SIEMENS

SIMATIC

模块化PID控制

手册

2003年11月版

A5E01156034-01

安全指南

本手册包括了保证人身安全与保护本产品及相连设备所应遵守的注意事项。这些注意事项在手册中均以下列符号加以突出，并根据危险等级标识如下:

危险

表示如果不采取适当的预防措施，将会导致死亡、严重的人身伤害或重大的财产损失。

警告

表示如果不采取适当的预防措施，可能导致死亡、严重人身伤害或重大财产损失。

当心

表示如果不采取适当的预防措施，可能导致轻微的人身伤害。

当心

表示如果不采取适当的预防措施，可能导致财产损失。

注意

提醒您注意有关产品、产品使用的特别重要的信息，或文档的某些特定部分。

合格人员

只有**合格人员**才允许安装和操作该设备。合格人员是指被授权按照既定安全惯例和标准，对线路、设备和系统进行调试、接地和标记的人员。

正确应用

请注意如下事项:

警告

本设备及其部件只能用于产品目录或技术说明书中所描述的范畴，并且只能与由西门子公司认可或推荐的第三方厂商提供的设备或部件一起使用。

只有正确地运输、保管、配置和安装，并且按照推荐的方式操作和维护，产品才能正常、安全地运行。

注册商标

SIMATIC®、SIMATIC HMI®和SIMATIC NET®是SIEMENS AG的注册商标。

本文档中的其它一些标志也是注册商标，如果任何第三方出于个人目的而使用，都会侵犯商标所有者的权利。

版权所有 © Siemens AG 2003 保留所有权利

未经明确的书面许可，不得复制、传播或使用本手册或所含内容。违者应对造成的损失承担责任。保留所有权利，包括实用新型或设计的专利许可权及注册权。

免责声明

我们已检查过本手册中的内容与所描述的硬件和软件相符。由于差错在所难免，我们不能保证完全一致。我们会定期审查本手册中的内容，并在后续版本中进行必要的更正。欢迎提出改进意见。

Siemens AG
Bereich Automation and Drives
Geschaeftsgebiet Industrial Automation Systems
Postfach 4848, D- 90327 Nuernberg

Siemens Aktiengesellschaft

© Siemens AG 2003
技术数据如有改动，恕不另行通知。

A5E01156034-01

前言

手册用途

当用户为自己的控制任务选择、配置和分配控制器块中的参数时，本手册可以为用户提供很大帮助。

本手册介绍了控制器块的功能，并且说明了如何使用启动和组态工具。

阅读本手册所需的基本知识

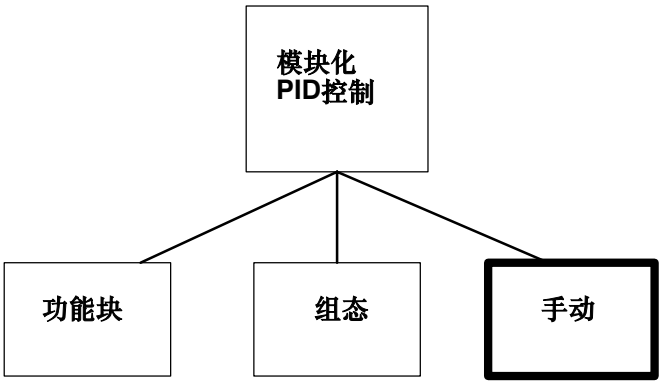
为了能够很好地理解本手册，用户应该熟悉自动化和过程控制工程设计领域。

此外，还应该知道如何在Windows操作系统下使用计算机或具有相似功能的设备(例如，编程设备)。由于模块化PID控制是基于STEP 7软件，所以用户还应该知道如何使用该软件。相关知识参见手册“使用STEP 7 V5.1编程”。

本手册在哪些地方适用？

本手册适用于软件包“模块化PID控制V5.0”和“模块化PID控制工具V5.0”。

该文档在整个信息环境中的位置



“模块化PID控制”数据包包括三个独立产品:

- “模块化PID控制FB”产品包含了功能块和实例。
- “模块化PID控制FB”产品主要包含了用于组态控制器块的工具。
  在后续部分中，我们将该产品称为“组态工具”。

适用对象

本手册是为下列读者编写的:

- S7程序员
- 控制系统的编程人员
- 操作员
- 维修人员

文本中的约定

为了使用户在手册中找到信息更容易，在手册中使用了一些约定:

- 首先在左页边的标题中快速浏览一下，获得各部分内容的主要信息。
- 处理特定标题的部分要么是回答与工具的功能相关的问题，要么是提供与必须或推荐的动作课程相关的信息。
- 指向与主题相关的更多信息的参考索引通过(*参见章节*或*部分x.y*)来指示。到其它文档的参考通过斜线/.../中的号码来指示。根据这些号码，用户可以在需要了解某个文档的完整标题时参见“附录”中的“参考”。
- 可以在手册 *“标准PID控制”* 中找到一个词汇表，其中包含了重要的控制器术语。

更多支持

如果有任何技术问题，请联系西门子代表或代理商。

可以在下列网址上找到联系人:

`http://www.siemens.com/automation/partner`

培训中心

西门子还提供了很多培训教程，使您熟悉SIMATIC S7自动化系统。请联系当地的培训中心或位于德国纽伦堡D 90327的培训总部，以获取详细资料:

电话: +49 (911) 895-3200

网址: `http://www.sitrain.com`

A&D技术支持中心

全球服务、全日制服务:

全球(纽伦堡) **技术支持中心** 每天24小时，全年365天为您服务 电话: +49 (180) 5050-222 传真: +49 (180) 5050-223 电子邮件: adsupport@ siemens.com GMT: +1：00		
欧洲/非洲(纽伦堡) **授权中心** 当地时间：周一至周五 上午8:00到下午5:00 电话: +49 (180) 5050-222 传真: +49 (180) 5050-223 电子邮件: adsupport@ siemens.com GMT: +1：00	**美国**(约翰逊城) **技术支持和授权中心** 当地时间：周一至周五， 上午8:00到下午5:00 电话: +1 (423) 262 2522 传真: +1 (423) 262 2289 电子邮件: simatic.hotline@ sea.siemens.com GMT: -5：00	**亚洲/澳大利亚**(北京) **技术支持和授权中心** 当地时间：周一至周五 上午8:00到下午5:00 电话: +86 10 64 75 75 75 传真: +86 10 64 74 74 74 电子邮件: adsupport.asia@ siemens.com GMT: +8：00
通常情况下，SIMATIC热线和授权热线的语言为德语和英语。		

Internet上的服务与支持

除了文档之外，还在Internet上在线提供专有技术信息:

http: //www.siemens.com/automation/service&support

可在其中查找下列内容:

- 公司简讯，不断提供产品的最新信息。
- 相应文档资料，可通过“Service & Support”(服务和支持)中的搜索功能查找。
- 论坛，世界各地的用户和专家可以在此交流经验。
- 当地自动化和驱动办事处。
- 在“Services”(服务)栏下的，关于现场服务、维修、备件的信息及其它信息。

目录

产品概述 - 模块化PID控制

1

1.1 产品模块化PID控制

模块化PID控制的原理

“模块化PID控制”软件产品包含了一系列**功能块**(FB)和**功能**(FC)，这些功能块和功能包含了创建控制器功能所需要的算法。因此，这是一个纯粹的软件控制器，可以在其中通过互连块来实现控制器功能。

在块库中，以实例的形式实现了一系列已经准备好，可供用户使用的控制器结构(单环路固定设置值控制器、比率控制器等)。可以复制并修改这些实例，以符合用户自己的控制任务。

当运行大量的控制环路时，通常的情况是部分环路的处理频率必须快于其它一些环路，而每个环路自身都必须以相等的时间间隔来处理。为了解决这类问题，特地提供了一个**环路时序表**(LP_SCHED)，通过它可以清晰简单地配置大型的控制系统。它同时确保了CPU上的负载均匀分配。

为了帮助用户安装并测试单个控制环路，该软件包还包含了一个*组态工具- “模块化PID控制工具”*。这包括一个环路监视器、一个用于操作和监视过程变量的曲线记录器，以及用于过程标识和PID参数优化的算法。

基本功能概述

在很多控制任务中，影响处理过程的经典PID控制器并不是唯一重要的元素；信号处理方面的要求往往更多。

因此，使用“模块化PID控制”软件包创建的控制器便包含了一系列的子功能；用户可以为这些子功能单独选择参数值。除了具有PID算法的实际控制器以外，还有一些功能也可以用于处理设定值和过程变量，以及用于调整计算出的可调节变量。

1.2 模块化PID控制的组件

模块化PID控制FB

“模块化PID控制FB”软件包由一个带有功能块的库和12个制作好的控制器实例组成。

可以通过SETUP程序在编程设备/PC上安装软件。在线帮助系统提供了与正在使用的子功能和单个参数相关的信息。

模块化PID控制功能

通过使用“启动和测试”工具，可以安装、设置和测试控制器的结构，以及优化PID参数。

*组态工具*包括一个环路监视器、一个曲线记录器以及用于设置或优化PID控制器参数的算法。在第5章中详细描述了*组态工具*。

模块化PID控制手册

关于此手册的内容的详细信息，请参考目录表。

1.3 环境和应用

硬件环境

使用“模块化PID控制”创建的控制器可以在S7-300和S7-400系列的可编程控制器(带有浮点和周期性中断的CPU)以及Win AC上运行。

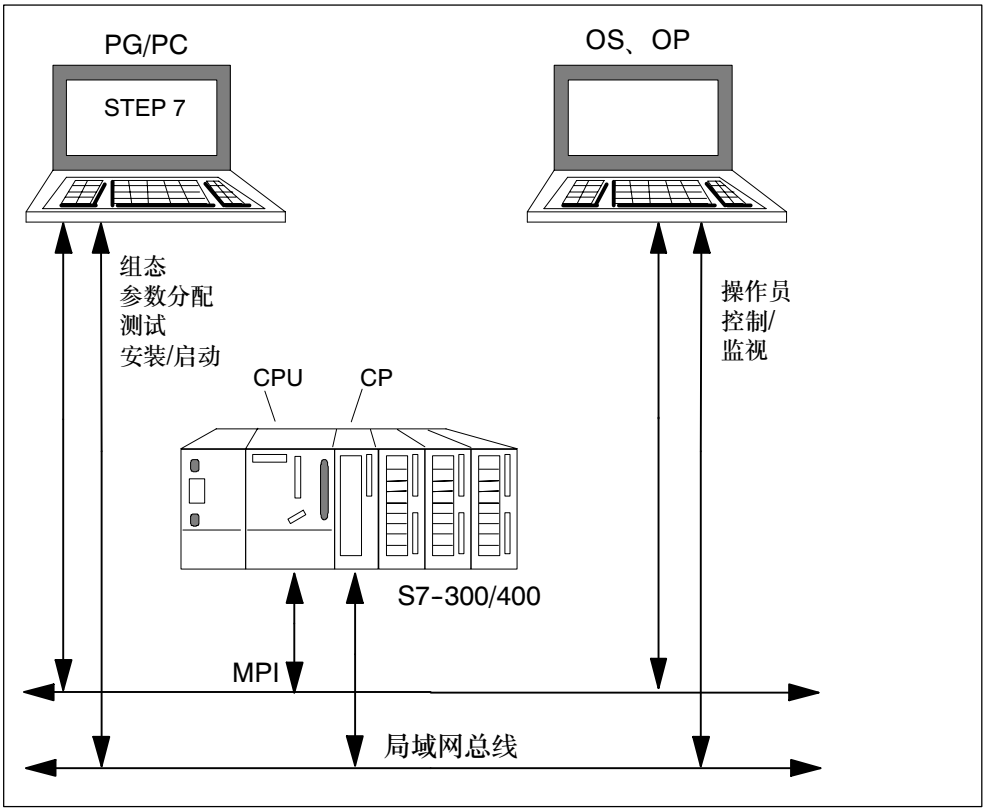


图1-1 “模块化PID控制”的环境

软件环境

模块化PID控制设计用在STEP 7程序组中。

模块化PID控制的配置软件可以在编程设备/PC上本地安装，也可以安装在网络中的中央网络驱动器上。

模块化PID控制的功能范围

不论是慢速过程(温度、罐液位)，还是非常快的过程(流速、电机速度)，都可以进行控制。可以实现下列控制器类型:

- 连续PID控制器
- 用于集成执行器的PID步进控制器
- 脉冲断开控制器

可以连接这些控制器，创建下列控制器结构中的某一种:

- 固定设定值控制器
- 层叠控制器
- 比率控制器
- 混合控制器
- 分段控制器
- 替代控制器
- 带有前馈控制的控制器
- 多变量控制器

功能描述

2

2.1 常规信息

方框图中参数和块名称所使用的约定惯例

参数的名称最大可以有8个字符长。

在命名参数时，有下列一些约定:

第一个字母:

Q	类型为BOOL (布尔型变量)的常规输出
SP	设定值
PV	过程变量
LMN	可调节变量或模拟量输出信号
DISV	干扰变量

随后的字母:

MAN	手动值
INT	内部
EXT	外部
_ON	布尔型变量，用于激活功能

调用数据

模块化PID控制软件包中的大部分块都需要环路特定的调用数据，例如完全重启动位和采样时间。这些数值是通过COM_RST和CYCLE传送的。

关于块参数(输入、输出和输入/输出参数)的注意事项

- **缺省:** 这是在创建实例时使用的缺省值。
- **允许使用的数值**: 为输入参数设置的数值不能超过允许的数值范围。在执行块时，系统**不**会检查数值范围。条目“技术取值范围”表示数值大约为 $\pm 10^{6}$ 的物理变量。

2.1.1 A_DEAD_B: 自适应死区

应用

如果过程变量受噪声影响，而且控制器实现最优设置，则噪声也将影响控制器输出。由于存在很高的切换频率(步进控制器)，这会增加执行器的磨损。抑制噪声可以防止控制器输出振荡。

方框图

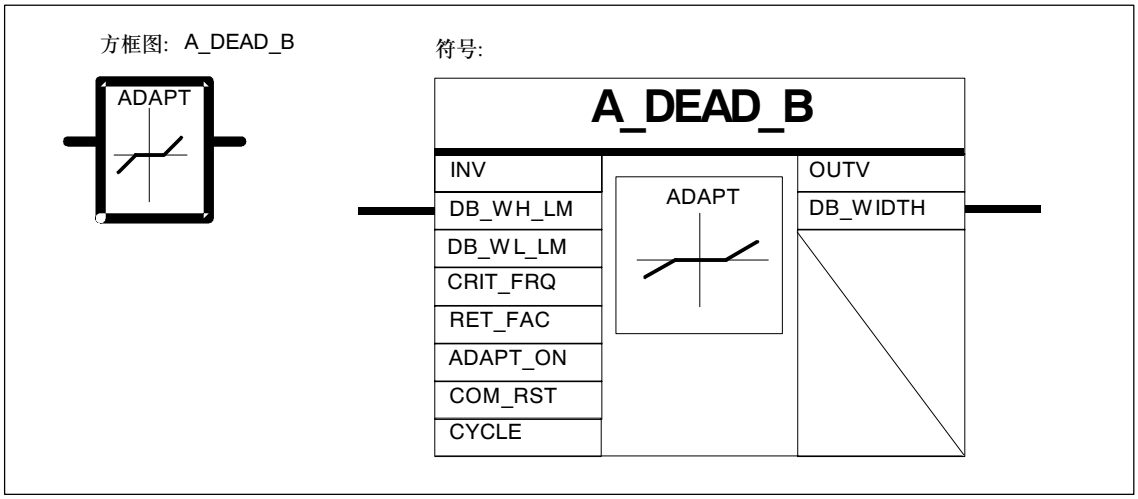


图2-1 A_DEAD_B，方框图和符号

功能描述

该块过滤错误信号的高频干扰信号输出。它在零点周围形成了一个死区。如果输入变量在该死区范围之内，则输出应用零值。系统会根据噪声信号的振幅，自适应调整死区的宽度。

该功能块根据下列函数运行:

OUTV = INV + DB_WIDTH when INV < -DB_WIDTH

OUTV = 0.0 when -DB_WIDTH ≤ INV ≤ +DB_WIDTH

OUTV = INV - DB_WIDTH when INV > +DB_WIDTH

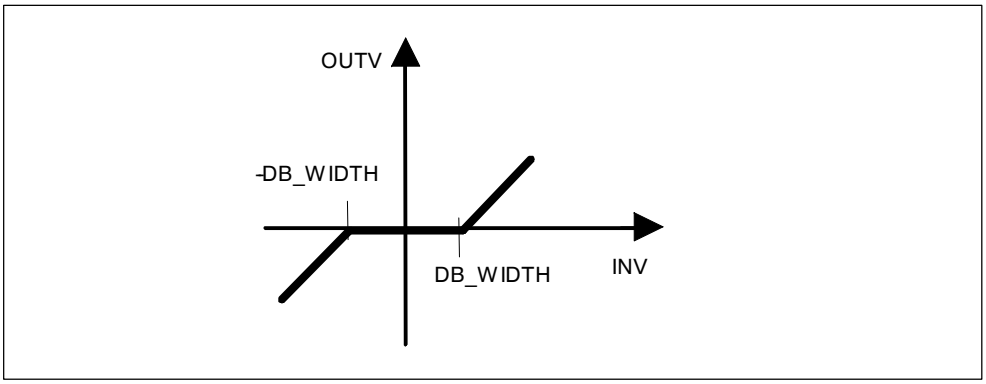


图2-2 OUTV = f(INV)

死区的自适应修改

为了保证控制器的稳定性，有效死区宽度DB_WIDTH向下由可选择的输入参数DB_WL_LM所限制。如果受噪声影响的输入信号INV在负(1)、正(2)方向上超过了当前设置的死区宽度，此后在时间段1/CRIT_FRQ内，再次在负(3)方向上超出当前死区宽度，则有效死区的数值将增加0.1。(也可以参见图2-4)。无论是在正方向上超过死区，还是在负方向上超过死区，都会启动此过程。无论何时，只要随后在相反方向上，在一半时间段内就超过了死区(3 -> 4)，则死区的数值会再次增加0.1。重复进行此过程，直到死区宽度与测量噪声的振幅相匹配。为了防止抑制小输入信号，有效死区宽度向上由输入DB_WH_LM限制。另一方面，如果在时间RET_FAC*1/CRIT_FRQ内未超过死区宽度，则死区的数值减小0.1。

CRIT_FRQ指定关键频率，系统将该频率处的信号分量检测为噪声。它的上限和下限如下:

0.01 CRIT_FRQ 1/(3*CYCLE)，其中CYCLE是以秒为单位的采样时间。

RET_FAC参数指定1/CRIT_FRQ的倍数，在此值之后将再次减小死区宽度。

只有在无噪声分量的输入变量逼近零时才会运行自适应逻辑。

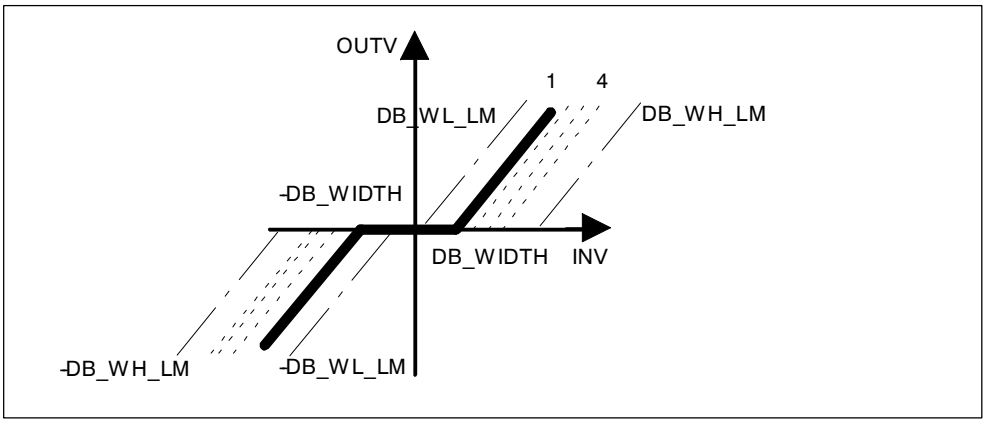


图2-3 死区自适应

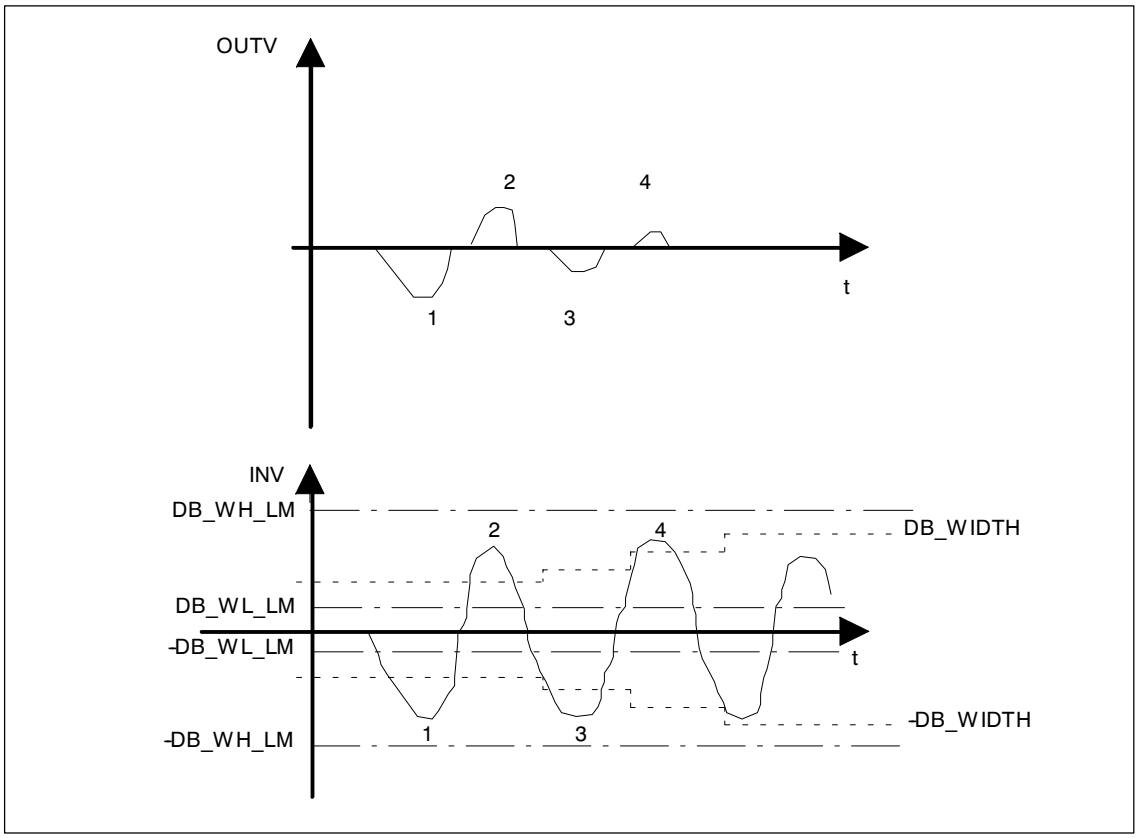


图2-4 死区自适应

输入参数

下表给出了A_DEAD_B的输入参数的数据类型和结构。

表2-1 A_DEAD_B的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	DB_WH_LM	死区宽度上限	技术范围 > DB_WL_LM	5.0
REAL	DB_WL_LM	死区宽度下限	技术范围 < DB_WH_LM	1.0
REAL	CRIT_FRQ	关键频率	≥ 0.01和 ≤1/(3 * CYCLE)	0.1
INT	RET_FAC	返回因子	≥ 1	1
BOOL	ADAPT_ON	自适应算法打开		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s

输出参数

下表给出了A_DEAD_B的输出参数的数据类型和结构。

表2-2 A_DEAD_B的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0
REAL	DB_WIDTH	有效死区宽度	0.0

完全重启动

在完全重启动期间，OUTV被设置为0.0，同时还会设置有效死区宽度，使得DB_WIDTH = DB_WL_LM。

正常操作

下列条件适用于自适应:

- 自适应关闭

  如果关闭了自适应(ADAPT_ON = FALSE，则会冻结最后的DB_WIDTH值，将其用作有效死区宽度DB_WIDTH。

- 自适应打开

  如果ADAPT_ON = TRUE，便可以将自适应算法包含到控制器计算中，该算法计算有效死区宽度。它会调整死区宽度，使其与输入变量上叠加的噪声信号相匹配；这样即使噪声分量的振幅发生波动，也可以有效抑制该噪声分量。

  如果块调用是周期性循环的，则必须关闭自适应(ADAPT_ON = FALSE)。

块内部限制

在此块中，输入参数的数值并没有限制；系统不会检查参数。

实例

如果在启动期间由于存在噪声而打开了自适应，而一段时间之后系统成功建立起稳定的死区宽度，则可以关闭自适应。在完全重启动之后，会一直保持自适应功能设置的死区宽度。

2.1.2 CRP_IN: 改变外围设备输入范围

应用

块调整模拟I/O的数值范围，使其与模块化控制器的内部表达式相符；例如，可以在过程变量分支中调用它。

方框图

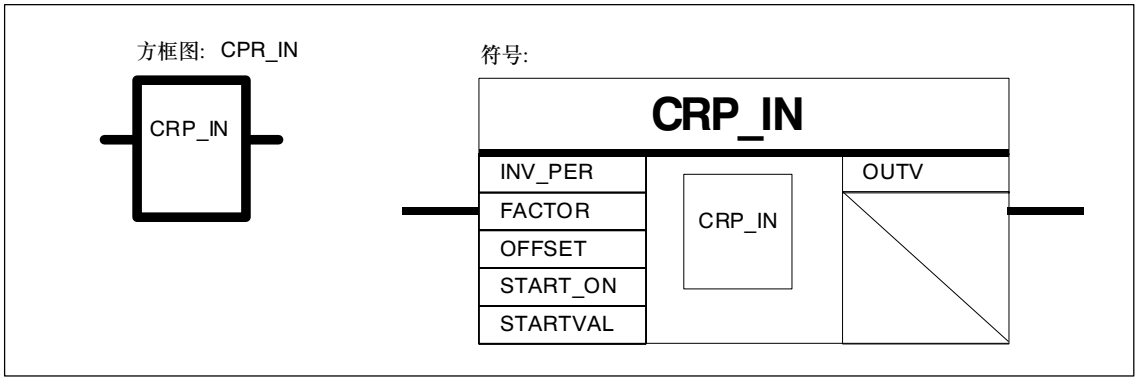


图2-5 CRP_IN，方框图和符号

功能描述

CRP_IN将外围设备格式的输入值转换成标准化浮点值，供模块化控制器使用。

外围设备值	输出值，%格式
32767	118.515
27648	100.000
1	0.003617
0	0.000
-1	-0.003617
-27648	-100.000
-32768	-118.519

可以使用线性标定因子和偏移量来调整浮点值。通过下式获得输出:

OUTV = INV_PER * 100/27648 * FACTOR + OFFSET

在安装、测试期间，或者在外围设备中发生故障时，可以将输出更改为启动值。如果设置了**START_ON** = TRUE，则**STARTVAL**中的数值输出到**OUTV**输出端子处。

注意

此处并不检查正/负溢出。

输入参数

下表给出了CRP_IN的输入参数的数据类型和结构。

表2-3 CRP_IN的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
WORD	INV_PER	输入变量外围设备	技术取值范围	0
REAL	FACTOR	线性标定因子		1.0
REAL	OFFSET	偏移量	技术取值范围	0.0
BOOL	START_ON	启动值打开		TRUE
REAL	STARTVAL	启动值	技术取值范围	0.0

输入参数

下表给出了CRP_IN的输出参数的数据类型和结构。

表2-4 CRP_IN的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

完全重启动

块并没有完全重启动例行程序。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

在此块中，输入参数的数值并没有限制；系统不会检查参数。

2.1.3 CRP_OUT: 改变外围设备输出范围

应用

该功能块调整模块化控制器的浮点值，使其与外围设备格式相符。

方框图

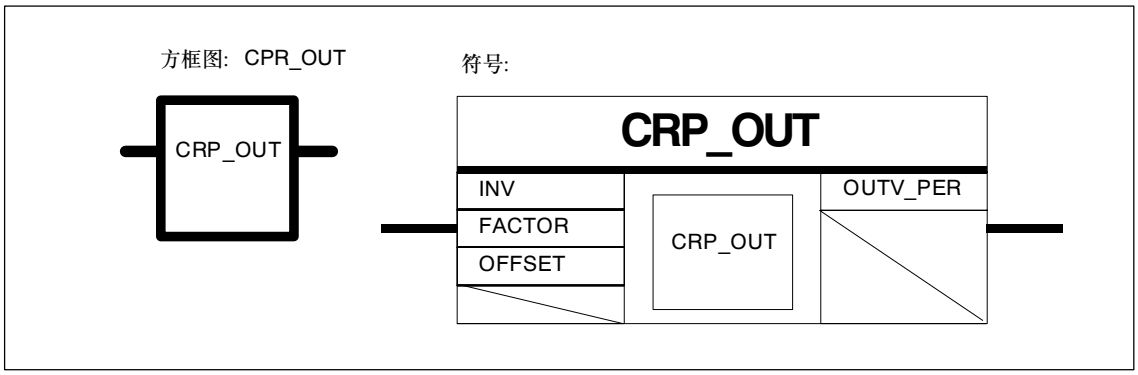


图2-6 CRP_OUT，方框图和符号

功能描述

CRP_OUT将输入数值(模块化控制器的标准化浮点值)转换成模拟量I/O的外围设备格式。

表2-5 输入值/外围设备值

输入值，%格式	外围设备值
118.515	32767
100.000	27648
0.003617	1
0.000	0
-0.003617	-1
-100.000	-27648
-118.519	-32768

可以使用线性标定因子和偏移量来调整浮点值。输出值的计算如下:

OUTV_PER = (INV * FACTOR + OFFSET) * 27648/100

注意

此处并不检查正/负溢出。

输入参数

下表给出了CRP_OUT的输入参数的数据类型和结构。

表2-6 CRP_OUT的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	FACTOR	线性标定因子		1.0
REAL	OFFSET	偏移量	技术取值范围	0.0

输出参数

下表给出了CRP_OUT的输出参数的数据类型和结构。

表2-7 CRP_OUT的输出参数

数据类型	参数	注释	缺省值
WORD	OUTV_PER	输出变量外围设备	0

完全重启动

块并没有完全重启动例行程序。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

在此块中，输入参数的数值并没有限制；系统不会检查参数。

2.1.4 DEAD_T：时滞

应用

该块可以用在比率控制器中；例如在将比率控制器的各个组件放在一起，各个组件有不同的距离要移动时便可以使用此块。

方框图

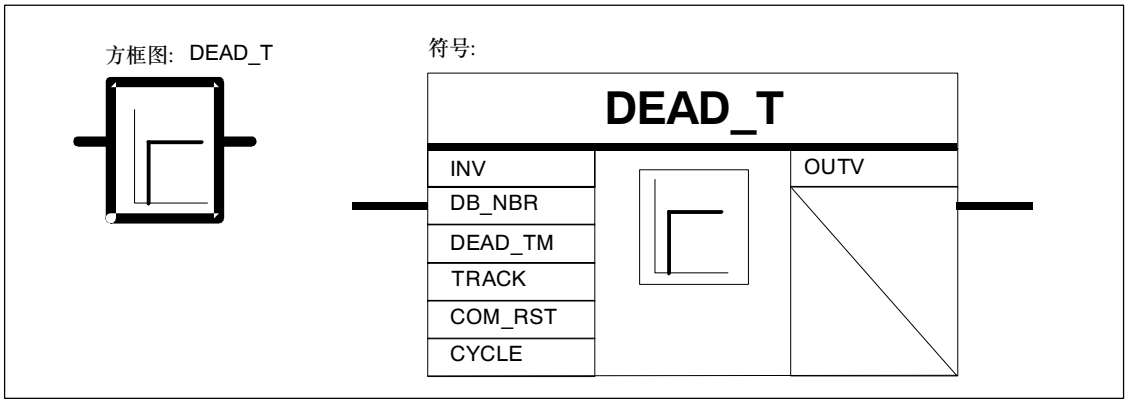


图2-7 DEAD_T，方框图和符号

功能描述

块将输入值的输出延迟一个可以选择的时间(时滞)。输入值缓存在共享数据块中。最大时滞取决于该数据块的长度。共享数据块DB_NBR中的数据的处理方式与环形缓冲区中的数据处理方式相同。

表2-8 输入值

编号	输入值	
0	INV[0]	⊓
1	INV[1]	| ↓
2	INV[2]	| ⇔ **OUTV/INV** 读/写指针
...	...	| ↓
...	...	| |
n	INV[n]	⊔ DEAD_TM = (n+1) • CYCLE
...	...	
m	INV[m]	

将读/写指针指示的位置读入，以及输出到OUTV中。在此之后，将INV写入到同一存储器位置。每次执行块时，读/写指针的存储器位置索引增加1。当达到n时，返回到0。

如果指定了时滞DEAD_TM，并有固定采样时间CYCLE时，数据块必须允许

$$\frac{\text{DEAD\_TM}}{\text{CYCLE}}$$

保存操作。保存操作(数据类型：REAL)占用4个字节。DEAD_TM必须是CYCLE的整数倍。

$$\text{DB长度 (以字节为单位)u} = \frac{\text{DEAD\_TM}}{\text{CYCLE}} * 4$$

如果TRACK = TRUE，将直接输出输入值。

注意

块不会检查具有编号DB_NBR的共享DB是否已经存在，以及参数DEAD_TM (时滞)和CYCLE(采样时间)是否与数据块的长度相匹配。如果参数分配不正确，则CPU切换到STOP，同时报告内部系统错误。

输入参数

下表给出了DEAD_T的输入参数的数据类型和结构。

表2-9　DEAD_T的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
BLOCK_DB	DB_NBR	数据块编号		DB 1
TIME	DEAD_TM	时滞	≥ CYCLE ≤ DB长度/4*CYCLE	10秒
BOOL	TRACK	跟踪OUTV = INV		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	1秒

输出参数

下表给出了DEAD_T的输出参数的数据类型和结构。

表2-10 DEAD_T的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

共享数据块DB_NBR

下表给出了共享数据块的数据类型和参数。

表2-11 共享数据块的参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV[0]	输入变量 [0]	技术取值范围	0.0
REAL	INV[1]	输入变量 [1]	技术取值范围	0.0
REAL	INV[2]	输入变量 [2]	技术取值范围	0.0
REAL	INV[3]	输入变量 [3]	技术取值范围	0.0

完全重启动

在完全重启动期间，删除保存的所有输入值，并且输出OUTV = 0.0。

正常操作

输入值的输出被延迟由“时滞”定义的时间。在线改变时滞设置可能导致输出值中发生阶跃变化。

- 跟踪

  如果打开了跟踪(TRACK = TRUE)，则输入值将会无任何延迟地传送到OUTV。输入值的缓冲不会被中断，这样在关闭跟踪的情况下，在设置的时滞时间过后，仍然可以输出输入值。如果TRACK = FALSE，OUTV跳转到INV[DEAD_TM]。

块内部限制

在此块中，输入参数的数值并没有限制； 系统不会检查参数。

实例

在采样时间为CYCLE = 1 s，时滞DEAD_TM = 4 s的情况下，必须缓存四个输入值。这样，数据区长度必须为16字节。

表2-12　双字/输入值

数据双字	输入值
0	INV[0]
4	INV[1]
8	INV[2]
12	INV[3]

2.1.5 DEADBAND：死区

应用

如果过程变量受噪声影响，而且控制器实现最优设置，则噪声也将影响控制器输出。由于存在很高的切换频率(步进控制器)，这会增加执行的磨损。抑制噪声可以防止控制器输出振荡。在使用死区来形成误差信号时，必须将偏移量DEADB_O设置成0.0。

方框图

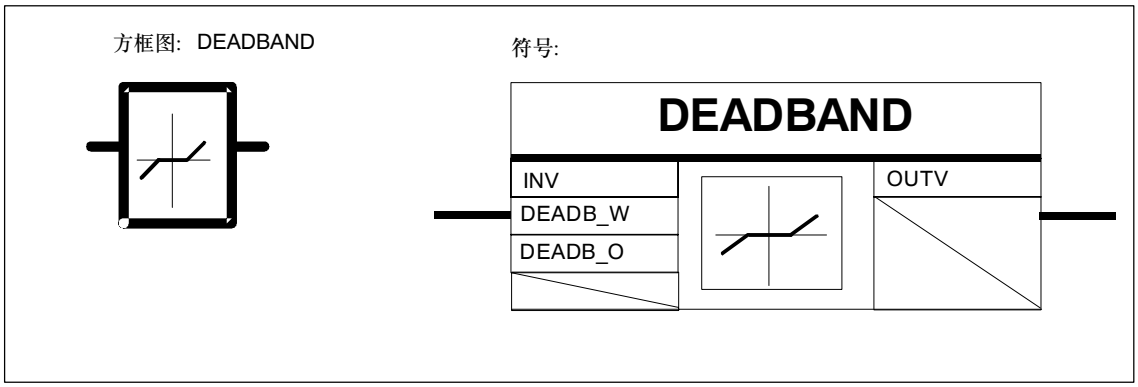


图2-8 DEADBAND，方框图和符号

功能描述

DEADBAND块在指定的零点附近，抑制输入变量INV中的小幅波动。在此死区之外，输出变量OUTV随输入变量线性增加。该功能块根据下列函数运行:

OUTV = INV + DEADB_W - DEADB_O
when INV < DEADB_O - DEADB_W

OUTV = 0.0 when DEADB_O - DEADB_W ≤ INV
and INV ≤ DEADB_O + DEADB_W

OUTV = INV - DEADB_W - DEADB_O
when INV > DEADB_O + DEADB_W

信号失真，失真量的大小为DEADB_W的值。死区的中间点由DEADB_O指定。

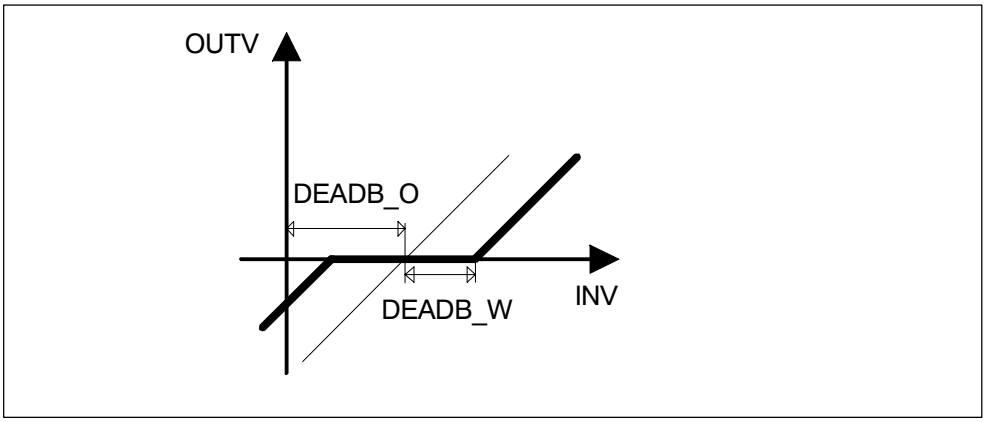


图2-9 OUTV = f(INV)

用户可以自行选择死区宽度DEADB_W和死区偏移量DEADB_O。

输入参数

下表给出了DEADBAND的输入参数的数据类型和结构。

表2-13 DEADBAND的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	DEADB_W	死区宽度	技术范围 ≥ 0.0	1.0
REAL	DEADB_O	死区偏移量	技术取值范围	0.0

输出参数

下表给出了DEADBAND的输出参数的数据类型和结构。

表2-14 DEADBAND的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

完全重启动

块没有完全重启动例行程序。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

在此块中，输入参数的数值并没有限制；系统不会检查参数。死区宽度只能为正值。

实例

图2-10说明了使用偏移量进行噪声抑制的情况。

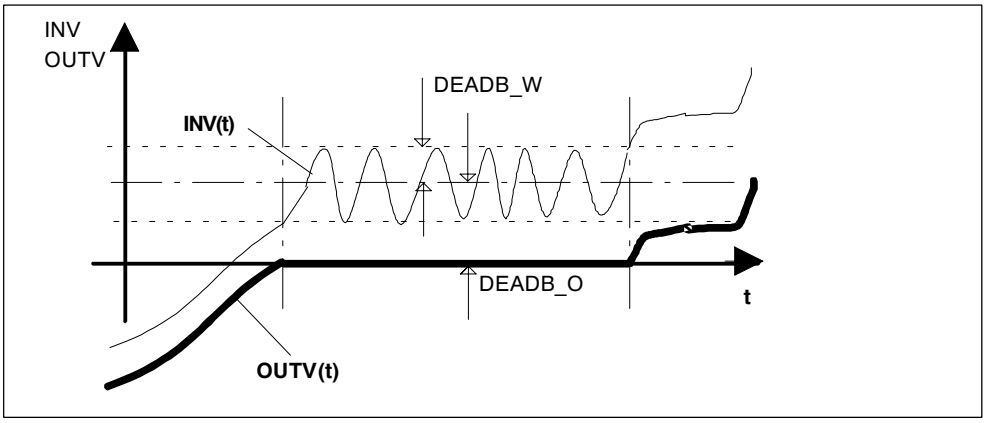


图2-10　　使用偏移量进行噪声抑制

2.1.6 DIF：微分器

应用

动态微分过程变量。也就是说，例如，可以通过移动的距离来计算速度。微分器可用于前馈控制，作为预控制器以及用于配置一个控制器。

方框图

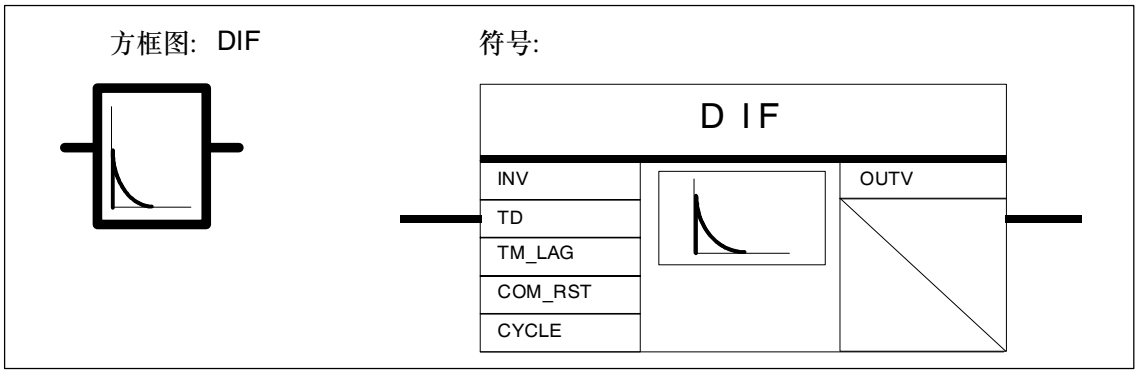


图2-11 DIF，方框图和符号

功能描述

该块将输入值对时间微分，使用一个一阶时滞环节对信号进行滤波。

输入参数

下表给出了DIF的输入参数的数据类型和结构。

表2-15 DIF的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
TIME	TD	微分时间值	≥ CYCLE	T#25s
TIME	TM_LAG	时间延迟		T#5s
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s

输出参数

下表给出了DIF的输出参数的数据类型和结构。

表2-16　DIF的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

完全重启动

在完全重启动期间，所有的信号输出都被设置成0。在内部，为微分器分配当前输入值**INV**。这样，如果输入变量保持不变，切换到常规运行就不会导致任何阶跃变化。

正常操作

在微分期间，块根据下列传送功能运行:

在Laplace范围内: OUTV(s) / INV(s) = TD / (1+TM_LAG*s)

微分器的时间响应由微分时间TD和时间延迟TM_LAG指定。下图说明了相应的阶跃响应。

步响应

图2-12和2-13说明了DIF (带和不带时滞)的阶跃响应。

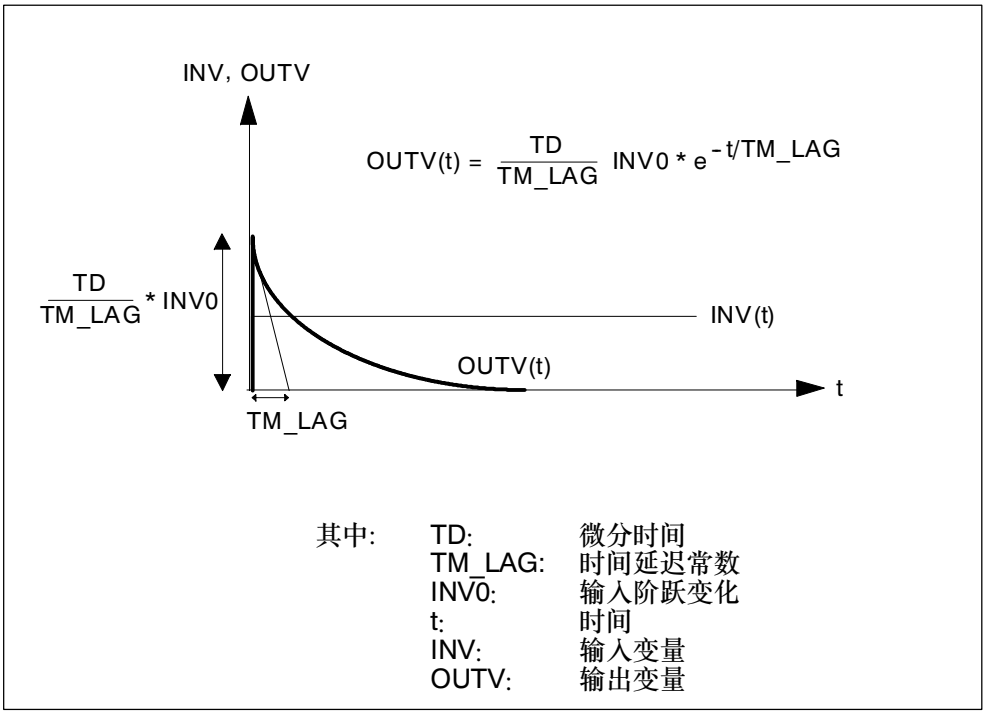


图2-12 DIF的阶跃响应

如果分配给TM_LAG的值小于或者等于CYCLE/2，则微分器的运行无时间延迟。通过因子TD/CYCLE，将输入阶跃变化施加到输出。一个周期之后，输出重新返回到0.0。

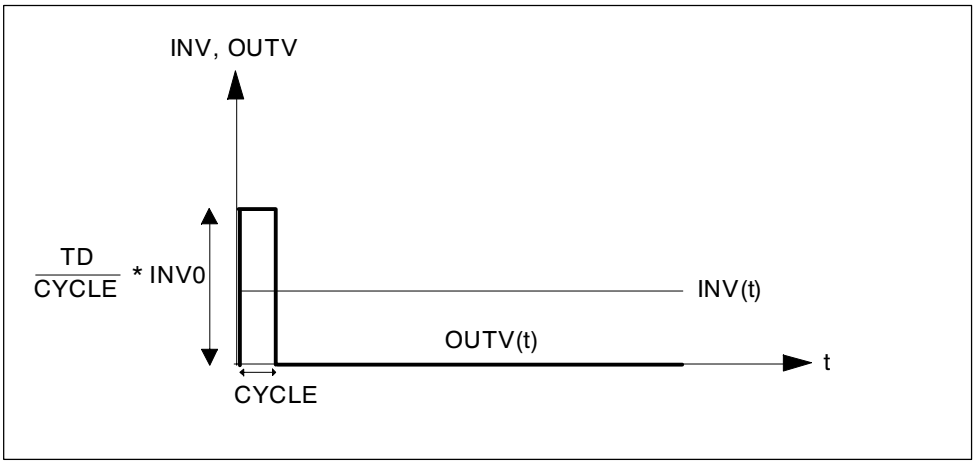


图2-13 无延迟的DIF的阶跃响应

块内部限制

微分时间向下限制为采样时间。时间延迟向下限制为采样时间的一半。

TD_{intern} = CYCLE　　when TD < CYCLE

TM_LAG_{intern} = CYCLE/2　　when TM_LAG < CYCLE/2

在此块中，其它输入参数的数值并没有限制；系统不会检查参数。

2.1.7 ERR_MON: 偏差信号监视

应用

块用于形成和监视偏差信号。

方框图

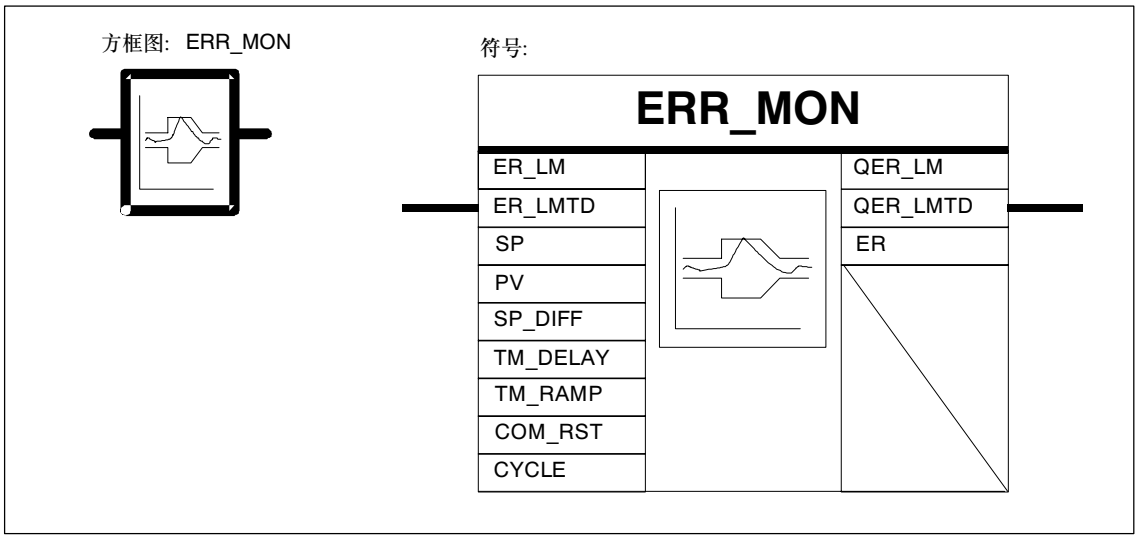
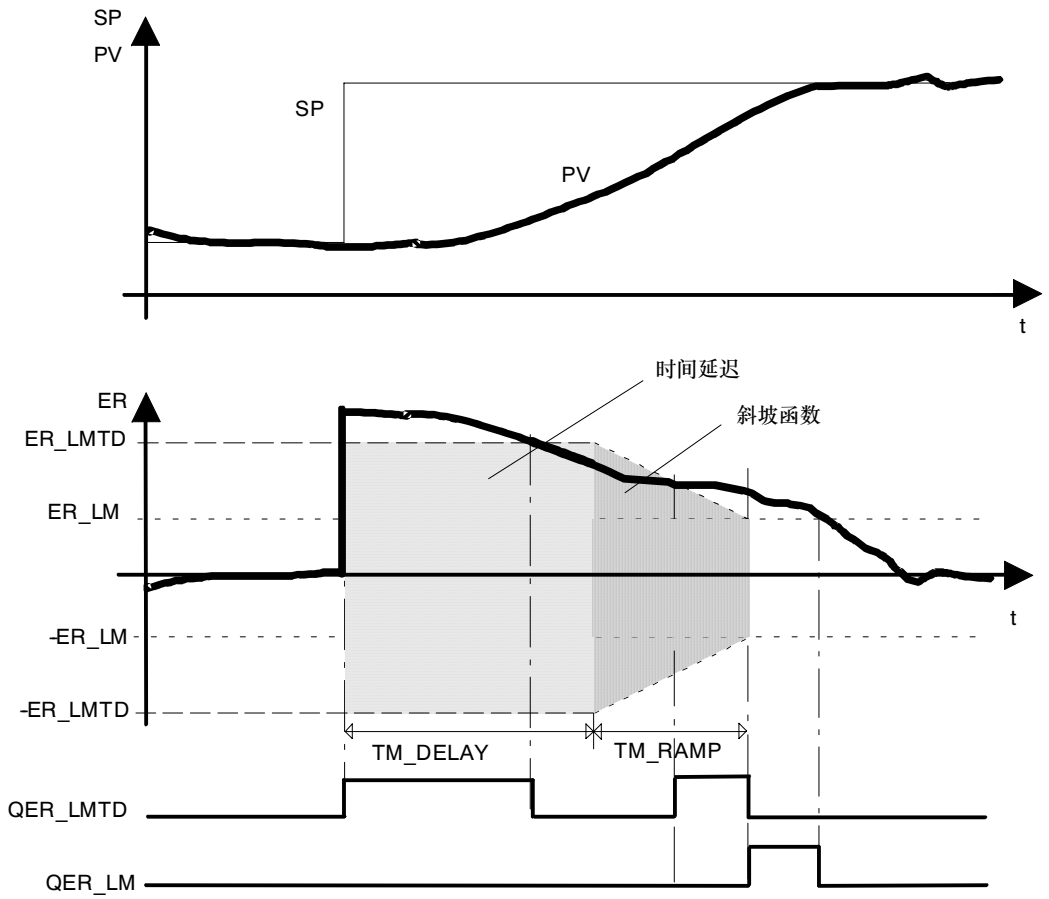


图2-14 ERR_MON，方框图和符号

功能描述

块计算偏差信号ER = SP - PV，并监视它是否在可选择的限制内。如果设定值中的变化大于SP_DIF，就会抑制限制值信号ER_LM的激活，抑制的时间长度可选择(TM_DELAY+TM_RAMP)；在此时间段内，会监视ER的高限制值ER_LMTD。如果超过了ER_LMTD，则会输出QER_LMTD = TRUE。一旦超出了延迟时间，ER_LMTD就会根据斜坡函数变为ER_LM。通过设定值变化来启动接通延迟。可以通过TM_RAMP参数选择斜坡的范围。



PV:	过程变量	TM_DELAY:	监视信号的时间延迟
SP:	设定值	TM_RAMP:	斜坡时间常数
ER:	出错信号	QER_LM:	已达到误差信号限制
t:	时间	QER_LMTD:	在时间延迟期间的误差信号限制并且已到达斜坡
ER_LM:	误差信号限制		
ER_LMTD:	时间延迟期间的误差信号限制		

图2-15 ERR_MON如何工作

输入参数

下表给出了ERR_MON的输入参数的数据类型和结构。

表2-17 ERR_MON的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	ER_LM	误差变量限制	技术范围 > 0.0，并且 < ER_LMTD	10.0
REAL	ER_LMTD	在时间延迟期间的误差信号限制	技术范围 > ER_LM	100.0
REAL	SP	设定值变量	技术取值范围	0.0
REAL	PV	过程变量	技术取值范围	0.0
REAL	SP_DIFF	设定值差值	技术范围 >0.0	10.0
TIME	TM_DELAY	监视信号的时间延迟		T#60s
TIME	TM_RAMP	斜坡的时间常数		T#60s
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s

输出参数

下表给出了ERR_MON的输出参数的数据类型和结构。

表2-18 ERR_MON的输出参数

数据类型	参数	注释	缺省值
BOOL	QER_LM	已达到误差信号限制	FALSE
BOOL	QER_LMTD	在时间延迟期间已到达误差信号限制	FALSE
REAL	ER	出错信号	0.0

完全重启动

在完全重启动期间，会将QER_LM和QER_LMTD信号以及误差信号输出ER全部复位。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

在此块中，输入参数的数值并没有限制； 系统不会检查参数。

2.1.8 INTEG: 积分器

应用

动态积分过程变量。也就是说，例如，可以通过速度计算出移动的距离。积分器可以用于配置控制器。

方框图

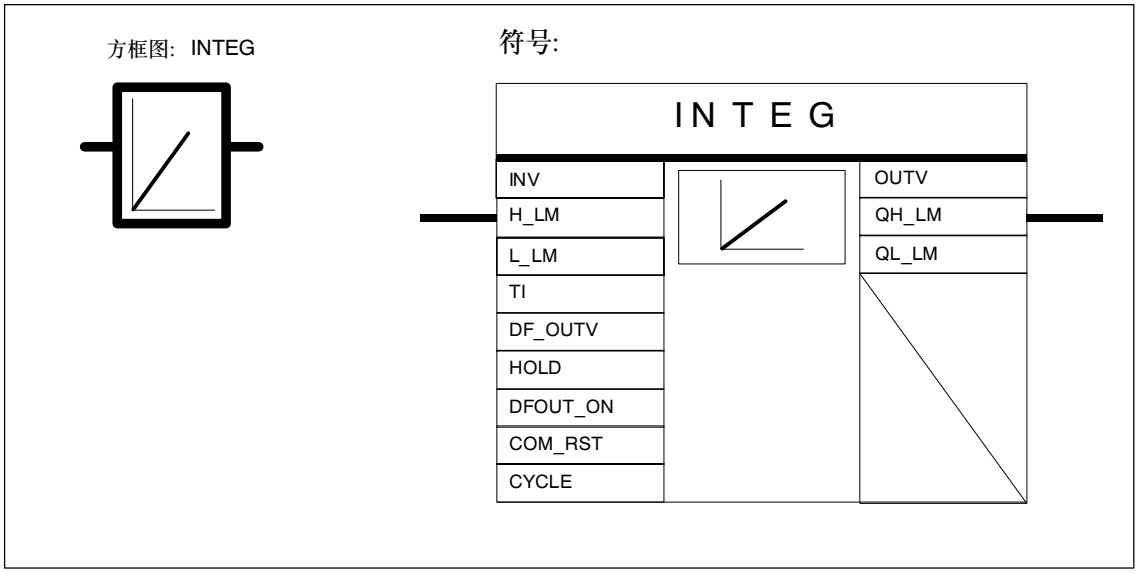


图2-16 INTEG，方框图和符号

功能描述

块计算输入变量在时间轴上的积分，并将积分值限制在上下限之内。输出变量的限制通过信号位来指示。

输入参数

下表给出了INTEG的输入参数的数据类型和结构。

表2-19 INTEG的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	H_LM	上限	技术范围 > L_LM	100.0
REAL	L_LM	下限	技术范围 < H_LM	0.0
TIME	TI	积分作用暂停	≥ CYCLE	T#25s
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
BOOL	HOLD	积分器暂停		FALSE
BOOL	DFOUT_ON	缺省输出变量打开		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s

输出参数

下表给出了INTEG的输出参数的数据类型和结构。

表2-20 INTEG的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0
BOOL	QH_LM	已达到上限	FALSE
BOOL	QL_LM	已达到下限	FALSE

完全重启动

在完全重启动期间，OUTV输出复位为0.0。如果设置了DFOUT_ON = TRUE，则输出DF_OUTV。在完全重启动期间，输出的限制保持有效，并且限制信号位也有效。当控制器切换到正常运行状态时，块从OUTV处开始积分。

如果希望在执行完全重启动时，让积分器在特定的运行点处启动，则必须在输入DF_OUTV处输入运行点。当在完全重启动例行程序中调用块时，则必须在周期性中断优先级中首先设置DFOUT_ON = TRUE，然后再复位成DFOUT_ON = FALSE。

正常操作

除正常操作以外，块还有下列一些模块:

模式	DFOUT_ON	HOLD
积分	FALSE	FALSE
积分器暂停	FALSE	TRUE
缺省输出变量	TRUE	ANY

- **积分**

在积分期间，块根据下列传送功能运行:

在Laplace范围内:
OUTV(s) / INV(s) = 1 / (TI∗s)

积分器的时间响应由复位时间TI指定。下图说明了相应的阶跃响应。

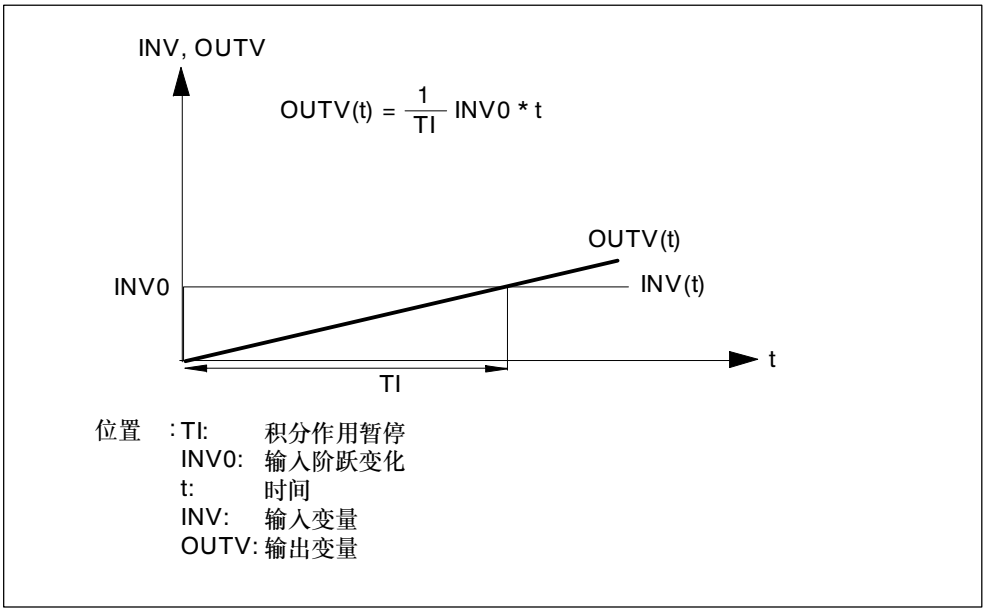


图2-17 INTEG的阶跃响应

输出和积分器缓冲限制为可选的限制值H_LM和L_LM。如果输出在此限制范围之内，通过信号位QH_LM和QL_LM对此进行指示。

- **积分器暂停**

如果HOLD设置为TRUE，则积分器保持在其当前输出值OUTV处。当HOLD复位为FALSE时，积分器将继续从当前输出值OUTV处进行积分运算。

- **输出处的缺省值**

如果设置了DFOUT_ON = TRUE，则DF_OUTV将应用到输出处。限制有效。如果复位了此参数，即设置了DF_OUTV_ON = FALSE，则积分器开始从数值DF_OUTV处启动积分运算。

块内部限制

复位时间向下由采样时间限制:

TI_{intern} = CYCLE when TI < CYCLE

在此块中，其它输入参数的数值并没有限制；系统不会检查参数。

实例

图2-18给出了DFOUT_ON、HOLD和限制的实例。

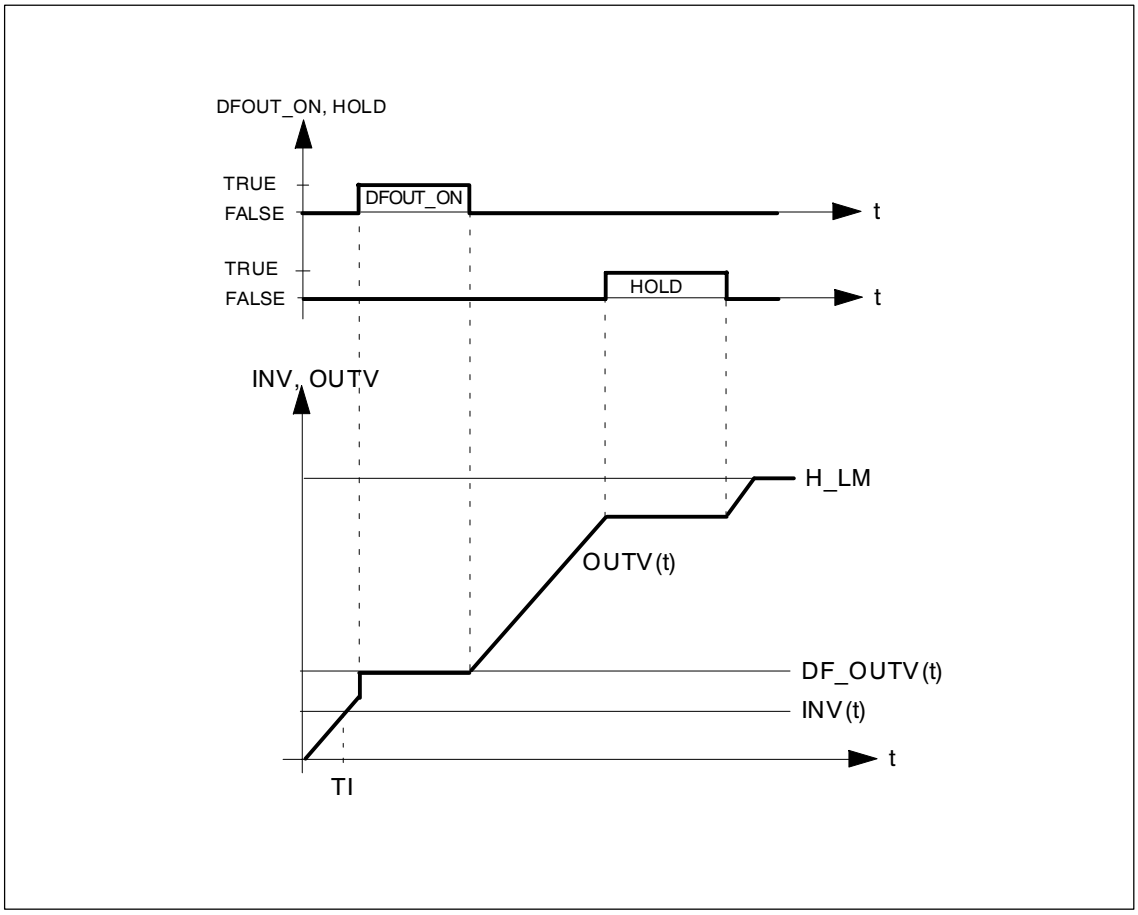


图2-18 DFOUT_ON、HOLD和限制的实例

2.1.9 LAG1ST: 一阶延迟元件

应用

该块可以用作延迟或滤波元件。

方框图

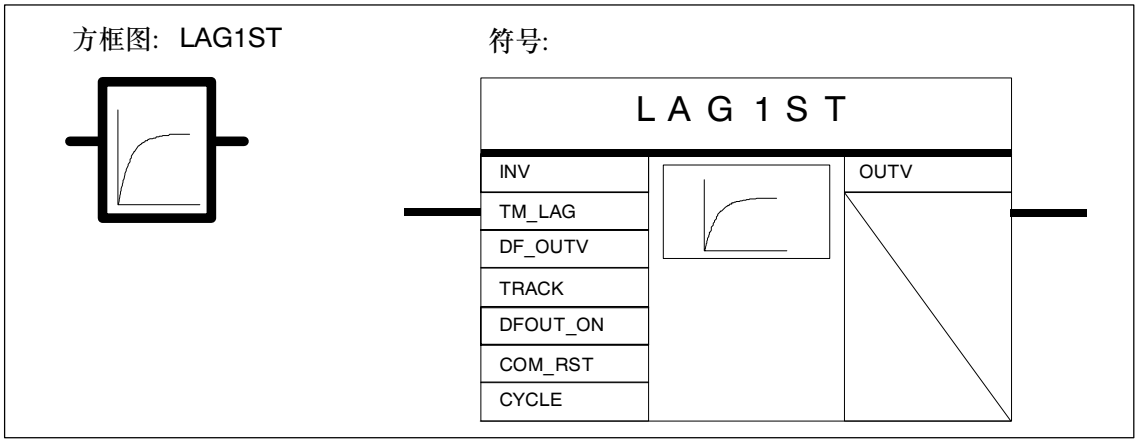


图2-19 LAG1ST，方框图和符号

功能描述

该块使用一阶时间延迟环节来滤波输入变量。时间延迟可以选择。

输入参数

下表给出了LAG1ST的输入参数的数据类型和结构。

表2-21 LAG1ST的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
TIME	TM_LAG	时间延迟		T#25s
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
BOOL	TRACK	跟踪OUTV = INV		FALSE
BOOL	DFOUT_ON	缺省输出变量打开		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s

输出参数

下表给出了LAG1ST的输出参数的数据类型和结构。

表2-22 LAG1ST的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

完全重启动

在完全重启动期间，输出OUTV复位为0.0。如果设置了DFOUT_ON = TRUE，则输出DF_OUTV。当控制器切换回到正常运行模式时，块继续从OUTV处开始运行。

正常操作

除了正常运行模式，该块还具有下列模式:

模式	DFOUT_ON	HOLD
过滤	FALSE	FALSE
跟踪	FALSE	TRUE
缺省输出变量	TRUE	ANY

- **过滤**

在进行滤波时，块根据下列传送功能运行:

在Laplace范围内: OUTV(s) / INV(s) = 1 / (1+TM_LAG∗s)

延迟元件的时间响应由时间延迟TM_LAG指定。下图说明了相应的阶跃响应。

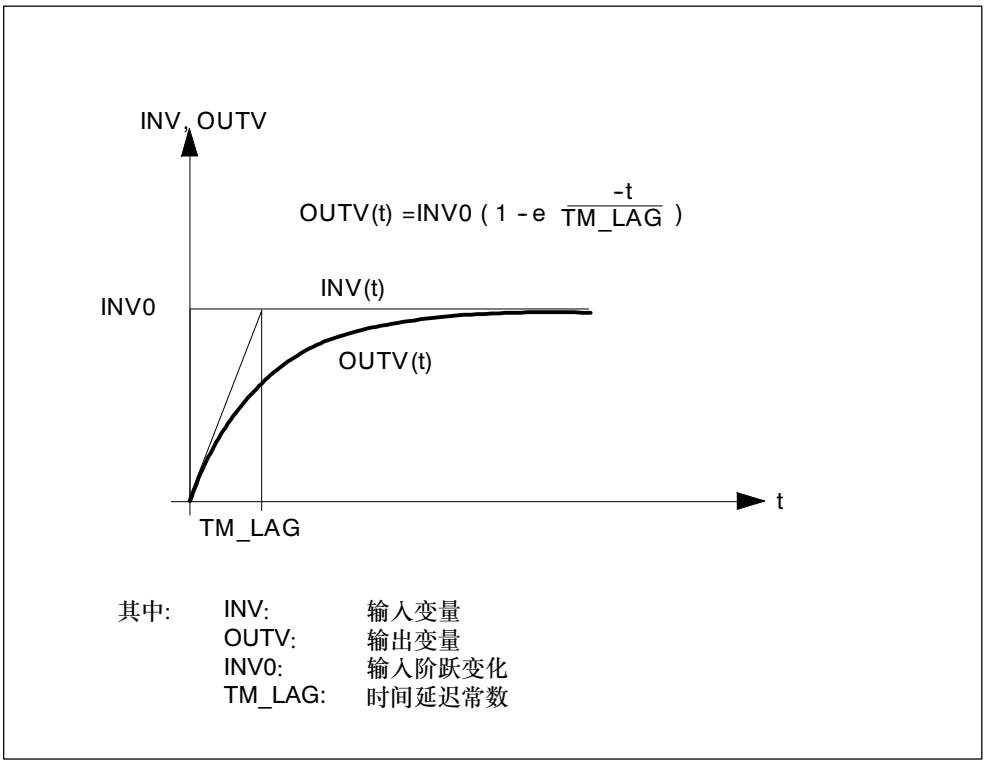


图2-20 LAG1ST的阶跃响应

- **跟踪**

如果TRACK = TRUE，输入值INV将切换到输出OUTV。

- **输出处的缺省值**

如果设置了DFOUT_ON = TRUE，则将DF_OUTV应用到输出处。如果复位了此参数，即设置了DF_OUTV_ON = FALSE，则延迟元件将从数值DF_OUTV处开始滤波。

块内部限制

时间延迟向下限制为采样时间的一半。

TM_LAG_{intern} = CYCLE/2　　When TM_LAG < CYCLE/2

在此块中，其它输入参数的数值并没有限制；系统不会检查参数。

实例

图2-21给出了使用DFOUT_ON和TRACK的实例。

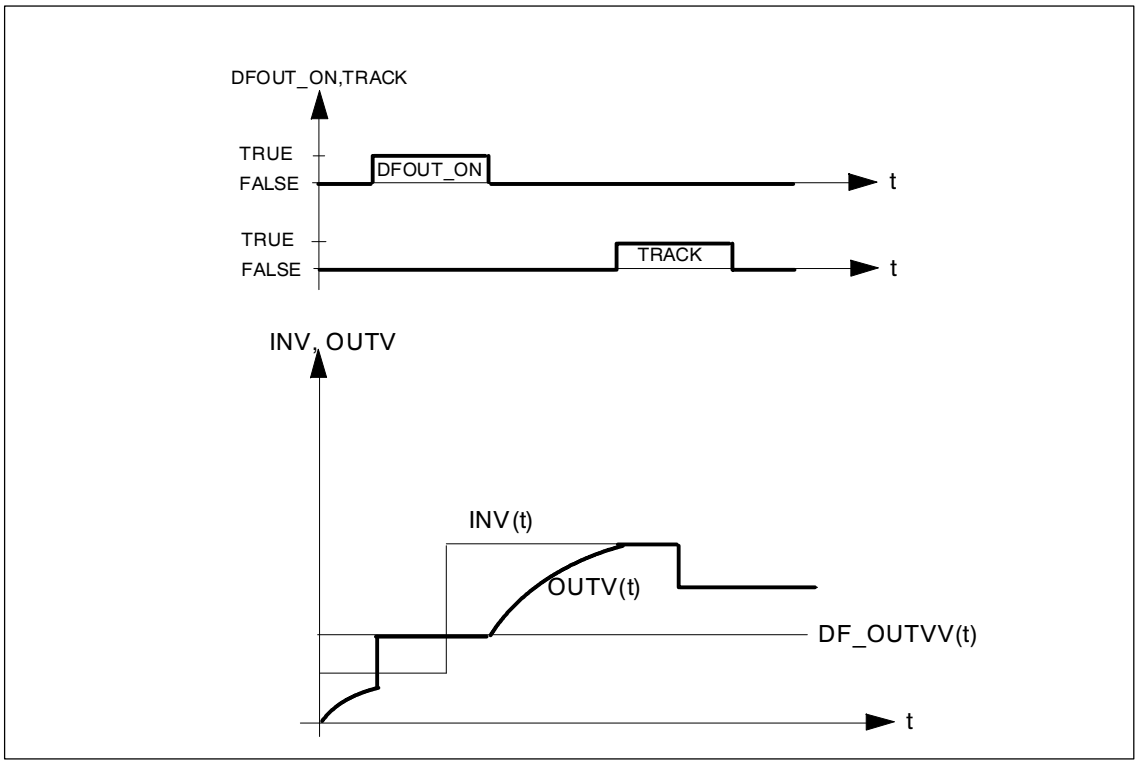


图2-21　使用DFOUT_ON和TRACK的实例

2.1.10 LAG2ND: 二阶延迟元件

应用

该块用于模拟预控制器和双回路控制器的系统组件。

方框图

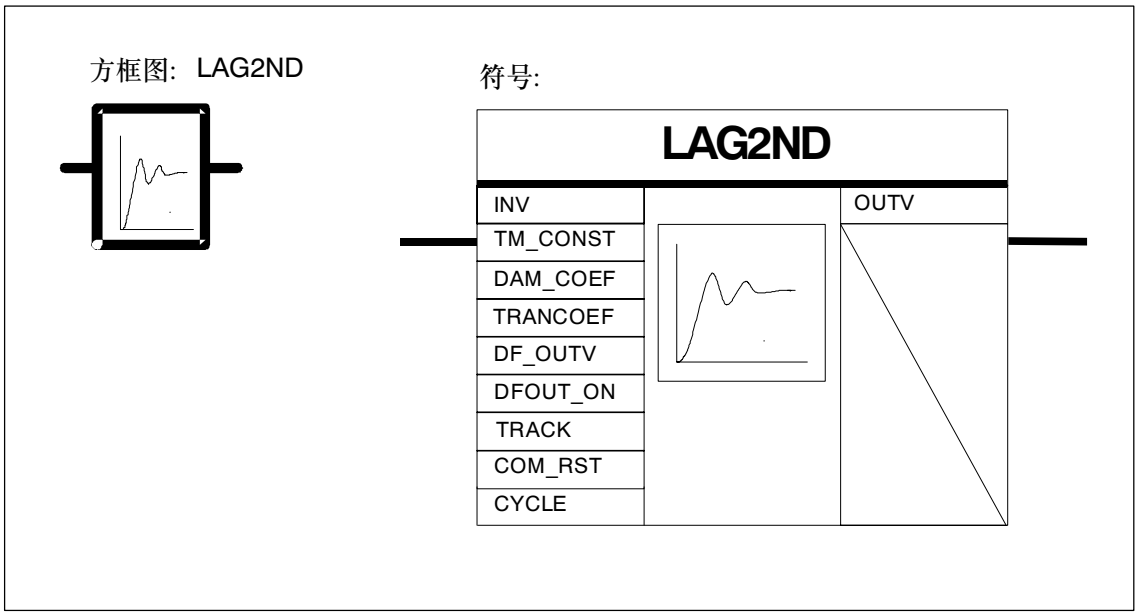


图2-22 LAG2ND，方框图和符号

功能描述

该块实现了一个二阶延迟元件，该元件具有振荡能力。

传送函数

Laplace范围内的传送函数是:

$$OUTV(s) / INV(s) = TRANCOEF / (1 + 2*DAM\_COEF*TM\_CONST*s + TM\_CONST^2*s^2)$$

如果DAM_COEF >= 1 (非周期情况)，传送元件可以表示为一系列带有两PT1元件的电路。

$$OUTV(s) / INV(s) = TRANCOEF / (1 + T1*s) * 1 / (1 + T2 * s)$$

按照下列方式重新计算时间常数:

$$T1 = TM\_CONST\ (DAM\_COEF + \sqrt{DAM\_COEF^2 - 1}\ )$$

$$T2 = TM\_CONST\ (DAM\_COEF - \sqrt{DAM\_COEF^2 - 1}\ )$$

方框图

图2-23给出了LAG2ND的方框图。

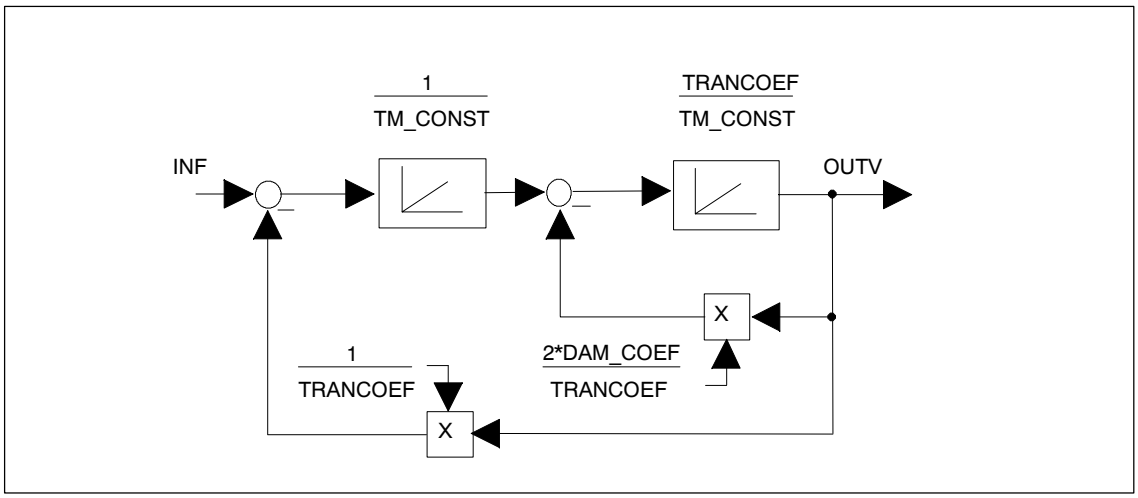


图2-23 LAG2ND的结构，由基本传送元件组成

步响应

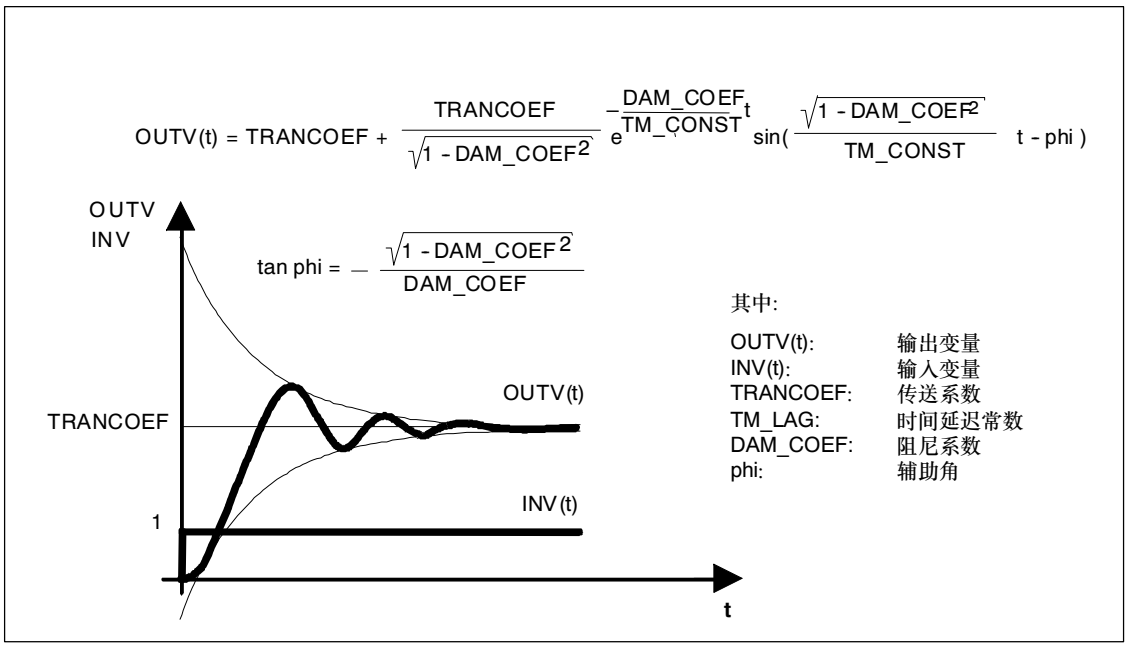


图2-24 具有振荡的LAG2ND元件的阶跃响应

输入参数

下表给出了LAG2ND的输入参数的数据类型和结构。

表2-23 LAG2ND的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
TIME	TM_CONST	时间常数		T#10s
REAL	DAM_COEF	阻尼系数		1.0
REAL	TRANCOEF	传送系数		1.0
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
BOOL	DFOUT_ON	缺省输出变量打开		FALSE
BOOL	TRACK	跟踪OUTV = INV		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s

输出参数

下表给出了LAG2ND的输出参数的数据类型和结构。

表2-24　LAG2ND的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

完全重启动

在完全重启动期间，输出OUTV设置为0.0。如果DFOUT_ON = TRUE，则会输出DF_OUTV。

正常操作

除了正常运行模式，该块还具有下列模式:

- 跟踪

如果设置了TRACK = TRUE，则OUTV = INV；内部历史值设置为INV。

- 输出处的缺省值

如果设置了DFOUT_ON = TRUE，则输出DF_OUTV，内部历史值设置为DF_OUTV。DFOUT_ON的优先级比TRACK高。

块内部限制

时间常数TM_CONST向下限制为采样时间的一半。

TM_CONST_{intern} = CYCLE/2　when TM_CONST < CYCLE/2

在此块中，其它输入参数的数值并没有限制；系统不会检查参数。

2.1.11 LIMALARM: 限制报警

应用

如果过程值(例如: 电机速度、温度或压力)超过或低于关键值，则系统中可能会发生非法或危险状态。必须及时检测到这类超限情况，并发信号通知系统发出适当的响应。

方框图

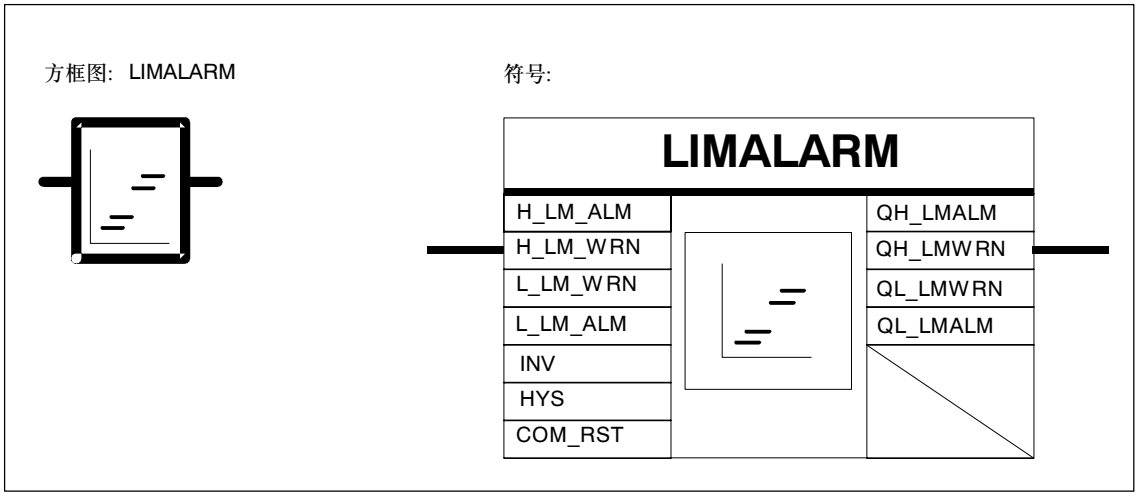


图2-25 LIMALARM，方框图和符号

功能描述

可以为输入变量INV选择四个限制值。如果达到并超过了其中一个限制值，就会输出一个限制信号。可以为关闭阈值设置一个滞后。

图2-26说明了LIMALARM块的工作原理:

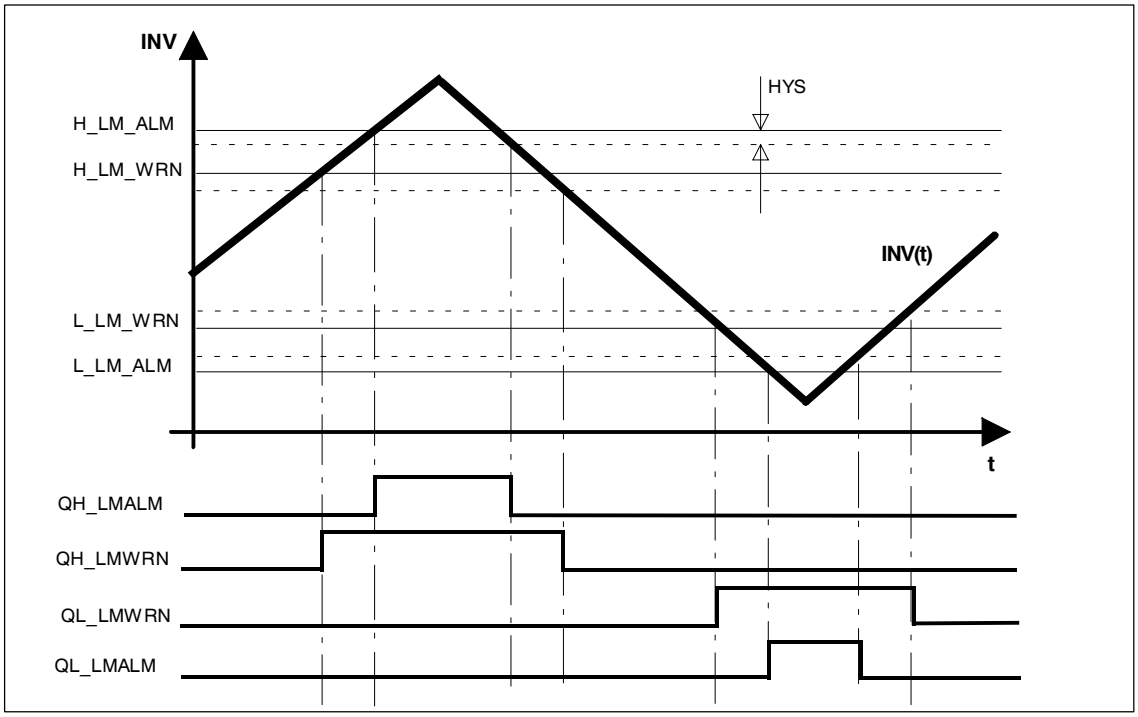


图2-26 LIMALARM块的工作原理

输入参数

下表给出了LIMALARM的输入参数的数据类型和结构。

表2-25 LIMALARM的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	H_LM_ALM	上限报警	技术范围 > H_LM_WRN	100.0
REAL	H_LM_WRN	上限警告	技术范围 > L_LM_WRN	90.0
REAL	L_LM_WRN	下限警告	技术范围 > L_LM_ALM	10.0
REAL	L_LM_ALM	下限报警	技术范围 < L_LM_WRN	0.0
REAL	INV	输入变量	技术取值范围	0.0
REAL	HYS	滞后	技术取值范围	1.0
BOOL	COM_RST	完全重启动		FALSE

输出参数

下表给出了LIMALARM的输出参数的数据类型和结构。

表2-26 LIMALARM的输出参数

数据类型	参数	注释	缺省值
BOOL	QH_LMALM	达到上限报警	FALSE
BOOL	QH_LMWRN	达到上限警告	FALSE
BOOL	QL_LMWRN	达到下限警告	FALSE
BOOL	QL_LMALM	达到下限报警	FALSE

完全重启动

在完全重启动期间，所有的信号输出都被设置成FALSE。

正常操作

该块根据下列函数工作:

QH_LMALM = TRUE	if	INV rises	and	INV >= H_LM_ALM
	or	INV falls	and	INV >= H_LM_ALM – HYS
QH_LMWRN = TRUE	if	INV rises	and	INV >= H_LM_WRN
	or	INV falls	and	INV >= H_LM_WRN – HYS
QL_LMWRN = TRUE	if	INV falls	and	INV <= L_LM_WRN
	or	INV rises	and	INV <= L_LM_WRN + HYS
QL_LMALM = TRUE	if	INV falls	and	INV <= L_LM_ALM
	or	INV rises	and	INV <= L_LM_ALM + HYS

只有满足下列条件，该块才能正常工作:

L_LM_ALM < L_LM_WRN < H_LM_WRN < H_LM_ALM

在输入H_LM_ALM、H_LM_WRN、L_LM_WRN和L_LM_ALM上设置了限制值。如果输入变量INV超过了限制值，则会设置输出信号位QH_LMALM、QH_LMWRN、QL_LMWRN和QL_LMALM。为了避免快速地反复置位和复位信号位，在复位输出之前输入值必须经过由滞后HYS定义的一段时间。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.12 LIMITER: 限制器

应用

如果参数是动态设置的(例如，通过过程变量计算出的设定值)，则这些参数可能会被设置成过程并不允许的数值。通过LIMITER，可以保持这些参数的数值在允许的范围之内。

方框图

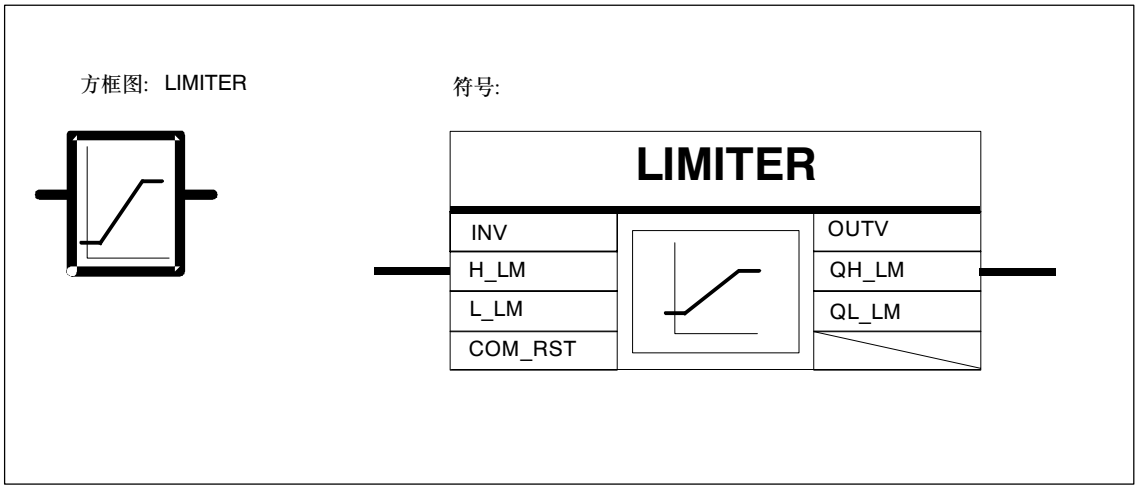


图2-27 LIMITER，方框图和符号

功能描述

当输入变量INV超出这些限制范围时，该块将输出变量OUTV限制在可选的上限H_LM和下限L_LM之间。OUTV的限制值是通过输出QH_LM和QL_LM发送出去的。

图2-28给出了LIMITER块的工作原理:

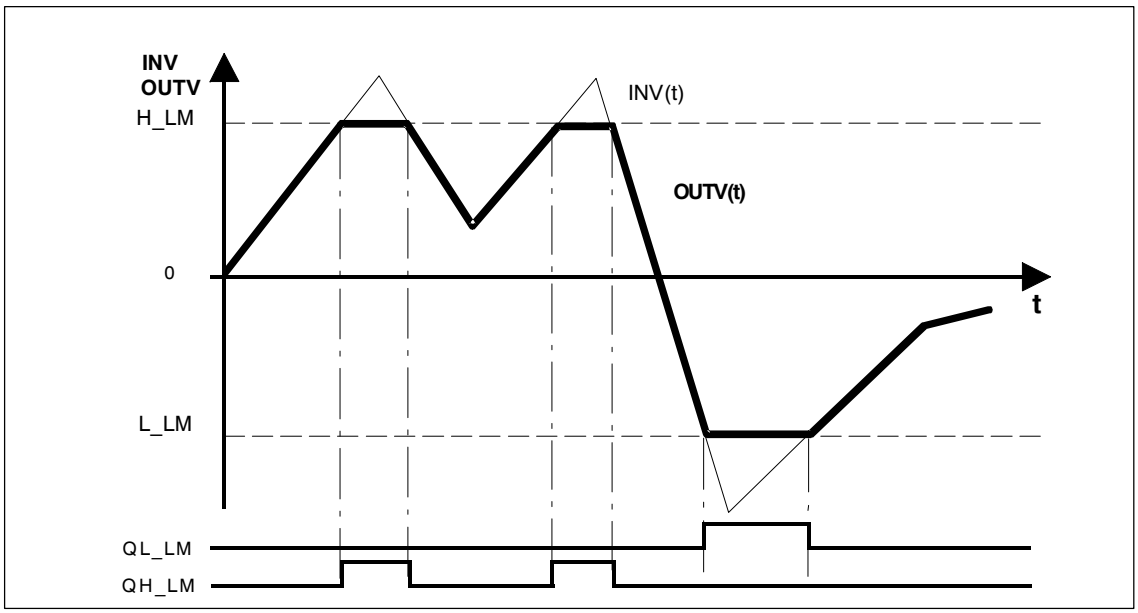


图2-28 LIMITER块的功能

输入参数

下表给出了LIMITER的输入参数的数据类型和结构。

表2-27 LIMITER的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	H_LM	上限	技术取值范围 > L_LM	100.0
REAL	L_LM	下限	技术取值范围 < H_LM	0.0
BOOL	COM_RST	完全重启动		FALSE

输出参数

下表给出了LIMITER的输出参数的数据类型和结构。

表2-28 LIMITER的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0
BOOL	QH_LM	已达到上限	FALSE
BOOL	QL_LM	已达到下限	FALSE

完全重启动

在完全重启动期间，所有的信号输出都被设置成FALSE。在OUTV上输出0.0。

正常操作

该块根据下列函数工作:

OUTV = H_LM, QH_LM = TRUE, QL_LM = FALSE if INV >= H_LM
OUTV = L_LM, QH_LM = FALSE, QL_LM = TRUE if INV <= L_LM
OUTV = INV, QH_LM = FALSE, QL_LM = FALSEif L_LM < INV < H_LM

只有满足下列条件，该块才能正常工作: L_LM < H_LM.

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.13 LMNGEN_C: 输出连续PID控制器

应用

该块用于构造一个连续PID控制器。它包含了控制器的调节值处理。

方框图

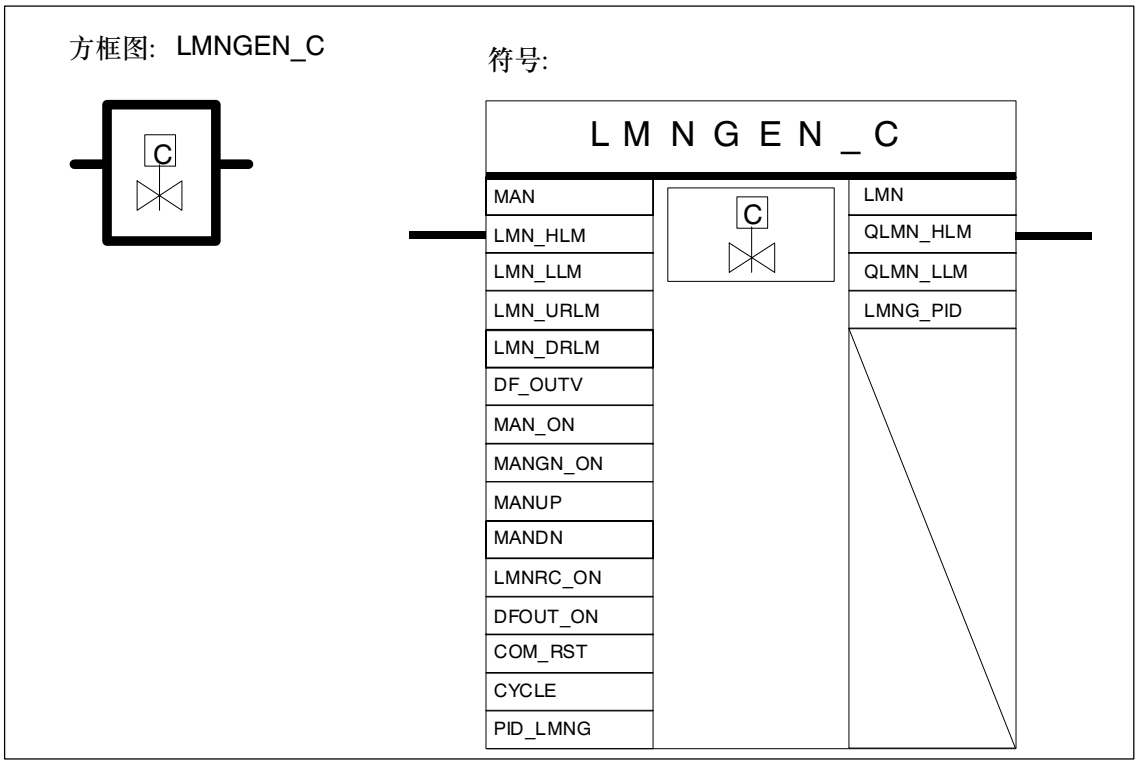


图2-29 LMNGEN_C，方框图和符号

功能描述

该块包含了手动-自动切换。在手动模式下，可以指定绝对值，或者通过开关增加或减小数值。可以将调节值和调节值的变化率限制到可选限制范围之内。

该块通常和PID算法块一起使用。

- **连续PID控制器: PID + LMNGEN_C**

图2-30给出了PID控制器的连接。

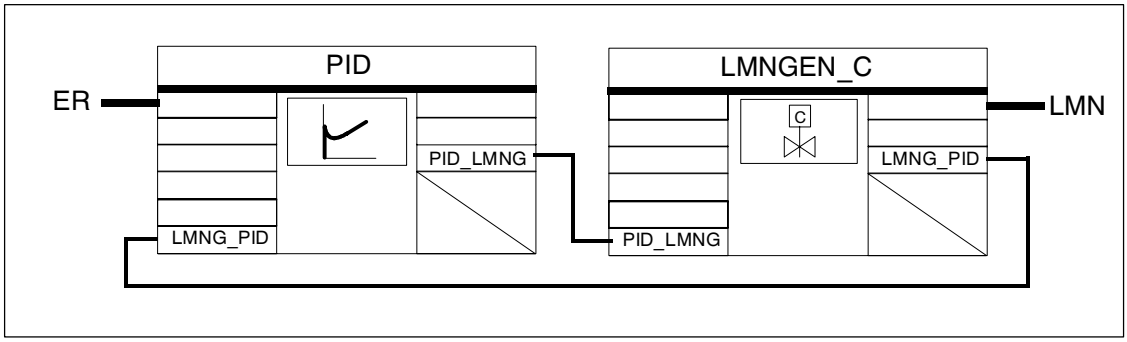


图2-30 连续PID控制器的连接

由于PID算法位于周期性中断优先级中，其周期时间已调整为与主系统时间常数相匹配，因此影响执行器的LMNGEN_C块可以位于更快的周期性中断优先级内，这样便可以允许操作员进行手动干预。使用结构化的输入/输出参数PID_LMNG和LMNG_PID来连接块。

输入参数

下表给出了LMNGEN_C的输入参数的数据类型和结构。

表2-29 LMNGEN_C的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	MAN	手动值	技术取值范围	0.0
REAL	LMN_HLM	调节值上限	技术范围 > LMN_ LLM	100.0
REAL	LMN_LLM	调节值下限	技术范围 < LMN_ HLM	0.0
REAL	LMN_URLM	调节值增加速率限制[1/s]	>0.0	10.0
REAL	LMN_DRLM	调节值减小速率限制[1/s]	>0.0	10.0
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
BOOL	MAN_ON	手动值打开		TRUE
BOOL	MANGN_ON	手动值发生器打开		FALSE
BOOL	MANUP	手动值增加		FALSE
BOOL	MANDN	手动值减小		FALSE
BOOL	LMNRC_ON	调节值变化速率打开		FALSE
BOOL	DFOUT_ON	缺省输出变量打开		FALSE

表2-29 LMNGEN_C的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s
STRUC	PID_LMNG	PID-LMNGEN接口		

输出参数

下表给出了LMNGEN_C的输出参数的数据类型和结构。

表2-30 LMNGEN_C的输出参数

数据类型	参数	注释	缺省值
REAL	LMN	调节值	0.0
BOOL	QLMN_HLM	达到调节值上限	FALSE
BOOL	QLMN_LLM	达到调节值下限	FALSE
STRUC	LMNG_PID	PID-LMNGEN接口	

完全重启动

在完全重启动期间，不论缺省位DFOUT_ON的状态如何，缺省值DF_OUTV都会切换到LMN输出。在完全重启动期间，输出的限制值和限制信号位也都有效。当控制器切换到正常运行模式时，块继续从DF_OUTV处开始运行。

正常操作

除了正常运行模式，该块还具有下列模式:

模式	DFOUT_ON	MAN_ON
自动	FALSE	FALSE
手动	FALSE	TRUE
缺省输出变量	TRUE	ANY

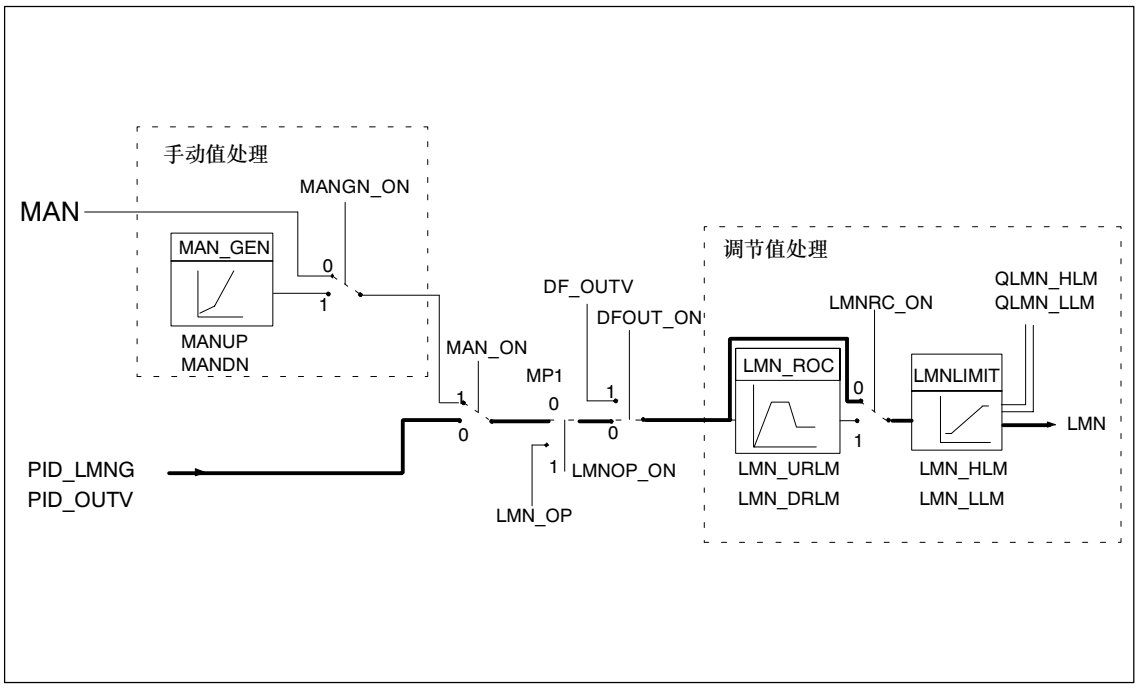


图2-31 连续PID控制器的调节值处理的方框图

- **自动模式**

如果没有选择其它模式，则PID算法计算出的数值被传送到调节值处理。如果激活了调节值变化速率功能(LMNRC_ON = TRUE)，则切换到自动模式并不会引起任何突然变化。

- **手动模式**

通过使用MAN_ON开关，可以切换到手动模式，中断控制回路。如果MANGN_ON = TRUE，则可以使用开关MANUP和MANDN，在限制值LMN_HLM和LMN_LLM之内，从当前数值开始增加或者减小调节值。变化率取决于限制:

在设置了MANUP或MANDN之后的头3秒钟之内:

dLMN/dt = (LMN_HLM - LMN_LLM) / 100 s

那么: dLMN/dt = (LMN_HLM - LMN_LLM) / 10 s

如果MANGN_ON = FALSE，并且MAN_ON = TRUE，则接入输入值MAN，作为调节值。

调节值的上限和下限必须应用到输入LMN_HLM和LMN_LLM上。此外，还可以限制调节值的变化率。在输入LMN_URLM和LMN_DRLM上设置增加变化率限制和减小变化率限制或者调节值，并通过开关LMNRC_ON激活。调节值出现在LMN输出上。调节值的限制由LMN_HLM和LMN_LLM完成，并通过位QLMN_HLM和QLMN_LLM进行指示。

- **缺省输出变量**

如果设置了DFOUT_ON = TRUE，则会将缺省值DF_OUTV应用到LMN输出。调节值限制有效，发信号对其进行指示。只要激活了变化率功能(LMNRC_ON = TRUE)，切换到“缺省输出变量”或者从“缺省输出变量”中切换出来就不会引起突然变化。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

通过*组态工具*影响调节值

“回路监视器”中的LMN显示和设置

*组态工具*本身有到LMNGEN_C块的接口。因此，始终可以中断调节变量分支(例如，当在安装了*组态工具*的编程设备或PC上工作时，用于测试目的)，并设置用户自己的调节值(LMN_OP)。(图2-32)。

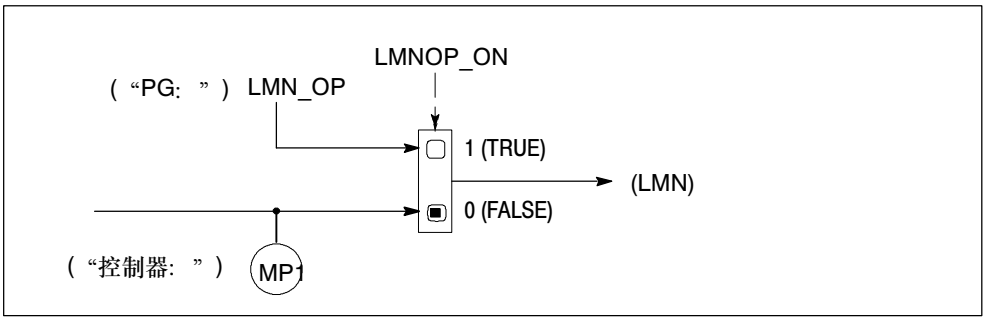


图2-32　　使用*组态工具干预调节值分支*

回路监视器有一个标识为“调节值”的域。此处，在上面的域(“控制器: ”)中，显示当前分配给测量点MP1的调节值。在下面的域(“PG: ”)中，可以设置参数LMN_OP。

通过*组态工具*切换到手动调节值

如果*组态工具*中的开关设置为“PG: ”，则在控制器FB中结构开关LMNOP_ON的开关信号设置为TRUE，而将LMN_OP引出到调节值LMN。

如果在调节变量分支中激活了变化率限制LMN_ROC，则可以在“PG: ”和“控制器: ”之间进行切换，而不会引起任何突变。调节变量所切换回到的值(MP1)可以在回路监视器的“控制器: ”显示域中看到。然后，LMN以在LMN_ROC上设置的变化率返回到此值。

只有在回路监视器中通过单击“发送”按钮将这些干预发送到可编程控制器之后，它们才会影响处理过程。

参数LMNOP_ON、LMN_OP和MP1是静态变量，不能作为块的输入/输出参数来获得。不能连接这些参数，并且只能通过*组态工具*使用和监视这些参数。

2.1.14 LMNGEN_S: 输出PID步进控制器

应用

该块用于为具有积分分量的执行器(例如，电机驱动器阀)构造一个PID步进控制器。它包含了控制器的调节值处理。该步进控制器的运行，可以有定位反馈信号，也可以没有定位反馈信号。

方框图

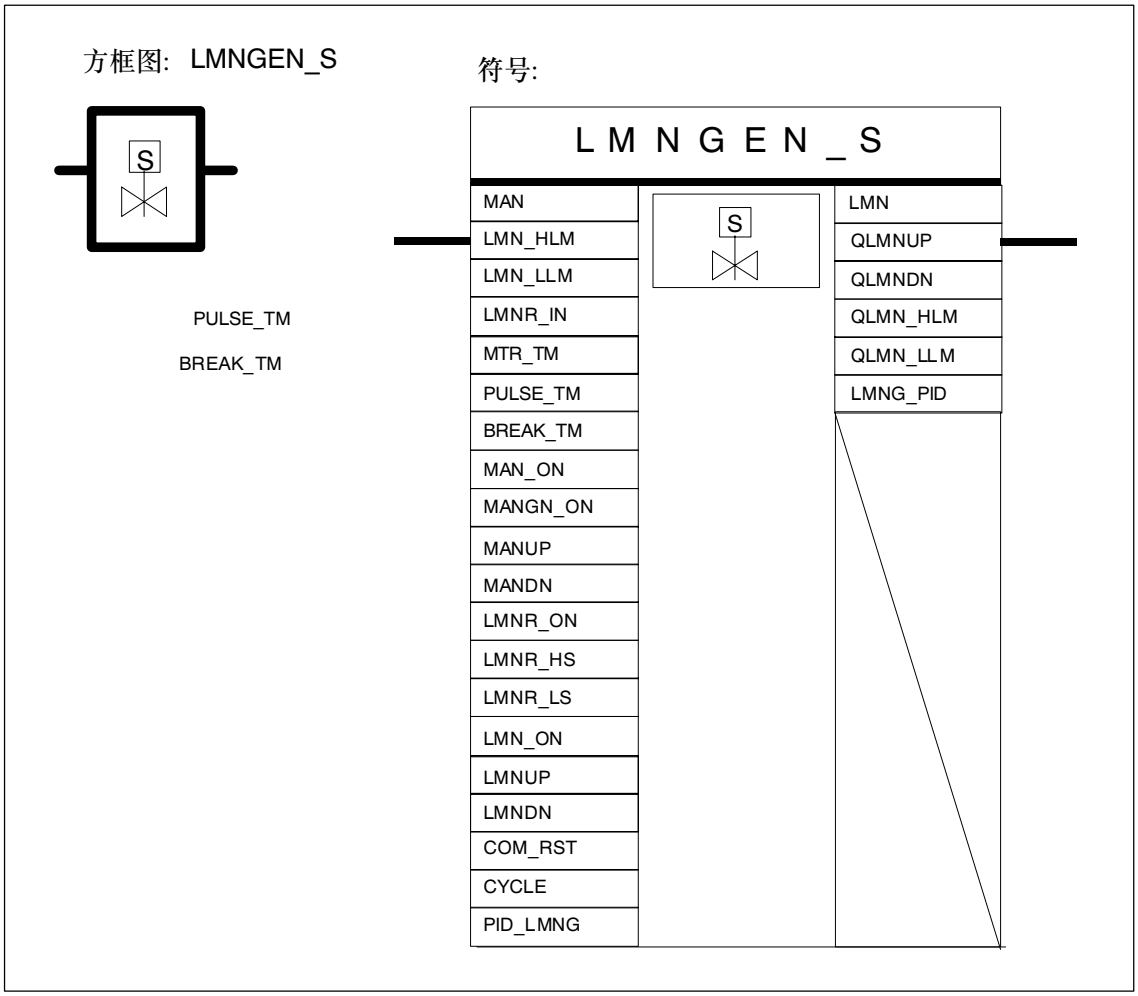


图2-33 LMNGEN_S，方框图和符号

功能描述

如果定位反馈信号可用，则该块可用作定位控制器。通过手动/自动选项，可以在PID算法提供的调节变量和手动值之间进行切换。可以输入一个手动值作为绝对值，也可以使用手动值发生器增加或减小手动值。块利用调节变量和定位反馈信号之间的差值生成脉冲信号，用于通过三步单元和脉冲发生器来控制执行器。调整三步单元的响应阈值可以减少控制器的切换频率。

在没有定位反馈信号的情况下，该块也可以正常工作。在一个积分器中计算PID算法的积分I动作和定位反馈信号，然后与剩余PD动作进行比较。然后将差值应用到三步单元和脉冲发生器，为最终的控制单元生成脉冲信号。调整三步单元的响应阈值可以减少控制器的切换频率。

该块通常和PID算法块一起使用。

- **PID步进控制器：PID + LMNGEN_S**

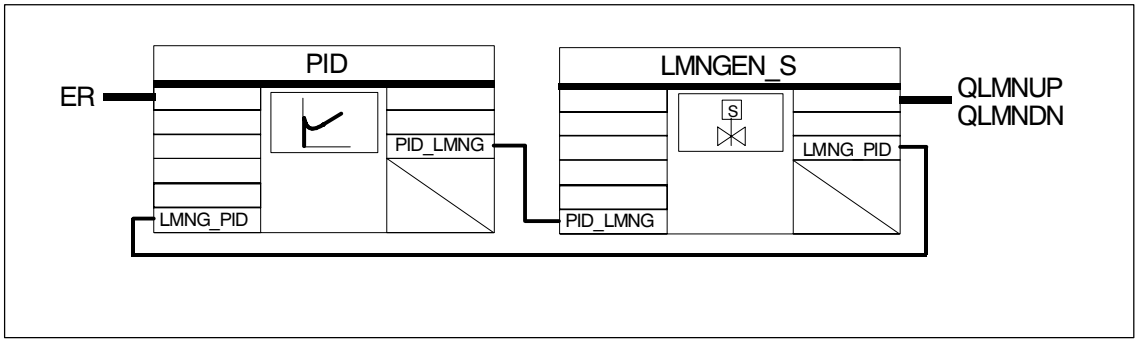


图2-34　PID步进控制器的连接

由于PID算法位于周期性中断优先级中，其周期时间已调整为与主系统时间常数相匹配，因此影响执行器的LMNGEN_S块可以位于更快的周期性中断优先级内，这样便可以允许操作员进行手动干涉。使用结构化的输入/输出参数PID_LMNG和LMNG_PID来连接块。

输入参数

下表给出了LMNGEN_S的输入参数的数据类型和结构。

表2-31 LMNGEN_S的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	MAN	手动值	技术取值范围	0.0
REAL	LMN_HLM	调节值上限	技术范围 > LMN_ LLM	100.0
REAL	LMN_LLM	调节值下限	技术范围 < LMN_ HLM	0.0
REAL	LMNR_IN	定位反馈信号	技术取值范围	0.0
TIME	MTR_TM	电机启动时间	≥ CYCLE	T#30s
TIME	PULSE_TM	最小脉冲时间	≥ CYCLE	T#3s
TIME	BREAK_TM	最小断开时间	≥ CYCLE	T#3s
BOOL	MAN_ON	手动值打开		TRUE
BOOL	MANGN_ON	手动值发生器打开		FALSE
BOOL	MANUP	手动值增加		FALSE
BOOL	MANDN	手动值减小		FALSE
BOOL	LMNR_ON	定位反馈信号打开		FALSE
BOOL	LMNR_HS	定位反馈信号的上限信号		FALSE
BOOL	LMNR_LS	定位反馈信号的下限信号		FALSE
BOOL	LMNS_ON	调节值信号打开		FALSE
BOOL	LMNUP	调节值信号增加		FALSE
BOOL	LMNDN	调节值信号减小		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s
STRUC	PID_LMNG	PID-LMNGEN接口		

输出参数

下表给出了LMNGEN_S的输出参数的数据类型和结构。

表2-32 LMNGEN_S的输出参数

数据类型	参数	注释	缺省值
REAL	LMN	调节值	0.0
BOOL	QLMNUP	调节值信号增加	FALSE
BOOL	QLMNDN	调节值信号减小	FALSE
BOOL	QLMN_HLM	达到调节值上限	FALSE
BOOL	QLMN_LLM	达到调节值下限	FALSE
STRUC	LMNG_PID	PID-LMNGEN接口	

完全重启动

在完全重启动期间，所有的信号输出都被设置成零。

正常操作

除了正常运行模式，该块还具有下列模式:

模式		LMNR_ON	LMN_ON	MAN_ON
带有定位反馈信号的步进控制器	自动模式	TRUE	FALSE	FALSE
	调节值手动模式	TRUE	FALSE	TRUE
	手动模式调节值输出信号	TRUE	TRUE	ANY
无定位反馈信号的步进控制器	自动模式	FALSE	FALSE	FALSE
	手动模式调节值输出信号	FALSE	TRUE	any

在模式“无定位反馈信号的步进控制器”中，控制器无法获得与阀位置相关的任何信息，因此指定手动值便无任何实际意义。在无定位反馈信号的步进控制器中，没有“调节值手动模式”。

警告

如果没有限位信号，控制器便无法识别阀停止限位；这样，举例来讲，便可能会发生在阀已经完全打开的情况下，控制器仍然输出打开阀门的信号。

如果没有限位信号，输入LMNR_HS和LMNR_LS必须设置为FALSE。

带有定位反馈信号的步进控制器

如果定位反馈信号可用，则必须设置LMNR_ON = TRUE。然后，块将作为定位控制器运行。

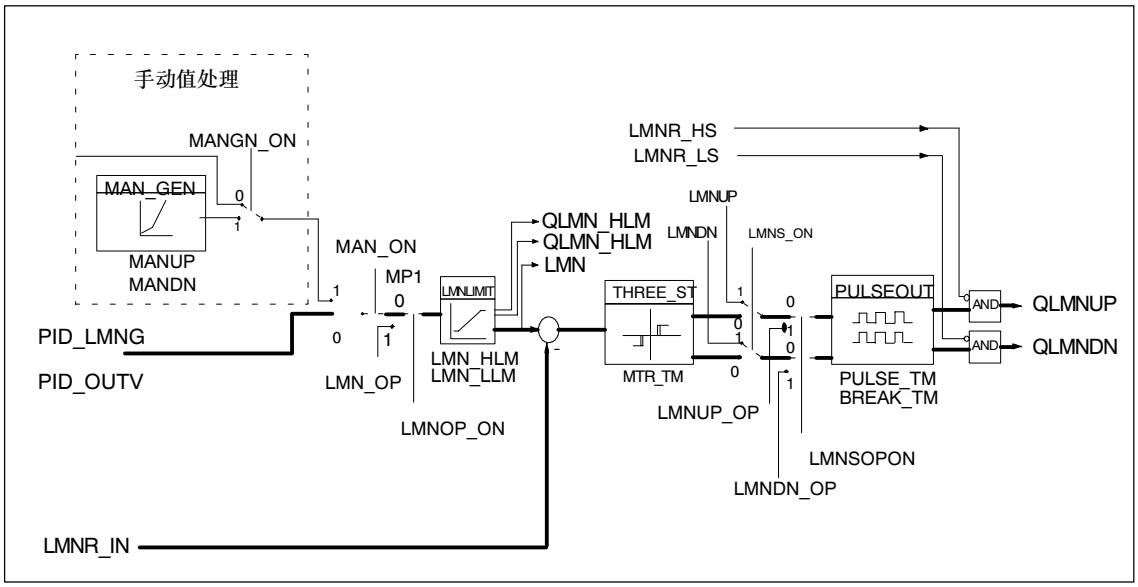


图2-35　　带有定位反馈信号的步进控制器

- **调节值手动模式**

在定位控制模式下，可以通过开关MAN_ON切换到手动模式，然后以绝对值形式在输入MAN上输入一个调节值，或者使用手动值发生器MAN_GEN。

如果设置了MANGN_ON = TRUE，则可以使用开关MANUP和MANDN，在限制值LMN_HLM和LMN_LLM的范围内，从其当前值处增加或减小调节值。如下所示，变化率取决与限制值:

在设置了MANUP或MANDN之后的头3秒钟之内:

dLMN/dt = (LMN_HLM - LMN_LLM) / 100 s

那么: dLMN/dt = (LMN_HLM - LMN_LLM) / 10 s

如果MANGN_ON = FALSE，并且MAN_ON = TRUE，则接入输入值MAN，作为调节值。

- **调节值输出信号的手动模式**

如果设置了LMNS_ON = TRUE，则可以直接影响二进制输出信号。通过开关LMNUP和LMNDN设置驱动信号输出QLMNUP和QLMNDN。最小脉冲时间PULSE_TM和最小断开时间BREAK_TM都被考虑在内。如果设置了限位开关LMNR_HS或LMNR_LS中的一个，则相应的输出信号QLMNUP或QLMNDN被禁用。

- **自动模式**

来自PID算法PID_LMNG.PID_OUTV的调节值被限制在可选数值LMN_HLM和LMN_LLM之间。调节值和定位反馈信号之间的差值被转换到带有滞后的三步单元中。从电机执行时间中计算出关闭阈值。要减少切换频率，需要修改接通阈值。脉冲发生器PULSEOUT确保最小脉冲和断开时间符合限制范围。如果设置了限位开关LMNR_HS或LMNR_LS中的一个，则相应的输出信号QLMNUP或QLMNDN被禁用。

无定位反馈信号的步进控制器

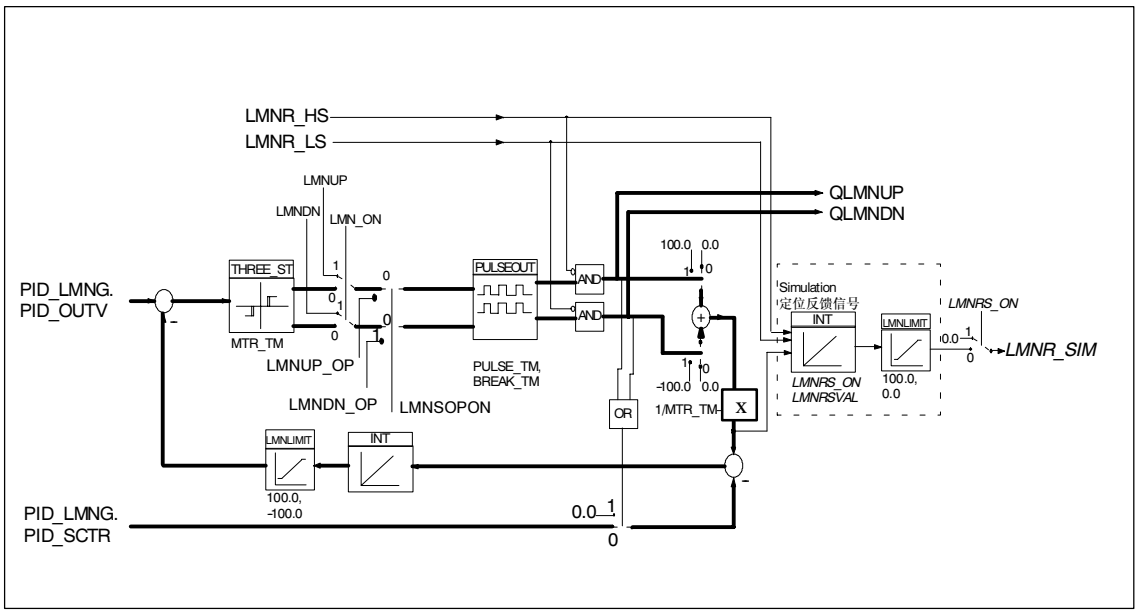


图2-36 无定位反馈信号的步进控制器

如果无定位反馈信号可用，则设置LMNR_ON = FALSE。

- **调节值输出信号的手动模式**

如果设置了LMNS_ON = TRUE，则可以直接影响二进制输出信号。通过开关LMNUP和LMNDN设置驱动信号输出QLMNUP和QLMNDN。最小脉冲时间PULSE_TM和最小断开时间BREAK_TM都考虑在内。如果设置了限位开关LMNR_HS或LMNR_LS中的一个，则相应的输出信号QLMNUP或QLMNDN被禁用。

- **自动模式**

PID算法的积分I动作和内部定位反馈信号之间的差值，在积分器INT中进行积分运算。积分器的输入是范围为100.0 / 0.0 /-100.0的差值；根据输出信号QLMNUP和QLMNDN的状态，该差值与电机执行时间MTR_TM和偏差信号与PID算法的复位时间TI相关的增益(PID_LMNG.PID_SCTR)相乘的乘积相关。积分器的输出构成了反馈信号，该信号与PID算法的剩余PD动作(PID.LMNG.PID_OUTV)相比较。差值转换到带有滞后的三步单元。从电机执行时间中计算出关闭阈值。

要减少切换频率，需要修改接通阈值。脉冲发生器PULSEOUT确保最小脉冲和断开时间符合限制范围。如果设置了限位开关LMNR_HS或LMNR_LS中的一个，则相应的输出信号QLMNUP或QLMNDN被禁用。

定位反馈信号的模拟

通过一个将电机驱动时间MTR_TM作为复位时间的积分器来模拟定位反馈信号。将参数LMNRS_ON从FALSE改成TRUE，启动初始值为LMNRSVAL的模拟。如果LMNRS_ON设置为FALSE，则为参数LMNR_SIM输出初始值LMNRSVAL。参数LMNRS_ON、LMNRSVAL和LMNR_SIM位于静态变量区中。通过*组态工具*为它们提供数值。如果LMNR_HS = TRUE，则向上积分被禁用。如果LMNR_LS = TRUE，则向下积分被禁用。

警告

仅模拟定位反馈信号。它不需要对应最终控制单元的实时定位反馈信号。

*组态工具*的PID过程辨识功能需要将定位反馈信号用作输入变量。

如果实际的定位反馈信号存在，则应该使用该信号。

块内部限制

在块内部，未对任何值进行限制；系统不会检查参数。

通过*组态工具*影响调节值

回路监视器中的LMN显示和设置

*组态工具*本身具有到LMNGEN_S块的接口。因此，当在编程设备或PC上工作时，始终可以中断调节值分支，并设置用户自己的调节值(图2-37)。

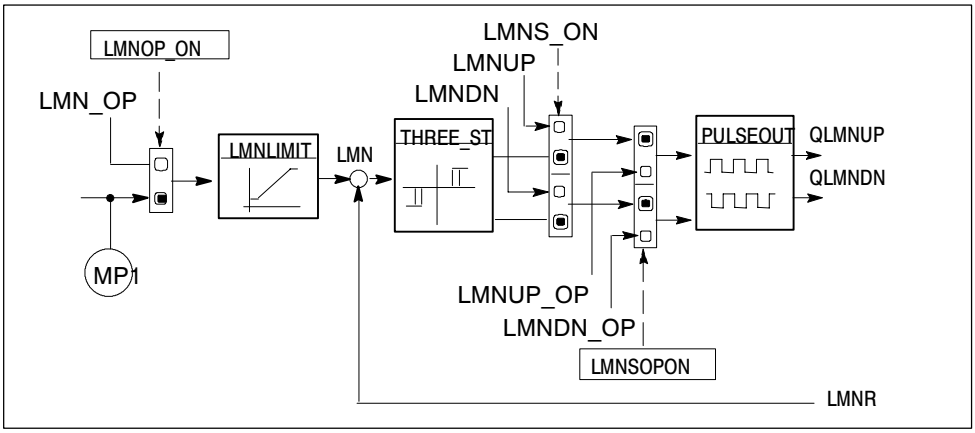


图2-37　　使用*组态工具*干预调节值分支

回路监视器有一个标识为“调节值”的域。此处，在上面的域(“控制器: ”)中，显示当前应用到测量点MP1的调节值。在下面的域(“PG: ”)中，可以设置参数LMN_OP。

通过*组态工具切换到手动调节值*

如果*组态工具*中的开关设置为“PG: ”，则在控制器FB中结构开关LMNOP_ON的开关信号设置为TRUE，而将LMN_OP引出到调节值LMN。

如果在调节变量分支中激活了变化率限制LMN_ROC，则可以在“PG: ”和“控制器: ”之间进行切换，而不会引起任何突变。调节变量所切换回到的值(MP1)可以在回路监视器的“控制器: ”显示域中看到。然后，LMN以在LMN_ROC上设置的变化率返回到此值。

如果调节值域中的开关“控制器: /PG: ”设置成“PG: ”，则设置参数LMNSOPON = TRUE，并可以在回路监视器中通过参数LMNUP_OP (向上) 或LMNDN_OP (向下)来控制执行信号输出。这适用于带有和不带定位反馈信号的步进控制器。只有在回路监视器中通过单击“发送”按钮，将这些干预传送到可编程控制器之后，它们才会影响实际过程。参数LMNOP_ON、LMN_OP、MP1、LMNSOPON、LMNUP_OP和LMNDN_OP是静态变量并且不能用作块的输入/输出参数。不能连接这些参数，并且只能通过*组态工具*使用和监视这些参数。

2.1.15 LP_SCHED: 回路调度程序

应用

当调用大量具有不同采样时间的控制回路时，以及调用具有大采样时间的控制回路时，周期性中断的优先级范围就可能不够用。通过使用回路调度程序LP_SCHED，可以在同一个周期性中断优先级中，以相同的时间间隔调用具有不同采样时间的多个控制回路。

并不强制使用回路调度程序。也可以使用无调度程序功能的OB直接调用控制器FB。

方框图

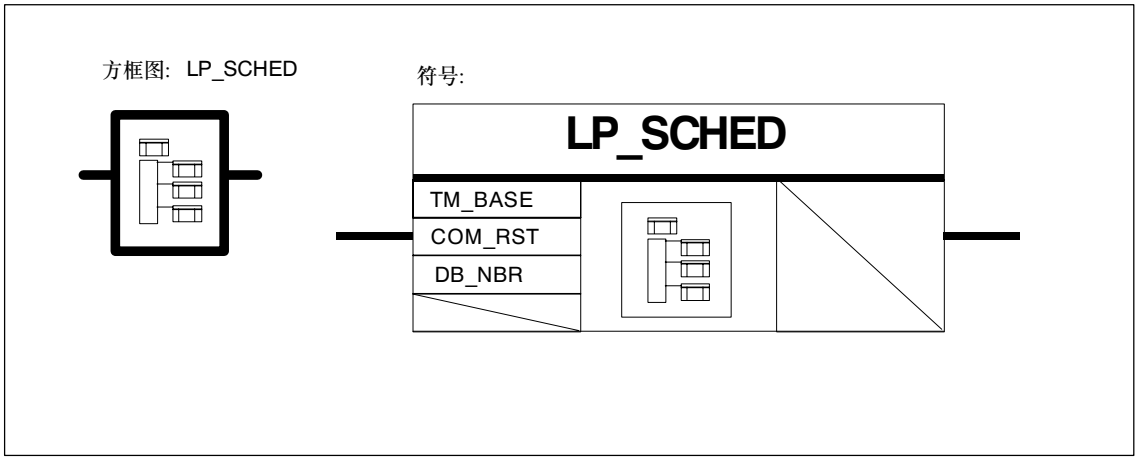
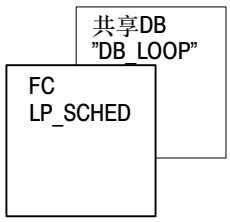


图2-38 LP_SCHED，方框图和符号

功能描述

在同一个周期性中断优先级中调度多个控制器是在LP_SCHED功能中实现的。必须在所有控制回路之前调用该块。用于执行各个块调用的数据存储在共享数据块(DB_LOOP)中。



LP_SCHED块处理共享数据块，并置位ENABLE位，以保持与控制回路的采样时间一致。它支持周期性中断优先级的时基，并根据控制回路的不同采样时间来调用各个控制回路。根据此优先级中各个回路的采样时间，来调用和处理这些控制回路。在完成块调用之后，必须复位ENABLE位。必须编程块调用和ENABLE位的复位。

必须手动禁用单个控制回路的调用。还可以复位单个控制回路(完全重启动)。

注意

该块并不会检查是否已经存在具有编号DB_NBR的共享DB，以及“最大回路编号”参数GLP_NBR是否符合DB长度。如果参数分配不正确，则CPU切换到STOP，同时报告内部系统错误。

输入参数

下表给出了LP_SCHED的输入参数的数据类型和结构。

表2-33 LP_SCHED的输入参数

数据类型	参数	注释	允许使用的数值
TIME	TM_BASE	时基	>= 1毫秒
BOOL	COM_RST	完全重启动	
BLOCK_DB	DB_NBR	数据块编号	

输出参数

下表给出了LP_SCHED的输出参数的数据类型和结构。

表2-34

数据类型	参数	注释
该块并没有任何输出参数。		

重启

当COM_RST = TRUE时，缺省设置如下:

当前控制回路编号:	ALP_NBR =	0

所有控制回路的调用数据在**GLP_NBR**时缺省如下:

释放:	ENABLE =	NOT MAN_DIS
探测时间:	CYCLE =	GV(MAN_CYC)
重启:	COM_RST =	TRUE
本地调用编号:	ILP_COU =	0

GV(MAN_CYC) = MAN_CYC取整为TM_BASE*GLP_NBR的倍数

共享数据块“DB_LOOP”

下表给出了用于控制器调用的共享数据块DB_LOOP。

表2-35

数据类型	参数	注释	允许使用的数值	缺省值
INT	GLP_NBR	最大回路数	1 - 256	2
INT	ALP_NBR	实际回路数	0 - 256	0
TIME	LOOP_DAT[1]. MAN_CYC	回路数据[1]手动采样时间	>= 1毫秒	T#1s
BOOL	LOOP_DAT[1]. MAN_DIS	回路数据[1]手动回路禁用		FALSE
BOOL	LOOP_DAT[1]. MAN_CRST	回路数据[1]手动完全重启动		FALSE
BOOL	LOOP_DAT[1]. ENABLE	回路数据[1]启用回路		FALSE
BOOL	LOOP_DAT[1]. COM_RST	回路数据[1]完全重启动		FALSE
INT	LOOP_DAT[1]. ILP_COU	回路数据[1]内部回路计数器		0
TIME	LOOP_DAT[1]. CYCLE	回路数据[1]采样时间	>= 1毫秒	T#1s

调用控制回路 使用LP_SCHED

通过三个输入参数，将LP_SCHED功能集成在CPU的调用策略中。周期性中断等级的时基在TM_BASE输入处指定。在**同一个**周期性中断等级(例如OB35)中，使用条件块调用来调用控制回路，并查询共享数据块中的ENABLE位。

如果是在完全重启动级别上进行调用，则设置输入COM_RST = TRUE。在周期性中断级中，必须将此参数复位为FALSE。使用输入参数DB_NBR分配共享数据块(表2-35)，该数据块带有用于周期性中断级中的控制回路的时间数据。

编程控制回路调用(共享的DB)

必须在无*组态工具*支持的情况下，编程回路调度程序。

数据块(DB_LOOP)包含要在周期性中断级中处理的控制回路总数(最大256)的参数，以及指示当前正在处理的控制回路的参数，如下所示:

GLP_NBR　最高控制回路号
ALP_NBR　处理周期中正在处理的控制回路的编号

通过控制回路在DB内的条目序列中的位置来指示每个控制回路的编号。

单个控制回路的调用数据，以结构化的形式存储在LOOP_DAT域中。如果要添加控制回路，必须通过在声明视图中修改ARRAY数据类型来匹配LOOP_DAT的域长度。例如，对于10个回路，应该输入ARRAY[1..10]。此外，还必须在数据视图中修改参数GLP_NBR。它一定不能高于域长度。

参数COM_RST和CYCLE必须链接到属于所调用的控制回路的FB中的相应输入参数上。必须由用户编程该连接。如果设置了ENABLE参数，则会调用相应的控制回路。在调用了控制器之后，必须复位ENABLE位。用户必须编程条件控制器调用和ENABLE位的复位。

通过使用可以手动置位的参数MAN_CYC / MAN_DIS / MAN_CRST，可以决定是否调用控制回路。可以在线(或者说是在运行期间)更改这些调用，只要仅仅是重写了参数，而不是重新生成整个DB。各个参数的含义如下:

MAN_CYC　相应控制器的采样时间(四舍五入到CYCLE中的TM_BASE * GLP_NBR的整数倍)。

MAN_DIS　禁用控制器调用。

MAN_CRST　控制器完全重启动。

处理调用

如果控制器调用数据的ENABLE信号置位，则根据DB中设置的参数处理每个回路。

数据块的工作顺序是从上到下。在每个周期中，回路调度程序向前移动一个回路号(ALP_NBR)；移动顺序是回路输入到DB中的顺序。然后，内部计数器ILP_COU减一。如果设置了ILP_COU = 0，则回路调度程序设置相应回路的ENABLE位。在调用完成之后必须复位ENABLE位，这必须由用户编程。

在处理CYCLE时，传送参数MAN_CYC:

CYCLE = GV (MAN_CYC),　　GV = 所有多个回路

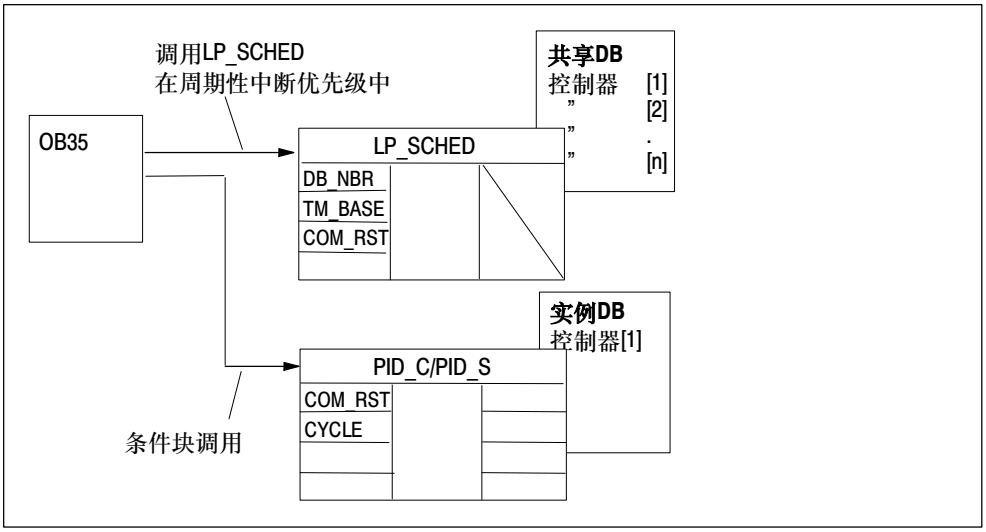


图2-39　　使用回路调度程序LP_SCHED进行控制器调用的原则

- 禁用所选回路:

  如果在DB中设置了 “MAN_DIS” 位，则将ENABLE位复位为FALSE，并且从回路调度程序的处理中将相关的回路排除在外。

- 复位所选择的回路(完全重启动):

  如果在DB中设置了 “MAN_CRST” 位，则会设置COM_RST = TRUE，然后复位MAN_CRST。然后处理控制回路的完全重启动例行程序。然后，在接下来的调用周期中，将COM_RST设置成FALSE。

注意

如果要插入或删除控制回路并重新生成整个DB，而又不希望执行回路调度程序完全重启动，则必须将内部回路计数器(ILP_COU[n])和当前回路号的参数ALP_NBR设置为零。

通过LP_SCHED调用回路的条件

为了确保特定控制器的相邻调用之间的时间间隔保持恒定，以使CPU上的负载均匀分布，在周期性中断级别的每个时基单元内只能处理一个控制回路。在分配采样时间MAN_CYC时，一定要记住与时基(TM_BASE)相关的下列条件:

- 单个回路的处理时间必须短于周期性中断级别的时基(TM_BASE)。
- 控制回路的采样时间(MAN_CYC)必须是时基和要处理的控制器数(GLP_NBR)的乘积的整数倍(GV)。

  MAN_CYC = GV (TM_BASE * GLP_NBR)。

回路调度实例

下面的实例说明了四个回路在一个周期性中断级别中的调用顺序(图2-40)。在每个时基单元中只能处理一个回路。回路调用的顺序和时间位移(TD1到TD5)所使用的顺序由共享数据块中调用数据的顺序决定。

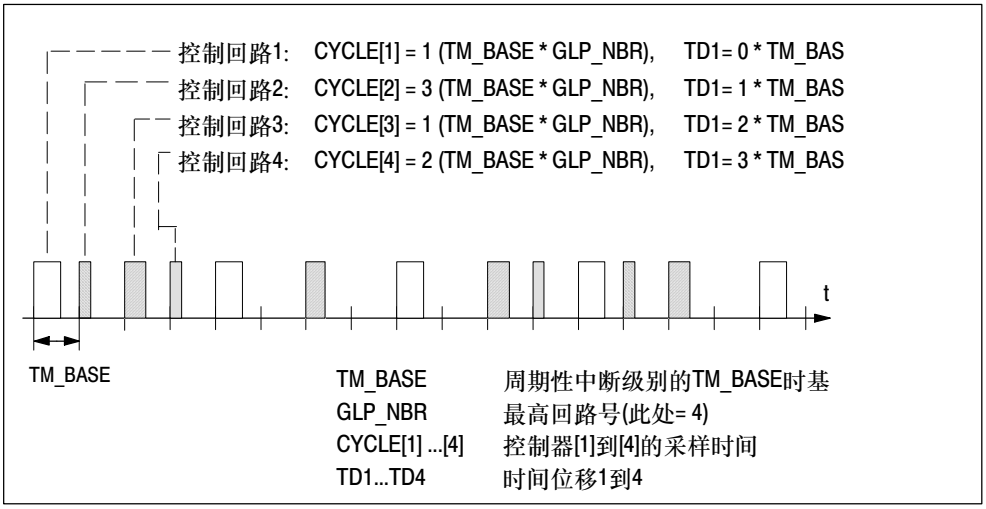


图2-40 以不同时间间隔调用的四个回路的调用顺序

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.16 NONLIN: 非线性静态功能

应用

通过NONLIN，可以使用可选择的功能来修改输入数值(例如，来自热电偶的测量值)。

方框图

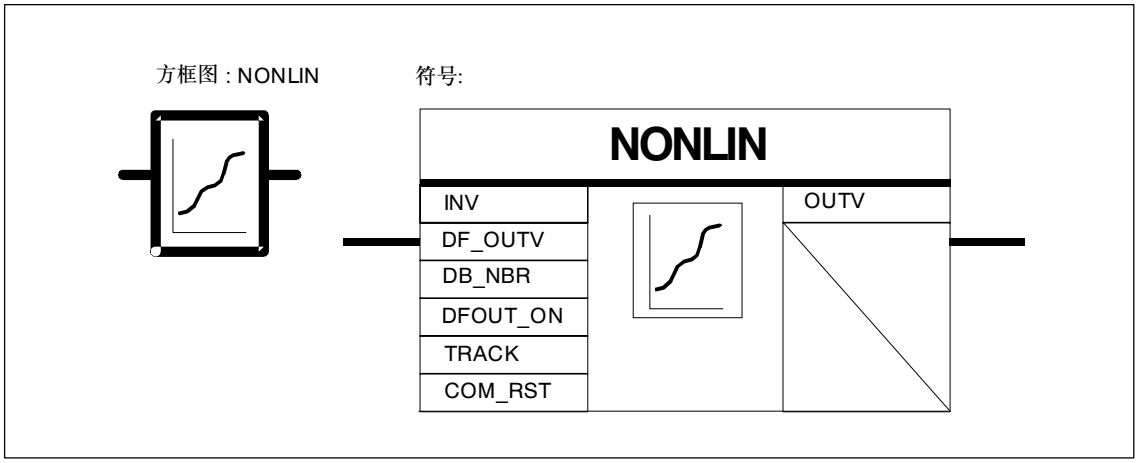


图2-41 NONLIN，方框图和符号

功能描述

功能的点存储在共享数据块中。NONLIN将相应的输出值分配给来自功能的当前输入值。如果输入值小于点0处的值，则使用点0和点1之间的斜率来外推该值。如果该数值高于最后一个点的数值，则使用最后一对点的斜率外推该值。为了使块能够提供了一个合理的结果，各个点的数值必须具有严格单调增加的斜率。

通过使用控制输入，可以直接输出所选择的数值或输入变量本身，作为输出变量。

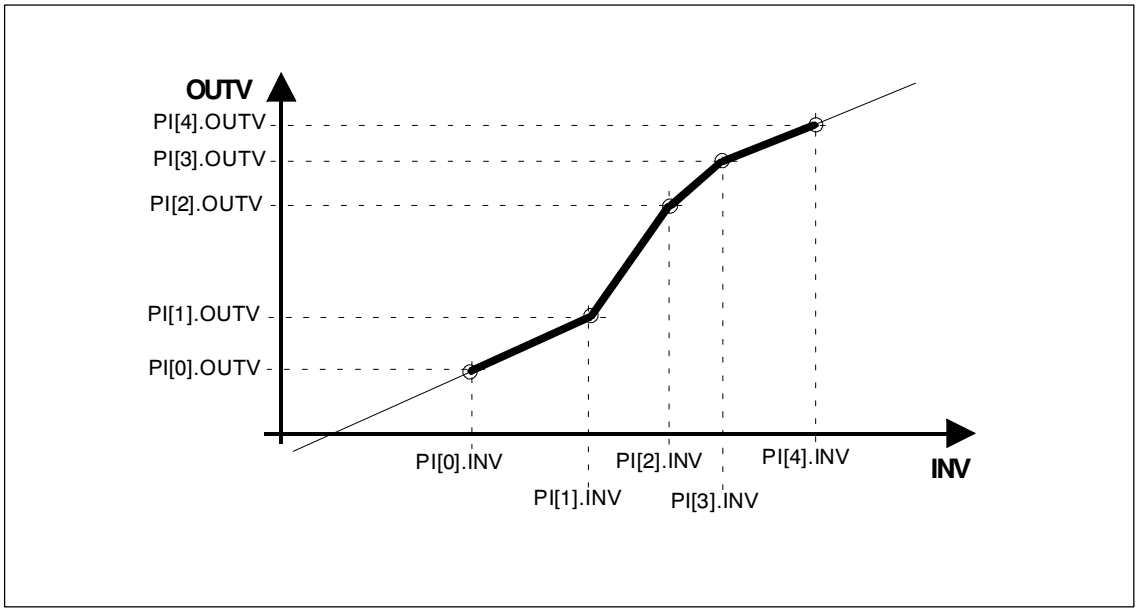


图2-42 OUTV = f(INV)

注意

该块并不会检查具有编号DB_NBR的共享DB是否存在，以及参数DB_NBR.NBR_PTS(点数)是否与数据块的长度相匹配。如果参数设置不正确，则CPU切换到STOP，同时报告内部系统错误。

输入参数

下表给出了NONLIN的输入参数的数据类型和结构。

表2-36 NONLIN的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
BLOCK_DB	DB_NBR	数据块编号		DB 1
BOOL	DFOUT_ON	缺省输出变量打开		FALSE
BOOL	TRACK	跟踪OUTV = INV		FALSE
BOOL	COM_RST	完全重启动		FALSE

输出参数

下表给出了NONLIN的输出参数的数据类型和结构。

表2-37 NONLIN的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

共享数据块DB_NBR

下表给出了**DB_NBR**的数据类型和参数(缺省有5个曲线点)。

数据类型	参数	注释	允许使用的数值	缺省值
INT	DB_NBR.NBR_PTS	最高坐标	1 - 255	4
REAL	DB_NBR.PI[0].OUTV	输出变量[0]，点0	技术取值范围	0.0
REAL	DB_NBR.PI[0].INV	输入变量[0]，点0	技术取值范围	0.0
REAL	DB_NBR.PI[1].OUTV	输出变量[1]，点1	技术取值范围	0.0
REAL	DB_NBR.PI[1].INV	输入变量[1]，点1	技术取值范围	0.0
REAL	DB_NBR.PI[2].OUTV	输出变量[2]，点2	技术取值范围	0.0
REAL	DB_NBR.PI[2].INV	输入变量[2]，点2	技术取值范围	0.0
REAL	DB_NBR.PI[3].INV	输入变量[3]，点3	技术取值范围	0.0
REAL	DB_NBR.PI[3].OUTV	输出变量[3]，点3	技术取值范围	0.0
REAL	DB_NBR.PI[4].OUTV	输出变量[4]，点4	技术取值范围	0.0
REAL	DB_NBR.PI[4].INV	输入变量[4]，点4	技术取值范围	0.0

完全重启动

在完全重启动期间，输出OUTV = 0.0。

如果DFOUT_ON=TRUE，则输出DF_OUTV。

正常操作

在正常操作模式下，在原始值的当前点通过内插法计算。

DFOUT_ON和TRACK在OUTV上有下列影响:

模式	DFOUT_ON	TRACK	OUTV
缺省输出变量	TRUE	ANY	DF_OUTV
TRACK	FALSE	TRUE	OUTV = INV
正常操作	FALSE	FALSE	OUTV = f(INV)

- **缺省输出变量**

如果设置了DFOUT_ON = TRUE，则输出DF_OUTV，并且OUTV上的变化是阶跃变化。在切换到DFOUT_ON = FALSE时，OUTV上的变化也是阶跃变化。

- **跟踪**

如果设置了TRACK=TRUE，则直接输出该输入值(OUTV=INV)。随DFOUT_ON的变化是一个阶跃变化。TRACK的优先级比DFOUT_ON低。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.17 NORM: 物理规格化

应用

通常，传感器所提供的过程变量的数值所处的范围并不适合用户使用(例如，0 ~ 10 V对应0 ~ 1200°C或0 ~ 10 V对应0 ~ 3000 rpm)。通过调整设置值或过程变量，可以使两个变量具有相同的数值范围。

方框图

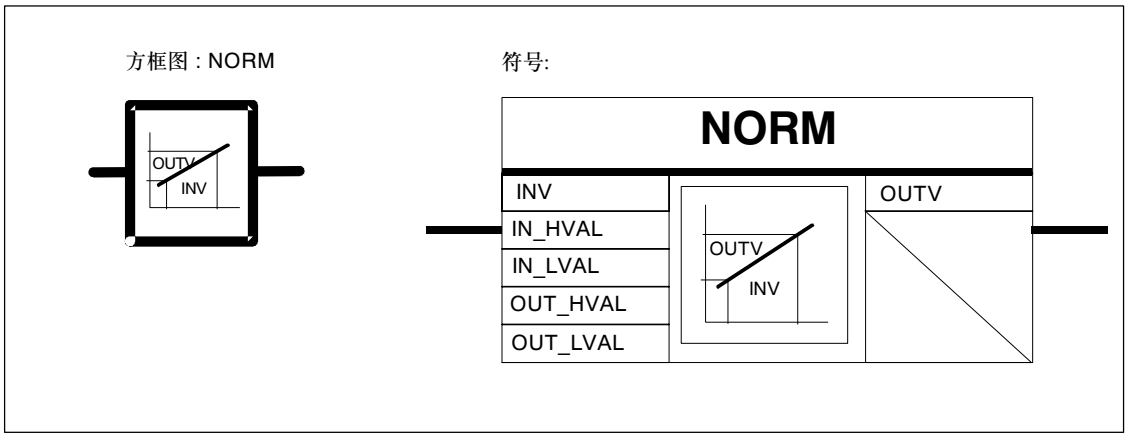


图2-43 NORM，方框图和符号

功能描述

该块对输入变量进行规格化，生成一个具有不同数值范围的输出变量。规格化曲线由两个点定义。

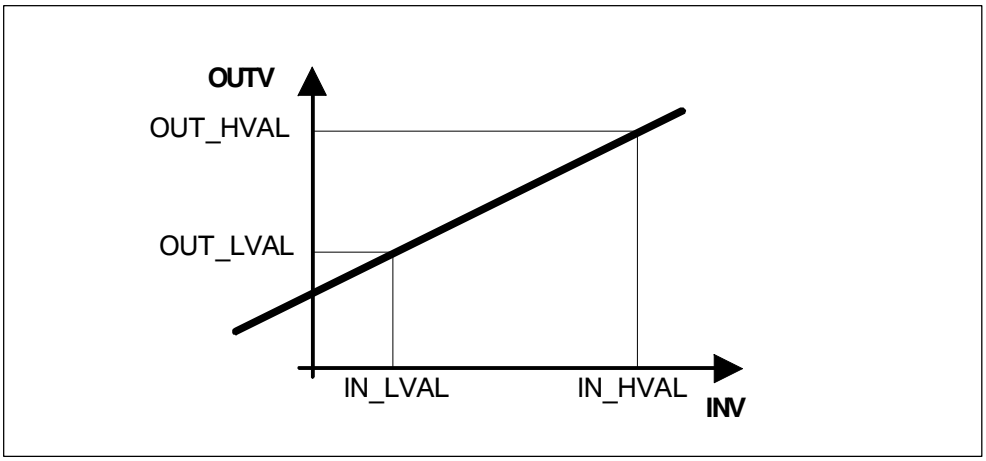


图2-44 规格化曲线

输入参数

下表给出了NORM的输入参数的数据类型和结构。

表2-38 NORM的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	IN_HVAL	物理输入数值高	技术范围 > IN_LVAL	100.0
REAL	OUT_HVAL	物理输出值高	技术范围 > OUT_LVAL	100.0
REAL	IN_LVAL	物理输入值低	技术范围 < IN_HVAL	0.0
REAL	OUT_LVAL	物理输出值低	技术范围 < OUT_HVAL	0.0

输出参数

下表给出了NORM的输出参数的数据类型和结构。

表2-39 NORM的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

完全重启动

块没有完全重启动例行程序。

正常操作

使用规格化曲线，将输入变量INV转换成输出变量OUTV。规格化曲线由点IN_HVAL、IN_LVAL、OUT_HVAL和OUT_LVAL定义。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.18 OVERRIDE: 倍率控制

应用

实现倍率控制需要该块。

方框图

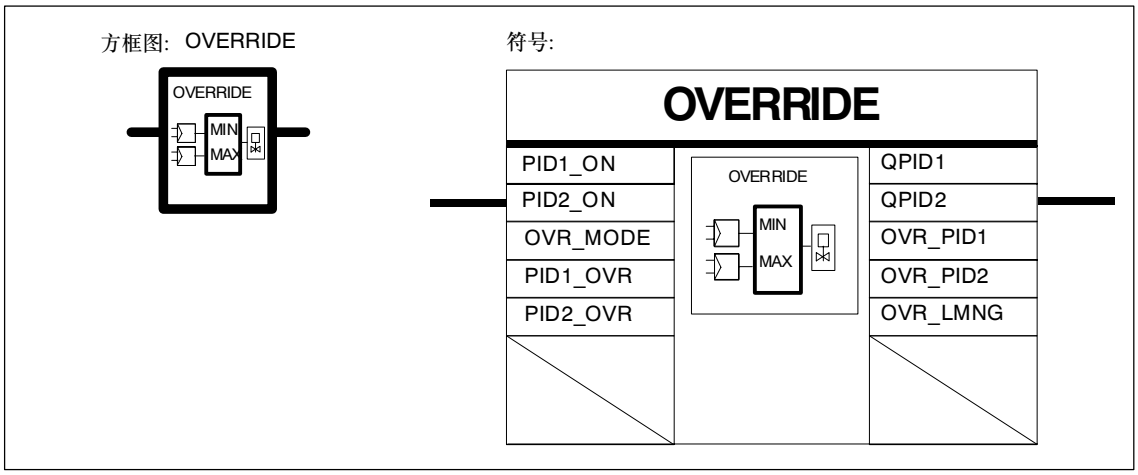


图2-45 OVERRIDE，方框图和符号

功能描述

两个PID控制器连接到一个执行器。两个PID控制器中的最大者(OVR_MODE = FALSE)或最小者(OVR_MODE = TRUE)应用到执行器上。为此，通过OVERRIDE块，使用调节值块LMNGEN_C或LMNGEN_S中的某一个连接两个PID块。

开关PID1_ON、PID2_ON和OVR_MOD的真值表如下所示。

PID1_ON	PID2_ON	OVR_MOD	功能
1	0		仅PID1算法有效
0	1		仅PID2算法有效
0	0	0	应用PID1和PID2的最大者
1	1	0	应用PID1和PID2的最大者
0	0	1	应用PID1和PID2的最小者
1	1	1	应用PID1和PID2的最小者

功能的处理与此信号状态无关。

在输出QPID1和QPID2上指示当前哪一个PID算法在使用中。

输入参数

下表给出了OVERRIDE的输入参数的数据类型和结构。

表2-40 OVERRIDE的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
BOOL	PID1_ON	PID控制器1打开		FALSE
BOOL	PID2_ON	PID控制器2打开		FALSE
BOOL	OVR_MODE	倍率模式FALSE = 最大，TRUE = 最小		FALSE
STRUC	PID1_OVR	PID-LMNGEN接口		
STRUC	PID2_OVR	PID-LMNGEN接口		

输出参数

下表给出了OVERRIDE的输出参数的数据类型和结构。

表2-41 OVERRIDE的输出参数

数据类型	参数	注释	缺省值
BOOL	QPID1	PID控制器1活动	
BOOL	QPID2	PID控制器2活动	
STRUC	OVR_LMNG	PID-LMNGEN接口	

完全重启动

块没有完全重启动例行程序。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

实例

图2-46给出了倍率控制的连接实例。

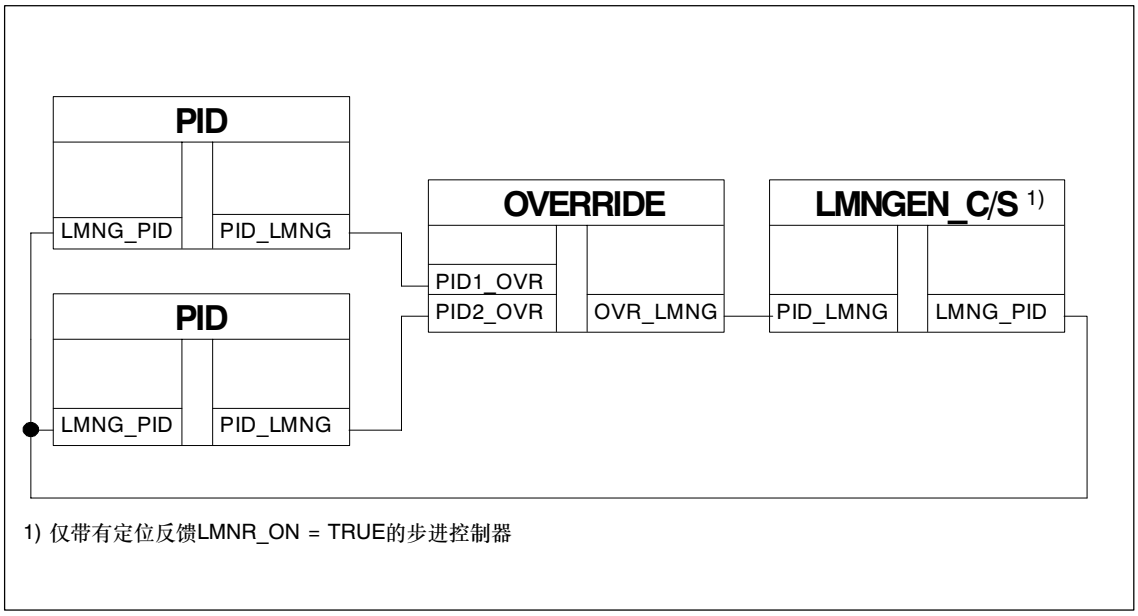


图2-46　　倍率控制连接实例

2.1.19 PARA_CTL: 参数控制

应用

该块用在带有参数切换的控制器结构中；在这种情况下，不同的运行范围需要不同的优化参数。

方框图

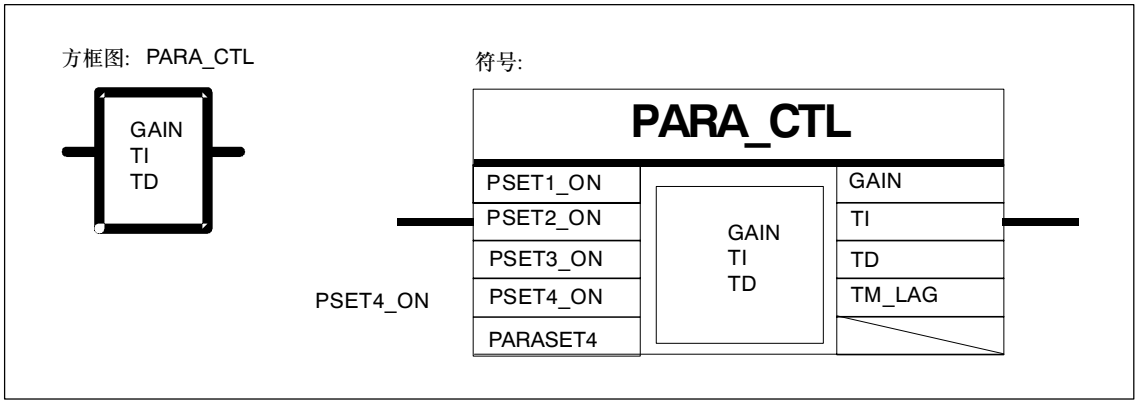


图2-47 PARA_CTL，方框图和符号

功能描述

可以将多个控制器参数集(GAIN、TI、TD和TM_LAG)传送到一个PID控制器。

通过开关PSET1_ON ... PSET4_ON，可以将四组参数设置中的一组应用到输出GAIN、TI、TD和TM_LAG。

```
if   PSET1_ON = 1   then   GAIN = PARASET1.GAIN
                           TI = PARASET1.TR
                           TD = PARASET1.TD
                           TM_LAG = PARASET1.TM_LAG
if   PSET2_ON = 1   then   GAIN = PARASET2.GAIN
                           TI = PARASET2.TR
                           TD = PARASET2.TD
                           TM_LAG = PARASET2.TM_LAG
if   PSET3_ON = 1   then   GAIN = PARASET3.GAIN
                           TI = PARASET3.TR
                           TD = PARASET3.TD
                           TM_LAG = PARASET3.TM_LAG
```

```
if      PSET4_ON = 1    then    GAIN = PARASET4.GAIN
                                TI = PARASET4.TR
                                TD = PARASET4.TD
                                TM_LAG = PARASET4.TM_LAG
```

如果设置了2个或多个开关，则具有最小参数号的开关具有最高优先级。如果没有设置任何开关，则传送第一组参数设置。

输入参数

下表给出了PARA_CTL的输入参数的数据类型和结构。

表2-42　PARA_CTL的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
BOOL	PSET1_ON	设置号码1		FALSE
BOOL	PSET2_ON	设置号码2		FALSE
BOOL	PSET3_ON	设置号码3		FALSE
BOOL	PSET4_ON	设置号码4		FALSE
STRUCT	PARASET1	参数集1		
STRUCT	PARASET4	参数集4		

输出参数

下表给出了PARA_CTL的输出参数的数据类型和结构。

表2-43　PARA_CTL的输出参数

数据类型	参数	注释	缺省值
REAL	GAIN	比例系数	1.0
TIME	TI	积分作用暂停	10秒
TIME	TD	微分时间	5秒
TIME	TM_LAG	时间延迟	2秒

控制器参数集

下表给出了数据类型和参数。

表2-44

数据类型	参数	注释	缺省值	缺省值
REAL	PARASET1.GAIN	比例增益1		1.0
TIME	PARASET1.TI	复位时间1		T#10s
TIME	PARASET1.TD	微分时间1		T#5s
TIME	PARASET1.TM_LAG	时间延迟1		T#2s
...	...		...	...
REAL	PARASET4.GAIN	比例增益4		1.0
TIME	PARASET4.TI	复位时间4		T#10s
TIME	PARASET4.TD	微分时间4		T#5s
TIME	PARASET4.TM_LAG	时间延迟4		T#2s

完全重启动

块没有完全重启动例行程序。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

实例

如果需要在参数集之间产生无阶跃的切换，则PI块和PARA_CTL块之间的GAIN参数(HROC_LIM)中必须包含斜坡变化率(参见图 2-48)。

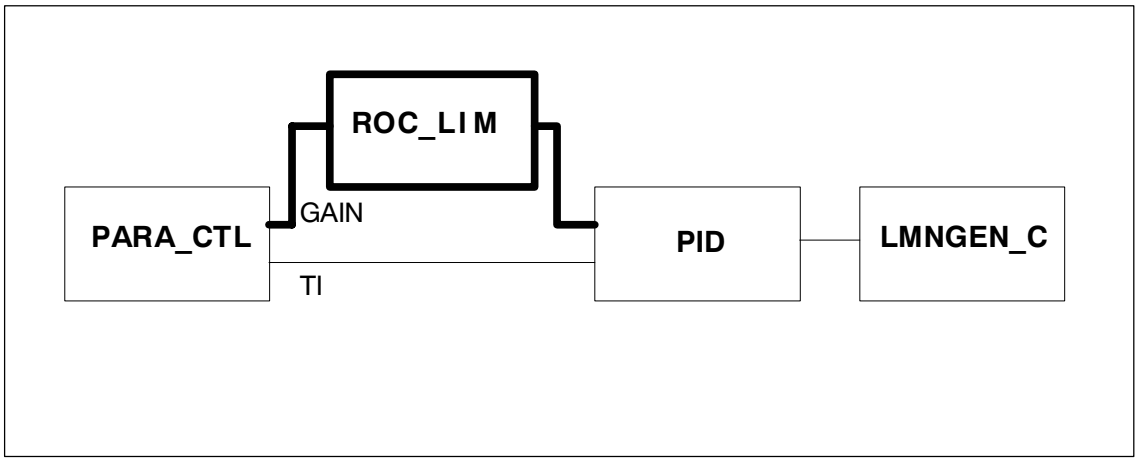


图2-48　　参数集之间的无阶跃切换

2.1.20 PID: PID算法

应用

该块包含了用于创建下列控制器类型的PID算法:

- 连续PID控制器:
  PID + LMNGEN_C
- 用于比例执行器的PID脉冲控制器:
  PID+LMNGEN_C + PULSEGEN
- 用于积分执行器的PID步进控制器:
  PID + LMNGEN_S

方框图

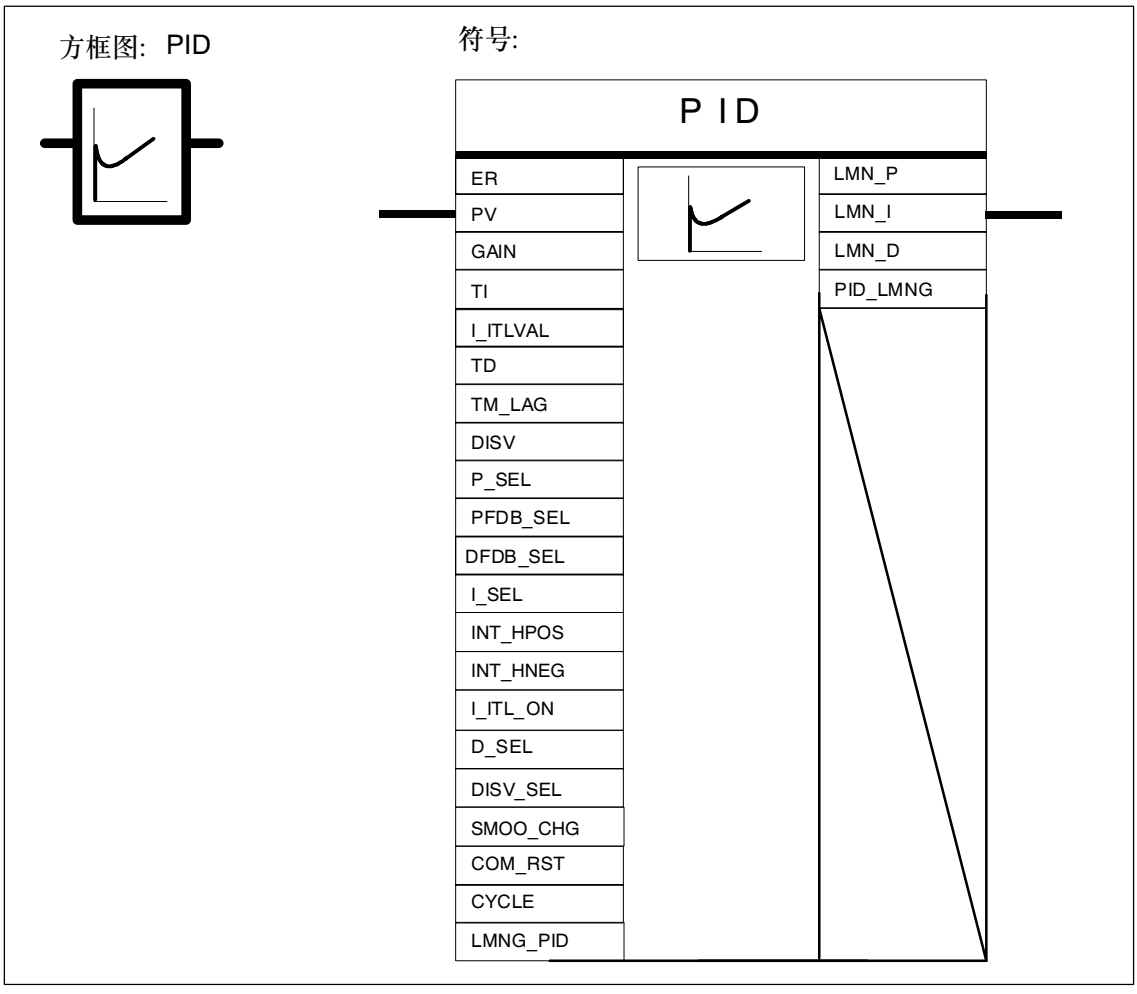


图2-49 PID，方框图和符号

功能描述

该块实现了PID算法。它被设计成纯粹的平行结构，只能单独作为定位算法使用。可以分别单独激活或取消激活比例、积分和微分作用。这使对P、PI、PD和PID控制器进行组态成为可能。

P和D动作的计算可以位于反馈路径中。通过决定P和D动作移到反馈路径中，如果在相同速度上补偿干扰，在控制器就不会引起任何突然变化。在通常情况下，并不需要使用设定值积分器，以避免在设定值中产生阶跃变化。

由于PID块位于周期性中断优先级中，该优先级的循环时间已调整为与主系统时间常数相匹配，这样用于处理执行器信号(LMNGEN_C和LMNGEN_S)的块便可以放在更快的周期性中断优先级中。

输入参数

下表给出了PID的输入参数的数据类型和结构。

表2-45　PID的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	ER	出错信号	技术取值范围	0.0
REAL	PV	过程变量(反馈中的比例或微分动作)	技术取值范围	0.0
REAL	GAIN	比例系数		1.0
TIME	TI	积分作用暂停		T#20s
REAL	I_ITLVAL	积分作用的初始化值	技术取值范围	0.0
TIME	TD	微分时间		T#10s
TIME	TM_LAG	微分作用的时间延迟		T#2s
REAL	DISV	干扰变量	技术取值范围	0.0
BOOL	P_SEL	比例作用打开		TRUE
BOOL	PFDB_SEL	反馈路径中的比例动作打开		FALSE
BOOL	DFDB_SEL	反馈路径中的微分动作打开		FALSE
BOOL	I_SEL	积分作用打开		TRUE
BOOL	INT_HPOS	保持正方向上的积分作用		FALSE
BOOL	INT_HNEG	保持负方向上的积分作用		FALSE
BOOL	I_ITL_ON	积分作用初始化		FALSE
BOOL	D_SEL	微分作用打开		FALSE
BOOL	DISV_SEL	干扰变量打开		TRUE

表2-45 PID的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
BOOL	SMOO_CHG	从手动模式平衡切换到自动模式		TRUE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	≥ 1毫秒	T#1s
STRUC	PID_LMNG	PID-LMNGEN接口		

输出参数

下表给出了PID的输出参数的数据类型和结构。

表2-46 PID的输出参数

数据类型	参数	注释	缺省值
REAL	LMN_P	比例分量	0.0
REAL	LMN_I	积分分量	0.0
REAL	LMN_D	微分分量	0.0
STRUC	PID_LMNG	PID-LMNGEN接口	

完全重启动

在完全重启动期间，所有输出和输入/输出参数都设置为零。

正常操作

除了正常运行模式，该块还具有下列模式:

模式	P_SEL	I_SEL	D_SEL
P控制器	TRUE	TRUE或FALSE	FALSE
PI控制器	TRUE	TRUE	FALSE
PD控制器	TRUE	FALSE	TRUE
PID控制器	TRUE	TRUE	TRUE

- **P控制器**

此处，关闭了D动作。对于I动作，可以使用初始值I_ITLVAL来指定工作点(I_SEL = TRUE和I_ITL_ON = TRUE)。如果工作点一直保持为恒零，则也可以关闭I动作。

传送函数

误差信号ER与增益GAIN相乘。

步响应

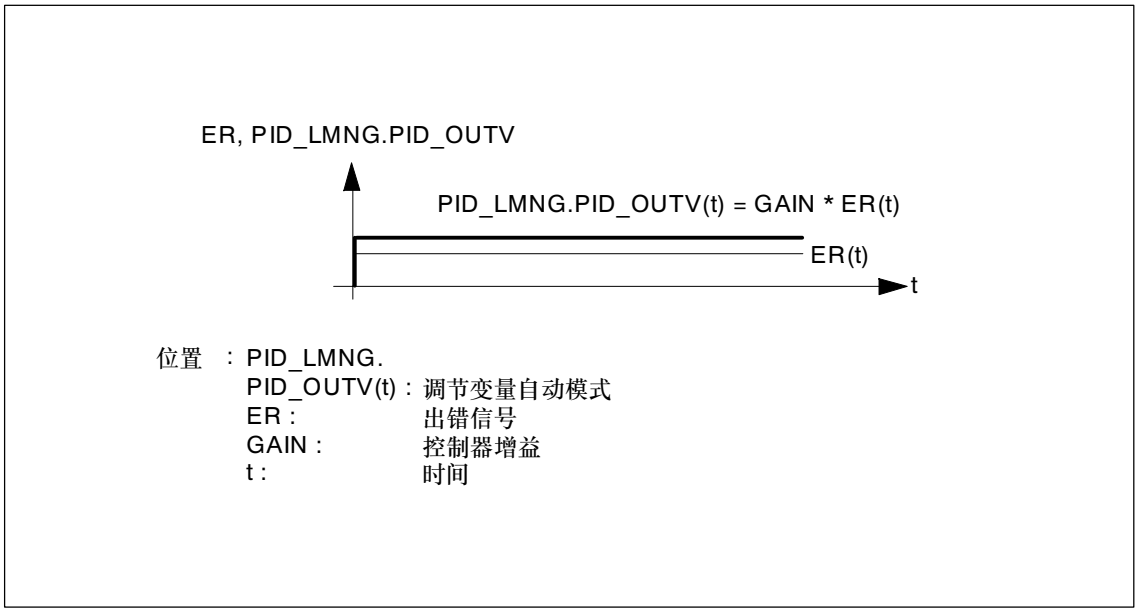


图2-50 P控制器的阶跃响应

- **PI控制器**

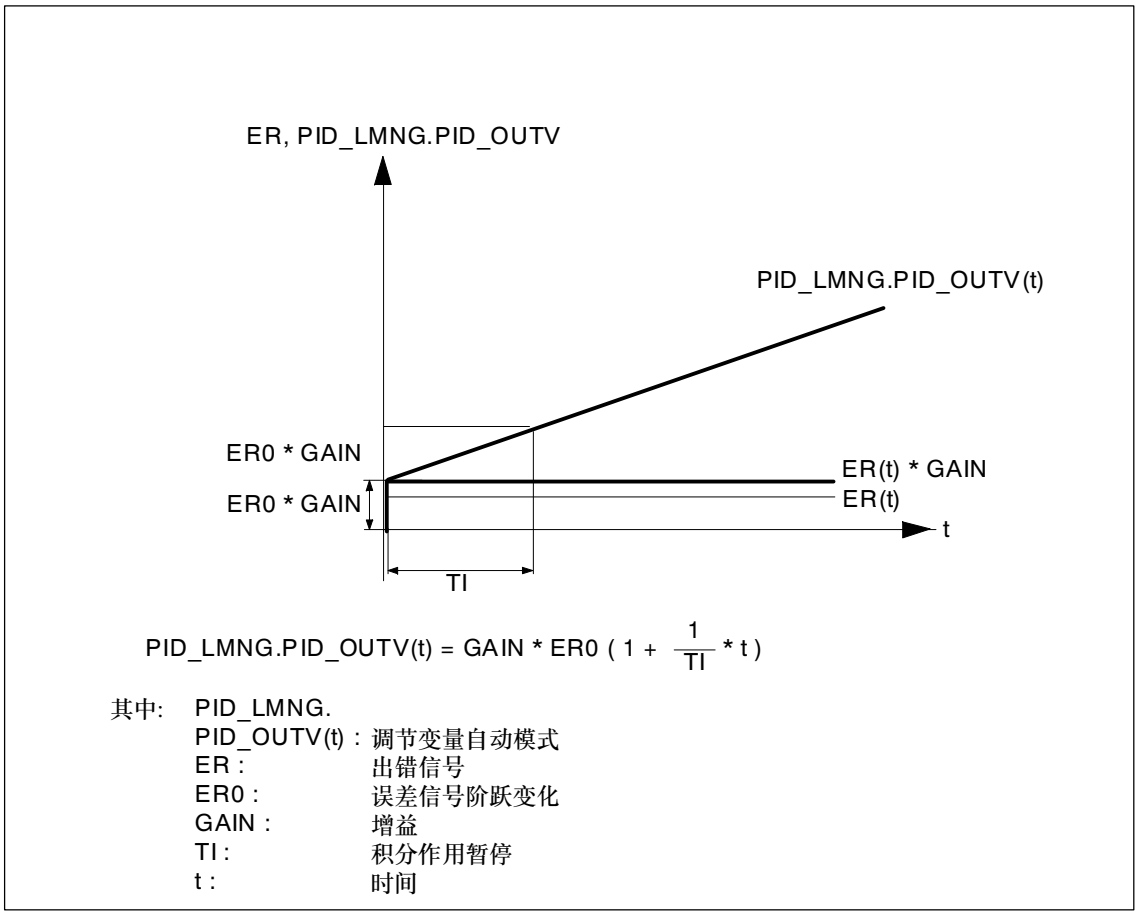
此处，关闭了D动作。

传送函数

误差信号ER被相乘，并被积分。Laplace范围内的传送函数是:

PID_LMNG.PID_OUTV(s) / ER(s) = GAIN* (1 + 1 / (TI*s))

步响应



其中: PID_LMNG.
PID_OUTV(t)：调节变量自动模式
ER：出错信号
ER0：误差信号阶跃变化
GAIN：增益
TI：积分作用暂停
t：时间

图2-51 PI控制器的阶跃响应

- **PD控制器**

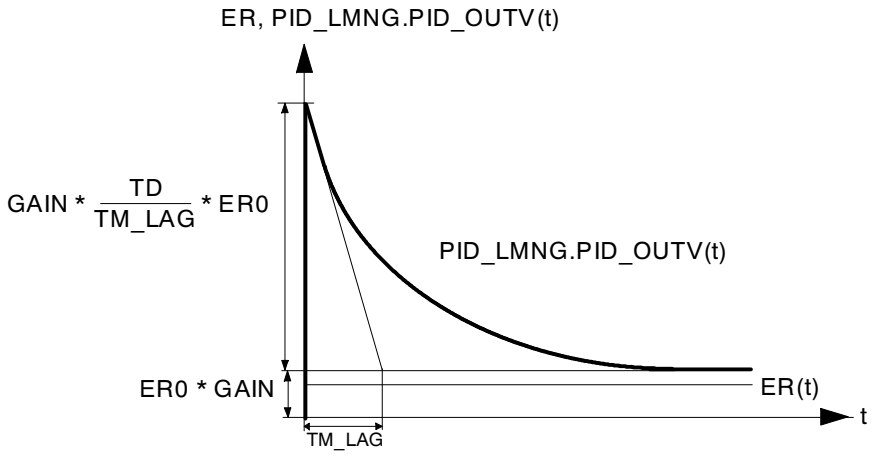
此处，关闭了I动作。

传送函数

误差信号ER被相乘，并被微分。Laplace范围内的传送函数是:

PID_LMNG.PID_OUTV(s) / ER(s) = GAIN* (1 + TD*s / (1 + TM_LAG*s))

步响应



$$PID\_LMNG.PID\_OUTV(t) = GAIN * ER0 \left(1 + \frac{TD}{TM\_LAG} * e^{-\frac{t}{TM\_LAG}}\right)$$

其中:

PID_LMNG.	
PID_OUTV(t)：	调节变量自动模式
ER：	出错信号
ER0：	误差信号阶跃变化
GAIN：	控制器增益
TD：	微分时间
TM_LAG：	时间延迟
t：	时间

图2-52　PD控制器的阶跃响应

- **PID控制器**

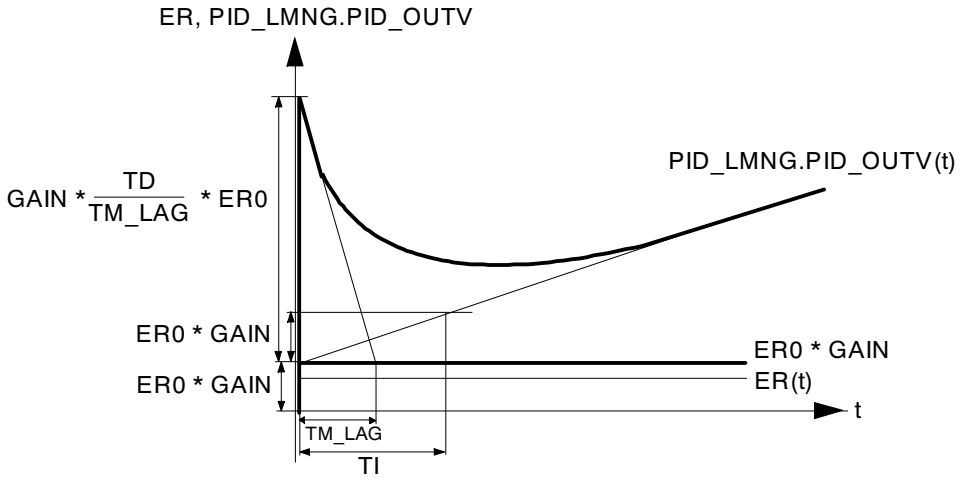
P、I和D动作都被激活。

传送函数

误差信号ER被相乘，并并积分和微分。Laplace范围内的传送函数是:

PID_LMNG.PID_OUTV(s) / ER(s) = GAIN* (1 + 1 / (TI*s) + TD*s / (1 + TM_LAG*s))

步响应



$$PID\_LMNG.PID\_OUTV(t) = GAIN * ER0 \left(1 + \frac{1}{TI} * t + \frac{TD}{TM\_LAG} * e^{-\frac{t}{TM\_LAG}} \right)$$

位置 ：PID_LMNG.
PID_OUTV(t)：调节变量自动模式
ER：出错信号
ER0：误差信号阶跃变化
GAIN：控制器增益
TI：积分作用暂停
TD：微分时间
TM_LAG：时间延迟
t：时间

图2-53 PID控制器的阶跃响应

方框图

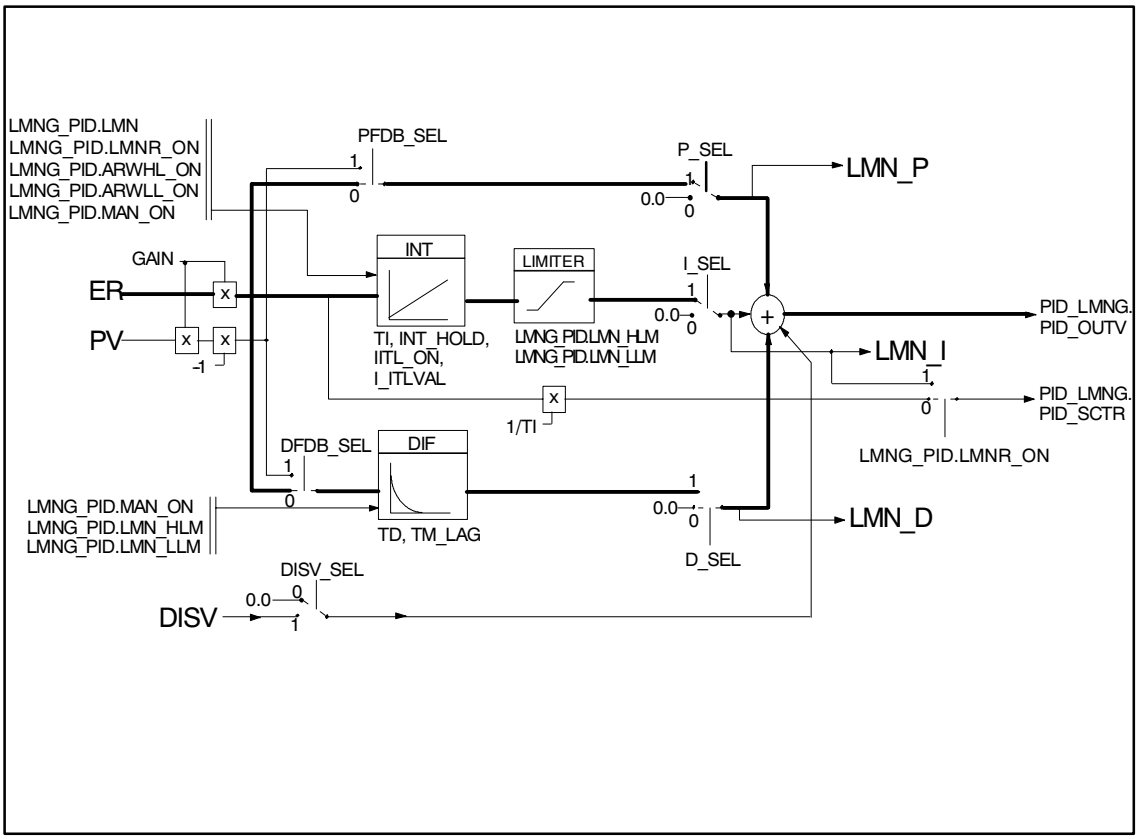


图2-54 PID算法的方框图

比例作用

可以通过P_SEL开关激活和取消激活比例作用。可以使用PFDB_SEL开关将比例作用包含在反馈路径中。这种情况下，过程变量PV用作P动作的输入。P动作表示误差信号ER (如果P动作位于反馈路径中，则是过程变量PV)和比例增益GAIN的乘积。

积分作用

可以通过I_SEL开关激活和取消激活积分动作。当取消激活I动作时，I动作和积分器的内部存储器都被设置为零。可以使用INT_HOLD冻结I动作。复位时间由复位时间常数TI决定。用户还可以为积分器分配自己的数值。 通过I_ITL_ON开关，将输入I_ITLVAL上的值传送打开积分器。如果限制了调节值的范围，则I动作保持在旧值处(反复位结束)。

微分作用

可以通过D_SEL开关激活和取消激活D动作。可以使用DFDB_SEL开关将D动作包含在反馈路径中。这样，D动作的输入值便是负的过程变量PV。时间响应由微分动作时间TD决定。在微分单元内集成了一个一阶时间延迟环节。在TM_LAG上输入时间延迟。

特别是在激活了D动作的快速系统中，很有可能会发生系统无法接受的干扰波动。这种情况下，可以通过在D动作中集成时间延迟来改善控制质量。通常，很小的TM_LAG便已足够。

前馈控制

可以在调节值上附加连接一个干扰值DISV。可以使用DISV_SEL开关激活和取消激活此干扰值。

块内部限制

- 复位时间向下被限制为采样时间的一半。
- 微分时间向下限制为采样时间。
- 时间延迟向下限制为采样时间的一半。

TI_{intern} = CYCLE/2	when TI < CYCLE/2
TD_{intern} = CYCLE	when TD < CYCLE
TM_LAG_{intern} = CYCLE/2	when TM_LAG < CYCLE/2

在此块中，其它输入参数的数值并没有限制；系统不会检查参数。

2.1.21 PULSEGEN：脉冲发生器

应用

该块用于构造带有用于比例执行器的脉冲输出的PID控制器。这就使用户可以实现带有脉宽调制的三步和两步控制器。

方框图

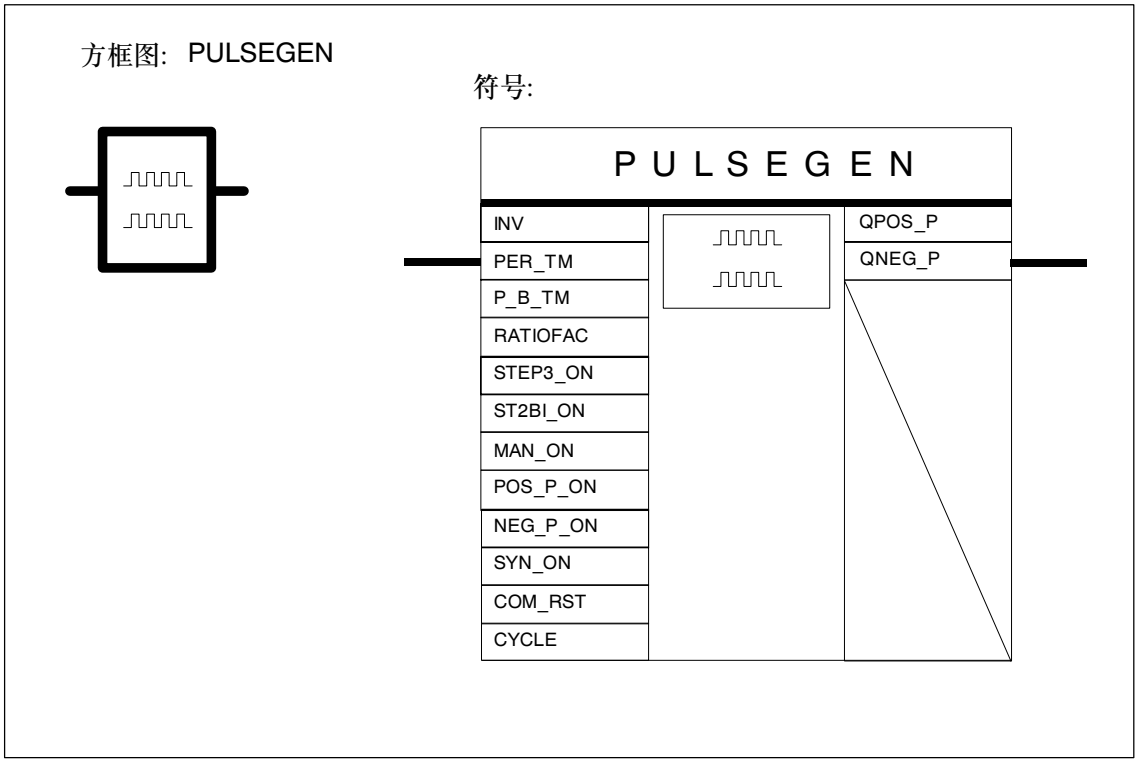


图2-55 PULSEGEN，方框图和符号

功能描述

通过使用脉宽调制，该块以恒定周期PER_TM将输入变量INV（= PID控制器的LMN)转换成脉冲序列。该周期对应于更新输入变量所使用的周期时间间隔。在每个周期内，脉冲的持续时间和输入变量成比例。因此，30%的输入变量意味着：持续时间为0.3 *周期时间的正脉冲，而在另外0.7 *周期时间内无脉冲信号。在每个周期的开始处，重新计算脉冲持续时间。

图2-56说明了脉宽调制。

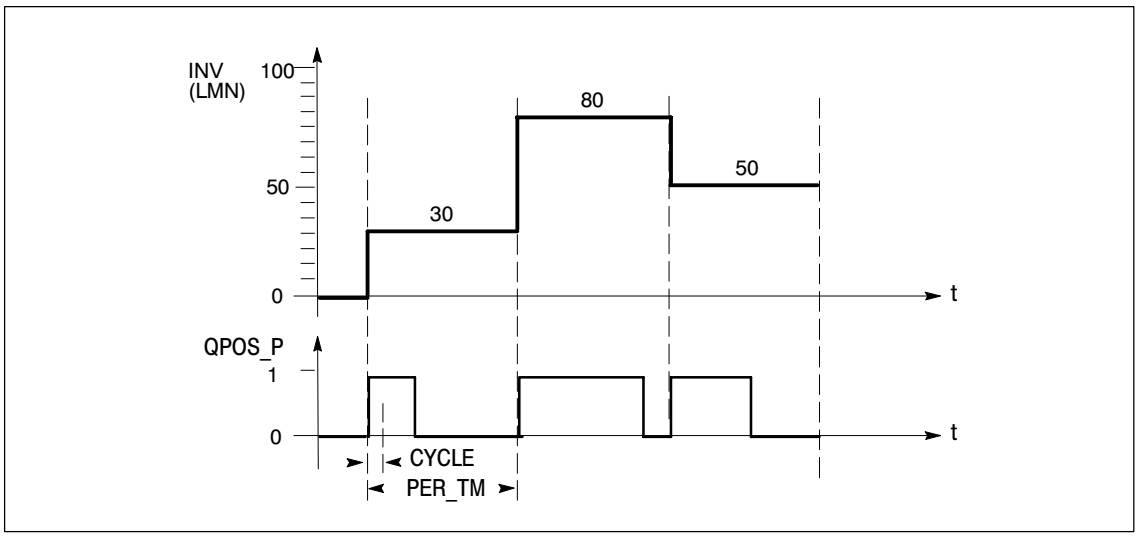


图2-56　　脉冲宽度调制

为了减小执行器的磨损，可以设置最小脉冲/断开时间。

在三步控制模式中，可以使用比率因子为加热和制冷补偿不同的时间常数。

该块通常和连续控制器一起使用。

- **两步或三步PID控制器:**
  PID+LMNGEN_C + PULSEGEN

图2-57给出了PID脉冲控制器的连接。

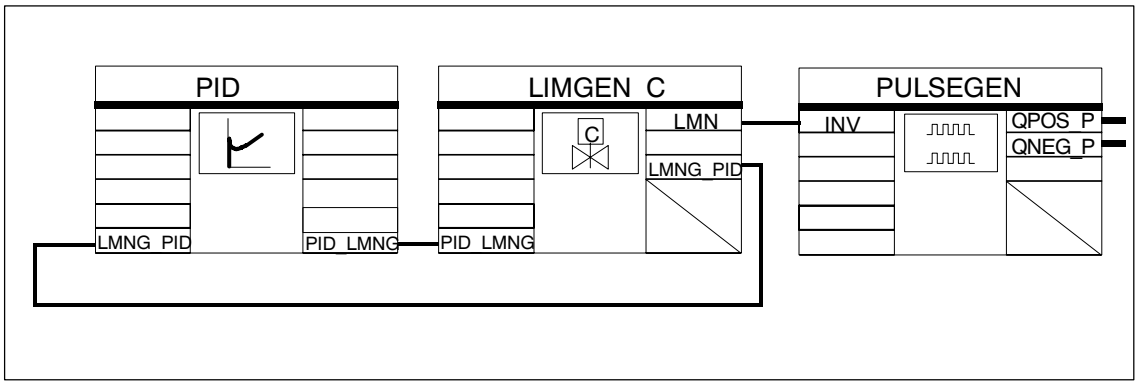


图2-57　　PID脉冲控制器的连接

输入参数

下表给出了PULSEGEN的输入参数的数据类型和结构。

表2-47 PULSEGEN的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	-100,0..100,0 (%)	0.0
TIME	PER_TM	周期时间	PER_TM >= 20*CYCLE	T#1s
TIME	P_B_TM	最小脉冲/断开时间	P_B_TM >= CYCLE	T#50ms
REAL	RATIOFAC	比率因子	0,1..10,0 (无维数)	1.0
BOOL	STEP3_ON	三步控制开启		TRUE
BOOL	ST2BI_ON	双极调节值范围的两步控制开启		FALSE
BOOL	MAN_ON	手动模式开启		FALSE
BOOL	POS_P_ON	正脉冲开启		FALSE
BOOL	NEG_P_ON	负脉冲开启		FALSE
BOOL	SYN_ON	同步开启		TRUE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	CYCLE >= 1ms	T#10ms

输出参数

下表给出了PULSEGEN的输出参数的数据类型和结构。

表2-48 PULSEGEN的输出参数

数据类型	参数	注释	缺省值
BOOL	QPOS_P	输出正脉冲	FALSE
BOOL	QNEG_P	输出负脉冲	FALSE

完全重启动

在完全重启动期间，所有的信号输出都被设置成零。

脉冲生成的精度

在CPU中实现此功能需要确定在控制器输出的时间段内，n倍二进制信号在周期内的各个点处的当前状态。n的数值越高，脉宽调制的精度越高。

由于连续PID控制器(PID-LMNGEN_C)位于慢周期性中断优先级中，其循环时间与主系统时间常数相匹配，因此必须将PULSEGEN块放置在较快的周期性中断优先级中。周期性中断优先级越快，可以输出的调节值精度就越高。通过使用快100倍的周期性中断优先级，可以获得的精度为调节值范围的1%。

可以自动将脉冲输出与更新输入变量INV (例如，PID_C)的块同步。这样就可确保在尽可能最短的时间内，以脉冲形式输出正在变化的输入变量。

自动同步

可以自动同步脉冲输出与控制器FB。这确保了系统能够在尽可能最短的时间内，以按比例修改的脉宽的二进制信号形式，输出调节变量LMN(t)的数值变化。

脉冲发生器按照与周期PER_TM相对应的时间间隔，判断输入变量INV。然而，由于INV是在较慢的周期性中断优先级中计算的，因此脉冲发生器应该在更新INV之后尽快将离散值转换成脉冲信号。为此，该块可以按照下述方式来同步它自己的周期的起始点:

如果INV已发生变化且块调用不是发生在该周期时间段的前两个或后两个调用周期内，则执行同步。重新计算脉宽，并且在下一个周期中，开始使用新周期时间段输出。

周期PER_TM必须与控制器的采样时间CYCLE相对应。

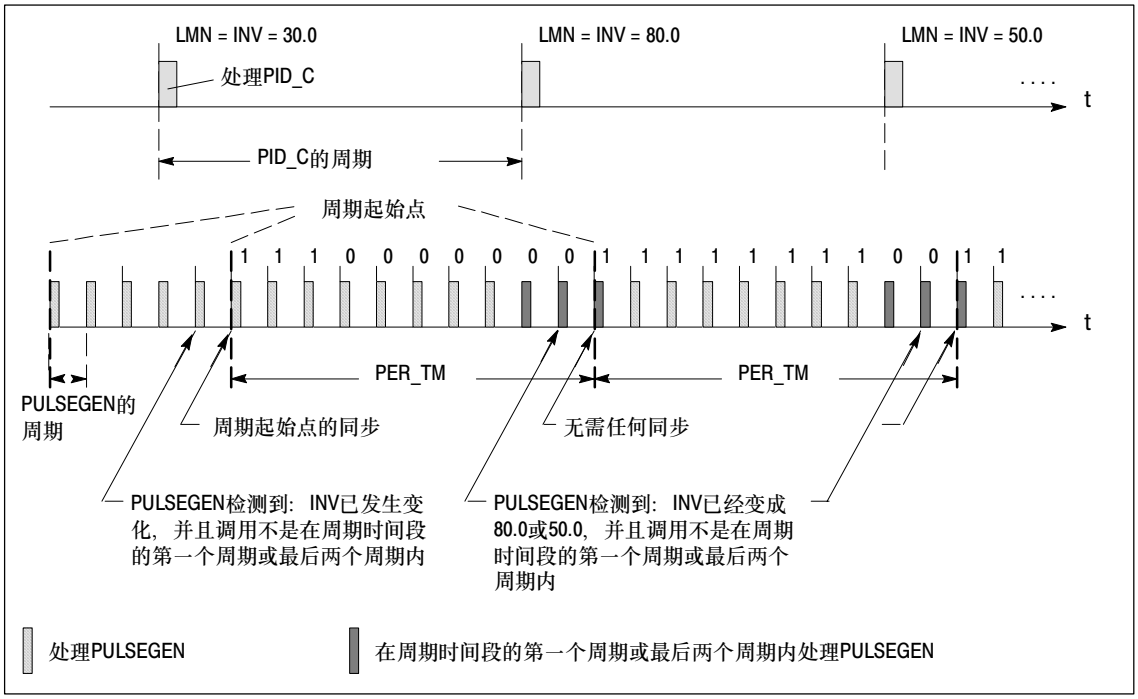


图2-58 周期时间段起始点同步

可以在输入 “SYN_ON” (= FALSE)上关闭自动同步。

注意

在同步之后的新周期时间段的开始，脉冲信号所模拟的INV的旧值(也就是LMN的值)不会太精确。

具有脉冲输出的控制器的模式

根据脉冲发生器的参数设置，将会配置带有三步输出的PID控制器，或带有双极或单极两步输出的PID控制器。下表给出了可能模式的开关组合。

开关 / 模式	MAN_ON	STEP3_ON	ST2BI_ON
三位控制器	FALSE	TRUE	ANY
带有双极输出的两步控制器 范围(-100 %到100 %)	FALSE	FALSE	TRUE
带有单极输出的两步控制器 范围(0 %到100 %)	FALSE	FALSE	FALSE
手动模式	TRUE	any	any

三步控制

在三步控制模式中，可以根据执行器和相关过程，为执行器信号生成三个状态，例如: 增加 - 关闭 - 减小，正向 - 停止 - 反向，加热 - 关闭 - 制冷等。根据要控制的过程的要求，将二进制输出信号QPOS_P和QNEG_P的状态分配给执行器的相应运行状态。下表给出了两个实例。

	加热 正向	关闭 停止	冷却 反向
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

计算最小脉冲时间或最小断开时间 P_B_TM可以防止出现特别短的打开时间和关闭时间，这样的短时间会降低开关和执行器的使用寿命。计算脉冲输出所使用的比例特征为自身应用了一个响应阈值。

注意

输入变量LMN的小绝对值可能会生成小于P_B_TM的脉宽，这类绝对值将被抑制。对于会生成大于PER_TM - P_B_TM的脉宽的大输入值，脉宽被设置为100%或-100%。

建议数值P_B_TM $\leq$ 0.1 * PER_TM。

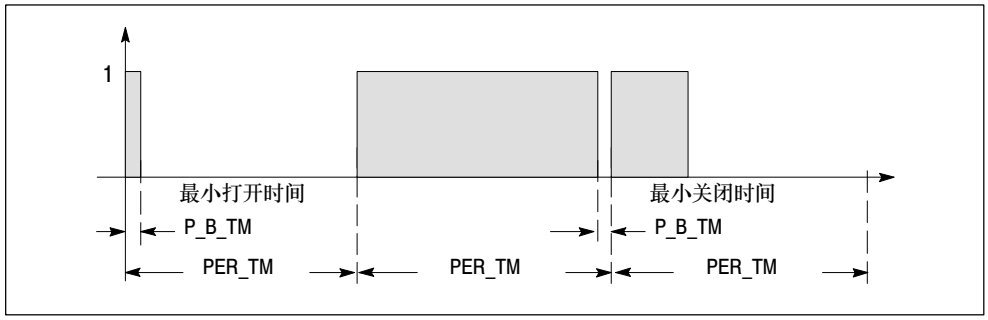


图2-59 脉冲输出如何打开和关闭

正脉冲或负脉冲的持续时间由输入变量(%格式)与周期时间段相乘得来:

$$\text{脉宽} = \frac{\text{INV}}{100} * \text{PER\_TM[s]}$$

抑制最小脉冲时间和最小断开时间会在转换特征曲线的范围开始和范围结束处生成“折线”。(图2-60)。

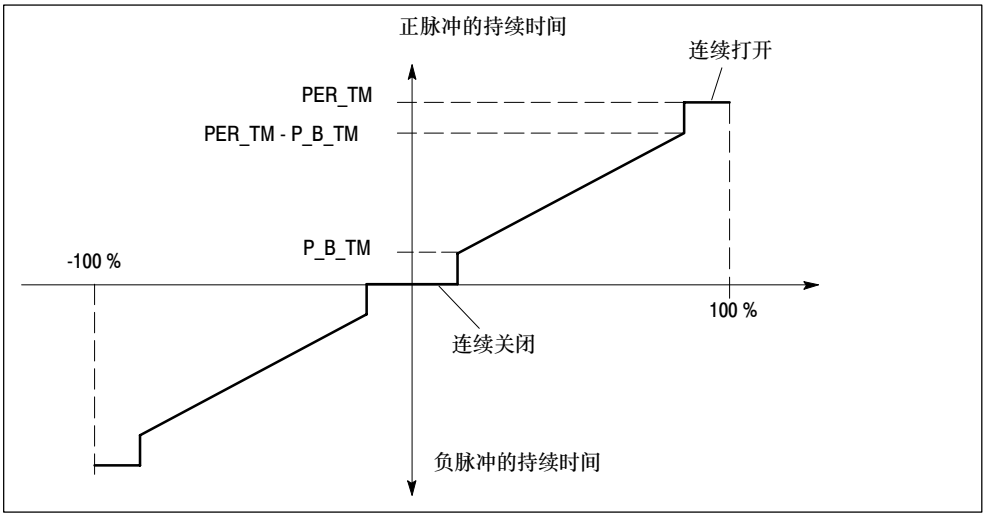


图2-60　　三步控制器的对称曲线 (比率因子 = 1)

不对称的三步控制器

通过比率因子 “RATIOFAC”，可以更改负脉冲持续时间和正脉冲持续时间之间的比率。在热过程中，可以为加热过程和制冷过程补偿不同的时间常数。

输入变量|INV|, 使用相同的绝对值时，如果负脉冲输出上的脉宽必须短于正脉冲上的脉宽，则必须设置小于1的比率因子(图2-61):

正脉冲 > 负脉冲: RATIOFAC < 1

负脉宽: $\frac{INV}{100} * PER\_TM * RATIOFAC$

正脉宽: $\frac{INV}{100} * PER\_TM$

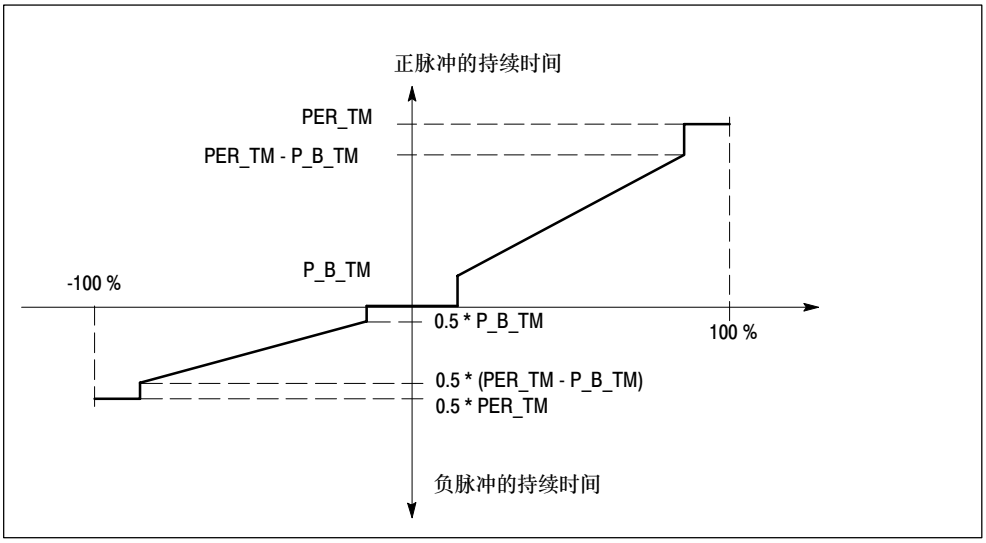


图2-61　　三步控制器的不对称曲线(比率因子 = 0.5)

如果需要相反结果，即对输入变量|INV|, 使用相同的绝对值时，如果正脉冲输出上的脉宽必须短于负脉冲上的脉宽，则必须设置大于1的比率因子:

正脉冲 < 负脉冲: RATIOFAC > 1

负脉宽:
$$\frac{INV}{100} * PER\_TM$$

正脉宽:
$$\frac{INV * PER\_TM}{100 * RATIOFAC}$$

比率因子还会影响最小脉冲时间和最小断开时间。从数学上讲，这意味着对于RATIOFAC < 1的情况，是负脉冲的打开阈值乘以比率因子；对于RATIOFAC > 1的情况，则是正脉冲的打开阈值除以比率因子。

两步控制

在两步控制中，只会将PULSEGEN的正脉冲输出QPOS_P连接到开/关执行机构。根据执行范围，即LMN = -100.0 ... 100.0 % 或LMN = 0.0 % ... 100.0 %，两步控制器有双极或单极执行范围。

在单极模式下，输入变量INV的值只能为0.0到100%之间的数值。

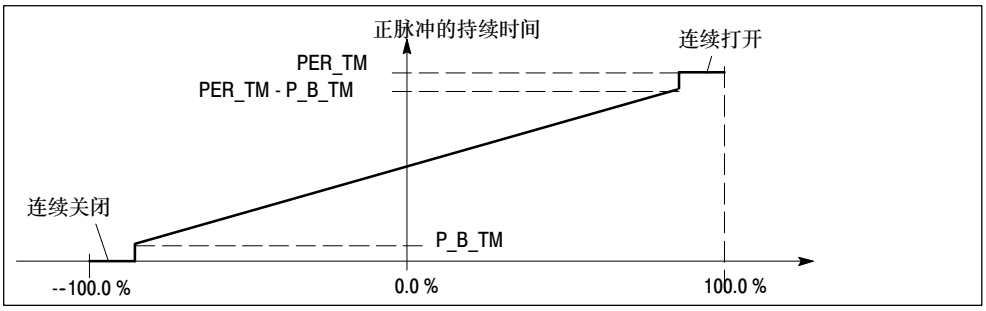


图2-62　　具有双极范围的两步控制器(-100%到100%)

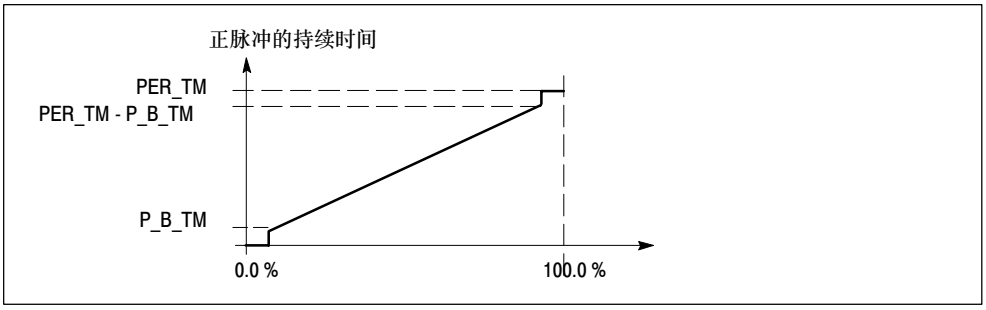


图2-63　　具有单极范围的两步控制器(0%到100%)

如果控制回路中的两步控制器连接需要一个逻辑反转的二进制信号，用于生成控制脉冲，则可以使用QNEG_P上的取反输出信号。

	接通	关闭
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

两步或三步控制中的手动模式

在手动模式(MAN_ON = TRUE)下，可以通过信号POS_P_ON和NEG_P_ON设置三步或两步控制器的输出，而不管INV的状态如何。

	POS_P_ON	NEG_P_ON	QPOS_P	QNEG_P
三位控制器	FALSE	FALSE	FALSE	FALSE
	TRUE	FALSE	TRUE	FALSE
	FALSE	TRUE	FALSE	TRUE
	TRUE	TRUE	FALSE	FALSE
两位控制器	FALSE	any	FALSE	TRUE
	TRUE	any	TRUE	FALSE

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.22 RMP_SOAK: 斜坡保持

应用

斜坡保持主要用作设定值发生器，使控制器在过程运行期间的不同时间点上设置不同的设定值。

方框图

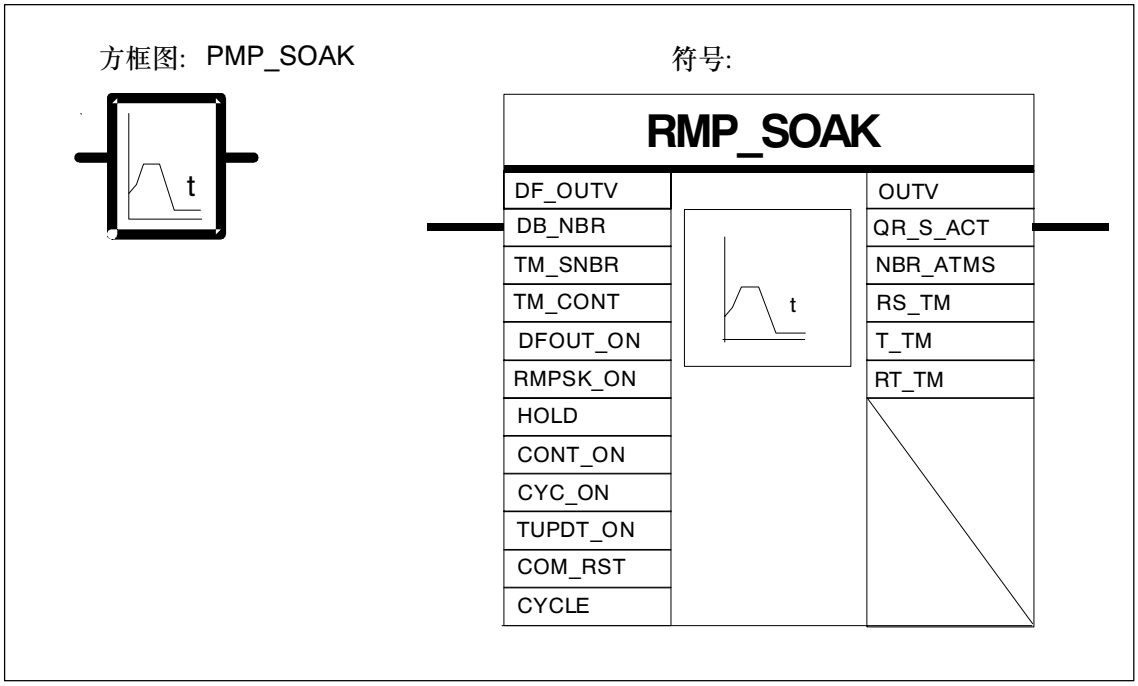


图2-64 RMP_SOAK，方框图和符号

功能描述

该块可用于执行其坐标存储在全局数据块中的曲线。在每个调用周期中，值按照时间顺序输出。值插在坐标之间的时间片内。

可通过控制输入选择下列模式:

- 斜坡保持打开
- 缺省输出变量
- 暂停处理
- 设置继续执行的时间片号和时间
- 重复打开(周期性模式)
- 总时间和剩余总时间更新打开

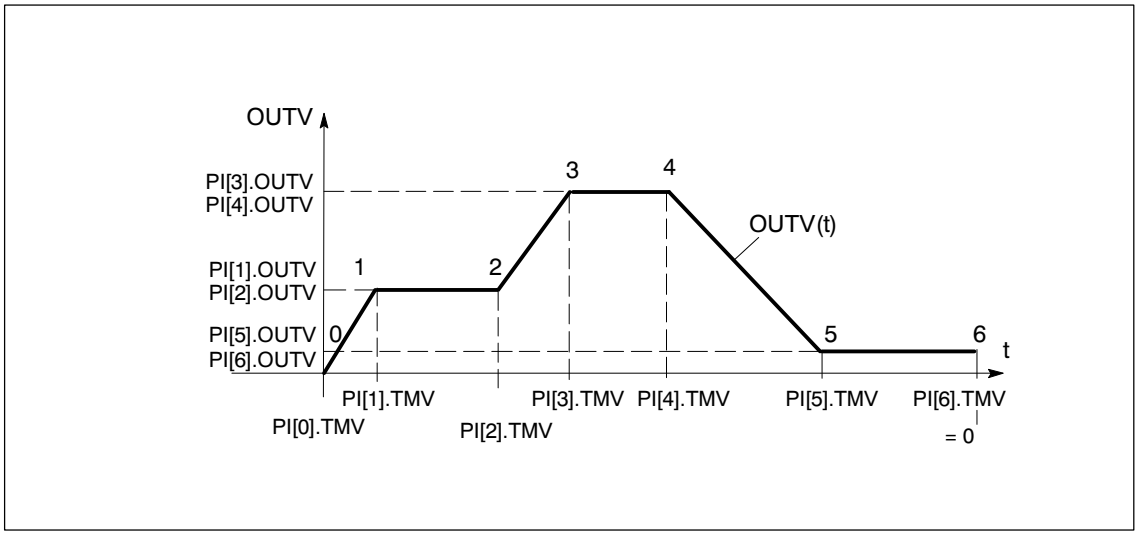


图2-65 带有起始点和6个坐标的斜坡保持实例

对于n个坐标，坐标n的时间值 = 0 ms (处理结束点)。

注意

该块不会检查编号为DB_NBR的共享数据块是否存在，以及参数DB_NBR.NBR_PTS(时间片数)与数据块长度是否匹配。如果参数分配不正确，则CPU切换到STOP，同时报告内部系统错误。

输入参数

下表列出了RMP_SOAK的输入参数的数据类型和结构。

表2-49 RMP_SOAK的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
BLOCK_DB	DB_NBR	数据块编号	取决于CPU	DB 1
INT	TM_SNBR	时间片编号	0 - 255	0
TIME	TM_CONT	继续执行的时间点(即时)	技术取值范围	T#0s
BOOL	DFOUT_ON	缺省输出变量打开		FALSE
BOOL	RMPSK_ON	斜坡保持打开		FALSE
BOOL	HOLD	暂停输出变量		FALSE
BOOL	CONT_ON	继续		FALSE
BOOL	CYC_ON	周期性重复打开		FALSE
BOOL	TUPDT_ON	总时间更新打开		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间	技术取值范围 ≥ 1ms	T#1s

输出参数

下表列出了RMP_SOAK的输出参数的数据类型和结构。

表2-50 RMP_SOAK的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0
BOOL	QR_S_ACT	斜坡保持激活	FALSE
INT	NBR_ATMS	活动时间片的编号	0
TIME	RS_TM	剩余时间片时间	T#0s
TIME	T_TM	总时间	T#0s
TIME	RT_TM	剩余总时间	T#0s

坐标(点)和点数NBRPTS存储在共享数据块(DB_NBR)中。输出从点0开始，在点NBR_PTS结束。

共享数据

(缺省带有起始点和4个点)

表2-51

数据类型	参数 DB_NBR	注释	允许使用的数值	缺省值
INT	NBR_PTS	点数 - 1	1 - 255	4
REAL	PI[0].OUTV	输出变量[0]	技术取值范围	0.0
TIME	PI[0].TMV	输出时间值[0]		T#1s
REAL	PI[1].OUTV	输出变量[1]	技术取值范围	0.0
TIME	PI[1].TMV	输出时间值[1]		T#1s
REAL	PI[2].OUTV	输出变量[2]	技术取值范围	0.0
TIME	PI[2].TMV	输出时间值[2]		T#1s
REAL	PI[3].OUTV	输出变量[3]	技术取值范围	0.0
TIME	PI[3].TMV	输出时间值[3]		T#1s
REAL	PI[4].OUTV	输出变量[4]	技术取值范围	0.0
TIME	PI[4].TMV	输出时间值[4]	0毫秒	T#0s

完全重启动

在完全重启动期间，输出OUTV设置为0.0。如果设置了DFOUT_ON=TRUE，则输出DF_OUTV。统计点0到NBRPTS之间的时间片(0到NBRPTS-1)的总数，可以从T_TM上获得该值。复位输出QR_S_ACT，并将输出NBR_ATMS和RS_TM设置为0。

正常操作

- 坐标参数NBR_PTS、PI[i].TMV和PI[i].OUTV存储在共享数据块中。
- 参数PI[i].TMV必须以TIME格式指定。
- 下面的示意图说明了点值和时间片的计数方式:

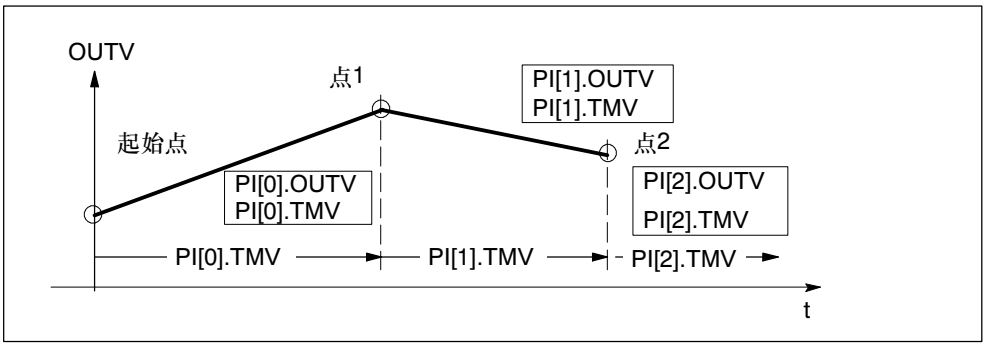


图2-66　　计数坐标和时间片

在正常操作中，斜坡保持根据下列函数进行插值，其中0 ≤ n < (NBR_PTS - 1):

$$OUTV(t) = PI[n+1].OUTV - \frac{RS\_TM}{PI[n].TMV}\,(PI[n+1].OUTV - PI[n].OUTV)$$

斜坡保持(Ramp Soak)的模式

为了影响控制输入，可以实现下列斜坡保持状态和模式:

1. 斜坡保持打开仅一次运行
2. 斜坡保持输出上的缺省值
3. 周期性的斜坡保持模式打开
4. 暂停斜坡保持
5. 要继续的时间片数和时间(重新定义剩余时间片RS_TM和点号TM_SNBR)
6. 更新总时间和剩余总时间

模式

当设置了其中一种模式时，将会应用如下表中所示的控制输入值:

表2-52　斜坡保持(Ramp Soak)的模式(RMP_SOAK)

模式	RMPSK_ON	DFOUT_ON	RMP_HOLD	CONT_ON	CYC_ON	TUPDT_ON	输出信号OUTV
1. 斜坡保持(Ramp Soak)打开	TRUE	FALSE	FALSE		FALSE		OUTV(t) 处理完成时保持的最终值。
2. 缺省输出变量	TRUE	TRUE					DF_OUTV
3. 周期性斜坡保持模式打开	TRUE	FALSE	FALSE		TRUE		OUTV(t) 完成时自动启动
4. 暂停斜坡保持	TRUE	FALSE	TRUE	FALSE			所保持的OUTV(t)的当前值 *)
5. 设置继续执行的时间片和时间	TRUE	FALSE	TRUE	TRUE			OUTV (旧) *)
			FALSE				斜坡保持使用新值继续运行。
6. 更新总时间						FALSE	并不影响OUTV
						TRUE	并不影响OUTV

*) 直到下一个点，曲线并不具有用户设置的斜率。

无论阴影域中的控制信号的数值如何，都会执行所选模式。

斜坡保持(Ramp Soak)打开

RMPSK_ON中从FALSE的TRUE的变化激活斜坡保持处理。在到达最后一个时间片(曲线中的最后一个点)之后，斜坡保持(曲线)结束。如果要手动重新启动该功能，必须首先将RMPSK_ON设置成FALSE，然后再设置回TRUE。

缺省输出变量，启动斜坡保持过程

如果要在特定的输出值上启动斜坡保持过程，则必须设置DFOUT_ON = TRUE。这种情况下，在输出上应用信号值DF_OUTV。

注意

恒定设定值DFOUT_ON的输出信号的优先级，比斜坡保持RMPSK_ON的启动信号的优先级高。

在从DFOUT_ON = FALSE中切换出来后，使用从设定值(DF_OUTV)起，到当前点数PI[NBR_ATMS].OUTV的输出值为止的线性变化率，来调整OUTV。

即使在输出上应用了固定设定值(RMPSK_ON = TRUE和DFOUT_ON = TRUE)，内部时间处理也会继续进行。

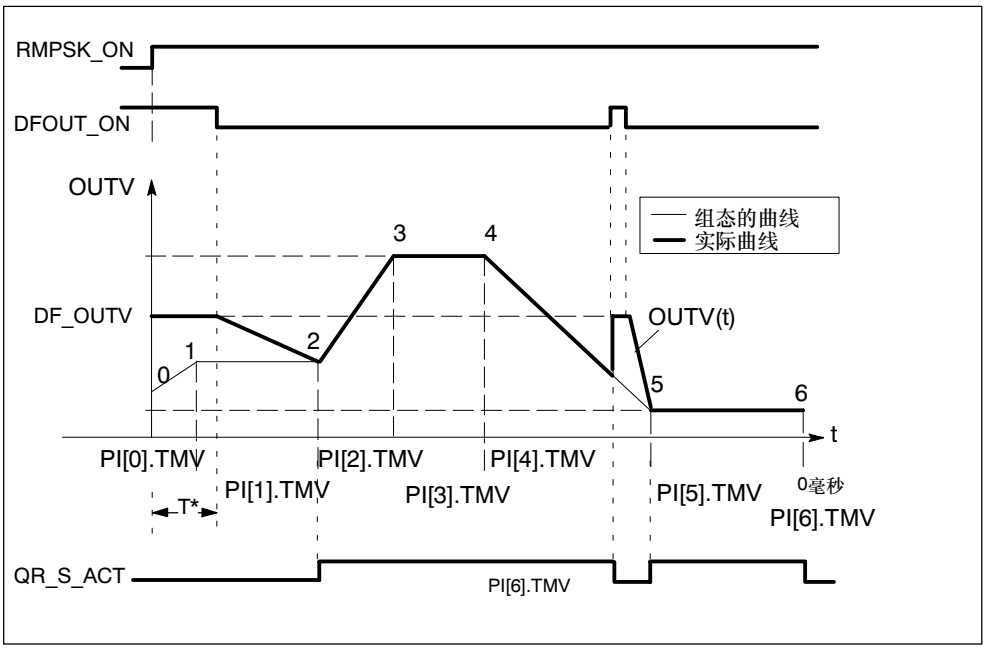


图2-67　　通过缺省信号DFOUT_ON影响斜坡保持

在通过RMPSK_ON = TRUE启动斜坡保持过程时，输出固定设定值DF_OUTV，一直到时间T*之后DFOUT_ON从TRUE变成FALSE (图2-67)。在此点处，已经超过了时间PI[0].TMV和时间PI[1].TMV的一部分。OUTV从DF_OUTV变成PI[2].OUTV，即点2。

只有从点2处开始，才开始输出组态的曲线；也就是从点2处开始，输出信号QR_S_ACT开始具有数值TRUE。如果系统正在处理斜坡保持过程时，缺省信号DFOUT_ON发生变化，则输出值立即OUTV跳到DF_OUTV，无任何延迟。

周期性斜坡保持模式打开

如果“周期性重复”模式激活(CYC_ON = TRUE)，当输出最后一个数值后，斜坡保持自动返回到起始点，重新运行该过程。

在最后一个点和起始点之间没有插值。为了获得平滑的转变，必须应用下式: PI[NBR_PTS].OUTV = PI[0].OUTV.

暂停斜坡保持

如果设置了RMP_HOLD = TRUE，则将保持输出变量的数值(包括时间处理)。当此参数复位(RMP_HOLD = FALSE)时，斜坡保持从停止点PI[x].TMV处继续运行。

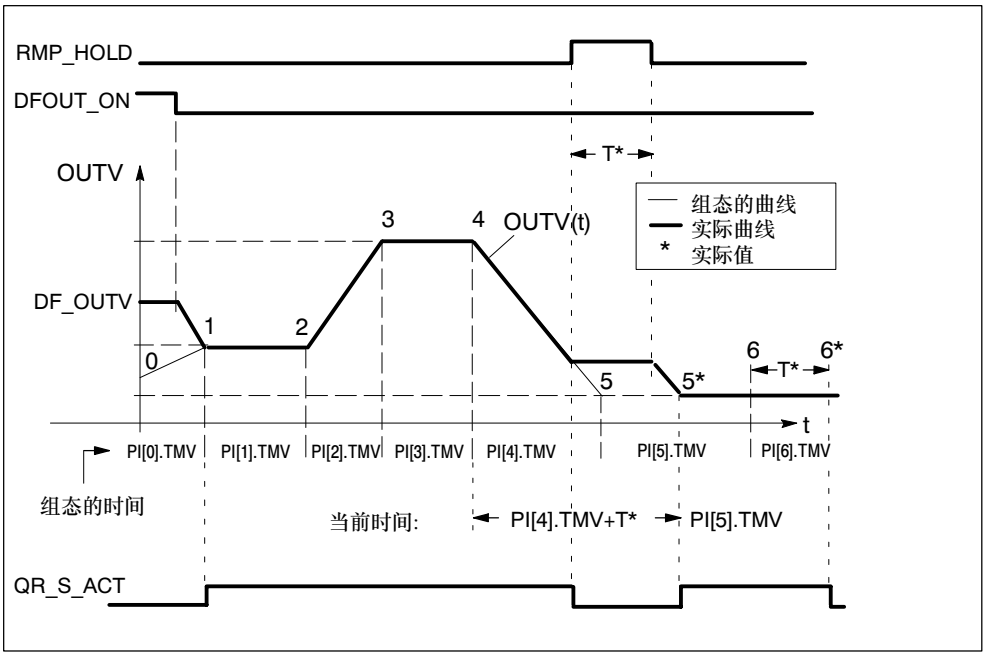


图2-68　　通过暂停信号RMP_HOLD影响斜坡保持

斜坡保持的执行时间被扩展了暂停时间T*。斜坡保持从点1处开始，按照组态的曲线运行，直到RMP_HOLD上发生信号变化(FALSE → TRUE)；然后从点5*到点6*，再次按照组态的曲线运行，即此时输出信号QR_S_ACT再次具有数值TRUE。(图2-68)。

如果设置了位CONT_ON，则已停止的斜坡保持将从指定的点TM_CONT继续运行。

要继续执行的时间片和时间

如果用于继续执行斜坡保持的控制输入CONT_ON设置为TRUE，则斜坡保持将从TM_CONT (继续运行的时间)处继续运行，直到点TM_SNBR (目标点)。时间参数TM_CONT确定了斜坡保持到达目标点TM_SNBR还需要的剩余时间。

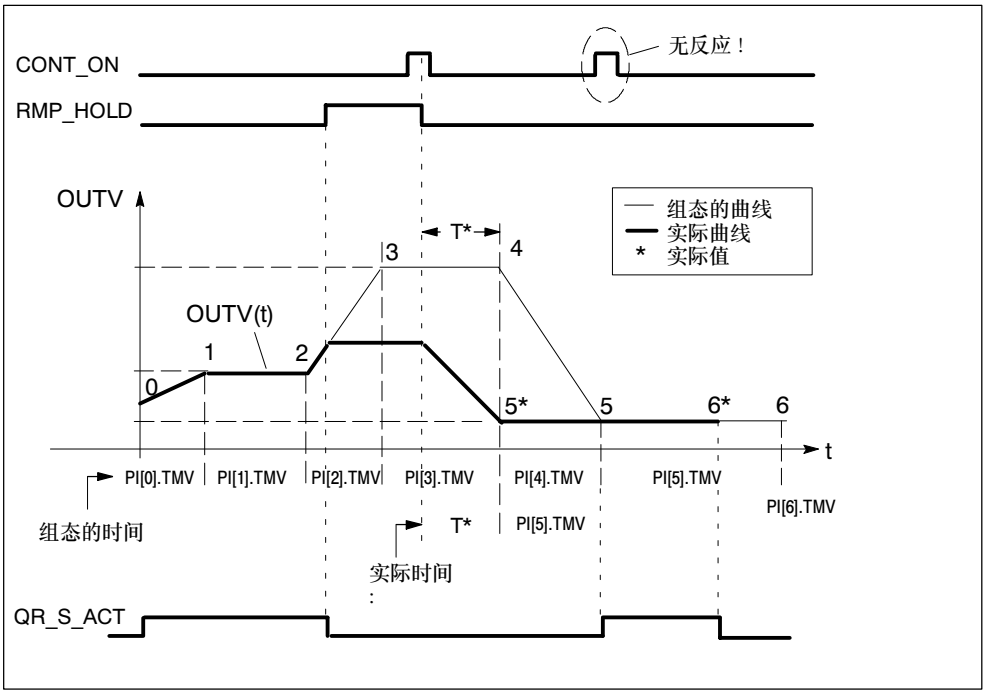


图2-69　影响斜坡保持(Ramp Soak)的暂停信号RMP_HOLD和继续信号CONT_ON

在实例(图2-69)中，如果RMP_HOLD = TRUE，CONT_ON = TRUE，并且如果设置了下列选项:

要继续执行的时间片	TM_SNBR = 5
和到所需点的剩余时间	TM_CONT = T*

对于斜坡保持的处理周期，将忽略所组态的点3和4。在RMP_HOLD上发生从TRUE到FALSE的信号变化后，从点5开始才能再次获得组态的曲线。

只有在斜坡保持运行完用户选择的曲线之后，才会置位输出QR_S_ACT。

更新总时间和剩余总时间

在每个周期内，更新当前点号NBR_ATMS，到达曲线点RS_TM之前的当前实际剩余时间，总时间T_TM 和到达曲线终点RT_TM的剩余总时间。

如果在线改变了PI[n].TMV，则总时间和到达曲线末尾的剩余总时间也会发生变化。当存在大量的时间片时，由于计算T_TM和RT_TM会极大增加功能块的运行时间，因此仅在完全重启动或当设置了TUPDT_ON = TRUE时才会进行计算。对各个曲线点之间的时间片PI[0 ... NBR_PTS].TMV进行求和，并在输出总时间T_TM和剩余总时间RT_TM上对此进行指示。

请注意，确定总时间意味着相对较长的CPU运行时间！

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.23 ROC_LIM: 变化率限制器

应用

当在输入上不得有阶跃变化时，使用斜坡函数。例如，当在电机和负载之间包括了齿轮传动装置时，如果电机转速增加太快便会使齿轮传动装置超载时便是这种情况。

方框图

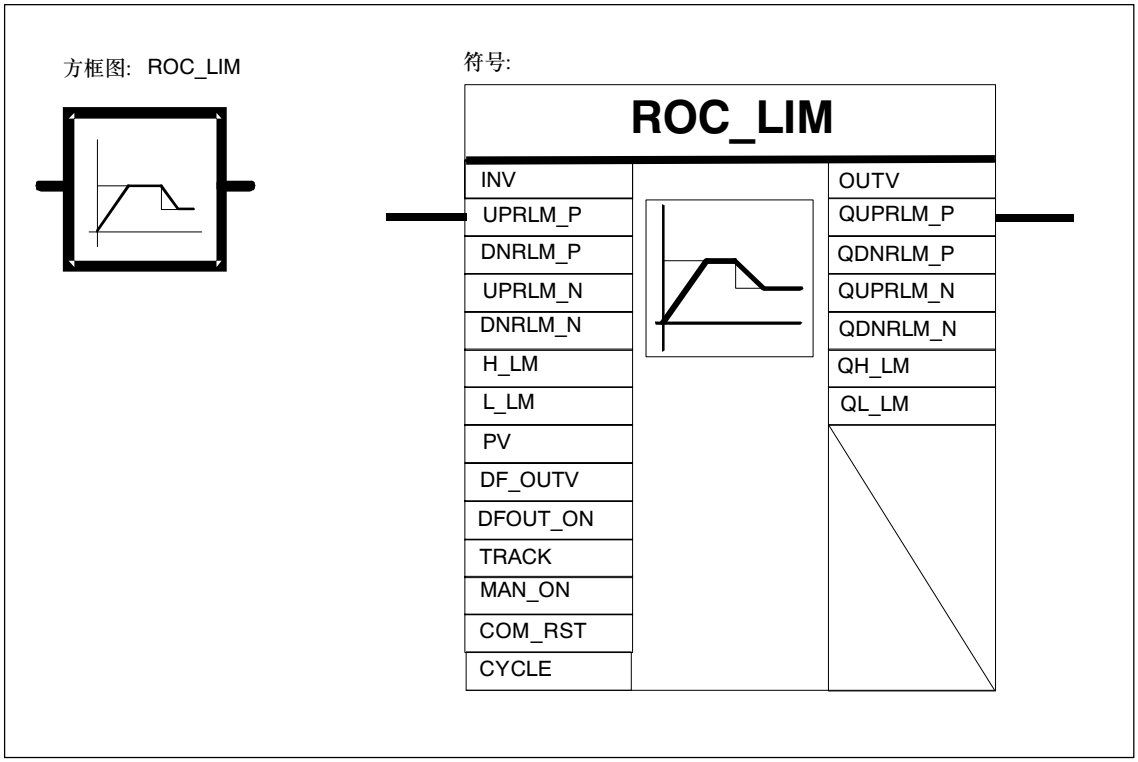


图2-70 ROC_LIM，方框图和符号

功能描述

该块限制输出值的变化率。阶跃变化变成了斜坡函数。可以为输入变量和输出变量，在正范围和负范围内选择两个斜坡(上升值和下降值)。控制输入设置下列模式:

- 缺省输出变量
- 跟踪
- 无阶跃自动手动切换

可以通过两个可选限制来限制输出变量的值。如果到达了上升或下降的变化率限制，或者到达上限/下限，在输出上对此进行指示。

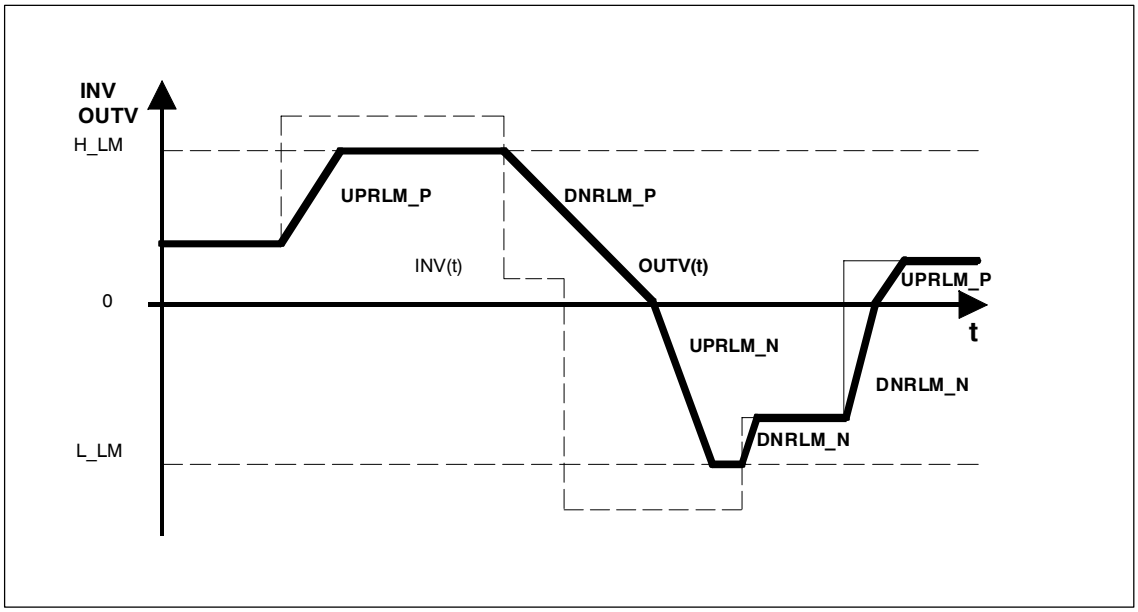


图2-71 斜坡函数

通过下列方式识别斜坡:

OUTV > 0，并且|OUTV|上升	UPRLM_P
OUTV > 0，并且|OUTV|上升	UPRLM_P
OUTV > 0，并且|OUTV|下降	DNRLM_P
OUTV < 0，并且|OUTV|上升	UPRLM_N
OUTV < 0，并且|OUTV|下降	DNRLM_N

输入参数

下表给出了ROC_LIM的输入参数的数据类型和结构。

表2-53

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	UPRLM_P	正范围中的向上速率限制	>0.0	10.0
REAL	DNRLM_P	正范围中的向下速率限制	>0.0	10.0
REAL	UPRLM_N	负范围中的向上速率限制	>0.0	10.0
REAL	DNRLM_N	负范围中的向下速率限制	>0.0	10.0
REAL	H_LM	上限	技术范围 > L_LM	100.0
REAL	L_LM	下限	技术范围 < H_LM	0.0
REAL	PV	过程变量	技术取值范围	0.0
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
BOOL	DFOUT_ON	缺省输出变量打开		FALSE
BOOL	TRACK	跟踪OUTV = INV		FALSE
BOOL	MAN_ON	手动模式开启		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间		T#1s

输出参数

下表给出了ROC_LIM的输出参数的数据类型和结构。

表2-54

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0
BOOL	QUPRLM_P	已到达正范围中的向上速率限制	FALSE
BOOL	QDNRLM_P	已到达正范围中的向下速率限制	FALSE
BOOL	QUPRLM_N	已到达负范围中的向上速率限制	FALSE
BOOL	QDNRLM_N	已到达负范围中的向下速率限制	FALSE
BOOL	QH_LM	已达到上限	FALSE
BOOL	QL_LM	已达到下限	FALSE

完全重启动

在完全重启动期间，输出OUTV复位为0.0。如果设置了DFOUT_ON=TRUE，则输出DF_OUTV。所有信号输出都设置为FALSE。

正常操作

斜率是直线限制曲线，与每秒的增加/减小相关。例如，如果在采样时间为1s/100ms/10ms时，为UPRLM_P设置了数值10.0，则当调用块时，如果INV > OUTV，则在OUTV上增加10.0/1.0/0.1，直到到达INV。如果输入变量超出H_LM或下降到低于L_LM，可以向上和向下限制输出变量。(例外：手动模式MAN_ON=TRUE；参见实例图2-72)

如果超过了其中一个限制值，则在输出QUPRLM_P、QDNRLM_P、QUPRLM_N、QDNRLM_N、QH_LM和QL_LM上对此进行指示。

- **缺省输出变量**

如果设置了DFOUT_ON = TRUE，则输出DF_OUTV。如果从TRUE变为FALSE，则OUTV从DF_OUTV变为INV；而如果从FALSE变为TRUE，则OUTV从INV变为DF_OUTV。

- **跟踪**

要跟踪(OUTV = INV)，需要设置位TRACK = TRUE。由于直接将输入变量转到输出变量，所以输入变量中的所有阶跃变化都会被输出。

- **无阶跃手动自动切换**

对于此模式，必须在偏差信号之前，直接将斜坡块包含到设定值分支中。过程变量连接到输入PV，而手动自动位则连接到输入MAN_ON。当切换到手动模式MAN_ON = TRUE时，输入PV上的数值立即转到输出OUTV上。由于设定值和过程变量相同，所以偏差信号变为零，控制器便处于稳定状态。当返回到自动模式MAN_ON = FALSE时，斜坡函数确保了输出OUTV逐渐从当前值PV变化到输入值INV。这样便形成了从手动模式到自动模式的无阶跃切换(参见图2-72)

如果设置了MAN_ON=TRUE，缺省输出变量模式具有较低的优先级，因此将被忽略。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

实例

如果MAN_ON、TRACK、DFOUT_ON设置为FALSE，则信号输出所具有的值如下所示:

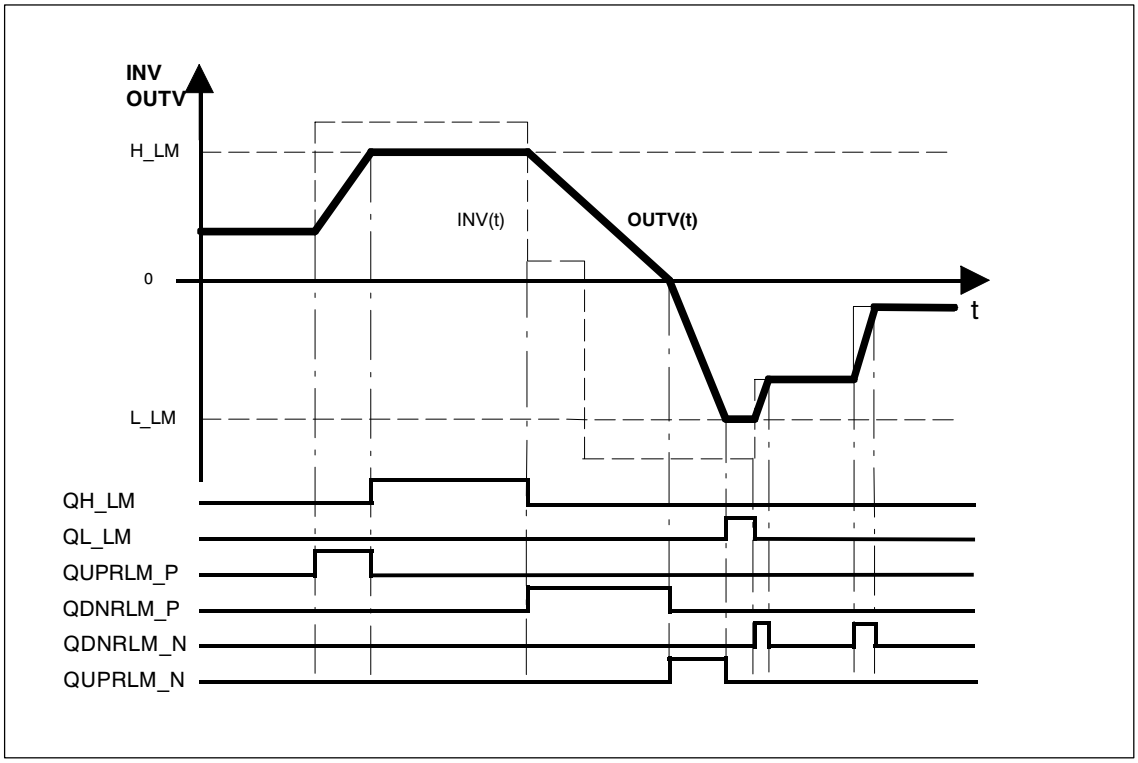


图2-72 实例(MAN_ON、TRACK、DFOUT_ON = FALSE; L_LM < 0.0 < H_LM)

如果设置了MAN_ON，则限制H_LM和L_LM无效。如果设置了TRACK，则输出输入值，不做任何改动。如果DFOUT_ON = TRUE，则总是输出DF_OUTV。

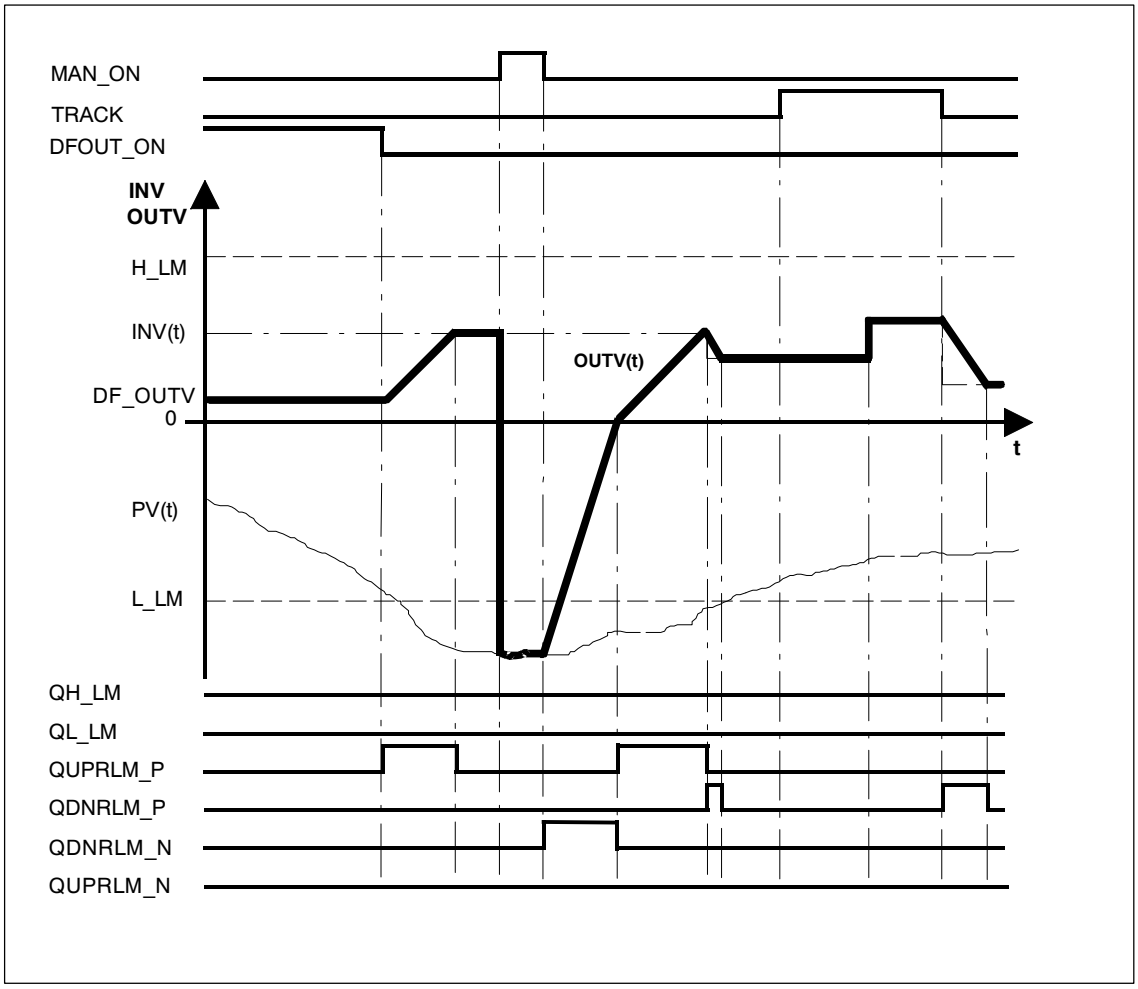


图2-73　　实例(L_LM < 0.0 < H_LM)

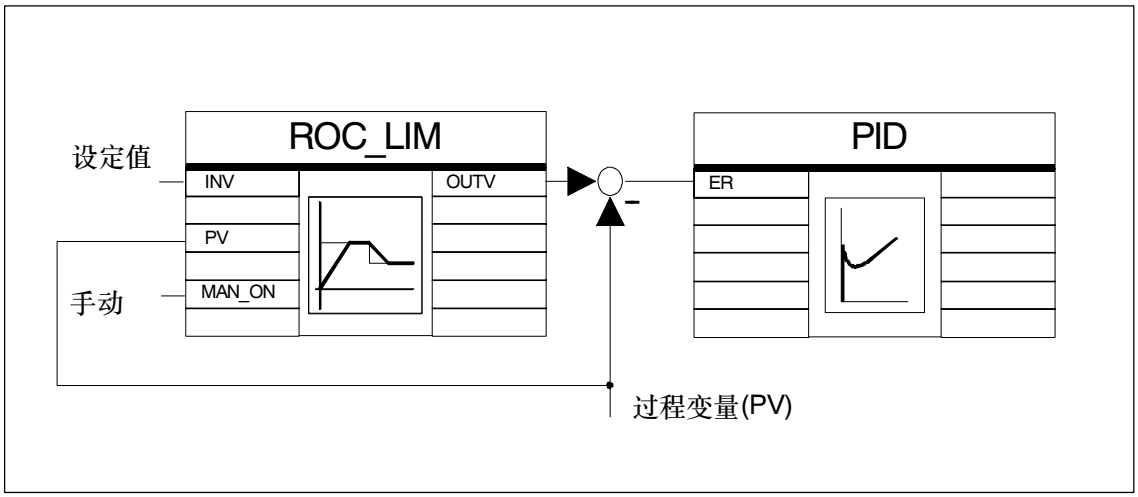


图2-74　　从手动到自动无阶跃切换实例

仅在正范围内使用的斜坡的实例

输入参数

下表给出了输入参数的数据类型和结构。

表2-55　输入参数

数据类型	参数	注释	参数分配
REAL	INV	输入变量	0.0
REAL	UPRLM_P	正范围中的向上速率限制	10.0
REAL	DNRLM_P	正范围中的向下速率限制	5.0
REAL	UPRLM_N	负范围中的向上速率限制	0.0
REAL	DNRLM_N	负范围中的向下速率限制	0.0
REAL	H_LM	上限	85.5
REAL	L_LM	下限	27.0
REAL	PV	过程变量	0.0
REAL	DF_OUTV	缺省输出变量	46.15
BOOL	DFOUT_ON	缺省输出变量打开	FALSE
BOOL	TRACK	跟踪OUTV = INV	FALSE
BOOL	MAN_ON	手动模式开启	FALSE
BOOL	COM_RST	完全重启动	FALSE
TIME	CYCLE	采样时间	T#1s

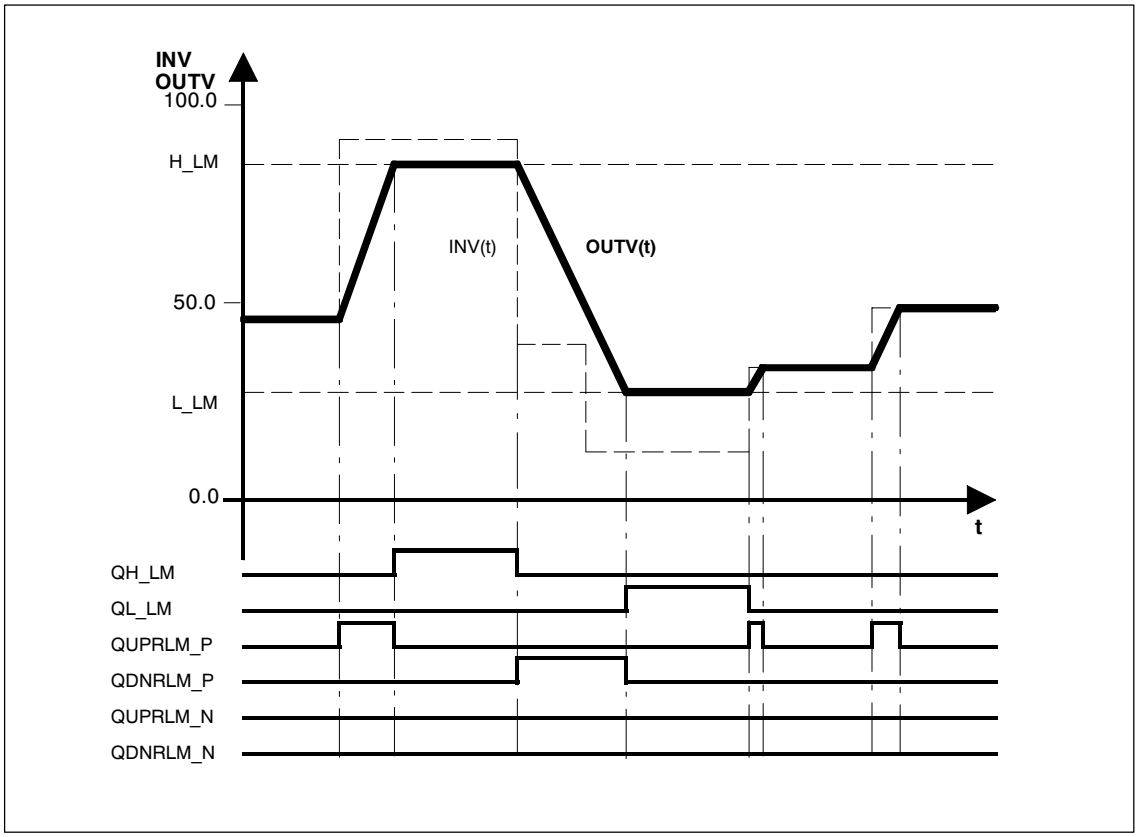


图2-75　　只具有值≥ 0.0的运行范围的实例

2.1.24 SCALE: 线性转换

应用

通常，传感器所提供的过程变量的数值所处的范围并不适合用户使用(例如，0 ~ 10 V对应0 ~ 1200 °C或0 ~ 10 V对应0 ~ 3000 rpm)。通过调整设置值或过程变量，可以使两个变量具有相同的数值范围。

方框图

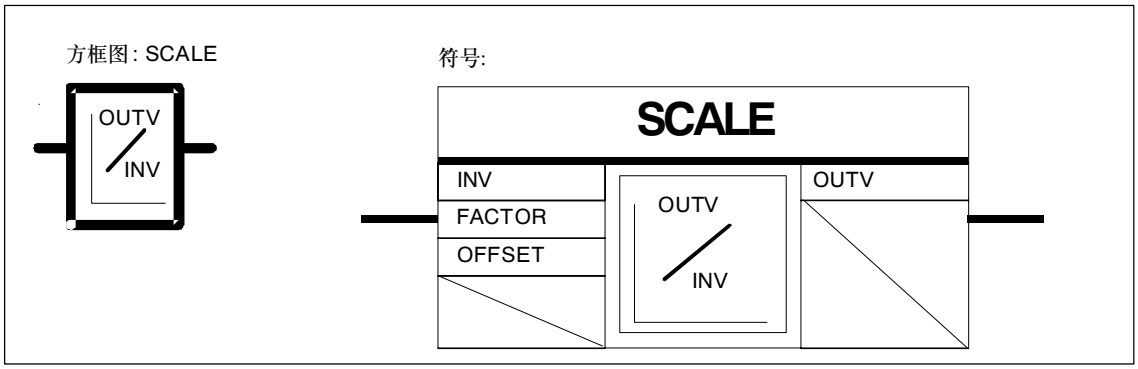


图2-76 SCALE，方框图和符号

功能描述

该块用于规格化模拟变量。规格化曲线由斜率(FACTOR)和OUTV在INV = 0和坐标轴OUTV = 0之间的距离定义。

算法

OUTV = INV * FACTOR + OFFSET

通过规格化曲线，将模拟变量INV应用到输出变量OUTV。规格化曲线由变量FACTOR和OFFSET定义。

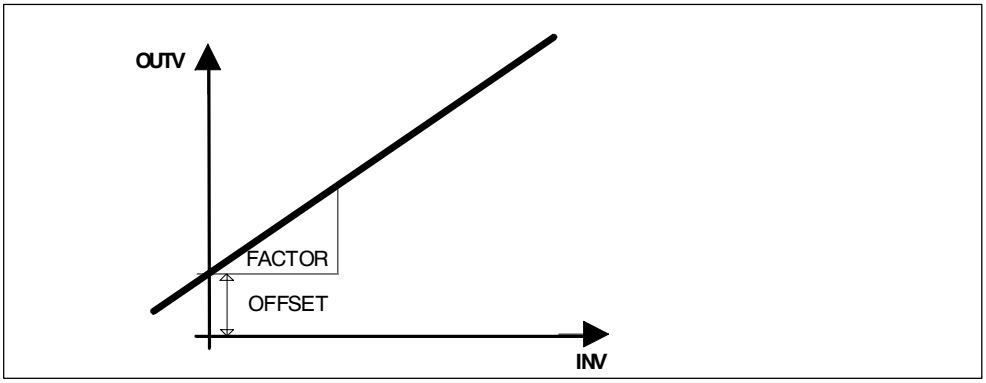


图2-77 带限制的规格化曲线

输入参数

下表给出了SCALE的输入参数的数据类型和结构。

表2-56 SCALE的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	FACTOR	线性标定因子		1.0
REAL	OFFSET	偏移量	技术取值范围	0.0

输出参数

下表给出了SCALE的输出参数的数据类型和结构。

表2-57 SCALE的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV	输出变量	0.0

完全重启动

块没有完全重启动例行程序。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

2.1.25 SP_GEN: 设定值发生器

应用

要手动输入设定值，可以通过SP_GEN块，使用两个输入来修改输出值。为了能够检测到小的变化，块应该具有≤100 ms的采样时间。

方框图

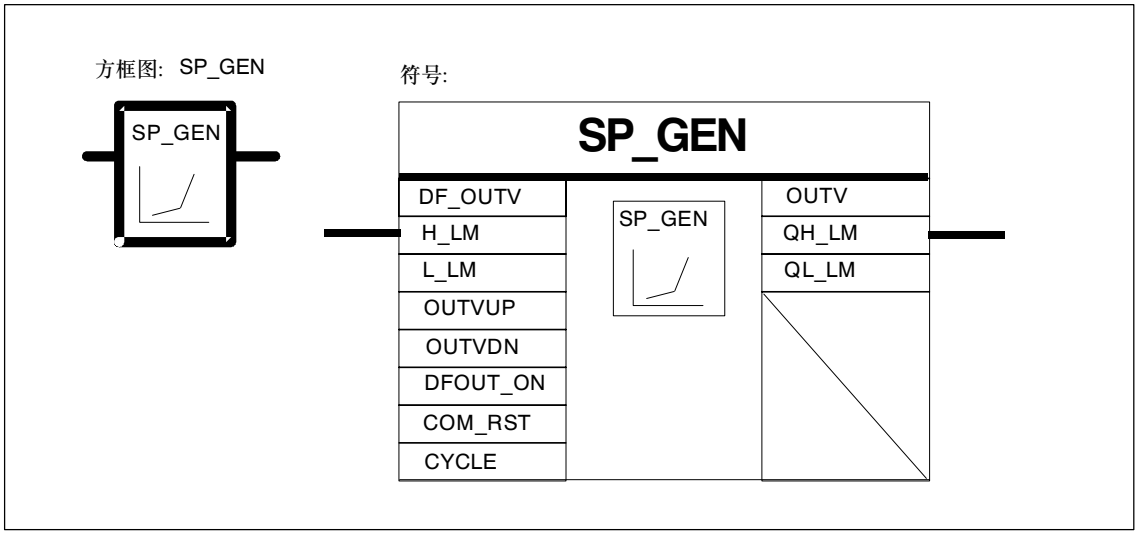


图2-78 SP_GEN，方框图和符号

功能描述

在输入OUTVUP和OUTVDN上，输出变量OUTV可以在限制H_LM和L_LM范围内连续增加或减小。变化率取决于OUTVUP和OUTVDN激活的时间长度。

在设置了OUTVUP或OUTVDN之后的头三秒钟内，变化率是

dV/dt = (H_LM - L_LM) / 100 秒；然后，它变为

dV/dt = (H_LM - L_LM) / 10秒。

OUTV的值是L_LM OUTV H_LM；如果限制了OUTV，则会显示一条消息。

通过DFOUT_ON，可以将OUTV分配给DF_OUTV。

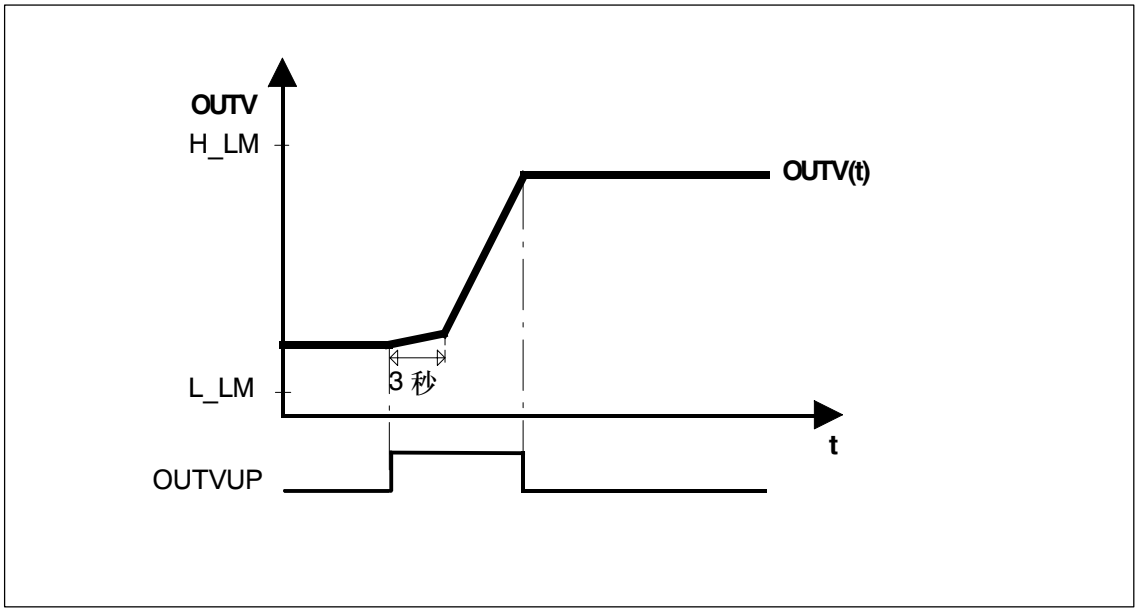


图2-79　　通过设置OUTVUP改变输出值

输入参数

下表给出了SP_GEN的输入参数的数据类型和结构。

表2-58　　SP_GEN的输入参数

数据类型	参数	注释	块内部限制	缺省值
REAL	DF_OUTV	缺省输出变量	技术取值范围	0.0
REAL	H_LM	上限	技术范围 > L_LM	100.0
REAL	L_LM	下限	技术范围 < H_LM	0.0
BOOL	OUTVUP	输出变量增加		FALSE
BOOL	OUTVDN	输出变量减小		FALSE
BOOL	DFOUT_ON	缺省输出变量打开		FALSE
BOOL	COM_RST	完全重启动		FALSE
TIME	CYCLE	采样时间		T#100ms

输出参数

下表给出了SP_GEN的输出参数的数据类型和结构。

表2-59　SP_GEN的输出参数

数据类型	参数	注释	完全重启动之后的缺省设置
REAL	OUTV	输出变量	0.0
BOOL	QH_LM	已达到上限	FALSE
BOOL	QL_LM	已达到下限	FALSE

完全重启动

在完全重启动期间，输出OUTV设置为0.0。如果设置了DFOUT_ON=TRUE，则输出DF_OUTV。在完全重启动期间，这些限制也有效。

正常操作

DFOUT_ON、OUTVUP和OUTVDN对OUTV有下列影响:

DFOUT_ON	OUTVDN	OUTVUP	OUTV
TRUE	ANY	ANY	DF_OUTV
FALSE	TRUE	TRUE	OUTV未变化
	FALSE		OUTV正在增加
	TRUE	FALSE	OUTV正在减小
	FALSE		OUTV未变化

- **缺省输出变量(DFOUT_ON = TRUE)**

如果设置了DFOUT_ON = TRUE，输出DF_OUTV。如果DF_OUTV的数值高于/低于H_LM/L_LM，它被限制为H_LM/L_LM，并且输出QH_LM/QL_LM=TRUE。OUTV中的变化是阶跃变化。到DFOUT_ON = FALSE的切换无任何突然变化。

- **减少输出值(OUTVDN=TRUE)**

如果OUTVDN=TRUE，OUTV按下式减小3秒钟

$$(H\_LM - L\_LM) * \frac{CYCLE}{100秒}$$

3秒钟之后，OUTV每个周期按下式减小

$$(H\_LM - L\_LM) * \frac{CYCLE}{10秒}$$

如果OUTV的数值小于L_LM，则它被限制为L_LM，并输出QL_LM=TRUE；如果设置了OUTVDN=FALSE，也会设置QL_LM=FALSE。OUTVDN的优先级比DFOUT_ON低。

- **增加输出值(OUTVUP=TRUE)**

如果设置了OUTVUP=TRUE，则会应用与OUTVDN上的变化率相同的变化率。

如果OUTV的数值大于H_LM，则它被限制为H_LM，并输出QH_LM=TRUE；如果设置了OUTVUP=FALSE，也会设置QH_LM=FALSE。OUTVUP的优先级比OUTVDN低。

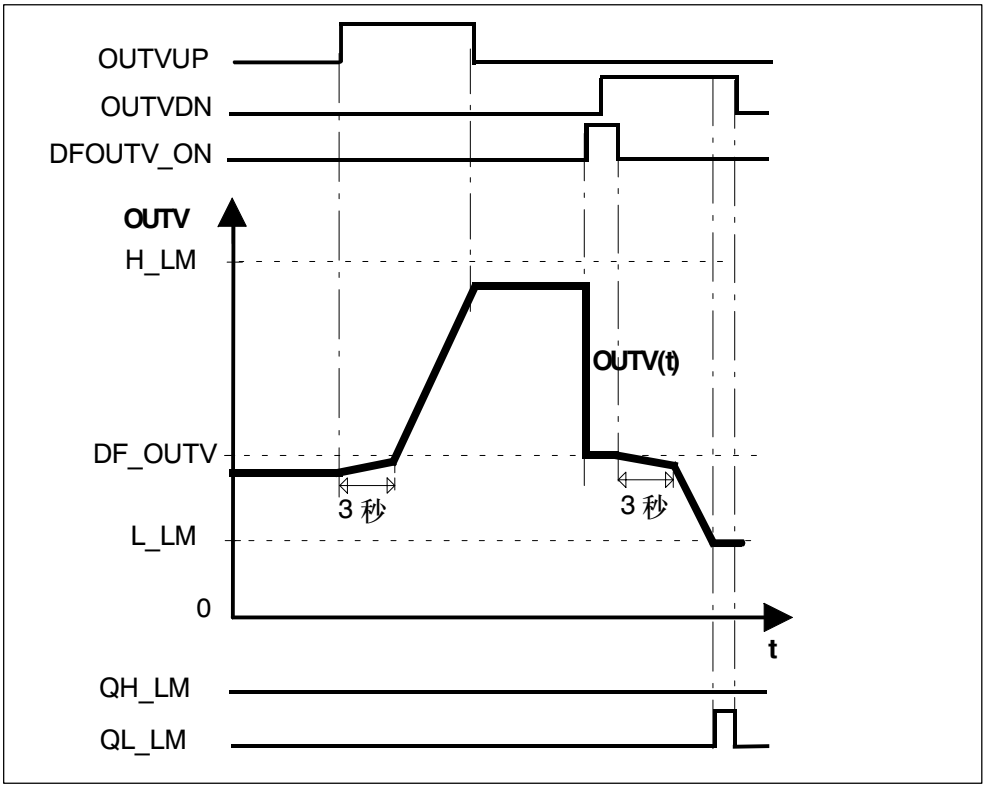


图2-80　　通过OUTVUP、OUTVDN和DFOUTV_ON影响OUTV

块内部限制

在块内部，未对任何值进行限制；　系统不会检查参数。

2.1.26 SPLT_RAN: 拆分范围

应用

在运用分段控制器时需要此块。

PID控制器的调节值范围被拆分成了多个子范围。该块必须每个子范围调用一次，并连接到其中一个调节值处理块LMNGEN_C或LMNGEN_S。

方框图

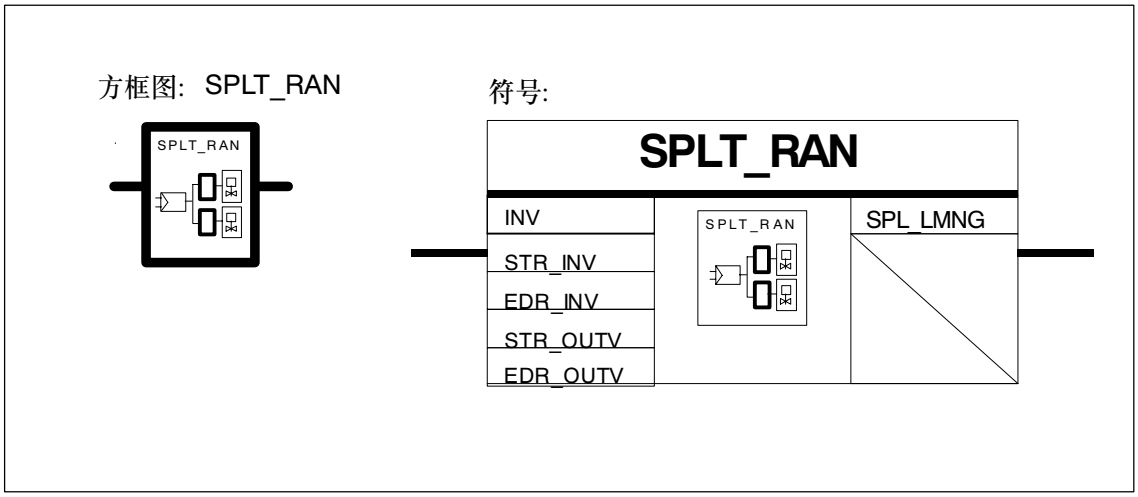


图2-81 SPLT_RAN，方框图和符号

功能描述

位于由STR_INV和EDR_INV所限制的范围之内的输入值被转换成位于由STR_OUTV和EDR_OUTV所限制的范围之内的输出值(参见图2-82)。

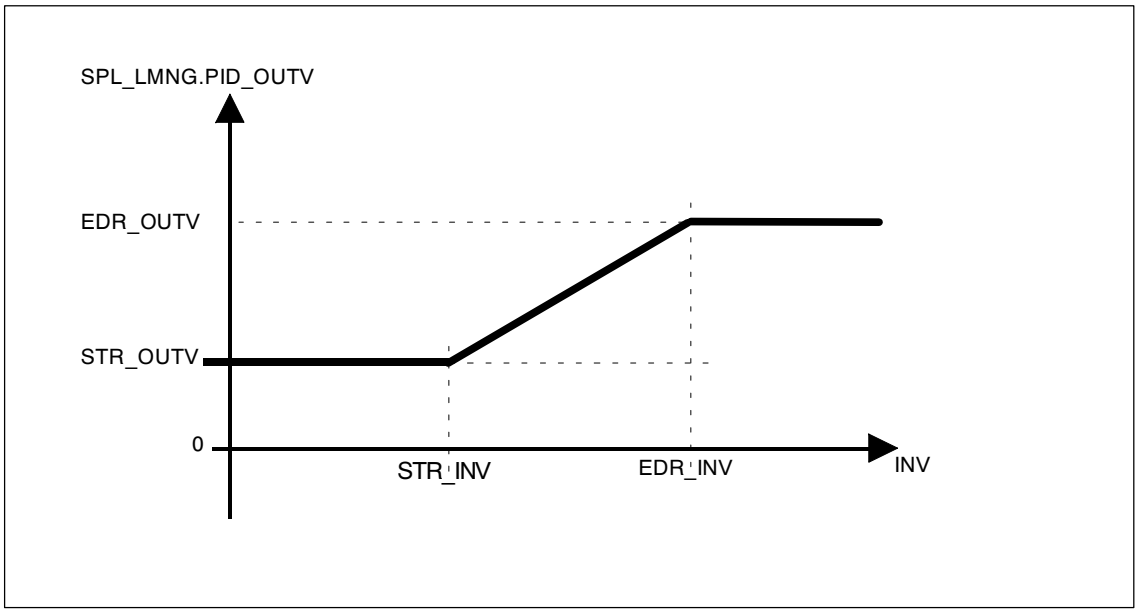


图2-82

输入参数

下表给出了SPLT_RAN的输入参数数据类型和结构。

表2-60 SPLT_RAN的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV	输入变量	技术取值范围	0.0
REAL	STR_INV	范围INV的起点	技术取值范围	0.0
REAL	EDR_INV	范围INV的终点	技术取值范围	50.0
REAL	STR_OUTV	范围OUTV的起点	技术取值范围	0.0
REAL	EDR_OUTV	范围OUTV的终点	技术取值范围	100.0

输出参数

下表给出了SPLT_RAN的输出参数的数据类型和结构。

表2-61 SPLT_RAN的输出参数

数据类型	参数	注释	缺省值
STRUC	SPL_LMNG	PID-LMNGEN接口	

完全重启动

块没有完全重启动例行程序。

正常操作

块没有除常规操作之外的其它模式。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

实例

通过SPLT_RAN，将PID块的输出值分配到两个调节值处理块LMNGEN_C和LMNGEN_S上。

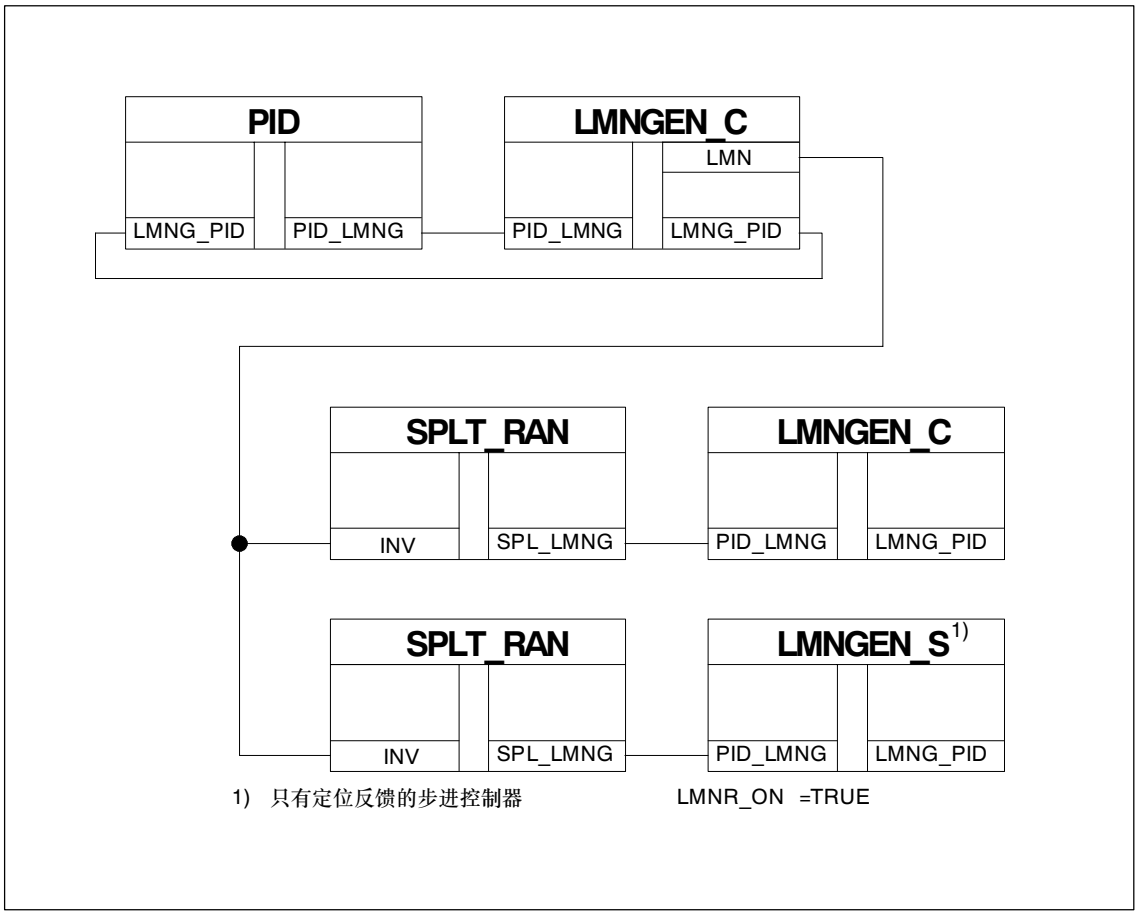


图2-83 带有PID和LMNGEN_S的 SPLT_RAN的连接

2.1.27 SWITCH: 开关

应用

该块用作两个输入/输出变量的输入和/或输出多路复用器。

方框图

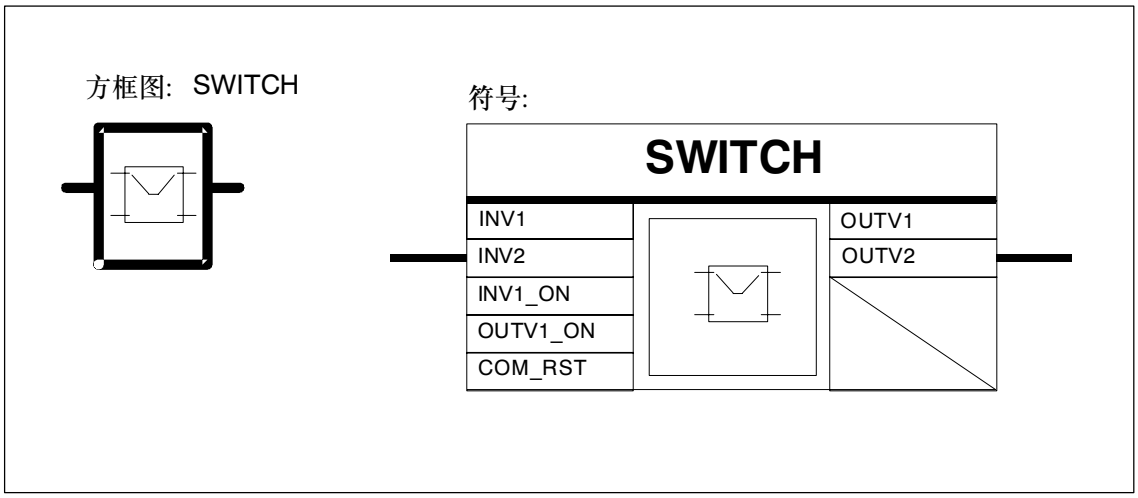


图2-84 SWITCH，方框图和符号

功能描述

该块根据下表，将两个模拟量输入值中的一个接到两个输出值中的一个上:

INV1_ON	OUTV1_ON	OUTV1	OUTV2
0	0	未变化	INV2
1	0	未变化	INV1
0	1	INV2	未变化
1	1	INV1	未变化

输入参数

下表给出了SWITCH的输入参数的数据类型和结构。

表2-62 SWITCH的输入参数

数据类型	参数	注释	允许使用的数值	缺省值
REAL	INV1	输入变量1	技术取值范围	0.0
REAL	INV2	输入变量2	技术取值范围	0.0
BOOL	INV1_ON	通过INV1连接		FALSE
BOOL	OUTV1_ON	通过OUTV1连接		FALSE
BOOL	COM_RST	完全重启动		FALSE

输出参数

下表给出了SWITCH的输出参数的数据类型和结构。

表2-63 SWITCH的输出参数

数据类型	参数	注释	缺省值
REAL	OUTV1	输出变量1	0.0
REAL	OUTV2	输出变量2	0.0

完全重启动

在完全重启动期间，设置OUTV1=0.0和OUTV2=0.0。

正常操作

该块除了正常操作模式以外，没有其它模式。

块内部限制

参数的值在块中不受限制。系统不会检查参数。

实例

3

3.1 使用模块化PID控制

总览

通过使用来自“模块化PID控制”的ModPID库中的块，可以创建自己的特定控制器。

项目zEn28_4_ModCon包含12个控制器结构(EXAMPLE01到EXAMPLE12)的实例。章节3.2到3.13描述了这12个实例，它们由第2章中所述的ModPID库的功能块组成。

实例及其使用

表3-1列出了 zEn28_4_ModCon项目中提供的实例。

表3-1 实例列表

实例	功能
EXAMPLE01	带有用于集成执行器的开关输出的固定设定值控制器
EXAMPLE02	带有连续输出的固定设置值控制器
EXAMPLE03	带有用于比例执行器的开关输出的固定设置值控制器
EXAMPLE04	单回路比例控制器
EXAMPLE05	多回路比例控制器
EXAMPLE06	混合控制器
EXAMPLE07	级联控制器
EXAMPLE08	带有预控制器的控制器
EXAMPLE09	带有前馈控制的控制器
EXAMPLE10	分段控制器
EXAMPLE11	倍率控制器
EXAMPLE12	多变量控制器

根据表3-1中的实例，可以看到最重要的一些块的调用和互连。

可以复制与您所需要的控制器结构最接近的实例作为模板；然后通过删除或添加块调用和互连来修改该模板。

注意

只有实例1到3包含过程模拟，可以在没有到连接过程的情况下运行。

实例4到12需要到过程的实际连接。在可以使用这些实例之前，必须为块(CRP_IN、LMNGEN_C、LMNGEN_S、SP_GEN ...)分配新参数值，过程值是通过这些新参数连接的。

您将自己控制器的结构(用户FB)实现为具有来自ModPID库的FB的本地实例的FB。您的用户FB包含了块调用和输入/输出参数的互连。既可以使用STL，也可以使用SCL创建自己的用户FB。

可以根据应用需要，在合适的组织块中调用这些控制器(用户FB)。

STL编程实例

下面的实例说明了如何调用来自ModPID库的块，以及如何使用STL互连这些块。

地址	声明	名称	类型
0.0	in	SP_UP	BOOL
0.1	in	SP_DOWN	BOOL
2.0	out	OUT	REAL
6.0	stat	DI_SP_GEN	FB 25
46.0	stat	DI_ROC_LIM	FB 22

```
STL                              解释
程序段1:
CALL    #DI_SP_GEN               //块调用
        OUTVUP  := #SP_UP
        OUTVDN  :=#SP_DOWN
L       #DI_SP_GEN.OUTV          //互连

T       #DI_ROC_LIM.INV
CALL    #DI_ROC_LIM              //块调用
        OUTV    :=#OUT
BE
```

SCL编程实例

下面的实例说明了如何调用来自ModPID库的块，以及如何使用SCL互连这些块。

```
FUNCTION_BLOCK  User FB
VAR_INPUT
      SP_UP:                    bool := FALSE;
      SP_DOWN:                  bool := FALSE;
END_VAR
VAR_OUTPUT
      OUT:                      real := 0.0;
END_VAR
VAR
      DI_SP_GEN:   SP_GEN;
      DI_ROC_LIM:  ROC_LIM;
END_VAR
BEGIN
      DI_SP_GEN(                //块调用 + 互连
      OUTVUP  := SP_UP,
      OUTVDN  :=SP_DOWN);
      DI_ROC_LIM(               //块调用 + 互连
      INV := DI_SP_GEN.OUTV);
      OUT := DI_ROC_LIM.OUTV;   //互连
END_FUNCTION_BLOCK
```

实例练习

实例1到3包含了一个完整的控制回路。它们特别适合练习使用。

通过使用标准工具“监视和修改变量”，可以简单地修改控制参数，然后就可以观察所做的改动在模拟的控制回路中造成的结果。

组态工具提供了一个带有回路监视器和曲线记录器的图形界面，允许进行过程识别。

3.2 实例1：带有开关输出的固定设置值控制器，用于带有过程模拟的集成执行器

总览

实例1的名称为EXAMPLE01，它包含了一个PID步进控制器(带有用于集成执行器的开关输出的固定设置值控制器)和模拟的工业过程。

控制闭环

图3-1给出了实例1的完整控制回路。

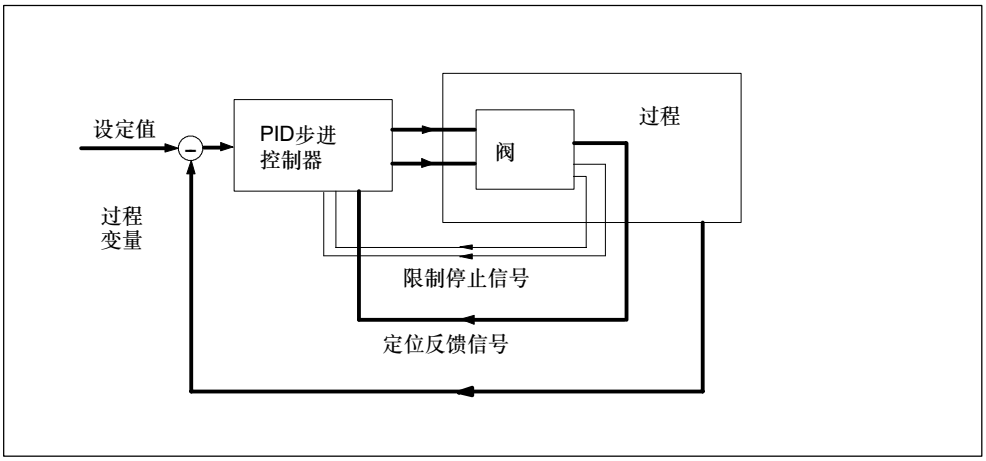


图3-1 实例1的控制回路

注意

必须设置DB50.DI_LMNGEN_S.MAN_ON为FALSE，这样才能使用组态工具的回路监视器功能。

块调用和互连

图3-2给出了实例1的块调用和互连。

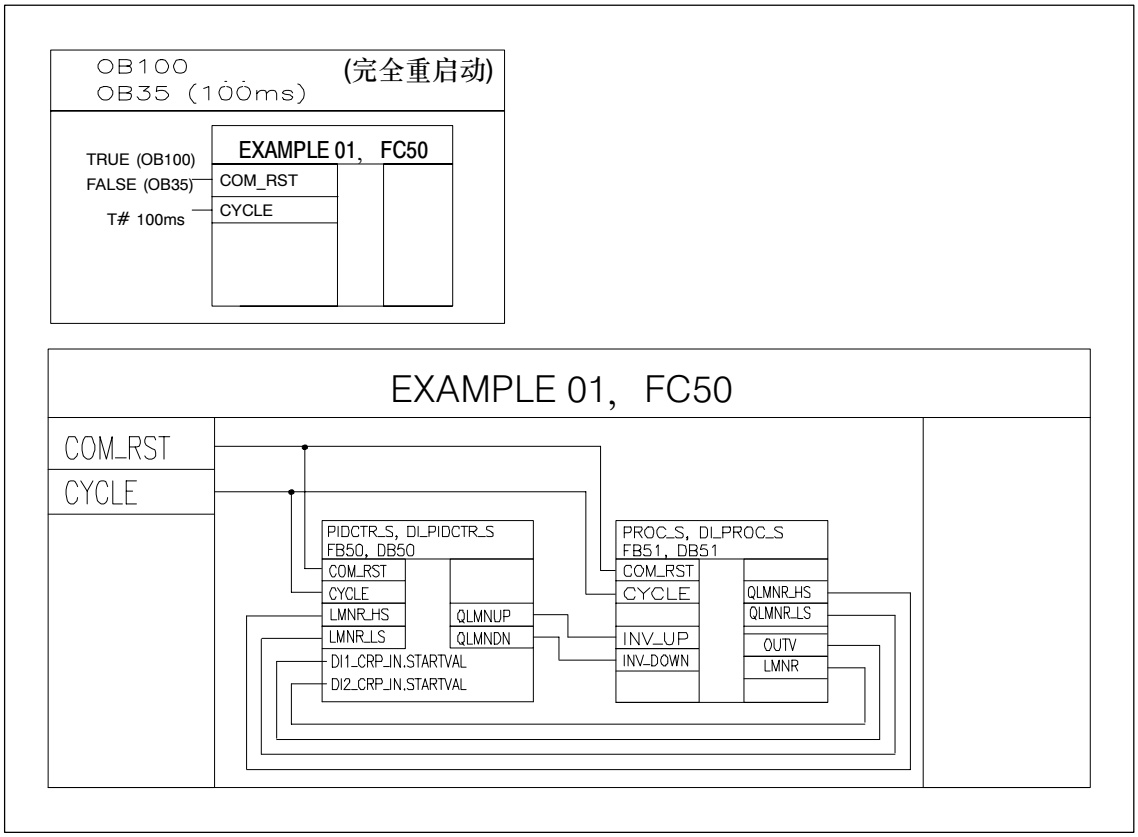


图3-2　　实例1的块调用和互连

3.2.1 PIDCTR_S: 带有用于集成执行器的开关输出的固定设置值控制器

应用

PIDCTR_C块实现了一个用于集成执行器的PID步进控制器(例如，工业过程中的电机驱动阀)。图3-3给出了PIDCTR_S的块互连。

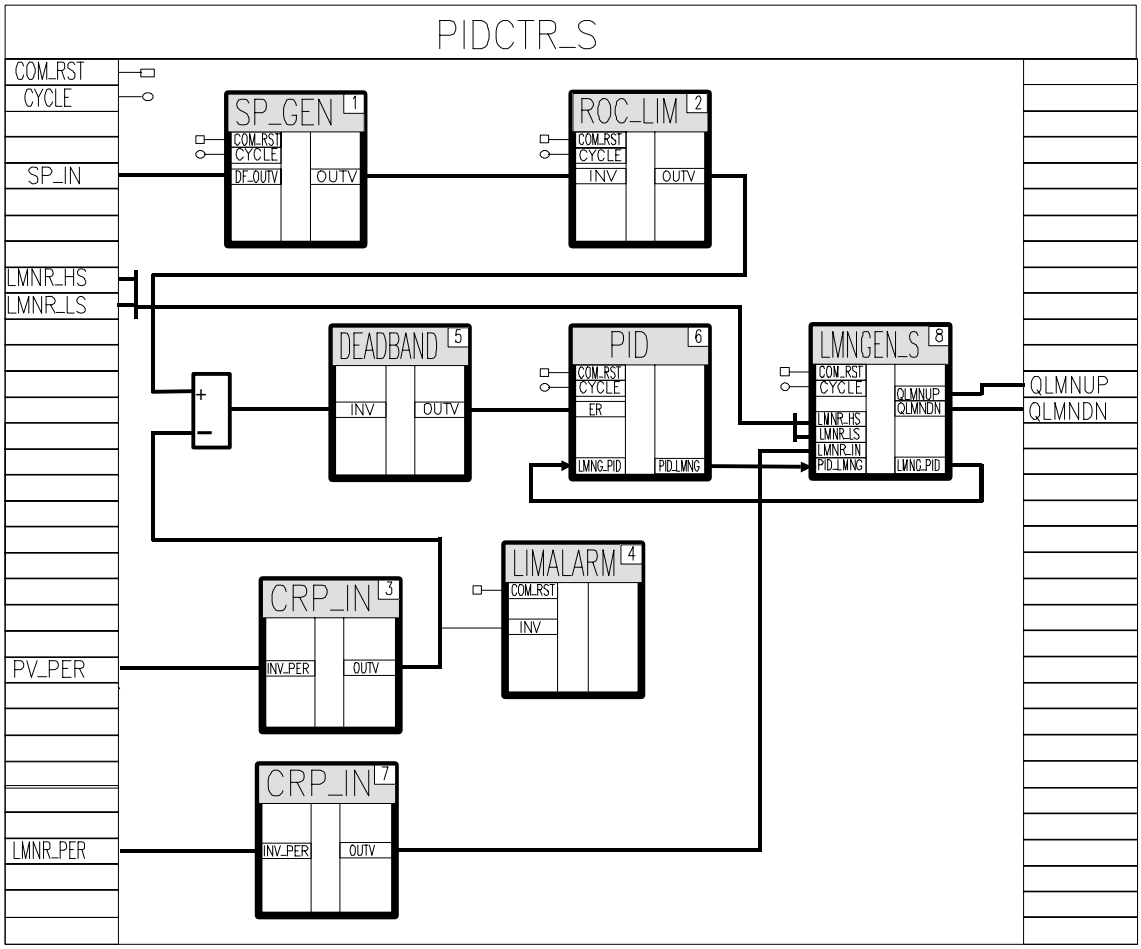


图3-3 PIDCTR_S的块互连

功能描述

设定值发生器SP_GEN设置设定值，该设定值的变化速率受限制器ROC_LIM的限制。CRP_IN将外围设备过程变量转换成浮点值，限制值监视器LIMALARM对其进行监视，检查它是否超出了所选择的限制值。误差信号通过DEADBAND发送到PID算法。通过第二个CRP_IN块读入位置反馈信号。调节值处理块LMNGEN_S设置输出信号QLMNUP和QLMNDN。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.2.2 PROC_S: 步进控制器的过程

应用

PROC_S块通过3阶时间延迟来模拟集成执行器。

图3-4显示PROC_S的方框图。

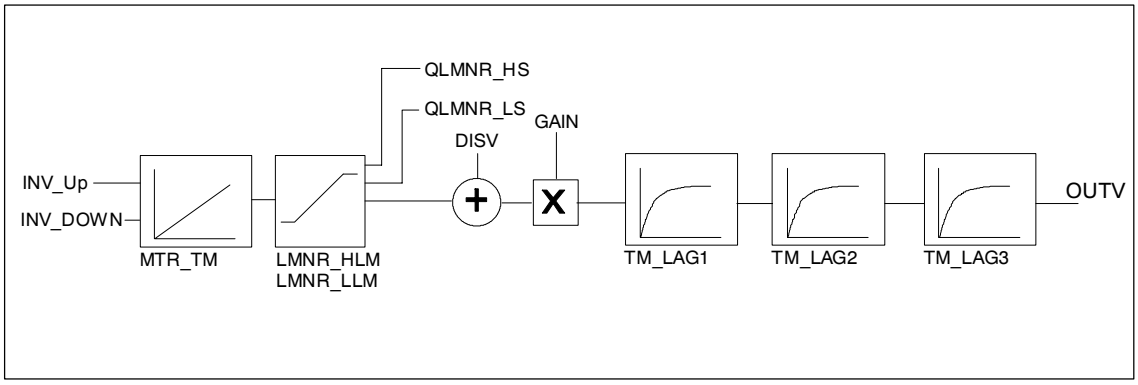


图3-4 PROC_S的方框图

功能描述

块模拟集成执行器和三个串级连接的1阶时间延迟。始终在执行值的输出上添加干扰变量**DISV**。电机执行时间**MTR_TM**是阀从一个限位停止到另一个限位停止所需要的时间。

完全重启动

在完全重启动期间，输出变量**OUTV**和内部存储的数值都设置为0。

3.3 实例2: 带有连续输出的固定设置值控制器，带有过程模拟

总览

实例2的名称为EXAMPLE02，它包含了一个连续PID控制器和一个模拟的工业过程。

控制闭环

图3-5给出了实例2的完整控制回路。

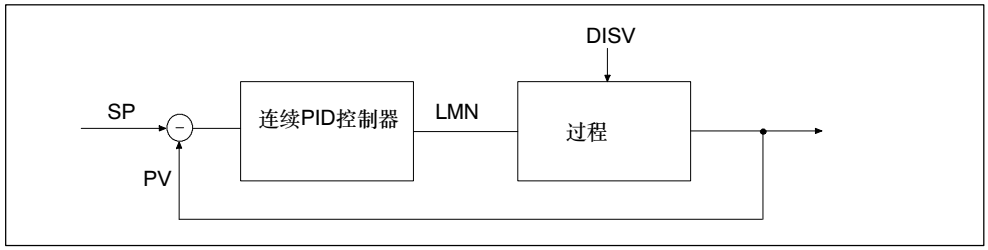


图3-5 实例2的控制回路

块调用和互连

图3-6给出了实例2的块调用和互连。

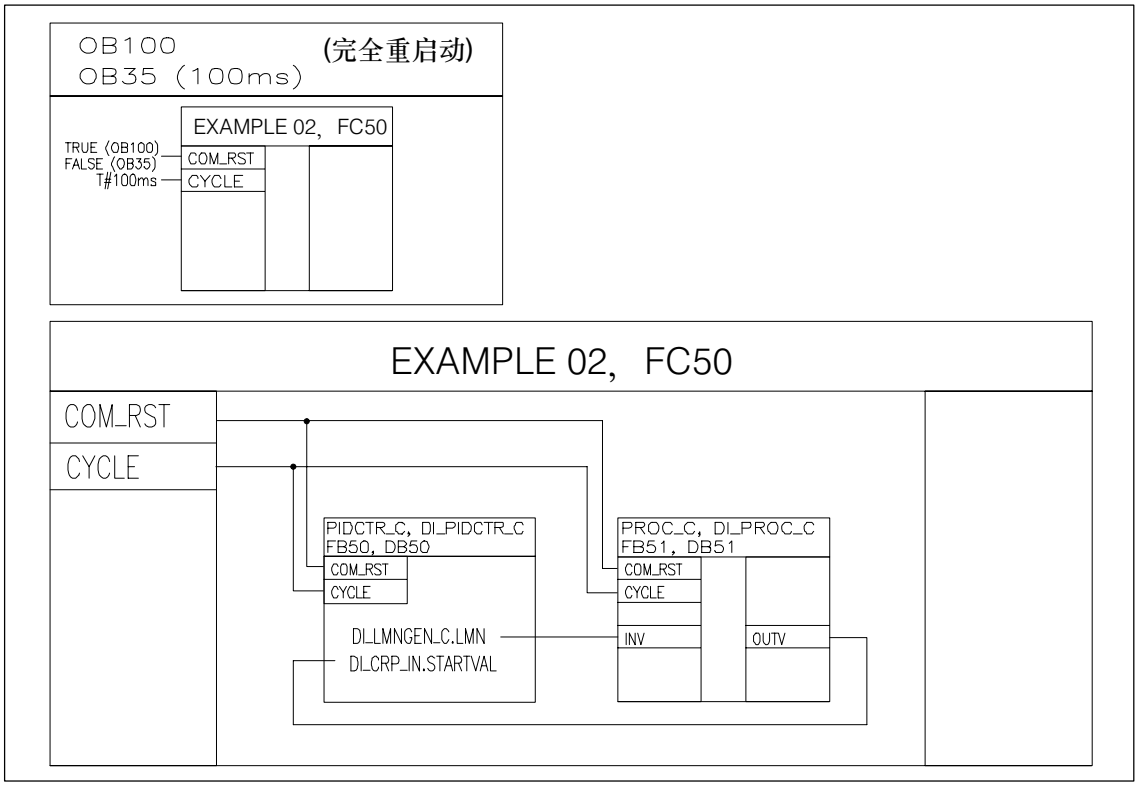


图3-6 实例2的块调用和互连

3.3.1 PIDCTR_C: 带有用于集成执行器的连续输出的固定设置值控制器

应用

PIDCTR_C块用作带有连续输出的固定设置值控制器。图3-7给出了PIDCTR_C的块互连。

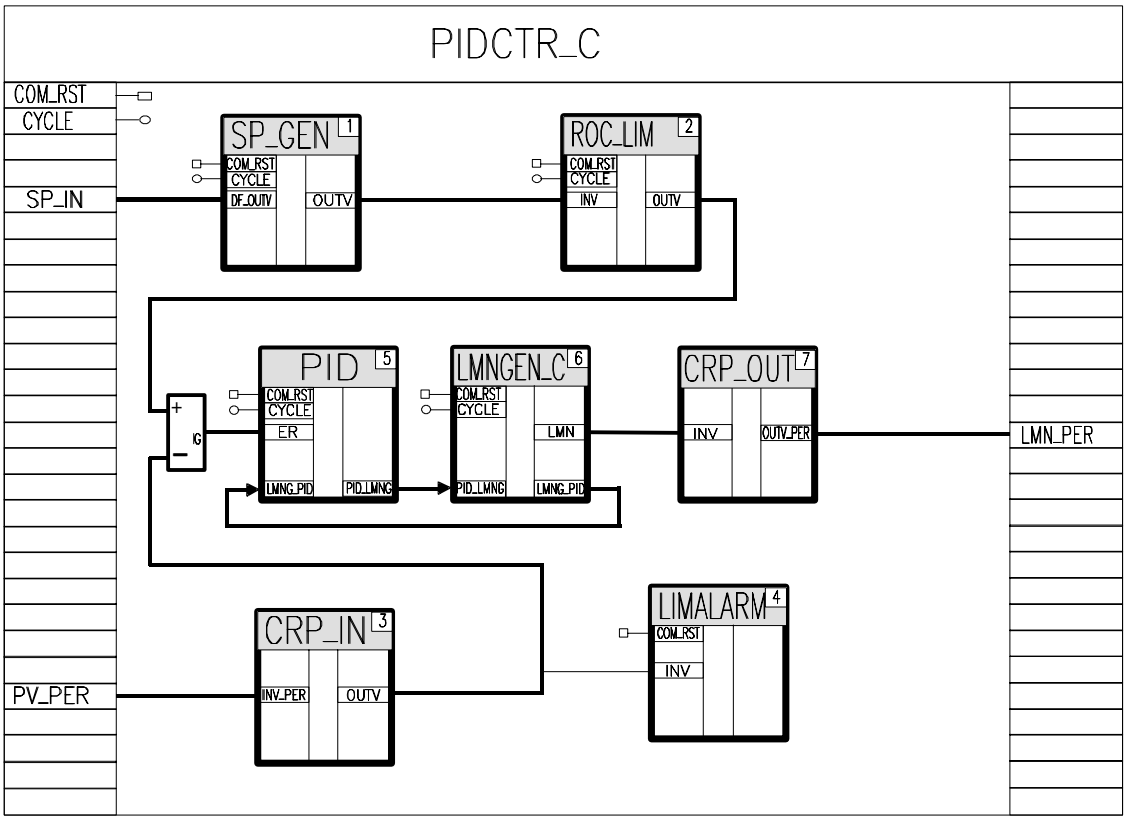


图3-7 PIDCTR_C的块互连

功能描述

设定值发生器SP_GEN设置设定值，该设定值的变化速率受限制器ROC_LIM的限制。CRP_IN将外围设备过程变量转换成浮点值，限制值监视器LIMALARM对其进行监视，检查它是否超出了所选择的限制值。偏差信号发送到PID算法。调节值处理块LMNGEN_C生成模拟调节变量LMN，该变量随后被CRP_OUT转换成外围设备格式。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.3.2 PROC_C: 用于连续控制器的自调整处理

应用

块PROC_C模拟一个3阶时间延迟。

图3-8给出了PROC_C的方框图。

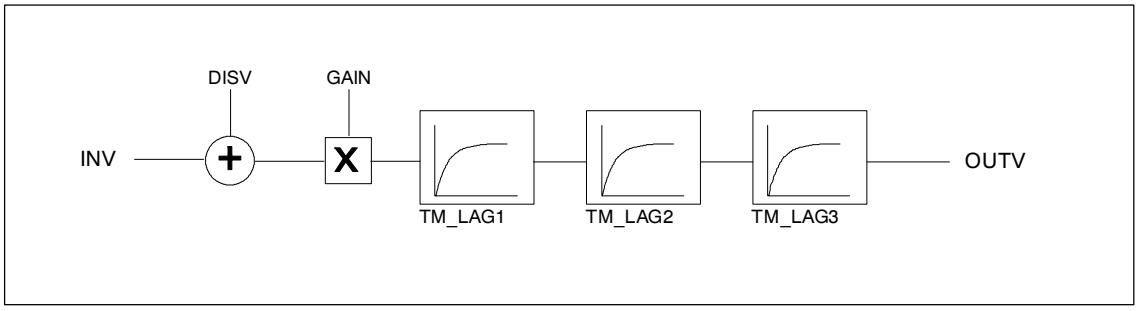


图3-8 PROC_C的方框图

功能描述

此块模拟三个串级连接的1阶时间延迟。始终在输入INV上添加干扰变量**DISV**。

完全重启动

在完全重启动期间，输出变量**OUTV**和内部存储的数值都设置为0。

3.4 实例3：带有开关输出的固定设置值控制器，用于带有过程模拟的比例执行器

总览

实例3的名称为EXAMPLE03，它包含了一个具有脉宽调制的连续PID控制器和一个模拟的工业过程。

控制闭环

图3-9给出了实例3的完整控制回路。

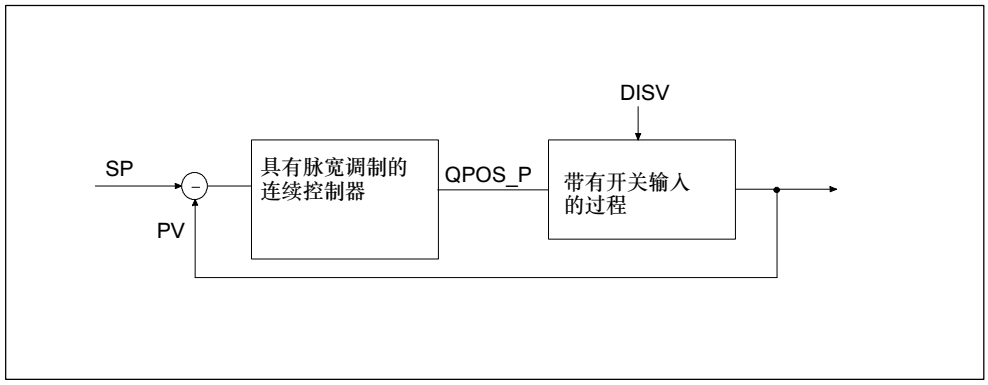


图3-9 实例3的控制回路

块调用和互连

图3-10给出了实例3的块调用和互连。

注意

必须通过“HW Konfig：硬件配置”将OB35的循环时间设置为10 ms。

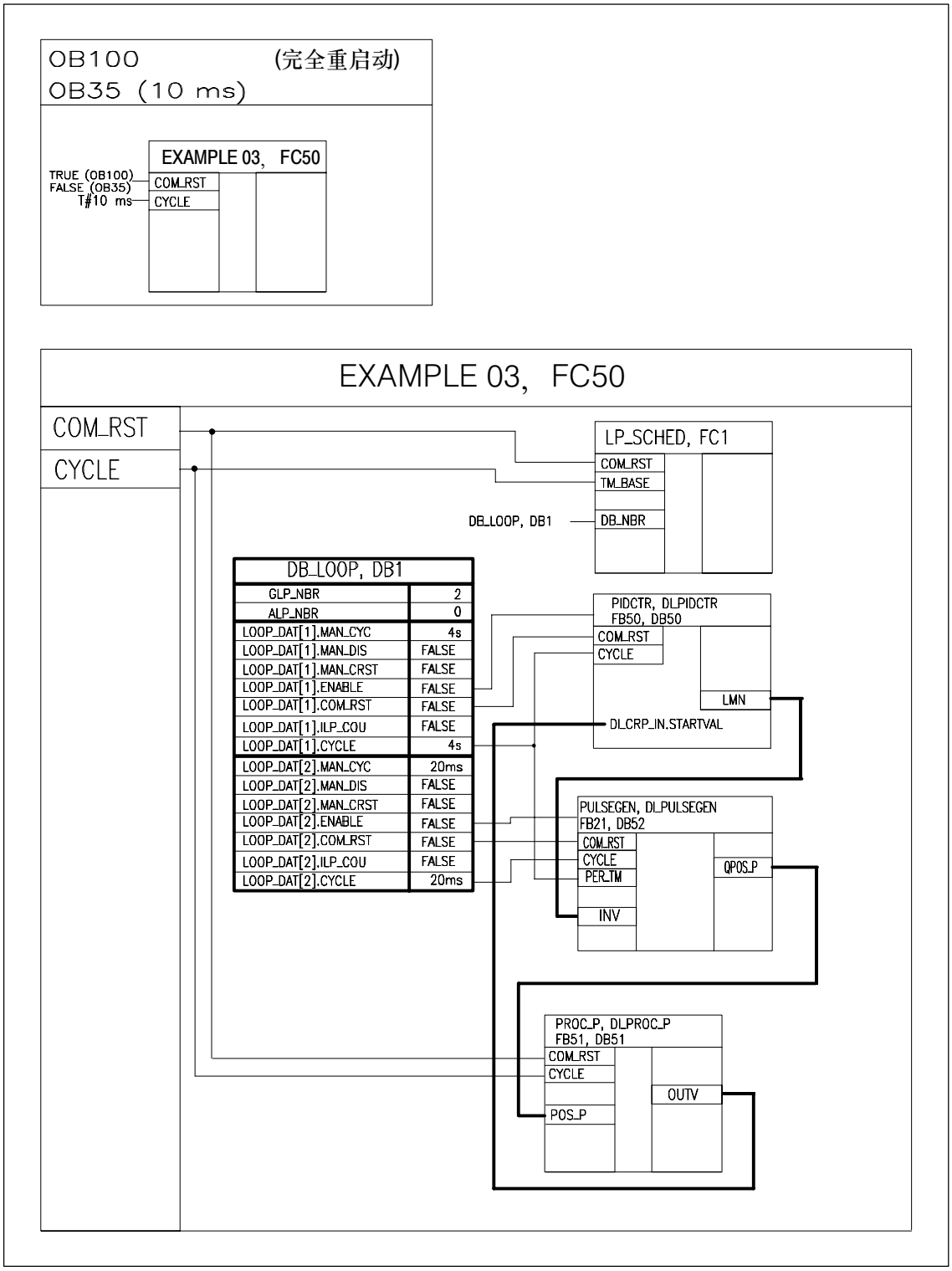


图3-10 实例3的块调用和互连

3.4.1 PIDCTR: 带有脉冲发生器的连续控制器的主控制器

应用

块PIDCTR实现了一个带有连续输出的PID控制器。它用于在脉冲中断控制器内计算模拟调节值。还可以在比率控制、混合控制和级联控制中用作主控制器。图3-11给出了PIDCTR的块互连。

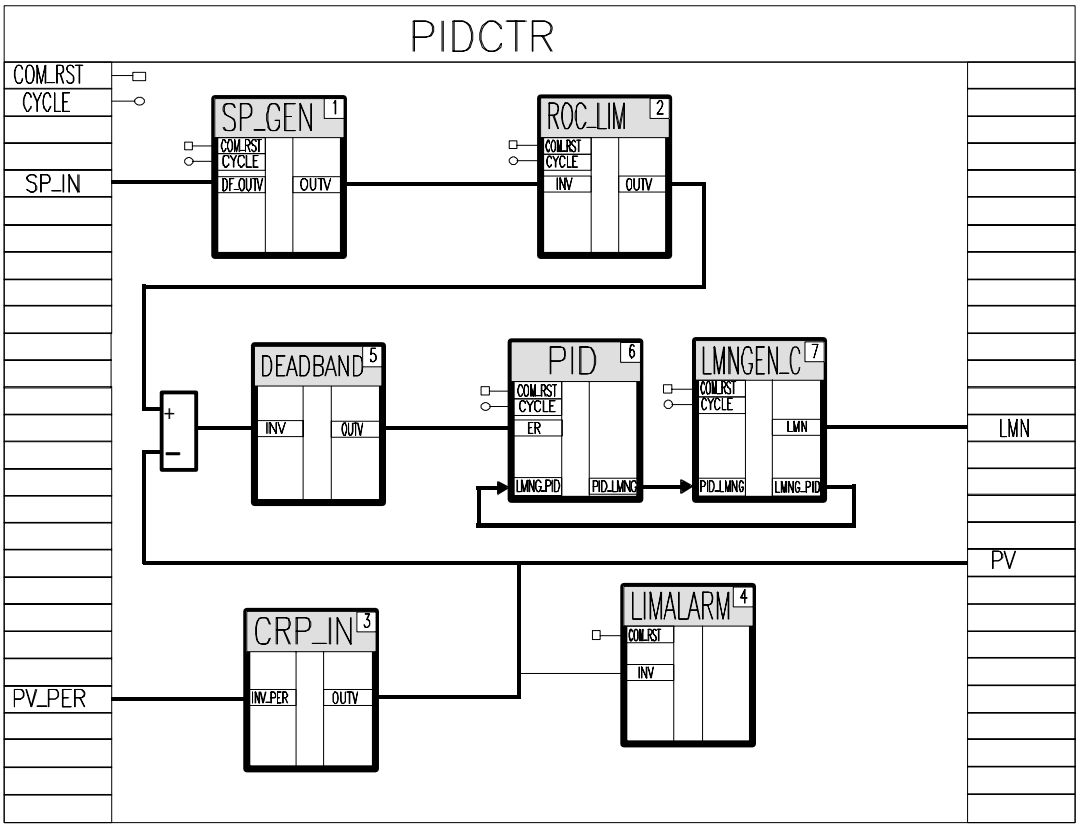


图3-11 PIDCTR的块互连

功能描述

设定值发生器SP_GEN设置设定值，该设定值的变化速率受限制器ROC_LIM的限制。CRP_IN将外围设备过程变量转换成浮点值，限制值监视器LIMALARM对其进行监视，检查它是否超出了所选择的限制值。偏差信号发送到PID算法。调节值处理块LMNGEN_C生成模拟调节值LMN。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.4.2 PROC_P: 带有脉冲发生器的连续控制器的过程

应用

PROC_P块模拟一个带有数字输入和3阶时间延迟的连续执行器阀。

图3-12给出了PROC_P的方框图。

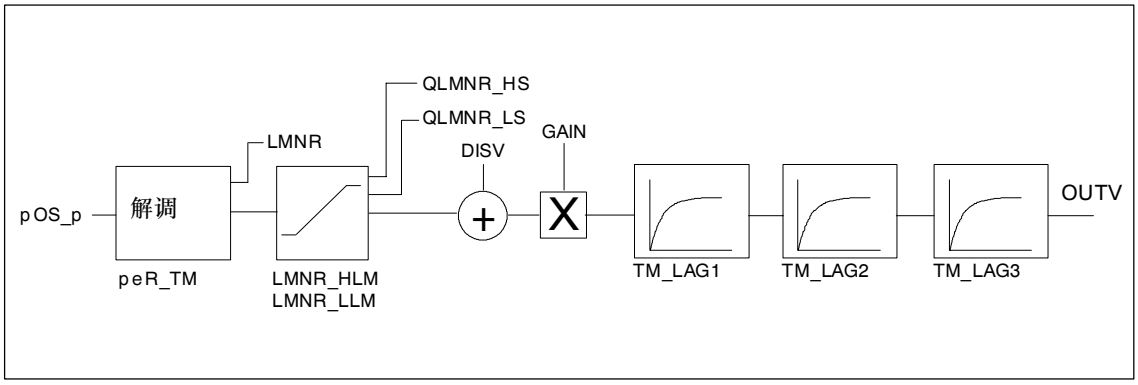


图3-12 PROC_P的方框图

功能描述

功能块在添加了干扰变量之后，将脉宽调制的二进制输入值转换成连续模拟量值，并通过1阶时间延迟来延迟输出信号。

完全重启动

在完全重启动期间，输出变量**OUTV**和内部存储的数值都设置为0。

3.5 实例4: 单回路比率控制器(RATIOCTR)

总览

实例4的名称为EXAMPLE04，它是一个单回路比率控制器。此实例中没有模拟的过程。

控制闭环

图3-13给出了实例4的应用的完整控制回路。

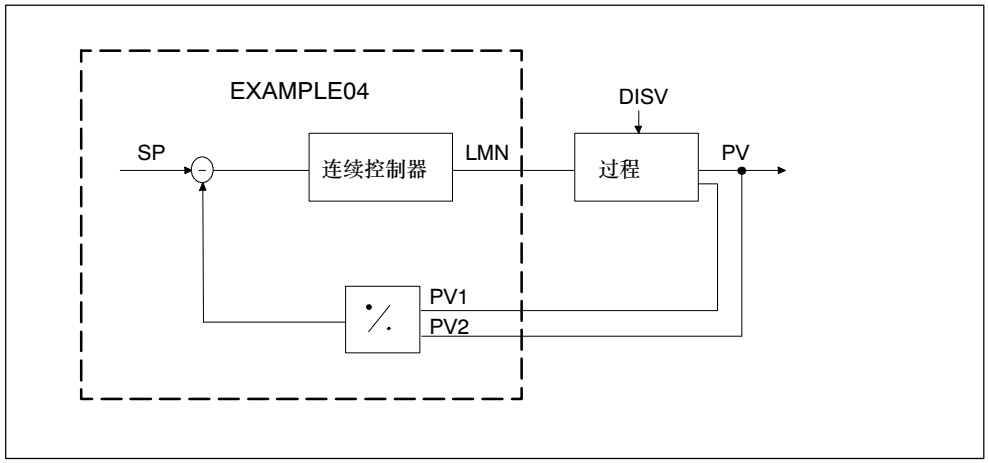


图3-13 实例4的控制回路

块调用

图3-14给出了实例4的块调用。

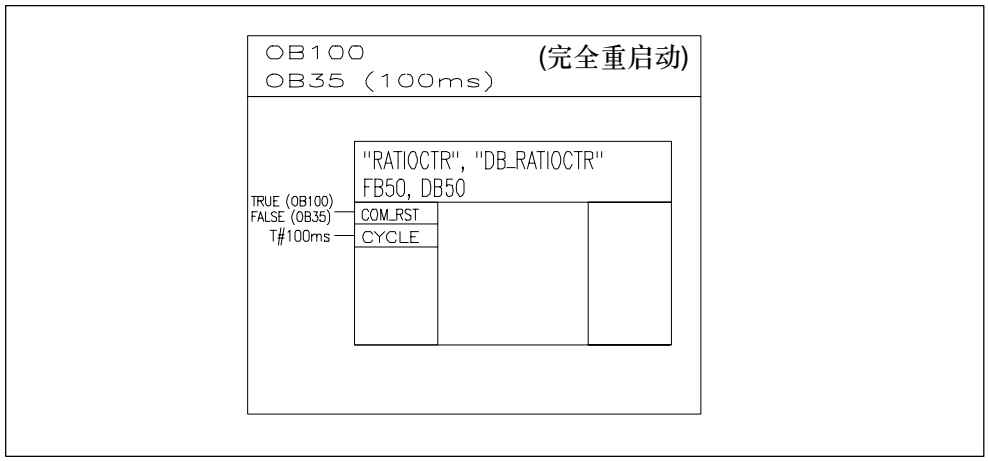


图3-14 实例4的块调用

应用

该块实现了一个用于连续执行器的单回路比率控制器。图3-15给出了RATIOCTR的块互连。

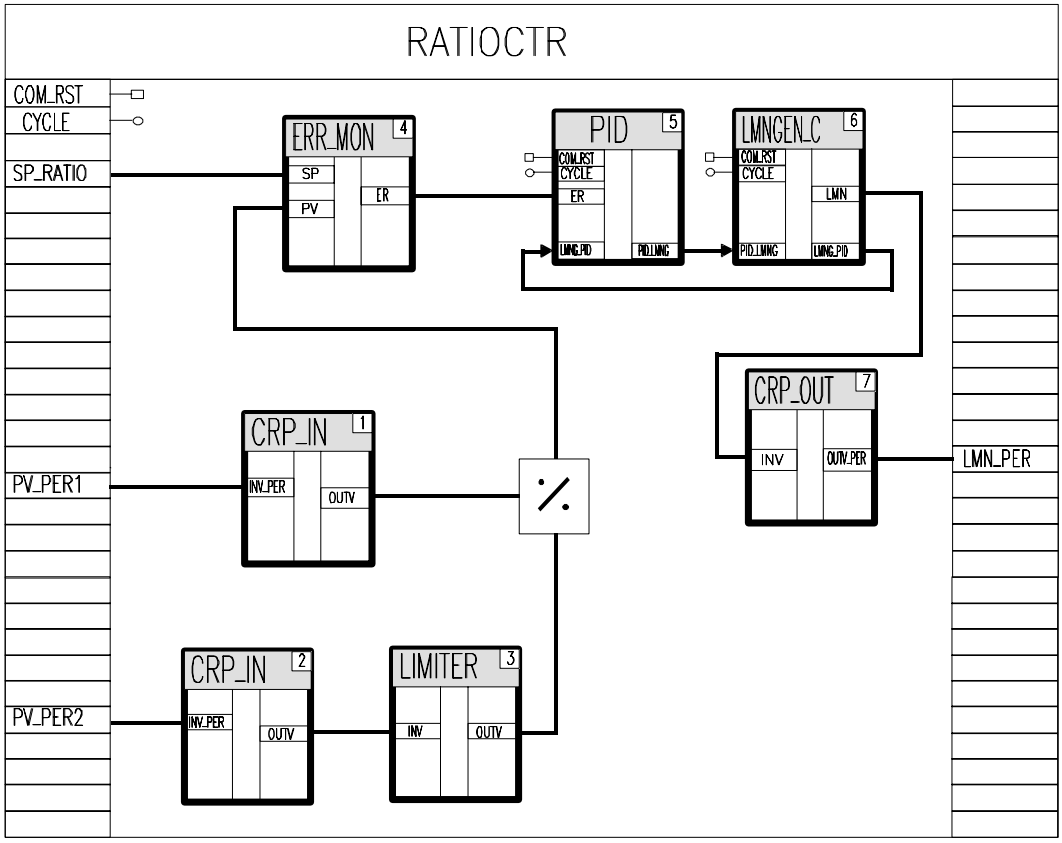


图3-15 RATIOCTR的块互连

功能描述

比率设定值是通过输入参数SP_RATIO设置的。CRP_IN将外围设备过程变量PV_PER1和PV_PER2转换成浮点值，然后生成比率值。浮点值PV_PER2的数值由LIMITER限制，这样就不会发生除数为零的情况。偏差信号发送到PID算法PID。调节值处理块LMNGEN_C生成模拟调节变量LMN，该变量随后被CRP_OUT转换成外围设备格式。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.6 实例5: 多回路比率控制器

总览

实例5的名称为EXAMPLE05，是一个多回路比率控制器。它包含了一个主控制器和三个次级控制器。此实例中没有模拟的工业过程。

控制闭环

图3-16给出了实例5的应用的完整控制回路。

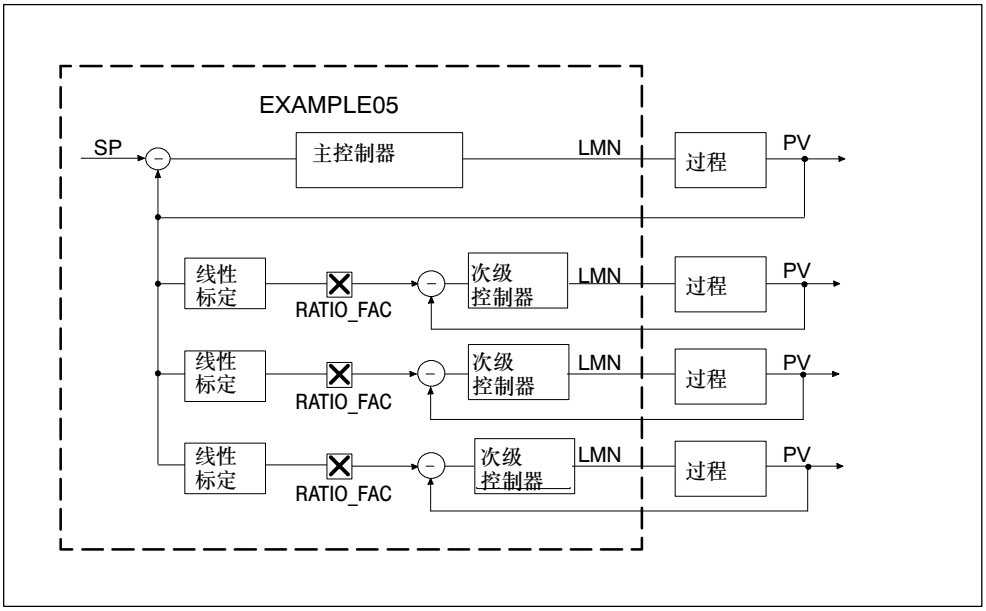


图3-16 实例5的控制回路

块调用和互连

图3-17给出了实例的块调用和互连。

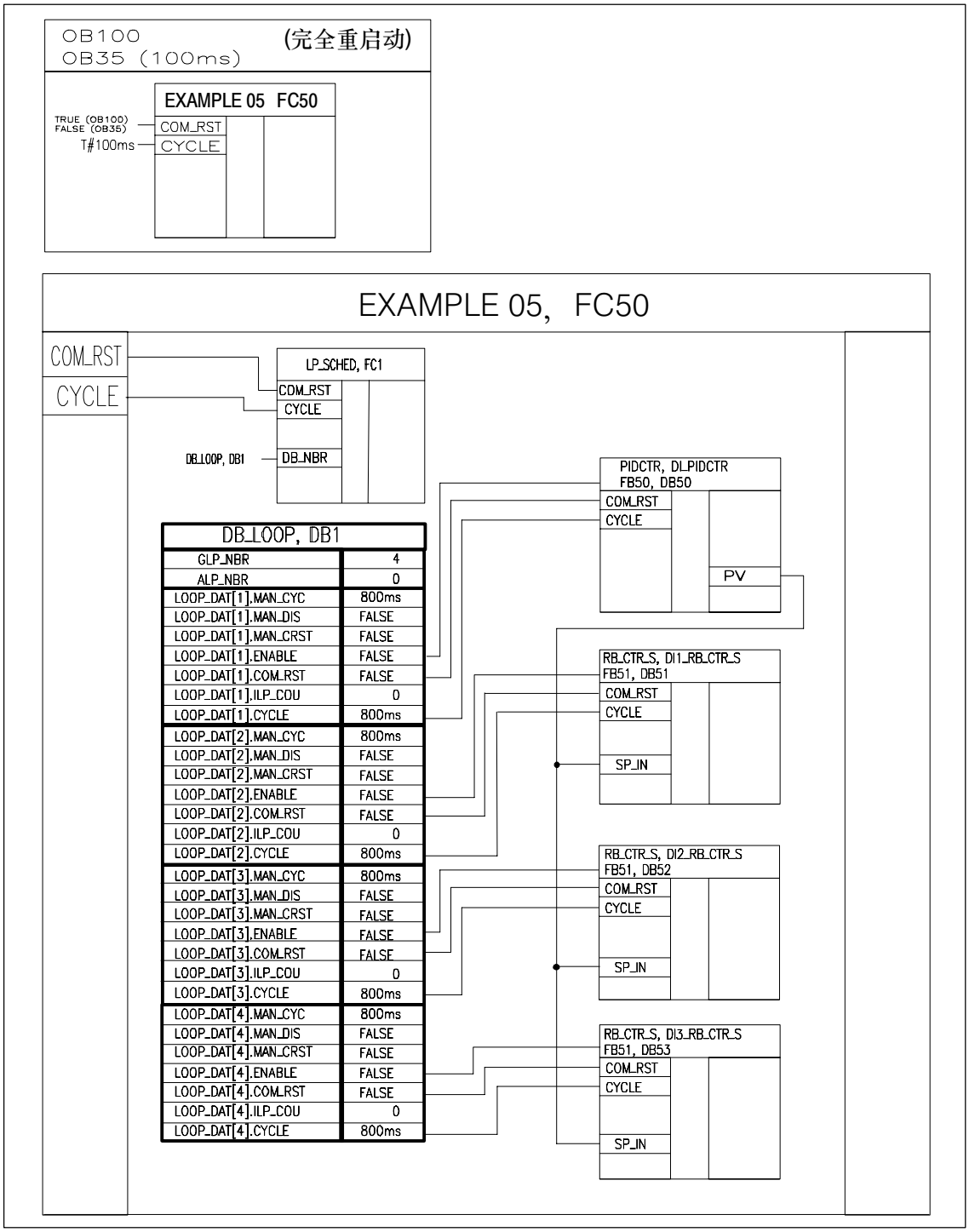


图3-17　　实例5的块调用和互连

主控制器

主控制器是来自实例3的块PIDCTR。它的功能描述在第3.4.1节(第3-14页)中。

次级控制器

次级控制器是块RB_CTR_S。此块是一个用于集成执行器的PID步进控制器，可以用作多回路比率或混合控制器中的次级控制器。图3-18给出了RB_CTR_S的块互连。

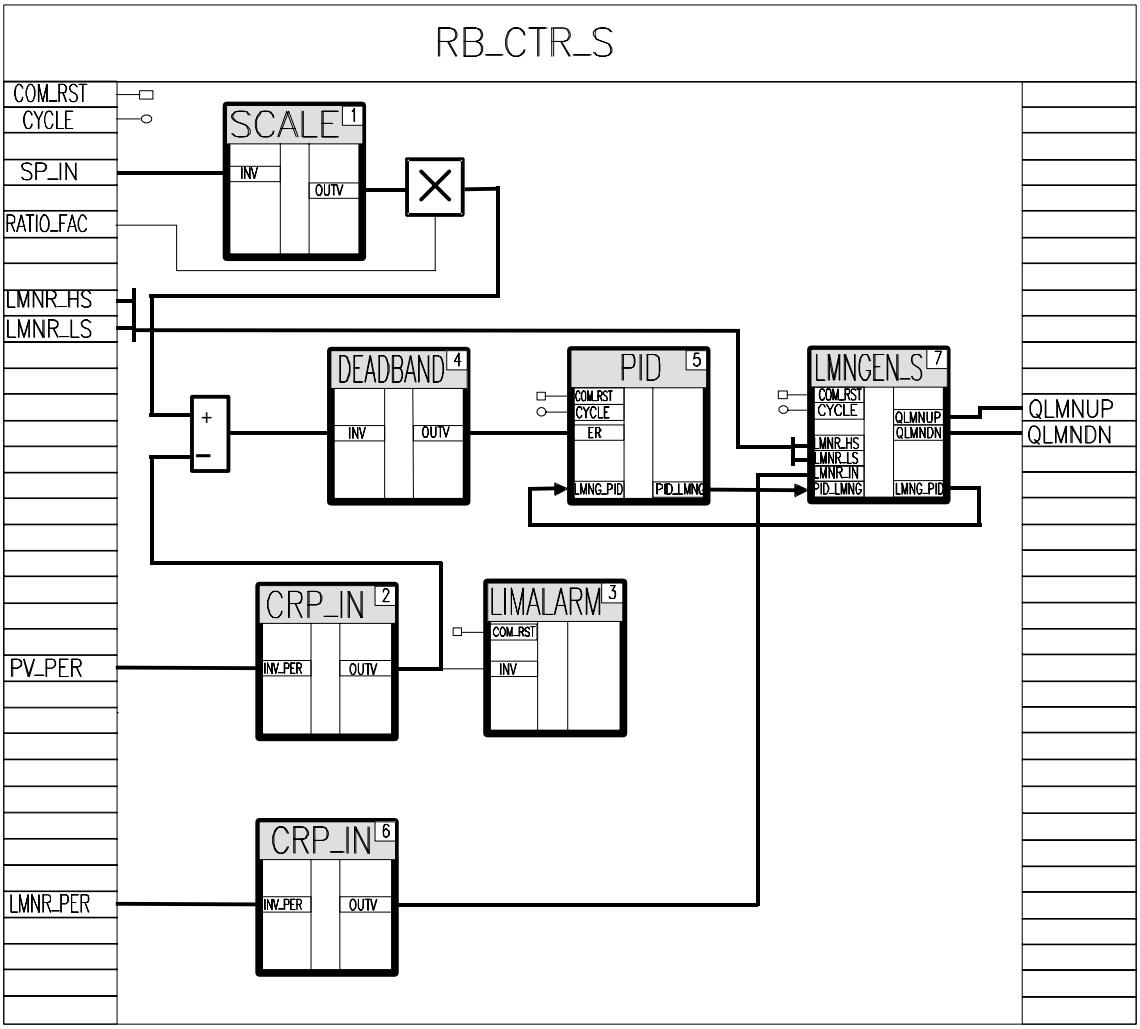


图3-18　　RB_CTR_S的块互连

次级控制器的功能描述

次级控制器RB_CTR_S的功能与实例1中的步进控制器PIDCTR_S (参见第3-4页)类似。通过线性标定因子，以及乘以所选择的比率或混合因子，将次级控制器的输入调整为与过程的输出相符。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.7 实例6: 混合控制器

总览

实例6的名称为EXAMPLE06，它是一个混合控制器。它包含了一个主控制器和三个次级控制器。此实例中没有模拟的工业过程。

控制闭环

图3-19给出了实例6的应用的完整控制回路。

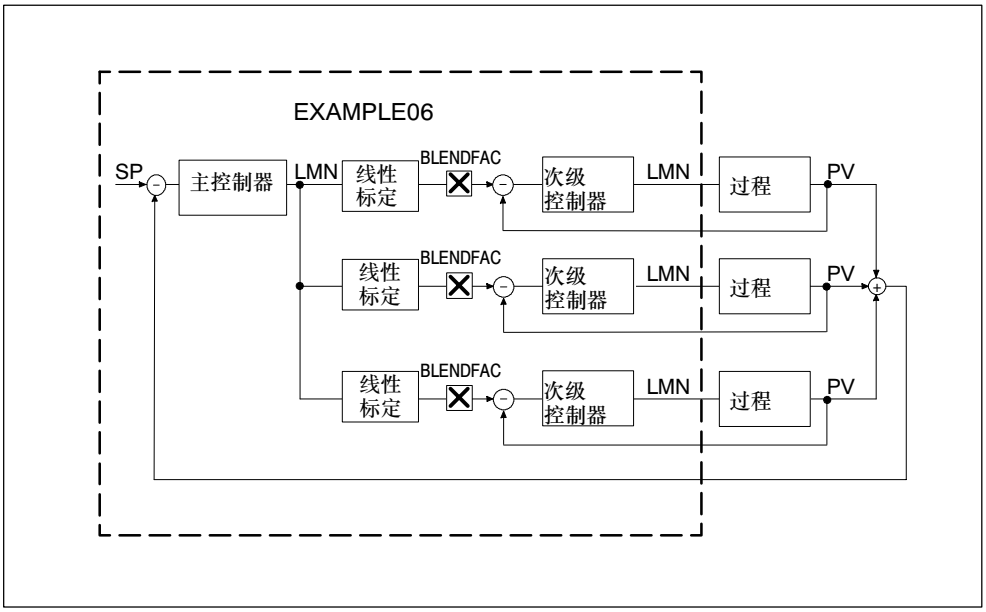


图3-19 实例6的控制回路

块调用和互连

图3-20给出了实例6的块调用和互连。

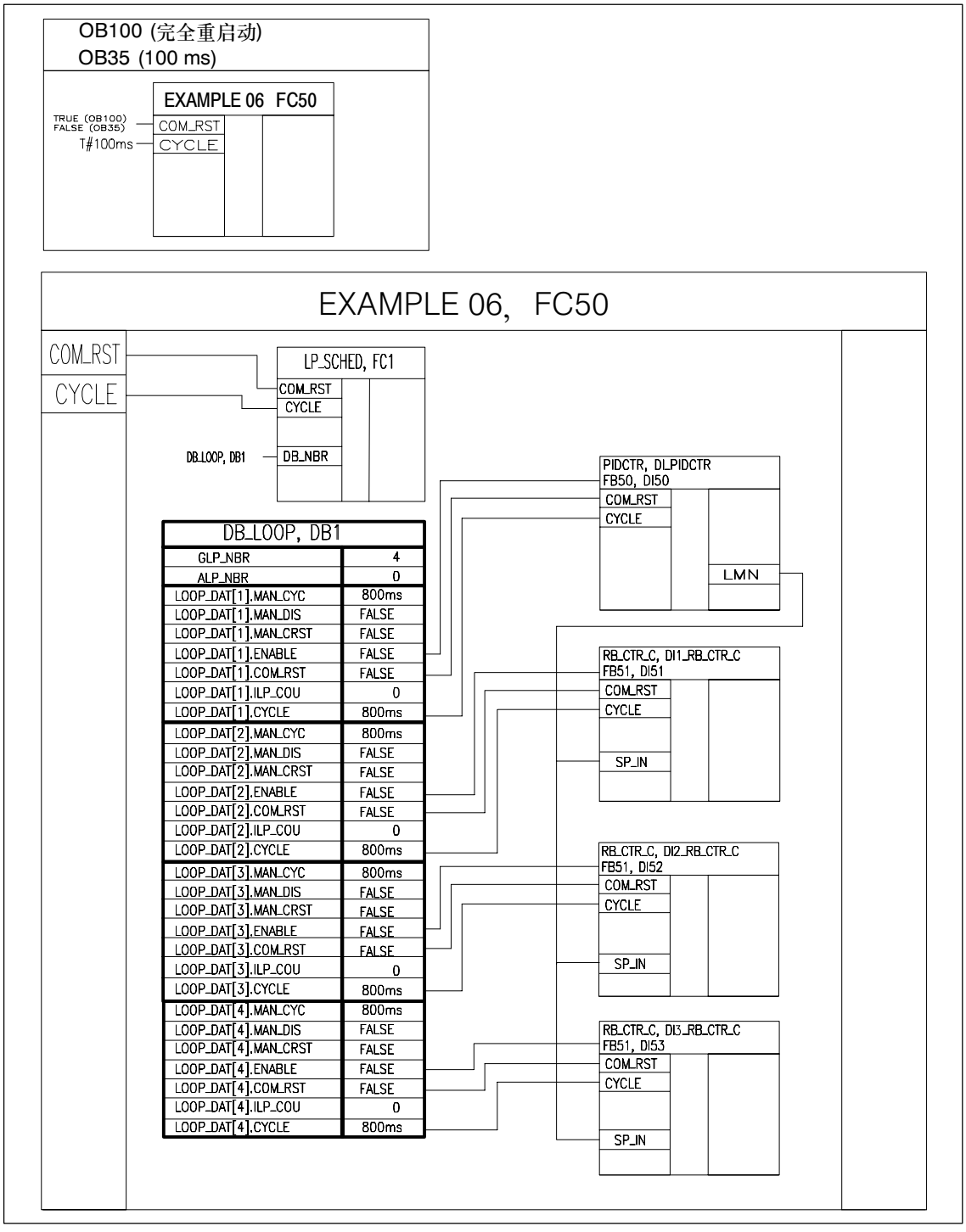


图3-20　　实例6的块调用和互连

主控制器

主控制器是来自实例3的块PIDCTR。它的功能描述在第3.4.1节(第3-14页)中。

次级控制器

次级控制器是块RB_CTR_C。此块是一个连续PID控制器，可以用作多回路比率或混合控制器中的次级控制器。图3-21给出了RB_CTR_C的块互连。

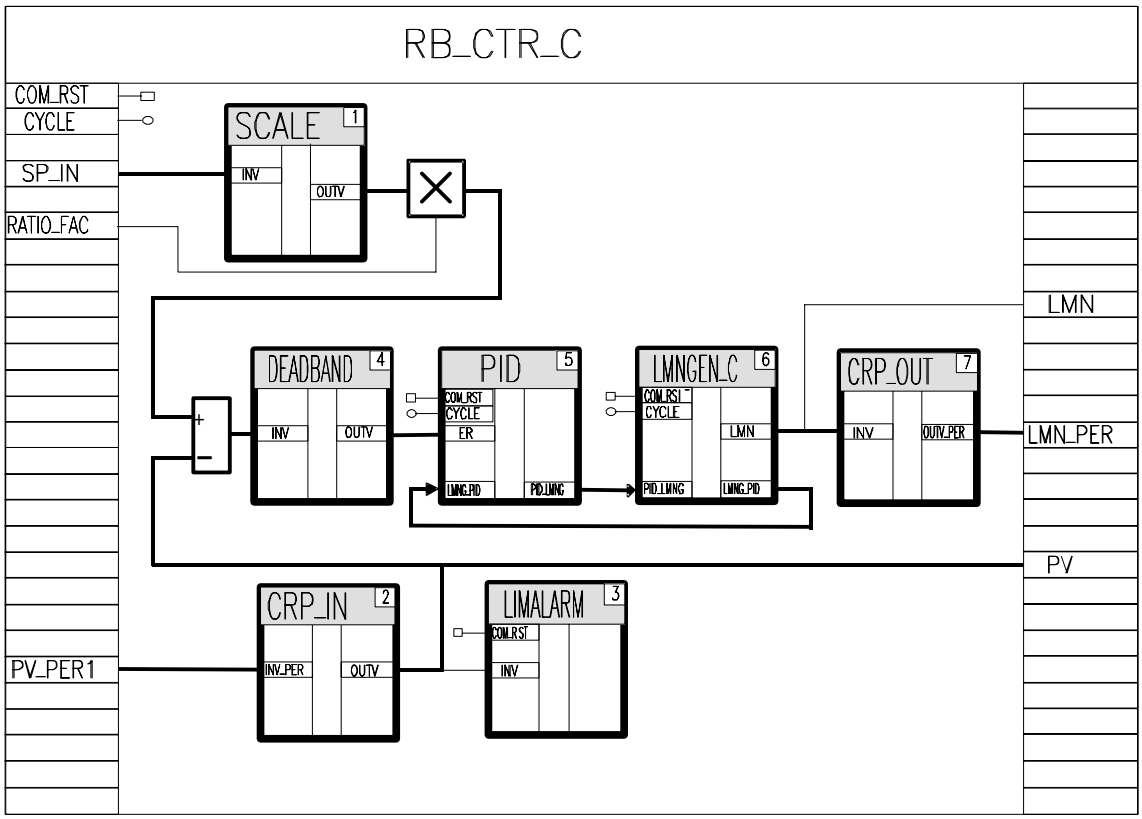


图3-21　　RB_CTR_C的块互连

次级控制器的功能描述

次级控制器RB_CTRL_C的功能与实例2 (参见第3-9页)中的连续PID控制器PIDCTR_C类似。通过线性标定因子，以及乘以所选择的比率或混合因子，将次级控制器的输入调整为与过程的输出相符。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.8 实例7: 级联控制

总览

实例7的名称为EXAMPLE07，它是一个级联控制。它包含了一个主控制器和一个次级控制器。此实例中没有模拟的工业过程。

控制闭环

图3-22给出了实例7的应用的完整控制回路。

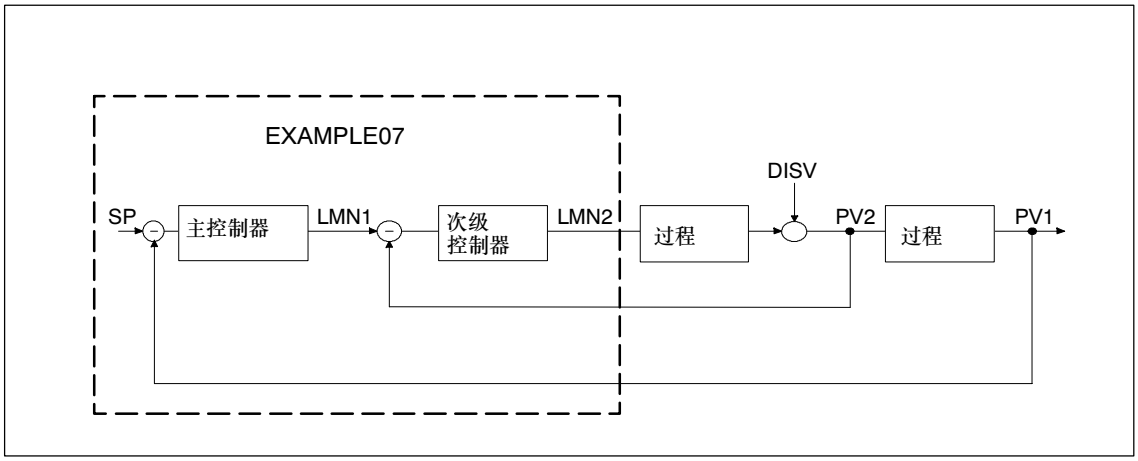


图3-22 实例7的控制回路

块调用和互连

图3-23给出了实例7的块调用和互连。

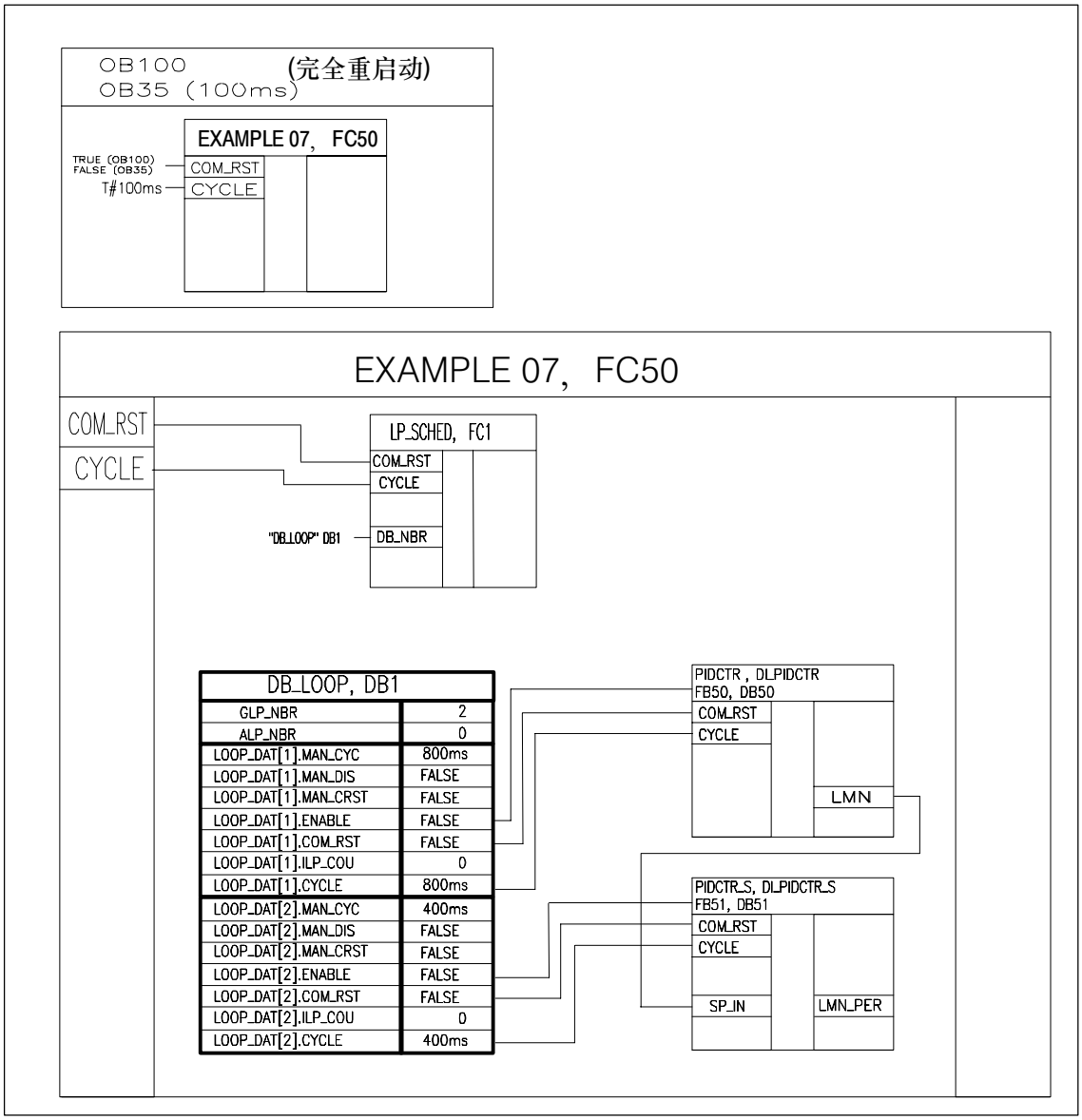


图3-23 实例7的块调用和互连

主控制器

主控制器是来自实例3的块PIDCTR。它的功能在3.4.1节中作了描述。

次级控制器

次级控制器是来自实例1的块PIDCTR_S。它的功能在3.2.1节(第3-6页)中作了描述。

3.9 实例8: 带有预控制器(CTRC_PRE)的控制器

总览

实例8的名称为EXAMPLE08，它是一个带有预控制器的控制器。此实例中没有模拟的工业过程。

控制闭环

图3-24给出了实例8的应用的完整控制回路。

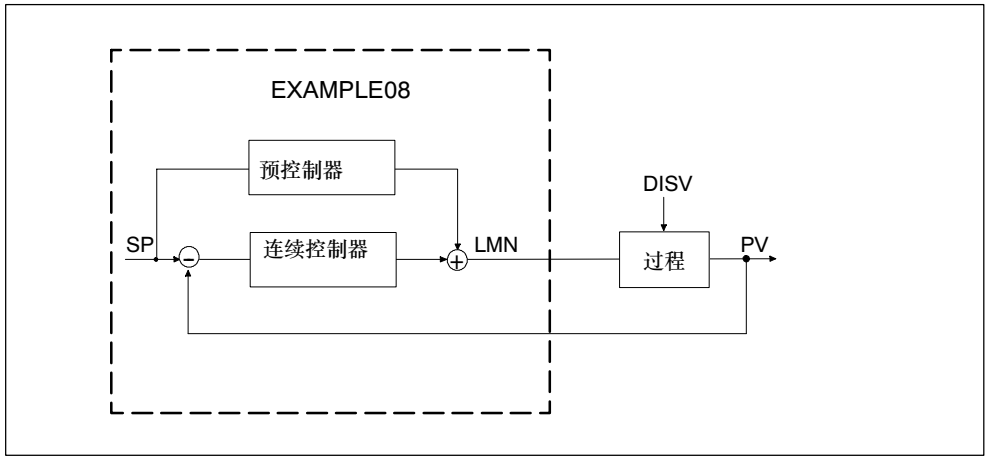


图3-24 实例8的控制回路

块调用

图3-25给出了实例8的块调用。

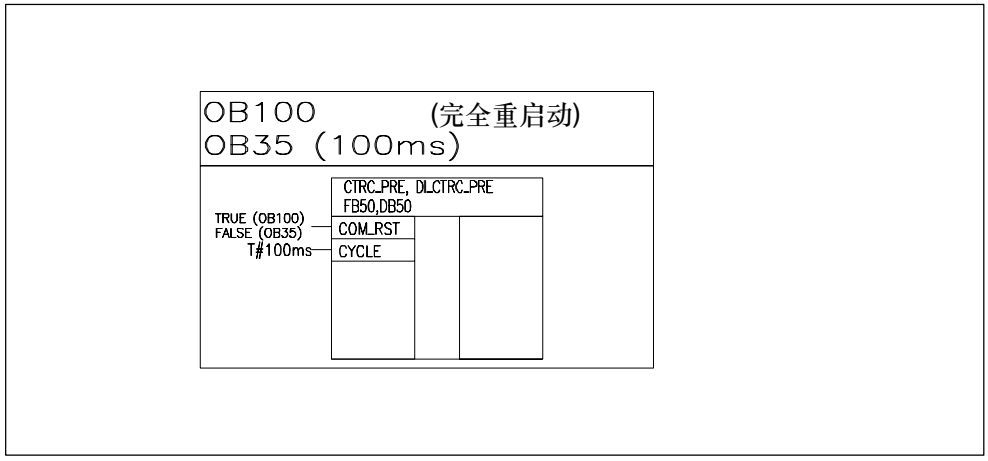


图3-25 实例8的块调用

应用

块CTRC_PRE是一个带有预控制器的连续PID控制器。图3-26给出了CTRC_PRE的块互连。

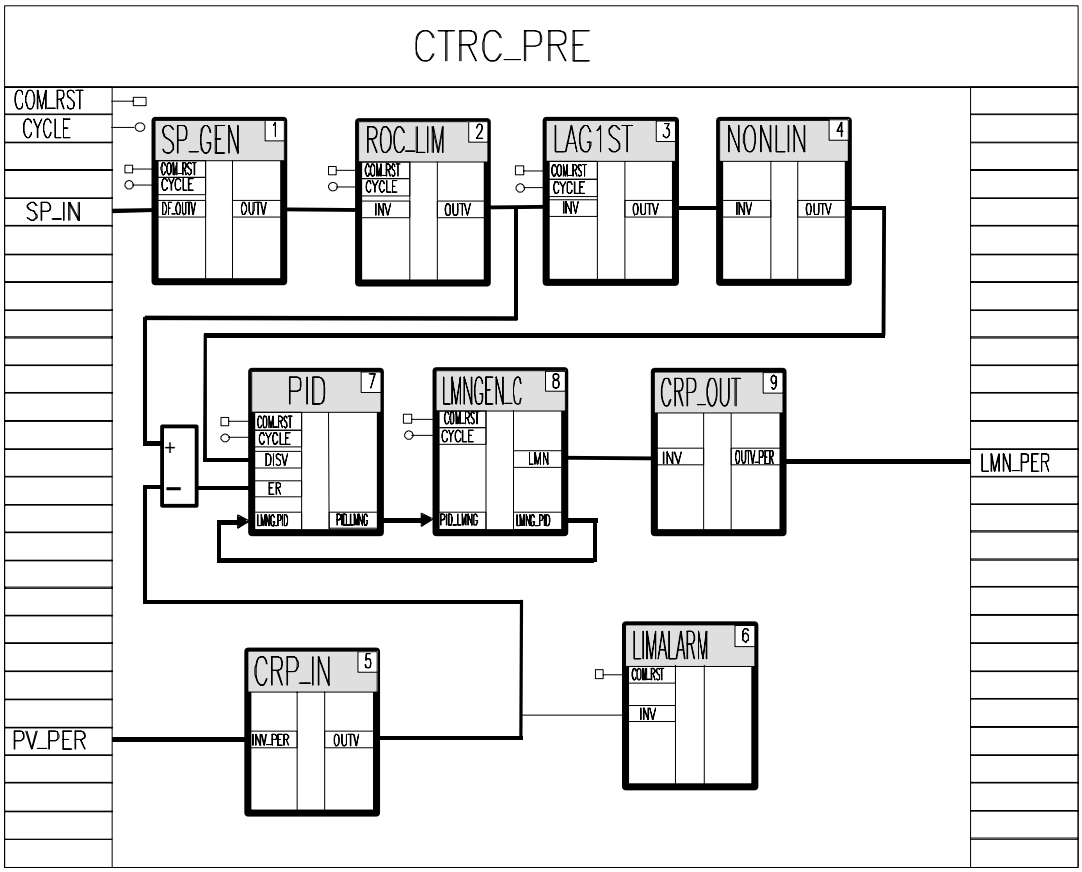


图3-26 CTRC_PRE的块互连

功能描述

带有预控制器的控制器的功能与实例2中的带有连续输出的固定设定值控制器PIDCTR_C类似。预控制器包含了一个具有静态、非线性特征的在结构上与PID算法平行的1阶时间延迟。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.10 实例9: 带有前馈控制的控制器(CTR_C_FF)

总览

实例9的名称为EXAMPLE09，它是一个带有前馈控制的控制器。此实例中没有模拟的工业过程。

控制闭环

图3-27给出了实例9的应用的完整控制回路。

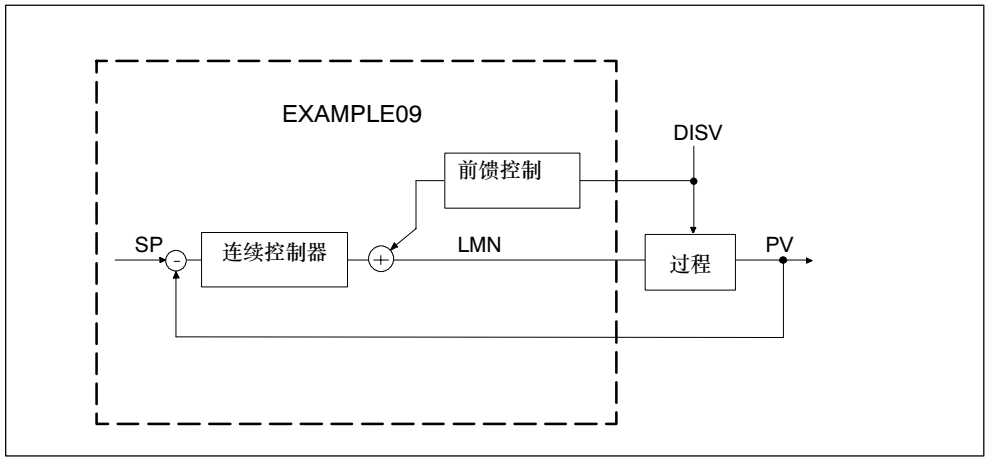


图3-27 实例9的控制回路

块调用

图3-28给出了实例9的块调用。

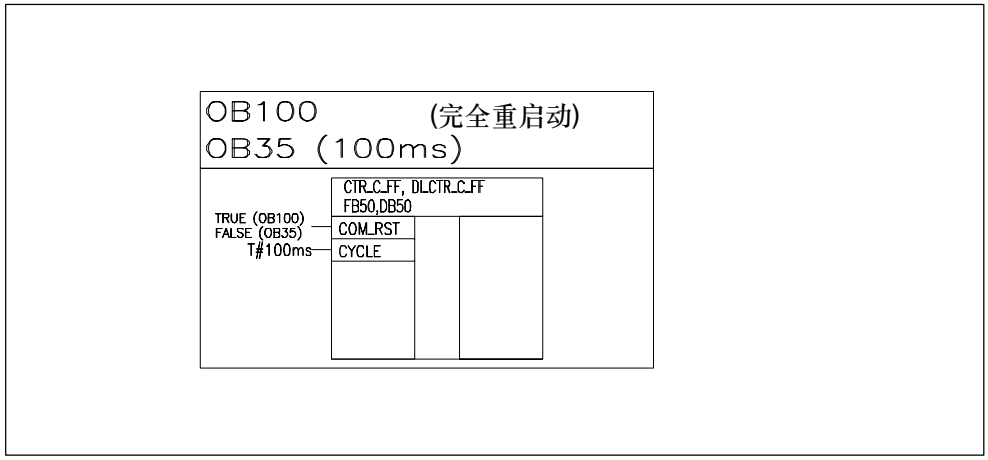


图3-28 实例9的块调用

应用

块CRT_C_FF是一个带有前馈控制的用于连续执行器的PID控制器。图3-29给出了CTR_C_FF的块互连。

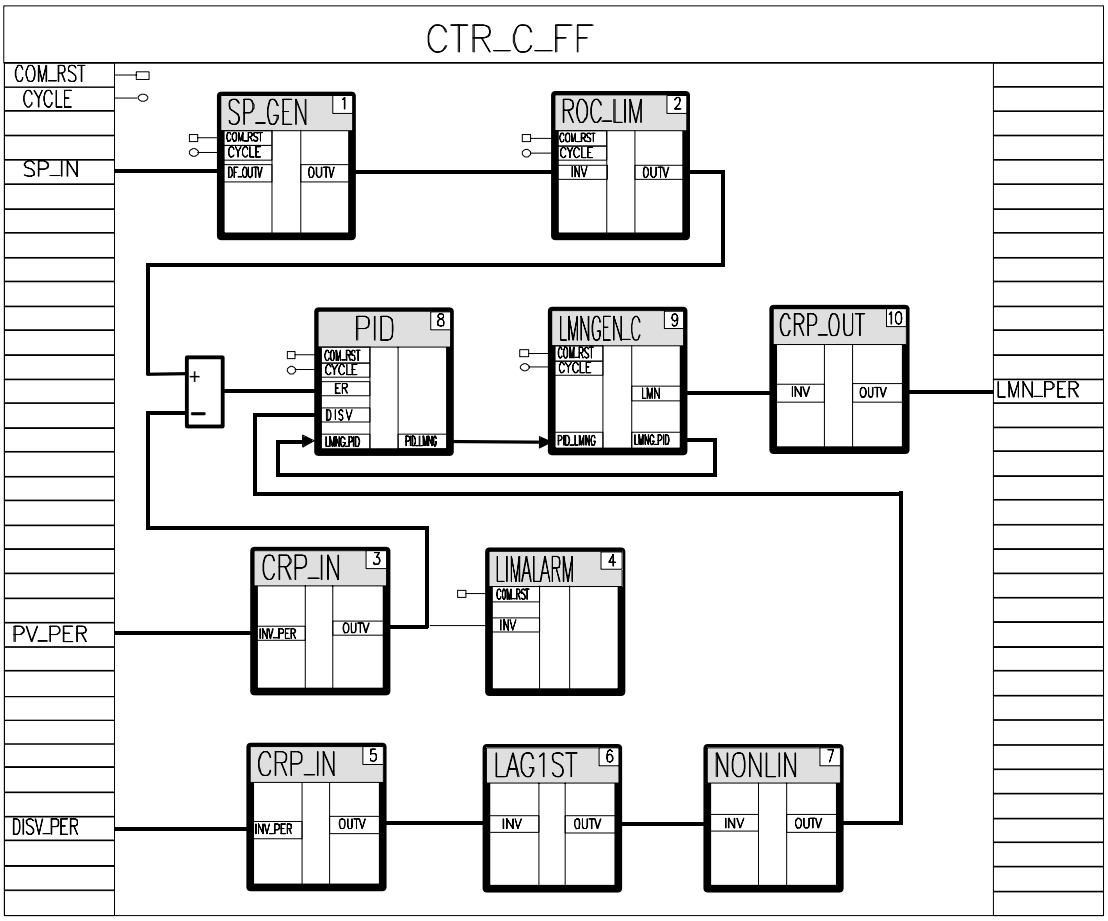


图3-29　　CTR_C_FF的块互连

功能描述

设定值发生器SP_GEN设置设定值，该设定值的变化速率受限制器ROC_LIM的限制。CRP_IN将外围设备过程变量转换成浮点值，限制值监视器LIMALARM对其进行监视，检查它是否超出了所选择的限制值。偏差信号发送到PID算法。CRP_IN将外围设备干扰值转换成浮点值，并通过LAG1ST进行滤波和线性化。调节值处理块LMNGEN_C生成模拟调节变量LMN，该变量随后被CRP_OUT转换成外围设备格式。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.11 实例10: 分段控制器(SPLITCTR)

总览

实例10的名称为EXAMPLE10，是一个分段控制器。此实例中没有模拟的工业过程。

控制闭环

图3-30给出了实例10的应用的完整控制回路。

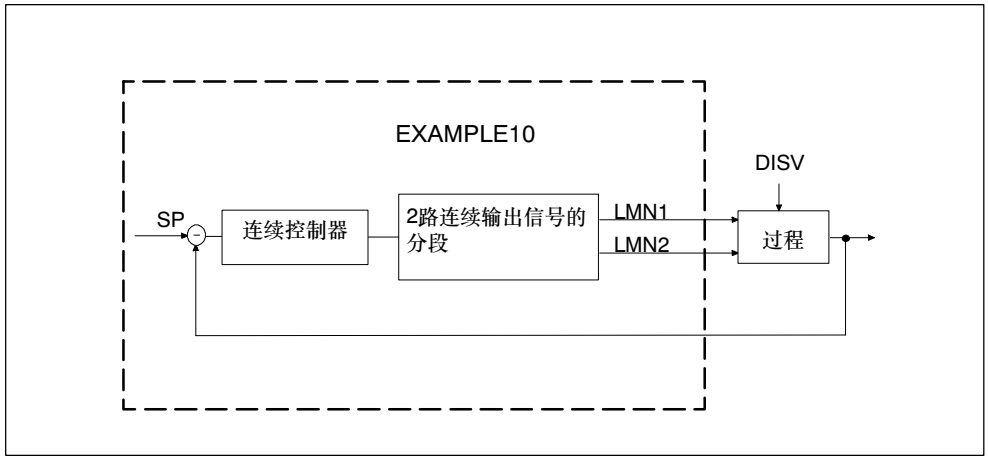


图3-30 实例10的控制回路

块调用

图3-31给出了实例10的块调用。

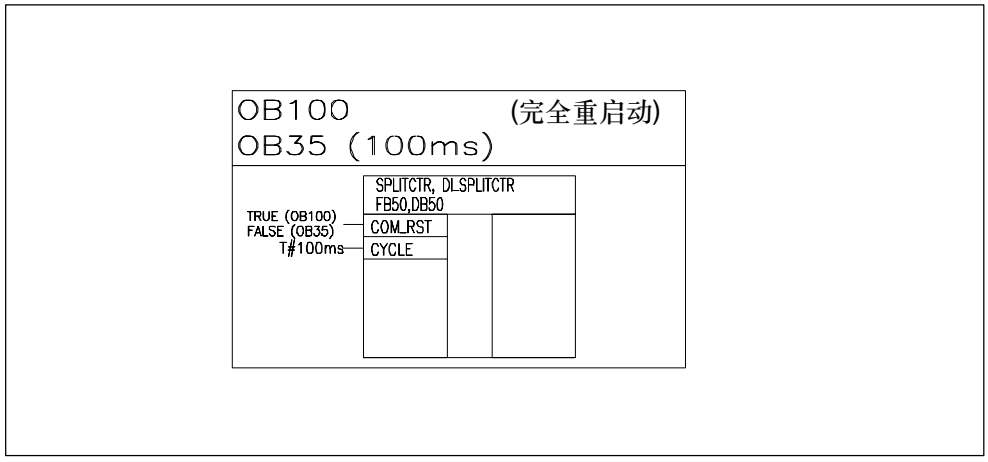


图3-31 实例10的块调用

应用

块SPLITCTR是一个带有用于2个连续执行器的分段的PID控制器。图3-32给出了SPLITCTR的块互连。

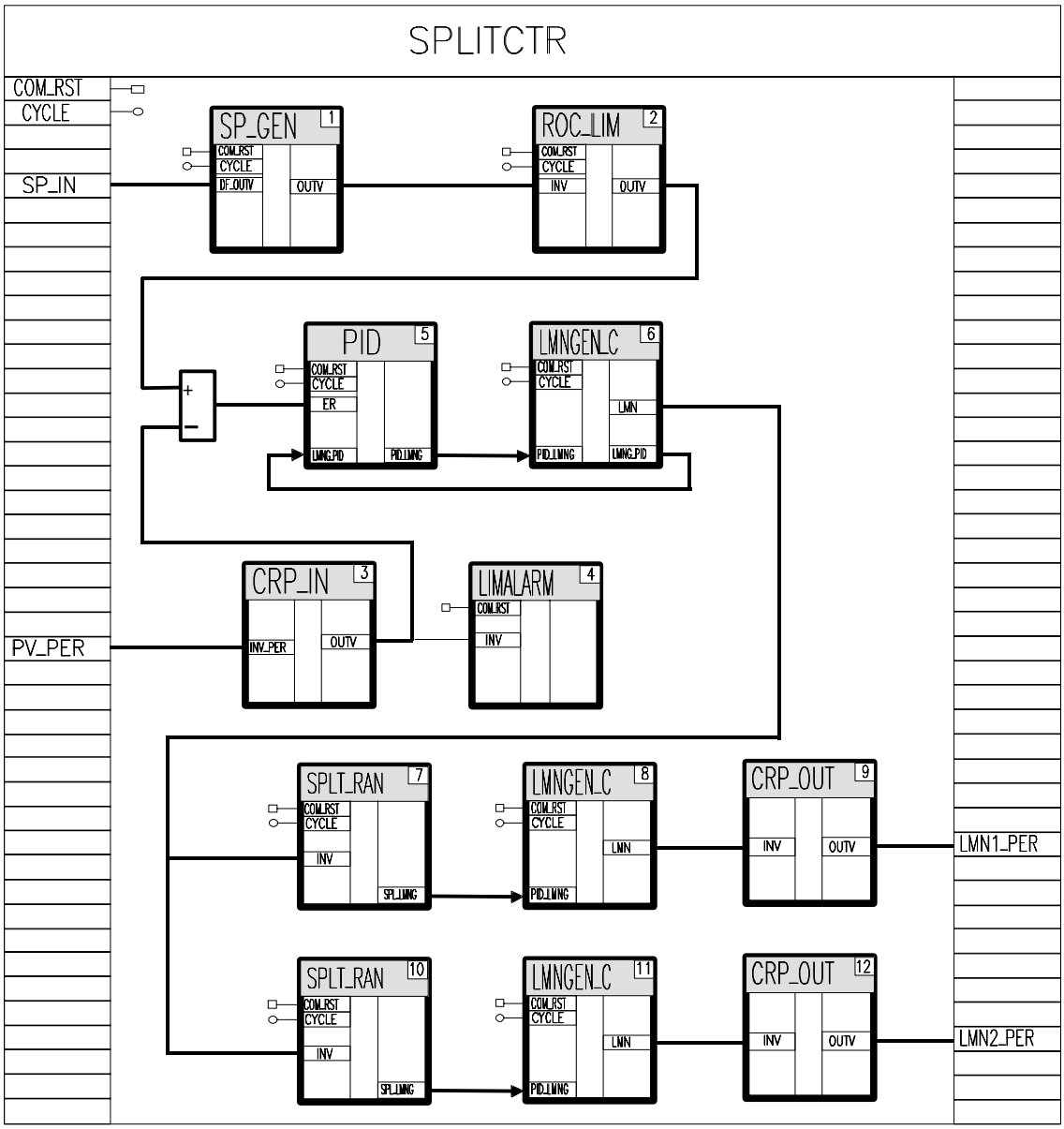


图3-32 SPLITCTR的块互连

功能描述

设定值发生器SP_GEN设置设定值，该设定值的变化速率受限制器ROC_LIM的限制。CRP_IN将外围设备过程变量转换成浮点值，限制值监视器LIMALARM对其进行监视，检查它是否超出了所选择的限制值。偏差信号发送到PID算法。调节值处理块LMNGEN_C生成模拟调节值LMN。调节值范围被两个SPLT_RAN块分成两个范围。对于每个范围，LMNGEN_C计算模拟调节变量，该变量稍后被CRP_OUT转换成外围设备格式。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.12 实例11：倍率控制器(OVR_CTR)

总览

实例11的名称为EXAMPLE11，它是一个倍率控制器。此实例中没有模拟的工业过程。

控制闭环

图3-33给出了实例11的应用的完整控制回路。

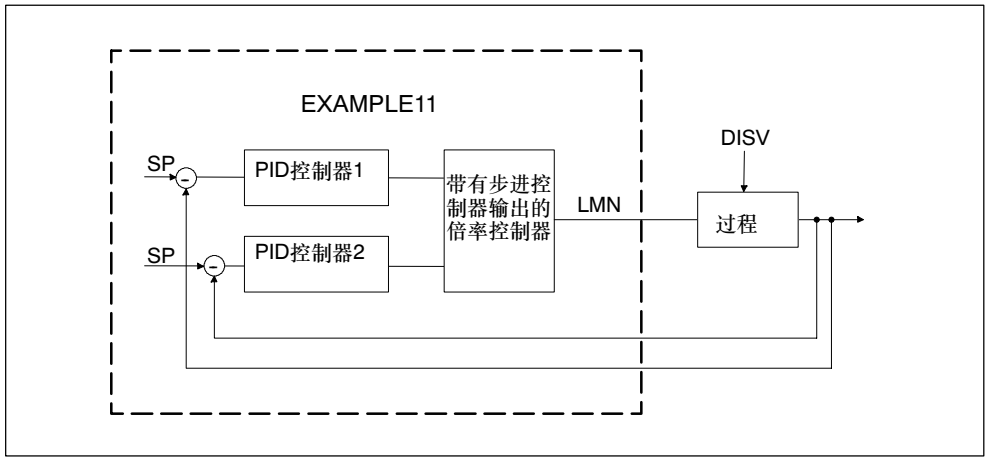


图3-33 实例11的控制回路

块调用

图3-34给出了实例11的块调用。

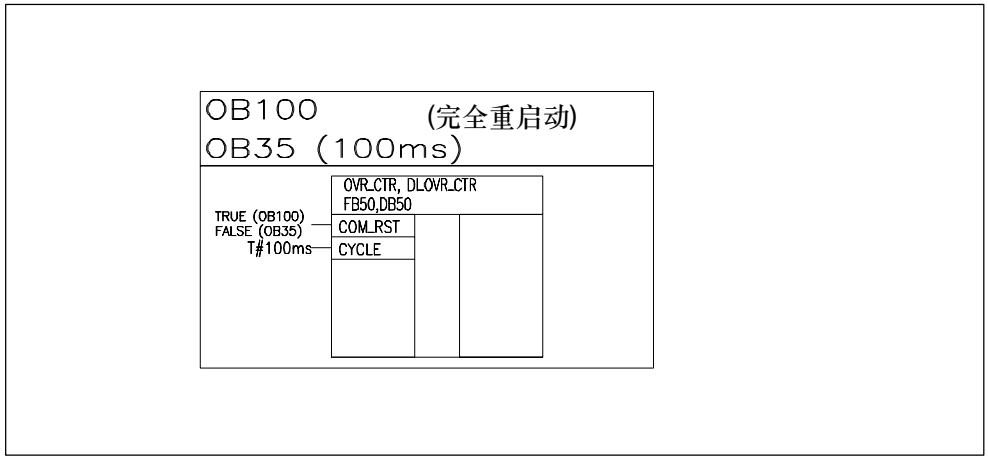


图3-34 实例11的块调用

应用

块OVR_CTR是一个倍率控制器。两个PID控制器连接到一个步进控制器输出。图3-35给出了OVR_CTR的块互连。

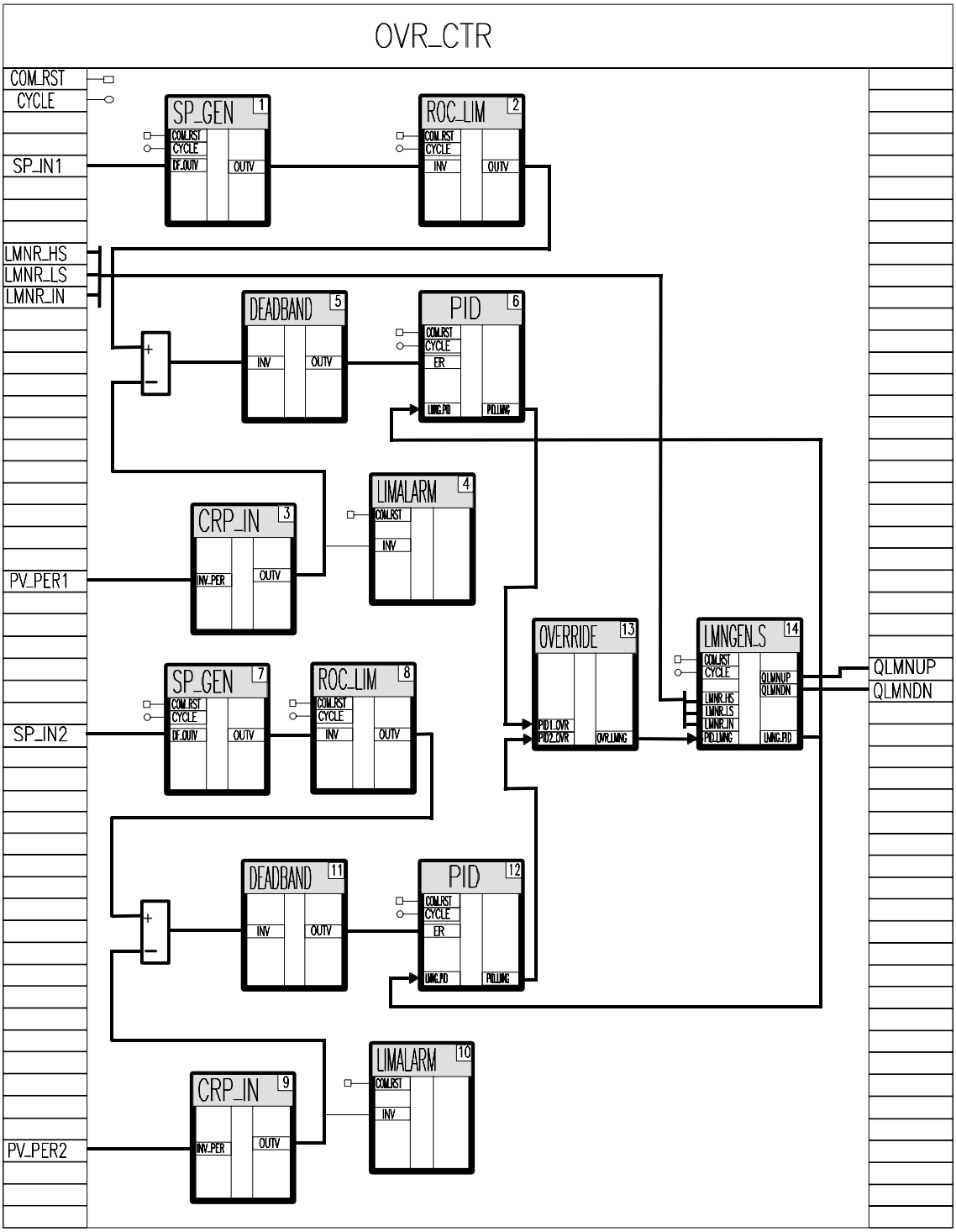


图3-35 OVR_CTR的块互连

功能描述

设定值发生器SP_GEN设置设定值，该设定值的变化速率受限制器ROC_LIM的限制。CRP_IN块将外围设备过程变量转换成浮点值，然后限制值监视器LIMALARM监视该浮点值，以检查它们是否超过了所选择的限制值。偏差信号被发送到PID算法。两个PID块的调节值应用到OVERRIDE块。在此处，确定两个调节值的最大值或最小值，然后传送到调节值处理块LMNGEN_S。在倍率控制器中，LMNGEN_S只能运行在“带有定位反馈信号的步进控制器”模式。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

3.13 实例12: 多变量控制器

总览

实例12的名称为EXAMPLE12，它是一个多变量控制器。此实例中没有模拟的工业过程。

控制闭环

图3-36给出了实例12的应用的完整控制回路。

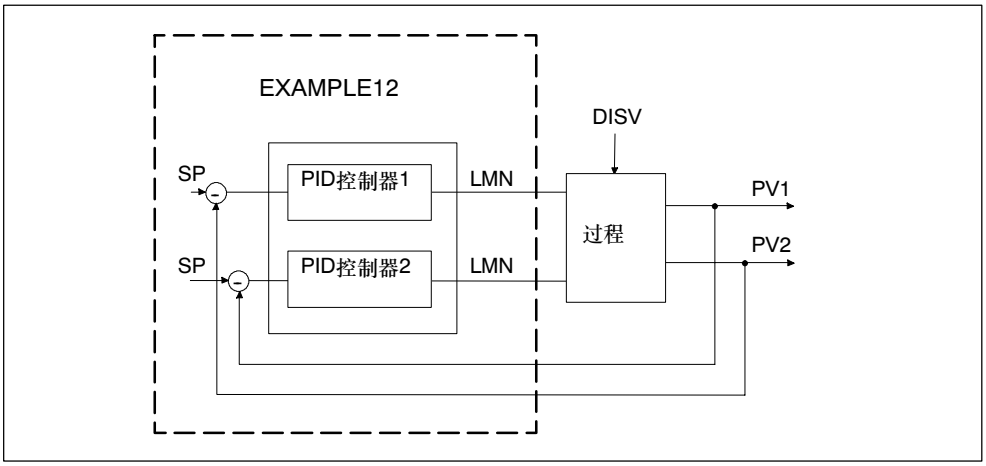


图3-36 实例12的控制回路

块调用

图3-37给出了实例12的块调用。

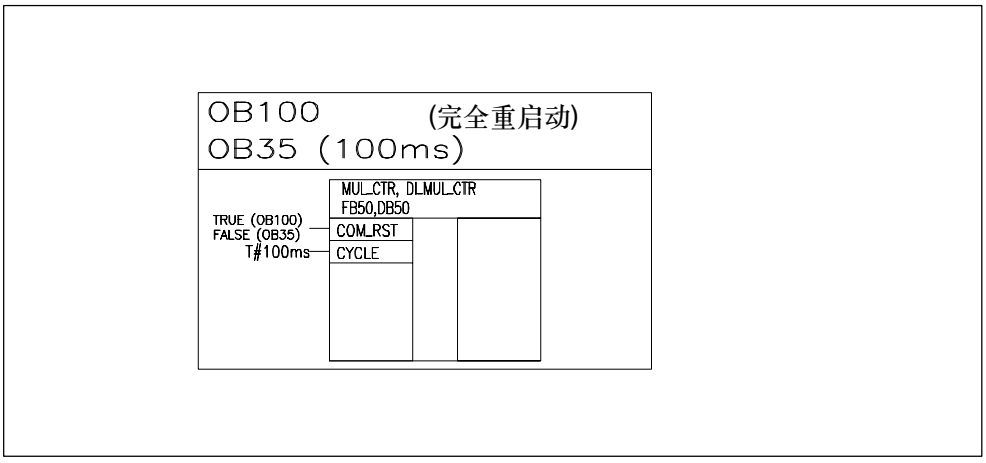


图3-37 实例12的块调用

应用

块MUL_CTR是一个多变量控制器。两个带有连续输出的PID控制器连接到过程中。图3-38给出了MUL_CTR的块互连。

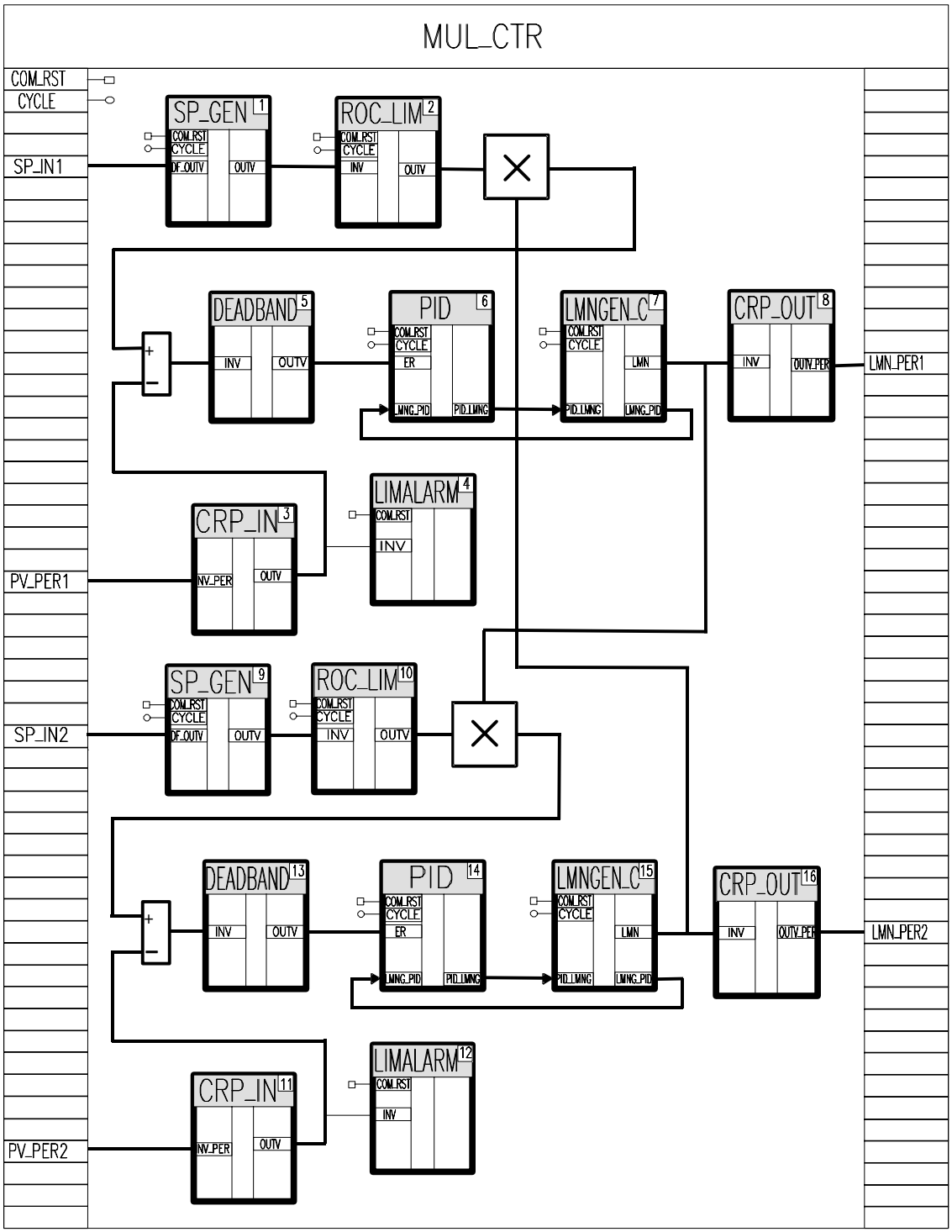


图3-38　　MUL_CTR的块互连

功能描述

每个PID控制器的功能与3.3.1节(第3-10页)中的连续控制器PIDCTR_C类似。

这种情况下，一个控制器的调节值与另外一个控制器的调节值相乘。

完全重启动

在完全重启动期间，分别调用每个块。具有完全重启动例行程序的块运行此例行程序。

技术数据

4

4.1 运行时间

块名称		CPU 313 单位为µs	CPU 314 单位为µs	CPU 315 CPU 315-2DP 单位为µs	CPU 412-1 CPU 413-1 CPU 413-2DP 单位为µs	CPU 414-1 CPU 414-2DP 单位为µs	CPU 416-1 CPU 416-2DP 单位为µs
A_DEAD_B	FB1	170	160	130	31	30	9
CRP_IN	FB2	60	60	60	21	30	6
CRP_OUT	FB3	220	210	180	42	30	12
DEAD_T	FB4	330	320	260	60	39	17
DEADBAND	FB5	210	200	160	32	30	10
DIF	FB6	710	690	550	92	55	26
ERR_MON	FB7	350	340	270	49	34	14
INTEG	FB8	510	500	400	74	44	20
LAG1ST	FB9	670	650	520	94	56	27
LAG2ND	FB10	1140	1110	880	158	86	44
LIMALARM	FB11	610	590	470	65	41	18
LIMITER	2	170	170	140	30	30	9
LMNGEN_C	3	410	390	320	64	41	18
LMNGEN_S	4	1470	1430	1160	184	115	57
NONLIN	5	410	400	320	74	45	20
NORM	6	430	420	330	67	42	19
OVERRIDE	7	180	170	150	35	30	9
PARA_CTL	8	150	140	120	30	30	9
PID	9	1460	1420	1150	184	113	56
PULSEGEN	FB20	200	200	170	50	33	12
RMP_SOAK	FB21	200	200	160	40	30	11
ROC_LIM	FB22	680	660	530	90	55	24
SCALE	3	130	120	100	23	30	8
SPLT_RAN	4	110	100	90	23	30	7
SP_GEN	5	350	340	270	58	35	15
开关	6	90	80	70	25	30	7
LP_SCHED	FC1	340	330	280	79	42	26

4.2 工作存储器要求

块名称		存储器中的FB长度 (单位：字节)	运行时的FB长度 (单位：字节)	存储器中的DB长度 (单位：字节)	运行时的DB长度 (单位：字节)
A_DEAD_B	FB1	898	692	186	44
CRP_IN	FB2	182	70	122	20
CRP_OUT	FB3	206	96	114	14
DEAD_T	FB4	532	394	142	22
DB_DEADT (带有10个历史记录值)	DB3			138	40
DEADBAND	FB5	232	120	114	16
DIF	FB6	410	268	158	30
ERR_MON	FB7	558	360	206	52
INTEG	FB8	488	314	168	36
LAG1ST	FB9	534	368	156	30
LAG2ND	FB10	690	516	190	46
LIMALARM	FB11	390	240	152	28
LIMITER	FB12	262	140	124	20
LMNGEN_C	FB13	1576	1280	276	80
LMNGEN_S	FB14	2578	2152	360	110
NONLIN	FB15	826	672	138	18
DB_NONLI (带有起始点和4个曲线点)	DB4			146	42
NORM	FB16	234	122	130	24
OVERRIDE	FB17	362	214	146	28
PARA_CTL	FB18	406	232	234	82
PID	FB19	1560	1242	340	98
PULSEGEN	FB20	1110	872	190	34
RMP_SOAK	FB21	1706	1500	212	62
DB_RMPSK (带有起始点和4个曲线点)	DB2			146	42
ROC_LIM	FB22	1242	980	222	50
SCALE	FB23	136	32	114	16
SPLT_RAN	FB24	304	180	138	28
SP_GEN	FB25	658	484	164	40
开关	FB26	238	116	118	18
LP_SCHED	FC1	1104	972	280	79
DB_LOOP (带有5控制环路)	DB1			190	64

4.3 经验规则

运行时间

可以通过下列公式来计算总运行时间:

所调用块的运行时间(来自模块化PID控制)
+ 块调用数***常量**
= 总运行时间

可以通过下列方式找到块调用数:

调用的块(来自模块化PID控制)
+ 调用的用户FB
= 块调用数

下列**常量**应用于CPU:

CPU	常量
CPU 313	105 μs
CPU 314	100 μs
CPU 315、CPU 315-2 DP	80 μs
CPU 412-1、CPU 413-1	23 μs
CPU 414-1、CPU 414-2 DP	12 μs
CPU 416-1、CPU 416-2 DP	9 μs

存储器要求

存储器要求的经验规则与工作存储器相关。

使用的FB需要的存储器(来自模块化PID控制)
+ 调用的FB的背景数据块所需要的存储器(来自模块化PID控制)
+ 块调用数*120字节

= 所需的总存储器

可以通过下列方式找到块调用数:

被调用块(来自模块化PID控制)
+ 调用的用户FB

= 块调用数

模块化PID控制的组态工具

5

要求

必须在编程设备/PC上正确安装了STEP 7。

软盘

光盘上提供有所需要的软件。

安装

要安装软件，请如下操作:

1. 在编程设备/PC的光盘驱动器中插入光盘。
2. 双击“控制面板”中的“添加/删除程序”图标，在WINDOWS下启动软件安装对话框。
3. 选择光盘驱动器，然后选择文件Setup.exe，启动安装过程。
   然后就会在用户的系统上安装组态工具。
4. 按照安装工具显示的提示，一步步进行操作。

自述文件

关于所提供的软件的重要最新信息，存储在自述文件中。可以在Windows开始菜单中，通过“开始 > SIMATIC > 产品信息 > 中文”，找到该文件。

用途

组态工具设计用于使用户能够更容易地安装、启动和测试控制器，这样用户可以将更多精力集中在手头上的实际控制问题。

组态工具的功能

每个功能都有其自己的窗口。还可以多次调用同一个功能；换句话说，例如，用户可以同时显示多个控制器的环路监视器。

监视控制器

通过使用曲线记录器功能，用户可以记录控制环路中所选变量的数值随时间的变化趋势，以及显示这些曲线。最多可以同时显示多达四个变量。

通过环路监视器功能，可以显示所选控制器的相关控制变量(设定值、可调节变量和过程变量)。

过程辨识

通过过程辨识功能，可以找到特定控制环路的最优控制器设置。过程的特征通过实验方式辨识。根据这些特征，计算出理想的控制器参数，并保存供将来使用。

手动控制

环路监视器功能使用户可以为控制环路的相关变量修改或设置新值。

集成帮助

组态工具有一个集成帮助系统，在用户使用该工具时为用户提供帮助。可以通过下列方式调用该帮助系统:

- 通过菜单命令**帮助 ▶ 目录**
- 通过按下F1键
- 通过单击各个对话框中的“帮助”按钮。

参考

A

更多读物

下列书籍介绍了控制工程设计的基础内容:

/350/ 用户手册: *SIMATIC 7,*
标准控制

/352/ J. Gißler, M. Schmid: Vom Prozeß zur Regelung. Analyse, Entwurf, Realisierung in der Praxis. Siemens AG. ISBN 3-8009-1551-0.

索引

字母

B

C

D

F

G

H

J

K

L

M

P

Q

R

S

T

W

X

Y