GE Oil & Gas

# Masoneilan* SVI* FF Function Blocks

## Instruction Manual (Rev B)


imagination at work

## About this Guide

This instruction manual applies to the following instruments and approved software:

- SVI FF
  - with firmware version 1.0.0.1 or higher
  - with ValVue* version 3.0
  - with handheld communicator with DD published for SVI FF

The information in this manual is subject to change without prior notice.

The information contained in this manual, in whole or part, shall not be transcribed or copied without GE Oil & Gas' written permission.

In no case does this manual guarantee the merchantability of the positioner or the software or its adaptability to a specific client needs.

Please report any errors or questions about the information in this manual to your local supplier or visit www.geoilandgas.com/valves.

## DISCLAIMER

**THESE INSTRUCTIONS PROVIDE THE CUSTOMER/OPERATOR WITH IMPORTANT PROJECT-SPECIFIC REFERENCE INFORMATION IN ADDITION TO THE CUSTOMER/OPERATOR'S NORMAL OPERATION AND MAINTENANCE PROCEDURES. SINCE OPERATION AND MAINTENANCE PHILOSOPHIES VARY, GE (GENERAL ELECTRIC COMPANY AND ITS SUBSIDIARIES AND AFFILIATES) DOES NOT ATTEMPT TO DICTATE SPECIFIC PROCEDURES, BUT TO PROVIDE BASIC LIMITATIONS AND REQUIREMENTS CREATED BY THE TYPE OF EQUIPMENT PROVIDED.**

**THESE INSTRUCTIONS ASSUME THAT OPERATORS ALREADY HAVE A GENERAL UNDERSTANDING OF THE REQUIREMENTS FOR SAFE OPERATION OF MECHANICAL AND ELECTRICAL EQUIPMENT IN POTENTIALLY HAZARDOUS ENVIRONMENTS. THEREFORE, THESE INSTRUCTIONS SHOULD BE INTERPRETED AND APPLIED IN CONJUNCTION WITH THE SAFETY RULES AND REGULATIONS APPLICABLE AT THE SITE AND THE PARTICULAR REQUIREMENTS FOR OPERATION OF OTHER EQUIPMENT AT THE SITE.**

**THESE INSTRUCTIONS DO NOT PURPORT TO COVER ALL DETAILS OR VARIATIONS IN EQUIPMENT NOR TO PROVIDE FOR EVERY POSSIBLE CONTINGENCY TO BE MET IN CONNECTION WITH INSTALLATION, OPERATION OR MAINTENANCE. SHOULD FURTHER INFORMATION BE DESIRED OR SHOULD PARTICULAR PROBLEMS ARISE WHICH ARE NOT COVERED SUFFICIENTLY FOR THE CUSTOMER/OPERATOR'S PURPOSES THE MATTER SHOULD BE REFERRED TO GE.**

**THE RIGHTS, OBLIGATIONS AND LIABILITIES OF GE AND THE CUSTOMER/OPERATOR ARE STRICTLY LIMITED TO THOSE EXPRESSLY PROVIDED IN THE CONTRACT RELATING TO THE SUPPLY OF THE EQUIPMENT. NO ADDITIONAL REPRESENTATIONS OR WARRANTIES BY GE REGARDING THE EQUIPMENT OR ITS USE ARE GIVEN OR IMPLIED BY THE ISSUE OF THESE INSTRUCTIONS.**

**THESE INSTRUCTIONS CONTAIN PROPRIETARY INFORMATION OF GE, AND ARE FURNISHED TO THE CUSTOMER/OPERATOR SOLELY TO ASSIST IN THE INSTALLATION, TESTING, OPERATION, AND/OR MAINTENANCE OF THE EQUIPMENT DESCRIBED. THIS DOCUMENT SHALL NOT BE REPRODUCED IN WHOLE OR IN PART NOR SHALL ITS CONTENTS BE DISCLOSED TO ANY THIRD PARTY WITHOUT THE WRITTEN APPROVAL OF GE.**

## Copyright

# Contents

## Document Changes

| Version/Date | Changes |
|---|---|
| A/05-2014 | Original release |
| B/5-2014 | Performed formatting changes. |

# 1. Introduction and Safety Information

## Introduction

This manual explains the functioning of the FOUNDATION Fieldbus blocks used in the SVI FF firmware. The basis for these block descriptions are extracted for the FOUNDATION Fieldbus document: FOUNDATION™ Specification Function Block Application Process Part 1 (FF-891), Part 2 (FF-892) and Part 4 (FF-893). These documents were then edited and amended to describe GE's implementation.

It is the customer's responsibility to know how to use their Host DCS with GE function blocks. GE does not provide specific instructions for each DCS and each function block (used individually or in combination).

## Safety Information

This section provides safety information including safety symbols that are used on the SVI FF and the safety symbol definition.

**CAUTION**

*Read this entire section before installation and operation.*

## Safety Symbols

SVI FF instructions contain **WARNINGS**, **CAUTIONS** labels and **Notes**, where necessary, to alert you to safety related or other important information. Total compliance with all **WARNING**, and **CAUTION** notices is required for safe operation.

**WARNING**

*Indicates a potentially hazardous situation, which if not avoided could result in serious injury.*

**CAUTION**

*Indicates a potentially hazardous situation, which if not avoided could result in property or data damage.*

**NOTE**

*Indicates important facts and conditions.*

## SVI FF Product Safety

The SVI FF digital valve positioner is intended for use with industrial compressed air or, natural gas systems only.

**NOTE**

*Installations using natural gas are Zone 0 or Div 1 installations.*

Ensure that an adequate pressure relief provision is installed when the application of system supply pressure could cause peripheral equipment to malfunction. Installation must be in accordance with local and national compressed air and instrumentation codes.

*General installation, maintenance or replacement*

- Products must be installed in compliance with all local and national codes and standards by qualified personnel using safe site work practices. Personal Protective Equipment (PPE) must be used per safe site work practices.
- Ensure proper use of fall protection when working at heights, per safe site work practices. Use appropriate safety equipment and practices to prevent the dropping of tools or equipment during installation.
- Under normal operation, compressed supply gas is vented from the SVI FF to the surrounding area, and may require additional precautions or specialized installations.

*Intrinsically Safe Installation*

Products certified as explosion proof or flame proof equipment or for use in intrinsically safe installations *MUST BE*:

- Installed, put into service, used and maintained in compliance with national and local regulations and in accordance with the recommendations contained in the relevant standards concerning potentially explosive atmospheres.
- Used only in situations that comply with the certification conditions shown in this document and after verification of their compatibility with the zone of intended use and the permitted maximum ambient temperature.
- Installed, put into service and maintained by qualified and competent professionals who have undergone suitable training for instrumentation used in areas with potentially explosive atmospheres.

**WARNING**

*Before using these products with fluids/compressed gases other than air or for non-industrial applications, consult the factory. This product is not intended for use in life support systems.*

*Under certain operating conditions, the use of damaged instruments could cause a degradation of the performance of the system, which can lead to personal injury or death.*

*Under certain operating conditions the SVI FF High Flow unit can produce noise levels greater than 85 dBA. Perform proper site monitoring and testing to verify the need for engineering or administrative controls to eliminate or reduce hazardous noise levels.*

*Installation in poorly ventilated confined areas, with any potential of gases other than oxygen being present, can lead to a risk of personnel asphyxiation.*

Use only genuine replacement parts which are provided by the manufacturer, to guarantee that the products comply with the essential safety requirements of the European Directives.

Changes to specifications, structure, and components used may not lead to the revision of this manual unless such changes affect the function and performance of the product.

*This page intentionally left blank.*

# 2. AO Function Block

## General

The AO function block receives the control signal from the transducer block and outputs it to the actuator. The major functions of the AO function block include (Figure 1):

- Scaling
- Setpoint limiters - for both the value and rate of change
- Simulation
- Valve position feedback
- Actions upon abnormality of upstream block
- Signal inversion

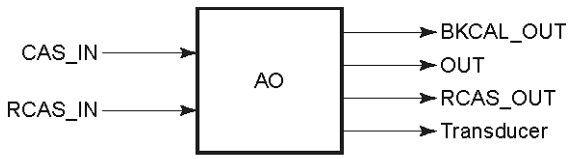


**Figure 1 Inputs/Outputs of AO Function Block**

The AO function block performs bi-directional signal handling (Figure 2): transfer of the valve control signal to the transducer block (forward path) and feedback of the valve position signal from the transducer block to the upstream block (backward path).

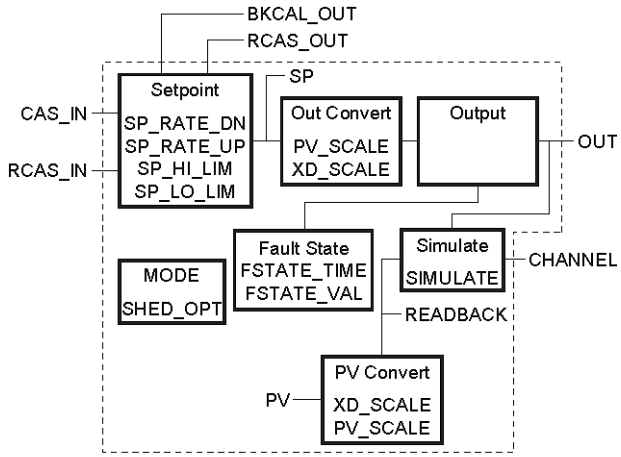


**Figure 2 Function Diagram of AO Function Block**

## Modes

The target mode for the AO function block can be set from five block modes: RCas, Cas, Auto, Man, and O/S. Regardless of the target mode, the AO block automatically enters the IMan or LO mode when a specified condition is met (such as when another function block enters a specific status) depending on the parameter settings.

## Forward Path

The following describes the signal input from the upstream block to the AO block and then passed to the transducer block. The upstream block is typically the PID controller block, and the control signal from the PID block is input as the source of computing the setpoint SP for the AO block.

The path for computing the SP differs depending on the mode: In Cas mode, CAS_IN is used for SP. In RCas mode, RCAS_IN is used for SP. If the value of CAS_IN or RCAS_IN, whichever is used, is greater than SP_HI_LIM (high limit) or less than SP_LO_LIM (low limit), the internal SP is set to the respective limits. Also, if the rate of change in the value of CAS_IN or RCAS_IN, whichever is used, is greater than SP_RATE_UP (rate-of increase limit) in the increasing direction, or than SP_RATE_DN (rate-of-decrease limit) in the decreasing direction, the change in internal SP is limited by the corresponding rate-of-change limit setting.

In RCas, Cas or Auto mode, the SP value is used for the AO block's output OUT, whose value is then passed to the transducer block via channel 1.

### Fault State

As for Fieldbus-enabled positioners including the SVI FF, not only a power failure but also other errors (such as a communication error) can cause the fail-safe action. For example, when the status of the CAS_IN input of the AO block from its upstream block indicates a specific status, such as a communication error, the case is regarded as an abnormality and fault state actions including a mode change are enacted.

When any of the following status exists for the moment of time specified in FSTATE_TIME, the block goes to the fault state and the mode changes to LO mode:

- Target mode is Cas, and the status of CAS_IN is *Bad: No Comm*
- Target mode is Cas, and the status of CAS_IN is *Good: IFS*
- Target mode is RCas, and the status of RCAS_IN is *Good: IFS*

In LO mode, the block holds the output (OUT) or outputs FSTATE_VAL, according to the setting of IO_OPTS. The factory setting is to hold the output.

## Backward Path

The backward path functions as:

1. The valve position signal from the transducer block is written to the AO block parameter READBACK.
2. Is scaled based on XD_SCALE and PV_SCALE for conversion to the process variable PV.
3. The value of PV is fed back to the PID block or an upper-level system as the valve position signal via the parameter BKCAL_OUT and RCAS_OUT.

If SIMULATE is set to *Enable*, the value of SIMULATE.Simulate_Value is always set in READBACK.

SIMULATE contains the following data:

| | |
|---|---|
| *Simulate Status* | Status to be set in simulation mode. |
| *Simulate Value* | Value to be set in simulation mode. |
| *Transducer Status* | Status of input from transducer. |
| *Transducer Value* | Value of input from transducer. |
| *Enable/Disable* | Whether to enable *2* or disable *1* simulation. |

## IO_OPTS and STATUS_OPTS

These parameters stipulate options about the block's signal processing and mode transitions. The settings of these options are made by setting or resetting the respective bits: on = true, off = false. Table 1 shows the options available in AO block IO_OPTS.

**Table 1 IO_OPTS of AO Block**

| Bit # | Meaning | Description |
|---|---|---|
| 1 | SP tracks PV if Man | Equalizes SP to PV when target is MAN mode. |
| 3 | SP tracks PV if LO | Equalizes SP to PV in LO mode. |
| 4 | SP tracks RCas or Cas if LO or Man (SP track retained target) | In LO mode, Equalizes SP to RCAS_IN if target mode is RCas and to CAS_IN if target mode is Cas. |
| 5 | Increase to close | Inverts the signal while it goes from SP through OUT. |
| 6 | Faultstate Type (Faultstate to value) | Uses a FSTATE_VALUE in LO mode. |
| 7 | Faultstate Type (Use Faultstate value on restart) | Uses a value preset for fault state also at a restart. |
| 8 | Target to Man | Sets the target mode to Man upon activation of the fault state. |
| 9 | PV for BKCAL_OUT | Sets the value of PV in BKCAL_OUT and RCAS_OUT. |

Only the Propagate Fault Backward option is available in AO block STATUS_OPTS.

**Table 2 STATUS_OPTS of AO Block**

| Bit # | Meaning | Description |
|---|---|---|
| 4 | Propagate Fault Backward | Stipulates the handling of the value, data status and related alarm of BKCAL_OUT and RCAS_OUT to be performed.<br>If this option is true, then:<br>□ Set the quality and sub-status components of the status of BKCAL_OUT to Bad and sensor failure, respectively.<br>□ Do nothing for the BKCAL_OUT value.<br>If this option is false, then:<br>□ Set the quality and sub-status components of the status of BKCAL_OUT to Bad and non specific, respectively.<br>□ Generates a block alarm. |

## Mode Shedding on Computer Failure

When the data status of RCAS_IN falls to Bad while the block is running in RCas (remote cascade) mode, mode shedding occurs in accordance with the setting in SHED_OPT. Table 3 shows the available selections for the AO block SHED_OPT setting.

**Table 3 SHED_OPT of AO Block**

| Bit # | Available Setting for SHED_OPT | Actions upon Computer Failure |
|---|---|---|
| 1 | Normal shed, normal return | Sets MODE_BLK.actual to Cas(*1), and leaves MODE_BLK.target unchanged. |
| 2 | Normal shed, no return | Sets both MODE_BLK.actual and MODE_BLK.target to Cas(*1). |
| 3 | Shed to Auto, normal return | Sets MODE_BLK.actual to Auto(*2), and leaves MODE_BLK.target unchanged. |
| 4 | Shed to Auto, no return | Sets both MODE_BLK.actual and MODE_BLK.target to Auto(*2). |
| 5 | Shed to Manual, normal return | Sets MODE_BLK.actual to Man, and leaves MODE_BLK.target unchanged. |
| 6 | Shed to Manual, no return | Sets both MODE_BLK.actual and MODE_BLK.target to Man. |
| 7 | Shed to retained target, normal return | If Cas is set in MODE_BLK.target,<br>□ sets MODE_BLK.actual to Cas<br>and<br>□ leaves MODE_BLK.target unchanged.<br>If Cas is not set in MODE_BLK.target,<br>□ sets MODE_BLK.actual to Auto(*2) and<br>□ leaves MODE_BLK.target unchanged. |
| 8 | Shed to retained target, No return | If Cas is set in MODE_BLK.target, sets:<br>□ MODE_BLK.actual to Cas, and<br>□ MODE_BLK.target to Cas, too.<br>If Cas is not set in MODE_BLK.target, sets:<br>□ MODE_BLK.actual to Auto(*2),<br>and<br>□ MODE_BLK.target to Cas. |

(*1) The modes to which the AO block can transfer are limited to those set in MODE_BLK.permitted, and the priority levels of modes (Figure 3). In fact, if Normal shed, normal return is set for SHED_OPT, the detection of a computer failure causes MODE_BLK.actual to change to Cas, Auto, or Man, whichever is set in MODE_BLK.permitted and has the lowest priority level.

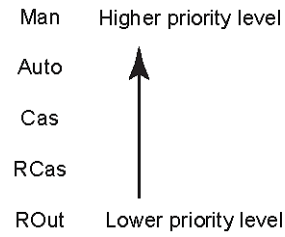


**Figure 3 Mode Priority Levels**

(*2) Only when Auto is set as permitted mode.

**NOTE**

*If a control block is connected as a cascade primary block of the AO block, a mode transition of the AO block to Cas occurs in the following sequence due to initialization of the cascade connection: RCas —> Auto —> Cas.*

## Initialization at Start

To prevent a sudden change in output when the AO block carries out the specified actions for the first time after the power is turned on, it:

1. Equalizes SP to PV if the Faultstate Type option (bit no. 7) in IO_OPTS is false.
2. Equalizes OUT to READBACK.

If the Faultstate Type option (bit no. 7) in IO_OPTS is true, it restores FSTATE_VAL in SP.

## Alarm Processing

When a condition shown in Table 4 is met, the AO block changes the bit statuses of BLOCK_ERROR accordingly and generates a block alarm.

**Table 4 BLOCK_ERROR in AO Block**

| Bit # | Name of Error Represented | Condition |
|---|---|---|
| 3 | Simulate Active | SIMULATE is active. |
| 4 | Local Override | Fault state is on, and Propagate Fault Backward is false. |
| 7 | Input Failure / process variable has BAD status | Propagate Fault Backward in STATUS_OPTS is false, and the sub-status component of the status of READBACK is sensor failure or device failure. |
| 15 | Out-of-Service | The target mode is O/S. |

## BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

## Channel Parameter Applications

**Table 5 Channels for Analog Output Blocks**

| Channel Value | Usage | Behavior |
|---|---|---|
| 0 | Not Configured | Prevents the Analog Output block from going into Auto Mode. |
| 1 | Position | Update the FINAL_VALUE parameter in the Transducer block. Readback is provided from FINAL_POSITION_VALUE.<br>Blocks the switching of the AO block to Auto mode if the TB.SETPOINT_SOURCE is not equal to 1 (Analog Output block).<br>Reports an error for the setting mismatch. |

## Minimal Configuration

XD_SCALE.UNIT = %

CHANNEL = 1 (POSITION CONTROL)

# AO Block Access

**Table 6 AO Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 16 | READBACK | 5 | | 5 | |
| 2 | TAG_DESC | | | | | 17 | CAS_IN | 5 | | 5 | |
| 3 | STRATEGY | | | | 2 | 18 | SP_RATE_DN | | | | 4 |
| 4 | ALERT_KEY | | | | 1 | 19 | SP_RATE_UP | | | | 4 |
| 5 | MODE_BLK | 4 | | 4 | | 20 | SP_HI_LIM | | 4 | | |
| 6 | BLOCK_ERR | 2 | | 2 | | 21 | SP_LO_LIM | | 4 | | |
| 7 | PV | 5 | | 5 | | 22 | CHANNEL | | | | 2 |
| 8 | SP | 5 | | 5 | | 23 | FSTATE_TIME | | | | 4 |
| 9 | OUT | 5 | | 5 | | 24 | FSTATE_VAL | | | | 4 |
| 10 | SIMULATE | | | | | 25 | BKCAL_OUT | | | 5 | |
| 11 | PV_SCALE | | 11 | | | 26 | RCAS_IN | | | 5 | |
| 12 | XD_SCALE | | 11 | | | 27 | SHED_OPT | | | | 1 |
| 13 | GRANT_DENY | | 2 | | | 28 | RCAS_OUT | | | 5 | |
| 14 | INPUT_OPTS | | | | 2 | 29 | UPDATE_EVT | | | | |
| 15 | STATUS_OPTS | | | | | 30 | BLOCK_ALM | | | | |
| Sub totals | | 23 | 26 | 23 | 9 | 31 | BLOCK_ERR_ DESC_1 | | | | |
| | | | | | | Sub totals | | 23 | 26 | 48 | 7 |
| | | | | | | Totals | | 23 | 26 | 48 | 68 |

# 3. PID Function Block

## General

The PID function block receives an input signal (Figure 4), performs PID control computation, and outputs the control signal, like a single-loop controller.

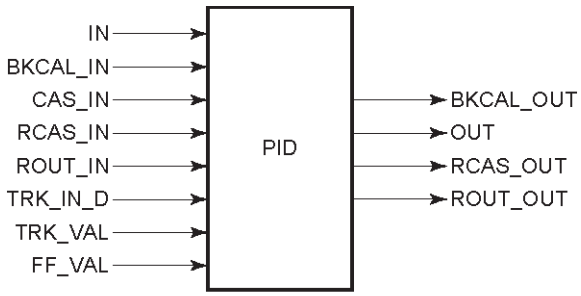


**Figure 4 Inputs/Outputs of PID Function Block**

In practice, it performs PID computation based on the setpoint (SP) set in the actual mode and the process value (PV), and generates a value of its output OUT so as to decrease the deviation. The PID block works with other function blocks such as the AI and AO blocks connected to it. The major functions of the PID block include (Figure 5):

- Filtering
- Setpoint limiters - both for the value and rate of change
- Scaling of process variable (PV), setpoint (SP), and output (OUT)
- PID control computation
- Control action bypass
- Feed-forward
- External-output tracking
- Measured-value tracking
- Output limiters
- Mode shedding upon computer failure
- Alarm generation

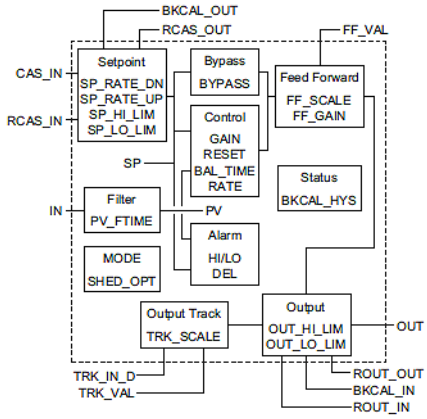


**Figure 5 Function Diagram of PID Function Block**

# Modes

The target mode for the PID function block can be set from five block modes: ROut, RCas, Cas, Auto, Man, and O/S. Regardless of the target mode, the PID block automatically enters the IMan or LO mode when a specified condition is met (such as when another function block enters a specific status), depending on the parameter settings.

## Input Processing

The input signal to IN is filtered through a lag filter whose time constant is set in PV_FTIME, and then set as the process variable (PV).

## Setpoint (SP) Limiters

The path for selecting the SP differs depending on the mode. In Cas mode, CAS_IN input is used for SP. In RCas mode, RCAS_IN input is used for SP. If the value of CAS_IN or RCAS_IN, whichever is used, is greater than SP_HI_LIM (high limit) or less than SP_LO_LIM (low limit), the internal SP is set to the respective limits. When the target mode is Auto or Man, and when SP-PV tracking is not specified at the same time, the rate of change in the setpoint is also limited (by the values of SP_RATE_UP and SP_RATE_DN).

## PID Computation

You can select from one of five PID computational models by setting the ALGORITM parameter. Figure 6 through Figure 10 visually describe each of the five user-selectable PID computational models.

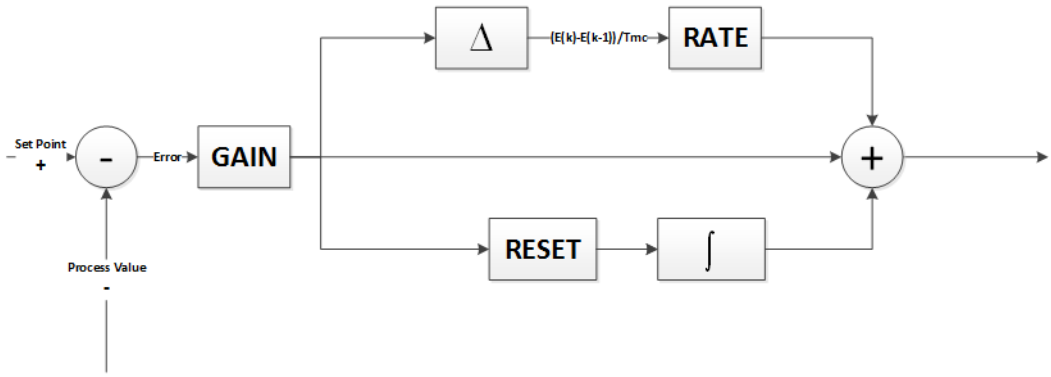


**Figure 6 Ideal Algorithm**

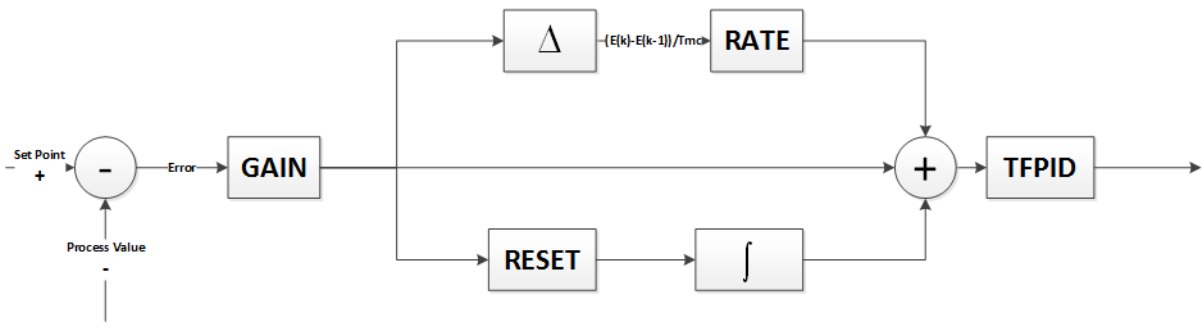


**Figure 7 Serial Algorithm**

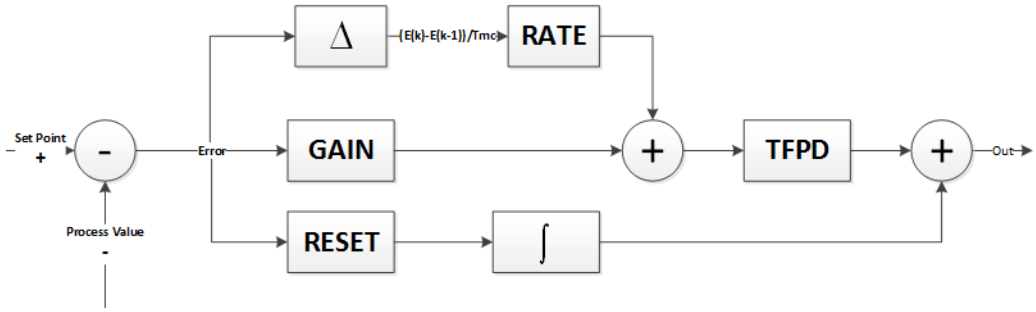


Figure 8 Parallel Algorithm

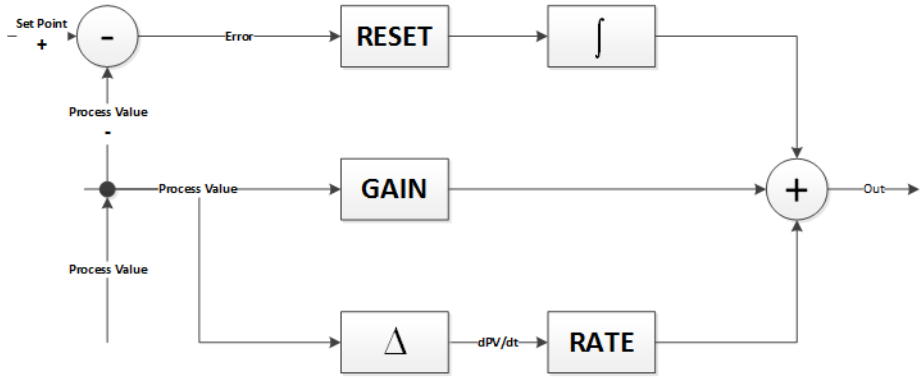


Figure 9 I-PD Algorithm

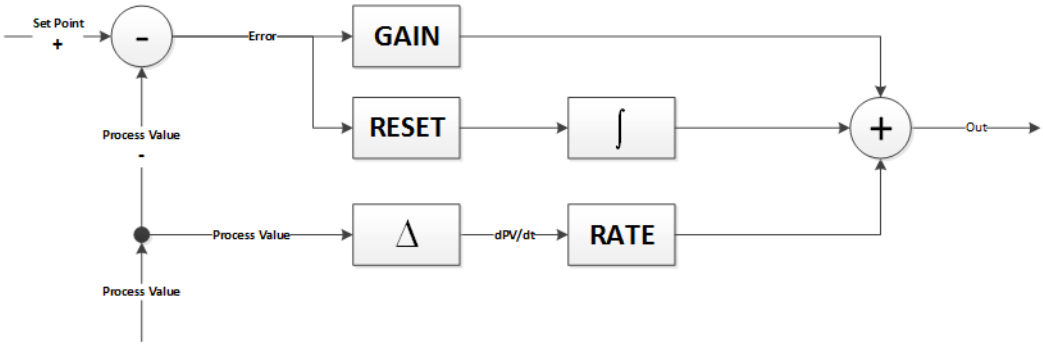


Figure 10 IP-D Algorithm

Table 7 shows the PID control parameters.

**Table 7 PID Control Parameters**

| Parameter | Description | Valid Range |
|---|---|---|
| GAIN | Proportional gain | 0.05 to 20 |
| RESET | Integral time | 4x macrocycle to 10,000 (seconds) |
| RATE | Derivative time | 4x macrocycle to infinity |
| ALGORITHM | Selects PID algorithm | See "PID Computation" on page 21. |
| TFPID | Filter time for PID algorithm | 4x macrocycle to infinity |
| TFPD | Filter time for proportional derivative portion | 4x macrocycle to infinity |

## Direction of Control Action

The direction of the control action is determined by the Direct Acting setting in CONTROL_OPTS (Table 8).

**Table 8 PID Control Parameters**

| Value of Direct Acting | Resulting Action |
|---|---|
| True | The output increases when the input PV is greater than the setpoint SP. |
| False | The output decreases when the input PV is greater than the setpoint SP. |

## Control Action Bypass

The PID control computation can be bypassed to set the SP value in the control output OUT (Figure 11). Setting BYPASS to *ON* bypasses the PID control computation.

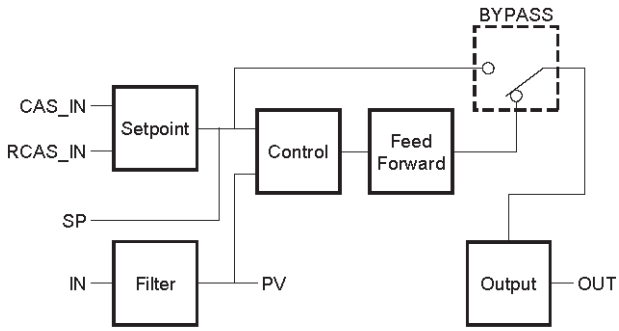


**Figure 11 Control Action Bypass**

## Feed-forward

Feed-forward adds a compensation input signal FF_VAL to the output of the PID control computation and is typically used for feed-forward control. In practice, the value of the change in FF_VAL is scaled to the range of the OUT, multiplied by the value of FF_GAIN, and then added to the PID control computation result, as illustrated by Figure 12.

When the status of FF_VAL is Bad, the value of LUV (Last usable value) is used instead of FF_VAL. If LUV contains no value, the feed-forward action is not carried out.

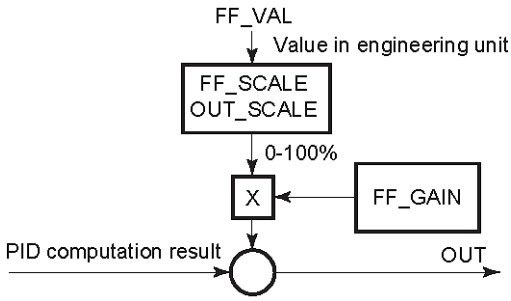


**Figure 12 Feed-forward**

## External-output Tracking (LO)

External-output tracking outputs the value of the remote output TRK_VAL set from outside the PID block (Figure 13). External tracking is performed when the block mode is LO.

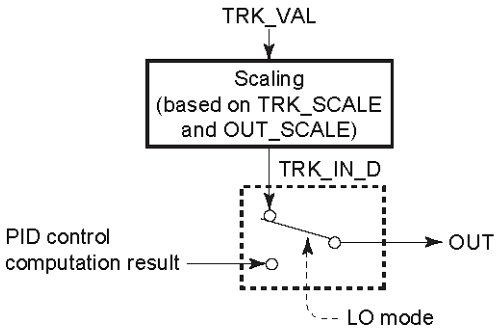


**Figure 13 External-value Tracking**

To change the block mode to LO:

1. Set Track Enable in CONTROL_OPTS (see "CONTROL_OPTS" on page 25) to true.
2. Set TRK_IN_D to true.

However, to change the block mode from Man to LO, Track in Manual must also be set as true in CONTROL_OPTS.

## Measured-value Tracking

Measured-value tracking, also referred to as SP-PV tracking, is the action of equalizing the setpoint SP to the measured value PV when the block mode (MODE_BLK.actual) is Man in order to prevent a sudden change in control output from being caused by a mode change to Auto.

While a cascade primary control block is performing automatic control in Auto or Cas mode, when the mode of its secondary control block is changed from Cas to Auto, the cascade connection is opened and the control action of the primary block stops. The SP of the primary controller can also be equalized to its cascade input signal CAS_IN in this case.

The settings for measured-value tracking are made in the parameter CONTROL_OPTS (Table 9).

## CONTROL_OPTS

CONTROL_OPTS is a parameter that stipulates control options (Table 9).

**Table 9 CONTROL_OPTS of PID Block**

| Bit # | Options in CONTROL_OPTS | Description |
|---|---|---|
| 0 | Bypass Enable | Switch for activating the control action bypass |
| 1 | SP-PV Track in Man | Equalizes SP to PV when MODE_BLK.target is set to Man. |
| 2 | SP-PV Track in Rout | Equalizes SP to PV when MODE_BLK.target is set to ROut. |
| 3 | SP-PV Track in LO or IMan | Equalizes SP to PV when MODE_BLK.actual is set to LO or IMan. |
| 4 | SP Track retained Target | Equalizes SP to RCAS_IN or CAS_IN when MODE_BLK.target is either in IMan, LO, Man or ROut and MODE_BLK.actual is set to RCas or Cas. |
| 5 | Direct Acting | Set the PID block to be a direct acting controller. |
| 7 | Track Enable | While this option is set, if the value of TRK_IN_D becomes *1*, the mode transfers to LO. |
| 8 | Track in Manual | Set this option when the mode should be transferred to LO even when MODE_BLK.target is set to Man. This option is invalid when Track Enable option is not set. |
| 9 | Use PV for BKCAL_OUT | Sets the value of PV in BKCAL_OUT and RCAS_OUT, instead of the value of SP. |
| 12 | Obey SP limits if Cas or RCas | Puts the setpoint high/low limits in force in the Cas or RCas mode. |
| 13 | No OUT limits in Manual | Disables the high/low limits for OUT in the Man mode. |

## Initialization and Manual Fallback (IMan)

Initialization and manual fallback denotes a set of abnormality handling actions in which a PID block changes mode to IMan (initialization manual) and suspends the control action. Initialization and manual fallback takes place only when the following condition is met:

- The quality component of BKCAL_IN.status (data status of BKCAL_IN) is Bad.

OR

- The quality component of BKCAL_IN.status is Good (c)

  AND

- The sub-status component of BKCAL_IN.status is FSA, LO, NI, or IR.

### STATUS_OPTS

Table 10 shows the options in STATUS_OPTS.

**Table 10 STATUS_OPTS of PID Block**

| Bit # | Options in CONTROL_OPTS | Description |
|---|---|---|
| 0 | IFS if BAD IN | Sets the sub-status component of OUT.status to IFS if IN.status is Bad except when PID control bypass is on. |
| 1 | IFS if BAD CAS IN | Sets the sub-status component of OUT.status to IFS if CAS_IN.status is Bad. |
| 2 | Use Uncertain as Good | Does not regard IN as being in Bad status when IN.status is Uncertain (to prevent mode transitions from being affected when it is Uncertain). |
| 5 | Target to Manual if BAD IN | Automatically changes the value of MODE_BLK.target to Man when IN falls to Bad status. |
| 9 | Target to next permitted mode if BAD CAS IN | Automatically changes the value of MODE_BLK.target to Auto (or to Man if Auto is not set in Permitted) when CAS_IN falls to Bad status. |

## Auto Fallback

Auto fallback is when a PID block changes mode from Cas to Auto and continues automatic PID control with the user-set setpoint. To enable the auto fallback action to take place:

- The Target to next permitted mode if BAD CAS IN option must be preset to true in STATUS_OPTS.

AND

- Auto must be preset in MODE_BLK.permitted.

If the above settings are made, auto fallback takes place automatically when the following condition is met:

- CAS_IN.status (data status of cascade setpoint) is Bad except when the control action bypass is on.

## Mode Shedding on Computer Failure

Mode shedding occurs in accordance with the SHED_OPT setting when:

- (1) the data status of RCAS_IN, which is the setting received from a computer as the setpoint SP, falls to Bad while the PID block is running in the RCas (remote cascade) mode,

or

- (2) the data status of ROUT_IN, which is the setting received from a computer as the remote output signal, falls to Bad while the PID block is running in the ROut (remote output) mode

**Table 11 STATUS_OPTS of PID Block**

| Available Setting for SHED_OPT | Actions upon Computer Failure |
|---|---|
| Normal shed, normal return | Sets MODE_BLK.actual to Cas(*1 below the table), and leaves MODE_BLK.target unchanged. |
| Normal shed, no return | Sets both MODE_BLK.actual and MODE_BLK.target to Cas(*1 below the table). |
| Shed to Auto, normal return | Sets MODE_BLK.actual to Auto(*2 below the table), and leaves MODE_BLK.target unchanged. |
| Shed to Auto, no return | Sets both MODE_BLK.actual and MODE_BLK.target to Auto(*2 below the table). |
| Shed to Manual, normal return | Sets MODE_BLK.actual to Man, and leaves MODE_BLK.target unchanged. |
| Shed to Manual, no return | Sets both MODE_BLK.actual and MODE_BLK.target to Man. |

**Table 11 STATUS_OPTS of PID Block**

| Available Setting for SHED_OPT | Actions upon Computer Failure |
|---|---|
| Shed to retained target, normal return | If Cas is set in MODE_BLK.target, - sets MODE_BLK.actual to Cas(*1 below the table) and<br>□ leaves MODE_BLK.target unchanged.<br>If Cas is not set in MODE_BLK.target,<br>□ sets MODE_BLK.actual to Auto(*2 below the table) and<br>□ leaves MODE_BLK.target unchanged. |
| Shed to retained target, no return | If Cas is set in MODE_BLK.target, sets:<br>□ MODE_BLK.actual to Cas, and<br>□ MODE_BLK.target to Cas(*1 below the table), too.<br>If Cas is not set in MODE_BLK.target, sets:<br>□ MODE_BLK.actual to Auto(*2 below the table), and<br>□ MODE_BLK.target to Cas. |

(*1) The modes to which the PID block can transfer are limited to those set in MODE_BLK.permitted, and the priority levels of modes (Figure 14). In fact, if Normal shed, normal return is set for SHED_OPT, detection of a computer failure causes MODE_BLK.actual to change to Cas, Auto, or Man, whichever is set in MODE_BLK.permitted and has the lowest priority level.

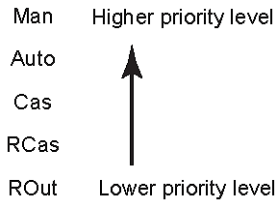


**Figure 14 Priority Levels**

(*2) Only when Auto is set as permitted mode.

**NOTE**

*If a control block is connected as a cascade primary block of the PID block in question, a mode transition of the PID block to Cas occurs in the following sequence due to initialization of the cascade connection: RCas or ROut —> Auto —> Cas.*

## Alarms

There are two kinds of alarms generated by a PID block: block and process alarms.

### Block Alarm (BLOCK_ALM)

The block alarm BLOCK_ALM is generated on occurrence of the errors in Table 12 (values set in BLOCK_ERR) and notifies the content of BLOCK_ERR.

**Table 12 Block Alarm (BLOCK_ALM)**

| Value of BLOCK_ERR | Condition |
|---|---|
| Input Failure | IN.status of the PID block is either of the following:<br>□ Bad-Device Failure<br>□ Bad-Sensor Failure |
| Local Override | MODE_BLK.actual of the PID block is LO. |
| Out of Service | MODE_BLK.target of the PID block is O/S. |

### Process Alarms

There are six types of process alarms (Table 13). Only one process alarm can be generated at a time, and the process alarm having the highest priority level from among those occurring at the same time is generated. The priority level is set for each process alarm type.

**Table 13 Process Alarms**

| Process Alarm | Cause of Occurrence | Parameter Containing Priority Level Setting |
|---|---|---|
| HI_HI_ALM | Occurs when the PV increases above the HI_HI_LIM value. | HI_HI_PRI |
| HI_ALM | Occurs when the PV increases above HI_LIM value. | HI_PRI |
| LO_ALM | Occurs when the PV decreases below the LO_LIM value. | LO_PRI |
| LO_LO_ALM | Occurs when the PV decreases below the LO_LO_LIM value. | LO_LO_LIM |
| DV_HI_ALM | Occurs when the value of [PV - SP] increases above the DV_HI_LIM value. | DV_HI_PRI |
| DV_LO_ALM | Occurs when the value of [PV - SP] decreases below the DV_LO_LIM value. | DV_LO_PRI |

## BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

# PID Block Access

**Table 14 PID Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 34 | SHED_OPT | | | | 1 |
| 2 | TAG_DESC | | | | | 35 | RCAS_OUT | | | 5 | |
| 3 | STRATEGY | | | | 2 | 36 | ROUT_OUT | | | 5 | |
| 4 | ALERT_KEY | | | | 1 | 37 | TRK_SCALE | | | | 11 |
| 5 | MODE_BLK | 4 | | 4 | | 38 | TRK_IN_D | 2 | | 2 | |
| 6 | BLOCK_ERR | 2 | | 2 | | 39 | TRK_VAL | 5 | | 5 | |
| 7 | PV | 5 | | 5 | | 40 | FF_VAL | | | 5 | |
| 8 | SP | 5 | | 5 | | 41 | FF_SCALE | | | | 11 |
| 9 | OUT | 5 | | 5 | | 42 | FF_GAIN | | | | 4 |
| 10 | PV_SCALE | | 11 | | 2 | 43 | UPDATE_EVT | | | | |
| 11 | OUT_SCALE | 5 | 11 | 5 | | 44 | BLOCK_ALM | | | | |
| 12 | GRANT_DENY | | 2 | | | 45 | ALARM_SUM | 8 | | 8 | |
| 13 | CONTROL_ OPTS | | | | 2 | 46 | ACK_OPTION | | | | 2 |
| 14 | STATUS_OPTS | | | | 2 | 47 | ALARM_HYS | | | | 4 |
| 15 | IN | | | 5 | | 48 | HI_HI_PRI | | | | 1 |
| 16 | PV_FTIME | | | | 4 | 49 | HI_HI_LIM | | | | 4 |
| 17 | BYPASS | | 1 | | | 50 | HI_PRI | | | | 1 |
| 18 | CAS_IN | 5 | | 5 | | 51 | HI_LIM | | | | 4 |
| 19 | SP_RATE_DN | | | | 4 | 52 | LO_PRI | | | | 1 |
| 20 | SP_RATE_UP | | | | 4 | 53 | LO_LIM | | | | 4 |
| 21 | SP_HI_LIM | | 4 | | | 54 | LO_LO_PRI | | | | 1 |
| 22 | SP_LO_LIM | | 4 | | | 55 | LO_LO_LIM | | | | 4 |
| 23 | GAIN | | | | 4 | 56 | DV_HI_PRI | | | | 1 |
| 24 | RESET | | | | 4 | 57 | DV_HI_LIM | | | | 4 |

**Table 14 PID Block Access (Continued)**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | BAL_TIME | | | | 4 | 58 | DV_LO_PRI | | | | 1 |
| 26 | RATE | | | | 4 | 59 | DV_LO_LIM | | | | 4 |
| 27 | BKCAL_IN | | | 5 | | 60 | HI_HI_ALM | | | | |
| 28 | OUT_HI_LIM | | 4 | | | 61 | HI_ALM | | | | |
| 29 | OUT_LO_LIM | | 4 | | | 62 | LO_ALM | | | | |
| 30 | BKCAL_HYS | | | | 4 | 63 | LO_LO_ALM | | | | |
| 31 | BKCAL_OUT | | | 5 | | 64 | DV_HI_ALM | | | | |
| 32 | RCAS_IN | | | 5 | | 65 | DV_LO_ALM | | | | |
| 33 | ROUT_IN | | | 5 | | 66 | BLOCK_ERR_ DESC_1 | | | | |
| Sub totals | | 28 | 43 | 53 | 41 | 67 | ALGORITHM | | | | |
| | | | | | | 68 | TFPD | | | | |
| | | | | | | 69 | TFPID | | | | |
| | | | | | | Sub totals | | 15 | 0 | 30 | 63 |
| | | | | | | Totals | | 43 | 43 | 83 | 104 |

*This page intentionally left blank.*

# 4. Analog Input

## Overview

The AI block takes the manufacturer's input data, selected by channel number, and makes it available to other function blocks at its output.

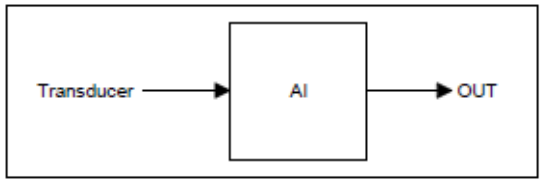


**Figure 15 Inputs/Outputs of Analog Input (AI) Function Block**

## Schematic

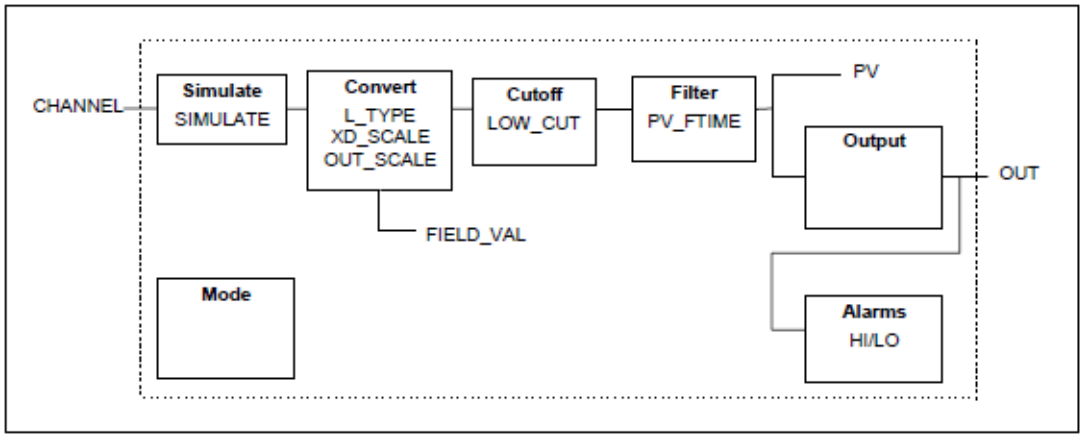


**Figure 16 Function Diagram of Analog Input Function Block**

# Description

Transducer scaling (XD_SCALE) is applied to the value from the channel to produce the FIELD_VAL in percent. The XD_SCALE units code must match the channel units code (if one exists), or the block will remain in O/S mode after being configured. A block alarm for units mismatch will be generated. The OUT_SCALE is normally the same as the transducer, but if L_TYPE is set to Indirect or Ind Sqr Root, OUT_SCALE determines the conversion from FIELD_VAL to the output. PV and OUT always have identical scaling. OUT_SCALE provides scaling for PV. The PV is always the value that the block places in OUT if the mode is Auto. If Man is allowed, someone may write a value to the output. The status prevents any attempt at closed loop control using the Man value, by setting the Limit value to Constant.

The LOW_CUT parameter has a corresponding *Low cutoff* option in the IO_OPTS bit string. If the option bit is true, any calculated output below the low cutoff value is changed to zero. This is only useful for zero based measurement devices, such as flow.

The PV filter, whose time constant is PV_FTIME, is applied to the PV, and not the FIELD_VAL. Equations:

*FIELD_VAL = 100*(channel value - EU@0%) / (EU@100% - EU@0%) [XD_SCALE] Direct: PV = channel value*

*Indirect: PV = (FIELD_VAL/100) * (EU@100% - EU@0%) + EU@0% [OUT_SCALE]*

*Ind Sqr Root: PV = sqrt(FIELD_VAL/100) * (EU@100% - EU@0%) + EU@0% [OUT_SCALE]*

**NOTE**

*Sqr Root is not recommended for SVI FF applications. This is reserved for future use.*

## Channel Parameter Applications

**Table 15 Channels for Analog Input Blocks**

| Channel Value | Usage | Behavior | XD_SCALE. UNIT |
|---|---|---|---|
| 0 | Not Configured | Prevents the Analog Input block from going into Auto Mode | |
| 7 | Supply Pressure | Provides the supply pressure value to the FF control | % |
| 8 | Actuator Pressure 1 | Provides the Actuator 1 pressure value to the FF control | PSI |
| 9 | Actuator Pressure 2 | Provides the Actuator 2 pressure value to the FF control | PSI |
| 10 | IP Current | Provides the IP current value to the FF control | mA |

**Table 15 Channels for Analog Input Blocks (Continued)**

| Channel Value | Usage | Behavior | XD_SCALE. UNIT |
|---|---|---|---|
| 11 | Pilot Pressure | Provides the pilot pressure value to the FF control | PSI |
| 12 | Temperature | Provides the board temperature to the FF control | °C |
| 13 | Final Value | Provides the Transducer block setpoint to the FF control | % |
| 14 | Final Position Value | Provides the Transducer block actual position to the FF control | % |
| 15 | Working SP | Provides the Transducer block working setpoint (after characterization) to the FF control | % |
| 16 | Working Position | Provides the Transducer block working position (before de-characterization) to the FF control | % |

## Supported Modes

O/S, Man, and Auto.

## Alarm Types

Standard block alarm plus standard HI_HI, HI, LO, and LO_LO alarms applied to OUT.

## Mode Handling

Standard transition in and out of O/S.

Standard transition from Man to Auto and back.

## Status Handling

The status values described in Output Parameter Formal Model of Part 1 of the FF891 specification apply, with the exception of the control sub-status values. The Uncertain - EU Range Violation status is always set if the OUT value exceeds the OUT_SCALE range, and no worse condition exists. The following options from STATUS_OPTS apply, where Limited refers to the sensor limits:

*Propagate Fault Forward*

*Uncertain if Limited*

*BAD if Limited*

*Uncertain if Man mode*

## Initialization

The PV filter must be initialized, but other than that, no special initialization is required. This is a pure calculation algorithm.

## BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

## Minimum Configuration

XD_SCALE.UNITS

OUT_SCALE.UNITS

CHANNEL

L-TYPE

# Analog Input Block Access

**Table 16 Analog Input Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 19 | FIELD_VAL | 5 | | 5 | |
| 2 | TAG_DESC | | | | | 20 | UPDATE_EVT | | | | |
| 3 | STRATEGY | | | | 2 | 21 | BLOCK_ALM | | | | |
| 4 | ALERT_KEY | | | | 1 | 22 | ALARM_SUM | 8 | | 8 | |
| 5 | MODE_BLK | 4 | | 4 | | 23 | ACK_OPTION | | | | 2 |
| 6 | BLOCK_ERR | 2 | | 2 | | 24 | ALARM_HYS | | | | 4 |
| 7 | PV | 5 | | 5 | | 25 | HI_HI_PRI | | | | 1 |
| 8 | OUT | 5 | | 5 | | 26 | HI_HI_LIM | | | | 4 |
| 9 | SIMULATE | | | | | 27 | HI_PRI | | | | 1 |
| 10 | XD_SCALE | | 11 | | | 28 | HI_LIM | | | | 4 |
| 11 | OUT_SCALE | | 11 | | | 29 | LO_PRI | | | | 1 |
| 12 | GRANT_DENY | | 2 | | | 30 | LO_LIM | | | | 4 |
| 13 | IO_OPTS | | | | 2 | 31 | LO_LO_PRI | | | | 1 |
| 14 | STATUS_OPTS | | | | 2 | 32 | LO_LO_LIM | | | | 4 |
| 15 | CHANNEL | | | | 2 | 33 | HI_HI_ALM | | | | |
| 16 | L_TYPE | | | | 1 | 34 | HI_ALM | | | | |
| 17 | LOW_CUT | | | | 4 | 35 | LO_ALM | | | | |
| 18 | PV_FTIME | | | | 4 | 36 | LO_LO_ALM | | | | |
| Sub totals | | 18 | 26 | 18 | 20 | 37 | BLOCK_ERR_ DESC_1 | | | | |
| | | | | | | Sub totals | | 13 | 0 | 13 | 26 |
| | | | | | | Totals | | 31 | 26 | 31 | 46 |

*This page intentionally left blank.*

# 5. Control Selector

## Overview

The Control Selector block is intended to select one of two or three control signals in a manner determined by SEL_TYPE, when the block is in Auto mode. The Control Selector should be distinguished from the Input Selector block, which is used for selecting a measurement from input or calculation blocks.

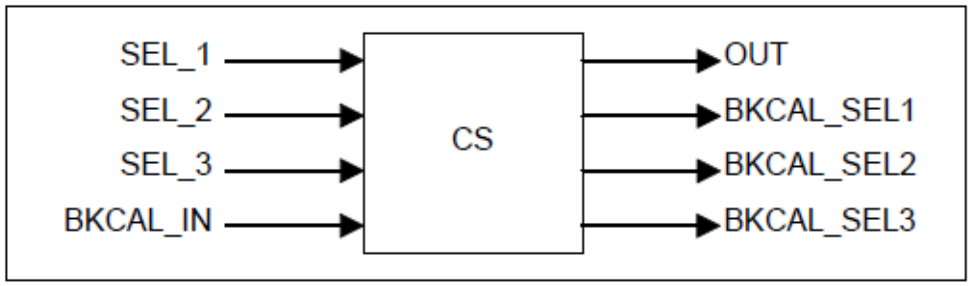


**Figure 17 Inputs Outputs of Control Selector Function Block**

## Schematic

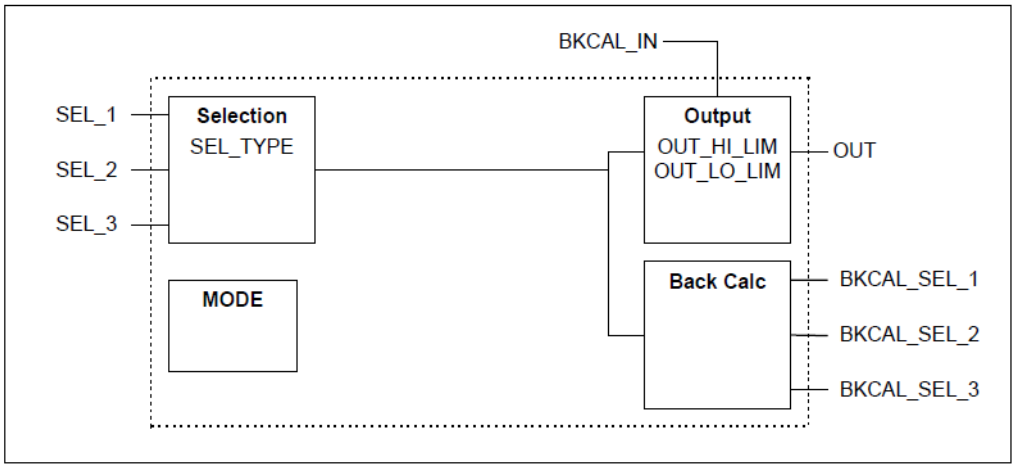


**Figure 18 Function Diagram of Control Selector Function Block**

## Description

All inputs to the selector block are assumed to have the same scaling as OUT, since any one of them may be selected to be OUT. Three separate BKCAL_SEL_N outputs are available, one for each SEL_N input. The status indicates those inputs that are not selected. Control blocks that are not selected are limited in one direction only, determined by the type of selector. The value of each BKCAL_SEL_N output is the same as OUT. The limits of back calculation outputs corresponding to deselected inputs are high for a low selector and low for a high selector, or one of each for a mid selector.

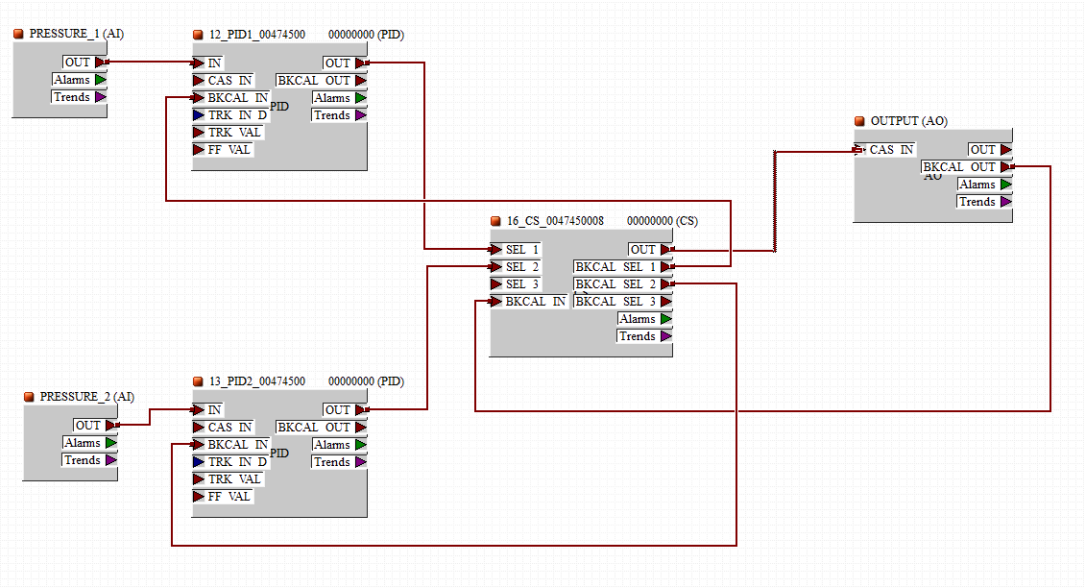


**Figure 19 Control Selector Block Example**

## Supported Modes

O/S, IMan, Man, and Auto.

## Alarm Types

Standard block alarm.

## Mode Handling

Standard transition in and out of O/S.

## Status Handling

If an input is Uncertain, it is treated as Bad unless the status option Use Uncertain as Good is set.

When any inputs is Bad, the actual mode goes to Man, as it does for a bad primary input to a PID. This condition sets IFS in the output status if the option IFS if BAD IN is set in STATUS_OPTS.

If the status of BKCAL_IN is NI (Not Invited) or IR, it is passed back on all three BKCAL outputs. This causes all initializable inputs to initialize to the BKCAL_IN value. Otherwise, if the status of BKCAL_IN is not normal, it is passed back on BKCAL_SEL_Y, where Y is the number of the selected input. The BKCAL_SEL_X outputs for deselected inputs just have the Not Selected status with the appropriate high or low limit set.

When the block is in Man mode, no input is selected. All three BKCAL outputs have a Not Invited status and Constant limits, with a value equal to OUT.

## Initialization

Standard.

## BLOCK_ERR_DESC_1 Parameter

Table 17 lists the common block errors that can occur.

**Table 17 Common Block Errors**

| Hex | Description |
|---|---|
| 0x40000000U | CHANNEL parameter is not set |
| 0x20000000U | Incorrect CHANNEL parameter value |
| 0x10000000U | |
| 0x08000000U | L_TYPE parameter is set for Direct, but input and output scale does not match |
| 0x02000000U | XD_SCALE upper and lower range not set correctly |
| 0x01000000U | OUT_SCALE upper and lower range not set correctly |
| 0x00800000U | PV_SCALE upper and lower range not set correctly |
| 0x00400000U | TRK_SCALE upper and lower range not set correctly |
| 0x00200000U | FF_SCALE upper and lower range not set correctly |
| 0x00100000U | BYPASS parameter is not set |
| 0x00080000U | SHED_OPT parameter is not set |
| 0x00040000U | SP_HI_LIM or SP_LO_LIM parameters are not set correctly |
| 0x00020000U | OUT_HI_LIM or OUT_LO_LIM parameters are not set correctly |
| 0x00010000U | ARITH_TYPE Parameter is not set |
| 0x00008000U | SEL_TYPE Parameter is not set |
| 0x00002000U | RANGE_HI or RANGE_LO parameters are not set correctly |
| 0x00001000U | COMP_HI_LIM or COMP_LO_LIM parameters are not set correctly |
| 0x00000040U | IN_ARR parameter values are not consistent |
| 0x00000020U | LOCKVAL Parameter is not set |
| 0x00000001U | Block is not scheduled |

# Control Selector Block Access

**Table 18 Control Selector Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 12 | SEL_2 | 5 | | 5 | |
| 2 | TAG_DESC | | | | | 13 | SEL_3 | 5 | | 5 | |
| 3 | STRATEGY | | | | 2 | 14 | SEL_TYPE | | | | 1 |
| 4 | ALERT_KEY | | | | 1 | 15 | BKCAL_IN | | | 5 | |
| 5 | MODE_BLK | 4 | | 4 | | 16 | OUT_HI_LIM | | 4 | | |
| 6 | BLOCK_ERR | 2 | | 2 | | 17 | OUT_LO_LIM | | 4 | | |
| 7 | OUT | 5 | | 5 | | 18 | BKCAL_SEL_1 | | | 5 | |
| 8 | OUT_SCALE | | 11 | | | 19 | BKCAL_SEL_2 | | | 5 | |
| 9 | GRANT_DENY | | 2 | | | 20 | BKCAL_SEL_3 | | | 5 | |
| 10 | STATUS_OPTS | | | | 2 | 21 | UPDATE_EVT | | | | |
| 11 | SEL_1 | 5 | | 5 | | 22 | BLOCK_ALM | | | | |
| Sub totals | | 18 | 26 | 18 | 20 | 23 | BLOCK_ERR | | | | |
| | | | | | | Sub totals | | 10 | 8 | 30 | 1 |
| | | | | | | Totals | | 28 | 23 | 48 | 8 |

*This page intentionally left blank.*

# 6. OS Function Block

## General

The output splitter block provides the capability to drive two control outputs from a single input (Figure 20). Each output is a linear function of some portion of the input. Back calculation support is provided using the same linear function in reverse. Cascade initialization is supported by a decision table for combinations of input and output conditions.

This block would normally be used in split ranging or sequencing of multiple valve applications.

A typical split range application has both valves closed when the splitter setpoint is about 50%. One of the valves opens fully as the input drops to 0%. The other valve opens as the input rises above 50%.

A typical sequencing application has both valves closed at 0% input. One valve opens fully as the SP rises to about 50%, and the other stays shut. The second valve opens as the input rises above 50%, and the first valve may remain open or shut off quickly.

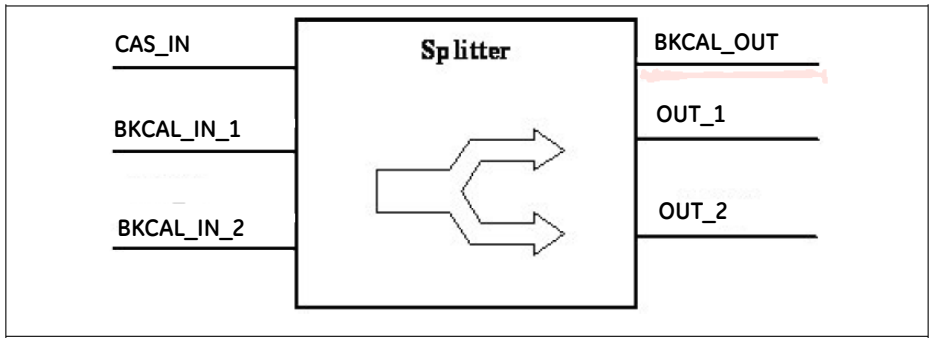


**Figure 20 Inputs/Outputs of OS Function Block**

The OS block receives a control signal and converts it into two signals in accordance with the predefined relationships. The major functions of the OS block include (Figure 21):

- Conversion of the setpoint (SP) value into two output values (OUT_1 and OUT_2) in accordance with the user-specified characteristics (set in IN_ARRAY and OUT_ARRAY)
- Generation of the output value to be fed back to the upstream block (BKCAL_OUT).

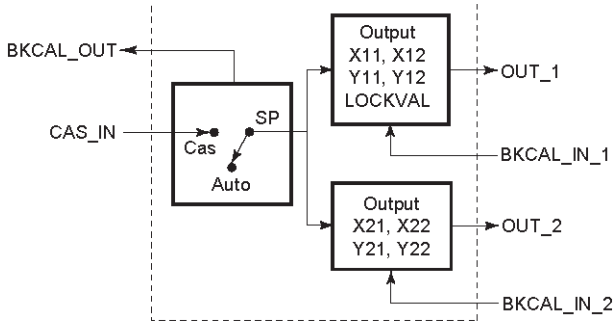


**Figure 21 Function Diagram of OS Function Block**

# Modes

Follows normal FF modes.

# Output Processing

The values of OUT_1 and OUT_2 with respect to the value of SP, which is the value of the input from the upstream block (CAS_IN) in the Cas mode or the local setpoint value in the Auto mode, are determined as shown in Figure 22.

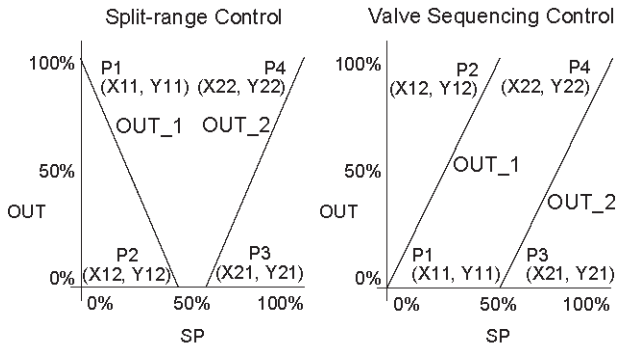


**Figure 22 Examples of Valve Operation Characteristics**

These characteristics are determined by the array element values in parameters:

- IN_ARRAY and OUT_ARRAY
- IN_ARRAY: [X11, X12, X21, X22]
- OUT_ARRAY: [Y11, Y12, Y21, Y22]

Coordinates P1 (X11, Y11) and P2 (X12, Y12) define the start and stop points of the characteristics for OUT_1, and P3 (X21, Y21) and P4 (X22, Y22) define those for OUT_2. These two operation characteristics may overlap each other, or start from the same point and have different slopes; however, all the following conditions must be met at all times. Settings of IN_ARRAY that do not meet one or more of these conditions cause a BLOCK_ERR, disabling the block from exiting the O/S mode.

X21 . X11
X12 > X11
X22 > X21

In areas outside the endpoints (i.e., start and stop points) of each operation characteristic, the output is retained at the Y value at the nearer end point. For OUT_1, however, depending on the setting of LOCKVAL, it is possible to:

Set the value of OUT_1 to Y11 in the areas outside the endpoints if SP is greater than X12 and if LOCKVAL is false.

When this action is enabled, the value set in HYSTVAL serves as hysteresis, which affects the output as follows (Figure 23):

When SP has increased beyond X12, OUT_1 is set to Y11.

Then, after SP has decreased below X12 minus HYSTVAL, OUT_1 returns to follow the set characteristic.

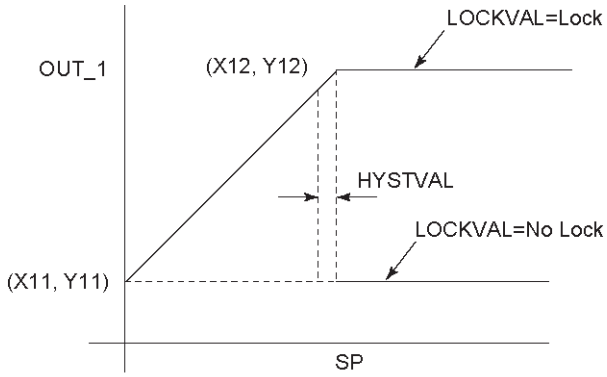


**Figure 23 LOCKVAL and HYSTVAL**

When both downstream blocks of the OS block are ready for cascade connection, the OS block connects the block on the side of OUT_1 first. For bumpless mode change on the side of OUT_2, the balancing time for connection can be set in BAL_TIME. When either downstream block alone is ready for cascade connection, the OS block connects it and enters the Cas mode. When neither downstream block is ready for cascade connection, the OS block mode is set to IMan.

## Backward Path (BKCAL_OUT)

The value of SP or a value calculated from the value of either BKCAL_IN_1 or BKCAL_IN_2, depending on the handshake status with the downstream blocks, is output through BKCAL_OUT. In normal operating conditions (i.e., BLK_MODE.actual is Cas or Auto), BKCAL_OUT is set to the value of SP.

## BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

## OS Block Access

**Table 19 OS Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 14 | CAS_IN | 5 | | 5 | 2 |
| 2 | TAG_DESC | | | | | 15 | BKCAL_OUT | | | 5 | |
| 3 | STRATEGY | | | | 2 | 16 | IN_ARRAY | | | | 16 |
| 4 | ALERT_KEY | | | | 1 | 17 | OUT_ARRAY | | | | 16 |
| 5 | MODE_BLK | 4 | | 4 | | 18 | LOCKVAL | | | | 1 |
| 6 | BLOCK_ERR | 2 | | 2 | | 19 | BKCAL_IN_1 | | | 5 | |
| 7 | SP | 5 | | 5 | | 20 | BKCAL_IN_2 | | | 5 | |
| 8 | OUT_1 | 5 | | 5 | | 21 | BAL_TIME | | | | 4 |
| 9 | OUT_2 | 5 | | 5 | | 22 | HYSTVAL | | | | 4 |
| 10 | OUT_1_ RANGE | | 11 | | 2 | 23 | UPDATE_EVT | | | | |
| 11 | OUT_2_ RANGE | 5 | 11 | 5 | | 24 | BLOCK_ALM | | | | |
| 12 | GRANT_DENY | | 2 | | | 25 | BLOCK_ERROR _DESC | | | | |
| 13 | STATUS_OPTS | | | | 2 | | | | | | |
| Sub totals | | 23 | 26 | 27 | 7 | Sub totals | | 23 | 26 | 23 | 7 |
| | | | | | | Totals | | 28 | 26 | 43 | 48 |

# 7. Multiple Analog Input Function Block (MAI)

## Overview

The MAI block makes available for the FF network eight analog variables of the I/O subsystem through its eight output parameters: OUT_1 through OUT_8, whose values must be expressed in engineering units.

Status indication in the OUT_x output parameters depends on the I/O subsystem and the transducer block. There is individual detection of sensor failure that can be indicated in the status of related OUT_x parameter.

## Schematic

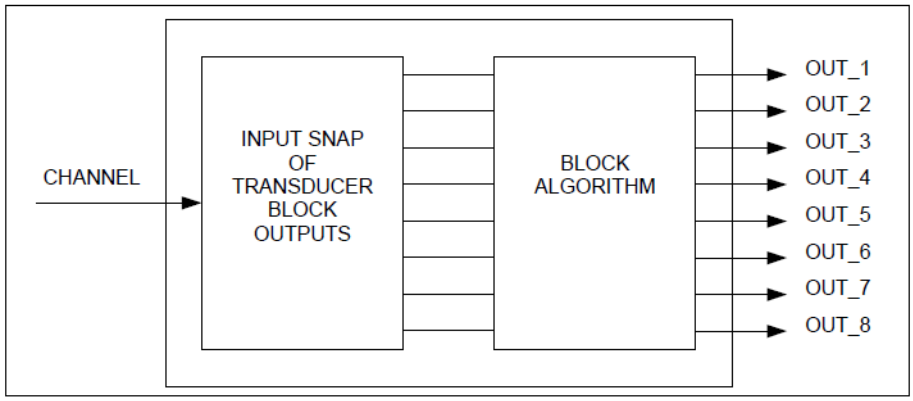


**Figure 24 Function Diagram of Multiple Analog Input Function Block**

## Usage

To integrate the positioner related parameters in the control system, you must set *CH_MAI_CHANNELS* to *20*.

## Supported Modes

O/S, Man and Auto modes are supported.

## Alarm Types

Standard block alarm are supported.

## Mode Handling

Standard transition in and out of O/S.

## Status Handling

The status values described in the Parameter Formal Model of Part 1 apply, with the exception of the control sub-status values.

## Initialization

The initialization is specific to the block algorithm.

## BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

# Multiple Analog Input Function Block Access

**Table 20 Multiple Analog Input Block Access**

| Index | Parameter Mnemonic (Parameter Name) | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 |
| 2 | TAG_DESC | | | | |
| 3 | STRATEGY | | | | 2 |
| 4 | ALERT_KEY | | | | 1 |
| 5 | MODE_BLK | 4 | | 4 | |
| 6 | BLOCK_ERR | 2 | | 2 | |
| 7 | CHANNEL | | | | 2 |
| 8 | OUT_1 (Supply Pressure) | 5 | | 5 | |
| 9 | OUT_2 (Actuator A Pressure) | 5 | | 5 | |
| 10 | OUT_3 (Actuator B Pressure) | 5 | | 5 | |
| 11 | OUT_4 (Pilot Pressure) | 5 | | 5 | |
| 12 | OUT_5 (IP Drive Current) | 5 | | 5 | |
| 13 | OUT_6 (Temperature) | 5 | | 5 | |
| 14 | OUT_7 (Working Setpoint) | 5 | | 5 | |
| 15 | OUT_8 (Working Position) | 5 | | 5 | |
| 16 | UPDATE_EVT | | | | |
| 17 | BLOCK_ALM | | | | |
| 18 | BLOCK_ERR_DESC_1 | | | | |
| Totals | | 48 | 2 | 48 | 7 |

*This page intentionally left blank.*

# 8. Input Selector (IS) Function Block

## General

The signal selector block provides selection of up to four inputs and generates an output based on the configured action (Figure 25). This block normally receives its inputs from AI blocks. The block performs maximum, minimum, middle, average and *first good* signal selection.

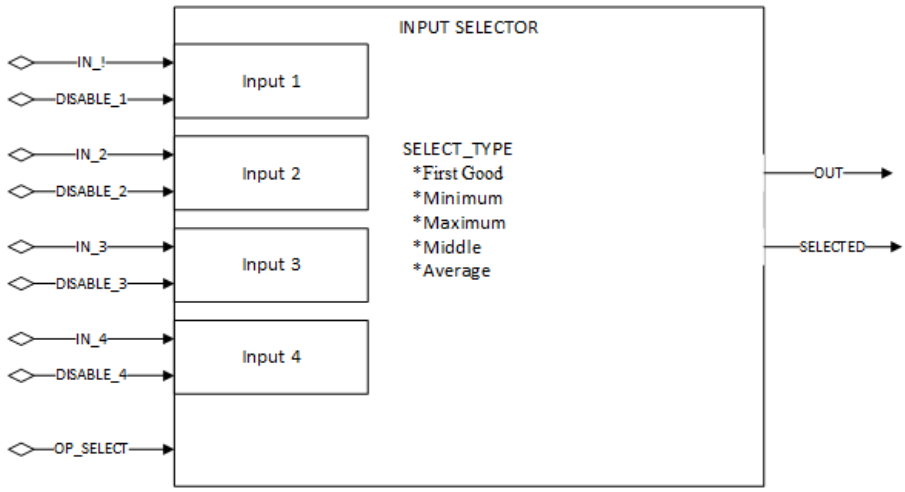


**Figure 25 Inputs/Outputs of Input Selector (IS) Function Block**

With a combination of parameter configuration options the block can function as a rotary position switch, or a validated priority selection based on the use of the first good parameter and the disable_n parameter (Figure 25). As a switch, the block can receive switching information from either the connected inputs or from an operator input. The block also supports the concept of a middle selection. Although the normal configuration for this feature would be with three signals, the block should generate an average of the middle two if four signals are configured or the average of two if three are configured and a bad status is passed to one of the inputs. Logic is provided for handling uncertain and bad signals in conjunction with configured actions. The intended application of this block is to provide control signal selection in the forward path only, therefore, no back calculation support is provided. SELECTED is a second output that indicates which input has been selected by the algorithm.

## Function Supported

This block is intended to be used in a forward path only and is not intended to receive signals from the output of a controller. There is no back calculation support or propagation of control status values. Because the block is usually connected to AI blocks, there is no need of back calculation. The processing of the block is as follows:

*Input Processing*

If DISABLE_n is true then don't process (ignore) the respective input IN_n.

Process the Use Uncertain as Good status options. Discard (ignore) inputs whose status is bad.

If there are no inputs left, or fewer than MIN_GOOD inputs, then set the value of SELECTED to zero. Do not do selection processing.

*Selection Processing*

If OP_SELECT is non-zero, the OP_SELECT value shall determine the selected input, regardless of the SELECT_TYPE selection. Set SELECTED to the number of the input used.

If SELECT_TYPE is First Good, transfer the value of the first remaining input to the output of the block. Set SELECTED to the number of the input used.

If SELECT_TYPE is Minimum, sort the remaining inputs by value. Transfer the lowest value to the output of the block. Set SELECTED to the number of the input with the lowest value.

If SELECT_TYPE is Maximum, sort the remaining inputs by value. Transfer the highest value to the output of the block. Set SELECTED to the number of the input with the highest value.

If SELECT_TYPE is Middle, sort the remaining inputs by value. If there are 3 or 4 values, discard the highest and lowest value. If two values are left, compute their average. Transfer the value to the output of the block. Set SELECTED to zero if an average was used, else set SELECTED to the number of the input with the middle value.

If SELECT_TYPE is Average compute the average of the remaining inputs and transfer the value to the output of the block. Set

SELECTED to the number of inputs used in the average.

Figure 26 shows an example use of Input Selector block configured to select the maximum value out of three input blocks.

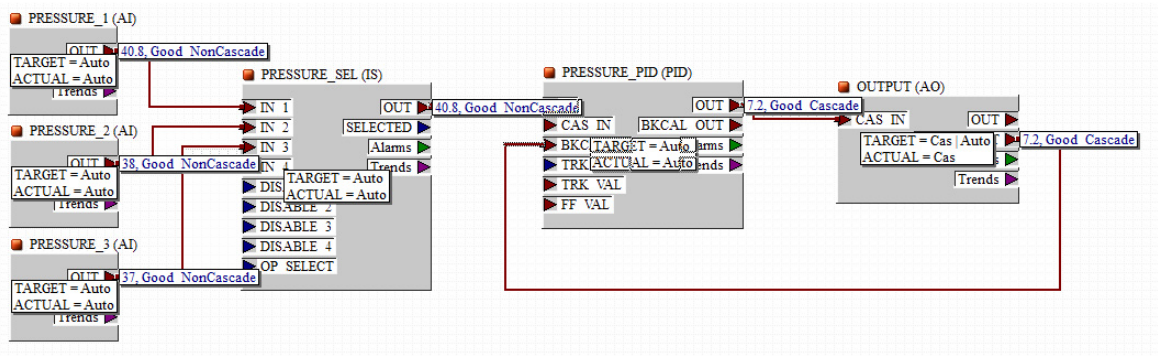


**Figure 26 Input Selector Block Input Selection Example**

### Supported Modes

O/S, Man, and Auto.

### Alarm Types

Standard block alarm.

### Mode Handling

Standard.

### Status Handling

If there are no inputs used, or fewer than MIN_GOOD inputs, then the status of OUT shall be set to Bad Non-specific.

The SELECTED output shall have Good(NC) status, unless the block is out of service.

Status options for Use Uncertain as Good and Uncertain if Manual shall be supported.

### Initialization

Standard.

### Power Failure Recovery

Standard.

## BLOCK_ERR_DESC_1 Parameter

See “BLOCK_ERR_DESC_1 Parameter” on page 42.

## Minimum Configuration

SELECT_TYPE

## IS Block Access

**Table 21 IS Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 13 | IN_3 | 5 | | 5 | |
| 2 | TAG_DESC | | | | | 14 | IN_4 | 5 | | 5 | |
| 3 | STRATEGY | | | | 2 | 15 | DISABLE_1 | 2 | | 2 | |
| 4 | ALERT_KEY | | | | 1 | 16 | DISABLE_2 | 2 | | 2 | |
| 5 | MODE_BLK | 4 | | 4 | | 17 | DISABLE_3 | 2 | | 2 | |
| 6 | BLOCK_ERR | 2 | | 2 | | 18 | DISABLE_4 | 2 | | 2 | |
| 7 | OUT | 5 | | 5 | | 19 | SELECT_TYPE | | | | 1 |
| 8 | OUT_RANGE | | 11 | | | 20 | MIN_GOOD | | | | 1 |
| 9 | GRANT_DENY | | 2 | | | 21 | SELECTED | 2 | | 2 | |
| 10 | STATUS_OPTS | | | | 2 | 22 | OP_SELECT | 2 | | 2 | |
| 11 | IN_1 | 5 | | 5 | | 23 | UPDATE_EVT | | | | |
| 12 | IN_2 | 5 | | 5 | | 24 | BLOCK_ALM | | | | |
| Sub totals | | 23 | 15 | 23 | 7 | 25 | BLOCK_ERR_ DESC_1 | | | | |
| | | | | | | Sub totals | | 23 | 15 | 23 | 7 |
| | | | | | | Totals | | 45 | 15 | 45 | 9 |

# 9. Discrete Output

## Overview

The DO block converts the value in SP_D to control the hardware identified by CHANNEL parameters selection.

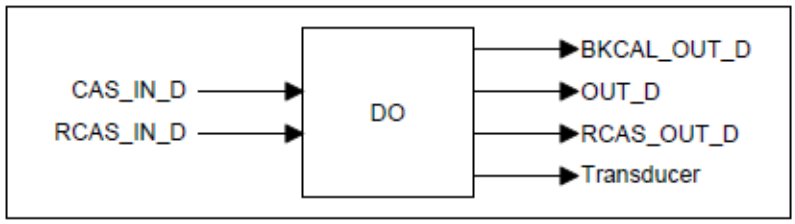


**Figure 27 Inputs Outputs of Discrete Output (AI) Function Block**

## Schematic

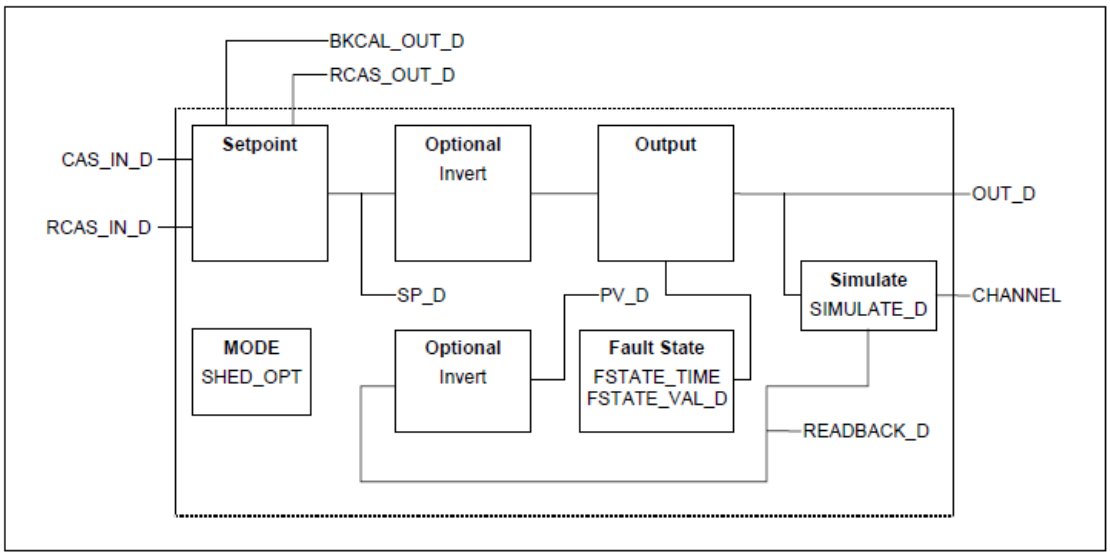


**Figure 28 Function Diagram of Discrete Output Function Block**

## Description

The SP_D supports the full cascade sub-function. Cas mode must be used to transfer the output of another block to the SP_D of the DO. There are additional I/O options which cause the SP_D value to track the PV_D value when the block is in an actual mode of LO or when the target mode for the block is Man.

If the hardware readback value is selected, it is used for READBACK_D, which, after accounting for the Invert I/O option, acts as the PV_D for this block. If not supported, READBACK_D is generated from OUT_D. The OUT_D and READBACK_D parameters both use XD_STATE. The PV_D and SP_D use PV_STATE.

The Invert I/O option can be used to do a Boolean NOT function between the SP_D and the hardware. The selected channel influences behavior as in Table 22.

### Channel Parameter Applications

**Table 22 Channels for Discrete Output Blocks**

| Channel Value | Domain | Usage | Behavior | Invert Behavior |
|---|---|---|---|---|
| 0 | NA | Not Configured | Prevents the Discrete Output block from going into Auto Mode. | NA |
| 2 | Position | Open/Close | Updates the FINAL_VALUE_D parameter in the Transducer block. Readback is provided from FINAL_POSITION_VALUE_D.<br>Block the switching of the DO block to Auto mode if the TB.SETPOINT_SOURCE is not equal to 2 (DO block in Open/Close mode).<br>Reports an error for the setting mismatch. | OUT_D = 1 if SP_D = 0.<br>OUT_D = 0 if SP_D not equal to 0. |
| 3 | Position | Discrete Position<br>(Range: 0 to 100%) | Updates the FINAL_VALUE_DINT parameter in the Transducer block. Readback is provided from FINAL_POSITION_VALUE_DINT.<br>Blocks the switching of the DO block to Auto mode if the TB.SETPOINT_SOURCE is not equal to 3 (Discrete Output block in Discrete Mode).<br>Reports an error for the setting mismatch. | If SP_D ≤ 100% OUT_D = 100% - SP_D.<br>If SP_D > 100% OUT_D = 0. |

**Table 22 Channels for Discrete Output Blocks (Continued)**

| Channel Value | Domain | Usage | Behavior | Invert Behavior |
|---|---|---|---|---|
| 4 | Discrete | Discrete Switch 1 | The DO.OUT value updates the TB.DISCRETE_OUTPUT_1 parameter, which controls the DO1 switch controlled by the APP processor. | |
| 5 | Discrete | Discrete Switch 2 | The DO.OUT value updates the TB.DISCRETE_OUTPUT_2 parameter, which controls the DO2 switch controlled by the APP processor. | OUT_D = 1 if SP_D = 0.<br>OUT_D = 0 if SP_D not equal to 0. |

## Supported Modes

O/S, LO, Iman, Man, Auto, Cas, and RCas. The Man mode can be used to force the output, in a PLC sense. It may be that Man mode is not permitted, but it must be supported so that Man mode may be entered when leaving O/S. The IMan mode is used to indicated that there is no path to the final element.

## Alarm Types

Standard block alarm.

## Mode Handling

Standard transition in and out of O/S.

## Status Handling

The status values described in Output Parameter Formal Model of Part 1 of the FF891 specification apply, with the exception of the control sub-status values. The following options from STATUS_OPTS apply:

*Propagate Fault Backward*

## Initialization

Standard.

## Fault State Recovery

Standard.

## Bad to Good Output Transition

Use the readback value, if available, or restore the NV value of OUT_D.

## BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

## DO Block Access

**Table 23 DO Block Access**

| Index | Parameter Mnemonic | VIEW_1 | VIEW_2 | VIEW_3 | VIEW_4 | Index | Parameter Mnemonic | VIEW_1 | VIEW_2 | VIEW_3 | VIEW_4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 14 | IO_OPTS | | | | 2 |
| 2 | TAG_DESC | | | | | 15 | STATUS_OPTS | | | | 2 |
| 3 | STRATEGY | | | | 2 | 16 | READBACK_D | 2 | | 2 | |
| 4 | ALERT_KEY | | | | 1 | 17 | CAS_IN_D | 2 | | 2 | |
| 5 | MODE_BLK | 4 | | 4 | | 18 | CHANNEL | | | | 2 |
| 6 | BLOCK_ERR | 2 | | 2 | | 19 | FSTATE_TIME | | | | 4 |
| 7 | PV_D | 2 | | 2 | | 20 | FSTATE_VAL_D | | | | 1 |
| 8 | SP_D | 2 | | 2 | | 21 | BKCAL_OUT_D | | | 2 | |
| 9 | OUT_D | 2 | | 2 | | 22 | RCAS_IN_D | | | 2 | |
| 10 | SIMULATE_D | | | | | 23 | SHED_OPT | | | | 1 |
| 11 | PV_STATE | | 2 | | | 24 | RCAS_OUT_D | | | 2 | |
| 12 | XD_STATE | | 2 | | | 25 | UPDATE_EVT | | | | |
| 13 | GRANT_DENY | | 2 | | 2 | 26 | BLOCK_ALM | | | | |
| Sub totals | | 14 | 8 | 14 | 5 | 27 | BLOCK_ERR_DESC_1 | | | | |
| | | | | | | Sub totals | | 4 | 0 | 10 | 12 |
| | | | | | | Totals | | 18 | 8 | 24 | 17 |

# 10. DI Function Block

## General

An SVI FF contains two DI function blocks, which individually transfer the valve-position high and low limit switch signals generated by the transducer block (Figure 29).

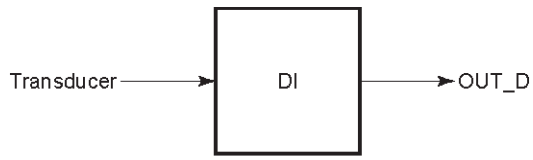


**Figure 29 Inputs/Outputs of DI Function Block**

The major functions of a DI function block include (Figure 30):

- Signal inversion (I/O processing option)
- Simulation
- Filtering (time delay)
- Alarm generation

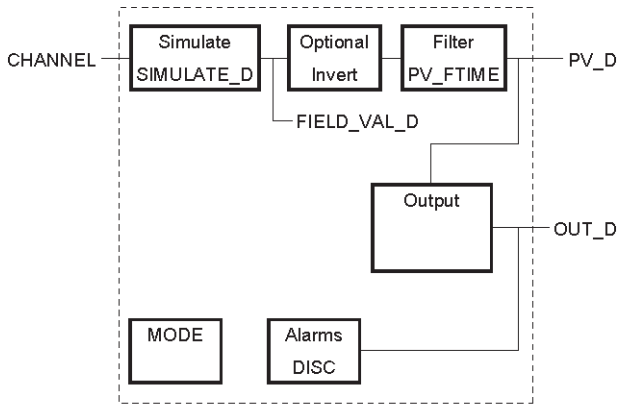


**Figure 30 Function Diagram of DI Function Block**

## Modes

The target mode for a DI function block can be set from three block modes: O/S, Auto, and Man.

## PV Value (PV_D)

A limit switch signal is transferred from the transducer block via a channel. Normally, the Transducer Value and Transducer Status values in SIMULATE_D are copied to FIELD_VAL_D, indicating the on/off status of the corresponding limit switch. If SIMULATE_D is set to *Enable*, the Simulate Value and Simulate Status values in SIMULATE_D are copied to FIELD_VAL_D.

SIMULATE_D contains the following data:

| | |
|---|---|
| *Simulate Status* | Status to be set in simulation mode |
| *Simulate Value* | Value to be set in simulation mode |
| *Transducer Status* | Status of input from transducer |
| *Transducer Value* | Value of input from transducer |
| *Enable/Disable* | Whether to enable *2* or disable *1* simulation |

The value of FIELD_VAL_D is copied to the process value PV_D. At this time, if the Invert option (bit 0) is specified as true, the on/off status is inverted (Table 24).

**Table 24 FIELD_VAL_D**

| Value of FIELD_VAL_D | Value of PV_D | |
|---|---|---|
| | Invert = False | Invert = True |
| 0 | 0 (off) | 1 |
| ≥1 | 1 (on) | 0 |

## Filtering

Transfer of a change in the value of FIELD_VAL_D to the value of PV_D can be delayed for a desired time period set in the parameter PV_FTIME (in seconds).

## Output

The value of the output OUT_D is generated based on the value of PV_D.

## IO_OPTS and STATUS_OPTS

These parameters stipulate options about block's signal processing and mode transitions.

The settings of these options are made by setting or resetting the respective bits: on = true, off = false. Table 25 shows the options available in the DI block IO_OPTS.

**Table 25 IO_OPTS of DI Block**

| Bit # | Meaning | Description |
|---|---|---|
| 0 | Invert | Inverts the on/off status. |

Table 26 shows the options available in the AO block STATUS_OPTS.

**Table 26 IO_OPTS of DI Block**

| Bit # | Meaning | Description |
|---|---|---|
| 3 | Propagate Fault Forward | Stipulates the handling of the value and data status of OUT_D when the quality component of the data status of SIMULATE_D falls to Bad and the substatus component falls to device failure or sensor failure.<br>If this option is true, then it:<br>□ Does not generate a block alarm.<br>□ Sets the status and value of SIMULATE_D in OUT_D.<br>If this option is false, then it:<br>□ Generates the *input failure* block alarm.<br>□ Set the quality and sub-status components of the status of OUT_D to Bad and non specific, respectively. |
| 8 | Uncertain if Man mode | Sets the status of OUT_D to uncertain when in Man mode. |

# Alarm Processing

## Block Alarms

When a condition shown in Table 27 is met in a DI block, the block changes the bit statuses of BLOCK_ERROR accordingly and generates a block alarm.

**Table 27 BLOCK_ERROR in DI Block**

| Bit # | Meaning | Description |
|---|---|---|
| 3 | Simulate Active | SIMULATE_D is active. |
| 7 | Input Failure / process variable has BAD status | Propagate Fault Backward in STATUS_OPTS is false, and the sub-status component of the status of READBACK is sensor failure or device failure. |
| 15 | Out of Service | The target mode is O/S. |

## Discrete Alarm

The parameter DISC_ALM is a discrete alarm of the parameter OUT_D.

When the value of OUT_D agrees with the value of DISC_LIM, the alarm state of DISC_ALM is set to active and an alert is generated.

# BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

# Channel Parameter Applications

**Table 28 Channels for Discrete Input Blocks**

| Channel Value | Usage | Behavior |
|---|---|---|
| 0 | Not Configured | Prevents the Discrete Input block from going into Auto Mode. |
| 17 | Discrete Input | Provides the state of the discrete input to the FF control. |
| 18 | State of Sw1 | Sends the state of the discrete Switch 1 from APP processor to the FF control. |
| 19 | State of Sw2 | Sends the state of the discrete Switch 2 from APP processor to the FF control. |

# DI Block Access

**Table 29 DI Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 13 | IO_OPTS | | | | 2 |
| 2 | TAG_DESC | | | | | 14 | STATUS_OPTS | | | | 2 |
| 3 | STRATEGY | | | | 2 | 15 | CHANNEL | | | | 2 |
| 4 | ALERT_KEY | | | | 1 | 16 | PV_FTIME | | | | 4 |
| 5 | MODE_BLK | 4 | | 4 | | 17 | FIELD_VAL_D | 2 | | 2 | |
| 6 | BLOCK_ERR | 2 | | 2 | | 18 | UPDATE_EVT | | | | |
| 7 | PV_D | 2 | | 2 | | 19 | BLOCK_ALM | | | | |
| 8 | OUT_D | 2 | | 2 | | 20 | ALARM_SUM | 8 | | 8 | |
| 9 | SIMULATE_D | | | | | 21 | ACK_OPTION | | | | 2 |
| 10 | XD_STATE | | 2 | | | 22 | DISC_PRI | | | | 1 |
| 11 | OUT_STATE | | 2 | | | 23 | DISC_LIM | | | | 1 |
| 12 | GRANT_DENY | | 2 | | | 24 | DISC_ALM | | | | |
| Sub totals | | 12 | 8 | 12 | 5 | 25 | BLOCK_ERR_ DESC_1 | | | | |
| | | | | | | Sub totals | | 10 | 0 | 10 | 14 |
| | | | | | | Totals | | 22 | 8 | 22 | 19 |

*This page intentionally left blank.*

# 11. AR (Arithmetic) Function Block

## General

The AR (Arithmetic) block (Figure 31) uses inputs as arguments on a pre-defined set of math functions to calculate an output.

The AR block is intended for use in calculating measurements from combinations of signals from sensors. It is not intended to be used in a control path, so it does not support control status propagation or back calculation. It has no process alarms.

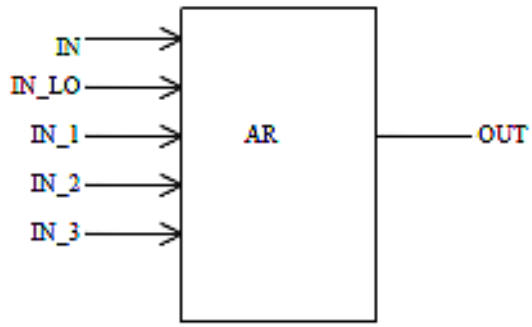


**Figure 31 Inputs/Outputs of AR Function Block**

The AR block has five inputs. The first two are dedicated to a range extension function that results in a PV, with status reflecting the input in use.

The remaining three inputs are combined with the PV in a selection of four term math functions that have been found useful in a variety of measurements. The inputs used to form the PV must come from devices with the desired engineering units, so that the PV enters the equation with the right units. Each of the additional inputs has a bias and gain constant. The bias can be used to correct for absolute temperature or pressure. The gain can be used to normalize terms within a square root function. The output also has gain and bias constants for any further adjustment required.

The major functions of the block are shown in Figure 32.

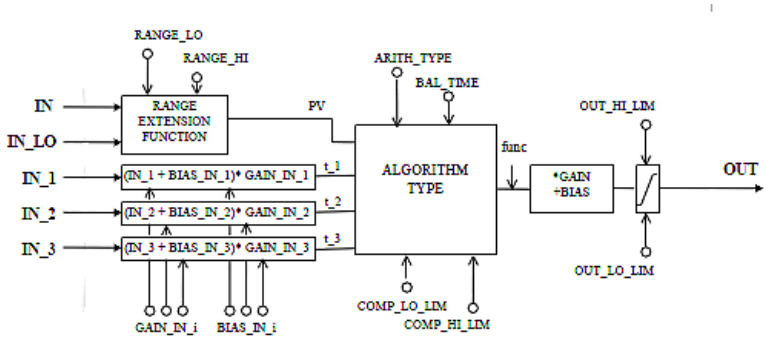


**Figure 32 Function Diagram of AR Function Block**

The range extension function has a graduated transfer, controlled by two constants referenced to IN. An internal value, g, is zero for IN less than RANGE_LO. It is one when IN is greater than RANGE_HI. It is interpolated from zero to one over the range of RANGE_LO to RANGE_HI. The equation for PV follows:

$$PV = g * IN + (1-g) * IN\_LO$$

If the status of IN_LO is unusable and IN is usable and greater than RANGE_LO, then set *g* to one. If the status of IN is unusable, and IN_LO is usable and less than RANGE_HI, then set *g* to zero. In each case, the PV should have a status of Good until the condition no longer applies. Otherwise, the status of IN_LO is used for the PV if g is less than 0.5, while IN is used for g greater than or equal to 0.5.

Six constants are used for the three auxiliary inputs. Each has a BIAS_IN_i and a GAIN_IN_i. The output has a BIAS and a GAIN static constant. For the inputs, the bias is added and the gain is applied to the sum. The result is an internal value called t_i in the function equations. The equation for each auxiliary input is the following:

$$t\_i = (IN\_i + BIAS\_IN\_i) * GAIN\_IN\_i$$

The flow compensation functions have limits on the amount of compensation applied to the PV, to assure graceful degradation if an auxiliary input is unstable. The internal limited value is f.

## Functions Supported

The following function types are supported:

*Flow compensation, linear*

Used for density compensation of volume flow.

func = f * PV

*Flow compensation, square root*

Usually, IN_1 is pressure, IN_2 temperature, and IN_3 is the compressibility factor Z.

func = f * PV

f = sqrt (t_1 / t_2 / t_3) [limited]

*Flow compensation, approximate*

Both IN_2 and IN_3 would be connected to the same temperature.

func = f * PV

f = sqrt (t_1 * t_2 * t_3 * t_3) [limited]

*BTU flow*

Where IN_1 is inlet temperature, and IN_2 the outlet temperature

func = f * PV

f = (t_1 - t_2) [limited]

*Traditional Multiply Divide*

func = f * PV

f = (t_1 / t_2) + t_3 [limited]

*Average*

func = (PV + t_1 + t_2 + t_3) / f

f = number of inputs used in computation (unusable inputs are not used).

*Traditional Summer*

func = PV + t_1 + t_2 + t_3

*Fourth order polynomial*

All inputs except IN_LO (not used) are linked together.

func = PV + t_1 ** 2 + t_2 ** 3 + t_3 ** 4

*Simple HTG compensated level*

Where PV is the tank base pressure, IN_1 is the top pressure, IN_2 is the density correction pressure, and GAIN is the height of the density tap.

func = (PV -t_1) / (PV - t_2)

After the value of func is calculated, it is multiplied by GAIN, and then BIAS is added to the result. Finally, high and low output limits are applied, and the result is the term PRE_OUT. If the mode is Auto, PRE_OUT becomes OUT.

### Supported Modes

O/S, Man, and Auto.

### Alarm Types

Standard block alarm.

### Mode Handling

The algorithm never changes the mode, even when inputs go bad.

If the mode is changed to Man, an internal value is set to the difference between OUT and the output of the selected function. When the mode is changed to Auto, the difference value exponentially decays to zero with a time constant of BAL_TIME.

The output of the calculation function appears in PRE_OUT.

### Status Handling

The INPUT_OPTS bit string controls use of auxiliary inputs with less than good status. The status of unused inputs are ignored.

The status of the output is that of the worst of the inputs used in the calculation after applying INPUT_OPTS.

### Initialization

Standard.

### Power Failure Recovery

Standard.

## BLOCK_ERR_DESC_1 Parameter

See "BLOCK_ERR_DESC_1 Parameter" on page 42.

# AR Block Access

**Table 30 AR Block Access**

| Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 | Index | Parameter Mnemonic | VIEW _1 | VIEW _2 | VIEW _3 | VIEW _4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ST_REV | 2 | 2 | 2 | 2 | 19 | RANGE_HI | | | | 4 |
| 2 | TAG_DESC | | | | | 20 | RANGE_LO | | | | 4 |
| 3 | STRATEGY | | | | 2 | 21 | BIAS_IN_1 | | | | 4 |
| 4 | ALERT_KEY | | | | 1 | 22 | GAIN_IN_1 | | | | 4 |
| 5 | MODE_BLK | 4 | | 4 | | 23 | BIAS_IN_2 | | | | 4 |
| 6 | BLOCK_ERR | 2 | | 2 | | 24 | GAIN_IN_2 | | | | 4 |
| 7 | PV | 5 | | 5 | | 25 | BIAS_IN_3 | | | | 4 |
| 8 | OUT | 5 | | 5 | | 26 | GAIN_IN_3 | | | | 4 |
| 9 | PRE_OUT | 5 | | 5 | | 27 | COMP_HI_LIM | | | | 4 |
| 10 | PV_SCALE | | 11 | | | 28 | COMP_LO_LIM | | | | 4 |
| 11 | OUT_RANGE | | 11 | | | 29 | ARITH_TYPE | | | | 4 |
| 12 | GRANT_DENY | | 2 | | | 30 | BAL_TIME | | | | 4 |
| 13 | INPUT_OPTS | | | | 2 | 31 | BIAS | | | | 4 |
| 14 | IN | | | 5 | | 32 | GAIN | | | | 4 |
| 15 | IN_LO | | | 5 | | 33 | OUT_HI_LIM | | | | 4 |
| 16 | IN_1 | | | 5 | | 34 | OUT_LO_LIM | | | | 4 |
| 17 | IN_2 | | | 5 | | 35 | UPDATE_EVT | | | | |
| 18 | IN_3 | | | 5 | | 36 | BLOCK_ALM | | | | |
| Sub totals | | 23 | 26 | 48 | 7 | 37 | BLOCK_ERR_ DESC_1 | | | | |
| | | | | | | From left column | | 23 | 26 | 48 | 7 |
| | | | | | | Totals | | 23 | 26 | 48 | 68 |

*This page intentionally left blank.*

# DIRECT SALES OFFICE LOCATIONS

AUSTRALIA
Brisbane
Phone: +61-7-3001-4319
Fax: +61-7-3001-4399

Perth
Phone: +61-8-6595-7018
Fax: +61-8-6595-7299

Melbourne
Phone: +61-3-8807-6002
Fax: +61-3-8807-6577

BELGIUM
Phone: +32-2-344-0970
Fax: +32-2-344-1123

BRAZIL
Phone: +55-11-2146-3600
Fax: +55-11-2146-3610

CHINA
Phone: +86-10-8486-4515
Fax: +86-10-8486-5305

FRANCE
Courbevoie
Phone: +33-1-4904-9000
Fax: +33-1-4904-9010

GERMANY
Ratingen
Phone: +49-2102-108-0
Fax: +49-2102-108-111

INDIA
Mumbai
Phone: +91-22-8354790
Fax: +91-22-8354791

New Delhi
Phone: +91-11-2-6164175
Fax: +91-11-5-1659635

ITALY
Phone: +39-081-7892-111
Fax: +39-081-7892-208

JAPAN
Chiba
Phone: +81-43-297-9222
Fax: +81-43-299-1115

KOREA
Phone: +82-2-2274-0748
Fax: +82-2-2274-0794

MALAYSIA
Phone: +60-3-2161-0322
Fax: +60-3-2163-6312

MEXICO
Phone: +52-5-310-9863
Fax: +52-5-310-5584

THE NETHERLANDS
Phone: +0031-15-3808666
Fax: +0031-18-1641438

RUSSIA
Veliky Novgorod
Phone: +7-8162-55-7898
Fax: +7-8162-55-7921

Moscow
Phone: +7 495-585-1276
Fax: +7 495-585-1279

SAUDI ARABIA
Phone: +966-3-341-0278
Fax: +966-3-341-7624

SINGAPORE
Phone: +65-6861-6100
Fax: +65-6861-7172

SOUTH AFRICA
Phone: +27-11-452-1550
Fax: +27-11-452-6542

SOUTH & CENTRAL
AMERICA AND THE CARIBBEAN
Phone: +55-12-2134-1201
Fax: +55-12-2134-1238

SPAIN
Phone: +34-93-652-6430
Fax: +34-93-652-6444

UNITED ARAB EMIRATES
Phone: +971-4-8991-777
Fax: +971-4-8991-778

UNITED KINGDOM
Wooburn Green
Phone: +44-1628-536300
Fax: +44-1628-536319

UNITED STATES
Massachusetts
Phone: +1-508-586-4600
Fax: +1-508-427-8971

Corpus Christi, Texas
Phone: +1-361-881-8182
Fax: +1-361-881-8246

Deer Park, Texas
Phone: +1-281-884-1000
Fax: +1-281-884-1010

Houston, Texas
Phone: +1-281-671-1640
Fax: +1-281-671-1735

* Masoneilan, SVI, and ValVue are registered trademarks of the General Electric Company. FOUNDATION Fieldbus is a trademark of the FOUNDATION Fieldbus organization. Other company names and product names used in this document are the registered trademarks or trademarks of their respective owners.



GEA31248B 05/2014