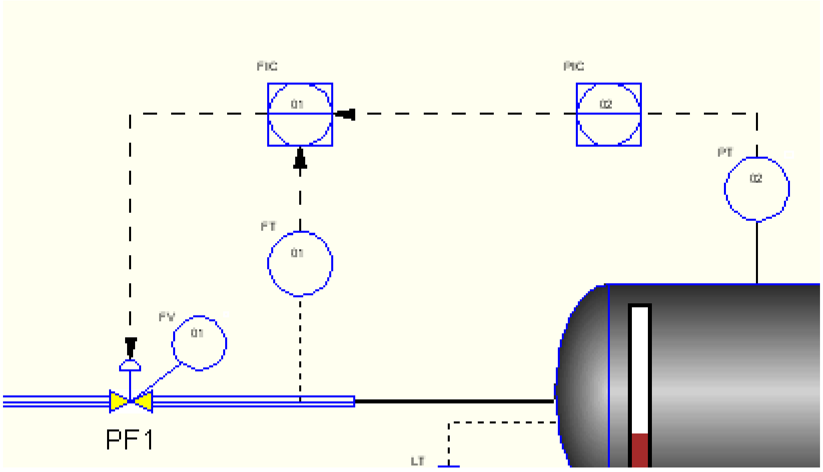
KONGSBERG K-Spice® DcsPidController

Module Reference



/B

Document history

Revision	Date	Description	Written by	Checked by	Approved by
Rev. A	May 2010	Created document and first XML version.	JS		
Rev. B	May 2013	Updated with description of manual output..	MB		

The reader

This manual is intended as a reference for the K-Spice® user. This manual is based on the assumption that the user is familiar with process modelling, as well as oil and gas production systems.

Note

Kongsberg Oil & Gas Technologies endeavours to ensure that all information in this document is correct and fairly stated, but does not accept liability for any errors or omissions.

Comments

To assist us in making improvements to the product and to this manual, we welcome comments and constructive criticism.

E-mail: kogt.documentation@kongsberg.com

Kongsberg Oil & Gas Technologies AS
www.kongsberg.com/kogt

Table of contents

1 Introduction

In the process industry the use of the PID control element (proportional-integral-derivative controller) is the most central method of controlling the operation of process plants. Advanced methods like model predictive control has also been developed, but the classic control methods are still the mostly used methods. The use of PID controller in a standard feedback control loop is easy to understand and well proven technology. A PID controller can also be combined with other control elements for more advanced control structures; for example feed forward control, cascade control, ratio control, selective control and split range control.

2 Description

This module simulates a PID controller which can be connected to a DCS system. The baisc functionality is similar to the generic PID controller simulated by the PidController module. The differences are described in the section 2.3 DCS Connection Functionality.

The controller output is found from a standard equation in position form. A PI-controller may be obtained by setting the derivative time to zero, a PD controller by setting the integral time to zero, and a P controller by setting both the integral and the derivative times to zero.

The controller may operate with an internal setpoint, or with an external setpoint, i.e. cascade control.

The controller supports manual, automatic and direct digital control modes and maintains bumpless transfer when switching between the modes. Other options are anti reset-windup control and feedback tracking.

2.1 Normal Connectivity

The controller block normally gets its *Measurement* input variable from a FieldTransmitter module. You can either connect from the control signal or the measured value. If you connect from the measured value the controller will work in engineering units. But if you connect from the control signal the controller will work in control signal unit. The controller output is normally sent to a valve, but can also be sent to another controller, an arithmetic operator or a motor.

2.2 DCS Connection Functionality

Compared to the generic PidController module, this block has some extra input variables:

- *InputSwitch*: Input switch between local, remote and manual control. This input variable is normally connected to the output OutputSwitch of a valve or an asynchronous machine.
- *DcsFeedback*: Dcs feedback signal. This input variable is normally connected to the output TargetPosition of a valve orTargetSignal of an asynchronous machine.

- *DcsOutput*: Dcs controller output. This input variable is normally connected to the output from a controller in the DCS system (PID.OP.DCS).
- *DcsSetpoint*: Dcs setpoint. This input variable is normally connected to the set point of a controller in the DCS system (PID.SP.DCS).
- *DcsMode*: Dcs Auto/Manual/Computer mode. This input variable is normally connected to the mode of a controller in the DCS system (PID.Mode.DCS).

The value of the input variable InputSwitch governs the use of the other extra input variables.

If *InputSwitch* = Local, then none of the other extra inputs are used, and the DcsPidController behaves similar to the DcsController.

If *InputSwitch* = Remote, then:

- The controller is tracking the *DcsFeedback*. This *DcsFeedback* tracking takes precedence over the *Feedback* tracking, governed by the *Tracking* input,
- The *Mode* of the controller is set equal to the *DcsMode*. If *DcsMode* = Manual, then the *ControllerOutput* is set equal to the *DcsOutput*.
- The controller uses the *DcsSetpoint* as an external set point.
- The trip actions described in the PCDA Mode section are not supported.

If *InputSwitch* = Manual, then the controller is tracking the *DcsFeedback*. This *DcsFeedback* tracking takes precedence over the *Feedback* tracking, governed by the *Tracking* input.

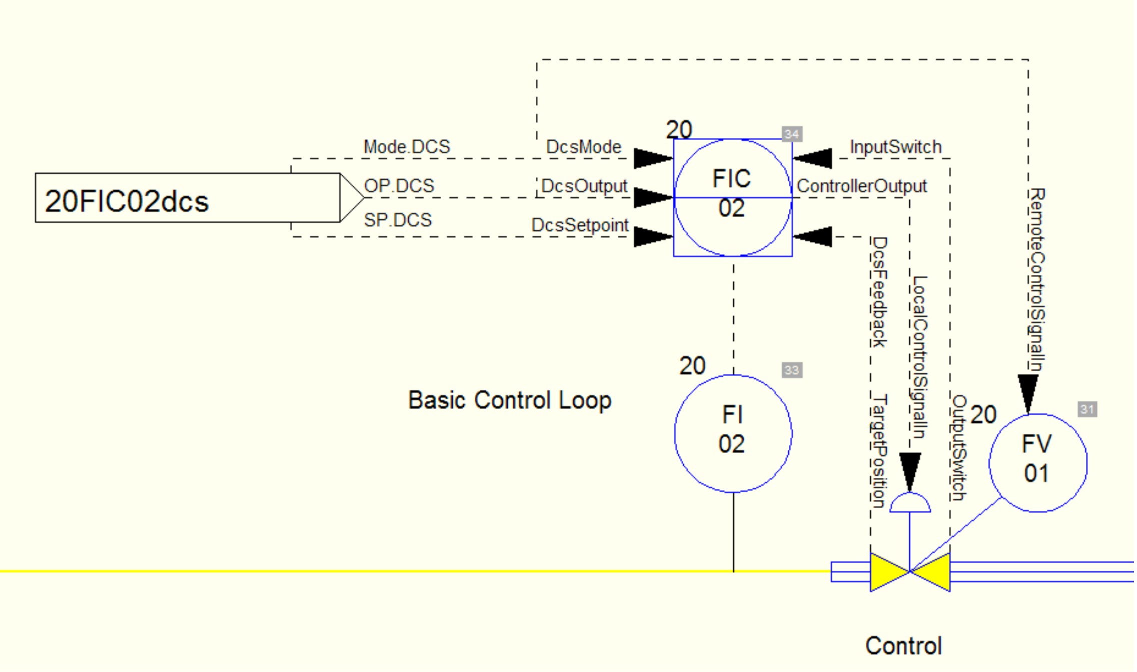
2.3 Example Configurations

Below, example configurations are shown for:

- Basic Control Loop
- Cascade Control
- Split Range

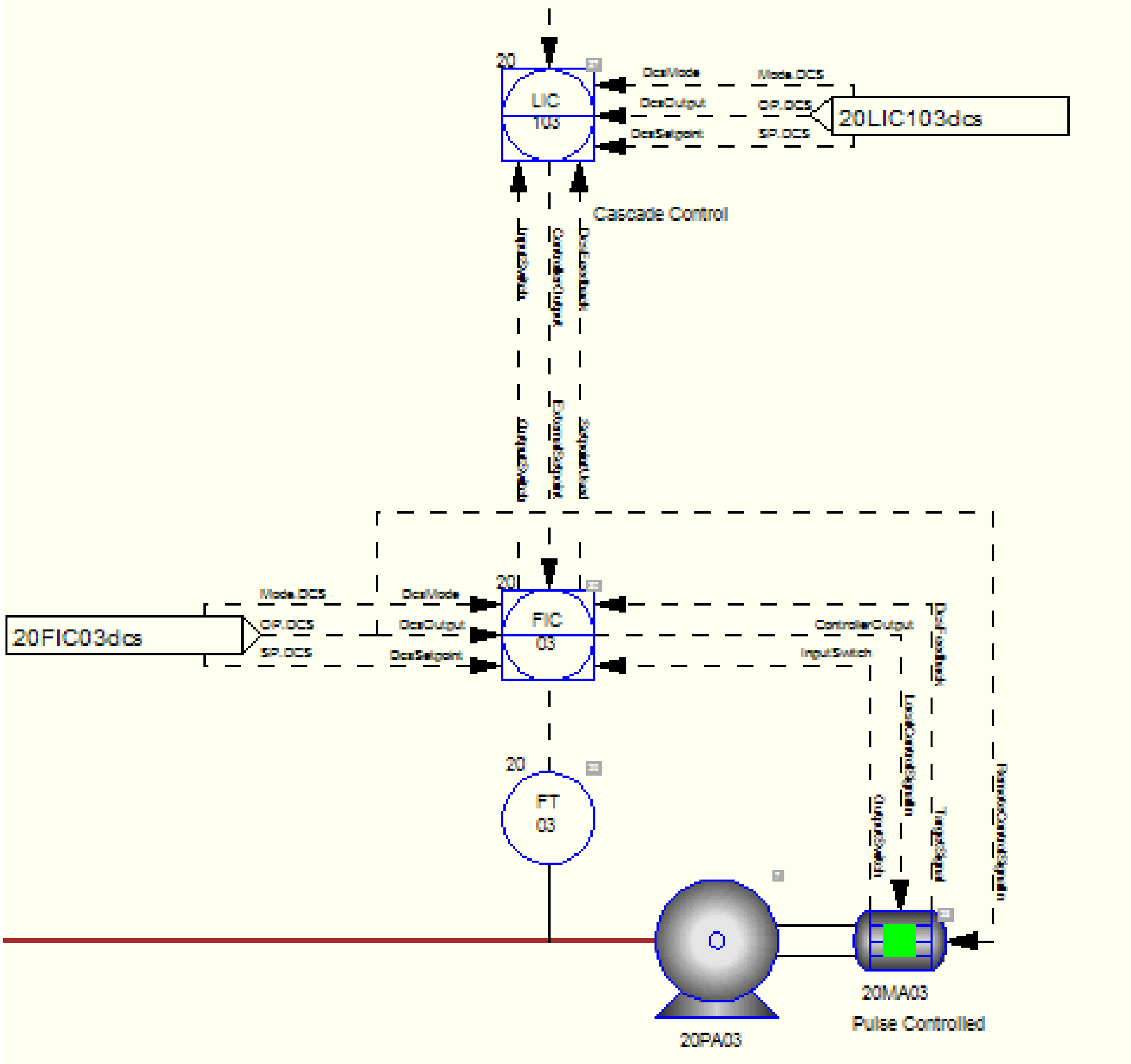
2.3.1 Basic Control Loop

Figure 1 Basic Control Loop



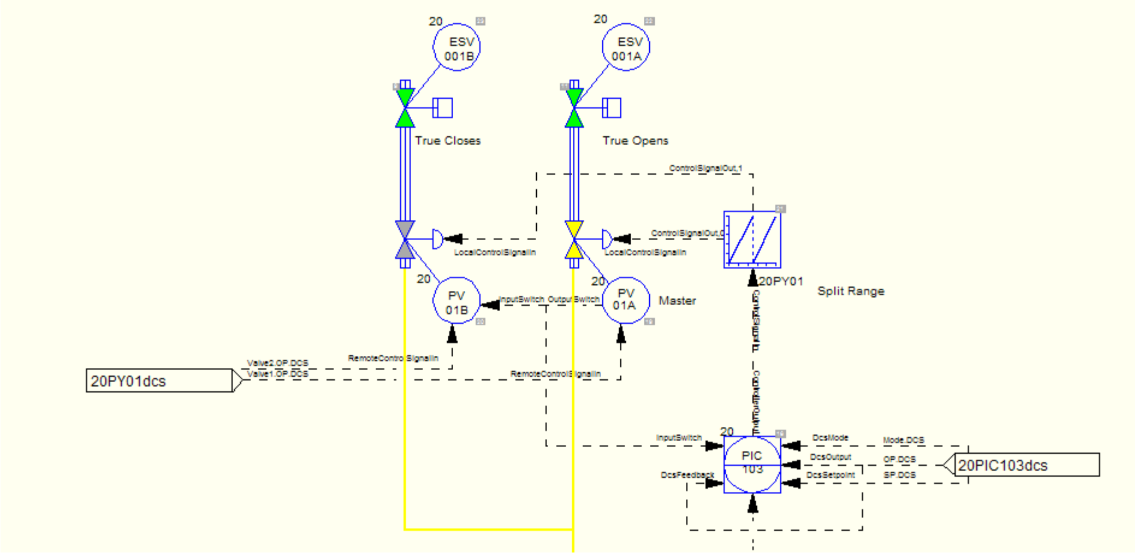
2.3.2 Cascade Control

Figure 2 Cascade Control



2.3.3 Split Range Control

Figure 3 Split Range Control



3 Theory

This section has detailed descriptions of the calculations for basic controller functionality and PCDA functionality.

3.1 Basic Controller Functionality

3.1.1 The Measurement Input

The Measurement input should be connected to an output from a FieldTransmitter module. It can be connected from a scaled control signal or an unscaled process value in engineering units. In any case the PidController module will adopt the unit category from the variable connected on Measurement and apply this on all its data items representing measurements, setpoints and their associates.

3.1.2 Input Range

The measurement range is set by the parameters RangeLowLimit and RangeHighLimit. The default range is 0–1 in the system unit, which is correct when the controller is working in control signal units.

3.1.3 Output Unit

By default the output of the controller ControllerOutput is in control signal units. This means that the system unit is fraction, i.e. 0–1, but the display unit can be different, for example %. The output unit can be changed using the parameter OutputUnitCategory. To see a list of the available unit categories you can open the Engineering Units panel (from the control menu select *“Engineering Units...”*). The unit categories is in the left column labelled *“Unit name”*. For example, if you enter **pressure** into OutputUnitCategory, the system unit for the Controller output will be Pa and the display unit will be the selected display unit for the **pressure** category.

3.1.4 Normalized Output Limits and Scaled Output Range

The limits for the normalized output from the PID algorithm are defined using the parameters OutputLowLimit and OutputHighLimit. These are in control signal unit, and the default limits are 0–1 (in system unit, fraction).

The scaled output range is set by the parameters OutputRangeLowLimit and OutputRangeHighLimit. The default range is 0–1 in the system unit, which is correct when control signal units is used on the output. The scaled output range is used to scale the normalized output from the PID algorithm to an engineering unit, typically for use as a setpoint in a cascaded controller.

If the output is in control signal unit it is actually normalized, so then it should not be scaled. If you then need different output limits than 0-1 you should instead use the normalized output limits, so that you not also unawares change the overall control loop gain.

If the parameter OutputUnitCategory is cleared then the unit category of the output is set back to control signal and the scaled output range is automatically set back to 0–1.

3.1.5 Output Rate of Change Limits

The controller can optionally be configured to have limits on how fast the output can increase or decrease. The rate of decrease and the rate of increase can be specified separately.

The parameter OutputClampDown sets the maximum rate of decrease for the output. This is specified as a rate of control signal change per unit time. Rate limiting can be turned off by setting OutputClampDown to zero.

The parameter OutputClampUp sets the maximum rate of increase for the output. This is specified as a rate of control signal change per unit time. Rate limiting can be turned off by setting OutputClampUp to zero.

Note

These limits also apply to output changes when the controller is in Manual mode.

3.1.6 On/Off Controller

The controller can be configured to represent an on/off controller. This is done by setting the OnOffMode parameter to **On/Off control**. A hysteresis parameter, Hysteresis can also be specified, which defines an interval where the controller output is retained. The controller will switch to the output high limit when the error is positive, and switch the low limit when the error is below – Hysteresis. The control algorithm is then:

$$C = \begin{cases} C_{max} & E > 0 \\ C_{min} & E < -h \end{cases}$$

Where:

$$E = \frac{M - S}{span}$$

and:

M = Measurement

S = Setpoint

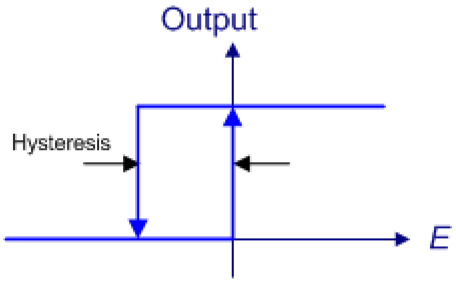
span = Span of the input range = OutputRangeHighLimit — OutputRangeLowLimit

C = Controller output, ControllerOutput

C_{max} = Output high limit, OutputHighLimit

C_{min} = Output low limit, OutputLowLimit

h = Hysteresis interval, Hysteresis



On/Off control with hysteresis and control action = Direct

If the control action is set to **Reverse**, the controller will instead switch to the high limit when the error is negative, and switch to the low limit when the error is bigger than Hysteresis.

3.1.7 PID Controller Algorithm

The module simulates a PID controller in position form if the OnOffMode parameter is set to **Modulating control**. The controller output is found from

$$C = K_p G_s \left(E + \frac{1}{T_i} \int_0^t E dt + T_d D \right) + b$$

Where:

K_p = Proportional gain, Gain

G_s = Scheduling gain factor, GainSchedule

T_i = Integral time, IntegralTime

T_d = Derivative time, DerivativeTime

t = time, s

d = derivation operator

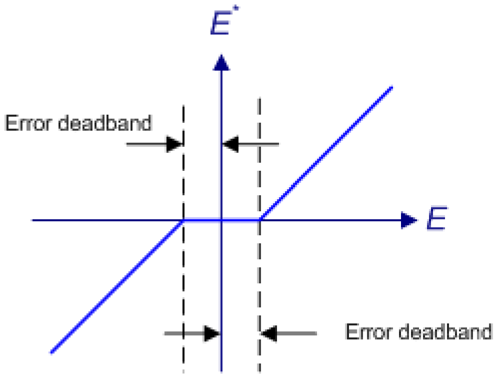
D = derivative term, see below

b = bias

A PI-controller may be obtained by setting the derivative time to zero, a PD-controller by setting the integral time to zero, and a P-controller by setting both the integral and the derivative times to zero.

3.1.8 Error Dead Band

The parameter ErrorDeadband can be used to specify an error dead band. It defines a range of error where no action occurs, i.e. the controller output is dead. The error used in the controller will be set to zero if the magnitude of the observed error is less than the deadband, see figure below. The PID algorithm will use the deadband error E^* instead of the measured error E.



Error Deadband

3.1.9 Derivative Term

If DerivativeTime is non-zero a derivative term will be calculated and used in the controller. In its unfiltered form, i.e. if DerivativeFiltering is zero, this term is:

$$D = \frac{E - E_p}{\Delta t}$$

Where:

E = Error, normalized, Error

E_p = Error at the previous time step

Δt = Time step

However, this form can be unstable with noisy signals. In this case a filter can be used. This filter is defined by a parameter DerivativeFiltering, R_D, which is the ratio between derivative time and the filter time constant, T_f:

$$R_D = \frac{T_d}{T_f}$$

$$D = \frac{E - E_p + T_f D_p}{\Delta t + T_f}$$

where D_p is the filtered value from the last time step.

3.1.10 Error-squared Controller

The UseErrorSquaredInI parameter can be set to **true** to use an integral error-squared algorithm, as shown in the equation below:

$$C = K_p G_s \left(E + \frac{1}{T_i} \int_0^t |E| \, E dt + T_d D \right) + b$$

The UseErrorSquaredInP parameter can be set to **true** to use an proportional error-squared algorithm, as shown in the equation below:

$$C = K_p G_s \left(|E| \, E + \frac{1}{T_i} \int_0^t E dt + T_d D \right) + b$$

These parameters can be used together.

3.1.11 Feed-forward Control

A feed-forward signal can be connected to the FeedForward input. This contains a value in output units. Any conditioning, scaling or dynamic compensation of the feedback signal must be done in a block (such as a gain relay) that sends its output to this input.

The FeedForward signal is simply added to the output calculated by the PID algorithm above, i.e.:

$$C += F$$

3.1.12 Controller Mode

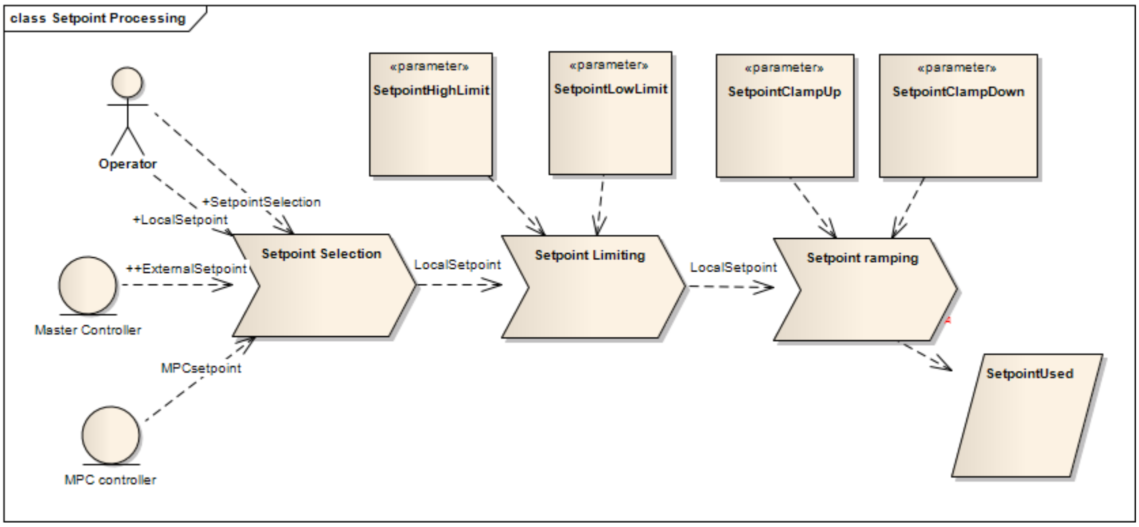
The controller mode is determined by the Mode input. This can have one of three values:

- **Auto** The output, ControllerOutput, is calculated using the control algorithm.

- **Manual** The output, ControllerOutput, can be set directly by the user if the ManualOutputOption flag is set to **Not used**. However, if ManualOutputOption has another value, the ManualOutput input is used. See the section *Externally-supplied manual output* below.
- **Computer** The output is set to the value of the ComputerOutput input.

3.1.13 Internal/External/MPC Settings / Cascade Control

The controller may operate with an internal/local set point, or with an external/remote setpoint, i.e. cascade control. The type of set point used is determined by the SetpointSelection input. The units of both the internal and the external setpoints are (must be) the same as the units of the measurement. The selected setpoint is further processed by checking against limits and rate of change limitations.



Setpoint processing

If SetpointSelection is set to **MPC**, the set point in the controller is read from the MPCsetpoint input. This allows a computer-implemented controller to manipulate the set point in parallel with a back-up cascade controller.

3.1.14 Set Point Limits

The parameters SetpointHighLimit and SetpointLowLimit, if non zero, impose limits on the set point. This applies to all set points used by the controller, internal, external and MPC.

3.1.15 Set Point Rate Of Change Clamping

Rate of change limits for the set point in the increasing direction can be implemented by setting a non-zero value of SetpointClampUp. The rate of change is given in a unit relative to the measurement range, for example in %/s. This means that the set point used in the model will only change by SetpointClampUp$*\Delta t$ in each time step.

The rate of change limit in the decreasing direction can be implemented using SetpointClampDown.

Note

Enter a value of zero to disable rate of change clamping. The set point will then move to its requested value in the next time step.

3.1.16 Set Point Tracking

The user can enable internal set point tracking by setting SetpointTracking to **true**.

3.1.17 Anti reset-windup control

Anti reset-windup control may be specified by setting the parameter AntiResetWindup to **On**. In that case, the error is not integrated if:

$$\begin{array}{c} C \leq C_{min} \wedge E < 0 \\ \vee \\ C \geq C_{max} \wedge E > 0 \end{array}$$

3.1.18 Signalling of Non Availability to Master Controller

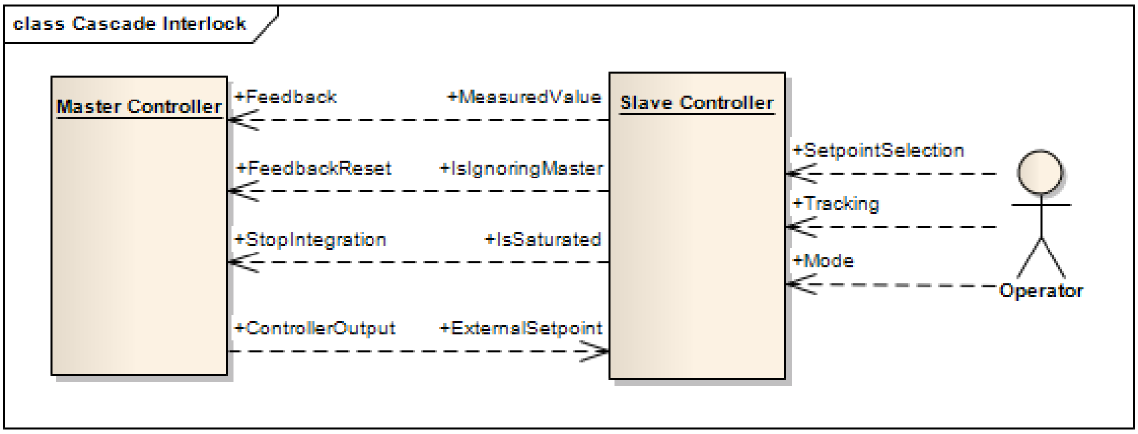
If the controller is a slave controller in a cascade setup, it will need to inform its master when it is ignoring the external set point. This will provide a bumpless transfer when the master regains control. This can be done by connecting the IsIgnoringMaster output on the slave to the FeedbackReset or Tracking input on the master. You will also need to connect the MeasuredValue output on the slave to the Feedback input on the master.

The IsIgnoringMaster output is **true** if one of the following is true:

- the controller is in manual control mode
- the controller is using its local setpoint
- the controller is tracking its feedback

The MeasuredValue output contains the current value of the measurement input to the slave controller.

Recall that the output of the master is the set point of the slave. This means that the master's output will be reset to the measurement input to the slave – so that there will be no bump in control signal when the master regains control.



Connecting a Master Controller to Slave Controller to ensure bumpless transfer between internal and external setpoint mode.

3.1.19 Signalling of Saturation to Master Controller

The IsSaturated output can be used to signal to a master controller that the slave's output is saturated. This is done by connecting it to the StopIntegration input on the master controller. The IsSaturated output is **true** if the normalized output is limited by either the high or low output limit.

If StopIntegration is **true**, the controller will not update its integral term while in automatic mode. However, if the controller itself also is saturated and not in feedback reset, it will ignore the StopIntegration signal. This is to avoid the controller getting stuck when both master and slave are saturated.

3.1.20 Externally-supplied manual output

The value to the written to the output when the controller is in manual can be supplied in the ManualOutput input. This value is then written to the output if the ManualOutputOption parameter is not set to **Do not use**.

The other values of ManualOutputOption are:

- **Use and writeback**: The value of ManualOutput is used to set ControllerOutput and the value is written back to the connected upstream module.
- **Use not writeback**: The value ManualOutput is used to set ControllerOutput, but it is not written back to the upstream module.

3.1.21 Computer control

When the controller is in computer control mode, i.e. Mode is **Computer**, it copies the value of ControllerOutput to the output ControllerOutput. This allows an external block or algorithm to calculate a control algorithm with the possibility of fall back to either the internal automatic control or manual operation.

The integral term in the controller is back-calculated so that a bumpless change can be made back to internal automatic control.

3.1.22 Bumpless transfer

Bumpless transfer of set point and/or tuning parameters is supported. It is based on back calculating the value of the controller integrated error that maintains the calculated output at the same value as it had before a change was made.

Bumpless Transfer of Set Point

Bumpless transfer of setpoint is supported if SetpointBumplessTransfer is **true**.

When the setpoint has been changed, the integrated error is back calculated to be the value that will give the same output with the new set point.

$$I = \left[\frac{C - b - F}{K_p G_s} - sign\left(E_p - \frac{S_p - S}{span} - T_d D_p\right)\right] T_i$$

where D_p is the derivative term from the last time step. The value of s depends on the chosen value of action. If action is Direct, *sign* is +1. If it is Reverse, *sign* is –1.

Bumpless Transfer of Parameters

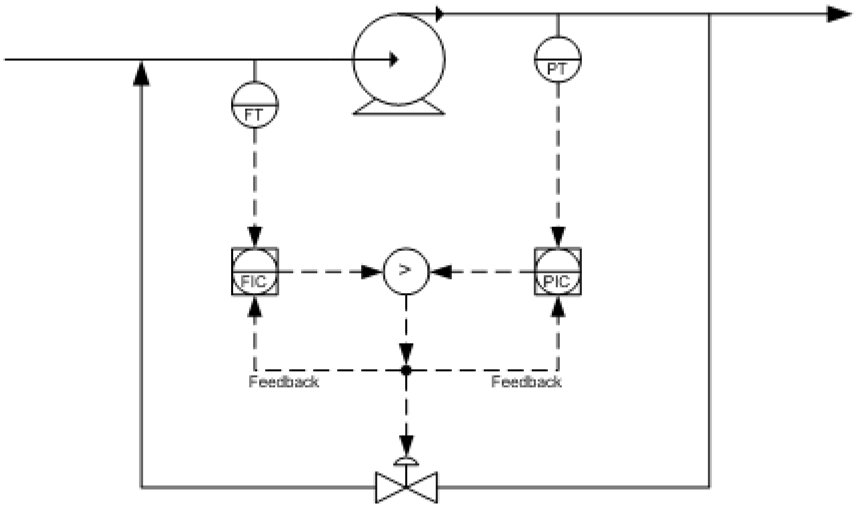
Bumpless transfer of gain and integral time is always on.

3.1.23 Feedback Tracking

If the Tracking flag is set, the output is set equal to the Feedback input variable, and the integral is updated so that it will give a bumpless transfer when the Tracking switch is reset.

3.1.24 Feedback Reset and Override Control

If the FeedbackReset flag is set, then the output is set equal to the Feedback input variable and the integral is updated before the actual PID control algorithm is executed. This functionality can be useful in override control for the purpose of preventing controller reset windup when the controller is not selected. The selector output is fed back to both controllers and the FeedbackReset flag is set on both controllers. Note that the feedback reset will not have any effect for the controller being selected, since the feedback signal will be equal to its own output.



Example of override control

3.1.25 Gain Scheduling

Gain scheduling can be implemented by externally calculating the gain scheduling factor GainSchedule, G_S. This factor is multiplied by the Gain in the PID algorithm.

3.1.26 Manual Output

The ManualOutputOption parameter governs if, and how, the input ManualOutput is used. The following options exist:

- **Do not use**: This is the normal operation. ManualOutput is not used.
- **Use with write-back**: The ManualOutput is transferred to the ControllerOutput if the controller is in manual mode, and the ControllerOutput is transferred back to the ManualOutput if the controller is in automatic mode. The value is also written back to the variable connected to the ManualOutput.
- **Use without write-back**: The ManualOutput is transferred to the ControllerOutput if the controller is in manual mode, and the ControllerOutput is transferred back to the ManualOutput if the controller is in automatic mode. In this case the value is not written back to the variable connected to the ManualOutput.

3.2 PCDA Mode

The controller can be set in PCDA mode by switching the PCDA parameter to **On**. Regardless of the setting of PCDA the controller supports all the basic functionality described above. When the controller is in PCDA mode it additionally supports:

- Trip signals.
- Instructor-induced failures.
- Valve position reporting and alarms.

3.2.1 Trip

The controller output can be set to a safe value in response to a trip signal. This safe value is specified in the parameter SafePosition. This is a value in the same units as the normal controller output.

Configuration

The trip action for the controller is only available if the PCDA flag is set to **On**.

Trip Signals

Process shutdown is initiated by the input signals TripSignal or PermissiveSignal. The TripSignal is **true** for trip and **false** for normal operation. The PermissiveSignal is the other way around. A trip is activated if any of the two turns to the bad position.

Blocking the Trip Signal

The effect of the trip signal can be blocked by setting the BlockTripSignals input to **true**.

Switching to Manual Mode

If TripOptions is set to **Lock**, the controller will be placed in manual mode when trip is activated.

If TripOptions is set to **Force**, the controller will remain in automatic mode when trip is activated. The integral will be back calculated to ensure bumpless transfer when the trip is cleared.

3.2.2 Coincidence

The Coincidence output will report a problem if there is a trip while the BlockTripSignals signal is **true**.

3.2.3 External Locks

Switching to Automatic Mode

If the SwitchToAuto input is **true**, the controller will be set and held in automatic mode. This input is overridden by a process shutdown.

Switching to External Setpoint

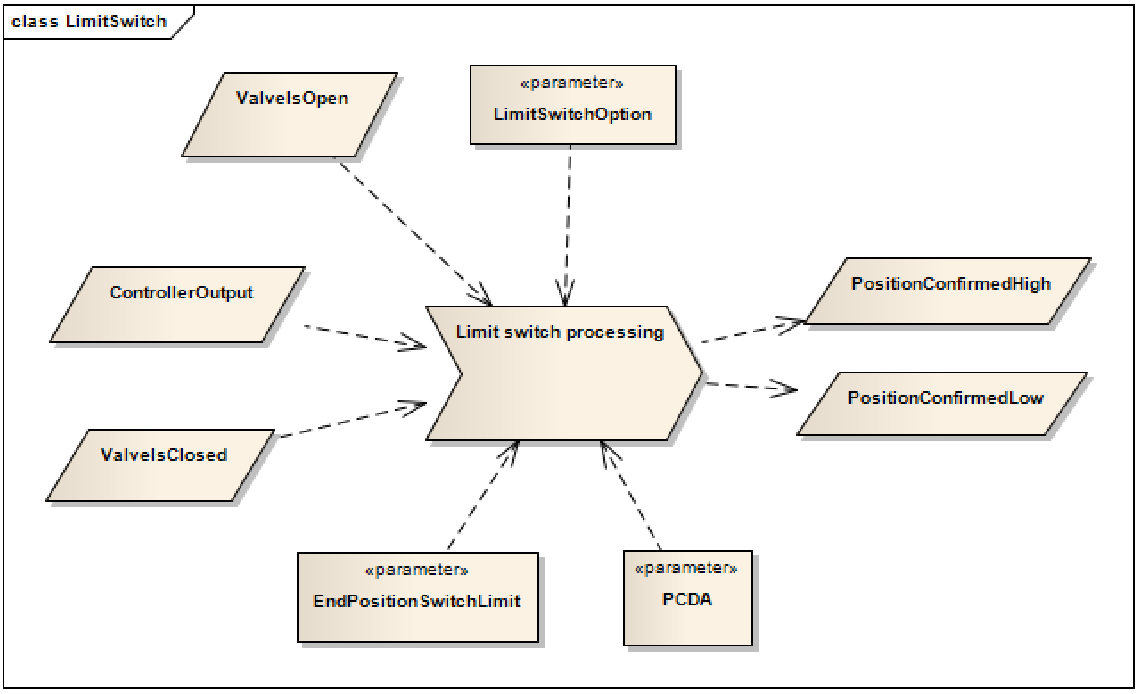
The SwitchToExternal input, if it is **true** will switch the controller to use the external setpoint, i.e. SetpointSelection is switched to **External**.

3.2.4 Valve Position Reporting

The inputs ValveIsOpen and ValveIsClosed can be connected to corresponding signals from a valve or valve actuator. The tag ValveIsOpen should be **true** when the valve is fully open, The tag ValveIsClosed should be **true** when the valve is fully closed.

The controller can then be configured to report on one of these limits, depending on the value of the LimitSwitchOption parameter.

- If LimitSwitchOption is **None** (0), the valve positions are ignored. The outputs PositionConfirmedLow and PositionConfirmedHigh are set to **true**.
- If LimitSwitchOption is **Low** (1), the valve closed position is processed. The output PositionConfirmedHigh is set to **true** but the value of PositionConfirmedLow depends on the value of the controller output. If the output is less than EndPositionSwitchLimit and the valve is closed, the PositionConfirmedLow output will be set to **true**. It is **false** otherwise.
- If LimitSwitchOption is **High** (2), the valve open position is processed. The output PositionConfirmedLow is set to **true** but the value of PositionConfirmedHigh depends on the value of the controller output. If the output is greater than 1-EndPositionSwitchLimit and the valve is open, the PositionConfirmedHigh output will be set to **true**. It is **false** otherwise.



Limit Switch Processing.

3.2.5 Failure Alarm

A failure alarm is triggered if a mismatch is detected between the shutdown signal and the limit switch status. Failure annunciation requires acknowledgement.

External Failure

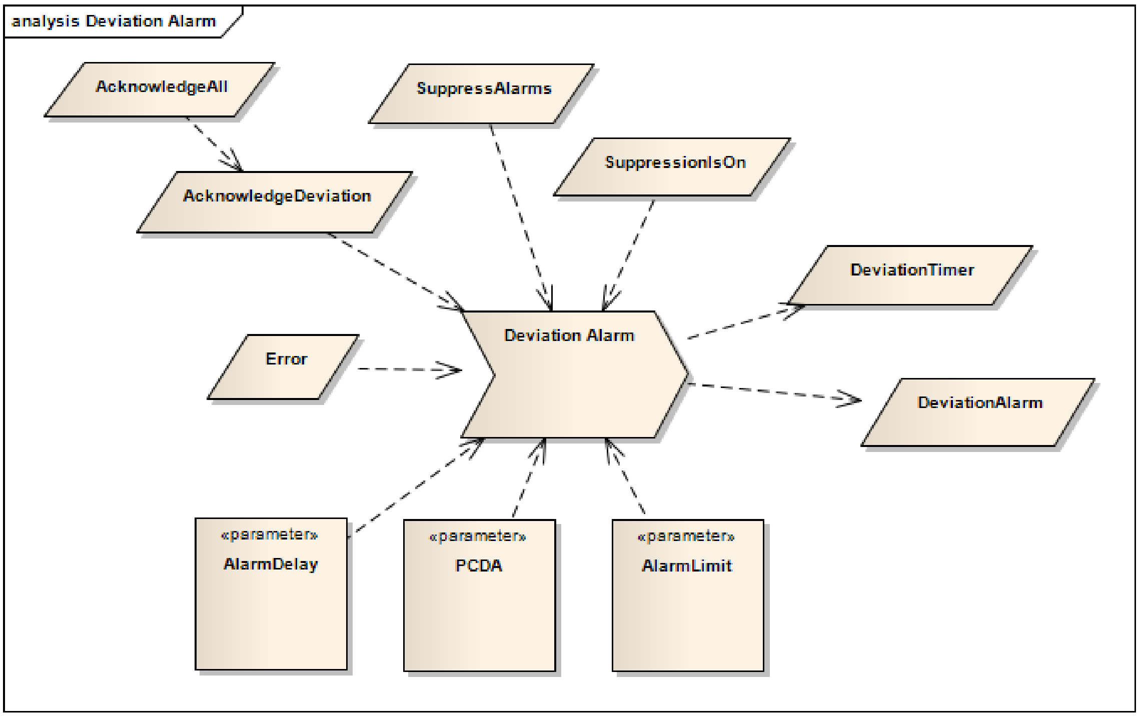
The failure alarm can also be triggered by setting the ExternalFailure input to **true**. The alarm will persist until this input is set to **false**.

3.2.6 Deviation Alarm

A deviation alarm is triggered if the absolute value of the normalized error has exceeded a threshold value, AlarmLimit, for a specified time, AlarmDelay.

The alarm is announced through the output DeviationAlarm, which has the value **true** for alarm. However, this signal is maintained until both the condition ceases and the alarm has been acknowledged. The alarm is acknowledged by setting the variable AcknowledgeDeviation to **true**.

The time remaining before the alarm is triggered is displayed in DeviationTimer if a deviation condition has occurred.



Deviation alarm calculations

3.2.7 Suppression of Alarms

Alarms can be suppressed using one or both of:

- The SuppressAlarms variable. This is used to simulate operator suppression of alarms.
- The SuppressionIsOn input. This can be connected from other modules in the simulation and is used to simulate automatic suppression by the control system.

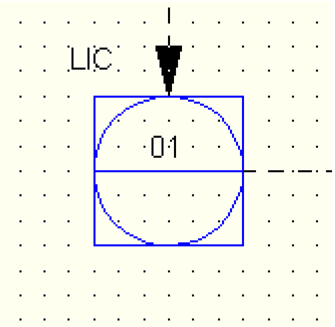
If either of these variables are **true**, the failure and deviation alarms are set to **false** and the acknowledge flags are reset to **false**.

4 Symbols and Dialogs

This section describes the user interface to the DcsPidController module, symbols, menu and dialogs.

4.1 Symbols

The following symbols can be used to represent a DcsPidController module:



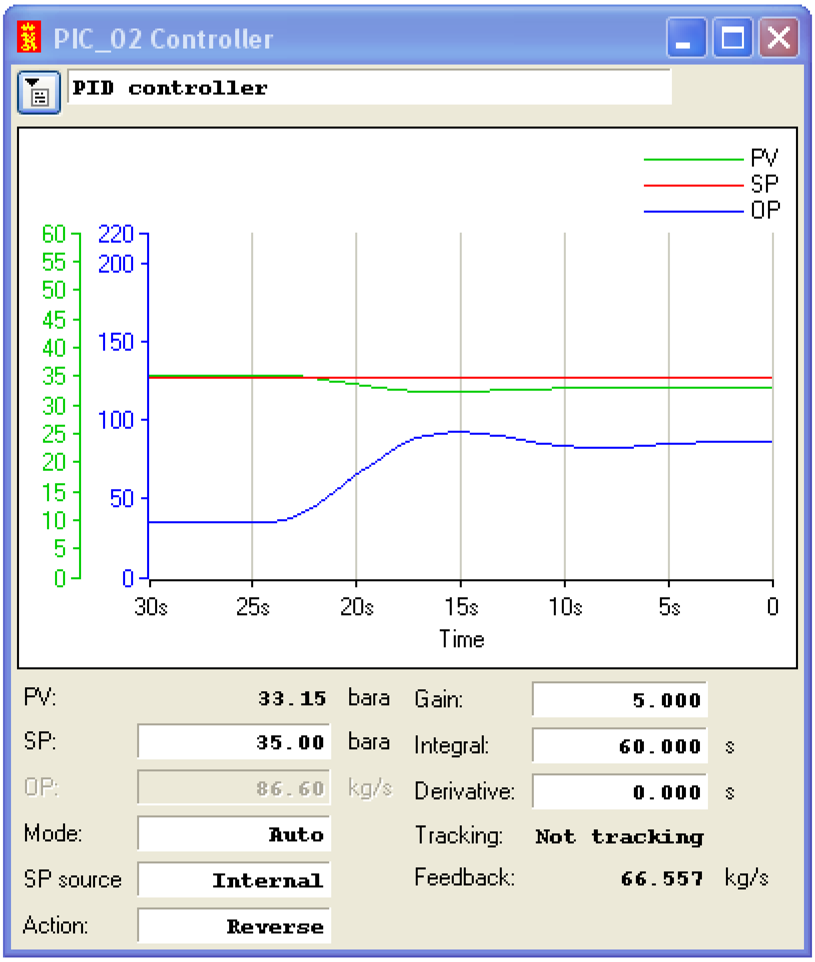
4.2 Dialogs

This section describes the dialogs belonging to the DcsPidController module.

Faceplate

The **PidController** faceplate dialogs shows the key values.

Figure 4 Faceplate Dialog



PIC_02 Controller

PID controller

PV: 33.15 bara

SP: 35.00 bara

OP: 86.60 kg/s

Mode: Auto

SP source Internal

Action: Reverse

Gain: 5.000

Integral: 60.000 s

Derivative: 0.000 s

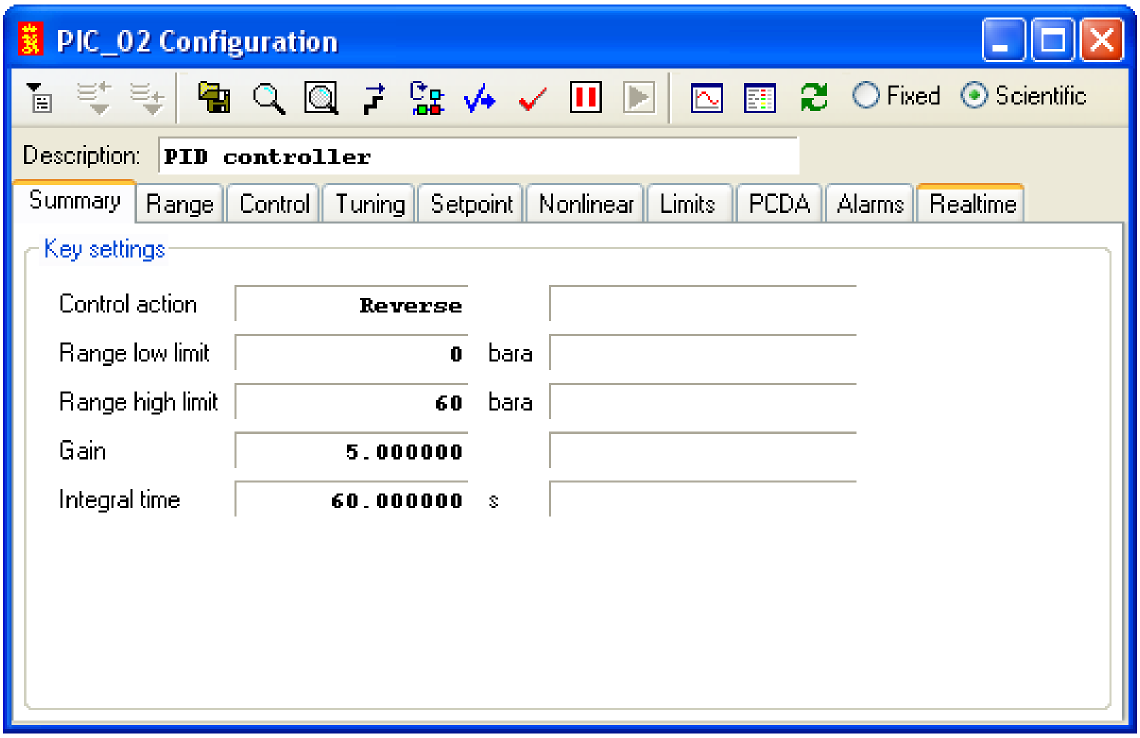
Tracking: Not tracking

Feedback: 66.557 kg/s

Configuration dialog

The Configuration dialog is displayed by choosing *Configuration…* from the block’s pop-up menu. It has 10 pages, one for each data item category. The summary page appears first and shows the key settings of the controller.

Figure 5 Summary page



Description: PID controller

Key settings

Control action	Reverse	
Range low limit	0	bara
Range high limit	60	bara
Gain	5.000000	
Integral time	60.000000	s

5 Module and Data Item Reference

5.1 Item Type Cross-Reference

5.1.1 Input Data Items

- BlockTripSignals, Block trip signals.
- ComputerOutput, Output used when Mode is set to "Computer"..
- DcsFeedback, Dcs feedback signal.
- DcsMode, Dcs Auto/Manual/Computer mode.
- DcsOutput, Dcs controller output.
- DcsSetpoint, Dcs setpoint.
- ExternalFailure, External failure.
- ExternalSetpoint, External setpoint.
- Feedback, Feedback signal.
- FeedbackReset, Reset switch, when true the controller output is reset to the feedback signal before each execution..
- FeedForward, Feedforward signal.
- GainSchedule, Gain Schedule.
- InputSwitch, Input switch between local, remote and manual control.
- ManualOutput, Output used when Mode is set to "Manual"..
- Measurement, Measurement value.
- Mode, Auto/Manual/Computer mode.
- MPCsetpoint, Setpoint from MPC system.
- PermissiveSignal, Permissive signal (false if tripped).
- SetpointSelection, Internal/External setpoint mode.
- StopIntegration, StopIntegration flag..
- SuppressionIsOn, When true the alarms are suppressed.
- SwitchToAuto, Switch controller to auto mode.
- SwitchToExternal, Switch controller to external setpoint mode.

- Tracking, Tracking switch, when true the controller output is tracking the feedback signal..
- TripSignal, Trip signal (true if tripped).
- ValveIsClosed, Feedback signal from limit switch low.
- ValveIsOpen, Feedback signal from limit switch high.

5.1.2 Output Data Items

- Coincidence, Coincidence status signal..
- ControllerActive, Show if the controller is active or not.
- ControllerOutput, Controller output.
- DeviationAlarm, Deviation alarm signal.
- EffectiveGain, Effective gain, including Gain Schedule.
- Error, Normalized deviation from setpoint.
- HasFailed, Failure status signal.
- InternalSetpoint, Internal setpoint.
- IsIgnoringMaster, Mode signal to master controller.
- IsSaturated, Saturation flag. When true the controller is saturated.
- MeasuredValue, Track value for cascade master.
- OutputMode, Output Auto/Manual/Computer mode.
- OutputSwitch, Output switch between local, remote and manual control.
- PositionConfirmedHigh, Position confirmed high.
- PositionConfirmedLow, Position confirmed low.
- PreviousMode, Auto/Manual mode used in the previous execution.
- SetpointUsed, Setpoint used.
- Tripped, Trip status of controller (true if tripped).

5.1.3 Parameter Data Items

- Action, Control action.
- AlarmDelay, Delay of alarms and warnings.
- AlarmLimit, Alarm limit for the normalized deviation.
- AntiResetWindup, Anti reset-windup option.
- Bias, Bias on output.
- DCStag, Corresponding DCS block (instructor station).
- DerivativeFiltering, Derivative filtering ratio (Td/Tdf).
- DerivativeTime, Derivative time.
- EndPositionSwitchLimit, Tolerance for position high/low confirm.
- ErrorDeadband, Error deadband.
- Gain, Gain.

- GainLA, Gain in LA mode, not in use.
- GainPL, Gain in PL mode, not in use.
- GainRT, Gain in RT mode, not in use.
- Hysteresis, Hysteresis interval for on/off control.
- IntegralTime, Integral time.
- IntegralTimeLA, Integral time in LA mode, not in use.
- IntegralTimePL, Integral time in PL mode, not in use.
- IntegralTimeRT, Integral time in RT mode, not in use.
- LimitSwitchOption, Limit switch configuration.
- ManualOutputOption, Options for use of manual output..
- OnOffMode, Switch for modulating/ON-OFF control..
- OutputClampDown, Output decreasing rate of change clamp.
- OutputClampUp, Output increasing rate of change clamp.
- OutputHighLimit, High limit for normalized output.
- OutputLowLimit, Low limit for normalized output.
- OutputRangeHighLimit, Scaled output range high limit.
- OutputRangeLowLimit, Scaled output range low limit.
- OutputUnitCategory, Unit category for controller output.
- PCDA, Supports PCDA functions.
- RangeHighLimit, Input range high limit.
- RangeLowLimit, Input range low limit.
- SafePosition, Safe controller output position at trip.
- SetpointBumplessTransfer, Bumpless transfer of setpoint.
- SetpointClampDown, Negative setpoint rate of change clamp.
- SetpointClampUp, Positive setpoint rate of change clamp.
- SetpointHighLimit, Maximum limit for setpoint.
- SetpointLowLimit, Minimum limit for setpoint.
- SetpointTracking, Setpoint tracking while in manual.
- TripOptions, Options for trip.
- TripPositionOption, Output position action when tripped.
- UseErrorSquaredInI, Use error squared in integral term.
- UseErrorSquaredInP, Use error squared in proportional term.
- UseSpecialRTparameters, Use special tuning parameters in RT mode, not in use..
- WriteBackExternalSetpoint, Write back external setpoint.

5.1.4 State Variable Data Items

- DeviationTimer, Timer for deviation warning annunciation.

- `FailTimer`, Timer for failure alarm annunciation.
- `FilteredDterm`, Filtered derivative term.
- `Integral`, The time integral of the normalized deviation.
- `NormalizedOutput`, Normalized controller output.
- `ParameterDisplaySwitch`, Switch for display of tuning parameters in the custom panel, not in use.
- `PreviousGain`, Effective gain from the previous execution.
- `PreviousIntegralTime`, Integral time from the previous execution.
- `PreviousSetpoint`, Setpoint from the previous execution.
- `ReferenceSetpoint`, Reference setpoint.
- `SuppressAlarms`, Suppress all alarms and warnings (from OS).

5.1.5 Working Variable Data Items

- `AcknowledgeAll`, Acknowledge all alarms and warnings.
- `AcknowledgeDeviation`, Acknowledge deviation warning.
- `AcknowledgeFailure`, Acknowledge failure alarm.
- `Saturation`, Flag indicating saturation.

5.2 Data Item Reference

5.2.1 Category Range: Signal scaling

Signal scaling

5.2.1.1 `RangeLowLimit`: Input range low limit

This defines the low limit for the measurement range. The input range span is used to normalize the error used in the PID algorithm.

Item Type	Variable Type	Units Category
parameter	double	The same unit as a data item Measurement

Range Checks and Default Values

The default value for this data item is: *0.0*.

5.2.1.2 `RangeHighLimit`: Input range high limit

This defines the high limit for the measurement range. The input range span is used to normalize the error used in the PID algorithm.

Item Type	Variable Type	Units Category
parameter	double	The same unit as a data item Measurement

Range Checks and Default Values

The default value for this data item is: *1.0*.

5.2.1.3 `OutputRangeLowLimit`: Scaled output range low limit

The scaled output range limits are used to scale the normalized output from the PID algorithm, if necessary. If the output is using control signal unit the output should not be scaled, so then the scaled output range limits should be 0-1.

Item Type	Variable Type	Units Category
parameter	double	

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *0.0*.

5.2.1.4 `OutputRangeHighLimit`: Scaled output range high limit

The scaled output range limits are used to scale the normalized output from the PID algorithm, if necessary. If the output is using control signal unit the output should not be scaled, so then the scaled output range limits should be 0-1.

Item Type	Variable Type	Units Category
parameter	double	

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *1.0*.

5.2.2 Category Tuning: Tuning

Tuning

5.2.2.1 `DerivativeTime`: Derivative time

Derivative time

Item Type	Variable Type	Units Category
parameter	double	time

Range Checks and Default Values

The engineering unit for data item here is: s.

The default value for this data item is: *0.0*.

The minimum permissible value of this data item is: 0.0. This limit is inclusive, i.e., the limit itself is allowed as a value.

5.2.2.2 `IntegralTime`: Integral time

Integral time

Item Type	Variable Type	Units Category
parameter	double	time

Range Checks and Default Values

The engineering unit for data item here is: s.

The default value for this data item is: *300.0*.

The minimum permissible value of this data item is: 0.0. This limit is inclusive, i.e., the limit itself is allowed as a value.

5.2.2.3 `Gain`: Gain

Gain

Item Type	Variable Type	Units Category
parameter	double	

Range Checks and Default Values

The default value for this data item is: *1.0*.

The minimum permissible value of this data item is: 0.0. This limit is inclusive, i.e., the limit itself is allowed as a value.

5.2.2.4 `DerivativeFiltering`: Derivative filtering ratio (Td/Tdf)

Derivative filtering ratio (Td/Tdf).

Item Type	Variable Type	Units Category
parameter	double	fraction

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *0.0*.

The minimum permissible value of this data item is: 0.0. This limit is inclusive, i.e., the limit itself is allowed as a value.

5.2.2.5 `ErrorDeadband`: Error deadband

The controller will ignore relative deviations from setpoint that are less than the deadband.

Item Type	Variable Type	Units Category
parameter	double	control_signal

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *0.0*.

The minimum permissible value of this data item is: 0.0. This limit is inclusive, i.e., the limit itself is allowed as a value.

5.2.3 Category Control: Control

Control

5.2.3.1 `DcsMode`: Dcs Auto/Manual/Computer mode

Dcs Auto/Manual/Computer mode, used if InputSwitch = Remote, takes precedence on Mode.

Item Type	Variable Type	Units Category
input	integer	

Enumerated Options

Option	Numerical Value	Description
Auto	0	Auto
Manual	1	Manual
Computer	2	Computer

5.2.3.2 `DcsFeedback`: Dcs feedback signal

Dcs feedback signal, used if InputSwitch = Remote or Manual, takes precedence on Feedback.

Item Type	Variable Type	Units Category
input	double	control_signal

5.2.3.3 `InputSwitch`: Input switch between local, remote and manual control

Input switch between local, remote and manual control.

Item Type	Variable Type	Units Category
input	integer	

Enumerated Options

Option	Numerical Value	Description
Local	0	Local
Remote	1	Remote
Manual	2	Manual

5.2.3.4 `OutputSwitch`: Output switch between local, remote and manual control

Output switch between local, remote and manual control.

Item Type	Variable Type	Units Category
output	integer	

Enumerated Options

Option	Numerical Value	Description
Local	0	Local
Remote	1	Remote
Manual	2	Manual

5.2.3.5 `Measurement`: Measurement value

Measurement value.

Item Type	Variable Type	Units Category
input	double	The same unit as a connected data item.

5.2.3.6 `Feedback`: Feedback signal

Feedback signal.

Item Type	Variable Type	Units Category
input	double	control_signal

5.2.3.7 `FeedbackReset`: Reset switch, when true the controller output is reset to the feedback signal before each execution.

Reset switch, when true the controller output is reset to the feedback signal before each execution.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.3.8 `Tracking`: Tracking switch, when true the controller output is tracking the feedback signal.

Tracking switch, when true the controller output is tracking the feedback signal.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.3.9 `StopIntegration`: StopIntegration flag.

When true, the controller will not update its integral term while in automatic mode. However, if the controller itself also is saturated and not in feedback reset, it will ignore the StopIntegration signal. This is to avoid the controller getting stuck when both master and slave are saturated. The IsSaturated output from the slave controller should be connected to this input to prevent windup of master when the slave is saturated.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.3.10 `Mode`: Auto/Manual/Computer mode

Auto/Manual/Computer mode.

Item Type	Variable Type	Units Category
input	integer	

Enumerated Options

Option	Numerical Value	Description
Auto	0	Auto
Manual	1	Manual
Computer	2	Computer

5.2.3.11 `SetpointSelection`: Internal/External setpoint mode

Internal/External setpoint mode.

Item Type	Variable Type	Units Category
input	integer	

Enumerated Options

Option	Numerical Value	Description
Internal	0	Internal
External	1	External
MPC	2	MPC

5.2.3.12 `FeedForward`: Feedforward signal

Feedforward signal.

Item Type	Variable Type	Units Category
input	double	control_signal

5.2.3.13 `ManualOutput`: Output used when Mode is set to "Manual".

Output used when Mode is set to "Manual".

Item Type	Variable Type	Units Category
input	double	control_signal

5.2.3.14 `ComputerOutput`: Output used when Mode is set to "Computer".

Output used when Mode is set to "Computer".

Item Type	Variable Type	Units Category
input	double	control_signal

5.2.3.15 `ControllerOutput`: Controller output

Controller output.

Item Type	Variable Type	Units Category
output	double	control_signal

5.2.3.16 `IsSaturated`: Saturation flag. When true the controller is saturated

The IsSaturated output can be used to signal to a master controller that the slave’s output is saturated. This output can be connected to the StopIntegration input on the master controller.

Item Type	Variable Type	Units Category
output	Boolean	

5.2.3.17 `IsIgnoringMaster`: Mode signal to master controller

If the controller is a slave controller in a cascade, it will need to inform its master when it is ignoring the external set point. This can be done by connecting the IsIgnoringMaster output on the slave to the FeedbackReset input on the master. You will also need to connect the MeasuredValue output on the slave to the feedback input on the master. The IsIgnoringMaster output is true if the slave controller is either in manual mode, is tracking or is in internal setpoint mode.

Item Type	Variable Type	Units Category
output	Boolean	

5.2.3.18 `MeasuredValue`: Track value for cascade master

The MeasuredValue output contains the current value of the Measurement input to the slave controller. This should be connected to the Feedback input of the master controller. Recall that the output of the master is the set point of the slave. This means that the master's output will track the input to the slave – so that there will be no bump in control signal when the master regains control.

Item Type	Variable Type	Units Category
output	double	The same unit as a data item Measurement

5.2.3.19 `Tripped`: Trip status of controller (true if tripped)

Trip status of controller (true if tripped).

Item Type	Variable Type	Units Category
output	Boolean	

5.2.3.20 `OutputMode`: Output Auto/Manual/Computer mode

Output Auto/Manual/Computer mode, used if InputSwitch = Remote, takes precedence on Mode.

Item Type	Variable Type	Units Category
output	integer	

Enumerated Options

Option	Numerical Value	Description
Auto	0	Auto
Manual	1	Manual
Computer	2	Computer

5.2.3.21 `Error`: Normalized deviation from setpoint

Normalized deviation from setpoint.

Item Type	Variable Type	Units Category
output	double	fraction

5.2.3.22 `ControllerActive`: Show if the controller is active or not

Throughput of the inverse of the Tracking switch, when true the controller is active (i.e. the Controller is NOT tracking the feedback signal).

Item Type	Variable Type	Units Category
output	Boolean	

5.2.3.23 `NormalizedOutput`: Normalized controller output

Normalized controller output.

Item Type	Variable Type	Units Category
state	double	control_signal

5.2.3.24 `Action`: Control action

Control action.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
Direct	0	Direct
Reverse	1	Reverse

5.2.3.25 `Saturation`: Flag indicating saturation

Flag indicating saturation.

Item Type	Variable Type	Units Category
working	integer	

Enumerated Options

Option	Numerical Value	Description
Rate clamped down	-2	Rate clamped down
Low	-1	Low
Off	0	Off
High	1	High
Rate clamped up	2	Rate clamped up

5.2.3.26 `Bias`: Bias on output

Bias on output.

Item Type	Variable Type	Units Category
parameter	double	control_signal

Range Checks and Default Values

The default value for this data item is: *0.0*.

5.2.3.27 `OutputUnitCategory`: Unit category for controller output

Unit category for controller output.

Item Type	Variable Type	Units Category
parameter	string	text

5.2.3.28 `ManualOutputOption`: Options for use of manual output.

This parameter governs if, and how, the input dataitem ManualOutput is used. The normal operation is that it is not used, i.e set to "Do not use". If it is set up to be used, either by "Use with write-back" or "Use without write-back", the ManualOutput is transferred to the ControllerOutput if the controller is in manual mode, and the ControllerOutput is transferred back to the ManualOutput if the controller is in automatic mode. Further, if it set to "Use with write-back", the value is also written back to the dataitem connected to the ManualOutput.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
Do not use	0	Do not use
Use with write-back	1	Use with write-back
Use without write-back	2	Use without write-back

5.2.3.29 `OnOffMode`: Switch for modulating/ON-OFF control.

Switch for modulating/ON-OFF control.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
Modulating control	0	Modulating control
On/Off control	1	On/Off control

5.2.3.30 `Hysteresis`: Hysteresis interval for on/off control

This parameter defines the interval for on/off control where the controller output is retained. The controller will switch to max output when the error is positive. When the error is below -Hysteresis it will switch to min output. If reverse action is selected the controller will instead switch to max output when the error becomes negative, and to min output when the error is above Hysteresis.

Item Type	Variable Type	Units Category
parameter	double	control_signal

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *0.0*.

5.2.3.31 `AntiResetWindup`: Anti reset-windup option

Anti reset-windup option.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
Off	0	Off
On	1	On

5.2.4 Category Setpoint: Setpoint management

Use this category to manage the set point used by the controller.

5.2.4.1 `DcsSetpoint`: Dcs setpoint

Dcs setpoint, used if InputSwitch = Remote.

Item Type	Variable Type	Units Category
input	double	The same unit as a connected data item.

Range Checks and Default Values

The default value for this data item is: *0.0*.

5.2.4.2 `ExternalSetpoint`: External setpoint

External setpoint.

Item Type	Variable Type	Units Category
input	double	The same unit as a connected data item.

Range Checks and Default Values

The default value for this data item is: *0.0*.

5.2.4.3 `SwitchToExternal`: Switch controller to external setpoint mode

Switch controller to external setpoint mode.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.4.4 `MPCsetpoint`: Setpoint from MPC system

Setpoint from MPC system.

Item Type	Variable Type	Units Category
input	double	The same unit as a data item Measurement

Range Checks and Default Values

The default value for this data item is: *0.0*.

5.2.4.5 `InternalSetpoint`: Internal setpoint

Internal setpoint.

Item Type	Variable Type	Units Category
output	double	The same unit as a data item Measurement

5.2.4.6 `SetpointUsed`: Setpoint used

This is the final setpoint after applying limits and rate of change clamps.

Item Type	Variable Type	Units Category
output	double	The same unit as a data item Measurement

5.2.4.7 `SetpointHighLimit`: Maximum limit for setpoint

Maximum limit for setpoint.

Item Type	Variable Type	Units Category
parameter	double	The same unit as a data item Measurement

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *1.0e+99*.

5.2.4.8 `SetpointLowLimit`: Minimum limit for setpoint

Minimum limit for setpoint.

Item Type	Variable Type	Units Category
parameter	double	The same unit as a data item Measurement

5.2.4.9 `SetpointClampUp`: Positive setpoint rate of change clamp

If this is set, the increase of *SetpointUsed* is clamped.

Item Type	Variable Type	Units Category
parameter	double	time_inverse

5.2.4.10 `SetpointClampDown`: Negative setpoint rate of change clamp

If this is set, the decrease of *SetpointUsed* is clamped.

Item Type	Variable Type	Units Category
parameter	double	time_inverse

5.2.4.11 `SetpointBumplessTransfer`: Bumpless transfer of setpoint

Bumpless transfer of setpoint.

Item Type	Variable Type	Units Category
parameter	Boolean	

5.2.4.12 `WriteBackExternalSetpoint`: Write back external setpoint

Write back external setpoint.

Item Type	Variable Type	Units Category
parameter	Boolean	

5.2.4.13 `SetpointTracking`: Setpoint tracking while in manual

Setpoint tracking while in manual.

Item Type	Variable Type	Units Category
parameter	Boolean	

5.2.4.14 `ReferenceSetpoint`: Reference setpoint

This is the last setpoint used while in auto mode.

Item Type	Variable Type	Units Category
state	double	The same unit as a data item Measurement

5.2.5 Category Nonlinear: Nonlinear PID control

Parameters and variables for non-linear or gain scheduling control

5.2.5.1 `GainSchedule`: Gain Schedule

Gain Schedule.

Item Type	Variable Type	Units Category
input	double	

5.2.5.2 `EffectiveGain`: Effective gain, including Gain Schedule

Effective gain including Gain Schedule.

Item Type	Variable Type	Units Category
output	double	

5.2.5.3 `UseErrorSquaredInP`: Use error squared in proportional term

When true the algorithm uses the square of the error in the proportional term.

Item Type	Variable Type	Units Category
parameter	Boolean	

5.2.5.4 `UseErrorSquaredInI`: Use error squared in integral term

When true the algorithm uses the square of the error in the integral term.

Item Type	Variable Type	Units Category
parameter	Boolean	

5.2.6 Category Limits: Limits, clamping and rate of change

Parameters and variables for setting limits, rate of change and clamping.

5.2.6.1 `OutputLowLimit`: Low limit for normalized output

Low limit for the normalized output from the PID algorithm.

Item Type	Variable Type	Units Category
parameter	double	control_signal

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *0.0*.

5.2.6.2 `OutputHighLimit`: High limit for normalized output

High limit for the normalized output from the PID algorithm.

Item Type	Variable Type	Units Category
parameter	double	control_signal

Range Checks and Default Values

The engineering unit for data item here is: fraction.

The default value for this data item is: *1.0*.

5.2.6.3 `OutputClampDown`: Output decreasing rate of change clamp

Output decreasing rate of change clamp.

Item Type	Variable Type	Units Category
parameter	double	time_inverse

5.2.6.4 `OutputClampUp`: Output increasing rate of change clamp

Output increasing rate of change clamp.

Item Type	Variable Type	Units Category
parameter	double	time_inverse

5.2.7 Category PCDA: Detailed PCDA emulation functions

Detailed PCDA emulation functions

5.2.7.1 `ValveIsClosed`: Feedback signal from limit switch low

Feedback signal from limit switch low.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.7.2 `ValveIsOpen`: Feedback signal from limit switch high

Feedback signal from limit switch high.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.7.3 `HasFailed`: Failure status signal

Failure status signal, true when the controller has a failure.

Item Type	Variable Type	Units Category
output	Boolean	

5.2.7.4 `Coincidence`: Coincidence status signal.

Coincidence status signal.

Item Type	Variable Type	Units Category
output	Boolean	

5.2.7.5 `DeviationAlarm`: Deviation alarm signal

Deviation alarm signal, turns true when |Error| > AlarmLimit for a time specified in AlarmDelay. Requires acknowledgement for resetting.

Item Type	Variable Type	Units Category
output	Boolean	

5.2.7.6 `PositionConfirmedHigh`: Position confirmed high

Position confirmed high.

Item Type	Variable Type	Units Category
output	Boolean	

5.2.7.7 `PositionConfirmedLow`: Position confirmed low

Position confirmed low.

Item Type	Variable Type	Units Category
output	Boolean	

5.2.7.8 `EndPositionSwitchLimit`: Tolerance for position high/low confirm

Tolerance for position high/low confirm.

Item Type	Variable Type	Units Category
parameter	double	control_signal

5.2.7.9 `PCDA`: Supports PCDA functions

Supports PCDA functions.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
Off	0	Off
On	1	On

5.2.7.10 `LimitSwitchOption`: Limit switch configuration

Limit switch configuration.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
None	0	None
Low	1	Low
High	2	High

5.2.8 Category External: External systems

Coordinating with external systems.

5.2.8.1 `DCStag`: Corresponding DCS block (instructor station)

Corresponding DCS block (instructor station)

Item Type	Variable Type	Units Category
parameter	string	

5.2.9 Category Calculation: Calculation details

Detailed working variables for the controller

5.2.9.1 `DcsOutput`: Dcs controller output

Dcs controller output

Item Type	Variable Type	Units Category
input	double	control_signal

5.2.9.2 `PreviousMode`: Auto/Manual mode used in the previous execution

Auto/Manual mode previous execution.

Item Type	Variable Type	Units Category
output	integer	

Enumerated Options

Option	Numerical Value	Description
Auto	0	Auto
Manual	1	Manual
Computer	2	Computer

5.2.9.3 `Integral`: The time integral of the normalized deviation

The time integral of the normalized deviation.

Item Type	Variable Type	Units Category
state	double	time

5.2.9.4 `PreviousIntegralTime`: Integral time from the previous execution

Integral time from the previous execution.

Item Type	Variable Type	Units Category
state	double	time

5.2.9.5 `PreviousGain`: Effective gain from the previous execution

Effective gain from the previous execution.

Item Type	Variable Type	Units Category
state	double	

5.2.9.6 `PreviousSetpoint`: Setpoint from the previous execution

Setpoint from the previous execution.

Item Type	Variable Type	Units Category
state	double	The same unit as a data item Measurement

5.2.9.7 `FilteredDterm`: Filtered derivative term

Filtered derivative term.

Item Type	Variable Type	Units Category
state	double	

5.2.10 Category Alarms: Alarms and warnings

Alarms and warnings of controller status.

5.2.10.1 `TripSignal`: Trip signal (true if tripped)

Trip signal (true if tripped).

Item Type	Variable Type	Units Category
input	Boolean	

5.2.10.2 `PermissiveSignal`: Permissive signal (false if tripped)

Permissive signal (false if tripped).

Item Type	Variable Type	Units Category
input	Boolean	

5.2.10.3 `BlockTripSignals`: Block trip signals

When true the trip signals are blocked.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.10.4 `SwitchToAuto`: Switch controller to auto mode

Switch controller to auto mode.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.10.5 `ExternalFailure`: External failure

External failure.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.10.6 `SuppressionIsOn`: When true the alarms are suppressed

When true the alarms are suppressed.

Item Type	Variable Type	Units Category
input	Boolean	

5.2.10.7 `TripOptions`: Options for trip

If set to "Lock" the controller will switch to manual when a trip signal is received. If set to "Force" the controller will remain in AUTO so when the trip signal is reset the controller resumes control. In either case, the controller output position at trip is governed by TripPositionOption. Note, however, that if set to "Lock", the ManualOutput will take presendence if this is set up to be used.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
Lock	0	Lock
Force	1	Force

5.2.10.8 `TripPositionOption`: Output position action when tripped

If set to "Safe position" the controller output will go to SafePosition when a trip signal is received. If set to "Freeze" the controller output will stay constant when a trip signal is received.

Item Type	Variable Type	Units Category
parameter	integer	

Enumerated Options

Option	Numerical Value	Description
Safe position	0	Safe position
Freeze	1	Freeze

5.2.10.9 `SafePosition`: Safe controller output position at trip

If TripPositionOption is set to "Safe position" the controller output turns to this position when trip is detected.

Item Type	Variable Type	Units Category
parameter	double	control_signal

5.2.10.10 `AlarmLimit`: Alarm limit for the normalized deviation

Alarm limit for the normalized deviation.

Item Type	Variable Type	Units Category
parameter	double	fraction

5.2.10.11 `AlarmDelay`: Delay of alarms and warnings

Delay of alarms and warnings.

Item Type	Variable Type	Units Category
parameter	double	time

5.2.10.12 `FailTimer`: Timer for failure alarm annunciation

Timer for failure alarm annunciation.

Item Type	Variable Type	Units Category
state	double	time

5.2.10.13 `DeviationTimer`: Timer for deviation warning annunciation

Timer for deviation warning annunciation.

Item Type	Variable Type	Units Category
state	double	time

5.2.10.14 `SuppressAlarms`: Suppress all alarms and warnings (from OS)

Suppress all alarms and warnings (from OS).

Item Type	Variable Type	Units Category
state	Boolean	

5.2.10.15 `AcknowledgeAll`: Acknowledge all alarms and warnings

Acknowledge all alarms and warnings.

Item Type	Variable Type	Units Category
working	Boolean	

5.2.10.16 `AcknowledgeFailure`: Acknowledge failure alarm

Acknowledge failure alarm.

Item Type	Variable Type	Units Category
working	Boolean	

5.2.10.17 `AcknowledgeDeviation`: Acknowledge deviation warning

Acknowledge deviation warning.

Item Type	Variable Type	Units Category
working	Boolean	

5.2.11 Category Realtime_NotInUse: Real-time parameters not in use

These parameters are used only if the controller is used in a real-time simulator. Not in use yet.

5.2.11.1 `UseSpecialRTparameters`: Use special tuning parameters in RT mode, not in use.

When true the controller will use different tuning parameters depending on whether the model runs real time (RT), lookahead (LA) or planning (PL) mode. Not in use.

Item Type	Variable Type	Units Category
parameter	Boolean	

5.2.11.2 `IntegralTimeRT`: Integral time in RT mode, not in use

Integral time in RT mode, not in use.

Item Type	Variable Type	Units Category
parameter	double	time

5.2.11.3 `GainRT`: Gain in RT mode, not in use

Gain in RT mode, not in use

Item Type	Variable Type	Units Category
parameter	double	

5.2.11.4 `IntegralTimePL`: Integral time in PL mode, not in use

Integral time in PL mode, not in use.

Item Type	Variable Type	Units Category
parameter	double	time

5.2.11.5 `GainPL`: Gain in PL mode, not in use

Gain PL in mode, not in use.

Item Type	Variable Type	Units Category
parameter	double	

5.2.11.6 `IntegralTimeLA`: Integral time in LA mode, not in use

Integral time in LA mode, not in use.

Item Type	Variable Type	Units Category
parameter	double	time

5.2.11.7 `GainLA`: Gain in LA mode, not in use

Gain in LA mode, not in use.

Item Type	Variable Type	Units Category
parameter	double	

5.2.11.8 `ParameterDisplaySwitch`: Switch for display of tuning parameters in the custom panel, not in use

Switch for display of tuning parameters in the custom panel, not in use.

Item Type	Variable Type	Units Category
state	integer	

Enumerated Options

Option	Numerical Value	Description
INPUTS	0	INPUTS
PL	1	PL
RT_AND_LA	2	RT_AND_LA
RT	3	RT