。 • 8000000h： 计算 PID 参数期间出错

ActivateRecoverMode	说明
TRUE	如果发生下列一个或多个错误且在发生错误前自动模式已激活，则 PID_Temp 切换到“含错误监视功能的替代输出值”模式： • 0000002h：Input_PER 参数的值无效。 • 0000200h：Input 参数的值无效。 • 0000400h：输出值计算失败。 • 0001000h： Setpoint 参数或 SubstituteSetpoint 的值无效。 当错误不再处于未决状态时，PID_Temp 切换回自动模式。 如果在“含错误监视功能的替代输出值”模式下发生以下错误，则只要错误待决，PID_Temp 就会将 PID 输出值设为 Config.Output.Heat.PidLowerLimit： • 0020000h：变量 SubstituteOutput 的值无效。值的数字格式无效。 此行为与 SetSubstituteOutput 无关。
	手动模式 如果发生一个或多个错误且在错误发生前手动模式已激活，PID_Temp 将保持手动模式。 如果在手动模式下发生以下错误，则只要此错误未决，PID_Temp 就会将 PID 输出值设为 SubstituteOutput: • 0010000h：ManualValue 参数的值无效。 值的数字格式无效。 如果错误 0010000h 在手动模式未决，又发生以下错误，则只要此错误待决，PID_Temp 就会将 PID 输出值设为 Config.Output.Heat.PidLowerLimit： • 0020000h：变量 SubstituteOutput 的值无效。值的数字格式无效。 此行为与 SetSubstituteOutput 无关。

预调节和精确调节

ActivateRecoverMode	说明
FALSE	PID_Temp 将在出现错误时切换到“未激活”模式。只能通过 Reset 的下降沿或 ModeActivate 的上升沿激活控制器。
TRUE	如果发生以下错误，PID_Temp 将保持激活模式： • 0000020h： 精确调节期间不允许预调节。 以下错误将被忽略： • 0010000h： ManualValue 参数的值无效。 • 0020000h：变量 SubstituteOutput 的值无效。 出现其它错误时，PID_Temp 将取消调节并切换到启动调节时的模式。

8.3.3.10 PID_Temp 警告变量

如果多个警告同时处于待决状态，将通过二进制加法显示 Warning 变量值。例如，如果显示警告 0000003h，警告 0000001h 和 0000002h 将同时处于待决状态。

Warning (DW#16#....)	说明
0000000	无警告处于待决状态。
0000001	预调节期间未发现拐点。
0000004	设定值被限制为组态的限值。
0000008	在所选计算方法中未定义所有必要的受控系统属性。因而，PID 参数是使用 TIR.TuneRuleHeat = 3 方法或 TIR.TuneRuleCool = 3 计算的。
0000010	由于 Reset = TRUE 或 ManualEnable = TRUE，无法更改工作模式。
0000020	调用 OB 的循环时间会限制 PID 算法的采样时间。 通过缩短 OB 循环时间来改进结果。
0000040	过程值超出其警告限值之一。
0000080	Mode 的值无效。工作模式不变。
0000100	手动值被限定在 PID 输出值的限值范围内。
0000200	不支持指定的调节规则。不计算任何 PID 参数。
0001000	无法达到替代输出值，因为它超出了输出值限值。
0004000	不支持选择指定的加热和/或制冷的输出值。 仅使用 OutputHeat 或 OutputCool 输出。
0008000	PIDSelfTune.SUT.AdaptDelayTime 的值无效。将使用默认值 0。
0010000	PIDSelfTune.SUT.CoolingMode 的值无效。将使用默认值 0。
0020000	用作主控制器（Config.Cascade.IsMaster 变量）的控制器不支持制冷激活（Config.ActivateCooling 变量）。PID_Temp 用作加热控制器。 将变量 Config.ActivateCooling 设为 FALSE。
0040000	Retain.CtrlParams.Heat.Gain，Retain.CtrlParams.Cool.Gain 序列 Config.CoolFactor 的值无效。 PID_Temp 仅支持比例增益（加热和制冷）和制冷系数使用正值。自动模式保持激活，且 PID 输出值为 0.0。积分分量停止。

只要消除了警告原因或使用有效参数重复操作后，以下警告就会立即消失：

- 0000001h
- 0000004h
- 0000008h
- 0000040h
- 0000100h

所有其它警告均在 Reset 或 ErrorAck 出现上升沿时清除。

8.3.3.11 PwmPeriode 变量

如果使用 OutputHeat_PWM 或 OutputCool_PWM 时 PID 算法采样时间 (Retain.CtrlParams.Heat.Cycle 或 Retain.CtrlParams.Heat.Cycle) 和脉宽调制的周期时间过大，则可在 Config.Output.Heat.PwmPeriode 或 Config.Output.Cool.PwmPeriode 参数中定义存在偏差的稍短周期时间来改善过程值的平滑度。

OutputHeat_PWM 中的脉宽调制时间

OutputHeat_PWM 输出中的 PWM 时间取决于 Config.Output.Heat.PwmPeriode：

- Heat.PwmPeriode = 0.0（默认值）

  加热的 PID 算法的采样时间 (Retain.CtrlParams.Heat.Cycle) 用作 PWM 的周期时间。

- Heat.PwmPeriode > 0.0

  该值将舍入为 PID_Temp 采样时间 (CycleTime.Value) 的整数倍并用作 PWM 的周期时间。

  该值必须满足以下条件：

  – Heat.PwmPeriode ≤ Retain.CtrlParams.Heat.Cycle

  – Heat.PwmPeriode > Config.Output.Heat.MinimumOnTime

  – Heat.PwmPeriode > Config.Output.Heat.MinimumOffTime

OutputCool_PWM 中的脉宽调制时间

OutputCool_PWM 输出中的 PWM 的周期时间取决于 Config.Output.Cool.PwmPeriode 和加热/制冷的方法：

- Cool.PwmPeriode = 0.0 且制冷系数 (Config.AdvancedCooling = FALSE)：

  加热的 PID 算法的采样时间 (Retain.CtrlParams.Heat.Cycle) 用作 PWM 的周期时间。

- Cool.PwmPeriode = 0.0 且 PID 参数切换 (Config.AdvancedCooling = TRUE)：

  制冷的 PID 算法的采样时间 (Retain.CtrlParams.Cool.Cycle) 用作 PWM 的周期时间。

- Cool.PwmPeriode > 0.0:

  该值将舍入为 PID_Temp 采样时间 (CycleTime.Value) 的整数倍并用作 PWM 的周期时间。

  该值必须满足以下条件：

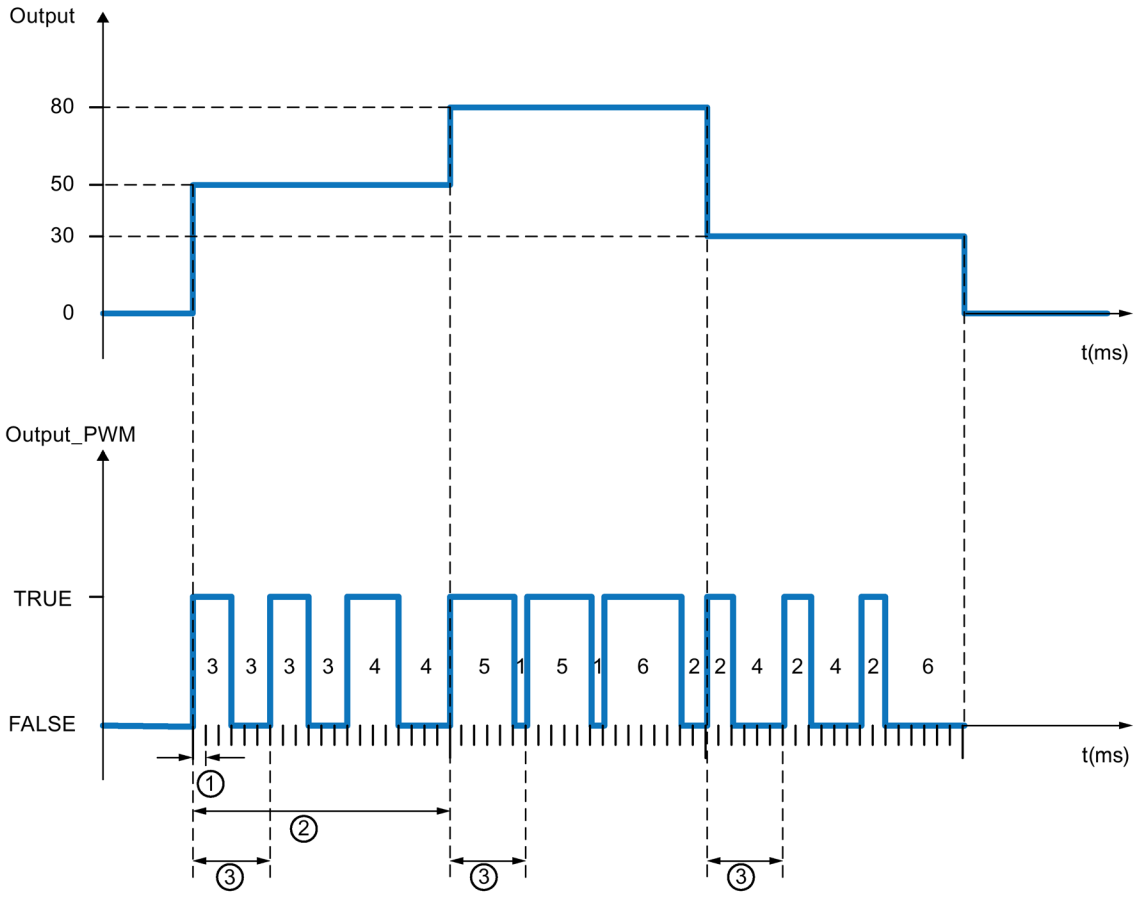
  – Cool.PwmPeriode ≤ Retain.CtrlParams.Cool.Cycle 或 Retain.CtrlParams.Heat.Cycle
  – Cool.PwmPeriode > Config.Output.Cool.MinimumOnTime
  – Cool.PwmPeriode > Config.Output.Cool.MinimumOffTime

Config.Output.Cool.PwmPeriode 仅在制冷输出激活 (Config.ActivateCooling =TRUE) 时有效。

使用 PwmPeriode 时，PWM 输出信号的精度由 PwmPeriode 与 PID_Temp 采样时间（OB 的周期时间）的关系决定。PwmPeriode 至少应为 PID_Temp 采样时间的 10 倍。

如果 PID 算法的采样时间不是 PwmPeriode 的整数倍，则在 PID 算法采样时间内 PWM 的最后一个周期都将相应延长。

OutputHeat_PWM 的示例



① PID_Temp 采样时间 = 100.0 ms（调用循环中断 OB 的周期时间，CycleTime.Value 变量）
② PID 算法采样时间 = 2000.0 ms（Retain.CtrlParams.Heat.Cycle 变量）
③ 加热的 PWM 时间 = 600.0 ms（Config.Output.Heat.PwmPeriode 变量）

8.3.3.12 IntegralResetMode 变量

IntegralResetMode 变量用于确定如何预分配积分作用 PIDCtrl.IOutputOld：

- 从“未激活”工作模式切换到“自动模式”时
- 参数 Reset 出现 TRUE -> FALSE 沿并且参数 Mode = 3 时

只有在激活了积分作用时，该设置才会在一个周期内有效（Retain.CtrlParams.Heat.Ti 和 Retain.CtrlParams.Cool.Ti > 0.0 变量）。

IntegralReset Mode	说明
0	平滑 已经预分配了 PIDCtrl.IOutputOld 的值，因此可以实现无扰动切换，即通过输出值 = 0.0（参数 PidOutputSum）启动“自动模式”，并且无论是否存在控制偏差（设定值 – 过程值），输出值都不会发生跳变。
1	删除 如果使用该选项，我们建议将比例作用的权重（Retain.CtrlParams.Heat.PWeighting 和 Retain.CtrlParams.Cool.PWeighting 变量）设为 1.0。 PIDCtrl.IOutputOld 的值已删除。任何控制偏差都会导致 PID 输出值发生跳变。输出值的跳变方向取决于有效的比例作用权重（Retain.CtrlParams.Heat.PWeighting 和 Retain.CtrlParams.Cool.PWeighting 变量）以及控制偏差： • 有效的比例作用权重 = 1.0： 输出值跳变与控制偏差的符号相同。 示例：如果过程值小于设定值（正控制偏差），则 PID 输出值会跳变至正值。 • 有效的比例作用权重 < 1.0： 对于较大的控制偏差，PID 输出值跳变与控制偏差的符号相同。 示例：如果过程值远远小于设定值（正控制偏差），则 PID 输出值会跳变至正值。 对于较小的控制偏差，PID 输出值跳变与控制偏差的符号不同。 示例：如果过程值略小于设定值（正控制偏差），则 PID 输出值会跳变至负值。通常不希望出现这种情况，因为这会导致控制偏差暂时增大。 组态的比例作用权重越小，控制偏差就越大，以便接收具有相同符号的 PID 输出值跳变。 如果使用该选项，我们建议将比例作用的权重（Retain.CtrlParams.Heat.PWeighting 和 Retain.CtrlParams.Cool.PWeighting 变量）设为 1.0。否则，可能会出现针对小控制偏差所说明的不良行为。您还可以使用 IntegralResetMode = 4。该选项确保 PID 输出值跳变与控制偏差的符号相同，无论组态的比例作用权重和控制偏差为何值。

IntegralReset Mode	说明
2	保持 PIDCtrl.IOutputOld 的值未更改。您可以使用用户程序定义一个新值。
3	预分配 自动预分配 PIDCtrl.IOutputOld 的值，如同在上一周期中 PidOutputSum = OverwriteInitialOutputValue。
4	类似于设定值更改（仅适用于版本 1.1 及更高版本的 PID_Temp） 自动预分配 PIDCtrl.IOutputOld 的值，以便使 PID 输出值跳变与自动模式下设定值从当前过程值更改为当前设定值时的 PI 控制器的行为类似。 任何控制偏差都会导致 PID 输出值发生跳变。PID 输出值跳变与控制偏差的符号相同。示例：如果过程值小于设定值（正控制偏差），则 PID 输出值会跳变至正值。这与组态的比例作用权重和控制偏差无关。

如果为 IntegralResetMode 分配的值不在有效值范围内，PID_Temp 的行为将与 IntegralResetMode 预分配时的情况相同：

- PID_Temp V1.0 及之前的版本：IntegralResetMode = 1
- PID_Temp 自 V1.1 起的版本：IntegralResetMode = 4

8.3.4 PID_Temp V1 的 CPU 处理时间和存储器要求

CPU 处理时间

自版本 1.0 起的 PID_Temp 工艺对象典型 CPU 处理时间（取决于 CPU 类型）。

CPU	典型 CPU 处理时间 (PID_Temp V1)
CPU 1211C ≥ V4.1	580 µs
CPU 1215C ≥ V4.1	580 µs
CPU 1217C ≥ V4.1	580 µs
CPU 1505S ≥ V1.0	50 µs
CPU 1510SP-1 PN ≥ V1.7	130 µs
CPU 1511-1 PN ≥ V1.7	130 µs
CPU 1512SP-1 PN ≥ V1.7	130 µs
CPU 1516-3 PN/DP ≥ V1.7	75 µs
CPU 1518-4 PN/DP ≥ V1.7	6 µs

存储器要求

自版本 V1.0 起的 PID_Temp 工艺对象背景数据块的存储器要求。

	PID_Temp V1 背景数据块的 存储器要求
装载存储器要求	约 17000 个字节
总工作存储器要求	1280 个字节
保持性工作存储器要求	100 个字节

8.4 PID 基本功能

8.4.1 CONT_C

8.4.1.1 CONT_C 说明

在 SIMATIC S7 自动化系统中可使用 CONT_C 指令控制具有连续输入和输出变量的工艺过程。 可分配参数来启用或禁用 PID 控制器的子功能并使其适应该过程。 除了设定值和过程值分支中的功能外，该指令还实现了一个完整的 PID 控制器，该控制器具有连续的输出值输出，并且允许手动影响输出值。

应用

可以使用该控制器作为 PID 固定设定值控制器，或在多回路控制系统中作为级联、混合或比率控制器。 控制器的功能基于带有模拟信号的采样控制器的 PID 控制算法，必要时还可按以下方法进行扩展：增加一个脉冲整形器环节，以便为带有比例执行器的两位或三位控制器生成脉宽调制的输出信号。

调用

CONT_C 指令具有一个初始化例程，在设置输入参数 COM_RST = TRUE 时将运行该例程。 初始化过程中，积分作用被设置为初始化值 I_ITVAL。 所有信号输出都被设置为零。完成初始化例程后，必须设置 COM_RST = FALSE。

只有以固定时间间隔调用块时，在控制块中计算的值才是正确的。 因此，应在循环中断 OB（OB 30 到 OB 38）中调用控制块。 在 CYCLE 参数中输入采样时间。

如果将指令 CONT_C 作为多重背景数据块调用，将不会创建任何工艺对象。没有参数分配接口或调试接口可用。 必须直接在多重背景数据块中为 CONT_C 分配参数,并通过监视表格进行调试。

错误信息

错误消息字 RET_VAL 不由块进行评估。

8.4.1.2 CONT_C 的工作原理

设定值分支

在输入 SP_INT 中输入浮点格式的设定值。

过程值分支

可以 I/O 或浮点格式输入过程值。 函数 CRP_IN 按照以下规则将 I/O 值 PV_PER 转换为浮点格式的值 -100 到 +100 %：

CRP_IN = PV_PER * 100 / 27648 的输出

函数 PV_NORM 根据以下规则标定 CRP_IN 的输出：

PV_NORM 的输出 = (CRP_IN 的输出) *PV_FAC + PV_OFF

PV_FAC 的默认值为 1，PV_OFF 的默认值为 0。

形成误差信号

设定值与过程值之间的差值是误差信号。 要抑制由于调节变量量化而产生的小幅持续振荡（例如，使用 PULSEGEN 进行脉宽调制时），可将误差信号应用于死区 (DEADBAND)。 DEADB_W = 0 时，死区关闭。

PID 算法

PID 算法作为位置算法运行。 比例、积分 (INT) 和微分 (DIF) 作用并行连接在一起，可以单独激活或禁用。 这样便可组态 P、PI、PD 和 PID 控制器。 也可以组态纯 I 控制器。

手动值处理

可以在手动模式和自动模式之间切换。 在手动模式下，调节变量被修正为手动选择的值。

积分作用 (INT) 内部设置为 LMN - LMN_P - DISV，微分作用 (DIF) 内部设置为 0 并同步。 因此，可以平滑地切换到自动模式。

调节值处理

可以使用 LMNLIMIT 函数将调节值限制为所选值。 输入变量超过限值时，报警位会给予指示。

函数 LMN_NORM 按照以下规则对 LMNLIMIT 的输出进行标准化：

LMN = (LMNLIMIT 的输出) * LMN_FAC + LMN_OFF

LMN_FAC 的默认值为 1，LMN_OFF 的默认值为 0。

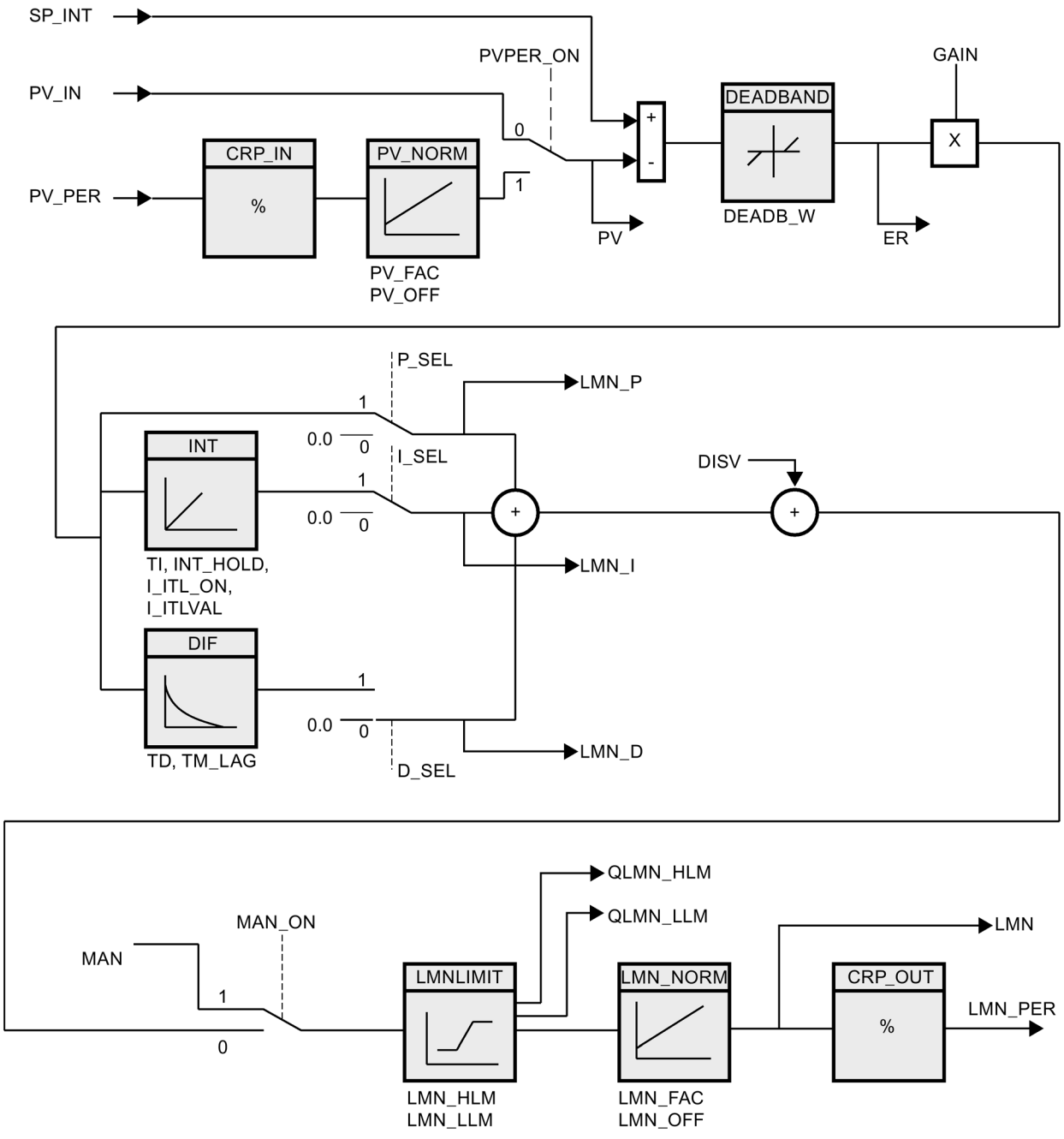
调节值也可以使用 I/O 格式。 函数 CRP_OUT 按照以下规则将 LMN 浮点值转换为 I/O 值：

LMN_PER = LMN * 27648 / 100

前馈控制

可在 DISV 输入中添加扰动变量。

8.4.1.3 CONT_C 方框图



8.4.1.4 输入参数 CONT_C

表格 8- 13

参数	数据类型	默认值	说明
COM_RST	BOOL	FALSE	该指令具有一个初始化例程，在对输入“重启”进行置位时将处理该例程。
MAN_ON	BOOL	TRUE	如果输入“启用手动模式”被置位，则控制回路会中断。 手动值将被设置为调节值。
PVPER_ON	BOOL	FALSE	如果要从 I/O 读取过程值，输入 PV_PER 必须与 I/O 互连，且输入“启用过程值 I/O”必须置位。
P_SEL	BOOL	TRUE	可在 PID 算法中单独开启或关闭 PID 作用。 置位输入“启用 P 作用”后，P 作用打开。
I_SEL	BOOL	TRUE	可在 PID 算法中单独开启或关闭 PID 作用。 置位输入“I 作用打开”后，I 作用打开。
INT_HOLD	BOOL	FALSE	可冻结积分作用的输出。 为此，必须置位输入“I 作用保持”。
I_ITL_ON	BOOL	FALSE	可在输入 I_ITLVAL 设置积分作用的输出。 为此，必须置位输入“设置 I 作用”。
D_SEL	BOOL	FALSE	可在 PID 算法中单独开启或关闭 PID 作用。 置位输入“启用 D 作用”后，D 作用打开。
CYCLE	TIME	T#1s	块调用之间的时间间隔必须恒定。 “采样时间”输入用于指定块调用之间的时间。 CYCLE >= 1ms
SP_INT	REAL	0.0	“内部设定值”输入用于指定设定值。 允许值从 -100 到 100 %，或者是物理变量 1)。
PV_IN	REAL	0.0	在“过程值输入”处，可以将参数分配给调试值，或者互连浮点格式的外部过程值。 允许值从 -100 到 100 %，或者是物理变量 1)。
PV_PER	WORD	W#16#0000	I/O 格式的过程值在“过程值 I/O”输入处与控制器互连。
MAN	REAL	0.0	“手动值”输入用于通过操作员界面功能设置手动值。 允许值从 -100 到 100 %，或者是物理变量 2)。
GAIN	REAL	2.0	“比例增益”输入用于指定控制器放大率。

参数	数据类型	默认值	说明
TI	TIME	T#20s	“积分时间”输入用于确定积分作用的时间响应。 TI >= CYCLE
TD	TIME	T#10s	“微分作用时间”输入用于确定微分作用的时间响应。 TD >= CYCLE
TM_LAG	TIME	T#2s	D 作用的时间滞后 D 作用算法包含延迟，用于在“D 作用的时间滞后”输入中延迟分配参数。 TM_LAG >= CYCLE/2
DEADB_W	REAL	0.0	将死区应用到系统偏差。 “死区宽度”输入用于确定死区的大小。 DEADB_W >= 0.0 (%) 或物理变量 1)
LMN_HLM	REAL	100.0	调节值始终限制在上限和下限之间。 “调节值的上限”输入用于指定上限。 允许实数值从 LMN_LLM 开始，或者是物理变量 2)。
LMN_LLM	REAL	0.0	调节值始终限制在上限和下限之间。 “调节值的下限”输入用于指定下限。 允许实数值最大为 LMN_HLM，或者是物理变量 2)。
PV_FAC	REAL	1.0	“过程值因子”输入与过程值相乘。 该输入用于标定过程值的范围。
PV_OFF	REAL	0.0	“过程值偏移量”输入与过程值相加。 该输入用于标定过程值的范围。
LMN_FAC	REAL	1.0	“调节值因子”输入与调节值相乘。 该输入用于标定调节值的范围。
LMN_OFF	REAL	0.0	“调节值偏移量”输入与过程值相加。 该输入用于标定调节值的范围。
I_ITLVAL	REAL	0.0	可在输入 I_ITL_ON 设置积分作用的输出。 将初始化值应用于输入“I 作用的初始化值”。 允许值从 -100.0 到 100.0 (%)，或者是物理变量 2)。
DISV	REAL	0.0	对于前馈控制，扰动变量与输入“扰动变量”互连。 允许值从 -100.0 到 100.0 (%)，或者是物理变量 2)。

1) 设定值和过程值分支中的参数具有相同的单位

2) 调节值分支中的参数具有相同的单位

8.4.1.5 CONT_C 输出参数

表格 8- 14

参数	数据类型	默认值	说明
LMN	REAL	0.0	有效“调节值”以浮点格式在“调节值”输出中输出。
LMN_PER	WORD	W#16#0000	I/O 格式的调节值在“调节值 I/O”输入中与控制器互连。
QLMN_HLM	BOOL	FALSE	调节值始终限制在上限和下限之间。 输出“达到调节值上限”表示已达到上限。
QLMN_LLM	BOOL	FALSE	调节值始终限制在上限和下限之间。 输出“达到调节值下限”表示已达到下限。
LMN_P	REAL	0.0	“P 作用”输出包含调节变量的比例作用。
LMN_I	REAL	0.0	“I 作用”输出包含调节变量的积分作用。
LMN_D	REAL	0.0	“D 作用”输出包含调节变量的微分作用。
PV	REAL	0.0	有效的过程值在“过程值”输出中输出。
ER	REAL	0.0	在“误差信号”输出中输出有效系统偏差。

8.4.2 CONT_S

8.4.2.1 CONT_S 说明

CONT_S 指令在 SIMATIC S7 自动化系统中用于通过具有积分行为的执行器的二进制输出值输出信号来控制工艺过程。 在参数分配期间，可以通过激活或取消激活 PI 步进控制器的子功能来使控制器适应受控系统。 除了过程值分支中的功能以外，该指令还实现了一个完整的比例积分作用控制器，该控制器具有二进制输出值输出，并且还允许手动影响输出值。 步进控制器在没有位置反馈信号的情况下运行。

应用

可以将该控制器作为 PI 固定设定值控制器使用，或在级联、混合或比率控制器的辅助控制回路中使用，但不能将其作为主控制器使用。 控制器的功能基于采样控制器的 PI 控制算法，其附加功能还可从模拟量执行信号生成二进制输出信号。

调用

CONT_S 指令具有一个初始化例程，在设置输入参数 COM_RST = TRUE 时将运行该例程。 所有信号输出都被设置为零。完成初始化例程后，必须设置 COM_RST = FALSE。

只有以固定时间间隔调用块时，在控制块中计算的值才是正确的。 因此，应在循环中断 OB（OB 30 到 OB 38）中调用控制块。 在 CYCLE 参数中输入采样时间。

如果将指令 CONT_S 作为多重背景数据块调用，将不会创建任何工艺对象。没有参数分配接口或调试接口可用。 必须直接在多重背景数据块中为 CONT_S 分配参数,并通过监视表格进行调试。

错误信息

错误消息字 RET_VAL 不由块进行评估。

8.4.2.2 CONT_S 工作模式

设定值分支

在输入 SP_INT 中输入浮点格式的设定值。

过程值分支

可以 I/O 或浮点格式输入过程值。 函数 CRP_IN 按照以下规则将 I/O 值 PV_PER 转换为浮点格式的值 -100 到 +100 %：

CRP_IN = PV_PER * 100 / 27648 的输出

函数 PV_NORM 按照以下规则对 CRP_IN 的输出进行标准化：

PV_NORM 的输出 = (CRP_IN 的输出) * PV_FAC + PV_OFF

PV_FAC 的默认值为 1，PV_OFF 的默认值为 0。

形成误差信号

设定值与过程值之间的差值是误差信号。 为了抑制由于调节变量量化（例如，由于控制阀操作值的精度有限)所引起的小幅恒定振荡，可将死区应用于误差信号 (DEADBAND)。 DEADB_W = 0 时，死区关闭。

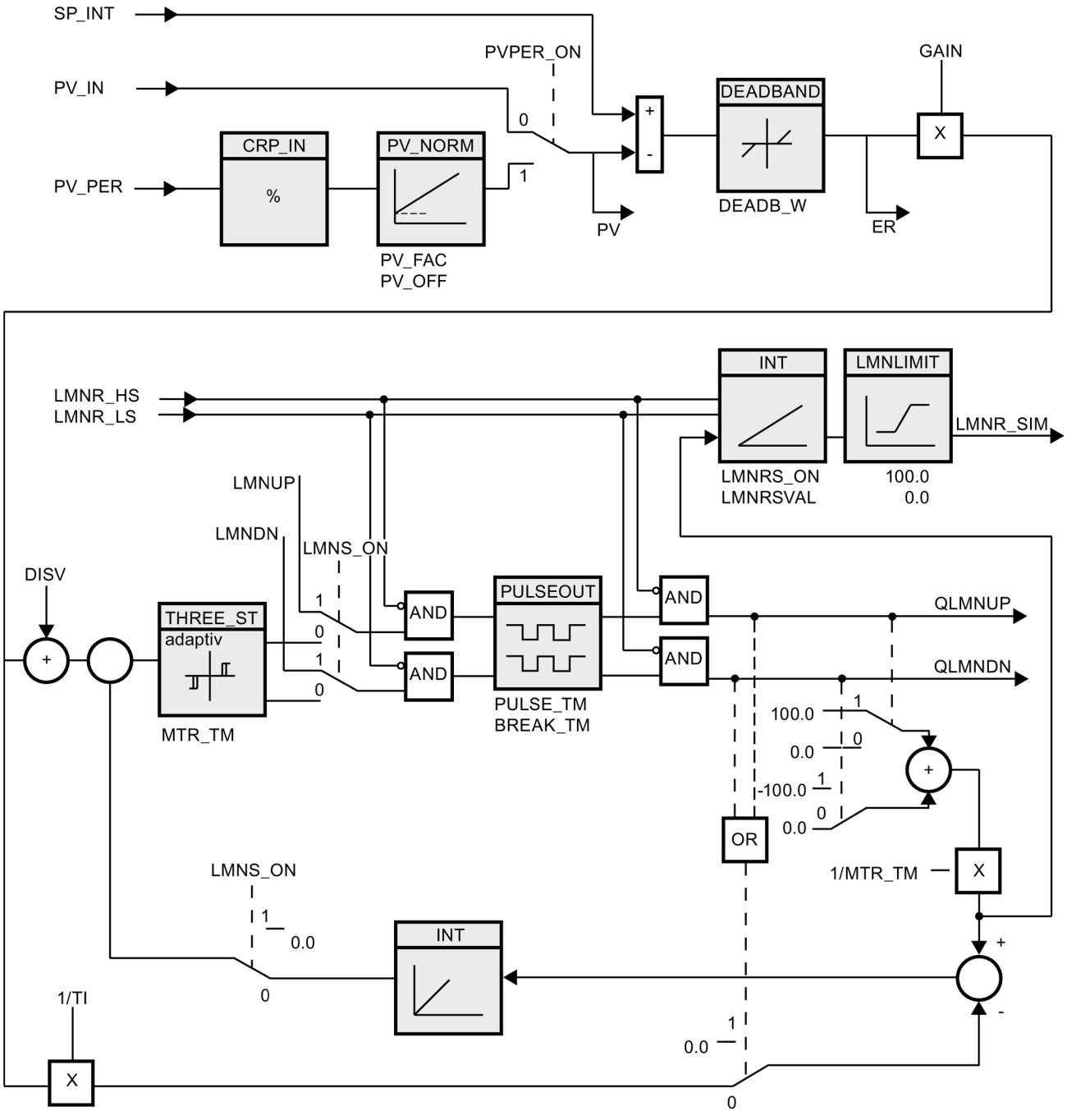
PI 步进算法

指令在没有位置反馈的情况下运行。 PI 算法的 I 作用和假定的位置反馈信号在一个 积分作用 (INT) 中计算，并作为反馈值与其余 P 作用进行比较。 差值将应用到三位元件 (THREE_ST) 以及为控制阀生成脉冲的脉冲整形器 (PULSEOUT)。 通过调整三位元件的响应阈值可以降低控制器的切换频率。

前馈控制

可在 DISV 输入中添加扰动变量。

8.4.2.3 CONT_S 方框图



8.4.2.4 CONT_S 输入参数

表格 8- 15

参数	数据类型	默认值	说明
COM_RST	BOOL	FALSE	该块具有一个初始化例程，在对输入“重启”进行置位时将处理该例程。
LMNR_HS	BOOL	FALSE	在输入“位置反馈的上端停止位信号”中互连信号“控制阀位于上端停止位”。 LMNR_HS=TRUE 表示： 控制阀位于上端停止位。
LMNR_LS	BOOL	FALSE	在输入“位置反馈的下端停止位信号”中互连信号“控制阀位于下端停止位”。 LMNR_LS=TRUE 表示控制阀位于下端停止位。
LMNS_ON	BOOL	FALSE	在“启用调节信号的手动模式”处将调节值信号处理模式切换为手动模式。
LMNUP	BOOL	FALSE	在输入“调节值信号上升”中，在调节值信号的手动模式下操作输出信号 QLMNUP。
LMNDN	BOOL	FALSE	在输入“调节值信号下降”中，在调节值信号的手动模式下操作输出信号 QLMNDN。
PVPER_ON	BOOL	FALSE	如果要从 I/O 读取过程值，输入 PV_PER 必须与 I/O 相关联，且输入“启用过程值 I/O”必须置位。
CYCLE	TIME	T#1s	块调用之间的时间间隔必须恒定。 “采样时间”输入用于指定块调用之间的时间。 CYCLE >= 1ms
SP_INT	REAL	0.0	“内部设定值”输入用于指定设定值。 允许值从 -100 到 100 %，或者是物理变量 1)。
PV_IN	REAL	0.0	在“过程值输入”处，可以将参数分配给调试值，或者互连浮点格式的外部过程值。 允许值从 -100 到 100 %，或者是物理变量 1)。
PV_PER	WORD	W#16#0000	I/O 格式的过程值在输入“过程值 I/O”中与控制器互连。
GAIN	REAL	2.0	“比例增益”输入用于指定控制器放大率。
TI	TIME	T#20s	“积分时间”输入用于确定积分作用的时间响应。 TI >= CYCLE
DEADB_W	REAL	1.0	将死区应用到系统偏差。 “死区宽度”输入用于确定死区的大小。 允许值从 0 到 100 %，或者是物理变量 1)。

参数	数据类型	默认值	说明
PV_FAC	REAL	1.0	“过程值因子”输入与过程值相乘。 该输入用于标定过程值的范围。
PV_OFF	REAL	0.0	“过程值偏移量”输入与过程值相加。 该输入用于标定过程值的范围。
PULSE_TM	TIME	T#3s	可在参数“最短脉冲周期”中分配最小脉冲时间。 PULSE_TM >= CYCLE
BREAK_TM	TIME	T#3s	可在参数“最小中断时间”中分配最小中断时间。 BREAK_TM >= CYCLE
MTR_TM	TIME	T#30s	执行器从一个限定停止位置移动到另一个限定停止位置所需要的时间在“电机运行时间”参数中输入。 MTR_TM >= CYCLE
DISV	REAL	0.0	对于前馈控制，扰动变量与输入“扰动变量”互连。 允许值从 -100 到 100 %，或者是物理变量 2)。

1) 设定值和过程值分支中的参数具有相同的单位

2) 调节值分支中的参数具有相同的单位

8.4.2.5 CONT_S 输出参数

表格 8- 16

参数	数据类型	默认值	说明
QLMNUP	BOOL	FALSE	如果置位输出“调节值信号上升”，则应打开控制阀。
QLMNDN	BOOL	FALSE	如果置位输出“调节值信号下降”，则应关闭控制阀。
PV	REAL	0.0	有效的过程值在“过程值”输出中输出。
ER	REAL	0.0	在“误差信号”输出中输出有效系统偏差。

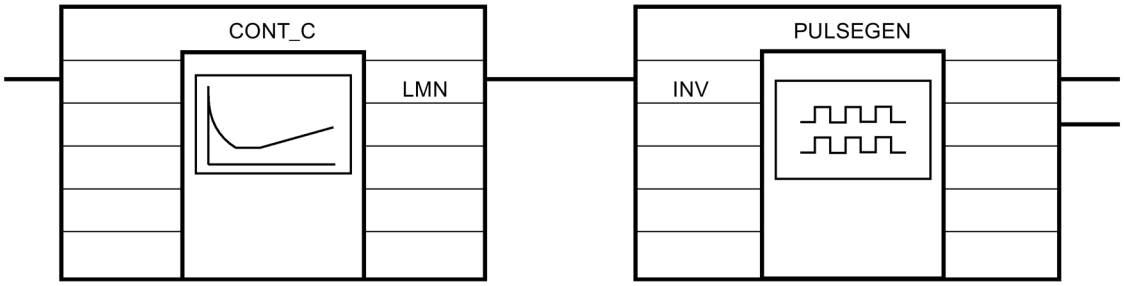
8.4.3 PULSEGEN

8.4.3.1 PULSEGEN 说明

指令 PULSEGEN 用于构造具有比例执行器脉冲输出的 PID 控制器。PULSEGEN 通过脉宽调制将输入值 INV（= PID 控制器的 LMN）转换成具有恒定周期持续时间的脉冲序列，该周期持续时间对应于更新输入值时所用的循环时间。

应用

可以用 PULSEGEN 指令来组态具有脉宽调制的两步或三步 PID 控制器。 该函数通常与连续控制器 CONT_C 一起使用。



调用

PULSEGEN 指令具有一个初始化例程，在设置输入参数 COM_RST = TRUE 时将运行该例程。 所有信号输出都被设置为零。完成初始化例程后，必须设置 COM_RST = FALSE。

只有以固定时间间隔调用块时，在控制块中计算的值才是正确的。 因此，应在循环中断 OB（OB 30 到 OB 38）中调用控制块。 在 CYCLE 参数中输入采样时间。

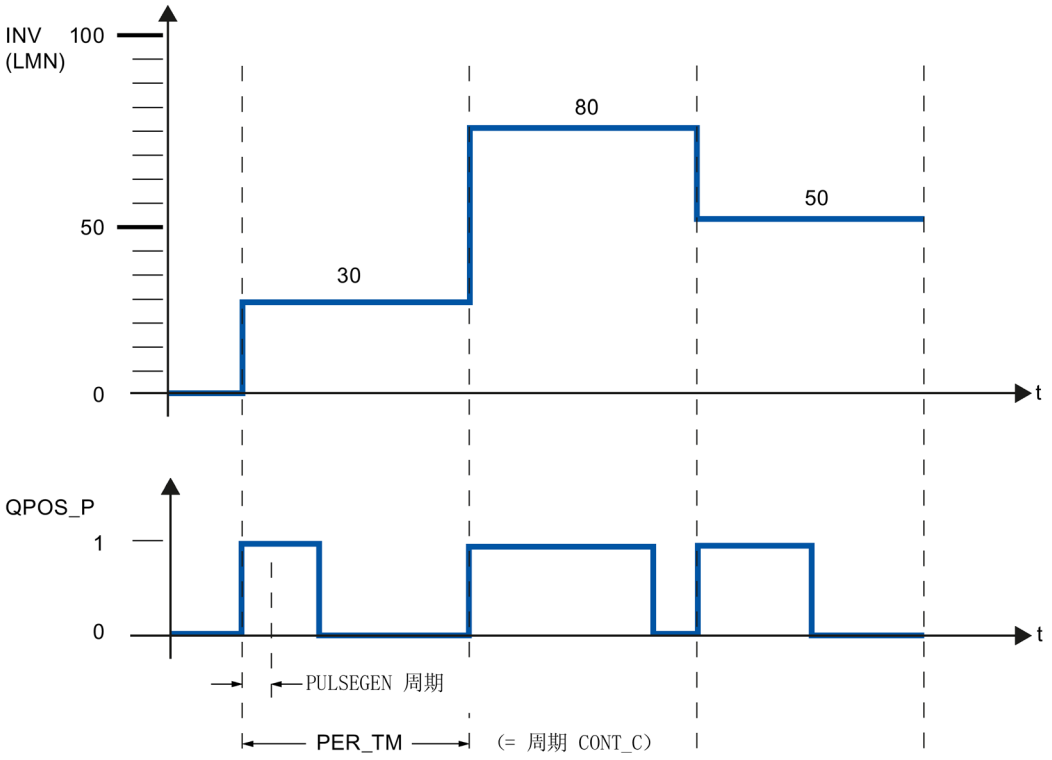
出现错误时的响应

错误消息字 RET_VAL 不由块进行评估。

8.4.3.2 PULSEGEN 的工作模式

脉宽调制

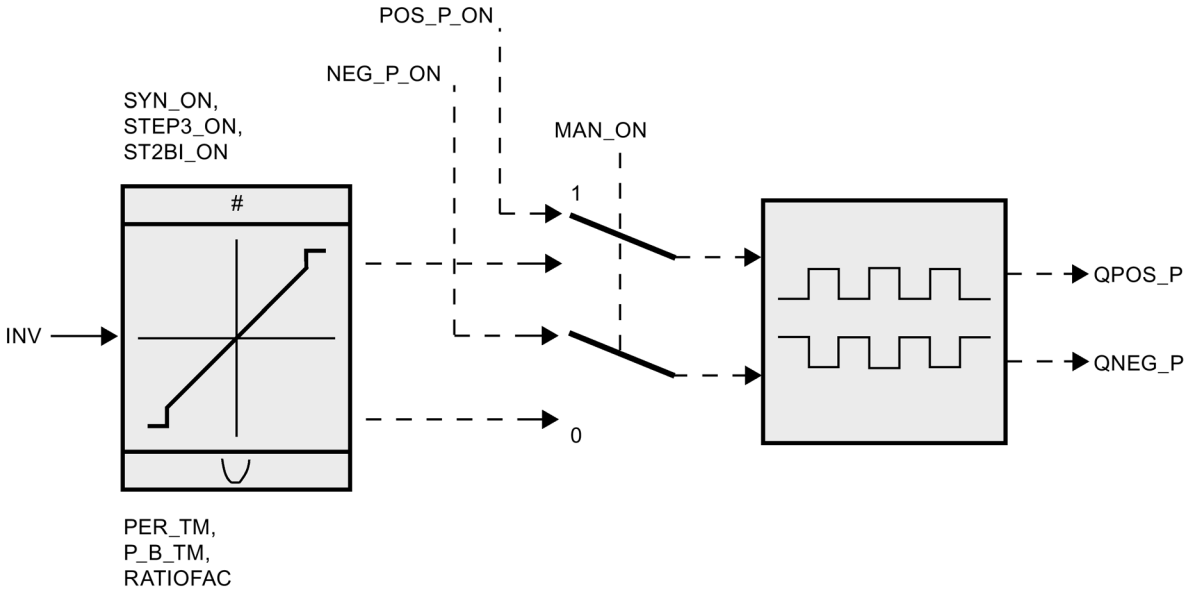
在每个周期持续时间内，脉冲的持续时间和输入变量成比例。 通过 PER_TM 分配的周期与 PULSEGEN 指令的处理周期不同。 相反，PER_TM 周期由 PULSEGEN 指令的多个处理周期组成，因此每个 PER_TM 周期中 PULSEGEN 调用的次数决定了脉冲宽度的精度。



每个 PER_TM 中 30 % 的输入变量和 10 次 PULSEGEN 调用表示以下结果：

- 前三次 PULSEGEN 调用时 QPOS_P 输出为“1”（10 次调用的 30%）
- 后七次 PULSEGEN 调用时 QPOS_P 输出为“0”（10 次调用的 70%）

方框图



调节值的精度

“采样比率”为 1:10（CONT_C 调用与 PULSEGEN 调用之比）时，此示例中的调节值精度将限制为 10%，换言之，只能在输出 QPOS_P 以 10% 为步长的脉冲持续时间对设置的输入值 INV 进行模拟。

精度将随每次 CONT_C 调用中 PULSEGEN 调用的次数的增加而提高。

例如，如果调用 PULSEGEN 的频率是调用 CONT_C 频率的 100 倍，则获得的操作值范围的精度为 1%。

说明

调用频率的减速比必须由用户编程设定。

自动同步

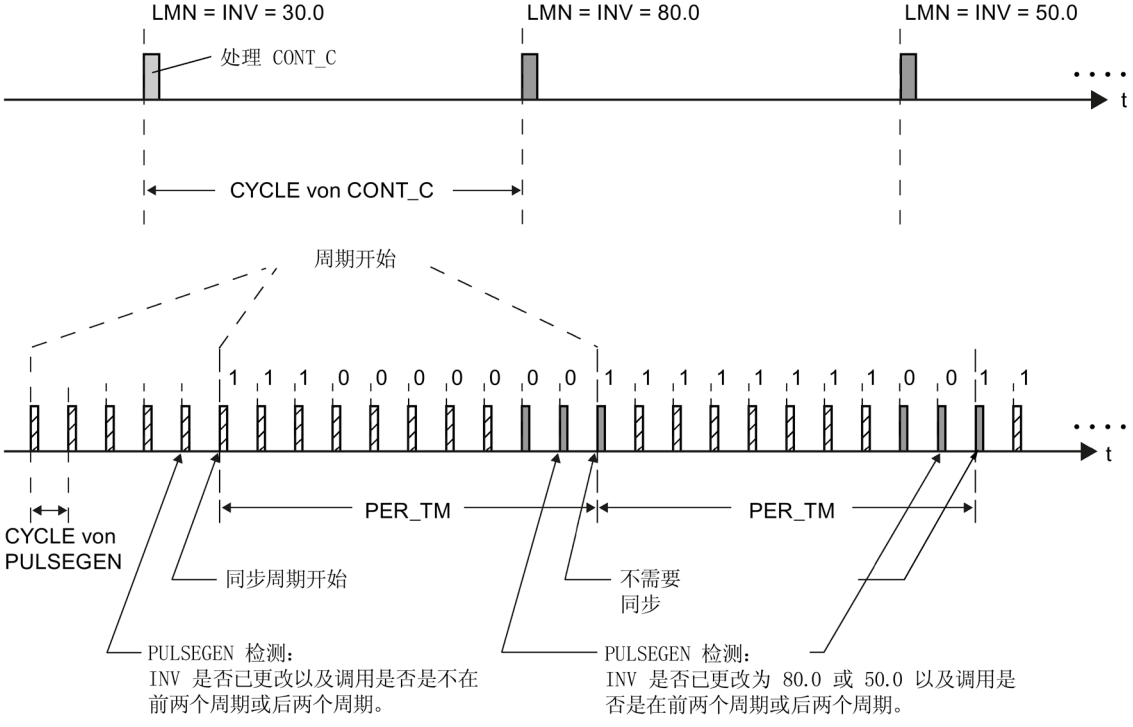
可以使脉冲输出与更新输入变量 INV 的指令（例如 CONT_C）自动同步。 这样可以确保尽快将输入变量的变化输出为脉冲。

脉冲整形器以对应周期持续时间 PER_TM 的时间间隔评估输入值 INV，并将该值转换成相应长度的脉冲信号。

但是，由于通常以较慢的循环中断等级计算 INV，因此在 INV 更新之后，脉冲整形器应尽快开始将离散值转换为脉冲信号。

为此，块可以使用以下步骤来与周期的起始点同步：

如果 INV 发生变化，且块调用不在周期的第一个或最后两个调用循环中，则执行同步。脉冲持续时间将重新计算，并在下一个循环与新周期一起输出。



如果 SYN_ON = FALSE，自动同步将关闭。

说明

如果旧的 INV 值（即 LMN 的值）映射到脉冲信号，则开始新周期和后续同步通常会导致某种不精确的情况产生。

8.4.3.3 PULSEGEN 的工作模式

模式

根据分配给脉冲整形器的参数，可以组态带有三位输出或者带有双极性或单极性两位输出的 PID 控制器。 下表给出了可能的模式所对应的开关组合的设置。

模式	MAN_ON	STEP3_ON	ST2BI_ON
三位控制	FALSE	TRUE	任意
具有双极的两步控制 调节范围（-100 % 到 100 %）	FALSE	FALSE	TRUE
带单极性的两位控制 调节范围（0 % 到 100 %）	FALSE	FALSE	FALSE
手动模式	TRUE	任意	任意

两步/三步控制的手动模式

在手动模式 (MAN_ON = TRUE) 下，无论 INV 为何值，均可使用信号 POS_P_ON 和 NEG_P_ON 设置三步或两步控制器的二进制输出。

控制	POS_P_ON	NEG_P_ON	QPOS_P	QNEG_P
三位控制	FALSE	FALSE	FALSE	FALSE
	TRUE	FALSE	TRUE	FALSE
	FALSE	TRUE	FALSE	TRUE
	TRUE	TRUE	FALSE	FALSE
两位控制	FALSE	任意	FALSE	TRUE
	TRUE	任意	TRUE	FALSE

8.4.3.4 三位控制

三位控制

在“三步控制”模式下，可以生成执行信号的三种状态。 为此，将二进制输出信号 QPOS_P 和 QNEG_P 的状态值分配给执行器的相应工作状态。 下表给出了温度控制的示例：

输出信号	加热	灭	冷却
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

通过特性曲线按输入变量计算脉冲持续时间。 特性曲线的形状由最小脉冲持续时间或最小间隔及比率因子定义。 比率因子的标准值为 1。

曲线中的“转折”由最小脉冲持续时间或最小间隔引起。

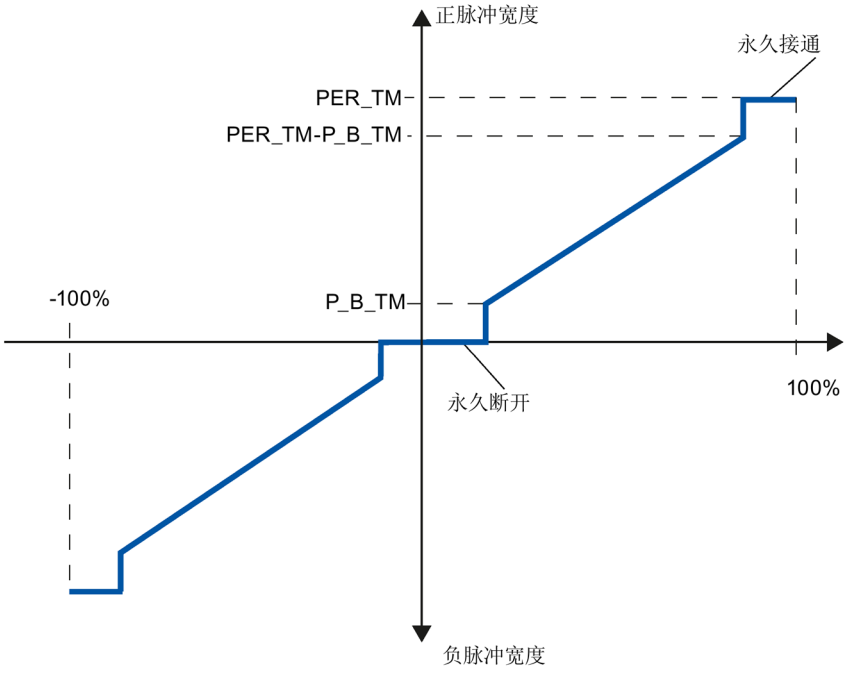
最小脉冲持续时间或最小间隔

正确分配的最小脉冲持续时间或最小间隔 P_B_TM 可以防止短暂开/关次数，避免由此而缩短开关元件和执行器的使用寿命。 如果由输入变量 LMN 的较小绝对值产生的脉冲持续时间小于 P_B_TM，则这些绝对值将被抑制。如果较大输入值生成的脉冲持续时间大于 PER_TM - P_B_TM，这些输入值将被设置为 100% 或 -100%。

用输入变量（以 % 表示）乘以周期持续时间来计算正或负脉冲的持续时间：

脉冲持续时间 = INV / 100 * PER_TM

下图显示了三步控制器的对称特性曲线（比率因子 = 1）。



非对称三步控制

使用比率因子 RATIOFAC 可以更改正脉冲与负脉冲持续时间的比率。 例如，在热过程中，可为加热和冷却过程使用不同的系统时间常数。

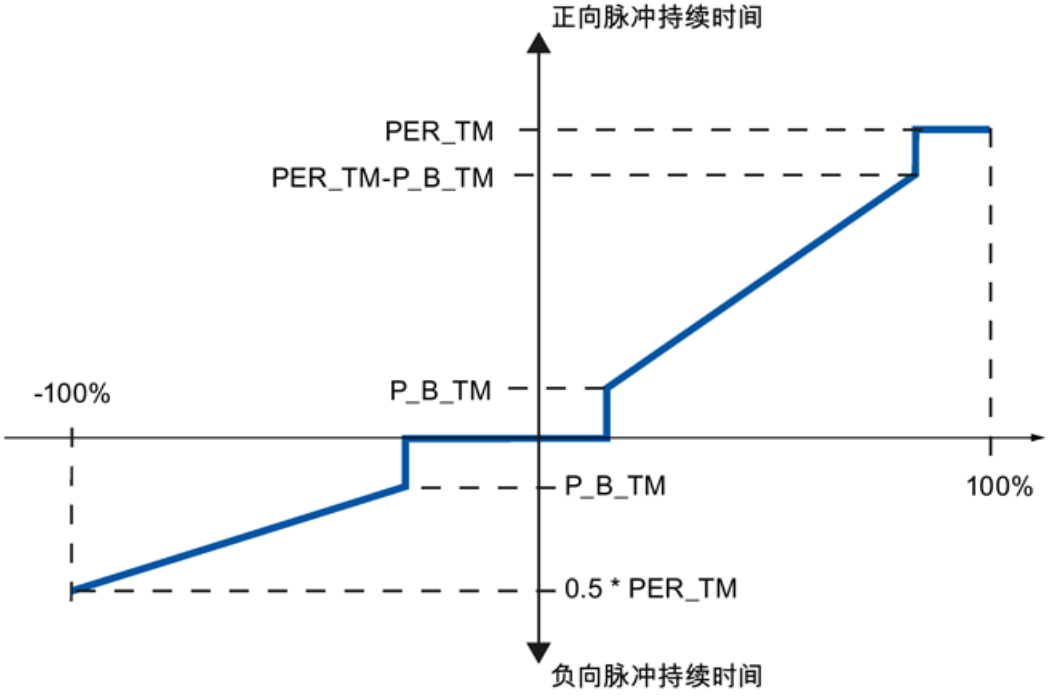
比率因子 < 1

将输入变量与周期持续时间相乘所得到的负向脉冲输出的脉冲持续时间与比率因子相乘。

正向脉冲持续时间 = INV /100 * PER_TM

负向脉冲持续时间 = INV / 100 * PER_TM * RATIOFAC

下图显示了三步控制器的非对称特性曲线（比率因子 = 0.5）：



比率因子 > 1

将输入变量与周期持续时间相乘所得到的正向脉冲输出的脉冲持续时间除以比率因子。

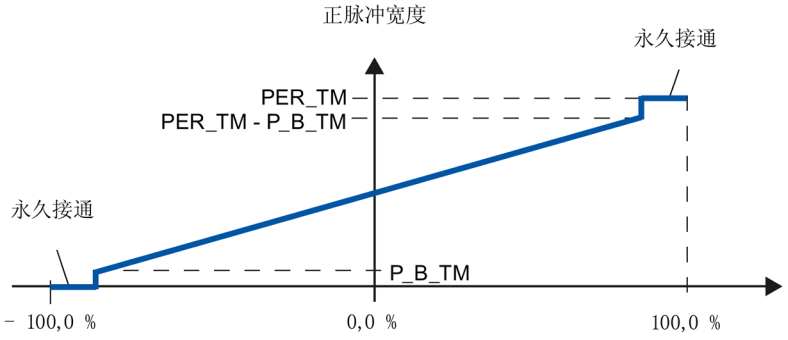
正向脉冲持续时间 = INV / 100 * PER_TM / RATIOFAC

负向脉冲持续时间 = INV / 100 * PER_TM

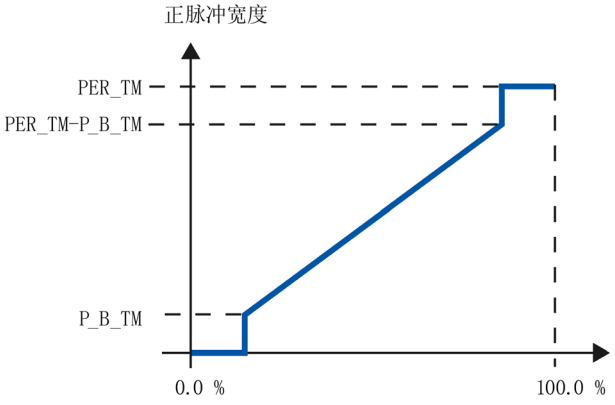
8.4.3.5 两位控制

在两步控制中，只会将 PULSEGEN 的正向脉冲输出 QPOS_P 连接到开/关执行器。 根据所使用的调节值范围，两步控制器具有双极或单极调节值范围。

具有双极调节变量范围的两步控制（-100% 到 100%）



具有单极调节变量范围的两步控制（0% 到 100%）



如果控制回路中的两步控制器的连接需要执行脉冲逻辑取反的二进制信号，则可在 QNEG_P 获得取反的输出信号。

脉冲	执行器开启	执行器关闭
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

8.4.3.6 PULSEGEN 输入参数

输入参数的值在块中不受限制。 没有参数检查。

表格 8- 17

参数	数据类型	默认值	说明
INV	REAL	0.0	在输入参数“输入变量”中连接模拟调节变量。 允许介于 -100 到 100 % 之间的值。
PER_TM	TIME	T#1s	在参数“周期持续时间”中输入脉宽调制的恒定周期持续时间。 该时间对应于控制器的采样时间。 脉冲整形器采样时间与控制器采样时间的比率决定脉宽调制的精度。 PER_TM >=20*CYCLE
P_B_TM	TIME	T#50 ms	可在参数“最小脉冲/中断时间”中分配最小脉冲/中断时间。 P_B_TM >= CYCLE
RATIOFAC	REAL	1.0	使用“比率因子”输入参数可以更改正向脉冲持续时间与负向脉冲持续时间的比率。 例如，在热处理中，可以为加热和冷却补偿不同的时间常数（例如，在使用电加热和水冷却的工艺中）。 允许介于 0.1 到 10.0 之间的值。
STEP3_ON	BOOL	TRUE	在输入参数“启用三步控制”中激活适当的模式。 在三步控制中，两个输出信号都处于激活状态。
ST2BI_ON	BOOL	FALSE	在输入参数“启用双极性调节值范围的两位控制”中，可以在“双极性调节值范围的两位控制”和“单极性调节值范围的两位控制”模式之间选择。STEP3_ON = FALSE 是必需的。
MAN_ON	BOOL	FALSE	通过设置输入参数“启用手动模式”可手动设置输出信号。
POS_P_ON	BOOL	FALSE	对于处于手动模式下的三步控制，可在输入参数“正向脉冲开启”中操作输出信号 QPOS_P。 在两步控制的手动模式下，QNEG_P 始终设置为与 QPOS_P 反向。
NEG_P_ON	BOOL	FALSE	对于处于手动模式下的三步控制，可在输入参数“负向脉冲开启”中操作输出信号 QNEG_P。 在两步控制的手动模式下，QNEG_P 始终设置为与 QPOS_P 反向。
SYN_ON	BOOL	TRUE	通过设置输入参数“启用同步”，可以使脉冲输出自动与更新输入变量 INV 的块同步。 这样可以确保尽快将输入变量的变化输出为脉冲。
COM_RST	BOOL	FALSE	该块具有一个初始化例程，在对输入“重启”进行置位时将处理该例程。
CYCLE	TIME	T#10ms	块调用之间的时间间隔必须恒定。 “采样时间”输入用于指定块调用之间的时间。 CYCLE >= 1ms

8.4.3.7 PULSEGEN 输出参数

表格 8- 18

参数	数据类型	默认值	说明
QPOS_P	BOOL	FALSE	如果要输出脉冲，输出参数“输出信号正向脉冲”将被置位。 在三步控制中，此项始终为正向脉冲。 在两步控制中，QNEG_P 始终设置为与 QPOS_P 反向。
QNEG_P	BOOL	FALSE	如果要输出脉冲，输出参数“输出信号负向脉冲”将被置位。 在三步控制中，此项始终是负向脉冲。 在两步控制中，QNEG_P 始终设置为与 QPOS_P 反向。

8.4.4 TCONT_CP

8.4.4.1 TCONT_CP 说明

指令 TCONT_CP 用于控制具有连续或脉冲控制信号的温度处理过程。 控制器功能基于 PID 控制算法及其它适用于温度过程的功能。 为改进对温度过程的控制响应，该块包括了一个控制区，并在设定值阶跃变化时减少比例分量。

该指令可以通过控制器优化功能自行设置 PI/PID 参数。

应用

控制器控制一个执行器；换句话说，使用一个控制器可进行加热或冷却操作，但不能同时进行这两种操作。 如果将该块用于冷却，必须为 GAIN 分配一个负值。 控制器的这种反转意味着，例如温度上升时，调节变量 LMN 会增大，冷却操作也随之加强。

调用

必须等距调用指令 TCONT_CP。 要达到该目的，可以使用循环中断优先级等级（例如，S7-300 的 OB35）。

TCONT_CP 指令具有一个初始化例程，在设置输入参数 COM_RST = TRUE 时将运行该例程。 初始化过程中，积分作用被设置为初始化值 I_ITVAL。 所有信号输出都设置为零。 在执行完初始化例程后，块将 COM_RST 重新设置成 FALSE。 如果需要在 CPU 重启时执行初始化，则可在 OB 100 中调用此块 (COM_RST = TRUE)。

如果将指令 TCONT_CP 作为多重背景数据块调用，将不会创建任何工艺对象。没有参数分配接口或调试接口可用。 必须直接在多重背景数据块中为 TCONT_CP 分配参数,并通过监视表格进行调试。

参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

8.4.4.2 TCONT_CP 的工作模式

设定值分支

在输入 SP_INT 中输入浮点格式的设定值，作为物理值或者百分比值。 用于形成控制偏差的设定值和过程值必须采用相同的单位。

过程值选项 (PVPER_ON)

根据 PVPER_ON，可读取 I/O 格式或浮点数格式的过程值。

PVPER_ON	过程值输入
TRUE	通过输入 PV_PER 中的模拟量 I/O (PIW xxx) 读取过程值。
FALSE	在输入 PV_IN 处采集浮点格式的过程值。

过程值格式转换 CRP_IN (PER_MODE)

函数 CRP_IN 按照下列规则并根据 PER_MODE 开关设置，将 I/O 值 PV_PER 转换为浮点格式：

PER_MODE	CRP_IN 的输出	模拟量输入类型	单位
0	PV_PER * 0.1	热电偶； PT100/NI100；标准	°C；°F
1	PV_PER * 0.01	PT100/NI100；气候型；	°C；°F
2	PV_PER * 100/27648	电压/电流	%

过程值标定 PV_NORM（PF_FAC，PV_OFFS）

函数 PV_NORM 根据以下规则计算 CRP_IN 的输出：

“PV_NORM 的输出” = “CRP_IN 的输出” * PV_FAC + PV_OFFS

有以下用途：

- 以 PV_FAC 为过程值因子、PV_OFFS 为过程值偏移量进行过程值调整。
- 将温度值标定为百分比值

  如果要以百分比的形式输入设定值，现在必须将测得的温度值转换成百分比值。
- 将百分比值标定为温度值

  如果想要以物理温度单位输入设定值，现在必须将测得的电压/电流值转换成温度值。

参数计算：

- PV_FAC = PV_NORM 的范围/CRP_IN 的范围；
- PV_OFFS = LL (PV_NORM) - PV_FAC * LL(CRP_IN)；

  其中，LL：下限

标定通过默认值（PV_FAC = 1.0 和 PV_OFFS = 0.0）关闭。 在 PV 输出中输出有效过程值。

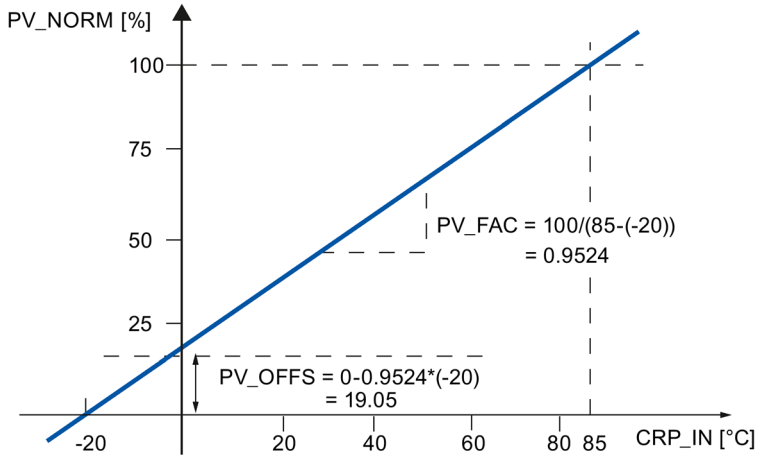
说明

对于脉冲控制，必须在快速脉冲调用中将过程值传送到块中（原因： 平均值过滤）。 否则，控制质量会变差。

过程值标定示例

如果要以百分比的形式输入设定值，并且 CRP_IN 的温度范围为 -20 到 85 °C，则必须将温度范围标准化为百分比值。

下图给出的示例说明了如何将 -20 到 85 °C 的温度范围修改为 0 到 100% 的内部标定：



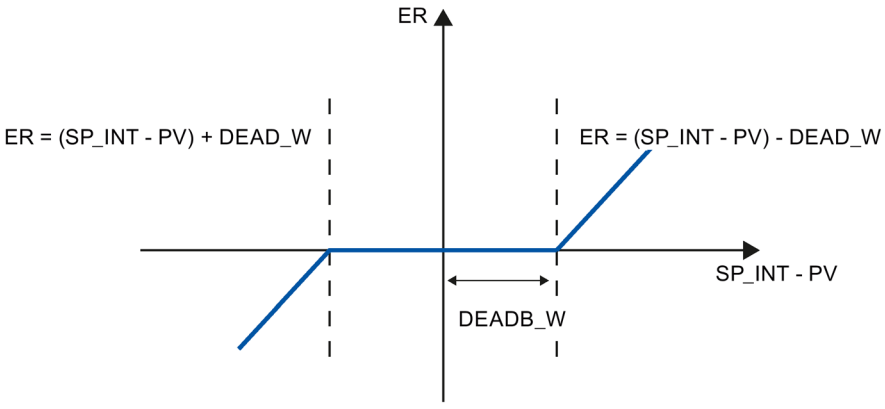
形成控制偏差

在到达死区之前，设定值与过程值的差值就是控制偏差。

设定值与过程值的单位必须相同。

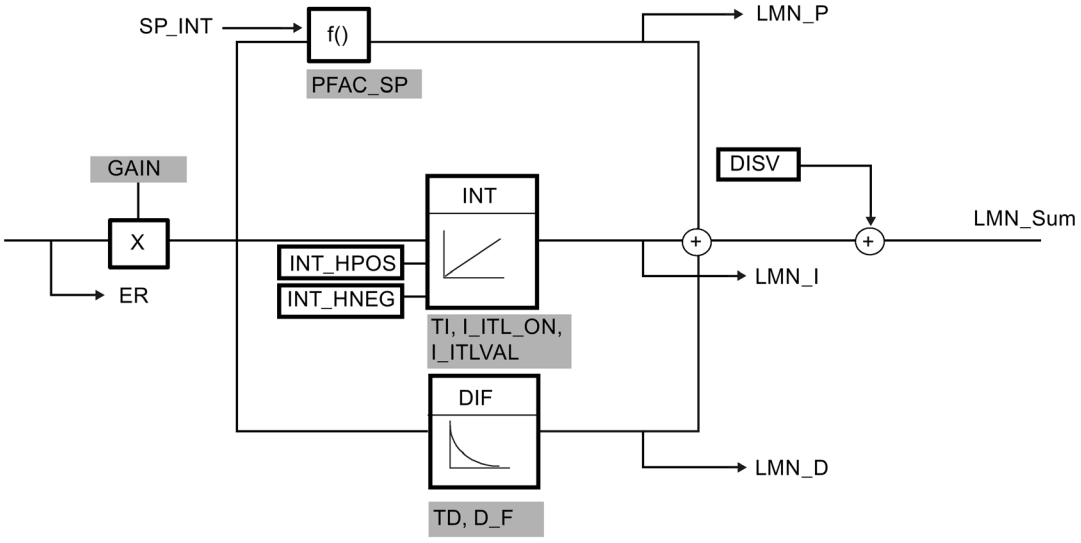
死区 (DEADB_W)

为了抑制由于调节变量量化所引起的小幅持续振荡（例如，在使用 PULSEGEN 进行脉宽调制时），可对控制偏差使用死区 (DEADBAND)。 DEADB_W = 0.0 时，死区禁用。 控制偏差的有效性由 ER 参数指示。



PID 算法

下图显示了 PID 算法的方框图。



PID 算法（GAIN、TI、TD、D_F）

PID 算法作为位置算法运行。 比例、积分 (INT) 和微分 (DIF) 作用是并行连接在一起的，可以单独激活或禁用。 这样便可组态 P、PI、PD 和 PID 控制器。

控制器调节功能支持 PI 控制器和 PID 控制器。 使用负 GAIN 实现控制器反转（冷却控制器）。

如果将 TI 和 TD 设置为 0.0，则将在工作点获得一个纯 P 控制器。

在时间范围内的阶跃响应是：

$$LMN\_Sum(t) = GAIN*ER(0)(1+ \frac{1}{TI} * t+D\_F * e^{\frac{-t}{TD/D\_F}}$$

其中：

LMN_Sum(t) 是控制器自动模式中的调节变量。

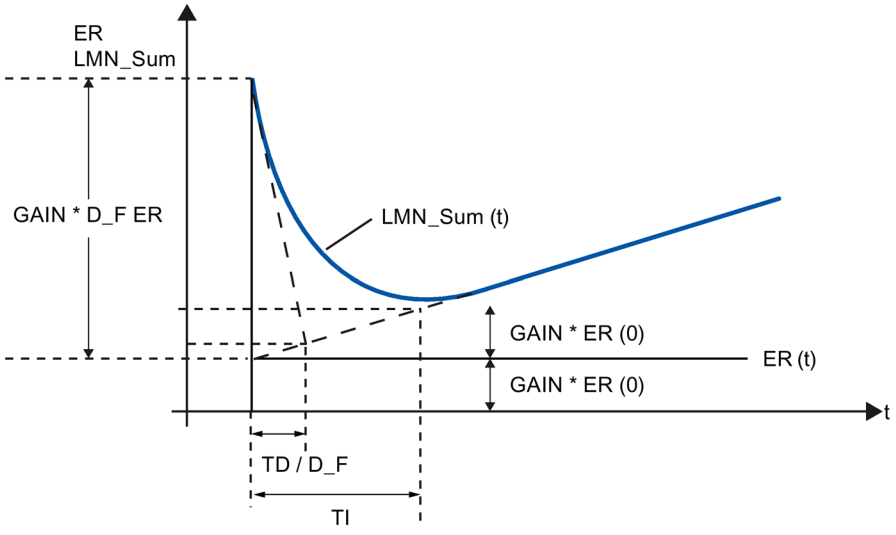
ER (0) 是标准化控制偏差的阶跃高度

GAIN 是控制器增益

TI 是积分时间

TD 是微分作用时间

D_F 是微分因子



积分作用（TI、I_ITL_ON、I_ITLVAL）

在手动模式下，使用以下公式进行修正： LMN_I = LMN - LMN_P - DISV.

如果输出值受限，则积分作用将停止。 如果控制偏差使积分作用移回到输出范围方向，则将再次启用积分作用。

也可通过以下方法来修改积分作用：

- 通过 TI = 0.0 禁用控制器的积分作用
- 当设定值发生变化时，弱化比例作用
- 控制区
- 在线修改输出值的限值

当设定值发生变化时，弱化比例作用 (PFAC_SP)

为了防止超调，可以使用参数“针对设定值更改的比例因子”(PFAC_SP) 来弱化比例作用。通过 PFAC_SP，可在 0.0 到 1.0 之间连续选择，以确定设定值发生变化时比例作用的效果：

- PFAC_SP = 1.0：如果设定值发生变化，则比例作用完全有效
- PFAC_SP = 0.0：如果设定值发生变化，则比例作用无效

也可通过补偿积分作用来弱化比例作用。

微分作用（TD、D_F）

- 通过 TD = 0.0 可禁用控制器的微分作用
- 如果微分作用处于激活状态，则下列关系成立：

  TD = 0.5 * CYCLE * D_F

带工作点的 P 或 PD 控制器的参数设置

在用户界面中，可禁用积分作用 (TI = 0.0)，也可禁用微分作用 (TD = 0.0)。 然后进行如下参数设置：

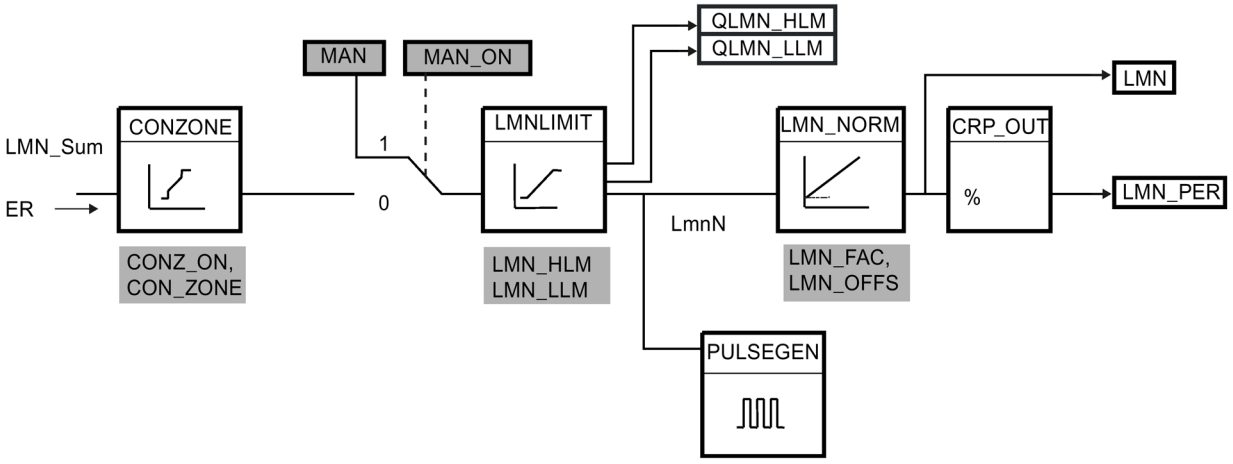
- I_ITL_ON = TRUE
- I_ITLVAL = 工作点；

前馈控制 (DISV)

可在 DISV 输入中添加扰动变量。

计算输出值

下图显示的是输出值计算过程的方框图：



参数组态接口
指令调用接口
参数组态接口、调用接口

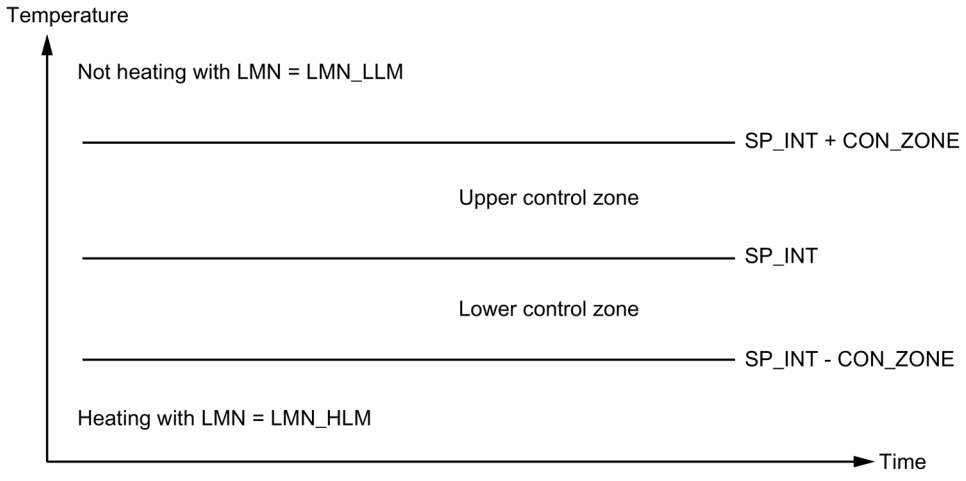
控制区（CONZ_ON、CON_ZONE）

如果 CONZ_ON = TRUE，则控制器在控制区范围内工作。 也就是说，控制器按照以下算法进行工作：

- 如果过程值 PV 超出设定值 SP_INT 的数值大于 CON_ZONE，则值 LMN_LLM 将作为调节变量输出。
- 如果过程值 PV 小于设定值 SP_INT 的数值大于 CON_ZONE，则输出为 LMN_HLM。
- 如果过程值 PV 位于控制区 (CON_ZONE) 范围内，则通过 PID 算法 LMN_Sum 获取输出值。

说明

将调节变量由 LMN_LLM 或 LMN_HLM 更改为 LMN_Sum 时以控制区的 20% 的滞后为前提。



说明

在手动启用控制区之前，请确保控制区范围不会过窄。 如果控制区范围过窄，则调节变量和过程值会产生振荡。

控制区的优点

当过程值进入控制区时，D 作用会导致调节变量数值急剧下降。 这意味着仅当激活 D 作用时，控制区才有用。 如果没有控制区，只有减小 P 作用才能从本质上减小调节变量。 如果最小或最大调节变量都远离新工作点所需的调节变量，则控制区会促使快速稳定，而不会产生过调或欠调。

手动值处理（MAN_ON、MAN）

可以在手动与自动模式之间切换。 在手动模式下，调节变量被修正为手动选择的值。

积分作用 (INT) 内部设置为 LMN - LMN_P - DISV，微分作用 (DIF) 内部设置为 0 并同步。 因此，可以平滑地切换到自动模式。

说明

MAN_ON 参数在调节期间无效。

输出值的限值 LMNLIMIT（LMN_HLM、LMN_LLM）

LMNLIMIT 函数用于将输出值限制为限值 LMN_HLM 和 LMN_LLM。 如果达到了这些限制值，则通过消息位 QLMN_HLM 和 QLMN_LLM 进行指示。

如果输出值受限，则积分作用将停止。 如果控制偏差使积分作用移回到输出范围方向，则将再次启用积分作用。

在线更改调节值限值

如果输出值的范围缩小，并且输出值的不受限新值超出了限值范围，则积分作用会发生改变，从而改变输出值。

输出值的减小幅度与输出值限值的变化幅度相同。 如果输出值在改变之前不受限制，其将被设置为新的限值（此处指输出值的上限）。

输出值的标定 LMN_NORM（LMN_FAC、LMN_OFFS）

函数 LMN_NORM 按照以下规则对输出值进行标准化：

LMN = LmnN * LMN_FAC + LMN_OFFS

有以下用途：

- 以 LMN_FAC 为输出值因子、以 LMN_OFFS 为输出值偏移量进行输出值标定。

输出值也可以使用 I/O 格式。 函数 CRP_OUT 按照以下规则将 LMN 浮点值转换为 I/O 值：

LMN_PER = LMN * 27648/100

标定通过默认值（LMN_FAC = 1.0 和 LMN_OFFS = 0.0）关闭。 有效的输出值将被发送至输出 LMN。

保存控制器参数 SAVE_PAR

如果将当前控制器参数分类为可以使用，则可以在手动更改之前将这些参数保存在指令 TCONT_CP 的背景数据块中专门为此提供的结构参数中。 优化控制器时，调节前有效的值将覆盖所保存的参数。

PFAC_SP、GAIN、TI、TD、D_F、CONZ_ON 和 CONZONE 被写入到结构 PAR_SAVE 中。

重新装载保存的控制器参数 UNDO_PAR

使用此功能可再次为控制器激活上次保存的控制器参数设置（仅在手动模式下）。

在 PI 和 PID 参数 LOAD_PID 之间切换 (PID_ON)

经过调节后，PI 参数和 PID 参数将存储在 PI_CON 结构和 PID_CON 结构中。 根据 PID_ON，可以在手动模式下使用 LOAD_PID 将 PI 或 PID 参数写入到有效的控制器参数中。

PID 参数 PID_ON = TRUE	PI 参数 PID_ON = FALSE
• GAIN = PID_CON.GAIN • TI = PID_CON.TI • TD = PID_CON.TD	• GAIN = PI_CON.GAIN • TI = PI_CON.TI

说明

仅当控制器增益不等于 0 时，才能通过 UNDO_PAR 或 LOAD_PID 将控制器参数写回到控制器中：

仅当相应的 GAIN <> 0 时，才能使用 LOAD_PID 复制参数（PI 或 PID 参数）。 这种策略考虑到了尚未进行任何调节或 PID 参数丢失的情况。 如果 PID_ON = TRUE 且 PID.GAIN = FALSE，则将 PID_ON 设置为 FALSE 并复制 PI 参数。

- 调节功能可对 D_F、PFAC_SP 进行预设。 然后用户可修改这些参数。LOAD_PID 不会更改这些参数。
- 使用 LOAD_PID 时，始终重新计算控制区 (CON_ZONE = 250/GAIN)，即使 CONZ_ON = FALSE。

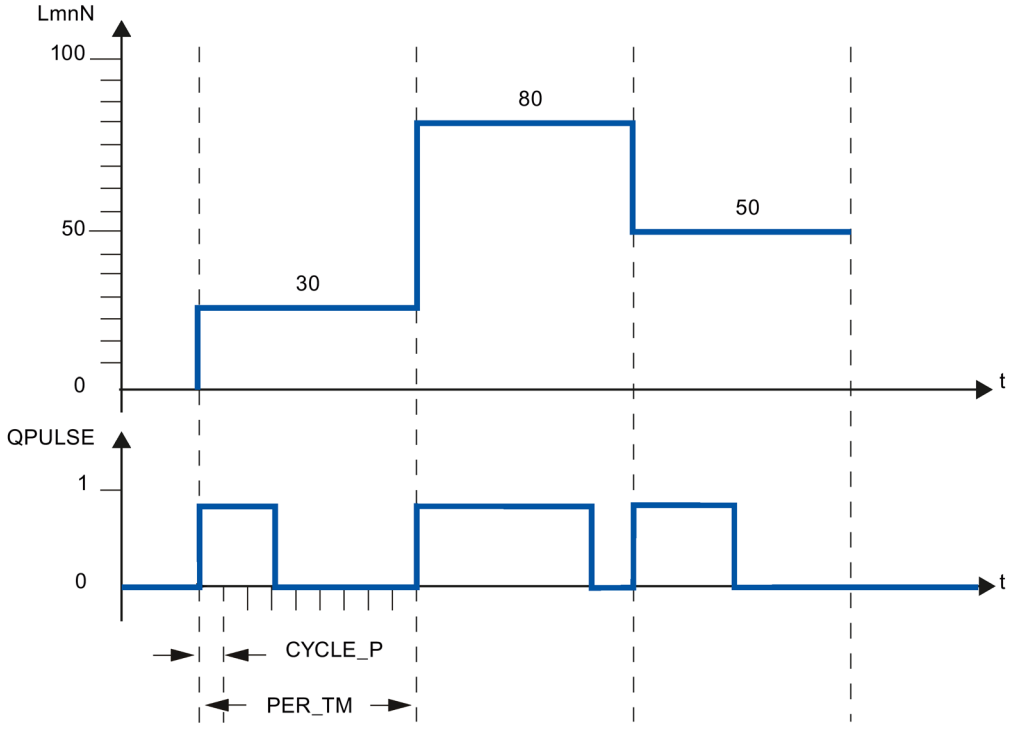
参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

8.4.4.3 脉冲发生器的工作原理

PULSEGEN 功能通过脉宽模块将模拟调节值 LmnN 转换为周期持续时间为 PER_TM 的脉冲序列。PULSEGEN 通过 PULSE_ON = TRUE 打开并按照周期 CYCLE_P 进行处理。



因此，调节值 LmnN = 30% 及每个 PER_TM 周期 10 次 PULSEGEN 调用意味着：

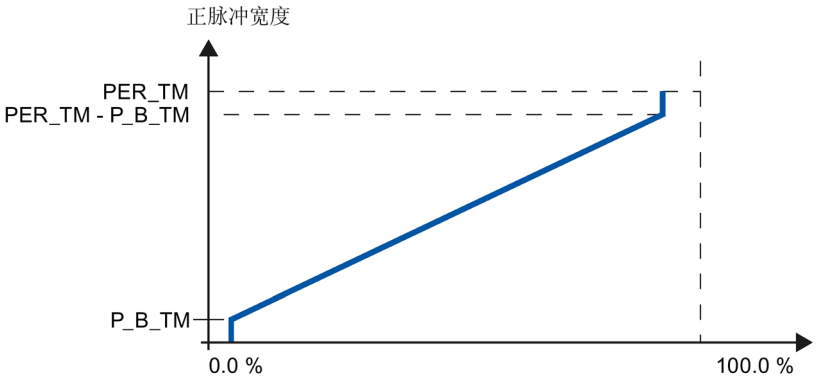
- 前三次 PULSEGEN 调用时输出 QPULSE 为 TRUE
  （10 次调用的 30%）
- 后七次 PULSEGEN 调用时输出 QPULSE 为 FALSE
  （10 次调用的 70%）

每个脉冲重复周期的脉冲持续时间与受控变量成比例，计算方式如下：

脉冲持续时间 = PER_TM * LmnN /100

通过抑制最小脉冲时间或中断时间，转换的特征曲线在开始和结束区域产生“拐点”。

下图展示了带有单极性受控变量范围（0% 至 100%）的两位控制：



最小脉冲时间或最小中断时间 (P_B_TM)

短时开启或关闭操作会影响执行器以及精密控制设备的使用寿命。 这可通过设置最小脉冲时间或最小中断时间 P_B_TM 来避免。

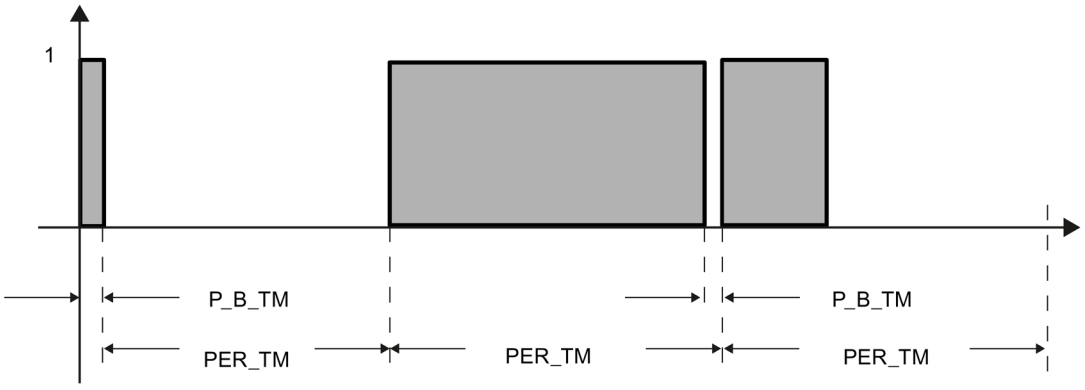
如果由输入变量 LmnN 中的较小绝对值产生的脉冲持续时间小于 P_B_TM，则这些较小绝对值产生的脉冲将被抑制。

如果较大输入值产生的脉冲持续时间大于 PER_TM - P_B_TM，则这些较大输入值产生的脉冲将被设置为 100%。 这将减少脉冲生成的动态性。

建议将值设置成 P_B_TM ≤ 0,1 * PER_TM，以获得最小脉冲时间和最小中断时间。

上图曲线中的“拐点”是由最小脉冲时间或最小中断时间引起的。

以下示意图说明了脉冲输出的开关响应：



脉冲生成的精度

脉冲发生器的采样时间 CYCLE_P 与周期持续时间 PER_TM 相比越小，脉宽调制的精确就越高。 要实现足够精确的控制，应该应用以下关系：

CYCLE_P ≤ PER_TM/50

调节值以 ≤ 2 % 的分辨率转换为脉冲。

说明

在脉冲整形器周期内调用控制器时，必须注意以下事项：

在脉冲整形器周期内调用控制器将导致对过程值取平均值。 因此，在输出 PV 处，输入 PV_IN 和 PV_PER 的值可能不同。 如果要跟踪设定值，必须在调用整个控制器处理 (QC_ACT = TRUE) 时保存输入参数 PV_IN 的过程值。 如果在这些调用时间之间调用脉冲整形器，则必须给输入参数 PV_IN 和 SP_INT 提供已保存的过程值。

参见

TCONT_CP 说明 (页 532)

TCONT_CP 的工作模式 (页 533)

TCONT_CP 方框图 (页 546)

TCONT_CP 输入参数 (页 548)

TCONT_CP 输出参数 (页 549)

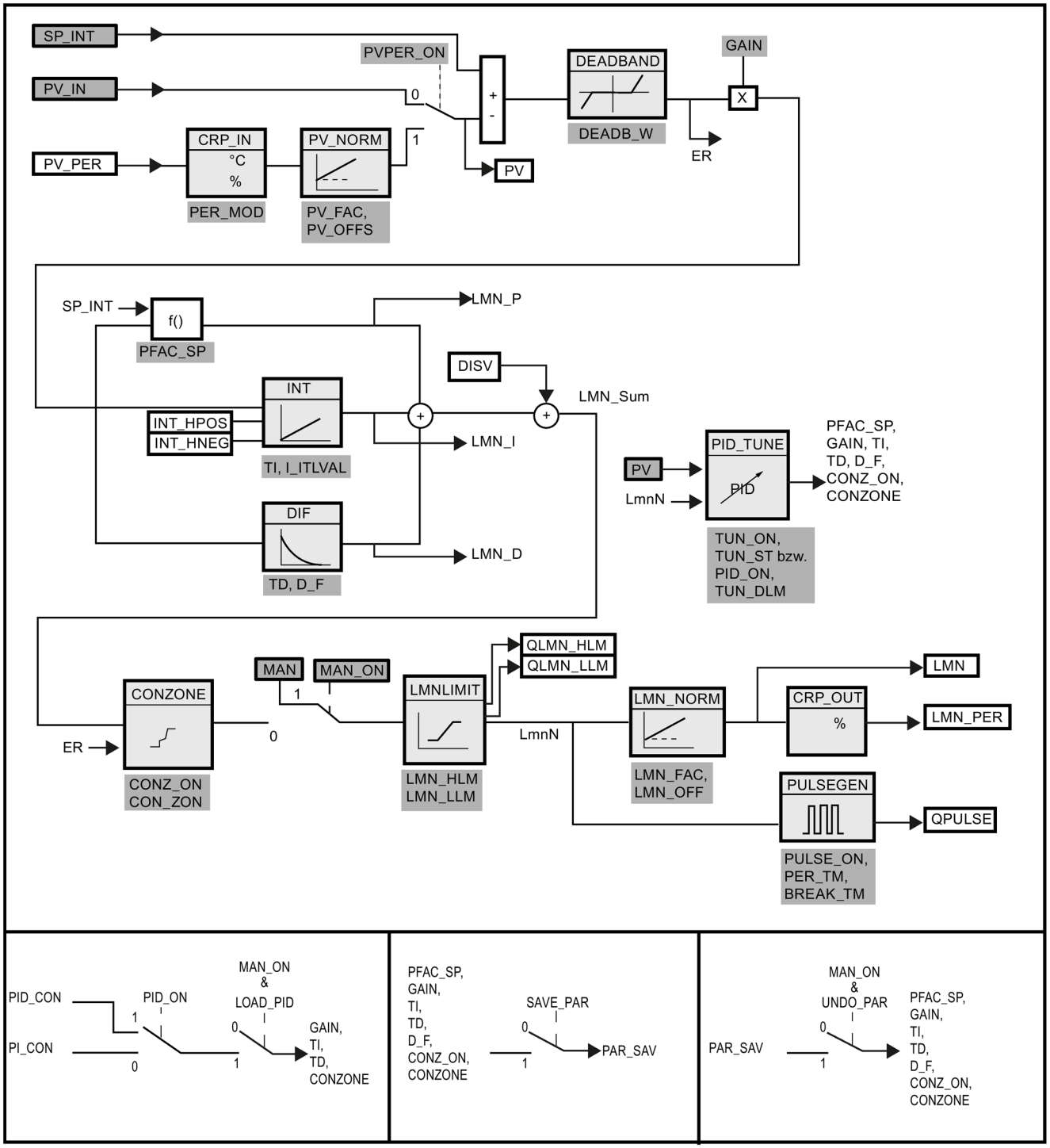
TCONT_CP 输入/输出参数 (页 550)

静态变量 TCONT_CP (页 551)

参数 STATUS_H (页 557)

参数 STATUS_D (页 558)

8.4.4.4 TCONT_CP 方框图



参见

TCONT_CP 说明 (页 532)

TCONT_CP 的工作模式 (页 533)

脉冲发生器的工作原理 (页 543)

TCONT_CP 输入参数 (页 548)

TCONT_CP 输出参数 (页 549)

TCONT_CP 输入/输出参数 (页 550)

静态变量 TCONT_CP (页 551)

参数 STATUS_H (页 557)

参数 STATUS_D (页 558)

8.4.4.5 TCONT_CP 输入参数

表格 8- 19

参数	地址	数据类型	默认值	说明
PV_IN	0.0	REAL	0.0	在“过程值输入”处，可以将参数分配给调试值，或者互连浮点格式的外部过程值。 有效值取决于所用的传感器。
PV_PER	4.0	INT	0	I/O 格式的过程值在输入“过程值 I/O”中与控制器互连。
DISV	6.0	REAL	0.0	对于前馈控制，扰动变量与输入“扰动变量”互连。
INT_HPOS	10.0	BOOL	FALSE	积分作用的输出可在正向保持。 为此，必须将 INT_HPOS 设置为 TRUE。 在级联控制中，主控制器的 INT_HPOS 连接到次级控制器的 QLMN_HLM。
INT_HNEG	10.1	BOOL	FALSE	可以在负方向上保持积分作用的输出。 为此，必须将 INT_HNEG 设置为 TRUE。 在级联控制中，主控制器的 INT_HNEG 连接到次级控制器的 QLMN_LLM。
SELECT	12.0	INT	0	如果脉冲整形器开启，则有几种方法可以调用 PID 算法和脉冲整形器： • SELECT = 0：以快速循环中断优先级等级调用控制器，处理 PID 算法和脉冲整形器。 • SELECT = 1：在 OB1 中调用控制器，仅处理 PID 算法。 • SELECT = 2：以快速循环中断优先级等级调用控制器，仅处理脉冲整形器。 • SELECT = 3：以慢速循环中断优先级等级调用控制器，仅处理 PID 算法。

参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

8.4.4.6 TCONT_CP 输出参数

表格 8- 20

参数	地址	数据类型	默认值	说明
PV	14.0	REAL	0.0	有效的过程值在“过程值”输出中输出。 有效值取决于所用的传感器。
LMN	18.0	REAL	0.0	有效“调节值”以浮点格式在“调节值”输出中输出。
LMN_PER	22.0	INT	0	I/O 格式的调节值在“调节值 I/O”输出中与控制器互连。
QPULSE	24.0	BOOL	FALSE	在输出 QPULSE 对调节值进行脉宽调制。
QLMN_HLM	24.1	BOOL	FALSE	调节值始终限制在上限和下限之间。 输出 QLMN_HLM 说明已达到上限。
QLMN_LLM	24.2	BOOL	FALSE	调节值始终限制在上限和下限之间。 输出 QLMN_LLM 说明已达到下限。
QC_ACT	24.3	BOOL	TRUE	此参数指示是否在下一次调用块时处理连续控制组件（仅当 SELECT 的值为 0 或为 1 时才相关）。

参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

参数 STATUS_H (页 557)

参数 STATUS_D (页 558)

8.4.4.7 TCONT_CP 输入/输出参数

表格 8- 21

参数	地址	数据类型	默认值	说明
CYCLE	26.0	REAL	0.1 s	设置 PID 算法的采样时间。 在阶段 1，整定器计算采样时间并将该采样时间输入到 CYCLE 中。 CYCLE > 0.001 s
CYCLE_P	30.0	REAL	0.02 s	在此输入中，设置脉冲整形器作用的采样时间。 在阶段 1，TCONT_CP 指令计算采样时间，并将该采样时间输入到 CYCLE_P 中。 CYCLE_P > 0.001 s
SP_INT	34.0	REAL	0.0	“内部设定值”输入用于指定设定值。 有效值取决于所用的传感器。
MAN	38.0	REAL	0.0	“手动值”输入用于设置手动值。 在自动模式下，它可跟踪调节值。
COM_RST	42.0	BOOL	FALSE	该块具有一个初始化例程，在置位输入 COM_RST 时将处理该例程。
MAN_ON	42.1	BOOL	TRUE	如果输入“启用手动模式”被置位，则控制回路会中断。 手动值 MAN 将被设置为调节值。

参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

8.4.4.8 静态变量 TCONT_CP

表格 8- 22

参数	地址	数据类型	默认值	说明
DEADB_W	44.0	REAL	0.0	将死区应用到控制偏差。 “死区宽度”(Deadband width) 输入决定死区的大小。 有效值取决于所用的传感器。
I_ITLVAL	48.0	REAL	0.0	可在输入 I_ITL_ON 设置积分器的输出。 初始化值应用于输入“I 作用的初始化值”。 在重启 COM_RST = TRUE 期间，将 I 作用设为初始化值。 允许介于 -100 到 100 % 之间的值。
LMN_HLM	52.0	REAL	100.0	输出值始终限制在上限和下限之间。 输入“调节值上限”(Manipulated value high limit) 指定上限。 LMN_HLM > LMN_LLM
LMN_LLM	56.0	REAL	0.0	输出值始终限制在上限和下限之间。 输入“调节值下限”(Manipulated value low limit) 指定下限。 LMN_LLM < LMN_HLM
PV_FAC	60.0	REAL	1.0	“过程值因子”输入与“过程值 I/O”相乘。 该输入用于标定过程值的范围。
PV_OFFS	64.0	REAL	0.0	“过程值偏移量”输入与“过程值 I/O”相加。 该输入用于标定过程值的范围。
LMN_FAC	68.0	REAL	1.0	“输出值因子”输入与输出值相乘。 该输入用于标定输出值的范围。
LMN_OFFS	72.0	REAL	0.0	“输出值偏移量”输入与输出值相加。 该输入用于标定输出值的范围。
PER_TM	76.0	REAL	1.0 s	在参数 PER_TM 中输入脉宽调制的周期持续时间。 周期持续时间与脉冲整形器采样时间的关系决定着脉宽调制的精度。 PER_TM ≥ CYCLE
P_B_TM	80.0	REAL	0.02 s	可在参数“最小脉冲/中断时间”中分配最小脉冲或中断时间。P_B_TM 在内部限制在 > CYCLE_P 之内。
TUN_DLMN	84.0	REAL	20.0	控制器调节的过程激发是由 TUN_DLMN 中的输出值阶跃变化引起的。 允许介于 -100 到 100 % 之间的值。

参数	地址	数据类型	默认值	说明
PER_MODE	88.0	INT	0	可使用此开关输入 I/O 模块的类型。 然后，在 PV 输出中对输入 PV_PER 中的过程值进行如下标定： • PER_MODE = 0：热电偶；PT100/NI100；标准 PV_PER * 0.1 单位： °C，°F • PER_MODE = 1：PT100/NI100；气候型 PV_PER * 0.01 单位： °C，°F • PER_MODE = 2：电流/电压 PV_PER * 100/27648 单位： %
PVPER_ON	90.0	BOOL	FALSE	如果要从 I/O 读取过程值，输入 PV_PER 必须与 I/O 互连，且输入“启用过程值 I/O”必须置位。
I_ITL_ON	90.1	BOOL	FALSE	可在输入 I_ITLVAL 设置积分器的输出。 必须为此置位输入“设置 I 作用”。
PULSE_ON	90.2	BOOL	FALSE	如果设置 PULSE_ON = TRUE，则会激活脉冲整形器。
TUN_KEEP	90.3	BOOL	FALSE	仅当 TUN_KEEP 切换为 FALSE 时，才会切换到自动模式。
ER	92.0	REAL	0.0	有效的控制偏差通过输出“控制偏差”输出。 有效值取决于所用的传感器。
LMN_P	96.0	REAL	0.0	“P 作用”输出包含调节变量的比例作用。
LMN_I	100.0	REAL	0.0	“积分作用”输出包含调节变量的积分作用。
LMN_D	104.0	REAL	0.0	“D 作用”输出包含调节变量的微分作用。

参数	地址	数据类型	默认值	说明
PHASE	108.0	INT	0	在输出 PHASE 中指示控制器调节的当前阶段。 • PHASE = 0：无调节模式；自动模式或手动模式 • PHASE = 1：启动调节准备就绪；检查参数、等待激发、测量采样时间 • PHASE = 2：实际调节： 使用常量输出值搜索拐点。 在背景数据块中输入采样时间。 • PHASE = 3：计算过程参数。 在进行调节之前保存有效的控制器参数。 • PHASE = 4：控制器设计 • PHASE = 5：根据新的调节变量跟踪控制器 • PHASE = 7：验证过程类型
STATUS_H	110.0	INT	0	STATUS_H 通过在加热过程中对拐点的搜索指示诊断值。
STATUS_D	112.0	INT	0	STATUS_D 通过加热过程中的控制器设计指示诊断值。
QTUN_RUN	114.0	BOOL	0	已应用整定调节变量，整定已启动并仍处于阶段 2（搜索拐点）。
PI_CON	116.0	STRUCT		PI 控制器参数
GAIN	+0.0	REAL	0.0	PI 控制器增益 %/物理单位
TI	+4.0	REAL	0.0 s	PI 积分时间 [s]
PID_CON	124.0	STRUCT		PID 控制器参数
GAIN	+0.0	REAL	0.0	PID 控制器增益
TI	+4.0	REAL	0.0s	PID 积分时间 [s]
TD	+8.0	REAL	0.0s	PID 微分作用时间 [s]
PAR_SAVE	136.0	STRUCT		PID 参数保存在此结构中。
PFAC_SP	+0.0	REAL	1.0	设定值变化的比例因子 允许使用介于 0.0 到 1.0 之间的值。

参数	地址	数据类型	默认值	说明
GAIN	+4.0	REAL	0.0	控制器增益 %/物理单位
TI	+8.0	REAL	40.0 s	积分时间 [s]
TD	+12.0	REAL	10.0 s	微分作用时间 (s)
D_F	+16.0	REAL	5.0	微分因子 允许使用介于 5.0 到 10.0 之间的值。
CON_ZONE	+20.0	REAL	100.0	控制区域范围 如果控制偏差大于控制区范围，则将输出输出值上限作为输出值。 如果控制偏差小于负控制区范围，则将输出输出值下限作为输出值。 CON_ZONE ≥ 0.0
CONZ_ON	+24.0	BOOL	FALSE	启用控制区
PFAC_SP	162.0	REAL	1.0	存在设定值变化时，PFAC_SP 指定 P 作用的有效性。 该值将在 0 和 1 之间进行设置。 • 1: 如果设定值发生变化，P 作用完全有效。 • 0: 如果设定值发生变化，P 作用无效。 允许使用介于 0.0 到 1.0 之间的值。
GAIN	166.0	REAL	2.0	“比例增益”输入用于指定控制器放大率。 为 GAIN 加上负号可反转控制的方向。 %/物理单位
TI	170.0	REAL	40.0 s	“积分时间”（积分作用时间）输入用于定义积分器的时间响应。
TD	174.0	REAL	10.0 s	“微分作用时间”（微分时间）输入用于确定微分器的时间响应。
D_F	178.0	REAL	5.0	此微分因子决定 D 作用的延迟。 D_F = 微分作用时间/“D 作用的延迟” 允许使用介于 5.0 到 10.0 之间的值。
CON_ZONE	182.0	REAL	100.0	如果控制偏差大于控制区范围，则将输出输出值上限作为输出值。 如果控制偏差小于负控制区范围，则将输出输出值下限作为输出值。 有效值取决于所用的传感器。
CONZ_ON	186.0	BOOL	FALSE	可以用 CONZ_ON =TRUE 来启用控制区域。

参数	地址	数据类型	默认值	说明
TUN_ON	186.1	BOOL	FALSE	如果 TUN_ON=TRUE，则将对输出值取平均值，直到由于设定值阶跃变化或 TUN_ST=TRUE 而启用输出值激发 TUN_DLMN 为止。
TUN_ST	186.2	BOOL	FALSE	如果在控制器调节期间操作点的设定值保持恒定，则 TUN_ST=1 将激活输出值阶跃变化（变化量为 TUN_DLMN）。
UNDO_PAR	186.3	BOOL	FALSE	从数据结构 PAR_SAVE 加载控制器参数 PFAC_SP、GAIN、TI、TD、D_FCONZ_ON 和 CON_ZONE （仅在手动模式下）。
SAVE_PAR	186.4	BOOL	FALSE	在数据结构 PAR_SAVE 中保存控制器参数 PFAC_SP、GAIN、TI、TD、D_F、CONZ_ON 和 CON_ZONE。
LOAD_PID	186.5	BOOL	FALSE	根据 PID_ON，从数据结构 PI_CON 或 PID_CON 加载控制器参数 GAIN、TI,TD（仅在手动模式下）
PID_ON	186.6	BOOL	TRUE	在输入 PID_ON 中，可以指定已调整的控制器作为 PI 控制器还是作为 PID 控制器运行。 • PID 控制器： PID_ON = TRUE • PI 控制器： PID_ON = FALSE 但是，对于某些过程类型，尽管 PID_ON = TRUE，仍然只能设计 PI 控制器。
GAIN_P	188.0	REAL	0.0	已识别的过程增益。 如果是过程类型 I，则对 GAIN_P 的估计往往会过低。
TU	192.0	REAL	0.0	已识别的过程时间延迟。 TU ≥ 3*CYCLE
TA	196.0	REAL	0.0	已识别的过程恢复时间。 如果是过程类型 I，则对 TA 的估计往往会过低。
KIG	200.0	REAL	0.0	调节变量激发从 0 增加至 100 % 时的最大过程值上升 [1/s] GAIN_P = 0.01 * KIG * TA
N_PTN	204.0	REAL	0.0	此参数指定过程的顺序。 也可以接受“非整数值”。 允许使用介于 1.01 到 10.0 之间的值。
TM_LAG_P	208.0	REAL	0.0	PTN 模型的时间常数（仅适用于 N_PTN >= 2 的实际值）。
T_P_INF	212.0	REAL	0.0	从过程激发到拐点的时间。
P_INF	216.0	REAL	0.0	从过程激发到拐点的过程值变化。 有效值取决于所用的传感器。
LMN0	220.0	REAL	0.0	调节开始时的输出值 在阶段 1 中进行检测（平均值）。 允许介于 0 到 100 % 之间的值。

参数	地址	数据类型	默认值	说明
PV0	224.0	REAL	0.0	调节开始时的过程值
PVDT0	228.0	REAL	0.0	调节开始时的过程值转换速率 [1/s] 符号已调整。
PVDT	232.0	REAL	0.0	当前过程值转换速率 [1/s] 符号已调整。
PVDT_MAX	236.0	REAL	0.0	每秒过程值的最大变化量 [1/s] 拐点处过程值最大微分（符号已调整，始终 > 0）；用于计算 TU 和 KIG。
NOI_PVDT	240.0	REAL	0.0	PVDT_MAX 中的噪声作用 (%) 噪声作用越高，控制参数精度越差（反应越慢）。
NOISE_PV	244.0	REAL	0.0	过程值中的绝对噪声 第 1 阶段中最大过程值和最小过程值之间的差。
FIL_CYC	248.0	INT	1	平均值过滤器的循环次数 过程值通过 FIL_CYC 周期来确定。如果需要，可将 FIL_CYC 从 1 增加到最大值 1024。
POI_CMAX	250.0	INT	2	拐点之后的最大周期数 此时间用于查找测量噪声的另一个（例如更好的）拐点。 该时间过后，才会完成调节。
POI_CYCL	252.0	INT	0	拐点之后的周期数

参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

8.4.4.9 参数 STATUS_H

STATUS_H	说明	解决方法
0	默认控制器参数或无控制器参数/无新的控制器参数	
10000	调整完成 + 找到适合的控制器参数	
2xxxx	调整完成 + 控制器参数不确定	
2xx2x	未达到拐点（仅当通过设定值阶跃更改激发时）	如果控制器产生振荡，则应弱化控制器参数，或使用较小的调节值偏差 TUN_DLMN 重复进行测试。
2x1xx	估计错误 (TU < 3*CYCLE)	降低 CYCLE 并重试。 仅 PT1 过程的特殊情况： 不重复测试；如果需要，可以调低控制器参数。
2x3xx	估计错误 TU 过高	在更好的条件下重复进行测试。
21xxx	估计错误 N_PTN < 1	在更好的条件下重复进行测试。
22xxx	估计错误 N_PTN > 10	在更好的条件下重复进行测试。
3xxxx	参数分配错误导致第 1 阶段的调整取消：	
30002	有效的调节值偏差 < 5%	更正调节值偏差 TUN_DLMN。
30005	采样时间 CYCLE 和 CYCLE_P 的偏差大于测量值的 5%。	将 CYCLE 和 CYCLE_P 与循环中断优先级等级的循环时间进行比较，并注意任何循环调度程序。 检查 CPU 负载。 过载的 CPU 会导致采样时间延长，与 CYCLE 或 CYCLE_P 不一致。

说明

如果在阶段 1 或 2 取消调节，则将设置 STATUS_H = 0。 但是，STATUS_D 仍然显示上一控制器计算的状态。

STATUS_D 的值越高，控制过程的序号就越高，TU/TA 的比率就越大，从而控制器参数的控制作用就越平缓。

参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

8.4.4.10 参数 STATUS_D

STATUS_D	说明
0	不计算任何控制器参数。
110	N_PTN <= 1.5 过程类型 I 快速
121	N_PTN > 1.5 过程类型 I
200	N_PTN > 1.9 过程类型 II（过渡范围）
310	N_PTN >= 2.1 过程类型 III 快速
320	N_PTN > 2.6 过程类型 III
111, 122, 201, 311, 321	已通过阶段 7 对参数进行修正。

说明

STATUS_D 的值越高，控制过程的序号就越高，TU/TA 的比率就越大，从而控制器参数的控制作用就越平缓。

参见

脉冲发生器的工作原理 (页 543)

TCONT_CP 方框图 (页 546)

8.4.5 TCONT_S

8.4.5.1 TCONT_S 说明

TCONT_S 指令用于 SIMATIC S7 自动化系统中，可以使用具有积分行为的执行器的二进制调节值输出信号来控制工艺温度过程。 此功能基于采样控制器的 PI 控制算法。 步进控制器在没有位置反馈信号的情况下运行。

应用

也可以将串级控制中的控制器用作辅助控制器。 通过设定值输入 SP_INT 指定执行器位置。 在这种情况下，必须将过程值输入和参数 TI（积分时间）设置为零。 温度控制方面的应用示例包括使用脉冲中断激活的加热功率控制，以及使用蝶阀进行的冷却控制。 要完全关闭阀，调节变量 (ER*GAIN) 应该为负值。

调用

必须等距调用指令 TCONT_S。 要达到该目的，可以使用循环中断优先级等级（例如，S7-300 的 OB35）。 在 CYCLE 参数中指定采样时间。

如果将指令 TCONT_S 作为多重背景数据块调用，将不会创建任何工艺对象。没有参数分配接口或调试接口可用。 必须直接在多重背景数据块中为 TCONT_S 分配参数,并通过监视表格进行调试。

CYCLE 采样时间

CYCLE 采样时间与两次调用的时间差（考虑缩减比率的循环中断 OB 的循环时间）一致。

控制器采样时间不应超出计算出的控制器积分时间 (TI) 的 10%。 通常，必须将采样时间设置为非常低的值，以获得需要的步进控制器精度。

所需精度 G	MTR_TM	CYCLE = MTR_TM*G	注释
0.5 %	10 s	0.05 s	采样时间由所需的步进控制器的精度决定。

启动

TCONT_S 指令具有一个初始化例程，在设置输入参数 COM_RST = TRUE 时将运行该例程。 在执行完初始化例程后，块将 COM_RST 重新设置成 FALSE。 所有输出都被设置成各自的初始值。 如果需要在 CPU 重启时执行初始化，则可在 OB 100 中调用此块 (COM_RST = TRUE)。

参见

TCONT_S 方框图 (页 565)

8.4.5.2 TCONT_S 的工作模式

设定值分支

在输入 SP_INT 中输入浮点格式的设定值，作为物理值或者百分比值。 用于形成控制偏差的设定值和过程值必须采用相同的单位。

过程值选项 (PVPER_ON)

根据 PVPER_ON，可读取 I/O 格式或浮点数格式的过程值。

PVPER_ON	过程值输入
TRUE	通过输入 PV_PER 中的模拟量 I/O (PIW xxx) 读取过程值。
FALSE	在输入 PV_IN 处采集浮点格式的过程值。

过程值格式转换 CRP_IN (PER_MODE)

函数 CRP_IN 按照下列规则并根据 PER_MODE 开关设置，将 I/O 值 PV_PER 转换为浮点格式：

PER_MODE	CRP_IN 的输出	模拟量输入类型	单位
0	PV_PER * 0.1	热电偶； PT100/NI100；标准	°C；°F
1	PV_PER * 0.01	PT100/NI100；气候型；	°C；°F
2	PV_PER * 100/27648	电压/电流	%

过程值标定 PV_NORM（PF_FAC，PV_OFFS）

函数 PV_NORM 根据以下规则计算 CRP_IN 的输出：

“PV_NORM 的输出” = “CRP_IN 的输出” * PV_FAC + PV_OFFS

有以下用途：

- 以 PV_FAC 为过程值因子、PV_OFFS 为过程值偏移量进行过程值调整。
- 将温度值标准化为百分比值

  如果要以百分比的形式输入设定值，现在必须将测得的温度值转换成百分比值。
- 将百分比值标准化为温度值

  如果想要以物理温度单位输入设定值，现在必须将测得的电压/电流值转换成温度值。

参数计算：

- PV_FAC = PV_NORM 的范围/CRP_IN 的范围；
- PV_OFFS = LL (PV_NORM) - PV_FAC * LL(CRP_IN)；

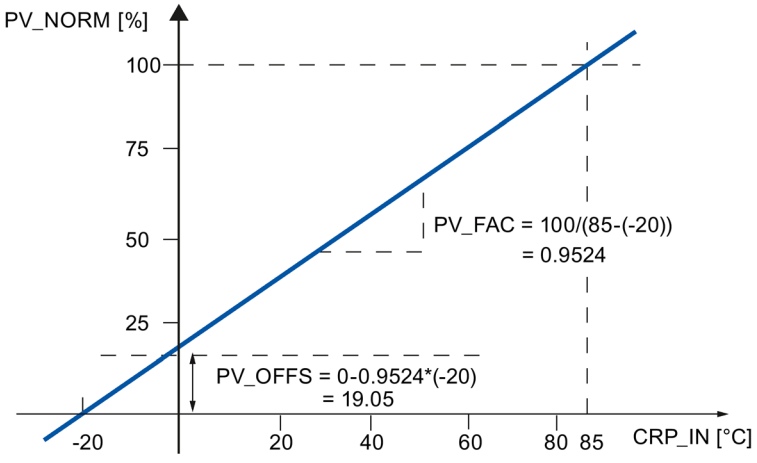
  其中，LL： 下限

标准化通过默认值（PV_FAC = 1.0 和 PV_OFFS = 0.0）关闭。 在 PV 输出中输出有效过程值。

过程值标准化示例

如果要以百分比的形式输入设定值，并且 CRP_IN 的温度范围为 -20 到 85 °C，则必须将温度范围标准化为百分比值。

下图给出的示例说明了如何将 -20 到 85 °C 的温度范围修改为 0 到 100% 的内部标定：



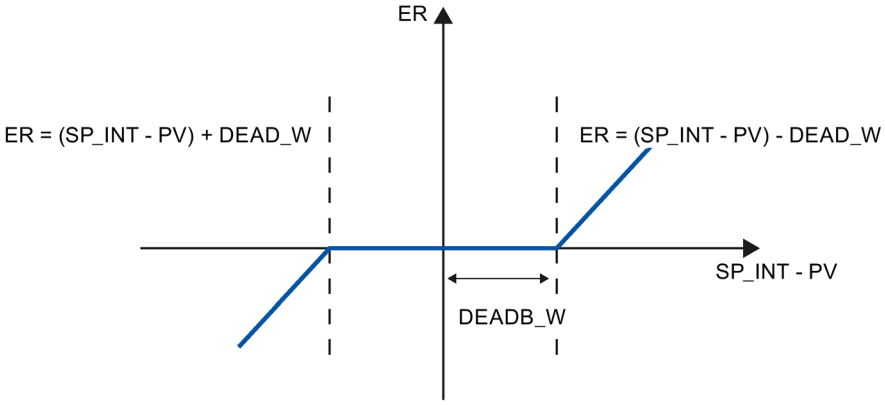
形成控制偏差

在到达死区之前，设定值与过程值的差值就是控制偏差。

设定值与过程值的单位必须相同。

死区 (DEADB_W)

为了抑制由于调节变量量化所引起的小幅持续振荡（例如，在使用 PULSEGEN 进行脉宽调制时），可对控制偏差使用死区 (DEADBAND)。 DEADB_W = 0.0 时，死区关闭。



PI 步进控制器算法

指令 TCONT_S 在没有位置反馈的情况下运行。 PI 算法的 I 作用和假定的位置反馈信号在积分器 (INT) 中计算，并作为反馈值与其余 P 作用进行比较。 差值将应用到三位元件 (THREE_ST) 以及为控制阀生成脉冲的脉冲整形器 (PULSEOUT)。 调整三位元件的响应阈值会降低控制器的切换频率。

当设定值发生变化时，会弱化 P 作用

为了防止过调，可以使用参数“用于设定值更改的比例因子”(PFAC_SP) 来弱化 P 作用。可以使用 PFAC_SP 在 0.0 和 1.0 之间进行连续选择，以决定设定值发生变化时 P 作用的有效程度：

- PFAC_SP = 1.0：如果设定值发生变化，P 作用完全有效
- PFAC_SP = 0.0：如果设定值发生变化，P 作用无效

像在连续控制器的情况中，如果电机运行时间 MTR_TM 比恢复时间 TA 小，且比率是 TU/TA < 0.2，则 PFAC_SP < 1.0 的值可以减小过调。 如果 MTR_TM 达到 TA 的 20%，则只能略有改进。

前馈控制

可在 DISV 输入中添加扰动变量。

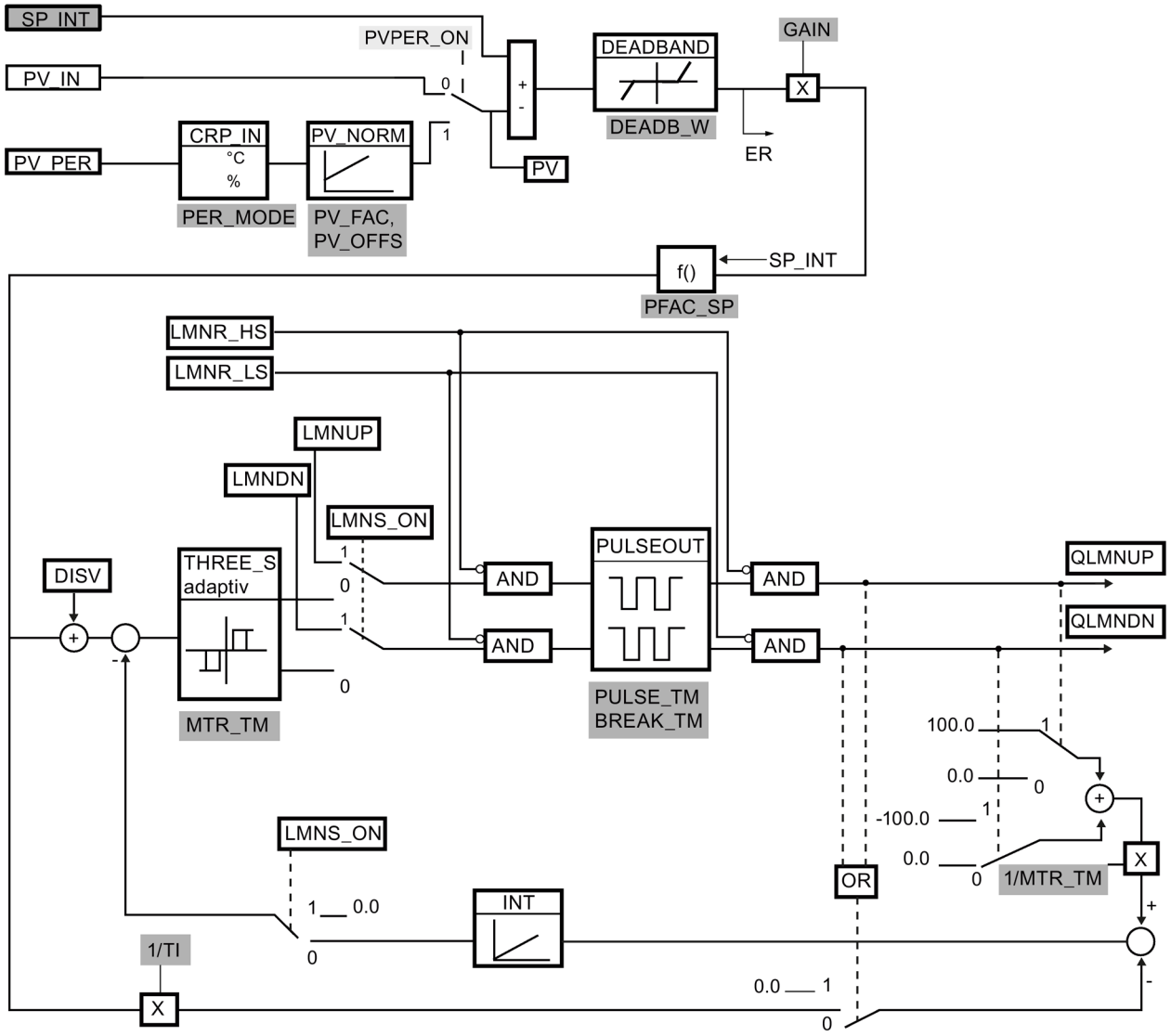
手动值处理（LMNS_ON、LMNUP、LMNDN）

可以通过 LMNS_ON 在手动与自动模式之间切换。 在手动模式下，执行器停止，积分作用 (INT) 在内部置位为 0。 可通过 LMNUP 和 LMNDN 将执行器调整为 OPEN 和 CLOSED。 因此，切换到自动模式会产生干扰。 由于 GAIN 的原因，现有控制偏差会导致内部调节变量的阶跃变化。 然而，执行器的积分分量将导致斜坡形的过程激发。

参见

TCONT_S 方框图 (页 565)

8.4.5.3 TCONT_S 方框图



参数组态接口
指令调用接口
参数组态接口、调用接口

参见

TCONT_S 说明 (页 559)

TCONT_S 的工作模式 (页 561)

TCONT_S 输入参数 (页 567)

TCONT_S 输出参数 (页 568)

TCONT_S 输入/输出参数 (页 568)

TCONT_S 静态变量 (页 569)

8.4.5.4 TCONT_S 输入参数

表格 8- 23

参数	地址	数据类型	默认值	说明
CYCLE	0.0	REAL	0.1 s	在此输入中，输入控制器的采样时间。 CYCLE ≥ 0.001
SP_INT	4.0	REAL	0.0	“内部设定值”输入用于指定设定值。 有效值取决于所用的传感器。
PV_IN	8.0	REAL	0.0	在“过程变量输入”中，可以将参数分配给调试值，或者互连浮点格式的外部过程值。 有效值取决于所用的传感器。
PV_PER	12.0	INT	0	I/O 格式的过程值在输入“过程值 I/O”中与控制器互连。
DISV	14.0	REAL	0.0	对于前馈控制，扰动变量与输入“扰动变量”互连。
LMNR_HS	18.0	BOOL	FALSE	在输入“位置反馈的上端停止位信号”中互连信号“控制阀位于上端停止位”。 • LMNR_HS=TRUE：控制阀位于上端停止位。
LMNR_LS	18.1	BOOL	FALSE	在输入“位置反馈的下端停止位信号”中互连信号“控制阀位于下端停止位”。 • LMNR_LS=TRUE： 控制阀位于下端停止位。
LMNS_ON	18.2	BOOL	TRUE	在“启用调节信号的手动模式”处将调节值信号处理模式切换为手动模式。
LMNUP	18.3	BOOL	FALSE	在调节信号的手动模式下，在输入参数“调节信号上升”中操作输出参数 QLMNUP。
LMNDN	18.4	BOOL	FALSE	在调节信号的手动模式下，在输入参数“调节信号下降”中操作输出参数 QLMNDN。

参见

TCONT_S 方框图 (页 565)

8.4.5.5 TCONT_S 输出参数

表格 8- 24

参数	地址	数据类型	默认值	说明
QLMNUP	20.0	BOOL	FALSE	如果置位输出“调节值信号上升”，则应打开控制阀。
QLMNDN	20.1	BOOL	FALSE	如果置位输出“调节值信号下降”，则应关闭控制阀。
PV	22.0	REAL	0.0	有效的过程值在“过程值”输出中输出。
ER	26.0	REAL	0.0	在“误差信号”输出中输出有效系统偏差。

参见

TCONT_S 方框图 (页 565)

8.4.5.6 TCONT_S 输入/输出参数

表格 8- 25

参数	地址	数据类型	默认值	说明
COM_RST	30.0	BOOL	FALSE	该块具有一个初始化例程，在置位输入 COM_RST 时将处理该例程。

参见

TCONT_S 方框图 (页 565)

8.4.5.7 TCONT_S 静态变量

表格 8- 26

参数	地址	数据类型	默认值	说明
PV_FAC	32.0	REAL	1.0	“过程值因子”输入与过程值相乘。 该输入用于标定过程值的范围。
PV_OFFS	36.0	REAL	0.0	“过程值偏移量”输入与过程值相加。 该输入用于标定过程值的范围。 有效值取决于所用的传感器。
DEADB_W	40.0	REAL	0.0	将死区应用到控制偏差。 “死区宽度”(Deadband width) 输入决定死区的大小。 DEADB_W ≥ 0.0
PFAC_SP	44.4	REAL	1.0	存在设定值变化时，PFAC_SP 指定 P 作用的有效性。 • 1: 如果设定值发生变化，P 作用完全有效。 • 0: 如果设定值发生变化，P 作用无效。 允许使用介于 0.0 到 1.0 之间的值。
GAIN	48.0	REAL	2.0	“比例增益”输入用于指定控制器放大率。 为 GAIN 加上负号可反转控制的方向。 %/物理单位
TI	52.0	REAL	40.0 s	“积分时间”（积分作用时间）输入用于定义积分器的时间响应。
MTR_TM	56.0	REAL	30 s	在“电机动作时间”参数中输入控制阀从一个停止位到另一个停止位的运行时间。 MTR_TM ≥ CYCLE
PULSE_TM	60.0	REAL	0.0 s	可以在“最短脉冲周期”参数中组态最短脉冲持续时间。
BREAK_TM	64.0	REAL	0.0 s	可在参数“最小中断时间”中分配最小中断时间。

参数	地址	数据类型	默认值	说明
PER_MODE	68.0	INT	0	可使用此开关输入 I/O 模块的类型。 然后，在 PV 输出中对输入 PV_PER 中的过程值进行如下标定： • PER_MODE = 0：热电偶；PT100/NI100；标准 PV_PER * 0.1 单位： °C，°F • PER_MODE = 1：PT100/NI100；气候型 PV_PER * 0.01 单位： °C，°F • PER_MODE = 2：电流/电压 PV_PER * 100/27648 单位： %
PVPER_ON	70.0	BOOL	FALSE	如果要从 I/O 读取过程值，输入 PV_PER 必须与 I/O 互连，且输入“启用过程值 I/O”必须置位。

参见

TCONT_S 方框图 (页 565)

8.4.6 集成的系统功能

8.4.6.1 CONT_C_SF

CONT_C_SF

指令 CONT_C_SF 集成在 S7-300 紧凑型 CPU 中。 加载期间，不得向 S7-300 CPU 传输该指令。 其功能范围与指令 CONT_C 的相同。

参见

CONT_C 说明 (页 509)

CONT_C 的工作原理 (页 510)

CONT_C 方框图 (页 512)

输入参数 CONT_C (页 513)

CONT_C 输出参数 (页 515)

8.4.6.2 CONT_S_SF

CONT_S_SF

指令 CONT_S_SF 集成在 S7-300 紧凑型 CPU 中。 加载期间，不得向 S7-300 CPU 传输该指令。 其功能范围与指令 CONT_S 的相同。

参见

CONT_S 说明 (页 516)

CONT_S 工作模式 (页 517)

CONT_S 方框图 (页 518)

CONT_S 输入参数 (页 519)

CONT_S 输出参数 (页 520)

8.4.6.3 PULSEGEN_SF

PULSEGEN_SF

指令 PULSEGEN_SF 集成在 S7-300 紧凑型 CPU 中。 加载期间，不得向 S7-300 CPU 传输该指令。 其功能范围与指令 PULSEGEN 的相同。

参见

PULSEGEN 说明 (页 521)

PULSEGEN 的工作模式 (页 522)

PULSEGEN 的工作模式 (页 525)

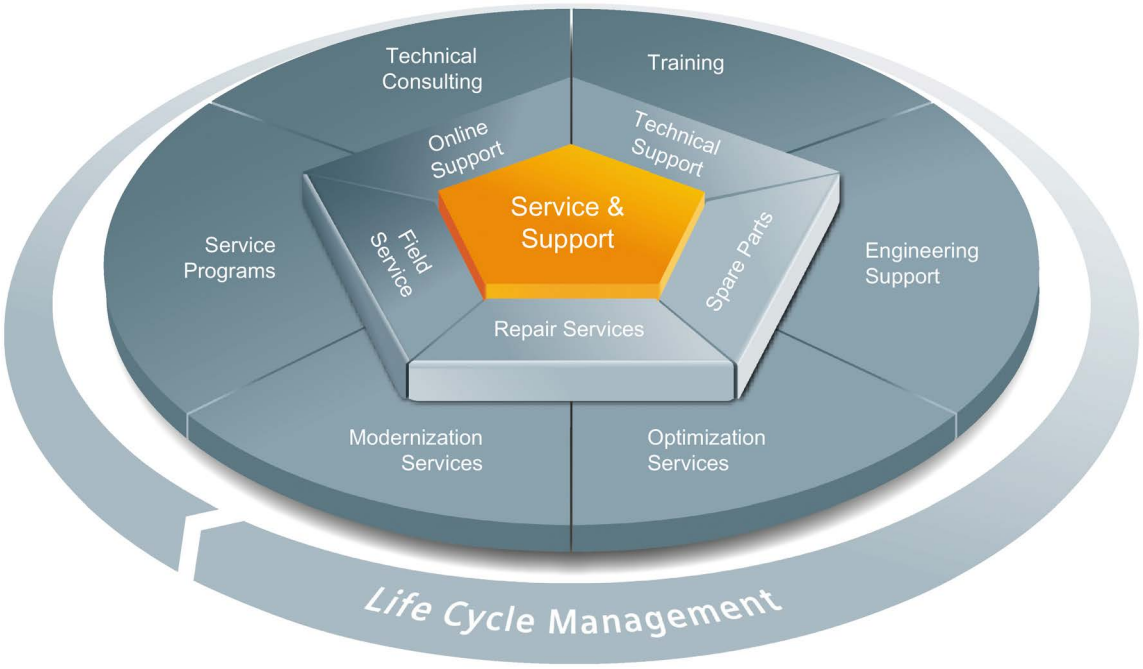
三位控制 (页 526)

两位控制 (页 529)

PULSEGEN 输入参数 (页 530)

PULSEGEN 输出参数 (页 531)

服务与支持



整个生命周期内的全面非凡服务

对于设备制造商、解决方案供应商以及工厂操作员而言： 西门子工业自动化与驱动技术集团将为制造和加工行业内所有领域中的各种不同用户提供全面服务。

为了配合我们的产品和系统，我们提供有集成的结构化服务，以便在您设备或工厂生命周期的每个阶段都提供有高价值的服务和支持：从规划和实施到调试，以及维护和现代化改造，一应俱全。

我们的服务和支持时刻伴在您的左右，为您解决所有的西门子自动化和驱动技术问题。我们在 100 多个国家为设备和工厂生命周期的所有阶段都提供有现场支持。

在您的身边，将有一支由经验丰富的专家所组成的团队，为您提供积极的支持和专业技术。 即使您与我们横跨多个大陆，我们的员工也将定期为您开展各种培训课程并与您保持密切的联系，以确保在各种领域为您提供可靠的服务。

在线支持

全面的在线信息平台，可以随时随地为您提供全面的服务与支持。

您可以在以下 Internet 地址上找到在线支持。

技术咨询

全面地为您的项目进行规划和设计： 我们的规划和设计内涵盖了实际状态的详细分析、目标定义、产品和系统问题咨询，以及自动化解决方案的创建，无所不及。

技术支持

除了为客户提供有关技术问题的专家建议，我们还提供大量针对我们产品和系统的按需服务。

您可以在以下 Internet 地址上找到技术支持。

培训

我们为您提供的各种实践专业知识，助您在激烈的竞争中处于不败之地。

您可以在以下 Internet 地址上找到培训课程。

工程组态支持

在项目工程组态和开发阶段，我们将专门针对您的要求进行量身定制的服务支持，涵盖了从自动化项目组态到实施的所有阶段。

现场服务

我们的现场服务为您提供调试和维护服务，以确保您的设备和工厂始终处于运行状态。

备件

在全世界的每个行业中，持久的可靠性是工厂和系统在运作时的必要条件。 我们通过遍布全球的网络和最优秀的物流链，从一开始就为您提供所需的支持，使工厂和系统运行通畅 。

维修

停机会在工厂中导致各种问题的产生并由此引发不必要的成本。 我们通过遍布全球的维修设施，可以帮助您将这两者的成本降至最低。

优化

在设备和工厂的服务寿命期间，通常有很大的空间来提高生产力或降低成本。

为了帮助您实现这一终极目标，我们提供了全面的优化服务。

现代化改造

在需要现代化改造时，您也将得到我们的支持，我们将提供有从规划阶段直到调试完成的全面服务。

服务计划

我们的服务计划是针对自动化和驱动系统或产品组特选的各种服务包。 各个服务之间相互协调以确保全面覆盖整个生命周期并对产品和系统的使用进行优化。

服务计划中的服务可以随时灵活更改并单独使用。

服务计划示例：

- 服务合同
- 工厂 IT 安全服务
- 驱动工程生命周期服务
- SIMATIC PCS 7 生命周期服务
- SINUMERIK 机床增效及制造信息化
- SIMATIC 远程支持服务

优势一览：

- 减少停机时间，提高生产力
- 量身定制各种服务，降低了维护成本
- 可预先计算并规划的成本
- 响应时间和备件交付时间有保障，服务十分可靠
- 客户服务人员将为额外任务提供支持以及解决方案
- 一站式全面服务，更少的联络，更多的专业技术

联系方式

在全球范围内就近为您提供各种服务： 针对工业自动化和驱动技术集团提供的所有产品，我们都为您提供咨询、销售、培训、服务、支持、备件等服务。

有关人员联系方式，请访问 Internet 上的联系方式数据库。

索引

G

R

Z H

F