

1 PIDConL 块基本特性介绍

在 PCS7 V7.1 版本中，PCS 7 AP Library V7.1 (APL) 提供了三种 PID 控制功能块：PIDConL、PIDConR 和 PIDStepL。其中 PIDConL 和 PIDConR 都是连续 PID 控制器，即输出是连续的控制值，相对应地，PIDStepL 输出的是脉冲信号，可以控制类似于电动阀之类的执行机构。

通过适当的参数设置和扩展组合，使用 PIDConL 功能块可以方便地实现如下控制：

- ✓ 固定设定值控制；
- ✓ 层级控制；
- ✓ 比值控制；
- ✓ 分程控制；
- ✓ 闭环史密斯预估控制；
- ✓ 超驰控制；

PIDConL 继承了 V7.0 下的 PID 功能块 CTRL_PID 的诸多特点，例如设定值的选择流程、手自动的切换方式等，但相比较而言，新 PID 功能块增加了如下功能：

- ✓ 在手自动之外增加了程序控制模式；
- ✓ 增加了设定的斜坡函数功能；
- ✓ 增加了偏差值的相关消息；
- ✓ 在比例和积分计算中增加了函数处理；
- ✓ 添加了控制带的功能；

这些新功能使得 PIDConL 在优化控制算法的同时，能够满足更多的控制需求。

2 PIDConL 块的基本使用

2. 1 PIDConL 块调用及基本管脚介绍

PIDConL 的功能块编号是 FB1874，在 CFC 编程窗口左侧的 Libraries 目录结构中“ PCS 7 AP Library V71”Blocks+Templates\Blocks”可以找到该功能块，如下图所示：

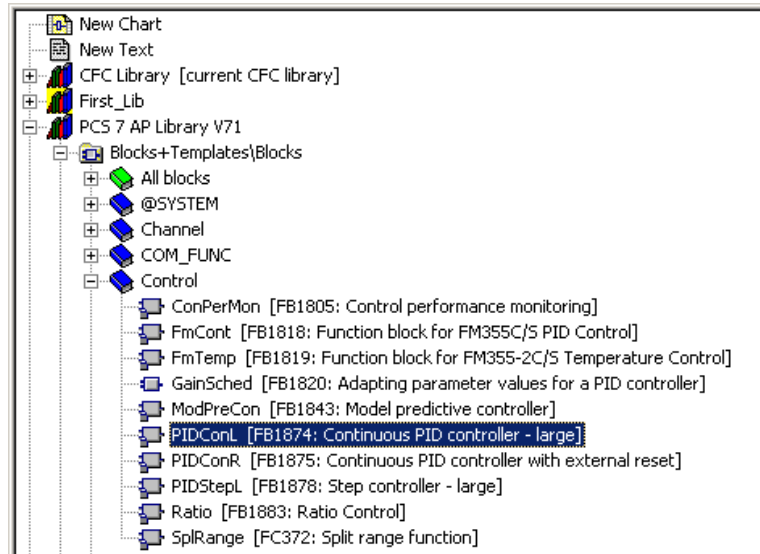


图 1 从库中选择插入功能块

PIDConL 功能块的基本管脚如下图所示，关于功能块或者全部管脚的相关信息可以查看在线帮助：

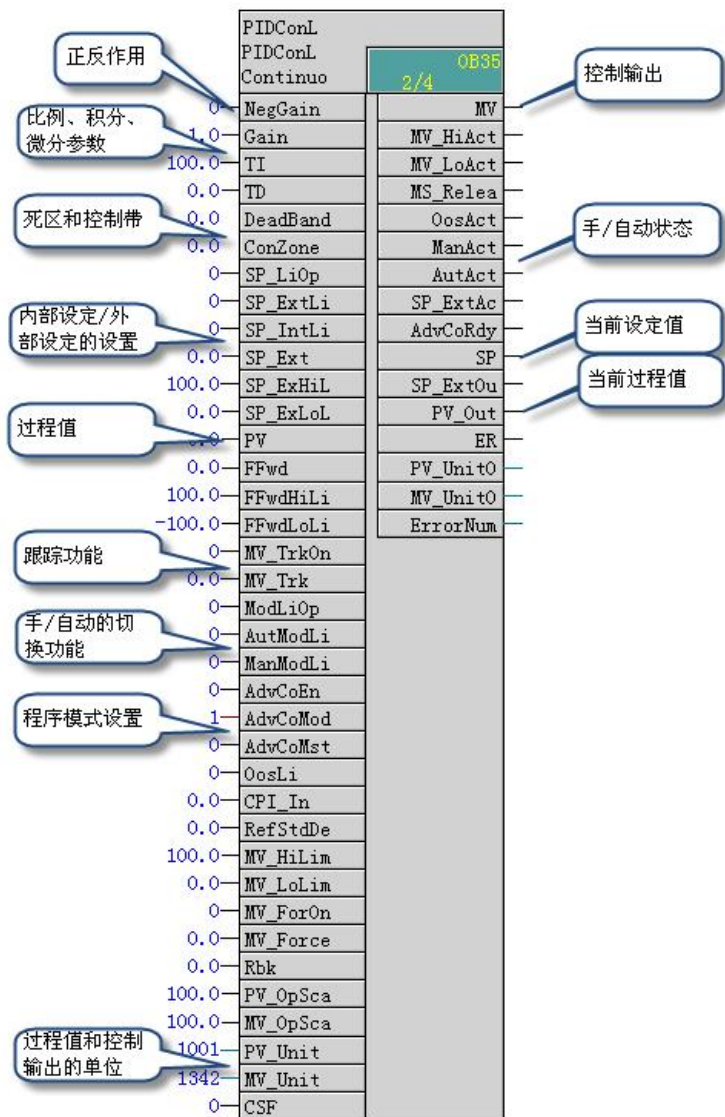


图 2 PIDConL 块管脚

在使用中，连接过程值的驱动块 PCS7AnIn 的 PV_Out 块到 PIDConL 的 PV 端，功能块的输出 MV 可以输出到驱动块 PCS7AnOu 的 PV_IN，即可实现基本的单回路 PID 控制。具体连接如下图所示：

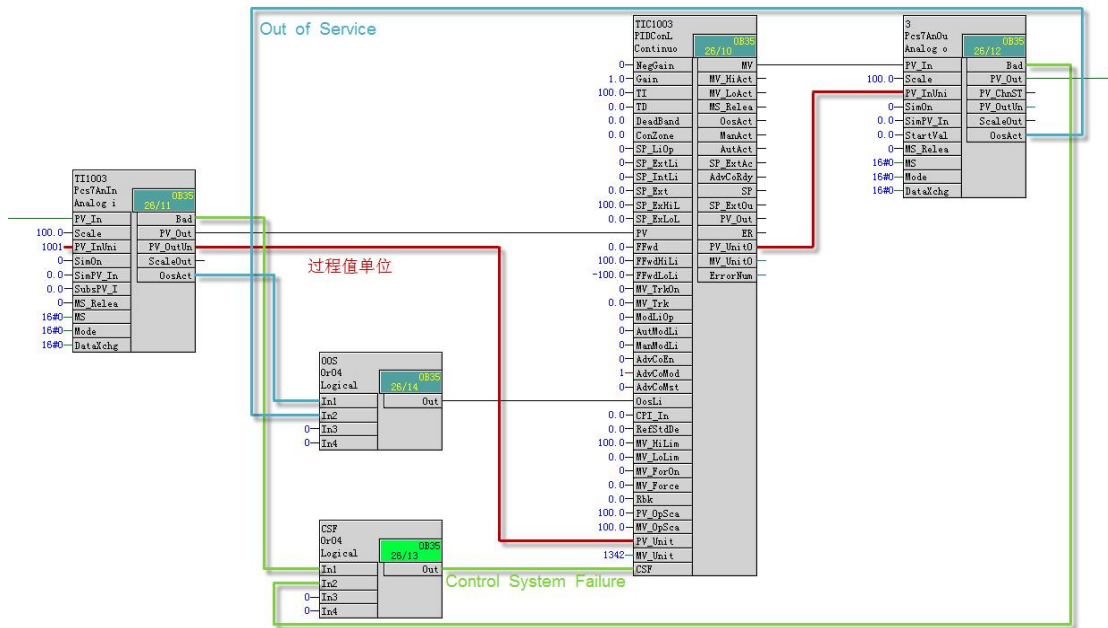


图 3 基本 PID 回路

完成 CFC 中的程序编写之后，进行 CFC 编译并下载程序。此外还需要对 OS 项目进行编译，以便在相应层级的过程画面中自动创建一个 PIDConL 的图标。

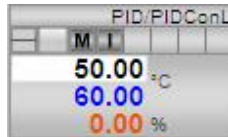


图 4 PIDConL 画面对象

2. 2 操作面板的基本使用

2. 2. 1 标准面板

在运行系统中点击过程画面中 PIDConL 的 ICON 图标即可打开功能块的标准视图：

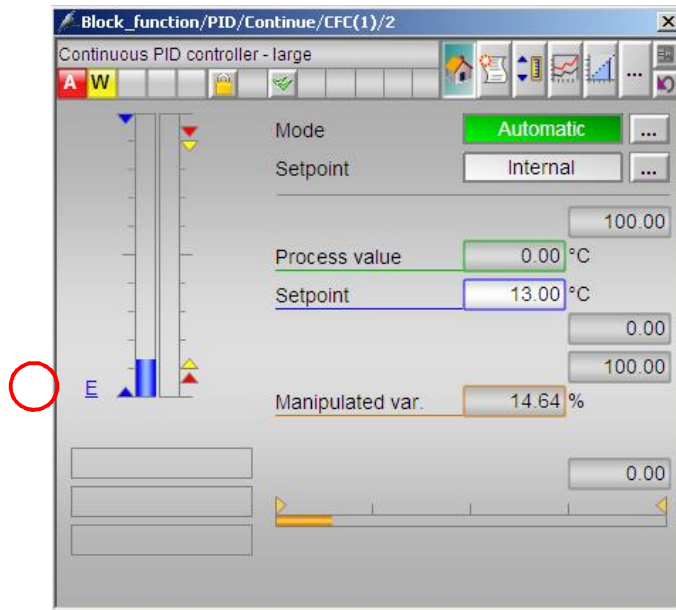


图 5 PIDConL 标准面板

PIDConL 的标准面板包括操作模式和设定值的选择、过程值的显示、设定值的设置、控制输出等等。在标准面板的左侧以柱状图的形式分别显示了当前过程值和设定值，需要注意的是在柱状图的左侧有一个字母 **E**，这表示的是目前有效的外部设定值的大小，如图中所示，外部设定值是 **0**。另外在开启斜坡功能的情况下，会有一个字母 **R** 来表示斜坡目标设定值的大小。

2. 2. 2 参数面板

标准面板中包含了回路最基本的信息，具体的控制参数设置都集中在参数面板中：

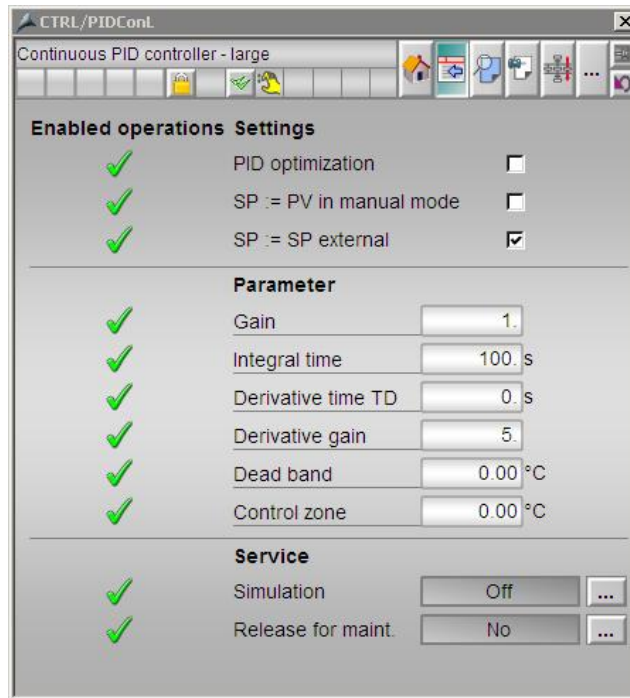


图 6 参数面板

参数面板主要包括激活 PID 优化、设置手动模式下 SP 的跟踪、控制参数的设置和仿真、维修释放设置等。需要注意的是，默认情况下“ SP:=SP external” 是激活的，即在使用外部设定值时，内部设定值会跟踪外部设定值。正常使用过程中，可以根据需要激活“ SP:=PV in manual mode” 选项，让手动状态下始终保持设定值和过程值相等。

此外，在标准视图的右上角有视图切换的按钮，依次包括了消息窗口、限制值、趋势图、斜坡处理、参数设置和 BATCH，点击相应的按钮即可进入具体的窗口，查看和设置功能块的相关信息。

- ✓ 消息窗口中记录了 PIDConL 的所有归档消息，可以对新的报警消息进行确认的操作；
- ✓ 限制值窗口可以对过程值 PV、偏差 ER、回读值、设定值和控制输出值的高低限范围；
- ✓ 趋势窗口中描出了过程值、设定值和控制输出的变化情况，在调试过程中可以提供参考；
- ✓ 斜坡处理窗口中可以设置设定值通道上的数据变化“柔化”处理的相关参数；
- ✓ 预览窗口是当前模块基本参数的一个概要显示，包括内/外部设定值、偏差、外部干扰、工作模式等。

2.3 如何切换操作模式

和传统的 PID 功能块一样，PIDConL 的手/自动切换可以在面板上完成，也可以在 CFC 中通过程序来完成，默认情况下允许用户在面板上切换：

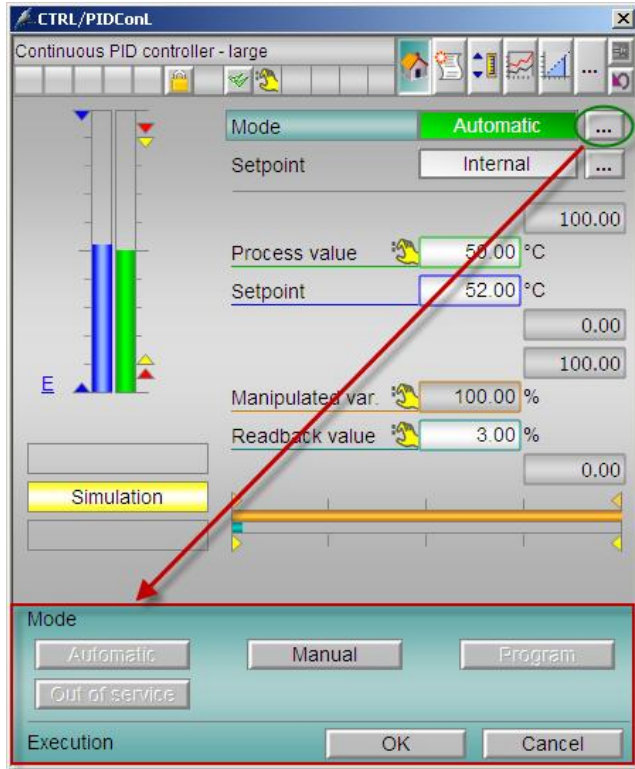


图 7 操作面板上的手/自动切换

点击标准面板下的“ Mode”项的右侧按钮可以弹出模式切换命令条，在其中选择需要的模式，并以“ OK”键确认即可。

如果需要通过用户程序来实现手/自动的切换，则会使用到如下表所示的功能块管脚：

表 1 手/自动切换相关的管脚

管脚	功能	信号类型
ModLiOp	模式选择的方式	STRUCT
AutModLi	通过程序来选择自动模式	STRUCT
ManModLi	通过程序来选择手动模式	STRUCT

通过参数 ModLiOp 来决定是由程序（CFC 或者 SFC）还是由面板（操作面板）来实现切换。例如需要通过程序将回路切换到手动模式，则需要配置参数：

ModLiOp=1
 AutModLi=0
 ManModLi=1

2. 4 如何使用外部设定值

从功能块的管脚可以看出，内部设定值和外部设定值是两个独立的参数，具体 PID 计算过程中选择哪个参数的数据可以在程序中配置，也可以在操作面板上来选择。

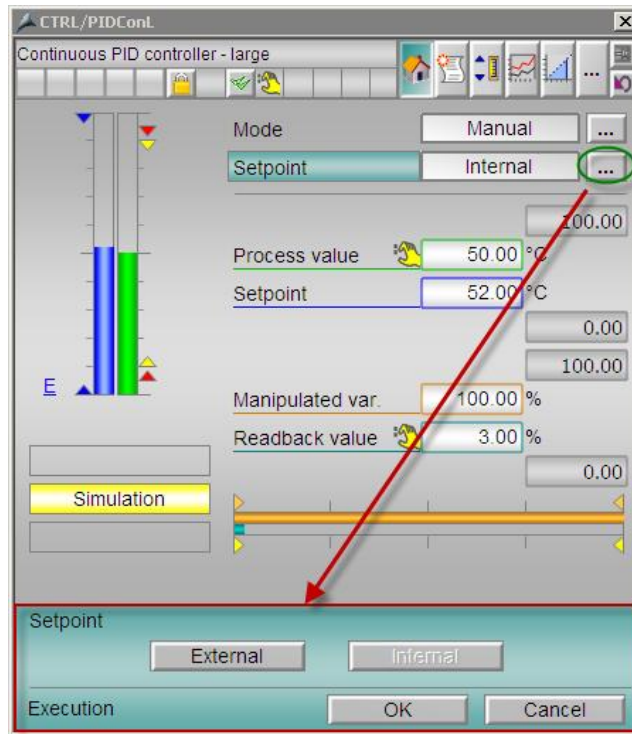


图 8 内部/外部设定值的选择

如上图所示，点击标准面板下的“Setpoint”右侧的按钮，在弹出的设定值命令条上选择内部或者外部，并以“OK”按钮确认即可。

用户程序中需要切换内部/外部设定值，会使用到下表中的功能块管脚：

表 2 内部/外部设定值选择相关的管脚

管脚	功能	信号类型
SP_LiOp	设定值源选择的方式	STRUCT
SP_ExtLi	通过程序来选择外部设定值	STRUCT
SP_IntLi	通过程序来选择内部设定值	STRUCT
SP_Int	内部设定值	STRUCT
SP_Ext	外部设定值	STRUCT

通过参数 SP_LiOp 来决定是由程序（CFC 或者 SFC）还是由面板（操作面板）来实现切换，例如需要由程序来选择外部设定值，则相关参数配置如下：

SP_LiOp=1

SP_ExtLi=1

SP_IntLi=0

如果 SP_ExtLi 和 SP_IntLi 都被设置为 1，那么内部设定值会有更高的优先级，即回路会使用内部设定值，此时回路不会产生错误消息。

设定值的多通道输入，是为了方便功能块的扩展，但也带来了内/外部设定值切换过程中由于设定值变化而可能导致回路波动的风险。所以，功能块提供了参数“ SP_TrkExt” 以确保在使用外部设定值的情况下，内部设定值保持和外部设定值的跟踪，这样在重新使用内部设定值的时候可以做到无扰切换。这个参数可以在操作面板上的参数画面中选择“ SP:=SP external” 来使能。（图 6 所示）

在自动状态中，手动控制值 Man 会跟踪控制输出 MV，以此可以确保自动到手动无扰切换。

2. 5 如何使用 PIDConL 的程序模式

程序模式指的是在使用外部软件来实现先进控制（APC）的过程中，PIDConL 提供相应的参数来实现设定值或者控制输出的远程设定，外部先进控制软件通常通过 OPC 方式进行通讯。

默认情况下，程序模式是关闭的，需要设置管脚 AdvConEn 为 1 才能激活该功能。

注：如果当前选择的是外部设定值，则无法切换到程序模式。

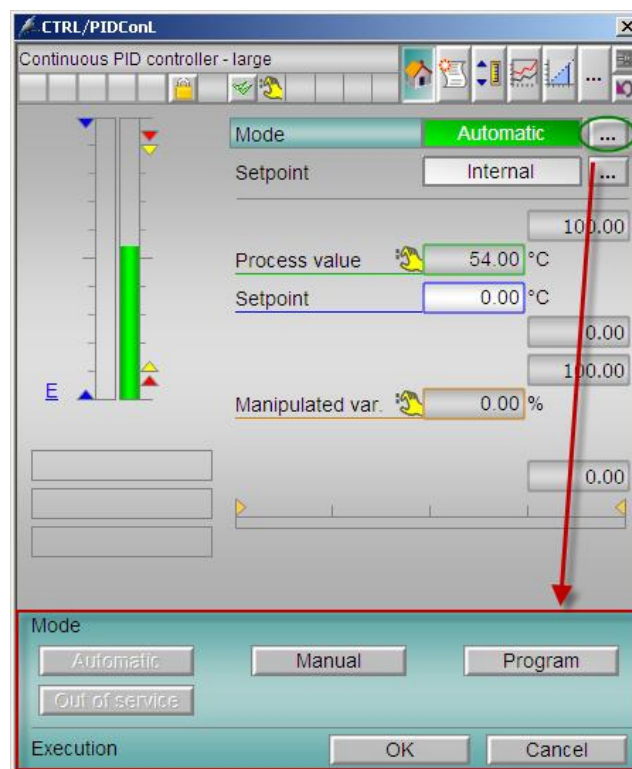


图 9 程序模式的选择

如上图所示，点击标准面板下的“ Mode” 右侧按钮，在弹出 Mode 命令条中选择“ Program” ，并点击“ OK” 按钮即可切换到程序模式。

在程序模式中，其又分为设定值程序控制模式和输出值程序控制模式。在设定值程序控制模式下，设定值窗口变为不可操作的状态，此时的设定值直接由参数 AdvCoMV 来设定：

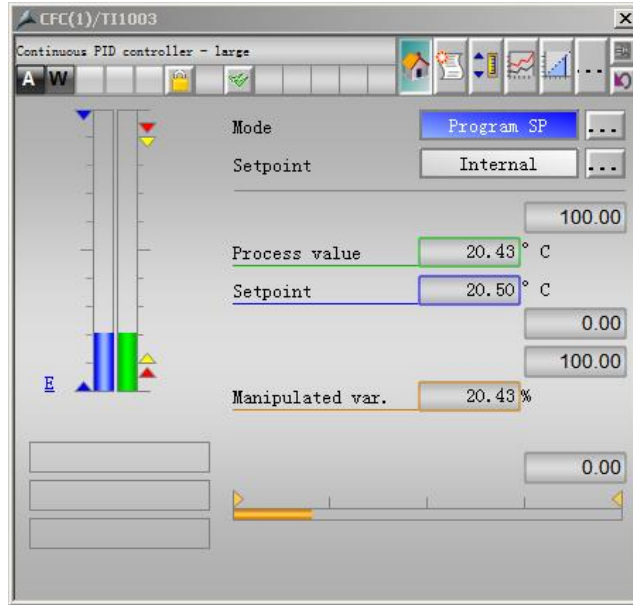


图 10 设置设定值的程序模式

如上图所示，当前操作模式是“Program SP”，通过 OPC 传递过来的设定值是 20.5，此时回路根据这个设定值进行自动控制。

在输出值程序控制模式下，控制器的 PID 运算停止，直接采用 AdvCoMV 作为控制器的输出。

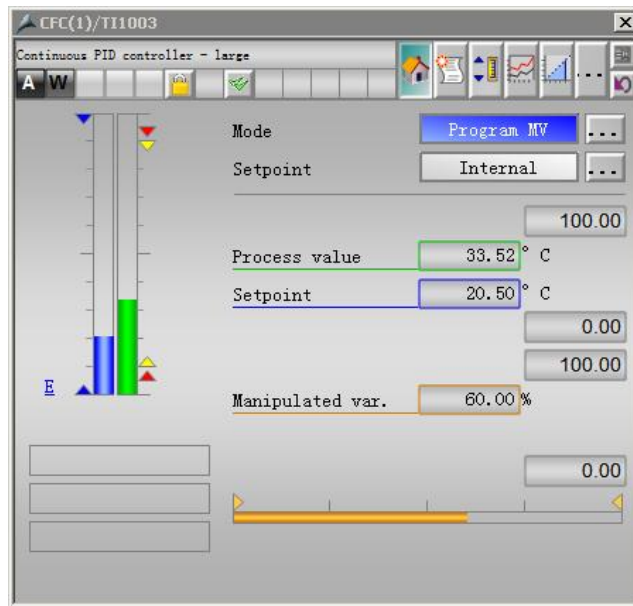


图 11 设置控制输出的程序模式

上图中，通过 OPC 过来的控制输出为 60.0，此时的 PIDConL 不进行 PID 运算，而直接采用相应的输出值。

具体采用设定值程序控制模式还是输出值程序控制模式是在 CFC 中通过参数 AdvCoModSP 来选择的。

程序模式需要使用到 CFC 中功能块的如下管脚：

表 3 程序模式相关的管脚

管脚	功能	信号类型
AdvCoEn	程序模式使能	STRUCT
AdvCoOn	通过面板来激活程序模式	BOOL
AdvCoModSP	程序模式的类型：设定值或直接控制输出	BOOL
AdvCoMstOn	是否使用边沿信号来激活或取消激活	STRUCT
AdvCoMV	外部程序的设置值	REAL

例如，需要使用一个外部应用程序通过 OPC 来设置 PIDConL 的设定值，参数可以这样设置：

AdvConEn=1

AdvCoModSP=1

之后，在回路处于自动状态下，就可以切换到程序模式了。

退出程序模式后的 PIDConL 的工作状态和程序模式的类型相关：如果采用远程设置设定值，那么退出之后工作在自动模式下；如果采用的是直接输出，则退出后工作在手动模式下。

2. 6 如何使用 PIDConL 的仿真功能

仿真功能（“ Simulation”）是为了方便调试而设置的，激活该功能，可以在操作面板上设置过程值和回读值（Rbk），开启仿真功能之后，连接的实际 PV 和 Rbk 被屏蔽。

在参数面板上点击“ Simulation”右侧按钮，然后在弹出的命令行中选择“ On”，并通过“ OK”按钮确认即可。

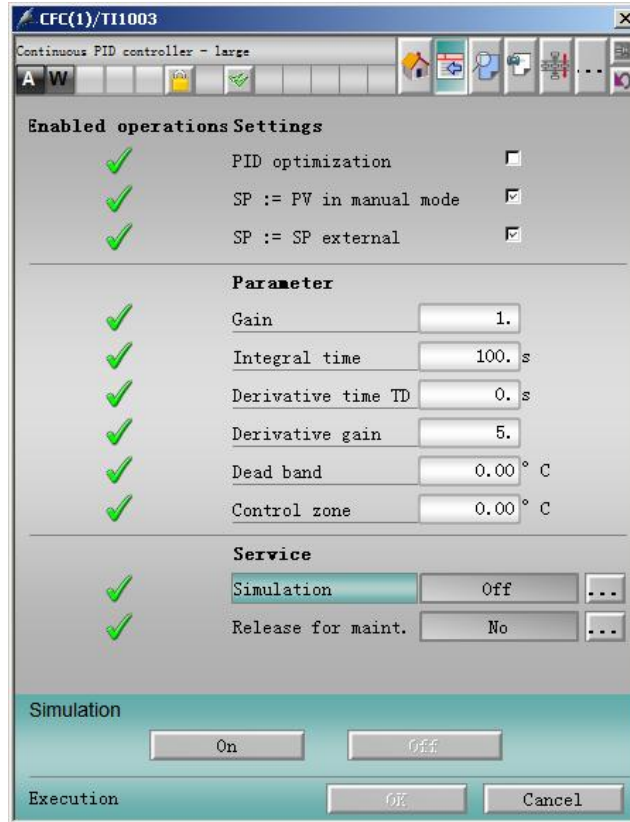


图 12 开启仿真功能

进入仿真模式之后，回路的过程值 PV 和回读值可以由操作员手动设置。

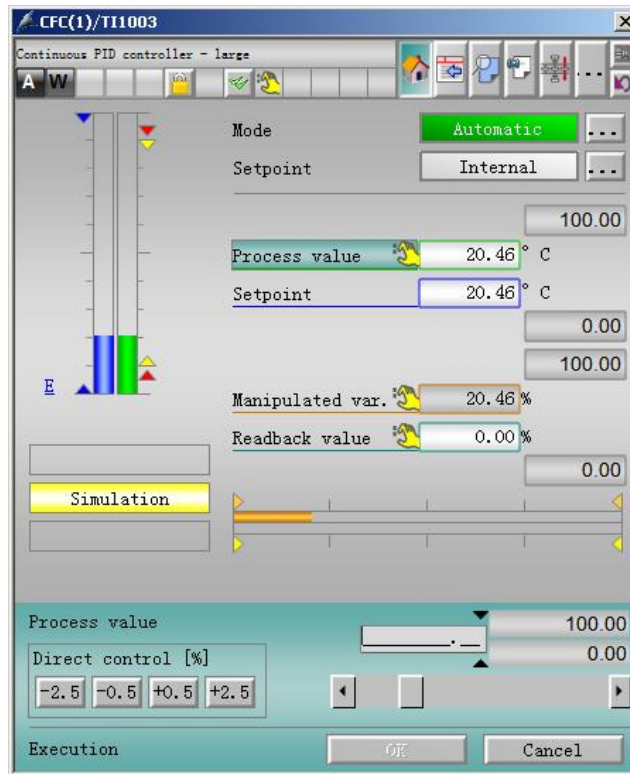


图 13 仿真状态下设置过程值

注：打开仿真功能需要当前登录用户具有 1100 号授权（Highest process controlling）

3 PIDConL 的应用

3. 1 分程控制

普通的控制回路中一个 PID 的输出控制一个执行机构——例如一个阀门来完成控制任务。某些工艺情况下，单个回路中存在两个阀门，它们之间在动作上存在一定的先后顺序，此时则需要分程控制方案。

例如，使用两个出口阀来调节一个反应罐中物料的液位，当物料液位过高时，增加 A 阀开度，如果 A 阀全开之后液位还是过高，则开启并不断增加 B 阀的开度；反之，如果物料液位过低，则先关小 B 阀，在 B 阀全关之后再关小 A 阀。两个阀的动作关系如下图所示：

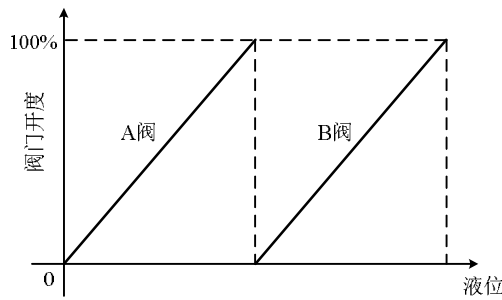


图 14 阀门动作关系

上图中，两个阀门的动作是相连续、服务于同一个调节回路的，为了实现分程控制，除了需要调用 PIDConL 块来进行 PID 运算之外，系统还提供了分程功能块 SplRange (FC372)，在 APL 功能库下的“Control”中可以找到，该功能块负责将控制输出“分解”成两个阀的控制信号，在实际的使用过程中，两个阀门的动作特性各不相同，需要根据实际情况配置相应的参数。

根据上例中两个阀门的动作关系，分别设置分程块 SplRange 的如下参数：

Out1Scale.High=0.0

Out1Scale.Low=100.0

Out2Scale.High=100.0

Out2Scale.Low=0.0

连接 SplRange 的输入管脚 In 到 PIDConL 的控制输出 MV，两个输出管脚 Out1 和 Out2 分别将其通过 CH_AO 驱动块输出到对应输出通道。

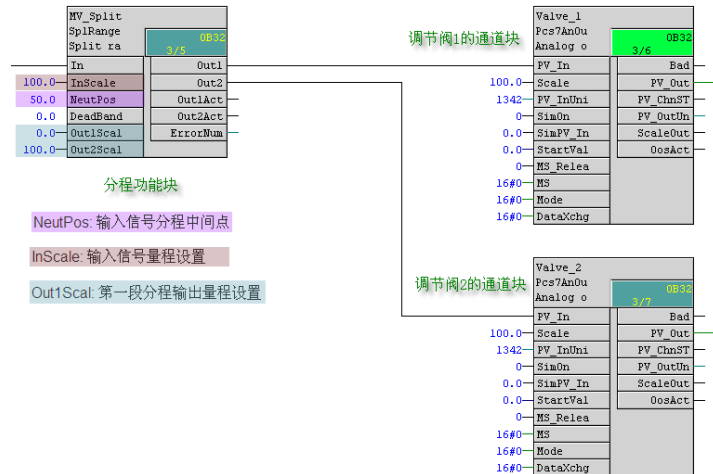


图 15 分程输出 CFC 块图

由于在分程控制中涉及到了两个 AO 通道，所以和简单回路不同，需要将两个驱动块的 QBad、Out of Service 等状态进行“逻辑或”运算之后送到 PIDConL 块中。

在 PCS 7 中还提供了分程控制的过程标签类型（Process tag type），打开“ PCS 7 AP Library V71”的“ Templates”，其中可以找到“ SplitrangeControl”，直接拷贝这个过程标签类型到层级或者项目的主数据中，之后可以通过过程对象视图（Process Object View）、IEA 文件导入或者直接修改参数的方式来批量添加分程控制程序。PCS 7 软件提供的过程标签类型除了完成分程控制之外，还增加了诸如带输出迟滞的回路仿真、控制器性能监视等功能。

3. 2 串级控制

串级控制是应用最为广泛的一种复杂控制系统，主要运用于控制精度要求较高、对象具有大迟滞特性的回路，通过将—个控制对象分解成两个“串联”的控制回路，从而达到降低回路的迟滞时间、限制过程值波动范围的目的。

—个典型的串级控制包括主调节回路和副调节回路，主调节回路的控制输出作为副调节回路的设定值，副调节回路的对象直接影响主回路的控制对象。两个调节回路有独立的测量值，而且副回路的响应速度要快于主回路，以便干扰能够地被抑制下来。如下图所示的控制系统方框图描述了串级系统的特性：

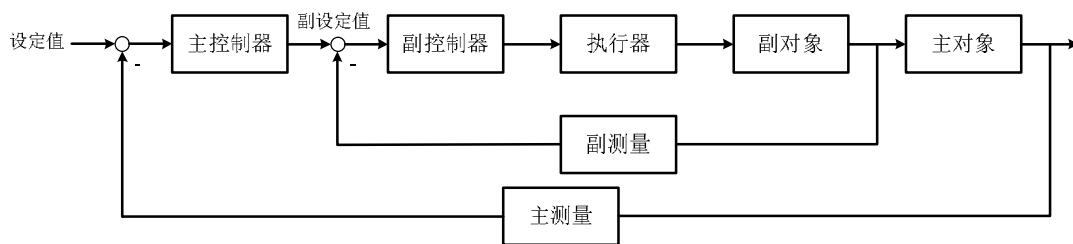


图 16 串级控制系统方框图

在 CFC 中使用 PIDConL 来搭建串级控制程序，除了上述描述的两个控制器之间的关联之外，还需要考虑以下几个方面：

- 串级模式和其他控制模式的无扰切换。作为串级模式的一大主要特点，要求其中的控制回路都处于自动模式，而且主回路的输出要用作副回路的实际设定值。在其他控制模式（例如手动等）下，主回路的输出要跟踪副回路的过程值，以防止副回路出现设定值跳变。为此，可以使用 PIDConL 的“ MV_TrkOn”和“ MV_Trk”两个参数分别来开启输出跟踪功能和设置跟踪值；

- 在调试串级回路时，需要先调节副回路，之后才是主回路，这就需要副回路能够在串级和常规单回路中进行切换。针对这个功能，可以使用 PIDConL 的外部设定值来连接主回路的功能块，此外，PIDConL 的输出管脚上的“ CascaCut”就是根据外部设定值投入使用与否来判断串级是否启用；

- 副回路的设定值取决于主回路的输出，这就需要主回路能够准确地将偏差的变化趋势传递到副回路中，但积分功能的—存在可能会延迟这个传递过程，从而导致副回路振荡幅度加大，难以回到稳定。尤其是在副回路的输出已经处于最高/最低的情况下，主回路的积分功能可能会让副回路的设定值继续

不断地增加/减少，而在主回路的偏差发生改变时，副回路不能快速地跟上响应。为了解决这个问题，可以使用 PIDConL 的正向/反向积分抑制功能，在副回路的控制块的输出达到上限之后，可以设置 “ IntHoldPos” 来抑制正向积分，反之亦然。

和分程控制一样，PCS 7 库中同样提供了一个串级控制的过程标签类型，在“ PCS 7 AP Library V71” 库的“ Templates” 中，名为“ CascadeControl”，可以直接使用这个过程标签类型来完成串级控制。如果使用两个 PID 功能块来编程，可以参考这个标签类型，完成如下管脚连接：

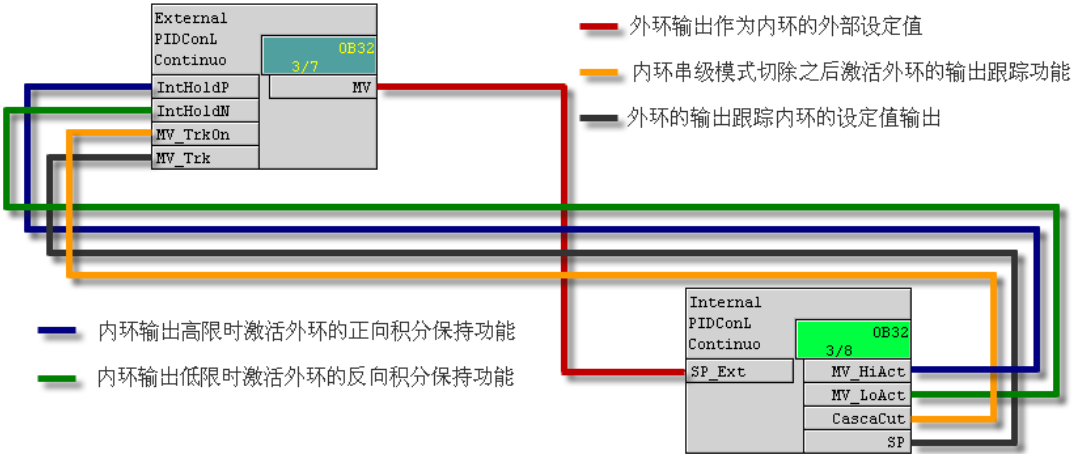


图 17 串级控制中两个内外环 PID 连接关系

APL 库中过程标签类型中除了串级控制相关的功能之外，为了方便调试还有带延迟的回路仿真、控制器性能监视等功能。