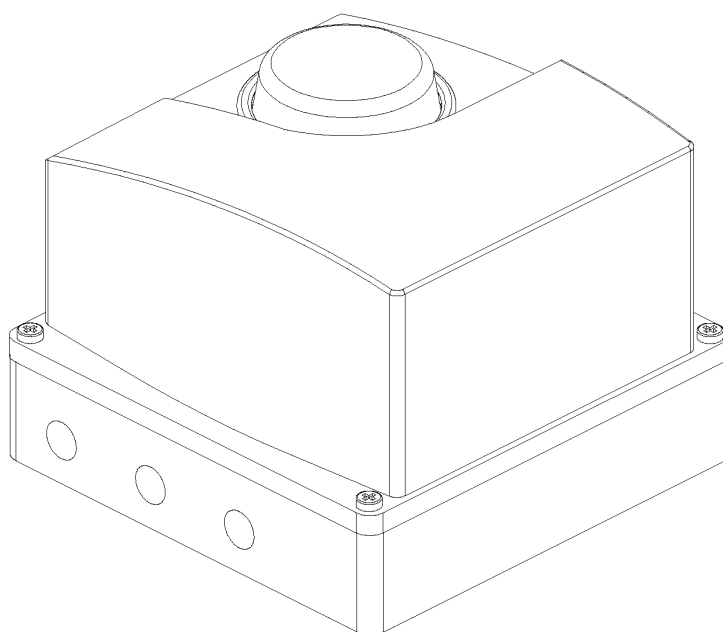




FOUNDATION FIELDBUS
VALVE POSITIONER
Series ND800FF
USER'S GUIDE v1.00

Table of Contents

1	INTRODUCTION	4
1.1	Abbreviations	4
1.2	Referenced documents	4
2	DD INSTALLATION	5
3	CONFIGURATION	6
3.1	Setting identification information	6
3.2	Positioner configuration	6
3.3	Control application configuration	6
4	FUNCTION BLOCKS	8
4.1	Parameters common to all blocks	8
4.2	Resource block	12
4.2.1	Overview	12
4.2.2	Parameters	12
4.2.2.1	General	12
4.2.2.2	Device Control	14
4.2.2.3	Device settings	14
4.2.2.4	Remote settings	15
4.2.2.5	Fault state	15
4.2.2.6	Alerts	15
4.2.3	Parameter access table	18
4.3	Transducer block	19
4.3.1	Overview	19
4.3.2	Parameters	19
4.3.2.1	General	19
4.3.2.2	Servo control	20
4.3.2.3	Valve assembly	22
4.3.2.4	Flow characterization	23
4.3.2.5	Calibration	24
4.3.2.5.1	Automatic travel calibration	24
4.3.2.5.2	Position sensor calibration	24
4.3.2.5.3	Pressure sensor calibration	24
4.3.2.5.4	Temperature measurement calibration	24
4.3.2.5.5	Position sensor temperature compensation	25
4.3.2.6	Valve diagnostics	25
4.3.2.6.1	General	25
4.3.2.6.2	Travel counters	26
4.3.2.6.3	Load factor trend	26
4.3.2.6.4	Travel deviation trend	27
4.3.2.6.5	Valve travel vs. time trend	28
4.3.2.6.6	Event log	28
4.3.2.7	Valve test	28
4.3.2.8	Alerts	30
4.3.3	Parameter access Table	31
4.4	Analog Output function block	33
4.4.1	Overview	33
4.4.2	Parameters	34
4.4.2.1	General	34
4.4.2.2	Process	34
4.4.2.3	Remote	35
4.4.2.4	Options	36
4.4.2.5	Scaling	36

Table of Contents

4.4.2.6	Limits	36
4.4.2.7	Fault state	37
4.4.2.8	Alerts	37
4.4.3	Parameter access table	38
4.5	PID controller function block	39
4.5.1	Overview	39
4.5.2	Parameters	40
4.5.2.1	General	40
4.5.2.2	Process	41
4.5.2.3	Remote	42
4.5.2.4	Options	43
4.5.2.5	Scaling	43
4.5.2.6	Limits	44
4.5.2.7	Feed forward control	44
4.5.2.8	Output tracking	44
4.5.2.9	Tuning	45
4.5.2.10	Alerts	47
4.5.3	Parameter access table	49
5	PARAMETER ATTRIBUTE DEFINITIONS	51

1 Introduction

1 Introduction

Fieldbus is not a replacement for 4-20 mA or Intelligent/Smart Transmitter Protocols, it provides much more. Fieldbus is a complete Control System Architecture enabling distribution of the control function to equipment in the field, it is therefore a replacement for the DCS Architecture of the 1970s.

One of the major advantages of Fieldbus is interoperability. The blocks described in this manual are used not only by devices from Metso Automation, but other Foundation Fieldbus devices too. No particular configurator is addressed in this manual, because the devices are independent of configurator due to the DD technology.

To achieve the desired control the devices must be installed, calibrated and configured. The configuration includes also the building of the control strategy. Installation instructions are presented in the ND800FF IMO /1/. ND800FF User's Guide covers the calibration and configuration.

This manual applies to following ND800FF revision (Resource block parameters):

MANUFAC_ID	Metso Automation (= 3589 = 0x0e05)
DEV_TYPE	800 (= 0x0320)
DEV_REV	03
DD_REV	02

ND800FF is designed according to Fieldbus Foundation specifications and IEC 1158-2.

Get the best results of the Fieldbus System by carefully reading these instructions.

1.1 Abbreviations

DD	Device Description
DDL	Device Description Language
AO	Analog Output
PD	Proportional Derivative
PID	Proportional Integral Derivative
SP	Setpoint
PV	Process Variable
FF	Fieldbus Foundation
MMI	Man Machine Interface
SCADA	Supervisory Control And Data Acquisition
SW	Software
HW	Hardware

1.2 Referenced documents

/1/ ND800FF Installation, Maintenance and Operation instructions (IMO).

2 DD Installation

Device Description (DD) is a set of three binary files that contains information about the field device. DD holds parameter descriptions, menus and calibration routines (methods). The host software i.e. Configurator, MMI, SCADA and OPC servers utilize the DD when communicating with the field device.

Each host software always has a certain directory called "release directory" which holds the DDs for each field device. If the ND800FF DD is not shipped with the host sw DD library, then the DD must be installed afterwards. Installation can be done in two ways. Most configurators have a special tool for this job, see the host sw documentation. The DD files can also just be copied from the shipped CD to the host sw release directory.

1. Locate the host sw release directory. It contains folders alike those in figure 1.
2. Drag and drop the 000e05 directory folder from the shipped CD to the release directory.

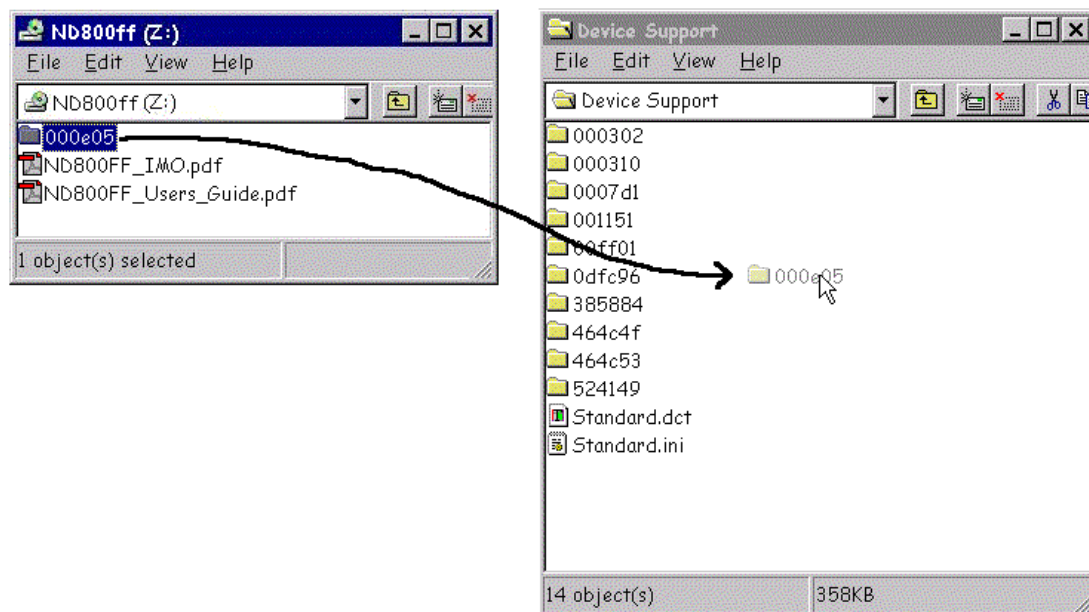


Figure 1. DD installation.

3 Configuration

This chapter describes the basic steps how to set up a basic control loop.

3.1 Setting identification information

Each fieldbus device must have a unique network address and physical device tag on the fieldbus before the communication can become operational.

Assignment of network address is performed automatically by the System management. In this case Configuration master (normally collocated with the Link Active Scheduler, LAS) specifies free address for a new fieldbus device. Default address for a new ND800FF is 248 (0xF8). The network address can also be manually changed by using the “Set Address” service of the fieldbus configuration software.

A new ND800FF positioner has default physical device tag defined as “Neles_Valve-1”. This tag can be manually changed by using the “Set tag” service of a fieldbus configuration software.

The function block tags must be unique across the fieldbus network before the control application can become operational. Default block tags in a new ND800FF positioner are: ResourceBlock for resource block, TransducerBlock for transducer block, AOBlock for Analog Output block and PIDBlock for Proportional Integral Derivative block. Block tags can be manually changed by using the “Set tag” service of the fieldbus configuration software.

3.2 Positioner configuration

The positioner configuration basic steps are:

1. Adjust the Assembly related parameters. See page 22.
2. Execute the Automatic travel calibration. See page 24.
3. Adjust the Transducer characterization, cutoff and limits. See pages 20 and 23.
4. Set the target modes of Resource and Transducer block to Auto.
5. Check the Resource and Transducer block BLOCK_ERR. If the block actual mode does not go to the target mode, BLOCK_ERR reports the reason for that.

3.3 Control application configuration

The control application configuration basic steps are:

1. Set the PID and AO signal scalings and limits.
2. Set the PID and AO parameters affecting to operation in case of abnormal situation i.e. PID CONTROL_OPTS, AO IO_OPTS, PID & AO STATUS_OPTS and SHED_OPT. See also resource block parameters SHED_RCAS and SHED_ROUT.
3. Set the PID and AO alarm limits and priorities.
4. Tune the PID controller. See page 45.
5. Build the control application. Link the block inputs and outputs, alarms and trends. See the fieldbus configurator sw documentation. Two basic control loops are presented in figures 2 and 3.
6. Download the function block schedule. See the fieldbus configurator sw documentation.
7. Set the target mode of AO to Cascade and PID to Auto.
8. Check the PID & AO BLOCK_ERR. If the block actual mode does not go to the target mode, BLOCK_ERR reports the reason for that.

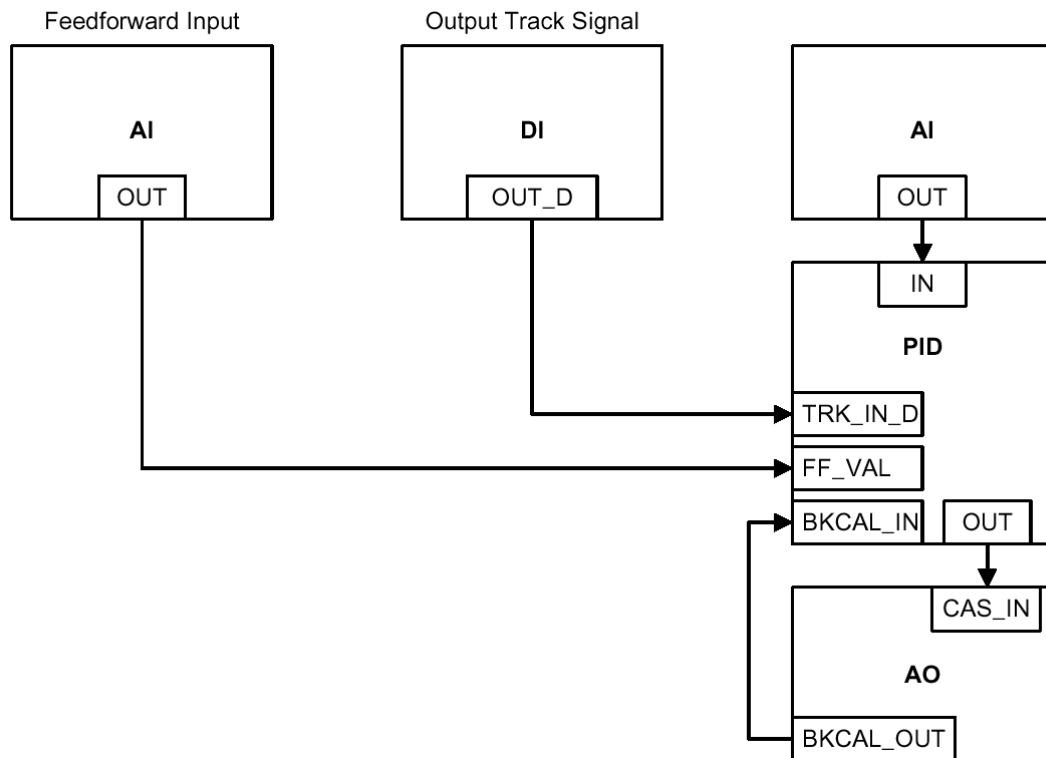


Figure 2. Feedback control loop with feedforward and track capability.

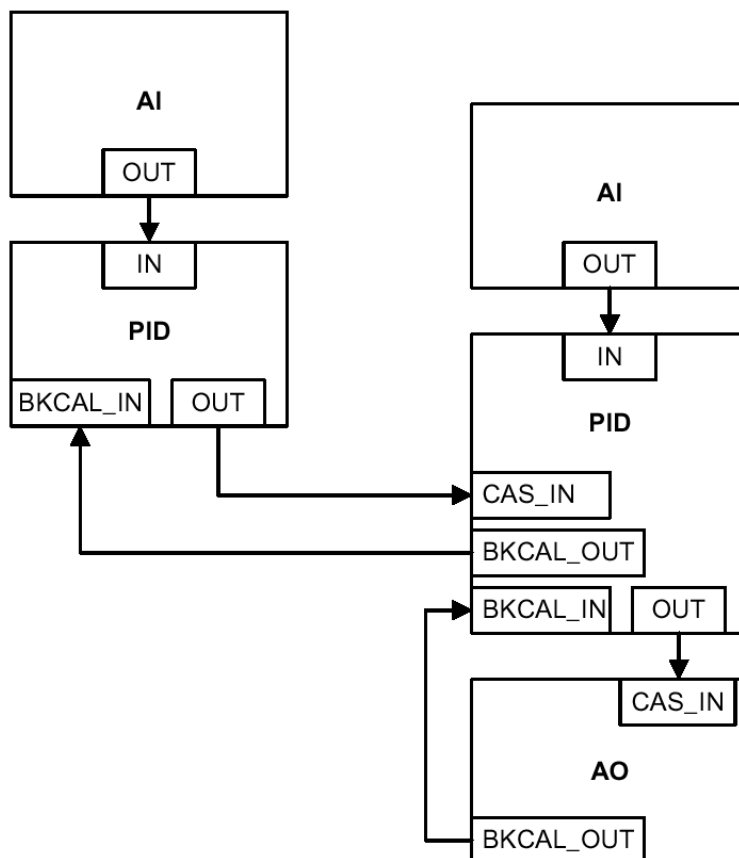


Figure 3. Cascade control loop.

4 Function blocks

Function blocks provide a general structure for specifying different types of device functions. ND800FF has four function blocks, Resource block, Transducer block, AO function block and PID control block. The functionality of these blocks is described in this chapter.

Block parameters are presented in the same order as they appear in the configurator software, if the configurator supports the DD menus.

4.1 Parameters common to all blocks

GRANT_DENY

The grant-deny parameter is used to allow the operator to grant and deny access permission to sets of function block parameters by other devices. The parameter has two attributes named Grant and Deny. The operation of these parameters is defined here, but the actual usage (if any) depends on the philosophy of the plant.

Grant - Depending on the philosophy of the plant, the operator or a higher level device (HLD), or a local operator's panel (LOP) in the case of Local, may turn on an item of the Grant attribute - Program, Tuning, Alarm, or Local. By doing or allowing this action, the operator gives up control of the selected parameters to the HLD or LOP. The function block does not check writes to any of the selected parameters for grant-deny permission. It is up to other devices to obey and enforce the rules, because the function block has no way to know who is writing to it. When the operator wants to regain control of the parameters, he clears the Grant item. The function block will then automatically set the corresponding Denied item. This indicates to the HLD or LOP that control has been taken away.

- Program - A higher level device may change the target mode, setpoint (if the block mode is Man or Auto), or output (if the block mode is Man) of the block.
- Tune - A higher level device may change the tuning parameters of the block.
- Alarm - A higher level device may change the alarm parameters of the block.
- Local - A local operator's panel or hand-held device may change the target mode, setpoint (if the block mode is Man or Auto), or output (if the block mode is Man) of the block.

Deny - The Denied attribute is provided for use by a monitoring application in an interface device and may not be changed by an operator. It allows the monitoring application to determine if control has been temporarily taken away during the execution of a batch program. This is done by first clearing one or all of the Denied items before execution of a batch program, then checking the Denied item after execution. The Grant item itself should not be checked for this condition, because the operator may have cleared and subsequently set the Grant item during batch program execution, a sequence that might be missed by a slowly scanning monitor program. The Denied item may not be cleared by the operator, thus latching the fact that control was taken away.

- Program - The Program permission item has been turned off.
- Tune Denied - The Tune permission item has been turned off.
- Alarm Denied - The Alarm permission item has been turned off.
- Local Denied - The Local permission item has been turned off.

MODE_BLK contains the actual, target, permitted, and normal modes of the block.

- ACTUAL is the current mode of the block, which may differ from the target based on operating conditions. Its value is calculated as part of block execution.
- TARGET is the mode requested by the operator. Only one mode from those allowed by the permitted mode parameter may be requested.
- PERMITTED defines the modes which are allowed for an instance of the block. The permitted mode is configured based on application requirement.
- NORMAL is the mode which the block should be set to during normal operating conditions.

IN, OUT, BKCAL_IN, BKCAL_OUT, RCAS_IN, RCAS_OUT, ROUT_IN, ROUT_OUT, SP, PV, READBACK, FINAL_VALUE, FINAL_POSITION_VALUE, FF_VAL, TRK_VAL, TRK_IN_D

All input and output parameters are structures composed of status and value, but some contained parameter (internal parameter, not accessible by other blocks) have also that data type, for example, RCAS_IN, ROUT_IN, SP and PV. The Status field is composed of three parts: Quality, Sub-Status and Limits.

Quality – It indicates the quality of the parameter value.

- Good Cascade – The quality of the value is good, and it may be part of a cascade structure.
- Good Non-Cascade – The quality of the value is good, and the block doesn't support a cascade path.
- Uncertain – The quality of the value is less than normal, but the value may still be useful.
- Bad – The value is not useful.

Sub-Status – The sub-status is a complement of the quality status and takes information to initialize or break a cascade control, alarms and others. There are different sets of sub-status for each quality.

Limits – It provides information whether the associated value is limited or not, as well the direction. The limits are classified as : *Not Limited, High Limited, Low Limited, Constant*.

When an input parameter is linked to an output parameter through the link object, the whole structure (status and value) is copied (local link) or received from the bus (external link). If the input is not linked, then the status may be set manually by the user, as well the value.

PV_SCALE, XD_SCALE, OUT_SCALE, FF_SCALE, TRK_SCALE

The high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the PV parameter and parameters which have the same scaling as PV.

- EU_100; The engineering unit value which represents the upper end of range of the associated block parameter.
- EU_0; The engineering unit value which represents the lower end of range of the associated block parameter.
- UNITS_INDEX; Units code index for the engineering unit descriptor for the associated block value.
- DECIMAL; The number of digits to the right of the decimal point which should be used by an interface device in displaying the specified parameter.

BLOCK_ERR

This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown.

- Other; Non-specific error active.
- BlockConfiguration; Error detected in block configuration.
- LinkConfiguration; Error detected in link configuration.
- SimulationActive; Simulation enabled in this block. For the resource block, Simulate Active will be used to indicate that the simulate hardware jumper is present. An active state (1) of this attribute will indicate that the jumper is present and that it is possible for the user to enable simulation in the AO function block. For AO block this indicates either the simulation is enabled or disabled.
- LocalOverride; Output tracking or faultstate active.
- DeviceFaultstate; Device faultstate set.
- Device needs maintenance soon
- InputFailure; Process variable has bad status.
- OutputFailure; Failure detected in output hardware.
- MemoryFailure; Memory error detected.
- LostStaticData; Static parameters cannot be recovered.
- LostNVData; Non-Volatile parameters cannot be recovered.
- ReadbackCheck; Failure detected in READBACK.
- Device needs maintenance now
- PowerUp; Recovery from power failure.
- OutOfService; Block actual mode is Out of Service.

BLOCK_ALM

The block alarm is used for all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.

- Unacknowledged - A discrete enumeration which is set to Unacknowledged when an update occurs, and set to Acknowledged by a write from a human interface device or other entity which can acknowledge that the alarm has been noticed.
- Update State - A discrete enumeration which gives an indication of whether the alert has been reported.
- Time Stamp - The time when evaluation of the block was started and a change in alarm/event state was detected that is unreported. The time stamp value will be maintained constant until alert confirmation has been received - even if another change of state occurs.
- Subcode - An enumeration specifying the cause of the alert to be reported. Enumerations are equal with the BLOCK_ERR.
- Value - The value of the associated parameter at the time the alert was detected.

UPDATE_EVT

An alert for any change in the static data, UPDATE_EVT, is included in each block. This alert can notify interface devices that keep track of changes that one or more changes have occurred. The relative parameter index and its associated block index is included in the alert, along with the new value of ST_REV. If more than one change was added since the last reported Update Alert, as known from the difference between the last copy of ST_REV and the one in the alert, it will be necessary for the interface device to update all static data. No alert will be generated while a block is in Out of Service mode, so that downloads will not generate many update alerts. ST_REV will be incremented for each change to static data that occurs while the block is in the O/S mode. On the transition out of O/S mode, an update alert may be generated if the value of ST_REV for the block does not match that of the last reported alert. Update Alert has a fixed priority of 2, therefore it is auto-acknowledged (no operator intervention is required).

- Unacknowledged - A discrete enumeration which is set to Unacknowledged when an update occurs, and set to Acknowledged by a write from a human interface device or other entity which can acknowledge that the alarm has been noticed.
- Update State - A discrete enumeration which gives an indication of whether the alert has been reported.
- Time Stamp - The time when evaluation of the block was started and a change in alarm/event state was detected that is unreported. The time stamp value will be maintained constant until alert confirmation has been received - even if another change of state occurs.
- Static Rev - the static revision of the block whose static parameter was changed and is being reported. It is possible for the present value of static revision to be greater than this because static can be changed at any time.
- Relative Index - The OD index of the static parameter whose change caused this alert, minus the FB starting index. If the update event was caused by a write to multiple parameters at the same time, then this attribute will be zero.

SHED_OPT

Defines action to be taken on remote control device timeout. See SHED_RCAS and SHED_ROUT, page 15.

- Undefined - Invalid
- Normal shed, normal return - Actual mode changes to the next lowest priority non-remote mode permitted but returns to the target remote mode when the remote computer completes the initialization handshake.
- Normal shed, no return - Target mode changes to the next lowest priority non-remote mode permitted. The target remote mode is lost, so there is no return to it.
- Shed to Auto, normal return - Actual mode changes to Auto on detection of a shed condition.
- Shed to Auto, no return - Target mode changes to Auto on detection of a shed condition.
- Shed to Manual, normal return - Actual mode changes to Man on detection of a shed condition.
- Shed to Manual, no return - Target mode changes to Man on detection of a shed condition.
- Shed to Retained target, normal return - Shed to previous target mode and return target remote mode after communications are re-established.
- Shed to Retained target, no return - Target mode changes to retained target mode.

4.2 Resource block

4.2.1 Overview

This block contains data that is specific to the hardware that is associated with the resource. All data is modeled as Contained, so there are no links to this block. The data is not processed in the way that a function block processes data, so there is no function schematic.

The mode is used to control major states of the resource. O/S mode stops all function block execution. The actual mode of the function blocks will be changed to O/S, but the target mode will not be changed. Auto mode allows normal operation of the resource. IMan shows that the resource is initializing or receiving a software download.

The RS_STATE parameter contains the operational state of the Function Block Application for the resource containing this resource block.

The RESTART parameter allows degrees of initialization of the resource. They are Run (1), Restart resource (2), Restart with defaults (3), and Restart processor (4).

SHED_RCAS and SHED_ROUT set the time limit for loss of communication from a remote device. These constants are used by each function block.

The parameter HARD_TYPES is a read only bitstring that indicates the types of hardware available as channel numbers.

Parameters MANUFAC_ID, DEV_TYPE, DEV_REV, DD_REV, and DD_RESOURCE are required to identify and locate the DD so that host sw can select the correct DD for use with the resource.

The MAX_NOTIFY parameter value is the maximum number of alert reports that this resource can have sent without getting a confirmation, corresponding to the amount of buffer space available for alert messages. A user can set the number lower than that, to control alert flooding, by adjusting the LIM_NOTIFY parameter value. If LIM_NOTIFY is set to zero, then no alerts are reported. The CONFIRM_TIME parameter is the time for the resource to wait for confirmation of receipt of a report before trying again.

4.2.2 Parameters

4.2.2.1 General

MANUFAC_ID

Manufacturer identification number - used by an interface device to locate the DD file for the resource.

DEV_TYPE

Manufacturer's model number associated with the resource - used by interface devices to locate the DD file for the resource.

DEV_REV

Manufacturer revision number associated with the resource - used by an interface device to locate the DD file for the resource.

DD_REV

Revision of the DD associated with the resource - used by an interface device to locate the DD file for the resource.

HARDWARE_REVISION

Hardware revision of the device.

SOFTWARE_REVISION

Software revision of the device.

ITK_VER

Major revision number of the interoperability test case used in certifying this device as interoperable. The format and range of the version number is defined and controlled by the Fieldbus Foundation. Note: The value of this parameter will be zero (0) if the device has not been registered as interoperable by the FF.

DEVICE_PCB_SN

Device printed circuit board serial number.

ST_REV

The revision level of the static data associated with the function block. The revision value will be incremented each time a static parameter value in this block is changed.

TAG_DESC

The user description of the intended application of the block.

STRATEGY

The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.

GRANT_DENY

See paragraph "Parameters common to all blocks", page 8.

DD_RESOURCE

String identifying the tag of the resource which contains the Device Description for this device.

HARD_TYPES

The types of hardware available as channel numbers.

MEMORY_SIZE

Available configuration memory in the empty resource. To be checked before attempting a download.

FREE_SPACE

Percent of memory available for further configuration. Zero in a preconfigured device.

FREE_TIME

Percent of the block processing time that is free to process additional blocks.

TEST_RW

Read/write test parameter - used only for conformance testing.

PPID_LICENCE

PPID licence number. Valid licence number enables the use of the PID controller block as a Predictive PID (PPID) controller.

4.2.2.2 Device Control

MODE_BLK

See paragraph "Parameters common to all blocks", page 8.

Resource block modes are:

- Out of Service (O/S) - O/S mode stops all function block execution. The actual mode of the function blocks in this resource will be changed to O/S, but the target mode will not be changed.
- Initialization Manual (IMan) - IMan shows that the resource is initializing or receiving a software download.
- Automatic (Auto) - Auto mode allows normal operation of the resource.

RESTART

Allows a manual restart to be initiated. Several degrees of restart are possible:

- Run - This is the passive state of the parameter.
- Restart resource - This restarts the device.
- Restart with defaults - This restarts the device with defaults. See PAR_DEFAULTS_TYPE.
- Restart processor - This restarts the processor.

PAR_DEFAULTS_TYPE

Parameter default values selection.

- FF Spec defaults; Default values defined in FF Specification. When these values are used, several parameters in resource block and other blocks must be configured before the device is ready for use. If the block configuration is not performed completely, the BLOCK_ERR "Configuration error" is reported.
- Metso Automation defaults; When these default values are used, very little configuration is needed. All parameters are initialized to legal values and Configuration error situation is avoided.

4.2.2.3 Device settings

CYCLE_TYPE

Identifies the block execution methods available for this resource.

CYCLE_SEL

Used to select the block execution method for this resource.

MIN_CYCLE_T

Time duration of the shortest cycle interval of which the resource is capable.

NV_CYCLE_T

Minimum time interval specified by the manufacturer for writing copies of NV parameters to non-volatile memory. NV memory is updated only if there has been a significant change in the dynamic value. The last value saved in NV memory will be available after restart. Zero means that NV data will only be copied to NV memory when an external write request is received.

FEATURES

Used to shows supported resource block options.

FEATURE_SEL

Used to select resource block options.

- Reports - If set, the device supports alert reports. If it is not set, the master must poll for alerts.
- Faultstate - If set, setting the SET_FSTATE parameter will force all output function blocks (AO and DO) in the resource to go to fault state. Individual output function block will go to Fault State due to a loss of communication to CAS_IN or IFS status in CAS_IN, regardless the selection of this feature.
- Hard W Lock - This enables the use of the hardware write lock switch. See WRITE_LOCK.
- Out Readback - If set, the AO.READBACK (valve position) runs backwards through the XD scaling to act as the PV for the AO block. If not set, READBACK is generated from AO.OUT. The OUT and READBACK parameters both use XD_SCALE. The PV and SP use PV_SCALE.

WRITE_LOCK

- Shows the state of the hardware switch. If set, no writes from anywhere are allowed. Block inputs will continue to be updated. See /1/ for the location of the switch. See FEATURE_SEL.

SIMULATION_SWITCH

This parameter shows the state of the HW simulation switch, which enables/disables the AO block simulation function. See /1/ for the location of the switch.

4.2.2.4 Remote settings

SHED_RCAS and SHED_ROUT set the time limit for loss of communication from a remote device. These constants are used by all function blocks that support a remote cascade mode. Shedding from RCAS/ROUT shall not happen when SHED_RCAS or SHED_ROUT is set to zero. See paragraph "Parameters common to all blocks", SHED_OPT, page 11.

4.2.2.5 Fault state

FAULT_STATE

Condition set by loss of communication to an output block, failure promoted to an output block or a physical contact. When faultstate condition is set, then output function blocks will perform their FSTATE actions.

SET_FSTATE

Allows the faultstate condition to be manually initiated by selecting Set. See FEATURE_SEL, page 15.

CLR_FSTATE

Writing a Clear to this parameter will clear the device faultstate state if the field condition, if any, has cleared.

4.2.2.6 Alerts

To enable alert reporting, select FEATURE_SEL "Reports", page 15.

ALERT_KEY

The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.

MAX_NOTIFY

Maximum number of unconfirmed alert notify messages possible.

LIM_NOTIFY

Maximum number of unconfirmed alert notify messages allowed.

CONFIRM_TIME

The minimum time between retries of alert reports.

ACK_OPTION

Selection of whether alarms associated with the function block will be automatically acknowledged.

RS_STATE

The overall state of the function block application state machine.

- Undefined - Invalid state.
- Start/Restart - This state will be entered after detection that power has been restored to a device. In this state, the memory and other hardware necessary for reliable operation will be tested. An important part of the recovery process is being able to restore static data, which includes both the “static” and “non-volatile” types of parameters. The validity of static memory will be tested. If the object’s static data is bad, then the object’s database will be set to its default values. A block will should be issued, with the subcode set to either “Lost static data” or “Lost NV data” as appropriate. After successfully initializing, the associated resource block should generate a block alarm with the subcode set to “Power-up”. If the hardware tests are successful, the resource state will transition to the initialization state. Otherwise, resource state will transition to the Failure state.
- Initialization - The initialization state is entered from the Start/Restart or Failure states. In the Initialization state, all unreported function block alarms will be automatically confirmed and acknowledged. Once the system is detected to be Operational, block execution may be scheduled and the resource state will transition to On-Line Linking.
- On-line Linking - This state will be entered from the On-Line and Initialization state. In this state, the status of defined links will be evaluated. If all defined links are established, then the resource state will transition to On-Line.
- On-line - The On-Line state will be entered from the On-Line Linking state. In this state, the status of defined links will be evaluated. If one or more defined links are detected as not established, then the resource state will transition to On-Line Linking
- Standby - This state will be entered if the mode of the resource block is changed to Out-of-Service (O/S). In this state the actual mode of all function blocks in the resource will be forced to O/S mode. The mode of transducer blocks may not be affected. This state will be maintained until the mode is changed to Auto. On a change in the resource block mode to Auto, the state will transition to Start/Restart.
- Failure - This state may be entered from any state except Standby. Transition to this state is caused by the detection of a memory or other hardware failure which would prevent reliable operation. The failure may pertain either to the whole device or only to the resource. Based on this state being active, a function block of the output class may change its output to a Fault State position. In this state, hardware status will be tested. If the hardware failure clears, then the state will transition to Initialization.

BLOCK_ERR

See paragraph “Parameters common to all blocks”, page 10.

BLOCK_ALM

See paragraph “Parameters common to all blocks”, page 10.

UPDATE_EVT

See paragraph “Parameters common to all blocks”, page 11.

WRITE_PRI

Priority of the alarm generated by clearing the write lock.

- 0 = the associated alert may clear when the priority is changed to 0, but it will never occur.
- 1 = the associated alert is not sent as a notification. If the priority is above 1, then the alert must be reported.
- 2 = reserved for alerts that do not require the attention of a plant operator, e.g. diagnostic and system alerts. Block alarm, error alarm, and update event have a fixed priority of 2.
- 3-7 = increasing higher priorities - advisory alarms.
- 8-15 = increasing higher priority - critical alarms.

WRITE_ALM

This alert is generated if the write lock parameter is cleared.

- Unacknowledged - A discrete enumeration which is set to Unacknowledged when an update occurs, and set to Acknowledged by a write from a human interface device or other entity which can acknowledge that the alarm has been noticed.
- Alarm State - A discrete enumeration which gives an indication of whether the alert has been reported.
- Time Stamp - The time when evaluation of the block was started and a change in alarm/event state was detected that is unreported. The time stamp value will be maintained constant until alert confirmation has been received - even if another change of state occurs.
- Subcode - An enumeration specifying the cause of the alert to be reported. Enumerations are equal with the BLOCK_ERR.
- Discrete Value - The value of the associated parameter at the time the alert was detected.

ALARM_SUM

The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.

- Current - The active status of each alarm.
- Unacknowledged - The unacknowledged state of each alarm.
- Unreported - The unreported status of each alarm.
- Disabled - The disabled state of each alarm.

4.2.3 Parameter access table

View object is a package which contains a predefined set of parameters. It is more efficient to read view objects from the device than polling the parameters separately. There are four different types of view objects

- VIEW_1 - View object defined to access the dynamic operating parameters of a block.
- VIEW_2 -View object defined to access the static operating parameters of a block.
- VIEW_3 - View object defined to access all dynamic parameters of a block. More than one View 3 object may be required to represent all dynamic parameters of the block.
- VIEW_4 - View object defined to access static parameters not included in VIEW_2. More than one VIEW_4 object may be required to represent these other static parameters of the block.

Table 1. Resource block parameter access table.

Index	Parameter Mnemonic	VIEW_1	VIEW_2	VIEW_3	VIEW_4	VIEW_4.1
1	ST_REV	2	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	RS_STATE	1		1		
8	TEST_RW					
9	DD_RESOURCE					
10	MANUFAC_ID				4	
11	DEV_TYPE				2	
12	DEV_REV				1	
13	DD_REV				1	
14	GRANT_DENY		2			
15	HARD_TYPES				2	
16	RESTART					
17	FEATURES				2	
18	FEATURE_SEL		2			
19	CYCLE_TYPE				2	
20	CYCLE_SEL		2			
21	MIN_CYCLE_T				4	
22	MEMORY_SIZE				2	
23	NV_CYCLE_T		4			
24	FREE_SPACE		4			
25	FREE_TIME	4		4		
26	SHED_RCAS		4			
27	SHED_ROUT		4			
28	FAULT_STATE	1		1		
29	SET_FSTATE					
30	CLR_FSTATE					
31	MAX_NOTIFY				1	
32	LIM_NOTIFY		1			
33	CONFIRM_TIME		4			
34	WRITE_LOCK		1			
35	UPDATE_EVT					
36	BLOCK_ALM					
37	ALARM_SUM	8		8		
38	ACK_OPTION				2	
39	WRITE_PRI				1	
40	WRITE_ALM					
41	ITK_VER				2	
42	PAR_DEFAULTS_TYPE					1
43	SIMULATION_SWITCH			1		
44	PPID_LICENCE					32
45	HARDWARE_REVISION					32
46	SOFTWARE_REVISION					32
47	DEVICE_PCB_SN					4
	Totals (bytes)	22	30	23	31	103

4.3 Transducer block

4.3.1 Overview

Transducer block contains the valve positioner user interface. Transducer block schematic is presented in figure 4.

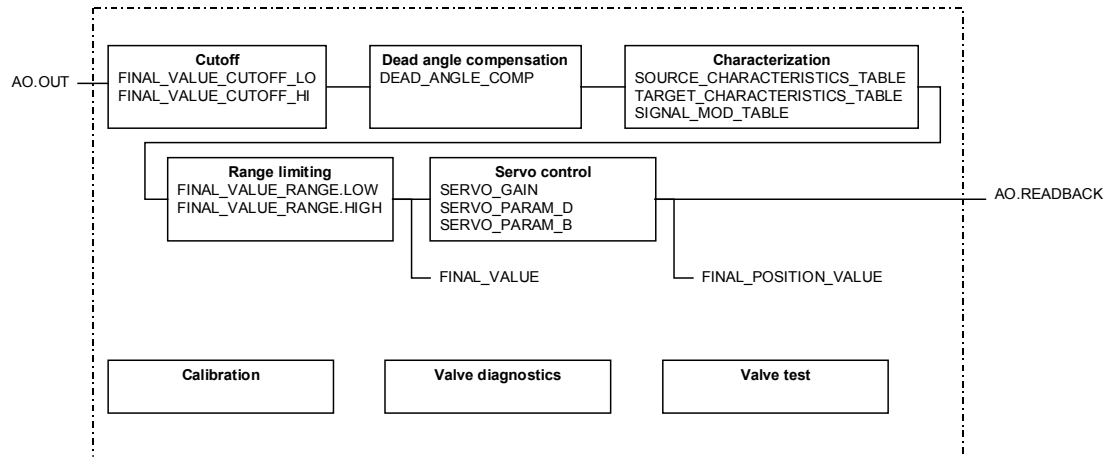


Figure 4 Transducer block schematic.

The transducer block input is the AO.OUT signal. This signal can be modified using the Cutoff, Dead angle compensation, Characterization and Range limiting functions.

FINAL_VALUE is the final valve position setpoint after all signal modifications. FINAL_POSITION_VALUE is the measured valve position, which is fed back to AO block as AO.READBACK signal.

The transducer block includes also the positioner Calibration, Valve diagnostics and Valve test functions.

4.3.2 Parameters

4.3.2.1 General

ST_REV

The revision level of the static data associated with the function block. The revision value will be incremented each time a static parameter value in this block is changed.

TAG_DESC

The user description of the intended application of the block.

STRATEGY

The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.

TRANSDUCER_DIRECTORY

A directory that specifies the number and starting indices of the transducers in the transducer block.

TRANSDUCER_TYPE

Identifies the transducer.

COLLECTION_DIRECTORY

A directory that specifies the number, starting indices, and DD Item IDs of the data collections in each transducer within a transducer block.

ACT_MAN_ID

The actuator manufacturer identification.

ACT_MODEL_NUMBER

The actuator model number.

ACT_SN

The actuator serial number.

VALVE_MAN_ID

The valve manufacturer identification.

VALVE_MODEL_NUMBER

The valve model number.

VALVE_SN

The valve serial number.

4.3.2.2 Servo control

MODE_BLK

See paragraph "Parameters common to all blocks", page 8.

Transducer block modes are:

- Out of Service (O/S) - The block is not being evaluated. The FINAL_VALUE maintained at last value.
- Manual (Man) - Manual mode indicates that the device cannot follow the AO.OUT. Manual mode is entered when the target mode is Auto and either MAN mode is selected from LUI or calibration is active or device failure is active.
- Automatic (Auto) - Normal operation, device follows the AO.OUT signal.

SERVO_GAIN

The servo controller gain value. This parameter is adjusted automatically during the Automatic Travel Calibration. After travel calibration the value is 1.0. For enhanced control use value 1.2.

SERVO_PARAM_D

Servo controller tuning parameter D.

- 0.0 for double acting actuators (factory set)
- 1.0 for single acting actuators (factory set)

SERVO_PARAM_B

Servo controller tuning parameter B. This parameter is adjusted automatically during the Automatic Travel Calibration. After travel calibration the value is 1.0.

The signal modifications accomplished using Dead angle compensation, Cut-Off and Range limit functions are presented in figure 5.

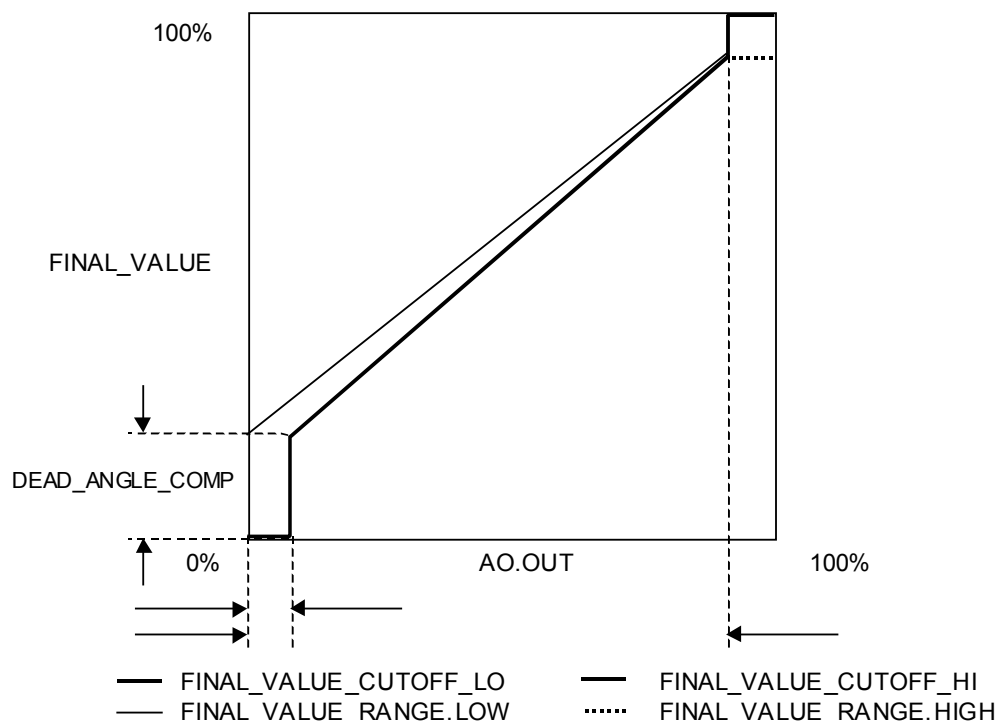


Figure 5. Dead angle compensation, Cut-Off and Range limit functions.

NOTE:

Setpoint Cut-Off function overrides the Limit function. If `FINAL_VALUE_CUTOFF_LO` is greater than 0%, the `FINAL_VALUE_RANGE.LOW` is not applied (has value of 0%). Correspondingly, if `FINAL_VALUE_CUTOFF_HI` is smaller than 100%, the Upper `FINAL_VALUE_RANGE.HIGH` is not applied (has value of 100%).

FINAL_VALUE_RANGE

- HIGH - Valve position upper limit.
- LOW - Valve position lower limit.
- `UNITS_INDEX`; Units code index for the engineering unit descriptor for the associated block value.
- `DECIMAL`; The number of digits to the right of the decimal point which should be used by an interface device in displaying the specified parameter.

FINAL_VALUE_CUTOFF_HI

If the `FINAL_VALUE` is more positive than this value, the valve is forced to its maximum high value (calibrated 100 %).

FINAL_VALUE_CUTOFF_LO

If the `FINAL_VALUE` is more negative than this value (default 2 %), the valve is forced to its maximum low value (fully closed against mechanical limit).

FINAL_VALUE

The final servo control setpoint after all signal modifications. Normally this value is calculated from the `AO.OUT` but while `AO` block simulation is active, this value can be written by the user. See paragraph "Parameters common to all blocks", page 9.

FINAL_POSITION_VALUE

The measured valve position, which is fed back to `AO` block as `AO.READBACK` signal. See paragraph "Parameters common to all blocks", page 9.

4.3.2.3 Valve assembly

NOTE:

These parameters are valve assembly related; you should thus check that the control valve assembly conforms to the setting of these parameters.

VALVE_TYPE

Valve type selection. Options are

- Rotary
- Linear

ACT_TYPE

Actuator type selection. Options are

- Undefined
- Single acting actuator
- Double acting actuator

SUPPLY_PRESSURE

Set this parameter to the average value of the positioner supply pressure. Supply pressure is used to calculate the load factor.

POSITIONER_FAIL_ACTION

Configuration of the action taken during the LOSS OF SUPPLY POWER (supply pressure is available). This action takes place ALSO when the positioner software notices a fatal device failure. In these both cases the spool valve feeds C1 (pneumatic connector) and releases C2 pressure. See /1/ how to set this parameter. Options are

- Close
- Open

ACT_FAIL_ACTION

Actuator fail safe position during LOSS OF SUPPLY PRESSURE. This parameter is informational only. Options are.

- Undefined (Double acting actuator)
- Self-Closing
- Self-Opening

POS_SENSOR_ROT

Defines relationship between the position sensor rotation and valve action. If the position sensor rotates clockwise when the valve closes, set Standard: clockwise to close. With inverse rotation set Nonstandard: counter-clockwise to close. See /1/ how to set this parameter.

DEAD_ANGLE_COMP

This adjustment is used to change the valve travel from fully closed to a desired travel with a small change in input signal near the 0% value of the input signal range. This can be used to compensate dead angle in a ball or segment valve such that it is equal to the a_0 adjustment in conventional Neles positioners. The a_0 adjustment is used especially with small ball valves which require a turn of several degrees from the closed position before any flow through the valve is perceptible. Suitable a_0 values for Metso Automation segment and ball valves can be found in /1/. See figure 5, page 21.

4.3.2.4 Flow characterization

When optimizing process control loop performance a linear relationship between the flow and valve position setpoint signal should be established. This can be achieved by using this flow characterization design tool.

Valve *inherent* flow characteristics curve tells the (normalized) valve capacity (effective flow cross-section) versus the valve position. This curve is specific to the physical valve design.

Valve *installed* flow characteristics curve tells the (normalized) flow versus the valve position. This curve is the result of the inherent flow characteristics curve and the process pressure behavior.

The flow characterization design tool changes the valve inherent flow characteristics curve to any desired curve. **SOURCE_CHARACTERISTICS_TABLE** tells the flow characteristics without any signal modifications. **TARGET_CHARACTERISTICS_TABLE** tells the desired characteristics. Based on this information, the tool calculates the required signal modification **SIGNAL_MOD_TABLE**, which is as well user editable.

SOURCE_CHARACTERISTICS_TABLE

This table tells the inherent flow characteristics without any signal modifications (flow characterization is off). The variable indexes of the table represent the valve opening (at range 0,5,10,15 ... 100 %) and the corresponding values are the relative flow coefficients.

TARGET_CHARACTERISTICS_TABLE

This table tells the desired inherent flow characteristics. The variable indexes of the table represent the valve opening (at range 0,5,10,15 ... 100 %) and the corresponding values are the relative flow coefficients.

FILL_SOURCE_CHARACTERISTICS_TBL & FILL_TARGET_CHARACTERISTICS_TBL

Options are:

- User defined - When this option is selected, the table can be entered by user.
- Linear - Fills the table with linear characteristic values.
- Equal percentage 1:25 - Fills the table with Equal percentage characteristic values.
- Equal percentage 1:33 - Fills the table with Equal percentage characteristic values.
- Equal percentage 1:50
- Quick opening (Equal percentage inverse 1:25)
- Quick opening (Equal percentage inverse 1:33)
- Quick opening (Equal percentage inverse 1:50)
- Neles L1-series butterfly valve - Fills the table with Neles specific characteristic values.
- Neles R-series segment valve - Fills the table with Neles specific characteristic values.
- Neles M-series ball valve - Fills the table with Neles specific characteristic values.
- Clear - Clears the table.

SIGNAL_MOD_TABLE

Signal modification table. The variable indexes of the table represent the input (at range 0,5,10,15 ... 100 %) and the corresponding values are the outputs.

CHARACTERIZATION

This parameter controls the characterization. Options are:

- Off - Shows the active state or turns the characterization off.
- On - Shows the active state or turns the characterization on.
- Calculate new Signal mod table from SOURCE and TARGET tables
- Clear Signal mod table
- Enable new Signal mod table using all table values
- Enable new Signal mod table using every second table value. This option fills the missing points using linear interpolation.

NOTE:

If the installed flow characteristics curve is known, this tool can ALTERNATIVELY be used to modify the installed characteristics. In this case all the words INHERENT in the tool must be understood as INSTALLED. The installed curve can be found by a simple process test or it can be calculated by the Nelprof software.

NOTE:

The dead angle compensation is done before the characterization.

4.3.2.5 Calibration

4.3.2.5.1 Automatic travel calibration

AUTO_TRAVEL_CALIBRATION

This calibration calibrates the position sensor range and tunes the servo control tuning parameters. This calibration must be done during the device commissioning. Options are:

- Idle - State indication.
- Start calibration - This selection starts the calibration.
- Stop calibration - This selection cancels the calibration.
- Calibration is initializing - State indication.
- Calibration is running - State indication.
- Calibration was stopped - State indication.
- Calibration was successful - State indication.
- Calibration was unsuccessful - State indication.

XD_CAL_LOC

The location of last positioner calibration. This describes the physical location at which the calibration was performed.

XD_CAL_DATE

The date of the last positioner calibration.

XD_CAL_WHO

The name of the person responsible for the last positioner calibration.

4.3.2.5.2 Position sensor calibration

This calibration is needed only when the position sensor module is replaced. Position sensor is factory calibrated. This calibration routine prompts and advises the user via dialog boxes.

4.3.2.5.3 Pressure sensor calibration

This calibration is needed only when the pressure sensor module is replaced. Pressure sensor is factory calibrated. This calibration routine prompts and advises the user via dialog boxes.

4.3.2.5.4 Temperature measurement calibration

This calibration calibrates the on-board temperature measurement. Temperature measurement is factory calibrated. This calibration routine prompts and advises the user via dialog boxes.

4.3.2.5.5 Position sensor temperature compensation

The position sensor temperature compensation adjustment is needed only when the position sensor module is replaced. Position sensor temperature compensation is factory calibrated.

POS_SENSOR_SN

Enter here the position sensor serial number.

POS_SENSOR_TEMP_COMP

Position sensor temperature compensation selector. Option are

- Compensation off. This selection turns the compensation off.
- Compensation on, User defined table. This selection turns the compensation on and uses the User defined table.
- Compensation on, Default table. This selection turns the compensation on and uses the default compensation table stored in the device memory. Use this selection if the compensation data is not available.

POS_SENSOR_TEMP_COMP_MIN_TABLE

Enter here the values that are shipped with the new position sensor.

POS_SENSOR_TEMP_COMP_MAX_TABLE

Enter here the values that are shipped with the new position sensor.

4.3.2.6 Valve diagnostics

The ND800FF valve controller continuously monitors control valve behavior. It records valve travel trends, histograms and possible alerts. Trends and histograms show how the valve has operated over a long time period. Alerts tell about the latest problems in control valve operation including control behavior and ND800FF valve controller faults. Valve and actuator travel distances and reversals are also counted to inform the user of the need for maintenance.

All the diagnostics information is very important for the process control and maintenance needs. Diagnostics information is on-line data which allows the process automation people to check the condition of the control valves anytime needed. This checking is done in real-time and it does not disturb the process. With these trends, histograms, travel distance and reversal counters and alert reports user can schedule the maintenance needs for the control valves. This is highly important predictive maintenance and it allows control valves and therefore the whole process to work more accurate and more reliably.

4.3.2.6.1 General

OPERATION_TIME

Device operation time. This comprises the total time during which the valve controller has been operating, and is displayed in hours. Resetting the diagnostics does not affect the total time.

ACT_PRESSURE

The actuator pressure difference can be used to check that the control valve is working properly. With the aid of pressure information, possible friction and actuator leakages can be discovered by comparing prevailing and previous pressure curves. Actuator pressure is the pressure difference in double-acting actuators and the pressure opposed to the spring in single-acting actuators.

DEV_TEMPERATURE

This parameter shows the temperature inside the ND800FF valve controller on the PCB (Printed Circuit Board). The temperature information can be used for checking that the environmental temperature is within specified ambient temperature limits. This ensures reliability of the ND800FF valve controller.

RESET_DIAGNOSTICS

This parameter resets the following diagnostic data

- Event history
- Travel Deviation trend
- Valve travel vs. time trend
- Load Factor trend
- Valve travel counters
- Actuator travel counters

4.3.2.6.2 Travel counters

VALVE_FULL_STROKES

Valve number of full strokes. The distance the valve has traveled in full strokes. One full stroke means valve movement from 0 to 100%. E.g. if valve moves from 40% to 50% full strokes increases by 0.1.

VALVE_FULL_STROKES_WARNING_LIMIT - Warning limit for the distance the valve has traveled in full strokes.

VALVE_REVERSALS

The number of changes in valve movement direction.

VALVE_REVERSALS_WARNING_LIMIT

Warning limit for the number of changes in valve movement direction.

ACTUATOR_FULL_STROKES

Actuator number of full strokes. The distance the actuator has traveled in full strokes.

ACT_FULL_STROKES_WARNING_LIMIT

Warning limit for the distance the actuator has traveled in full strokes.

ACTUATOR_REVERSALS

The number of changes in actuator movement direction.

ACTUATOR_REVERSALS_WARNING_LIMIT

Warning limit for the number of changes in actuator movement direction.

4.3.2.6.3 Load factor trend

This trend shows the load factor of the actuator as a percentage. In the case of a single acting actuator, the load factor shows the actuator load with respect to the present spring force, i.e., a load factor of 100% indicates that the actual load may exceed the spring force. For double acting actuators, the load factor shows the actuator load with respect to the user-given supply pressure level, i.e., a load factor of 100% indicates that the actual load may exceed maximum attainable pressure difference being equal to the supply pressure. The trend can be used for analysing the condition of the control valve. A high load factor indicates the presence of high friction or an undersized actuator if the given supply pressure is equal to actual supply pressure level. The load factor is not updated when the valve is appropriately fully open or closed.

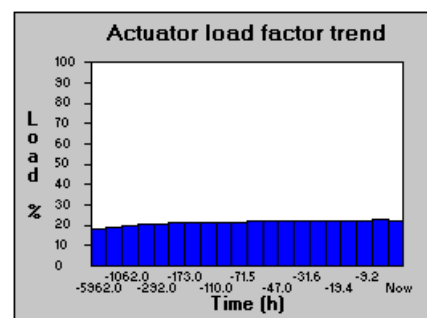


Figure 6. Actuator load factor trend.

Since the DDL does not support graphics, this trend is presented as two tables;

- LOAD_FACTOR_TREND_TIME_TABLE - X-axis values; Operating hours from this point backwards.
- LOAD_FACTOR_TREND_FACTOR_TABLE - Corresponding Y-axis values; Load factor values.

LOAD_FACTOR_WARNING_LIMIT

Warning limit for the load factor value, default 85 %.

LOAD_FACTOR_ALARM_LIMIT

Alarm limit for the load factor value, default 90 %.

4.3.2.6.4 Travel deviation trend

This trend shows the deviation between the setpoint and actual travel during total valve operation time. The travel deviation trend can be used to analyse present and predict future behaviour of the control valve. The travel deviation is not updated when the valve is appropriately fully open or closed.

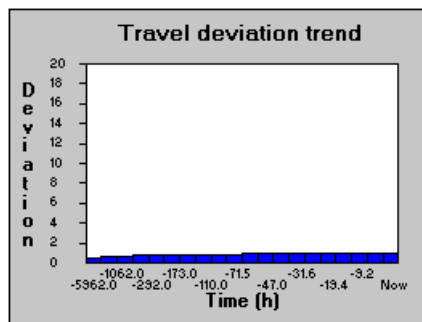


Figure 7. Travel deviation trend.

Since the DDL does not support graphics, this trend is presented as two tables;

- TRAVEL_DEVIATION_TREND_TIME_TBL - X-axis values; Operating hours from this point backwards.
- TRAVEL_DEVIATION_TREND_ERROR_TBL - Corresponding Y-axis values; Deviation values.

TRAVEL_DEVIATION_WARNING_LIMIT

Warning limit for the travel deviation value, default 5 %.

TRAVEL_DEVIATION_ALARM_LIMIT

Alarm limit for the travel deviation value, default 10 %.

4.3.2.6.5 Valve travel vs. time trend

The histogram illustrates the valve's opening history as percentages of total valve operation time. Valve opening is on the horizontal axis and percentages of operation time on the vertical axis. This histogram can be used to check valve sizing. If the valve operates mostly in small openings, this indicates that the valve is oversized for the application.

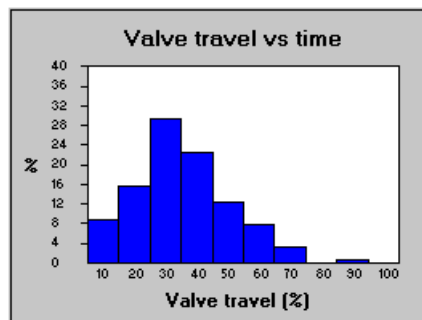


Figure 8. Valve travel vs. time trend.

Since the DDL does not support graphics, this trend is presented as a table.

VALVE_TRAVEL_VS_TIME_TREND

The variable indexes of the table represent the valve travel travel (X-axis values at range 10 - 100 %) and the corresponding values are the percentages of total operation time time (Y-axis values).

ELAPSED_TREND_TIME

This is the time during which the data for the statistics displayed have been collected, and is displayed in hours.

4.3.2.6.6 Event log

Event log holds 20 latest events and failures stored in the ND800FF memory. Event log is presented in four tables;

- EVENT_LOG_EVENT_TABLE - Events.
- EVENT_LOG_COUNT_TABLE - Number of consecutively occurred events.
- EVENT_LOG_TIME_TABLE_1 - Event log time stamps are splitted into two separate tables. If there are consecutively occurred events, the time stamp reports the first event occurrence time.
- EVENT_LOG_TIME_TABLE_2 - Second time stamp table.

4.3.2.7 Valve test

To ensure that the control valves are working properly they can be tested with the Testing function. Two control valve performance tests, the Hysteresis loop and Step response, can be run with the device. The Step response test tells how rapidly the valve reacts to the input signal, while the Hysteresis loop test tells how large is the dynamic hysteresis + dead band, which indicates friction quantity. The control valve test results can be used to diagnose the condition of the control valve, thus showing the need for possible future maintenance work. Running control valve tests is, therefore, very important predictive maintenance work and ensures optimal performance of the control valve in every situation. Note that control valve testing affects process control by driving the control valve independently of the input signal from the control system !

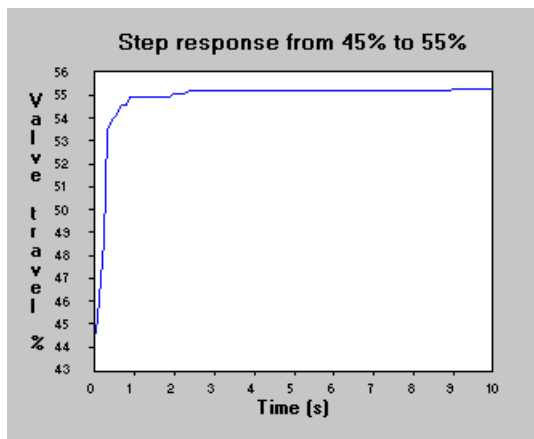


Figure 9. Step response test.

VALVE_TEST_xxx

The valve test is controlled with parameters named VALVE_TEST_xxx.

- VALVE_TEST_TYPE - Valve test type; Step response test or Hysteresis loop test.
- VALVE_TEST_STARTING_SETPOINT - Test starting setpoint. **Hysteresis test**; This is the opening where the test is started. **Step response test**; This is the initial opening of the valve prior to the step.
- VALVE_TEST_LOOP_OR_STEP_SIZE - Hysteresis loop or step size.
- VALVE_TEST_DURATION - **Hysteresis test**; This is the total time used to run the hysteresis loop. Minimum time allowed is dependent of the loop size. The time must be more than 50 seconds for each 10 % (i.e. 50 s for 10 % loop and 500 s for 100 % loop. **Step response test**; This is the time period in which the valve opening is measured. The step is made at t = 0 seconds.
- VALVE_TEST_START - Controls the test execution. Options are:
 - Idle - Test state indication.
 - Start test - Starts the configured test.
 - Stop test - Cancels the test.
 - Test is initializing - Test state indication.
 - Test is running - Test state indication.
 - Test was stopped - Test state indication.
 - Test was successful - Test state indication.
 - Test was unsuccessful - Test state indication.

LAST_VALVE_TEST_xxx

Parameters named LAST_VALVE_TEST_xxx contain the settings of the last successfully ran valve test.

- LAST_VALVE_TEST_TYPE - Last ran test type; Step response test or Hysteresis loop test.
- LAST_VALVE_TEST_STARTING_SP - Last ran test starting setpoint.
- LAST_VALVE_TEST_LOOP_OR_STEP_SIZE - Hysteresis loop or step size.
- LAST_VALVE_TEST_DURATION - See VALVE_TEST_DURATION.

Since the DDL does not support graphics, the test results are presented as tables. These tables contains 50 measurement pairs.

- LAST_VALVE_TEST_SP_OR_TIME_TBL_1 - The X-axis data is splitted into two separate tables. **Hysteresis test**; Valve position setpoint. **Step response test**; Time in seconds.
- LAST_VALVE_TEST_SP_OR_TIME_TBL_2 - The second X-axis data table.
- LAST_VALVE_TEST_POSITION_TABLE_1 - The Y-axis data splitted into two separate tables as well. This table contains the valve position measurement in both test cases.
- LAST_VALVE_TEST_POSITION_TABLE_2 - The second Y-axis data table.

4.3.2.8 Alerts

ALERT_KEY

The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.

UPDATE_EVT

See paragraph "Parameters common to all blocks", page 11.

BLOCK_ERR

See paragraph "Parameters common to all blocks", page 10. Whenever any XD_ERROR is active, BLOCK_ERR "Other" is reported.

XD_ERROR

This parameter reports the errors detected in the valve package. This is a standard FF parameter.

Table 2. XD_ERROR

Error	Description
Unspecified error	An error has occurred that was not identified.
General error	An error has occurred that could not be classified as one of the errors below.
Calibration error	An error occurred during calibration of the device or a calibration error has been detected during operation of the device.
Configuration error	An error occurred during configuration of the device or a configuration error has been detected during operation of the device.
Electronics Failure	An electronic component has failed.
Mechanical Failure	A mechanical component has failed.
I/O Failure	An I/O failure has occurred
Data Integrity Error	Indicates that data stored within the system may no longer be valid due to NVM checksum failure or Data verify after write failure.
Software Error	The software has detected an error. This could be caused by an improper interrupt service routine, an arithmetic overflow or a watchdog timer.
Algorithm Error	The algorithm used in the transducer block produced an error. This could be due to an overflow or data reasonableness failure.

XD_ERROR_EXTENSION_PART_1 & 2

This parameter is a manufacturer specific extension to the XD_ERROR parameter.

Table 3. XD_ERROR_EXTENSION PART 1 & 2

Error	Description
Pneumatic prestage valve 1 control failure	See /1/ for actions.
Pneumatic prestage valve 2 control failure	See /1/ for actions.
Position feedback ADC low limit failure	See /1/ for actions.
Position feedback ADC high limit failure	See /1/ for actions.
Pressure sensor failure	See /1/ for actions.
EEPROM error	See /1/ for actions.
RAM error	See /1/ for actions.
ROM error	See /1/ for actions.
Processor failure	See /1/ for actions.
Board to board communication timeout	The communication between the circuit boards is broken.
Write was not successful	The write operation from the bus did not succeed.
Valve full strokes warning limit exceeded	Valve full strokes warning limit has been exceeded.
Valve reversals warning limit exceeded	Valve reversals warning limit has been exceeded.
Actuator full strokes warning limit exceeded	Actuator full strokes warning limit has been exceeded.
Actuator reversals warning limit exceeded	Actuator reversals warning limit has been exceeded.
Deviation warning limit exceeded	The first value in the LOAD_FACTOR_TREND_FACTOR_TABLE has crossed the LOAD_FACTOR_WARNING_LIMIT. That value is a mean value during the time period specified in the first element of the LOAD_FACTOR_TREND_TIME_TABLE.
Load factor warning limit exceeded	The first value in the TRAVEL_DEVIATION_TREND_ERROR_TBL has crossed the TRAVEL_DEVIATION_WARNING_LIMIT. That value is a mean value during the time period specified in the first element of the TRAVEL_DEVIATION_TREND_TIME_TBL.
Deviation alarm limit exceeded	The first value in the TRAVEL_DEVIATION_TREND_ERROR_TBL has crossed the TRAVEL_DEVIATION_ALARM_LIMIT. That value is a mean value during the time period specified in the first element of the TRAVEL_DEVIATION_TREND_TIME_TBL.
Load factor alarm limit exceeded	The first value in the LOAD_FACTOR_TREND_FACTOR_TABLE has crossed the LOAD_FACTOR_ALARM_LIMIT. That value is a mean value during the time period specified in the first element of the LOAD_FACTOR_TREND_TIME_TABLE.
Pneumatics problem	Problem detected in pneumatics.
Friction problem	High friction detected in actuator or valve.
Travel deviation alert	The load factor ON-LINE measurement value has crossed the TRAVEL_DEVIATION_WARNING_LIMIT.
Load factor alert	The load factor ON-LINE measurement value has crossed the LOAD_FACTOR_WARNING_LIMIT.

BLOCK_ALM

See paragraph "Parameters common to all blocks", page 10. In addition to BLOCK_ERR, also the changes in the XD_ERROR parameter are reported.

4.3.3 Parameter access Table

View object is a package which contains a predefined set of parameters. It is more efficient to read view objects from the device than polling the parameters separately. There are four different types of view objects

- VIEW_1 - View object defined to access the dynamic operating parameters of a block.
- VIEW_2 -View object defined to access the static operating parameters of a block.
- VIEW_3 - View object defined to access all dynamic parameters of a block. More than one View 3 object may be required to represent all dynamic parameters of the block.
- VIEW_4 - View object defined to access static parameters not included in VIEW_2. More than one VIEW_4 object may be required to represent these other static parameters of the block.

Table 4. Transducer block parameter access table.

Index	Parameter Mnemonic	V I E W																											
		1	2	3	3.1	3.2	3.3	3.4	3.5	4	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	4.10	4.11	4.12	4.13	4.14	4.15				
1	ST_REV	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	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	UPDATE_EVT																												
8	BLOCK_ALM																												
9	TRANS-DUCER_DIRECTORY																												
10	TRANSDUCER_TYPE	2	2	2						2																			
11	XD_ERROR	1		1																									
12	COLLEC-TION_DIRECTORY																												
13	FINAL_VALUE	5		5																									
14	FINAL_VALUE_RANGE		11																										
15	FI-NAL_VALUE_CUTOFF_HI									4																			
16	FI-NAL_VALUE_CUTOFF_LO									4																			
17	FINAL_POSITION_VALUE	5		5																									
18	SERVO_GAIN									4																			
19	ACT_FAIL_ACTION									1																			
20	ACT_MAN_ID									4																			
21	ACT_MODEL_NUM									32																			
22	ACT_SN									32																			
23	VALVE_MAN_ID									4																			
24	VALVE_MODEL_NUM									32																			
25	VALVE_SN									32																			
26	VALVE_TYPE									1																			
27	XD_CAL_LOC										32																		
28	XD_CAL_DATE										7																		
29	XD_CAL_WHO										32																		
30	SERVO_PARAM_D										4																		
31	SERVO_PARAM_B										4																		
32	ACT_TYPE										1																		
33	SUPPLY_PRESSURE										4																		
34	POSI-TIONER_FAIL_ACTION										1																		
35	POS SENSOR ROT										1																		
36	DEAD_ANGLE_COMP										4																		
37	SOURCE_CHARACTERISTICS_TBL													84															
38	FILL_SOURCE_CHARACTERISTICS_TBL													1															
39	TARGET_CHARACTERISTICS_TBL														84														
40	FILL_TARGET_CHARACTERISTICS_TBL														1														
41	SIGNAL_MOD_TABLE															84													
42	CHARACTERIZATION															1													
43	AUTO_TRAVEL_CALIBRATION											1																	
44	ACT_PRESSURE			4																									
45	DEV_TEMPERATURE			4																									
46	OPERATION_TIME			4																									
47	EVENT_LOG_EVENT_TABLE																40												
48	EVENT_LOG_COUNT_TABLE																20												
49	EVENT_LOG_TIME_TABLE																	80											

Function blocks / Transducer block

In dex	Parameter Mnemonic	V I E W																									
		1	2	3	3.1	3.2	3.3	3.4	3.5	4	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	4.10	4.11	4.12	4.13	4.14	4.15		
	E 1																										
50	EVENT_LOG_TIME_TBL E 2																		80								
51	TRAVEL_DEVIATION_TREND_TIME_TBL				80																						
52	TRAVEL_DEVIATION_TREND_ERROR_TBL					80																					
53	TRAVEL_DEVIATION_WARNING_LIMIT												4														
54	TRAVEL_DEVIATION_ALARM_LIMIT												4														
55	VALVE_TRAVEL_VS_TIME_TREND						40																				
56	ELAPSED_TREND_TIME						4																				
57	LOAD_FACTOR_TREND_TIME_TABLE							80																			
58	LOAD_FACTOR_TREND_FACTOR_TABLE								80																		
59	LOAD_FACTOR_WARNING_LIMIT												4														
60	LOAD_FACTOR_ALARM_LIMIT												4														
61	VALVE_FULL_STROKES			4																							
62	VALVE_FULL_STROKES_WARNING_LIMIT												4														
63	VALVE_REVERSALS			4																							
64	VALVE_REVERSALS_WARNING_LIMIT												4														
65	ACTUATOR_FULL_STROKES			4																							
66	ACT_FULL_STROKES_WARNING_LIMIT												4														
67	ACTUATOR_REVERSALS			4																							
68	ACTUATOR_REVERSALS_WARNING_LIMIT												4														
69	RESET_DIAGNOSTICS											1															
70	VALVE_TEST_TYPE											2															
71	VALVE_TEST_STARTING_SETPOINT											4															
72	VALVE_TEST_LOOP_OR_STEP_SIZE											4															
73	VALVE_TEST_DURATION											4															
74	VALVE_TEST_START											1															
75	LAST_VALVE_TEST_TYPE											2															
76	LAST_VALVE_TEST_STARTING_SP											4															
77	LAST_VALVE_TST_LOOP_OR_STEP_SIZE											4															
78	LAST_VALVE_TEST_DURATION											4															
79	LAST_VALVE_TEST_SP_OR_TIME_TBL_1																			100							
80	LAST_VALVE_TEST_SP_OR_TIME_TBL_2																				100						
81	LAST_VALVE_TEST_POSITION_TABLE_1																					100					
82	LAST_VALVE_TEST_POSITION_TABLE_2																						100				
83	POS_SENSOR_SN											4															
84	POS_SENSOR_TEMP_COMP											1															
85	POS_SENSOR_TEMP_COMP_MIN_TABLE																							104			
86	POS_SENSOR_TEMP_COMP_MAX_TABLE																								104		
87	XD_ERROR_EXTENSION_PART_1			4																							
88	XD_ERROR_EXTENSION_PART_2			2																							
	Totals (bytes)	21	15	55	82	82	46	82	82	88	106	73	54	87	87	87	62	82	82	102	102	102	102	106	106	106	

4.4 Analog Output function block

4.4.1 Overview

The AO block converts the value in CAS_IN / RCAS_IN signal to valve position setpoint and passes it to the transducer block through an internal channel reference. The AO block inputs and outputs are presented in figure 10.

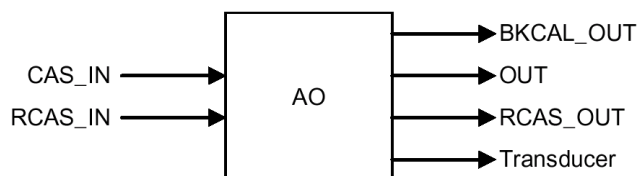


Figure 10. AO block inputs and outputs.

AO schematic is presented in figure 11.

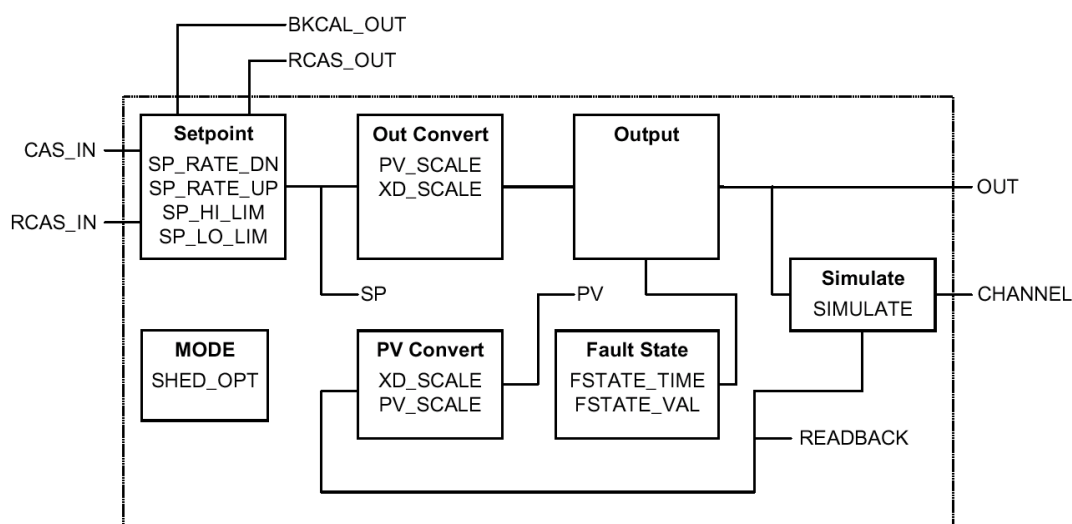


Figure 11. AO block schematic.

The PV_SCALE range is used to convert the SP to percent of span. The XD_SCALE range is used to convert percent of span to the valve position setpoint. This allows portions of the SP span to cause full span movement of the output. The Increase to close option in IO_OPTS can be used to invert the span.

The SP supports the full cascade sub-function. In fact, Cas mode must be used to transfer the output of another block to the SP of the AO. The SP has standard ramp rate and absolute limits. There are additional I/O options which will cause the SP value to track the PV value when the block is in an actual I/O mode of LO or Man. If the hardware supports a readback value, such as valve position, that value should be run backwards through the XD scaling to act as the PV for this block. If not supported, READBACK is generated from OUT. The OUT and READBACK parameters both use XD_SCALE. The PV and SP use PV_SCALE.

There is an option to use either the SP value after limiting or the PV value for the BKCAL_OUT value.

4.4.2 Parameters

4.4.2.1 General

ST_REV

The revision level of the static data associated with the function block. The revision value will be incremented each time a static parameter value in this block is changed.

TAG_DESC

The user description of the intended application of the block.

STRATEGY

The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.

GRANT_DENY

See paragraph "Parameters common to all blocks", page 8.

CHANNEL

The number of the logical hardware channel that is connected to this AO block. This information defines the transducer to be used going to or from the physical world.

4.4.2.2 Process

MODE_BLK

See paragraph "Parameters common to all blocks", page 8.

AO block modes are:

- Out of Service (O/S) - The block is not being evaluated. The output is maintained at last value. Setpoint is maintained at last value.
- Initialization Manual (IMan) - AO block actual mode is under way to change to the target mode. If actual mode stays on IMan, READBACK signal from the transducer block may have BAD status. The setpoint may be maintained or, optionally, initialized to the process variable parameter value.
- Local Override (LO), - In the local override mode, the block output is being set to track the value of the FSTATE_VAL parameter. The algorithm initializes so no bump is experienced when the mode switches from LO back to the target mode. The setpoint may be maintained or, optionally, be initialized to the process variable parameter value.
- Manual (Man) - The block output is not being calculated, although it may be limited. It is directly set by the operator through an interface device. The algorithm initializes so no bump is experienced when the mode switches. The setpoint may be maintained or, optionally, be initialized to the process variable parameter value or to the setpoint value associated with the previous (retained) target mode.
- Automatic (Auto) - A local setpoint value is used by the normal block algorithm in determining the primary output value. The local setpoint value may be written to by an operator through an interface device.
- Cascade (Cas) - A setpoint value supplied by another function block through the Cascade input parameter is used by the normal block algorithm in determining the primary output value. This connection between function blocks is defined by a link object.
- Remote-Cascade (RCas) - The block setpoint is being set by a Control Application running on an interface device through the remote-cascade in parameter. Based on this setpoint, the normal block algorithm determines the primary output value. A remote-cascade out parameter is maintained by the block to support initialization of the control application when the block mode is not remote-cascade.

PV

Either valve position setpoint calculated from AO.OUT value (default) or measured valve position calculated from the AO.READBACK value. See FEATURE_SEL, page 15 and paragraph “Parameters common to all blocks”, page 9.

CAS_IN

This parameter is the remote setpoint value, which must come from another Fieldbus block, or a DCS block through a defined link. This setpoint is used in Cascade mode. See paragraph “Parameters common to all blocks”, page 9.

SP

The AO block setpoint calculated from CAS_IN (Cascade mode), RCAS_IN (Remote cascade mode) or entered by the user (Auto mode). See SP_HI_LIM, SP_LO_LIM, SP_RATE_UP and SP_RATE_DN at page 36. See paragraph “Parameters common to all blocks”, page 9.

OUT

The AO block output (valve position setpoint) either calculated from the SP (Auto, Cascade, Remote cascade) or tracks FSTATE_VAL (fault state active) or entered by user (Manual mode). This signal is passed to the transducer block through an internal channel reference. Transducer block still may apply some signal modifications to this signal. See PV_SCALE, XD_SCALE, page 36, IO_OPTS, page 36. See paragraph “Parameters common to all blocks”, page 9.

READBACK

This indicates the measured valve position, in transducer units (%). See paragraph “Parameters common to all blocks”, page 9.

BKCAL_OUT

The value and status required by an upper block’s BKCAL_IN so that the upper block may prevent reset windup and provide bumpless transfer to closed loop control. Depending on IO_OPTS, the value is either SP (default) or PV. See IO_OPTS, page 36. See paragraph “Parameters common to all blocks”, page 9.

SIMULATE

This parameter acts as a switch at the interface between the AO function block and the transducer block. When the enable switch is on, the simulate value and status become the READBACK value and status, and the transducer block is ignored. The status can be used to simulate transducer faults. The transducer attribute value and status reflect the transducer read-back value and status when simulation is enabled and the transducer maintains last output and ignores the OUT of the AO block. It is necessary to show that a block has a simulated value, without touching the status of parameters that may be linked elsewhere. The block alarm parameter will provide this visibility through the simulate active attribute. When disabled, the simulation parameter should take on the value and status it would supply if enabled. See FEATURE_SEL, page 15.

- Simulate Status - Used for the transducer status when simulation is enabled.
- Simulate Value - Used for transducer value when simulation is enabled.
- Transducer Status - Status of value supplied by transducer.
- Transducer Value - Current value supplied by transducer.
- Enable/disable - Enable/disable simulation.

4.4.2.3 Remote

SHED_OPT

Defines action to be taken on remote control device timeout. See paragraph “Parameters common to all blocks”, page 11.

RCAS_IN

Target setpoint and status provided by a supervisory Host. See paragraph "Parameters common to all blocks", page 9.

RCAS_OUT

Block setpoint and status after ramping - provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change. See paragraph "Parameters common to all blocks", page 9.

4.4.2.4 Options

IO_OPTS

Contains options which the user may select to alter the AO block processing.

- SP tracks PV if Man; Permits the setpoint to track the process variable when the target mode of the block is Man.
- SP tracks PV if LO; Permits the setpoint to track the process variable when the actual mode of the block is LO.
- SP tracks RCas or Cas if LO or Man; Permits the setpoint to track the RCas or Cas parameter based on the retained target mode when the actual mode of the block is LO or Man. When SP-PV track options are enabled, then SP Track retained target will have precedence in the selection of the value to track when the actual mode is Man and LO.
- Increase to close; Indicates whether the output value should be inverted before it is communicated to the I/O channel.
- Fault state to value; The output action to take when failure occurs. If set, go to the FSTATE_VAL. If not set, freeze.
- Fault state restart; Use the value of FSTATE_VAL if the device is restarted, otherwise use the non-volatile value. This does not act like fault state, just uses the value.
- Target to Man if fault state activated; Set the target mode to Man, thus losing the original target, if Fault State is activated. This latches an output block into the manual mode.
- PV for BKCAL_OUT; The BKCAL_OUT value is normally the working SP. This option changes it to the PV.

STATUS_OPTS

Options which the user may select in the block processing of status.

- Propagate Fault Backward - If the status from the actuator is Bad, Device failure or Fault State Active or Local Override is active, propagate this as Bad, Device Failure or Good Cascade, Fault State Active or Local Override to BKCAL_OUT respectively without generating an alarm. The use of these sub-status in BKCAL_OUT is determined by this option. Through this option, the user may determine whether alarming (sending of an alert) will be done by the block or propagated upstream for alarming.

4.4.2.5 Scaling

PV_SCALE

The scale of the PV, as default 0-100%. Any unit can be used. See paragraph "Parameters common to all blocks", page 9.

XD_SCALE

The scale of the transducer, as default 0-100%. Since the valve position is expressed as percentage of the valve movement span, the only valid unit is %. See paragraph "Parameters common to all blocks", page 9.

4.4.2.6 Limits

SP_HI_LIM

The setpoint high limit is the highest setpoint operator entry that can be used for the block.

SP_LO_LIM

The setpoint low limit is the lowest setpoint operator entry that can be used for the block.

SP_RATE_UP

Ramp rate at which upward setpoint changes are acted on in Auto mode, in PV units per second. If the ramp rate is set to zero or the block is in a mode other than Auto, then the setpoint will be used immediately.

SP_RATE_DN

Ramp rate at which downward setpoint changes are acted on in Auto mode, in PV units per second. If the ramp rate is set to zero or the block is in a mode other than Auto, then the setpoint will be used immediately.

4.4.2.7 Fault state

The Fault State parameters determine the response of an output block if one or more of the following conditions exists for a time that exceeds FSTATE_TIME;

- Loss of communications to CAS_IN
- There is Initiate Fault State status at CAS_IN when the target mode is CAS
- There is Initiate Fault State status at RCAS_IN when the target mode is RCAS.

If one of these conditions exist, then the block will go to the defined Fault State. The FAULT_STATE parameter of the resource block may also put this block into the Fault State. The IO_OPTS "Faultstate to value" determines whether the action is simply to hold, or to move to FSTATE_VAL. The "Target to Manual if IFS" I/O option may be used to latch the Fault State. This will cause a Fault State block alarm and the target mode to automatically change to manual. The target mode may be manually changed from manual when conditions are correct.

FSTATE_TIME

The time in seconds from detection of failure of the output block remote setpoint to the output action of the block output if the condition still exists.

FSTATE_VAL

The preset OUT value to use when failure occurs. This value will be used if the IO_OPTS "Faultstate to value" is selected.

4.4.2.8 Alerts

To enable alert reporting, select FEATURE_SEL "Reports", page 15.

ALERT_KEY

The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.

BLOCK_ERR

See paragraph "Parameters common to all blocks", page 10.

BLOCK_ALM

See paragraph "Parameters common to all blocks", page 10.

UPDATE_EVT

See paragraph "Parameters common to all blocks", page 11.

4.4.3 Parameter access table

View object is a package which contains a predefined set of parameters. It is more efficient to read view objects from the device than polling the parameters separately. There are four different types of view objects

- VIEW_1 - View object defined to access the dynamic operating parameters of a block.
- VIEW_2 -View object defined to access the static operating parameters of a block.
- VIEW_3 - View object defined to access all dynamic parameters of a block. More than one View 3 object may be required to represent all dynamic parameters of the block.
- VIEW_4 - View object defined to access static parameters not included in VIEW_2. More than one VIEW_4 object may be required to represent these other static parameters of the block.

Table 5. AO block parameter access table.

Index	Parameter Mnemonic	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	PV	5		5	
8	SP	5		5	
9	OUT	5		5	
10	SIMULATE				
11	PV_SCALE		11		
12	XD_SCALE		11		
13	GRANT_DENY		2		
14	IO_OPTS				2
15	STATUS_OPTS				2
16	READBACK	5		5	
17	CAS_IN	5		5	
18	SP_RATE_DN				4
19	SP_RATE_UP				4
20	SP_HI_LIM		4		
21	SP_LO_LIM		4		
22	CHANNEL				2
23	FSTATE_TIME				4
24	FSTATE_VAL				4
25	BKCAL_OUT			5	
26	RCAS_IN			5	
27	SHED_OPT				1
28	RCAS_OUT			5	
29	UPDATE_EVT				
30	BLOCK_ALM				
	Totals (bytes)	33	34	48	28

4.5 PID controller function block

4.5.1 Overview

The PID block is key to many control schemes and is used almost universally, with the exception of PD, which is used when the process itself does the integration. As long as an error exists, the PID function will integrate the error, which moves the output in a direction to correct the error. PID blocks may be cascaded when the difference in process time constants of a primary and secondary process measurement makes it necessary or desirable. The PID controller block inputs and outputs are presented in figure 12.

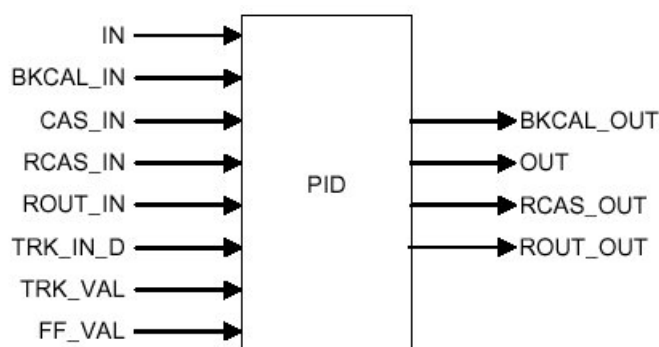


Figure 12. PID controller block inputs and outputs.

PID controller block schematic is presented in figure 13.

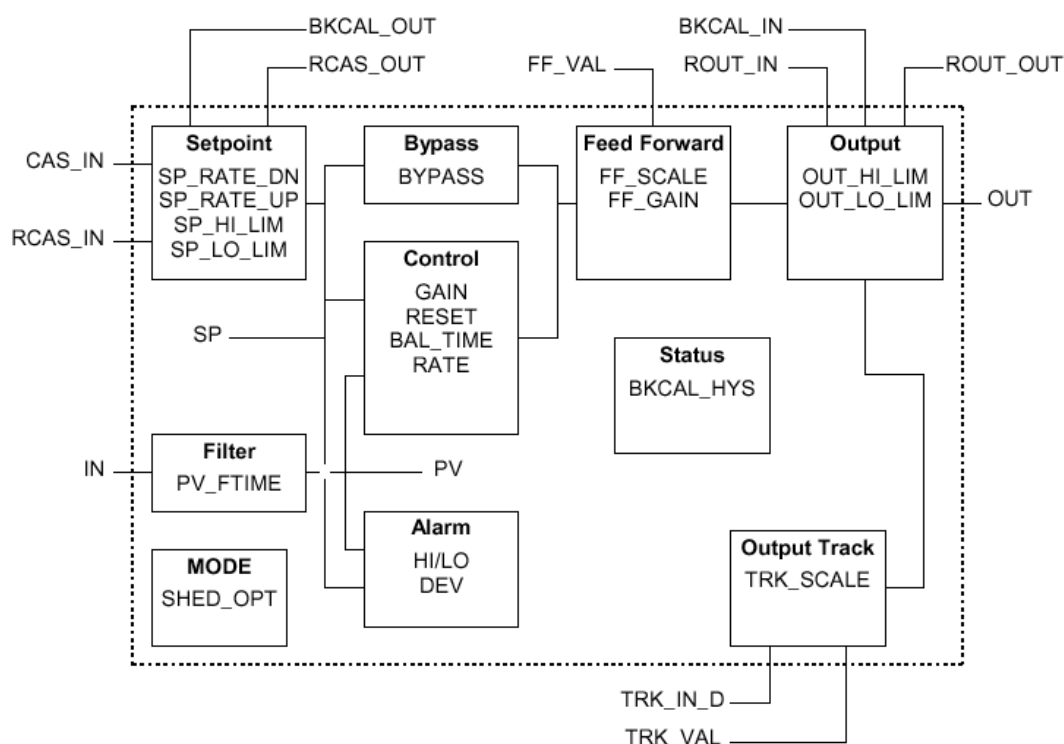


Figure 13. PID controller block schematic.

The Process Value to be controlled is connected to the IN input. This value is passed through a filter whose time constant is PV_FTIME. The value is then shown as the PV, which is used in conjunction with the SP in the PID algorithm. A PID will not integrate if the limit status of IN is constant. A full PV and DV alarm sub-function is provided. The PV has a status, although it is a Contained parameter. This status is a copy of IN's status unless IN is good and there is a PV or block alarm.

The full cascade SP sub-function is used, with rate and absolute limits. There are additional control options which will cause the SP value to track the PV value when the block is in an actual mode of IMan, LO, Man or ROut. Limits do not cause SP-PV tracking.

There is a switch for BYPASS, which is available to the operator if the Bypass Enable control option is true. Bypass is used in secondary cascade controllers that have a bad PV. The Bypass Enable option is necessary because not all cascade control schemes will be stable if BYPASS is true. BYPASS can only be changed when the block mode is Man or O/S. While it is set, the value of SP, in percent of range, is passed directly to the target output, and the value of OUT is used for BKCAL_OUT. When the mode is changed to Cas, the upstream block is requested to initialize to the value of OUT. On the transition out of bypass, the upstream block is requested to initialize to the PV value, regardless of the "Use PV for BKCAL_OUT" option.

GAIN, RESET, and RATE are the tuning constants for the P, I, and D terms, respectively. Gain is a dimensionless number. RESET and RATE are time constants expressed in seconds. There are existing controllers that are tuned by the inverse value of some or all of them, such as proportional band and repeats per minute. The human interface to these parameters should be able to display the user's preference.

The Direct Acting control option, if true, causes the output to increase when the PV exceeds the SP. If false, the output will decrease when the PV exceeds the SP. It will make the difference between positive and negative feedback, so it must be set properly, and never changed while in an automatic mode. The setting of the option must also be used in calculating the limit state for BKCAL_OUT.

The output supports the feed forward algorithm. The FF_VAL input brings in an external value which is proportional to some disturbance in the control loop. The value is converted to percent of output span using the values of parameter FF_SCALE. This value is multiplied by the FF_GAIN and added to the target output of the PID algorithm. If the status of FF_VAL is Bad, the last usable value will be used, because this prevents bumping the output. When the status returns to good, the block will adjust its integral term to maintain the previous output.

The output supports the track algorithm.

There is an option to use either the SP value after limiting or the PV value for the BKCAL_OUT value.

4.5.2 Parameters

4.5.2.1 General

ST_REV

The revision level of the static data associated with the function block. The revision value will be incremented each time a static parameter value in this block is changed.

TAG_DESC

The user description of the intended application of the block.

STRATEGY

The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.

GRANT_DENY

See paragraph "Parameters common to all blocks", page 8.

TUNE_IN

This parameter is intended to be used with controller tuning program like TuneUp. Value of this signal is summed to controller output when controller is operating in RCAS, CAS or AUTO mode.

CURRENT_TIME

Presents current time used on the device.

4.5.2.2 Process

MODE_BLK

See paragraph “Parameters common to all blocks”, page 8.

PID controller block modes are:

- Out of Service (O/S) - The block is not being evaluated. The output is maintained at last value. Setpoint is maintained at last value.
- Initialization Manual (IMan) - The block output is being set in response to the back-calculation input parameter status. When the status indicates there is no path to the final output element, then the PID block initializes to provide for bumpless transfer when the condition clears. The setpoint may be maintained or, optionally, initialized to the process variable parameter value.
- Local Override (LO), - In the local override mode, the block output is being set to track the value of the TRK_VAL parameter. The algorithm must initialize so no bump is experienced when the mode switches from LO back to the target mode. The setpoint may be maintained or, optionally, be initialized to the process variable parameter value.
- Manual (Man) - The block output is not being calculated, although it may be limited. It is directly set by the operator through an interface device. The algorithm initializes so no bump is experienced when the mode switches. The setpoint may be maintained or, optionally, be initialized to the process variable parameter value or to the setpoint value associated with the previous (retained) target mode.
- Automatic (Auto) - A local setpoint value is used by the normal block algorithm in determining the primary output value. The local setpoint value may be written to by an operator through an interface device.
- Cascade (Cas) - A setpoint value supplied by another function block through the Cascade input parameter is used by the normal block algorithm in determining the primary output value. This connection between function blocks is defined by a link object.
- Remote-Cascade (RCas) - The block setpoint is being set by a Control Application running on an interface device through the remote-cascade in parameter. Based on this setpoint, the normal block algorithm determines the primary output value. A remote-cascade out parameter is maintained by the block to support initialization of the control application when the block mode is not remote-cascade.
- Remote-Output (ROut) - The block output is being set by a Control Application running on an interface Device through the remote-output in parameter. The algorithm must initialize so no bump is experienced when the mode switches. A remote-output out parameter is maintained by the block to support initialization of the control application when the block mode is not remote-output. The setpoint may be maintained or, optionally, be initialized to the process variable value.

BYPASS

The normal control algorithm may be bypassed through this parameter. When bypass is set, the setpoint value (in percent) will be directly transferred to the output. To prevent a bump on transfer to/from bypass, the setpoint will automatically be initialized to the output value or process variable, respectively, and the path broken flag will be set for one execution. See CONTROL_OPTS, page 43.

SP

The PID controller setpoint calculated from CAS_IN (Cascade mode), RCAS_IN (Remote cascade mode) or entered by the user (Auto mode). See SP_HI_LIM, SP_LO_LIM, SP_RATE_UP, SP_RATE_DN and SP_RAMP at page 44. See paragraph "Parameters common to all blocks", page 9.

IN

The primary input value of the block i.e. the raw measurement signal from the process. See paragraph "Parameters common to all blocks", page 9.

PV

The controlled process variable scaled and filtered from the IN signal. See PV_SCALE, page 43 and PV_FTIME, page 47 and paragraph "Parameters common to all blocks", page 9.

CAS_IN

This parameter is the remote setpoint value, which must come from another Fieldbus block, or a DCS block through a defined link. This setpoint is used in Cascade mode. See paragraph "Parameters common to all blocks", page 9.

OUT

The PID controller output calculated by the control algorithm (Cas, Rcas and Auto mode) or copied from ROUT_IN (Rout mode) or entered by user (Man mode). See OUT_SCALE, page 43 and paragraph "Parameters common to all blocks", page 9.

BKCAL_IN

The value and status from a lower block's BKCAL_OUT that is used to prevent reset windup and to initialize the control loop. See paragraph "Parameters common to all blocks", page 9.

BKCAL_OUT

The value and status required by an upper block's BKCAL_IN so that the upper block may prevent reset windup and provide bumpless transfer to closed loop control. Depending on CONTROL_OPTS, the value is either SP (default) or PV. See CONTROL_OPTS, page 43. See paragraph "Parameters common to all blocks", page 9.

4.5.2.3 Remote

SHED_OPT

Defines action to be taken on remote control device timeout. See paragraph "Parameters common to all blocks", page 11.

RCAS_IN

Target setpoint and status provided by a supervisory Host to a analog control or output block. See paragraph "Parameters common to all blocks", page 9.

RCAS_OUT

Block SP or PV (depending on CONTROL_OPTS) and status after ramping - provided to a supervisory Host for back calculation and to allow action to be taken under limiting conditions or mode change. See paragraph "Parameters common to all blocks", page 9.

ROUT_IN

Target output and status provided by a Host to the control block for use as the output (ROut mode). See paragraph "Parameters common to all blocks", page 9.

ROUT_OUT

Block output and status - provided to a Host for back calculation in ROut mode and to allow action to be taken under limited conditions or mode change. See paragraph "Parameters common to all blocks", page 9.

4.5.2.4 Options

CONTROL_OPTS

Options which the user may select to alter the calculations done in a control block.

- Bypass Enable; This parameter, if true, allows BYPASS to be set. Some control algorithm applications cannot provide closed loop control if bypassed.
- SP-PV Track in Man; Permits the setpoint to track the process variable when the target mode of the block is Man.
- SP-PV Track in Rout; Permits the setpoint to track the process variable when the actual mode of the block is ROut.
- SP-PV Track in LO or IMan; Permits the setpoint to track the process variable when the actual mode of the block is LO or IMan.
- SP Track retained target; Permits the setpoint to track the RCas or Cas parameter based on the retained target mode when the actual mode of the block is IMan, LO, Man, or ROut.
- Direct Acting; Defines the relationship between a change in PV and corresponding change in output. When Direct is selected, an increase in PV results in an increase in the output.
- Track Enable; This enables the external tracking function. If true, the value in TRK_VAL will replace the value of OUT if TRK_IN_D becomes true and the target mode is not Man.
- Track in Manual; This enables TRK_VAL to replace the value of OUT when the target mode is Man and TRK_IN_D is true. The actual mode will then be LO.
- Use PV for BKCAL_OUT; The BKCAL_OUT and RCAS_OUT values are normally the working SP. This option changes it to the PV.
- Obey SP limits if Cas or Rcas; Normally the setpoint will not be restricted to the setpoint limits except when entered by a human interface device. However, if this option is selected, the setpoint will be restricted to the setpoint absolute and rate limits in the Cas and RCas modes.
- No OUT limits in Manual; Do not apply OUT_HI_LIM or OUT_LO_LIM when target and actual modes are Man. Trust the operator to do the right thing.

STATUS_OPTS

Options which the user may select in the block processing of status.

- IFS if BAD IN; Set Initiate fail safe status in the OUT parameter if the status of the IN parameter is BAD.
- IFS if BAD CAS_IN; Set Initiate fail safe status in the OUT parameter if the status of the CAS_IN parameter is BAD.
- Use Uncertain as Good; If the status of the IN parameter is Uncertain, treat it as Good. Otherwise, treat it as BAD.
- Target to Manual if BAD IN; Set the target mode to Man if the status of the IN parameter is BAD. This latches a PID block into the Man state if the input ever goes bad.
- Target to next permitted mode if BAD CAS_IN; Set the target mode to next permitted mode if the target mode is CAS and the status of CAS_IN is BAD. This latches a control block into the next permitted mode if the CAS_IN is being used in control and the status goes bad.

4.5.2.5 Scaling

PV_SCALE

The scale of the process variable. See paragraph "Parameters common to all blocks", page 9.

OUT_SCALE

The scale of the PID output. See paragraph "Parameters common to all blocks", page 9.

4.5.2.6 Limits

SP_HI_LIM

The setpoint high limit is the highest setpoint operator entry that can be used for the block.

SP_LO_LIM

The setpoint low limit is the lowest setpoint operator entry that can be used for the block.

SP_RATE_UP

Ramp rate at which upward setpoint changes are acted on in Auto mode, in PV units per second. If the ramp rate is set to zero or the block is in a mode other than Auto, then the setpoint will be used immediately.

SP_RATE_DN

Ramp rate at which downward setpoint changes are acted on in Auto mode, in PV units per second. If the ramp rate is set to zero or the block is in a mode other than Auto, then the setpoint will be used immediately.

SP_RAMP

Setpoint ramping time in seconds when block actual mode is changed from MAN to AUTO.

OUT_HI_LIM

Limits the maximum output value.

OUT_LO_LIM

Limits the minimum output value.

BKCAL_HYS

The amount that the block output must change away from its output limit before the limit status is turned off, expressed as a percent of the span of the output.

4.5.2.7 Feed forward control

FF_GAIN

The gain that the feed forward input is multiplied by before it is added to the calculated control output.

FF_SCALE

The feedforward input scale. See paragraph "Parameters common to all blocks", page 9.

FF_VAL

The feed forward value and status. See paragraph "Parameters common to all blocks", page 9.

4.5.2.8 Output tracking

TRK_SCALE

The high and low scale values, engineering units code, and number of digits to the right of the decimal point, associated with TRK_VAL. See paragraph "Parameters common to all blocks", page 9.

TRK_IN_D

This discrete input is used to initiate external tracking of the block output to the value specified by TRK_VAL. See paragraph "Parameters common to all blocks", page 9.

TRK_VAL

This input is used as the track value when external tracking is enabled by TRK_IN_D. See paragraph "Parameters common to all blocks", page 9.

4.5.2.9 Tuning

The process model based controller tuning tool consists of parameters named TUNING_xxx, PROPOSED_xxx and ACTIVATE_PROPOSED_TUNING_PARAM.

TUNING_xxx

These parameters define the first order and dead time process model and tuning method. Any time any of these parameters is changed, new proposed tuning parameters are calculated. Proposed tuning parameters are presented on the parameters named PROPOSED_xxx. These parameters can also be changed by the user. Proposed tuning parameters are not activated (copied over actual tuning parameters) until ACTIVATE_PROPOSED_TUNING_PARAM is set to "Activate".

- TUNING_PROCESS_GAIN - Process gain estimate. Dimensionless. See figure 14.
- TUNING_PROCESS_TIME_CONSTANT - Process time constant estimate, in seconds. See figure 14.
- TUNING_PROCESS_DEAD_TIME - Process dead time estimate, in seconds. See figure 14.
- TUNING_METHOD - Defines desired controller type to be tuned (PI, PID, P-PI) and tuning method (Ziegler-Nichols or Lambda tuning with lambda values 0.5 - 5.0). Tuning parameter lambda defines closed loop step response (controller in Auto mode, step change to set-point) speed (time constant) in comparison to open loop response (controller in Man mode, step change to output) speed. Closed loop time constant = LAMBDA x PROCESS_TIME_CONSTANT.

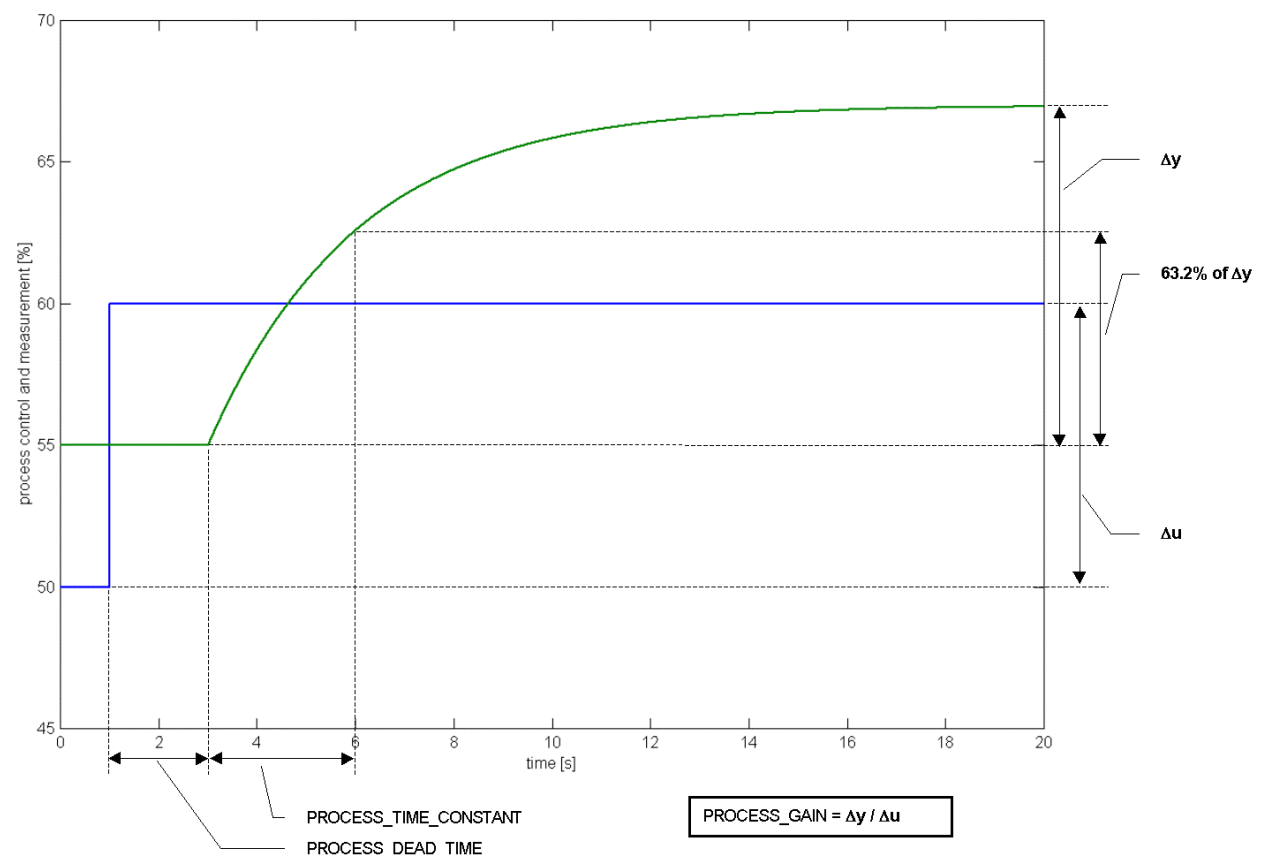


Figure 14. First order and dead time process step response.

PROPOSED_xxx

- PROPOSED_ALGORITHM_TYPE - Switch to select between standard (PID) and predictive (PPID) algorithm.
- PROPOSED_GAIN - Dimensionless value used by the block algorithm in calculating the block output.
- PROPOSED_RESET - The integral time constant, in seconds per repeat.
- PROPOSED_RATE - Defines the derivative time constant, in seconds.
- PROPOSED_DERIVATIVE_GAIN_LIMIT - Maximum gain of derivative part is limited to the value GAIN x DERIVATIVE_GAIN_LIMIT.
- PROPOSED_DERIVATIVE_ACTION_SRC - Selector for the signal in derivative calculation.
- PROPOSED_DEAD_TIME_COMPENSATION - The estimated value of process dead time in seconds. Maximum value is configured PID sample time x 200.
- PROPOSED_LAMBDA - Closed loop response speed in comparison to open loop response speed. Closed loop time constant = LAMBDA x PROCESS_TIME_CONSTANT.

ACTIVATE_PROPOSED_TUNING_PARAM

Activates proposed tuning parameters. When "Activate" is selected, proposed tuning parameters are activated (copied over actual tuning parameters). After that, ACTIVATE_PROPOSED_TUNING_PARAM changes to "Activated" indicating successful operation. Whenever PROPOSED_TUNING_PARAMETERS differ from actual tuning parameters, "Unactivated" is indicated.

ALGORITHM_TYPE - Switch to select between standard (PID) and predictive (PPID) algorithm. PID controller can be configured to operate either as standard PID controller or advanced predictive PID controller based on lambda structure (PPID). PPID controller is specially designed to provide robust control for processes with relatively long or varying dead time. The control algorithm used in the PPID case is presented in block diagram form in figure 15. Similar controller structure is used also in the case of standard PID algorithm. See PPID_LICENCE, page 13.

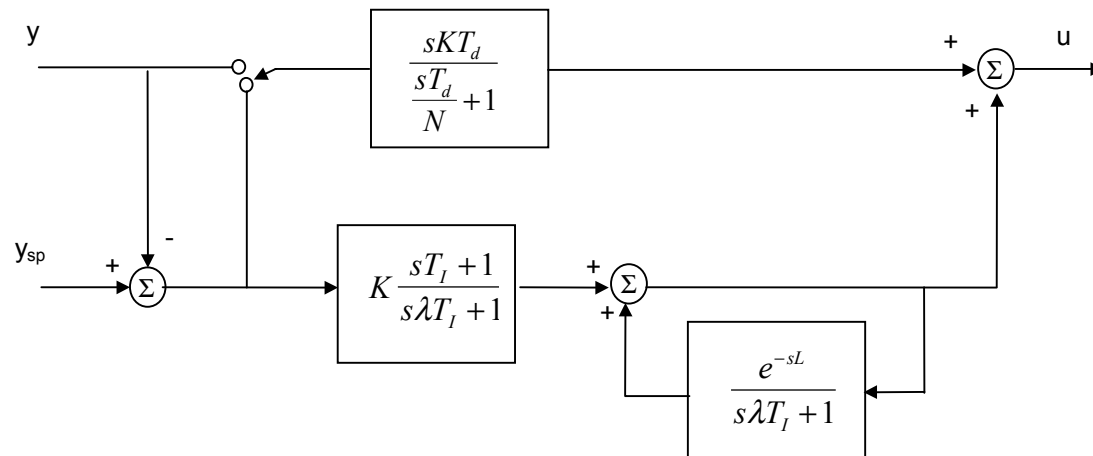


Figure 15. Block diagram of the control algorithm.

GAIN - Controller gain, dimensionless.

RESET - The integral time constant, in seconds per repeat.

RATE - Defines the derivative time constant, in seconds.

DERIVATIVE_GAIN_LIMIT

Maximum gain of derivative part is limited to the value GAIN x DERIVATIVE_GAIN_LIMIT.

DERIVATIVE_ACTION_SOURCE

Selector for the signal in derivative calculation. Options are.

- Measurement
- Control error

DEAD_TIME_COMPENSATION

The estimated value of process dead time in seconds. Maximum value is scheduled PID sample time x 200. This parameter is additional tuning parameter of the predictive controller (PPID).

LAMBDA

Closed loop response speed in comparison to open loop response speed. Closed loop time constant = LAMBDA x PROCESS_TIME_CONSTANT. This parameter is additional tuning parameter of the predictive controller (PPID).

PV_FTIME

Time constant of a single exponential filter between IN and PV, in seconds.

BAL_TIME

This parameter is used to specify the time constant (in seconds) at which the integral term will move to obtain balance when the output is limited and the mode is Auto, Cas, or RCas.

4.5.2.10 Alerts

To enable alert reporting, select FEATURE_SEL "Reports", page 15.

ALERT_KEY

The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.

BLOCK_ERR

See paragraph "Parameters common to all blocks", page 10.

BLOCK_ALM

See paragraph "Parameters common to all blocks", page 10.

UPDATE_EVT

See paragraph "Parameters common to all blocks", page 11.

ACK_OPTION

Selection of whether alarms associated with the function block will be automatically acknowledged.

ALARM_HYS

Amount the PV must return within the alarm limits before the alarm condition clears. Alarm Hysteresis is expressed as a percent of the PV span.

HI_HI_LIM, HI_LIM, LO_LIM, LO_LO_LIM, DV_LO_LIM, DV_HI_LIM

The setting for alarm limit in engineering units.

HI_HI_PRI, HI_PRI, LO_PRI, LO_LO_PRI, DV_LO_PRI, DV_HI_PRI

Priority of the alarm.

- 0 = the associated alert may clear when the priority is changed to 0, but it will never occur.
- 1 = the associated alert is not sent as a notification. If the priority is above 1, then the alert must be reported.
- 2 = reserved for alerts that do not require the attention of a plant operator, e.g. diagnostic and system alerts. Block alarm, error alarm, and update event have a fixed priority of 2.
- 3-7 = increasing higher priorities - advisory alarms.
- 8-15 = increasing higher priority - critical alarms.

HI_HI_ALM

The status for high high alarm and its associated time stamp. The high high alarm is generated when the PV value crosses the HI_HI_LIM value. The ALARM_HYS is valid here.

- Unacknowledged - A discrete enumeration which is set to Unacknowledged when an update occurs, and set to Acknowledged by a write from a human interface device or other entity which can acknowledge that the alarm has been noticed.
- Alarm State - A discrete enumeration which gives an indication of whether the alert has been reported.
- Time Stamp - The time when evaluation of the block was started and a change in alarm state was detected that is unreported. The time stamp value will be maintained constant until alert confirmation has been received - even if another change of state occurs.
- Subcode - In this case, always "Other".
- Value - The value of the associated parameter at the time the alert was detected.

HI_ALM

The status for high alarm and its associated time stamp. See HI_HI_ALM.

LO_ALM

The status for lo alarm and its associated time stamp. See HI_HI_ALM.

LO_LO_ALM

The status for lo lo alarm and its associated time stamp. See HI_HI_ALM.

DV_LO_ALM

The status for deviation low and its associated time stamp. The deviation low alarm is generated when the difference between SP and PV is crosses the DV_LO_LIM value. The ALARM_HYS is valid here. See HI_HI_ALM.

DV_HI_ALM

The status for deviation high and its associated time stamp. See DV_LO_ALM.

ALARM_SUM

The current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the function block.

- Current - The active status of each alarm.
- Unacknowledged - The unacknowledged state of each alarm.
- Unreported - The unreported status of each alarm.
- Disabled - The disabled state of each alarm.

4.5.3 Parameter access table

View object is a package which contains a predefined set of parameters. It is more efficient to read view objects from the device than polling the parameters separately. There are four different types of view objects

- VIEW_1 - View object defined to access the dynamic operating parameters of a block.
- VIEW_2 - View object defined to access the static operating parameters of a block.
- VIEW_3 - View object defined to access all dynamic parameters of a block. More than one View 3 object may be required to represent all dynamic parameters of the block.
- VIEW_4 - View object defined to access static parameters not included in VIEW_2. More than one VIEW_4 object may be required to represent these other static parameters of the block.

Table 6. PID controller block parameter access table.

Index	Parameter Mnemonic	VIEW_1	VIEW_2	VIEW_3	VIEW_4	VIEW_4.1
1	ST_REV	2	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	PV	5		5		
8	SP	5		5		
9	OUT	5		5		
10	PV_SCALE		11			
11	OUT_SCALE		11			
12	GRANT_DENY		2			
13	CONTROL_OPTS				2	
14	STATUS_OPTS				2	
15	IN			5		
16	PV_FTIME				4	
17	BYPASS		1			
18	CAS_IN	5		5		
19	SP_RATE_DN				4	
20	SP_RATE_UP				4	
21	SP_HI_LIM		4			
22	SP_LO_LIM		4			
23	GAIN				4	
24	RESET				4	
25	BAL_TIME				4	
26	RATE				4	
27	BKCAL_IN			5		
28	OUT_HI_LIM		4			
29	OUT_LO_LIM		4			
30	BKCAL_HYS				4	
31	BKCAL_OUT			5		
32	RCAS_IN			5		
33	ROUT_IN			5		
34	SHED_OPT				1	
35	RCAS_OUT			5		
36	ROUT_OUT			5		
37	TRK_SCALE				11	
38	TRK_IN_D	2		2		
39	TRK_VAL	5		5		
40	FF_VAL			5		
41	FF_SCALE				11	
42	FF_GAIN				4	
43	UPDATE_EVT					
44	BLOCK_ALM					
45	ALARM_SUM	8		8		
46	ACK_OPTION				2	
47	ALARM_HYS				4	
48	HI_HI_PRI				1	
49	HI_HI_LIM				4	
50	HI_PRI				1	
51	HI_LIM				4	
52	LO_PRI				1	
53	LO_LIM				4	
54	LO_LO_PRI				1	
55	LO_LO_LIM				4	
56	DV_HI_PRI				1	
57	DV_HI_LIM				4	
58	DV_LO_PRI				1	
59	DV_LO_LIM				4	
60	HI_HI_ALM					
61	HI_ALM					
62	LO_ALM					

Function blocks / PID controller function block

Index	Parameter Mnemonic	VIEW_1	VIEW_2	VIEW_3	VIEW_4	VIEW_4.1
63	LO_LO_ALM					
64	DV_HI_ALM					
65	DV_LO_ALM					
66	DERIVATIVE_ACTION_SOURCE					2
67	DERIVATIVE_GAIN_LIMIT					4
68	ALGORITHM_TYPE					2
69	DEAD_TIME_COMPENSATION					4
70	LAMBDA					4
71	TUNING_PROCESS_GAIN					4
72	TUNING_PROCESS_TIME_CONSTANT					4
73	TUNING_PROCESS_DEAD_TIME					4
74	TUNING_METHOD					2
75	PROPOSED_ALGORITHM_TYPE					2
76	PROPOSED_GAIN					4
77	PROPOSED_RESET					4
78	PROPOSED_RATE					4
79	PROPOSED_DERIVATIVE_GAIN_LIMIT					4
80	PROPOSED_DERIVATIVE_ACTION_SOURCE					2
81	PROPOSED_DEAD_TIME_COMPENSATION					4
82	PROPOSED_LAMBDA					4
83	ACTIVATE_PROPOSED_TUNING_PARAMETERS					1
84	SP_RAMP					4
85	TUNE_IN			4		
86	CURRENT_TIME			8		
	Totals (bytes)	43	43	95	104	65

5 Parameter attribute definitions

Parameter attributes are defined in table 7.

Table 7. Parameter attribute table.

Parameter Mnemonic	Standard / Man Specific	Class	Obj Type	Data Type/ structure	Store	Valid Range	Units	Spec default	MA default	Permission	Mode	Other	Range check
ACK_OPTION	Standard	C/Alarm	S	Bit String	S	0:auto ack disabl 1: auto ack enabl	na	0	0	ALARM			
ACT_FAIL_ACTION	Standard	C/Service	S	Uint8	S		E	Undefined	Undefined				
ACT_MAN_ID	Standard	C/Service	S	Uint32	N	Same as MANU-FAC_ID	E	000e05	000e05				
ACT_MODEL_NUM	Standard	C/Service	S	Visible String	N		none	-/-	-/-				
ACT_PRESSURE	Man Specific	C/Dynamic/Diagnostic	S	float	D		bar					Read only	
ACT_SN	Standard	C/Service	S	Visible String	N		none	-/-	-/-				
ACT_TYPE	Man Specific	C/Service	S	Uint8	S		E	Undefined	Undefined				
ACTIVATE_PROPOSED_TUNING_PARAMETER	Man Specific	C/Tune	S		S		E	0	0				
ACTUATOR_FULL_STROKES	Man Specific	C/Dynamic/Diagnostic	S	float	D		none					Read only	
ACT_FULL_STROKES_WARNING_LIMIT	Man Specific	C/Dynamic/Diagnostic	S	Uint32	S	0 - 4294967295	none	250 000	250 000				
ACTUATOR_REVERSALS	Man Specific	C/Dynamic/Diagnostic	S	Uint32	D		none					Read only	
ACTUATOR_REVERSALS_WARNING_LIMIT	Man Specific	C/Dynamic/Diagnostic	S	Uint32	S	0 - 4294967295	none	1 000 000	1 000 000				
ALARM_HYS	Standard	C/Alarm	S	Float	S	0 to 50 %	%	0.5%	0.5 %	ALARM		Positive	Yes
ALARM_SUM	Standard	C/Alarm	R	DS-74	mix		na			ALARM			
ALERT_KEY	Standard	C/Alarm	S	Uint8	S	1 to 255	none	0					Yes
ALGORITHM_TYPE	Man Specific	C/Operate	S	Uint16	S		E	0	Predictive				
AUTO_TRAVEL_CALIBRATION	Man Specific	C/Service	S	Uint8	S		E	Idle	Idle				
BAL_TIME	Standard	C/Operate	S	Float	S	Positive	Sec	0	0	TUNE		Positive	
BKCAL_HYS	Standard	C/Operate	S	Float	S	0 to 50 %	%	0.5%	0.5 %	TUNE		Positive	Yes
BKCAL_IN	Standard	I	R	DS-65	N		OUT						
BKCAL_OUT	Standard	O/Dynamic	R	DS-65	D		PV					Read only	
BLOCK_ALM	Standard	C/Alarm	R	DS-72	D		na					Read only	
BLOCK_ERR	Standard	C/Alarm	S	Bit String	D		E					Read only	
BYPASS	Standard	C/Operate	S	Uint8	S	1: Off, 2: On	E	0	1		Man		
CAS_IN	Standard	I	R	DS-65	D		PV						
CHANNEL	Standard	C	S	Uint16	S	1	none	0	1		O/S		Yes
CLR_FSTATE	Standard	C/Operate	S	Uint8	D	1:Off, 2:On	E	0	0	Note 9			
COLLECTION_DIRECTORY	Standard	C	A	Array of Uint 32	N		none					Read only	

Parameter attribute definitions

Parameter Mnemonic	Standard / Man Specific	Class	Obj Type	Data Type/ structure	Store	Valid Range	Units	Spec default	MA default	Permission	Mode	Other	Range check
CONFIRM_TIME	Standard	C/Alarm	S	Uint32	S		1/32	72000	72000	TUNE			Yes
CONTROL_OPTS	Standard	C/Operate	S	Bit String	S		na	0	2		O/S		
CURRENT_TIME	Man Specific	C	S	DS-21	D							DD Hidden	
CYCLE_SEL	Standard	C/Service	S	Bit String	S		na	0	0	Note 10			
CYCLE_TYPE	Standard	C/Service	S	Bit String	S	Set by mfr	na		Scheduled			Read only	
DD_RESOURCE	Standard	C	S	Visible String	S		na	null	null			Read only	
DD_REV	Standard	C	S	Uint8	S	Set by mfr	none		NC spec			Read only	
DEAD_ANGLE_COMP	Man Specific	C/Service	S	float	S	0-99	%	0	0				Yes
DEAD_TIME_COMPENSATION	Man Specific	C/Tune	R	Float	S	0 to 200 x PID period of execution		0	0			See Note 3	
DERIVATIVE_ACTION_SOURCE	Man Specific	C/Tune	S	Uint16	S		E	0	0				
DERIVATIVE_GAIN_LIMIT	Man Specific	C/Tune	S	Float	S	$8 \leq \text{DERIVATIVE_GAIN_LIMIT} \leq 20$		8	8				
DEV_REV	Standard	C	S	Uint8	S	Set by mfr	none		NC spec			Read only	
DEV_TEMPERATURE	Man Specific	C/Dynamic/Diagnostic	S	float	D		celsius					Read only	
DEV_TYPE	Standard	C	S	Uint16	S	Set by mfr	none					Read only	
DEVICE_PCB_SN	Man specific	C/Service	S	Uint 32	N		none					Read only	
DV_HI_ALM	Standard	C/Alarm	R	DS-71	D		PV					Read only	
DV_HI_LIM	Standard	C/Alarm	S	Float	S	0 to PV span, +INF	PV	+INF	+INF	ALARM			Yes
DV_HI_PRI	Standard	C/Alarm	S	Uint8	S	0 to 15	none	0	3	ALARM			Yes
DV_LO_ALM	Standard	C/Alarm	R	DS-71	D		PV					Read only	
DV_LO_LIM	Standard	C/Alarm	S	Float	S	-INF, -PV span to 0	PV	-INF	-INF	ALARM			Yes
DV_LO_PRI	Standard	C/Alarm	S	Uint8	S	0 to 15	none	0	3	ALARM			Yes
ELAPSED_TREND_TIME	Man Specific	C/Dynamic/Diagnostic	S	float	D		h					Read only	
EVENT_LOG_COUNT_TABLE	Man Specific	C/Dynamic/Diagnostic	A	20 x Uint8	S		none					Read only	
EVENT_LOG_EVENT_TABLE	Man Specific	C/Dynamic/Diagnostic	A	20 x Uint16	S		E					Read only	
EVENT_LOG_TIME_TABLE_1	Man Specific	C/Dynamic/Diagnostic	A	10 x Time value	S							Read only	
EVENT_LOG_TIME_TABLE_2	Man Specific	C/Dynamic/Diagnostic	A	10 x Time value	S							Read only	
POSITIONER_FAIL_ACTION	Man Specific	C/Service	S	Uint8	S		E	Close	Close				
FAULT_STATE	Standard	C/Dynamic	S	Uint8	N	1:Clear, 2:Active	E					Read only	
FEATURE_SEL	Standard	C/Service	S	Bit String	S		na	0	See Note 7				
FEATURES	Standard	C/Service	S	Bit String	S	Set by mfr	na		See Note 6			Read only	
FF_GAIN	Standard	C/Tune	S	Float	S		none	0	0	TUNE	Man		

Parameter attribute definitions

Parameter Mnemonic	Standard / Man Specific	Class	Obj Type	Data Type/ structure	Store	Valid Range	Units	Spec default	MA default	Permission	Mode	Other	Range check
FF_SCALE	Standard	C	R	DS-68	S		FF	0-100%	0-100%				
FF_VAL	Standard	I	R	DS-65	N		FF					Read only	
FILL_SOURCE_CHARACTERISTICS_TBL	Man Specific	C/Tune	S	Uint 8	S		E	none	none				Yes
FILL_TARGET_CHARACTERISTICS_TBL	Man Specific	C/Tune	S	Uint 8	S		E	none	none				Yes
FINAL_POSITION_VALUE	Standard	C/Dynamic	R	DS- 65	N		FVR					Read only	
FINAL_VALUE	Standard	C/Dynamic	R	DS- 65	N		FVR	0	0			See note 1	
FINAL_VALUE_CUTOFF_HI	Standard	C/Service	S	Float	S	FINAL_VALUE_RANGE, +INF	FVR	100	100		O/S		
FINAL_VALUE_CUTOFF_LO	Standard	C/Service	S	Float	S	FINAL_VALUE_RANGE, -INF	FVR	2	2		O/S		
FINAL_VALUE_RANGE	Standard	C/Service	R	DS- 68	S	0 - 100 %	%						
FINAL_VALUE_RANGE.DECIMAL	Standard	C/Service	S	Integer8	S			2	2				
FINAL_VALUE_RANGE.HIGH	Standard	C/Service	S	float	S	0 - 100	%	100	100				
FINAL_VALUE_RANGE.LO	Standard	C/Service	S	float	S	0 - 100	%	0	0				
FINAL_VALUE_RANGE.UNITS_INDEX	Standard	C/Service	S	Uint16	S	1342	none	1342	1342			Read only	
FREE_SPACE	Standard	C/Diagnostic	S	Float	D	0 - 100 %	%					Read only	
FREE_TIME	Standard	C/Diagnostic	S	Float	D	0 - 100 %	%					Read only	
FSTATE_TIME	Standard	C/Service	S	Float	S	Positive	Sec	0	0			Posit	Yes
FSTATE_VAL	Standard	C/Service	S	Float	S	PV_SCALE+/-10%	PV	0	0				Yes
GAIN	Standard	C/Tune	S	Float	S		none	0	0.5	TUNE			
GRANT_DENY	Standard	C/Service	R	DS-70	N		na	0	0				
HARD_TYPES	Standard	C/Service	S	Bit String	S	Set by mfr	na					Read only	
HARDWARE_REVISION	Man specific	C/Service	S	Visible String 32	N		none					Read only	
HI_ALM	Standard	C/Alarm	R	DS-71	D		PV					Read only	
HI_HI_ALM	Standard	C/Alarm	R	DS-71	D		PV					Read only	
HI_HI_LIM	Standard	C/Alarm	S	Float	S	PV_SCALE, +INF	PV	+INF	+INF	ALARM			Yes
HI_HI_PRI	Standard	C/Alarm	S	Uint8	S	0 to 15	none	0	8	ALARM			Yes
HI_LIM	Standard	C/Alarm	S	Float	S	PV_SCALE, +INF	PV	+INF	+INF	ALARM			Yes
HI_PRI	Standard	C/Alarm	S	Uint8	S	0 to 15	none	0	3	ALARM			Yes
IN	Standard	I	R	DS-65	N		PV					Read only	
IO_OPTS	Standard	C/Service	S	Bit String	S		na	0	0				
LAMBDA	Man Specific	C/Tune	R	Float	S	0.5 ≤ LAMBDA ≤ 5		0	1			See Note 3	
LAST_VALVE_TEST_DURATION	Man Specific	C/Service	S	float	S		s					Read only	

Parameter attribute definitions

Parameter Mnemonic	Standard / Man Specific	Class	Obj Type	Data Type/ structure	Store	Valid Range	Units	Spec default	MA default	Permission	Mode	Other	Range check
LAST_VALVE_TST_LOOP_OR_ST EP_SIZE	Man Specific	C/Service	S	float	S	0.1 - 100	%					Read only	
LAST_VALVE_TEST_POSITION_T ABLE_1	Man Specific	C/Service	A	25 x float	S		%					Read only	
LAST_VALVE_TEST_POSITION_T ABLE_2	Man Specific	C/Service	A	25 x float	S		%					Read only	
LAST_VALVE_TEST_SP_OR_TIME TBL_1	Man Specific	C/Service	A	25 x float	S		s, %					Read only	
LAST_VALVE_TEST_SP_OR_TIME TBL_2	Man Specific	C/Service	A	25 x float	S		s, %					Read only	
LAST_VALVE_TEST_STARTING_S P	Man Specific	C/Service	S	float	S	0 - 100	%					Read only	
LAST_VALVE_TEST_TYPE	Man Specific	C/Service	S	Uint16	S		E					Read only	
LIM_NOTIFY	Standard	C/Alarm	S	Uint8	S	0 to MAX_NOTIFY	none	MAX_NOTIFY	MAX_NOTIFY	TUNE			Yes
LO_ALM	Standard	C/Alarm	R	DS-71	D		PV					Read only	
LO_LIM	Standard	C/Alarm	S	Float	S	-INF, PV_SCALE	PV	-INF	-INF	ALARM			Yes
LO_LO_ALM	Standard	C/Alarm	R	DS-71	D		PV					Read only	
LO_LO_LIM	Standard	C/Alarm	S	Float	S	-INF, PV_SCALE	PV	-INF	-INF	ALARM			Yes
LO_LO_PRI	Standard	C/Alarm	S	Uint8	S	0 to 15	none	0	8	ALARM			Yes
LO_PRI	Standard	C/Alarm	S	Uint8	S	0 to 15	none	0	3	ALARM			Yes
LOAD_FACTOR_ALARM_LIMIT	Man Specific	C/Diagnostic	S	float	S		%	85	85				
LOAD_FACTOR_TREND_FACTOR TABLE	Man Specific	C/Dynamic/Diagnostic	A	20 x float	D		%					Read only	
LOAD_FACTOR_TREND_TIME_T A BLE	Man Specific	C/Dynamic/Diagnostic	A	20 x float	D		h					Read only	
LOAD_FACTOR_WARNING_LIMIT	Man Specific	C/Diagnostic	S	float	S		%	50	50				
MANUFAC_ID	Standard	C	S	Uint32	S	E, 0xE05	none		0xE05			Read only	
MAX_NOTIFY	Standard	C/Alarm	S	Uint8	S	Set by mfgr	none		15			Read only	
MEMORY_SIZE	Standard	C/Service, Diagnostic	S	Uint16	S	Set by mfgr	Kbytes					Read only	
MIN_CYCLE_T	Standard	C/Service	S	Uint32	S	Set by mfgr	1/32 msec					Read only	
MODE_BLK	Standard	C/Operate	R	DS-69	mix	See MODE	na	O/S	See Note 5	See Note 1		Note 2	
NV_CYCLE_T	Standard	C	S	Uint32	S		1/32 msec					Read only	
OPERATION_TIME	Man Specific	C/Dynamic/Diagnostic	S	float	D		h					Read only	
OUT	Standard	O	R	DS-65	N		OUT			See Note 1	Man		Yes
OUT	Standard	O/Dynamic	R	DS-65	N	OUT_SCALE ±10%	OUT			See Note 1	Man		Yes
OUT_HI_LIM	Standard	C/Alarm	S	Float	S	OUT_SCALE ±10%	OUT	100	100			Ordered. low	Yes
OUT_LO_LIM	Standard	C/Alarm	S	Float	S	OUT_SCALE ±10%	OUT	0	0			Ordered. high	Yes
OUT_SCALE	Standard	C/Operate	R	DS-68	S		OUT	0-100%	0-100%		Man		

Parameter attribute definitions

Parameter Mnemonic	Standard / Man Specific	Class	Obj Type	Data Type/ structure	Store	Valid Range	Units	Spec default	MA default	Permission	Mode	Other	Range check
PAR_DEFAULTS_TYPE	Man specific	C/Service	S	Uint8	S		E						
POS_SENSOR_ROT	Man Specific	C/Service	S	Uint8	S		E	Standard : clockwise to Close	Standard : clockwise to Close				
POS_SENSOR_SN	Man Specific	C/Service	S	Uint32	N		none	0	0				
POS_SENSOR_TEMP_COMP	Man Specific	C/Service	S	Uint8	N		E	off	off				
POS_SENSOR_TEMP_COMP_MAX_TABLE	Man Specific	C/Service	A	26 x float	N	0.1 - 99.9	%	99.9	99.9				
POS_SENSOR_TEMP_COMP_MIN_TABLE	Man Specific	C/Service	A	26 x float	N	0.1 - 99.9	%	0.1	0.1				
PPID_LICENCE	Man specific	C/Service	S	Visible String 32	S	3*dev_id	none		empty				
PROPOSED_ALGORITHM_TYPE	Man Specific	C/Tune	R	Uint16	S	E		0	1				
PROPOSED_DEAD_TIME_COMPENSATION	Man Specific	C/Tune	R	Float	S	Positive		0	0			See Note 4	
PROPOSED_DERIVATIVE_ACTION_SRC	Man Specific	C/Tune	R	Uint16	S	E		0	0				
PROPOSED_DERIVATIVE_GAIN_LIMIT	Man Specific	C/Tune	R	Float	S	$8 \leq \text{DERIVATIVE_GAIN_LIMIT} \leq 20$		8	8				
PROPOSED_GAIN	Man Specific	C/Tune	R	Float	S	Positive		0	0				
PROPOSED_LAMBDA	Man Specific	C/Tune	R	Float	S	$0.5 \leq \text{LAMBDA} \leq 5$		1	1			See Note 4	
PROPOSED_RATE	Man Specific	C/Tune	R	Float	S	Positive		0	0				
PROPOSED_RESET	Man Specific	C/Tune	R	Float	S	Positive		+INF	+INF				
PV	Standard	C/Operate	R	DS-65	D		PV					Read only	
PV_FTIME	Standard	C/Operate	S	Float	S	Positive	Sec	0	0	TUNE		Positive	
PV_SCALE	Standard	C/Operate	R	DS-68	S		PV	0-100%	0-100%		Man		
RATE	Standard	C/Tune	S	Float	S	Positive	Sec	0	0	TUNE		Positive	
RCAS_IN	Standard	C/Dynamic	R	DS-65	N		PV						
RCAS_OUT	Standard	C/Dynamic	R	DS-65	D		PV					Read only	
READBACK	Standard	C	R	DS-65	D		XD					Read only	
RESET	Standard	C/Tune	S	Float	S	Positive	Sec	+INF	30	TUNE		Positive	
RESET_DIAGNOSTICS	Man Specific	C/Diagnostic	S	Uint8	S		E	0	0				
RESTART	Standard	C	S	Uint8	D		E			See Note 10			
ROUT_IN	Standard	C	R	DS-65	D		OUT						
ROUT_OUT	Standard	C/Dynamic	R	DS-65	D							Read only	
RS_STATE	Standard	C/Dynamic, Diagnostic	S	Uint8	D	See Part 1 for enumeration	E					Read only	
SERVO_GAIN	Standard	C/Tune	S	float	S	0.1-3.0	none	1	1				Yes
SERVO_PARAM_B	Man Specific	C/Tune	S	float	S	0.0-3.0	none	1	1				Yes
SERVO_PARAM_D	Man Specific	C/Tune	S	float	S	0.0-3.0	none	0	0				Yes
SET_FSTATE	Standard	C/Operate	S	Uint8	D	1:Off, 2:Set	E	1	1	See Note 9			
SHED_OPT	Standard	C/Operate	S	Uint8	S	See SHED_OPT	E	0	1				
SHED_RCAS	Standard	C/Service	S	Uint32	S		1/32	640000	640000	TUNE			Yes

Parameter attribute definitions

Parameter Mnemonic	Standard / Man Specific	Class	Obj Type	Data Type/ structure	Store	Valid Range	Units	Spec de- fault	MA default	Permission	Mode	Other	Ran ge chec k
SHED_ROUT	Standard	C/Service	S	Uint32	S		msec						
SIGNAL_MOD_TABLE	Man Specific	C/Tune	R	21 x float + 1 x Uint 8	S	0 -100	%	0	0	TUNE			Yes
SIG- NAL_MOD_TABLE.CHARACTERIZ ATION	Man Specific	C/Tune	S	Uint 8	S		E	Uninitialized	off				Yes
SIMULATE	Standard	C	R	DS-82	D		none	disable	disable				
SIMULATION_SWITCH	Man specific	C/Service	S	Uint8	D		E					Read only	
SOFTWARE_REVISION	Man specific	C/Service	S	Visible String 32	N		none					Read only	
SOURCE_CHARACTERISTICS_TA BLE	Man Specific	C/Tune	A	21 x float	S	0-1	none	0	0				Yes
SP	Standard	C/Operate	R	DS-65	N	PV_SCALE ± 10%	PV			See Note 1			
SP_HI_LIM	Standard	C/Operate	S	Float	S	PV_SCALE ± 10%	PV	100	100			Ordered, low	Yes
SP_LO_LIM	Standard	C/Operate	S	Float	S	PV_SCALE ± 10%	PV	0	0			Ordered, high	Yes
SP_RAMP	Man Specific	C/Operate	S	Float	S	Positive		0	0				
SP_RATE_DN	Standard	C/Operate	S	Float	S	Positive	PV/Sec	+INF	+INF			Positive	
SP_RATE_UP	Standard	C/Operate	S	Float	S	Positive	PV/Sec	+INF	+INF			Positive	
ST_REV	Standard	C/Dynamic	S	Uint16	S		none	0	0			Read only	
STATUS_OPTS	Standard	C/Service	S	Bit String	S		na	0	0		O/S		
STRATEGY	Standard	C	S	Uint16	S		none	0	0				
SUPPLY_PRESSURE	Man Specific	C/Service	S	float	S	0.1-15	bar	5	5				Yes
TAG_DESC	Standard	C	S	Octet String	S		na	nulls	nulls				
TAR- GET_CHARACTERISTICS_TABLE	Man Specific	C/Tune	A	21 x float	S	positive	none	0	0				Yes
TEST_RW	Standard	C/Diagnostic	R	DS-85	D		none						
TRANSDUCER_DIRECTORY	Standard	C	A	Array of Uint 16	N		none					Read only	
TRANSDUCER_TYPE	Standard	C	S	Uint16	N		E					Read only	
TRAVEL_DEVIATION_ALARM_LIMI T	Man Specific	C/Diagnostic	S	float	S		%	10	10				
TRAVEL_DEVIATION_TREND_ER ROR_TBL	Man Specific	C/Dynamic/Di agnostic	A	20 x float	D		%					Read only	
TRAVEL_DEVIATION_TREND_TIM E_TBL	Man Specific	C/Dynamic/Di agnostic	A	20 x float	D		h					Read only	
TRAVEL_DEVIATION_WARNING_L IMIT	Man Specific	C/Diagnostic	S	float	S		%	2	2				
TRK_IN_D	Standard	I	R	DS-66	N		On/Off					Read only	
TRK_SCALE	Standard	C/Scaling	R	DS-68	S		TRK	0-100%	0-100%		Man		
TRK_VAL	Standard	I	R	DS-65	N		TRK					Read only	
TUNE_IN	Man Specific	C	S	Float	D								
TUNING.PROCESS_DEAD_TIME	Man Specific	C/Tune	R	Float	S	Positive		0	0				

Parameter attribute definitions

Parameter Mnemonic	Standard / Man Specific	Class	Obj Type	Data Type/ structure	Store	Valid Range	Units	Spec default	MA default	Permission	Mode	Other	Range check
TUNING.PROCESS_GAIN	Man Specific	C/Tune	R	Float	S	Positive		0	0				
TUNING.PROCESS_TIME_CONSTANT	Man Specific	C/Tune	R	Float	S	Positive		0	0				
TUNING.TUNING_METHOD	Man Specific	C/Tune	R	Uint16	S	E		1	1				
UPDATE_EVT	Standard	C/Alarm	R	DS-73	D		na					Read only	
VALVE_FULL_STROKES	Man Specific	C/Dynamic/Diagnostic	S	float	D		none					Read only	
VALVE_FULL_STROKES_WARNING_LIMIT	Man Specific	C/Dynamic/Diagnostic	S	Uint32	S	0 - 4294967295	none	250 000	250 000				
VALVE_MAN_ID	Standard	C/Service	S	Uint32	N	Same as MANUFAC_ID	E	000e05	000e05				
VALVE_MODEL_NUM	Standard	C/Service	S	Visible String	N		none	-/-	-/-				
VALVE_REVERSALS	Man Specific	C/Dynamic/Diagnostic	S	Uint32	D		none					Read only	
VALVE_REVERSALS_WARNING_LIMIT	Man Specific	C/Dynamic/Diagnostic	S	Uint32	S	0 - 4294967295	none	1 000 000	1 000 000				
VALVE_SN	Standard	C/Service	S	Visible String	N		none	-/-	-/-				
VALVE_TEST_DURATION	Man Specific	C/Service	S	float	S	See Note 8	s	10	10				
VALVE_TEST_LOOP_OR_STEP_SIZE	Man Specific	C/Service	S	float	S	0.1 - 100	%	20	20				Yes
VALVE_TEST_START	Man Specific	C/Service	S	Uint8	S		E	idle	idle				
VALVE_TEST_STARTING_SETPOINT	Man Specific	C/Service	S	float	S	0 - 100	%	20	20				Yes
VALVE_TEST_TYPE	Man Specific	C/Service	S	Uint16	S		E	Step	Step				
VALVE_TRAVEL_VS_TIME_TREND	Man Specific	C/Dynamic/Diagnostic	A	10 x float	D		%					Read only	
VALVE_TYPE	Standard	C/Service	S	Uint8	S		E	Rotary	Rotary				
WRITE_ALM	Standard	C/Alarm	R	DS-72	D		none						
WRITE_LOCK	Standard	C/Service	S	Uint8	S	1:Unlocked, 2: Locked	E	1	1	Note 9			
WRITE_PRI	Standard	C/Alarm	S	Uint8	S	0 to 15	none	0	2	ALARM			Yes
XD_CAL_DATE	Standard	C/Service	S	Date	S		none	1.1.2000	1.1.2000		O/S		
XD_CAL_LOC	Standard	C/Service	S	Visible String	S		none	-/-	-/-		O/S		
XD_CAL_WHO	Standard	C/Service	S	Visible String	S		none	-/-	-/-		O/S		
XD_ERROR	Standard	C/Alarm	S	Uint8	D		E					Read only	
XD_ERROR_EXTENSION_PART_1	Man Specific	C/Alarm	R	Uint32	D		E					Read only	
XD_ERROR_EXTENSION_PART_2	Man Specific	C/Alarm	R	Uint16	D		E					Read only	
XD_SCALE	Standard	C/Tune	R	DS-68	S		XD	0-100%	0-100%				

Note 1. Normally, the operator has permission to write these values, but PROGRAM or LOCAL remove that permission and grant it to a supervisory computer or a local control panel.

Note 2. MODE_BLK has a mixture of storage types.

Note 3. Parameter is Read only if ALGORITHM_TYPE is Standard.

Note 4. Parameter is Read only if PROPOSED_TUNING_PARAMETERS.ALGORITHM_TYPE is Standard.

Note 5. MA default target modes; Resource Auto, Transducer Auto, AO O/S, PID O/S.

- Note 6. Supported features: Reports, Fault state, Hard write lock, Output readback
Note 7. Selected features: Reports, Fault state, Hard write lock, Output readback
Note 8. Hysteresis 500 - 2000 s, Step 2 - 500 s
Note 9. The operator can control PROGRAM or LOCAL access to these values.
Note 10. Changing these parameters may be fatal to communication

Table legend:

Parameter Mnemonic:

List of parameters associated with the blocks addressed on the form.

Standard / Man Specific:

Indicates either parameter is standard or manufacturer specific.

Class:

The class attribute of a variable specifies how the variable is used by a field device. This attribute is intended to be used by host applications to implement permission levels and to organize how the variables are displayed.

I - Function block input. The input may be connected to a function block output or used as a constant.

O - Function block output. An output may be referenced by other function block inputs.

C - Parameter value contained in the block, available for the interface (operation, diagnostic) and/or configuration.

Dynamic - Variables modified by a field device without stimulus from the fieldbus network.

Diagnostic - Variables indicating device status.

Service - Used when performing routine maintenance.

Operate - Block parameters manipulated to control a block's operation (for instance, set point).

Alarm - Block parameters containing alarm limits.

Tune - Block parameters used to tune the algorithm of a block.

Local - Variables used locally by host applications. Local variables are not stored in a field device, but they can be sent to a field device. For example, a local variable may be used to guide the menu structure, that is, the user edits a variable and based on that value a new menu is presented. In this case, the local variable is never sent to a field device.

Object Type:

S - Simple variable value

R - Record

A - Array of simple variables

Data Type:

Name - Simple variable or array

DS-n - Data structure (Record) of index n.

Store:

S - Static. Writing to the parameter changes the static revision counter ST_REV

N - Non-volatile parameter which must be remembered through a power cycle, but is not under the static update code.

D - Dynamic. The value is calculated by the block, or read from another block.

Valid Range:

Range of values the parameters should be restricted to for use in the function block. Zero is always allowed as the state of a bit in a bit string. Unless otherwise stated, zero is the inverse of the described value. Zero is allowed in enumeration's, although it always means that the value is invalid. This is required for initialization of an unconfigured block. Plus or minus infinity (+INF or -INF) may be included in the valid range to indicate that it is permissible to use them to turn off a limit comparison, such as an alarm limit. A display device should display the word "None" instead if trying to show infinity.

Units:

All parameters with the same Units name have the same DD units relationship. Units names are PV, OUT, FF, TRK, and XD. The names Sec (seconds), % (percent) and On/Off (discrete state) are constant units. E means enumerated. The word "none" is used for the numbers with no specified units, and "na" is used for bit strings where units do not apply.

Spec default:

When PAR_DEFAULTS_TYPE is "FF Spec defaults", indicated value is inserted when restart with defaults is requested.

MA default:

When PAR_DEFAULTS_TYPE is "Metso Automation defaults", indicated value is inserted when restart with defaults is requested.

Permission:

Defines the setting of the GRANT_DENY parameter that will allow write access to this parameter, for instance devices that obey this parameter. Note 1 shows those cases where write access may be removed by an active Grant setting.

Mode:

Lowest priority target mode required to allow a change to the parameter to be made. The actual mode must match the target mode, so that the block is not in another mode than that chosen by the operator. Scaling changes are protected by mode because the block may be using scaling to calculate its output.

Other:

DD handling for Positive, Ordered, and Read Only. If the parameter is an Input, it is read only if it is linked, but writeable if it is not linked.

Range Check:

Flag to check that the value is within the Valid Range given in the table.