Honeywell

Advanced Process Manager Control Functions and Algorithms

AP09-600

TotalPlant

Honeywell

Implementation

Advanced Process Manager - 1

Advanced Process Manager Control Functions and Algorithms

AP09-600
Release 650
12/03

TotalPlant

Notices and Trademarks

Revision 05 – December 5, 2003

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any indirect, special or consequential damages. The information and specifications in this document are subject to change without notice.

TotalPlant and TDC 3000 are U. S. registered trademarks of Honeywell Inc.

Other brand or product names are trademarks of their respective owners.

Contacts

World Wide Web

The following Honeywell Web sites may be of interest to Industry Solutions customers.

Honeywell Organization	WWW Address (URL)
Corporate	http://www.honeywell.com
Industry Solutions	http://www.acs.honeywell.com
International	http://content.honeywell.com/global/

Telephone

Contact us by telephone at the numbers listed below.

	Organization	Phone Number	
United States and Canada	Honeywell International Inc. Industry Solutions	1-800-343-0228 1-800-525-7439 1-800-822-7673	Sales Service Technical Support
Asia Pacific	Honeywell Asia Pacific Inc. Hong Kong	(852) 23 31 9133	
Europe	Honeywell PACE Brussels, Belgium	[32-2] 728-2711	
Latin America	Honeywell International Inc. Sunrise, Florida U.S.A.	(954) 845-2600	

Honeywell International
Process Solutions
2500 West Union Hills Drive
Phoenix, AZ 85027
1-800 343-0228

About This Publication

This publication supports **TotalPlant®** Solution (TPS) system network Release 640. TPS is the evolution of TDC 3000^X.

This publication defines the Advanced Process Manager I/O and control functions that are available in software Releases 500 - 600. Used as a reference manual for process engineers, control-system engineers, and application engineers who design and implement data-acquisition and control strategies for TPS systems with Local Control Networks and Universal Control Networks.

The user should be familiar with the system control functions described in *System Control Functions* in the *Implementation/Startup & Reconfiguration - 2* binder before using this publication.

Detailed descriptions of the parameters mentioned in this publication can be found in the *Advanced Process Manager Parameters Reference Dictionary*.

Table of Contents

1 INTRODUCTION

1.1 General Description
1.1.1 I/O Functions
1.1.2 Control Functions
1.2 Common Functions
1.2.1 Point Form
1.2.2 Associated Display
1.3 Alarming
1.3.1 Alarm Priorities
1.3.2 Alarm Enable Status
1.3.3 Contact Cutout
1.3.4 Primary Module Point
1.3.5 Auxiliary Units
1.4 Red Tagging
1.5 References

2 I/O FUNCTIONS

2.1 Overview
2.2 High and Low Level Analog Input Point
2.2.1 PV Characterization
2.2.2 PV Range Checking and Filtering
2.2.3 PV Source Selection
2.2.4 Alarming
2.2.4.1 Disable Input Alarming When Using Open-Line Detection
2.3 Smart Transmitter Interface Point
2.3.1 Parameter Comparisons
2.3.2 STI IOP Functions
2.3.3 Smart Transmitter Communication Modes
2.3.4 Database Considerations
2.3.5 Point States
2.3.6 STI IOP Commands
2.4 Analog Output Point
2.4.1 Direct/Reverse Output
2.4.2 Output Characterization
2.4.3 Calibration Compensation
2.4.4 Disable Output Alarming when Using Open-Line Detection
2.5 Digital Input Point
2.5.1 Status Digital Input Point
2.5.2 Latched Digital Input Point
2.5.3 Accumulation Digital Input Point
2.5.4 Sequence Of Events Digital Input IOP
2.6 Digital Output Point
2.6.1 Pulse Width Modulated (PWM) Output Type
2.6.2 Status Output Type
2.6.3 Initialization Request Flag
2.7 Pulse Input Point
2.7.1 Operation
2.8 Serial Interface
2.8.1 Operation
2.8.2 Configuration
2.8.3 Checkpointing and Startup

Table of Contents

2.9 Serial Device Interface
2.9.1 Serial Device Interface Description
2.9.2 Manual/Auto Station Interface
2.9.3 Toledo Weigh Scale Interface

3 CONTROL FUNCTIONS OVERVIEW

3.1 Control Performance
3.1.1 Control Slot Types
3.1.2 Scan Frequency
3.1.3 Determining Processing and Memory Capacity
3.1.4 Performance Monitoring
3.2 Database Considerations
3.3 Point Execution State
3.3.1 Run-Idle-Run Transition
3.3.2 Warm Cold Restart
3.4 Peer-to-Peer Communication
3.4.1 Implementation
3.4.2 Restrictions
3.5 Hardware Reference Addressing
3.5.1 Find Names Function
3.6 Redundancy
3.7 Point Reservation
3.8 Status Messages

4 DIGITAL COMPOSITE POINT

4.1 Description
4.2 Point States
4.2.1 Normal States
4.2.2 Momentary States
4.2.3 Moving/Bad States
4.3 Mode Attributes
4.4 Interlocks
4.4.1 Permissive Interlocks
4.4.2 Override Interlocks
4.4.3 Configurable Interlock Bypass
4.4.4 Safety Interlocks for State 0 (SI0)
4.5 Commanded State (OP)
4.6 Digital Outputs
4.6.1 Latched Output
4.6.2 Pulsed Output
4.7 Output Connections
4.7.1 Digital Output Point—Latched Output
4.7.2 Digital Output Point—Pulsed Output
4.7.3 Logic Slot Flags
4.7.4 APM Box PV Flags

Table of Contents

4.7.5 Device Control Slot Flags
4.7.6 Array Point Flags
4.8 Read-Back Check
4.9 Initialization Manual
4.10 Digital Inputs
4.11 Input Connections
4.11.1 Digital Input Point—PV
4.11.2 Logic Slot Output
4.11.3 Logic Slot Flags
4.11.4 APM Box PV Flags
4.11.5 Device Control Slot Flags
4.11.6 Array Point Flags
4.11.7 Process Module Slot Flags
4.12 Current Input State (PV)
4.13 PV Source
4.13.1 PV Source Option
4.14 Alarming
4.14.1 Command Disagree, Command Fail, and Uncommanded Change
4.14.2 Off-Normal
4.15 Change-of-State Events
4.16 Local Manual Indication
4.17 Maintenance Statistics
4.17.1 Reset and Redtag

5 LOGIC POINT

5.1 Functions
5.2 Configuring the Logic Point
5.2.1 Logic Input Connections
5.2.2 Flags
5.2.3 Numerics
5.2.4 Logic Blocks
5.2.5 Output Connections
5.2.6 Generic Descriptors
5.3 Logic Block Algorithms
5.3.1 Null (Null)
5.3.2 AND Gate (AND)
5.3.3 OR Gate (OR)
5.3.4 NOT Gate (NOT)
5.3.5 NAND Gate (NAND)
5.3.6 NOR Gate (NOR)
5.3.7 XOR Gate (XOR)
5.3.8 Qualified OR Gate with 2 Inputs ON (QOR2)
5.3.9 Qualified OR Gate with 3 Inputs ON (QOR3)
5.3.10 Switch (SWITCH)
5.3.11 Compare Equal with Deadband (CE)
5.3.12 Compare Not Equal with Deadband (NE)
5.3.13 Compare Greater Than with Deadband (GT)
5.3.14 Compare Greater Than or Equal with Deadband (GE)
5.3.15 Compare Less Than with Deadband (LT)

Table of Contents

5.3.16 Compare Less Than or Equal with Deadband (LE)
5.3.17 Check for Bad (CHECKBAD)
5.3.18 Fixed-Size Pulse (PULSE)
5.3.19 Pulse with Maximum Time Limit (MAXPULSE)
5.3.20 Pulse with Minimum Time Limit (MINPULSE)
5.3.21 Delay (DELAY)
5.3.22 On Delay (ONDLY)
5.3.23 Off Delay (OFFDLY)
5.3.24 Watchdog Timer (WATCHDOG)
5.3.25 Flip-Flop (FLIPFLOP)
5.3.26 Change Detect (CHDETECT)
5.4 Logic Block Alarms
5.4.1 Configurable Alarms

6 PROCESS MODULE DATA POINTS

6.1 Purpose of the Process Module Data Point
6.1.1 CL/APM Sequence Program
6.1.2 Sequence Program/Unit Partitioning
6.1.3 Writing the Sequence Programs
6.1.4 Sequence Program Structure
6.1.5 Sequence Execution
6.1.6 Process Module Operating State
6.1.7 Sequence Execution Mode
6.1.8 Sequence Execution State
6.1.9 Sequence Overrides
6.1.10 Sequence Alarms
6.1.11 Sequence Messages
6.1.12 Restart Option

7 REGULATORY PV POINT

7.1 Features
7.2 PV Input Connections
7.3 PV Algorithm Calculation
7.4 PV Range Check and Filtering
7.4.1 PV Value Status
7.5 PV Source Selection
7.6 PV Alarm Detection
7.7 PV Algorithms
7.7.1 Data Acquisition (DATAACQ)
7.7.2 Flow Compensation (FLOWCOMP)
7.7.3 Middle of Three Selector (MIDOF3)
7.7.4 High Selector, Low Selector, Average (HILOAVG)
7.7.5 Summer (SUMMER)
7.7.6 Variable Dead Time with Lead-Lag Compensation (VDTLDLG)
7.7.7 Totalizer (TOTALIZR)
7.7.8 General Linearization (GENLIN)
7.7.9 Calculator (CALCULTR)

Table of Contents

8 **REGULATORY CONTROL POINT**

8.1 Functions
8.2 Control Input Connections
8.3 PV Source Selection (Pid, PosProp, and RatioCtl Algorithms)
8.4 Modes
8.4.1 Mode Attribute
8.4.2 Normal Mode
8.4.3 Normal Mode Attribute
8.4.4 Remote Cascade Handling (Pid Algorithms)
8.5 Safety Shutdown
8.6 External Mode Switching
8.7 Setpoint Handling
8.7.1 Setpoint (SP) Limits (Pid, PosProp, PidPosPr, and RatioCtl Algorithms)
8.7.2 SP Target-Value Processing (Pid, PosProp PidPosPr, and RatioCtl Algorithms)
8.7.3 PV Tracking (Pid Algorithms)
8.7.4 Deviation and Advisory Deviation Alarming
8.7.5 Bad Output Alarm
8.7.6 Ratio and Bias Options (Pid Algorithms)
8.7.7 SP Access Restrictions
8.8 Setpoint/Output (SP/OP) Tolerance Check
8.9 Alarm Detection
8.10 Control Output Processing
8.10.1 Initial Control Processing
8.10.2 Output High and Low Limits
8.10.3 Output Rate-of-Change Limits
8.10.4 Output Minimum-Change Limit
8.10.5 Output Limiting in Manual Mode
8.10.6 Control Output Connections
8.11 Initialization
8.11.1 PV Initialization
8.11.2 Control Initialization
8.12 Windup Protection
8.12.1 Windup Status
8.12.2 Status Propagation
8.13 Override Control
8.13.1 O/R Status and Feedback
8.13.2 Processing in an Override Control Strategy
8.13.3 Guidelines for Using Override Control
8.14 Control Algorithms
8.14.1 Proportional, Integral, Derivative (PID)
8.14.2 PID with Feedforward (PIDFF)
8.14.3 PID with External Reset-Feedback (PIDERFB)
8.14.4 Position Proportional Controller (POSPROP)
8.14.5 PID Position Proportional Controller (PIDPOSPR)
8.14.6 Ratio Control (RATIOCTL)
8.14.7 Ramp and Soak (RAMPSOAK)
8.14.8 Auto Manual (AUTOMAN)

Table of Contents

8.14.9 Incremental Summer (INCRSUM)
8.14.10 Switch (SWITCH)
8.14.11 Override Selector (ORSEL)

9 APM BOX FLAG, NUMERIC, STRING, TIME, AND TIMER GLOBAL VARIABLES

9.1 Global Variables
9.2 Box Flag Data Point
9.2.1 Alarming
9.3 Box Numeric Data Point
9.4 Box Timer Data Point
9.5 Box Time Variables
9.6 Box String Variables

10 ARRAY POINTS

10.1 Array Point Overview
10.1.1 Node Specific Configuration Requirements
10.1.2 Array Parameter Names
10.2 Array Point Use with APM Box Global Variables
10.2.1 Configuration
10.2.2 Array Point Example
10.3 Serial Interface to Array Point
10.3.1 Accessing SI Array Data
10.3.2 SI Array Point Configuration
10.3.3 Status and Error Checking
10.3.4 Read-Back Check for Device Control/Digital Composite Points
10.3.5 CL Notes

11 DEVICE CONTROL POINT

11.1 Features
11.2 General Description
11.3 Input Connections
11.3.1 Digital Inputs
11.3.2 Logic Inputs
11.3.3 Secondary Variable Input
11.3.4 Local Manual Input
11.4 Output Connections
11.4.1 Digital Outputs
11.4.2 Logic Output Connections
11.5 Alarms
11.6 Maintenance Statistics

INTRODUCTION
Section 1

This section contains an introduction to the Advanced Process Manager Control functions, and also provides references to other publications that are useful or necessary in implementing control system functions.

1.1 GENERAL DESCRIPTION

The Advanced Process Manager (APM) is designed to provide flexible and powerful process scanning and control capabilities. To do this, it uses a powerful multiprocessor architecture with separate microprocessors dedicated to perform specific tasks. As depicted in Figure 1-1, the APM consists of the Advanced Process Manager Module (APMM) and the I/O Subsystem.

The APMM consists of a Communication Processor and Modem, I/O Link Interface Processor, and Control Processor. APMM redundancy can be optionally provided. The Communication Processor is optimized to provide high performance network communications, handling such functions as network data access and peer-to-peer communications. The Control Processor is the APM resource dedicated to executing regulatory, logic, and sequence functions, including a powerful user programming facility. Because communication and I/O processing are performed by separate dedicated hardware, the full power of the control processor can be applied to control strategy implementation. The I/O Link Interface Processor is the interface to the I/O Subsystem.

The I/O Subsystem consists of a redundant I/O Link and the I/O Processors. These I/O Processors handle all field I/O for both data acquisition and control functions. The I/O Processors, for example, provide such functions as engineering unit conversion and alarm limit checking independent of the APMM. All control operations are performed within the APMM, with all data acquisition being performed in I/O Processors. The process engineer has complete flexibility of choice, within the maximum APM design limits, in the assignment of point types and control strategies. The interactive tools provided by both the **TotalPlant** Solution (TPS) System Universal Station and Universal Work Station are used to implement these selections. Refer to Section 31 in the *Engineer’s Reference Manual* for more information.

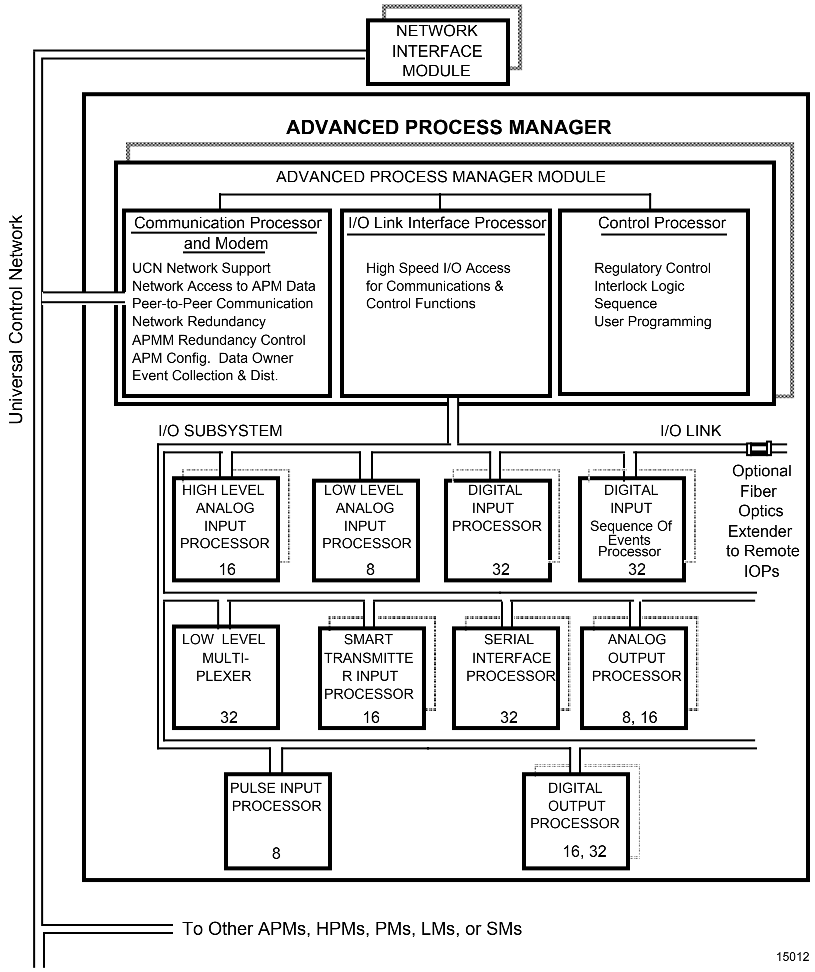

NETWORK INTERFACE MODULE
ADVANCED PROCESS MANAGER
ADVANCED PROCESS MANAGER MODULE
Communication Processor and Modem
UCN Network Support
Network Access to APM Data
Peer-to-Peer Communication
Network Redundancy
APMM Redundancy Control
APM Config. Data Owner
Event Collection & Dist.
I/O Link Interface Processor
High Speed I/O Access for Communications & Control Functions
Control Processor
Regulatory Control
Interlock Logic
Sequence
User Programming
Universal Control Network
I/O SUBSYSTEM
I/O LINK
Optional Fiber Optics Extender to Remote IOPs
HIGH LEVEL ANALOG INPUT PROCESSOR 16
LOW LEVEL ANALOG INPUT PROCESSOR 8
DIGITAL INPUT PROCESSOR 32
DIGITAL INPUT Sequence Of Events Processor 32
LOW LEVEL MULTI-PLEXER 32
SMART TRANSMITTE R INPUT PROCESSOR 16
SERIAL INTERFACE PROCESSOR 32
ANALOG OUTPUT PROCESSOR 8, 16
PULSE INPUT PROCESSOR 8
DIGITAL OUTPUT PROCESSOR 16, 32
To Other APMs, HPMs, PMs, LMs, or SMs
15012

Figure 1-1 — APM Architecture

1.1.1 I/O Functions

The I/O Processors, in conjunction with Field Termination Assemblies (FTAs), perform input and output scanning, and processing on all field I/O. A redundant I/O Link is standard for added security. I/O processing is performed separately from control processing functions so that I/O scan rates are completely independent of I/O quantity, controller loading, processing, and alarming. This partitioning of processing requirements allows more efficient use of control processor capability and future integration of additional I/O Processor types.

The following I/O Processors (IOPs) are available for the APM:

- Analog Input—High Level (16 points)
- Analog Input—Low Level (8 points)
- Analog Output (8 points)
- Analog Output (16 points)
- Digital Input (32 points)
- Digital Input Sequence Of Events (32 points)
- Digital Output (16 points)
- Digital Output (32 points)
- Low Level Multiplexer (32 points)
- Pulse Input (8 points)
- Smart Transmitter Interface (16 points)
- Serial Interface (32 points)
- Serial Device Interface (16 points), (not shown in Figure 1-1)

Up to 40 of the above I/O processors can be selected in any mix. An option allows redundant Analog Output and High Level Analog Input processors (up to 40 primary and 40 redundant IOPs). Each of the I/O processor functions is described in Section 2.

1.1.2 Control Functions

The APM provides a variety of control tools that can be customized to address a wide range of process automation needs. Functions from I/O scanning, through regulatory and logic control to more advanced control can be easily implemented through the APM. The APM toolbox includes a sophisticated regulatory control package, fully integrated interlock logic functions, and an advanced process engineer-oriented Control Language (CL/APM). CL/APM is an enhanced version of the Control Language implemented by Honeywell in the Multifunction Controller, Process Manager, and Application Module. This language facility includes the sequence structures needed to handle batch or hybrid applications as well as the computational capability needed for some continuous control tasks. Key to the power of this control capability is that each of the functions shares data freely within the APM and can share data from other devices on the Universal Control Network.

The following point types reside in the APMM:

- Regulatory PV
- Regulatory Control
- Digital Composite
- Logic
- Array
- Process Module
- Device Control
- Box Flag, Numeric, and Timer

Each of these data point types is described beginning with Section 3 of this publication. In addition, the descriptions of the Regulatory PV (RegPV) and Regulatory Control (RegCtl) points also contain detailed descriptions of the algorithms associated with the respective points.

1.2 COMMON FUNCTIONS

1.2.1 Point Form

Separate functional elements of the APM are used to implement various parts of typical control loops and control strategies. Each of these functional elements can be assigned a user-defined point name to allow for location-independent reference to the data associated with that function. For example, points are assigned by the user for analog input and analog output slots. The I/O Processor data (engineering-unit range for inputs, characterization option for outputs, etc.) is configured as part of the point-build process for these points. A separate point would be configured for each regulatory control (RegCtl) slot that would be linked to the assigned analog I/O points through input/output connections.

The APM provides a configurable parameter called PNTFORM (Point Form) that allows the user to define which points are to be used as the primary operator interface for point data. The PNTFORM parameter provides the user with two choices for point form: "Full" and "Component." Points that are configured as having "Full" point form include alarm-related parameters and sometimes, some other miscellaneous parameters. This information is needed when the point is to be used as the primary operator interface to the point's data.

Prior to Release 600, points that were configured as having "Component" point form did not require descriptor data and alarm-related parameters (PTDESC, EUDESC, and Keyword), because this type of information was suppressed for "Component" points.

Component point form should be used for points that provide inputs to the "Full" point, and for those points that handle the outputs from the "Full" points. "Component" points should be used as part of the "Full" point that has been designated a primary operator interface point.

NOTE

The maximum number of points per Network Interface Module (NIM) is 8000. Both full and component points should be counted when checking against this limit. Note that for the DI and DO portions of a digital composite point, the inputs and outputs from/to the digital composite point are not to be counted as part of the point total if they are implemented using the hardware reference source and destination addresses that are entered on the respective configuration form for the digital composite point.

CAUTION

Database security is provided to prevent an operator from starting an IOP that has an invalid database. After initial configuration, each IOP must be set valid. Validating the database of an IOP by selecting the VALIDATE IOP DB target on the display with the keylock in the Engineering position causes the IOP database to be accepted as is (that is, default/null configuration or configuration loaded from IDF). When the IOPs are marked valid, select SAVE DATA to checkpoint the configured data. You cannot checkpoint unless the IOP database is marked valid. Restoring the database from a valid checkpoint also sets the IOP database valid. An IOP cannot be switched from Idle to Run unless its database is valid.

The following paragraphs provide examples of the usage of the "Full" and "Component" point forms.

In Figure 1-2, a single-loop PID controller has the PV of a Smart Transmitter Interface point (FT100) connected to the PV of a RegCtl point (FIC100) that has been configured for a PID algorithm. The output (OP) of the RegCtl point is applied to an analog output (AO) point (FY100). In this case, FIC100 is the primary operator interface and is configured as a "Full" point. FT100 and FY100 are parts (components) of FIC100, and could be configured as "Component" points.

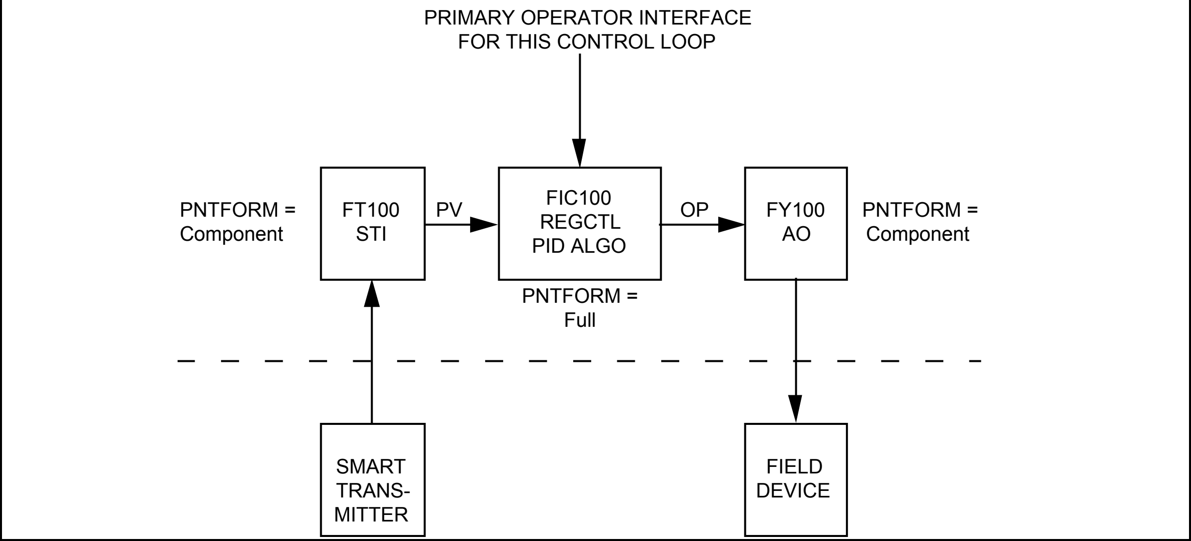


Figure 1-2 — Single-Loop Example

Figure 1-3 shows a single-loop PID controller that is controlling mass flow. The mass flow is computed by the flow compensation PV algorithm in regulatory PV (RegPV) point (FX101). FX101 receives three PVs representing the uncompensated flow, absolute pressure, and absolute temperature from AI points FT101, PT101, and TT101. FX101 provides the PV as mass flow to a RegCtl point (FIC101) that has been configured for a PID algorithm. FIC101 provides an output to an AO point (FY101).

In this example, FIC101 is the primary operator interface and would be configured as a "Full" point. FX101 is part of FIC101 and would be configured as a "Component" point. FT101, PT101, and TT101 could be configured as "Full" or "Component " points. "Full" would be selected for any of these points that require separate alarm reporting, such as alarming for uncompensated flow, line pressure, or fluid temperature. FY101 must be configured as a "Component" point.

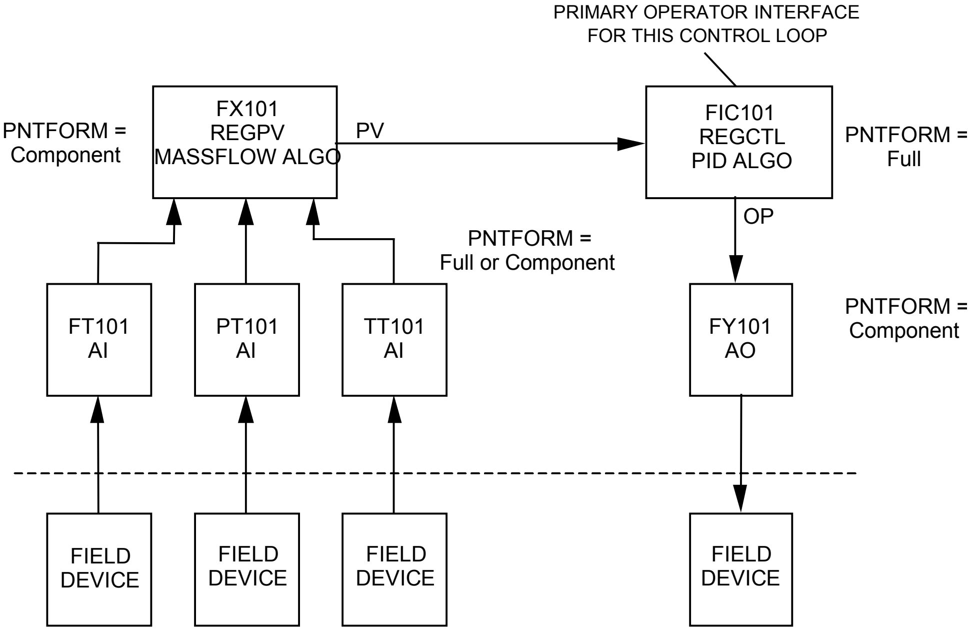
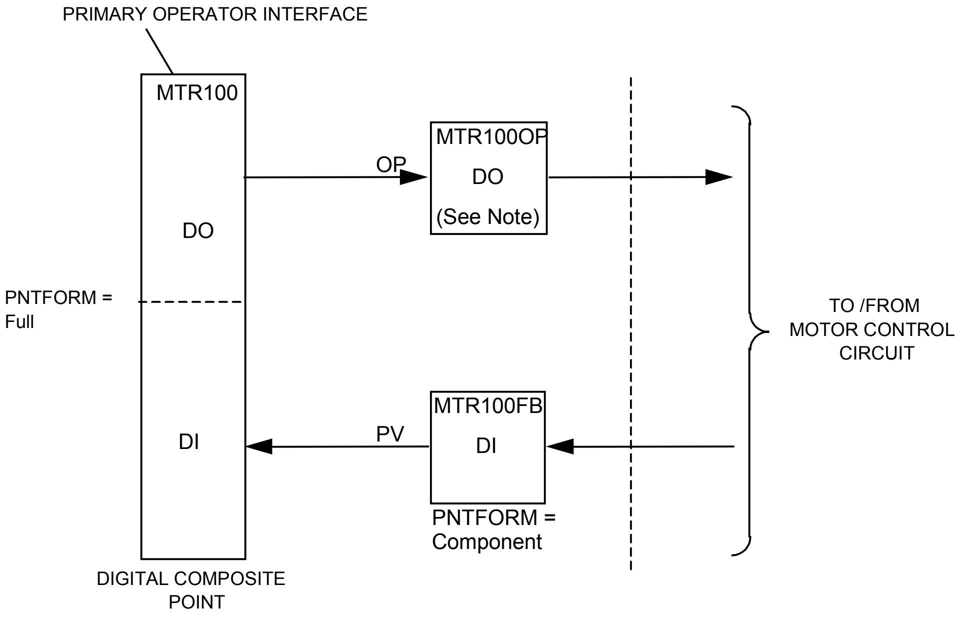


Figure 1-3 — APM — Mass Flow Example

Figure 1-4 shows a single-output, single-input digital composite point (MTR100) that interfaces a motor control circuit through digital output (DO) point MTR100OP, and digital input (DI) point MTR100FB. In this example, MTR100 is the primary operator interface and would be configured as a "Full" point. MTR100OP and MTR100FB are parts of MTR100 and would be configured as "Component" points.

Note that assigning point names in this example for the standard digital input point and digital output point is optional. These points can be alternatively referenced, using the hardware reference addresses that would be assigned as the source and destination on the configuration form for the digital composite point.



NOTE: Full-point form cannot be assigned to DO (digital output) points; they are always component points. Stand-alone digital output points require using a digital composite point that has 0 (zero) inputs and 1 output.

Figure 1-4 — Motor Control Example

2056

Figure 1-5 shows three types of stand-alone points. FT102 and CT105 are points that could be used for data acquisition; in this case these points would be configured as full points. For the stand-alone digital output (DO) point SW103, a digital composite (DigComp) point would be used and would be configured as having 0 (zero) inputs and 1
output.

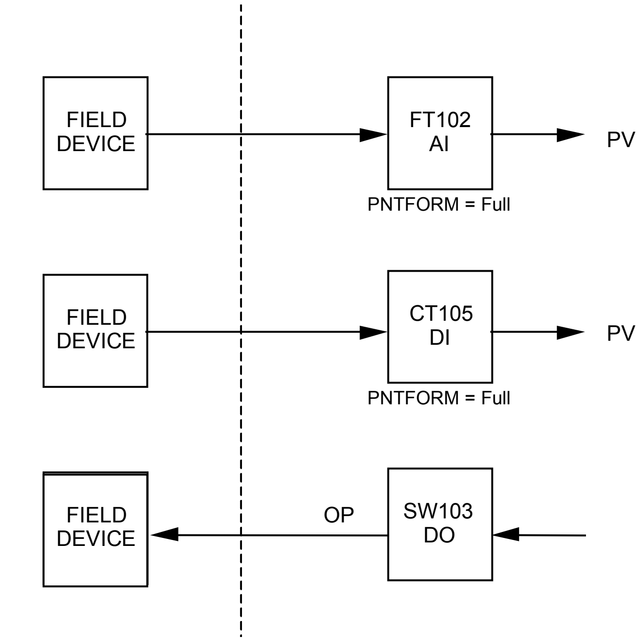


Figure 1-5 — Stand-Alone Points 2057

1.2.2 Associated Display

With Release 510 and later software, an Associated Display can be configured for each HPM point at build time by entering the name of a custom built schematic in the ASSOCDSP parameter. At operating time, the operator can call up that associated custom schematic from a point Detail Display or Group Display.

The ASSOCDSP parameter can be changed from the configuration page of the Detail Display (for points that have a configuration page). The keylevel must = Engineer.

1.3 ALARMING

The following paragraphs describe the common alarming functions that can be configured for APM points. The alarm parameters available depend on the point type. Refer to the individual point description for details. Alarm detection and reporting for a point can be configured only if the full point form is specified.

1.3.1 Alarm Priorities

For each point, a separate alarm priority can be specified for each alarm (for example, PV high alarm can be low priority but PV high high alarm can be emergency). The following alarm priorities are supported for APM points:

Emergency
High
Low
Journal
JnlPrint
Printer
NoAction

The Journal option causes alarms to be journaled only. The JnlPrint selection causes alarms to be journaled in the HM and printed. If the Printer option is selected, alarms are printed but not journaled. Note that if the printer is not working, alarms could be lost.

Alarm priority configuration information is maintained by the NIM.

1.3.2 Alarm Enable Status

Alarm enable status is applicable to full point forms and allows the user to enable, disable, and inhibit alarms. This function is accomplished through the ALENBST parameter. The alarm enable status function is resident in the NIM.

1.3.3 Contact Cutout

The main purpose of the contact cutout function is to prevent a proliferation of alarms from being reported to the operator. This function can be used to cut out alarms on a point when they are generated because of alarm conditions that have been detected at other points. Contact cutout is provided for all the point types in the APM and is implemented through the CONTCUT parameter. Another point or CL program must write to the CONTCUT parameter in order to change the Contact Cutout state.

When the contact cutout state is applied, alarms at the point are cut out. New alarms are not reported on the alarm summary display at the Universal Station, nor are they journaled on the History Module or Real Time Journal. This is the same way inhibited alarms are handled except Cutout alarms continue to be reported to the AM or CM60 for event initiated processing (EIP). A Contact Cutout True message is journaled for points that are in alarm when cutout. For points that are not in alarm when cut out, no message is journaled.

On event recovery (for example, node failover), the system journals a Contact Cutout True message for points that are currently in alarm and cut out.

When Contact Cutout is removed, points that were in alarm are checked and if the alarm condition still exists, the system generates a new alarm with a current timestamp. A Contact Cutout False message is journaled for points that returned to normal while cut out. No message is journaled for points that were not in alarm while cut out.

NOTE

When cutout is removed, it is possible to receive a Contact Cutout False message and an alarm for the same point. This would occur if the point had changed from an alarm state to the normal state, and back to alarm, while it was cut out.

A network configuration option provides two choices for handling **unacknowledged** alarms that exist on the US Alarm Summary display when Contact Cutout is applied:

- CLEAR IMMEDIATELY clears all of the point's unacknowledged alarms from the display.
- CLEAR WHEN ACKED (also called the Backlight Option) leaves unacknowledged alarms on the display and the time stamp is backlit. The alarms are cleared when they are acknowledged.

If the second option (above) is selected, the complete backlighting action is as follows:

- When Contact Cutout is applied, unacknowledged alarms on the Alarm Summary Display have their timestamp backlit (indicating that the alarm is inhibited).
- If an a point's alarm state returns to normal while the point is cutout, the timestamp remains backlit. When cutout is removed—

  backlighting is removed from the timestamp
  the point's priority indicator is backlit (indicating return to normal)
  a contact Cutout False message is journaled.
- If the NIM fails over while Contact Cutout is applied to any point, the point's tagname is backlit in addition to it's timestamp. As soon as communication is restored, backlighting is removed from the tagname.

When Contact Cutout is removed, backlighting is removed from cutout points.

1.3.3.1 UCN Contact Cutout Implementation

Contact Cutout is implemented by using a logic point or a CL program to write a Boolean variable from a primary control point into an alarmable UCN secondary point's CONTCUT parameter. A TRUE (ON) in this parameter will cut out alarming by the secondary point. The secondary point will not resume alarming until a logic point or CL program writes a FALSE (OFF) into the secondary's CONTCUT parameter.

Typical implementation of Contact Cutout uses a primary point's PTINAL parameter (point-in-alarm indicator) to control a secondary point's CONTCUT status. This is a convenient method for providing Contact Cutout for all primary alarms, and we recommend it as the safest and most secure implementation. However, any other valid primary point alarm parameter, such as PVHIFL, PVHHFL, PVLOFL, PVROCNFL, or any other valid primary point Boolean parameter can be used. When distributing a primary point's alarm status to the CONTCUT parameter of several points, use only one input connection to make the best use of peer-to-peer resources.

When implementing a Contact Cutout connection using a logic point, the Logic Bad Input handling Option parameter, LIBADOPT, must be set to OFF in case the primary point fails or is deleted while in alarm. This implementation ensures that the secondary point's CONTCUT parameter will be set to OFF in the event of a point failure. In addition, if the primary and secondary points are in different UCN nodes, the logic point should be located in the same node as the secondary point in case the node containing the primary point is idled or fails.

For the same reason, we recommend that CL programs be located in the same node as the secondary point. CL programs should include error trapping code to handle the different failure cases described above.

In general, Contact Cutout should be implemented using primary point alarming flags as previously described, because these alarm parameters automatically clear (to the OFF state) when the primary points are made INACTIVE or the node containing the primary point is idled. Note that when connecting CONTCUT to other parameters, such as box flags or digital composite PV flags, these parameters do not provide similar protection. Because CONTCUT cannot be changed by the operator, it is possible to end up with a secondary point temporarily or permanently disabled from alarming.

1.3.4 Primary Module Points

Primary Module (PRIMMOD) Points are used to collect alarms and events from points that are related for some purpose. All UCN points with alarming capability have a PRIMMOD parameter with which to identify a common PRIMMOD point. All points with a matching Primmod point are considered to be in the same alarm group.

The value for the Primmod parameter can be any point in the same NIM. The value is assigned during point build, but can later be stored by a CL program or from schematics. If a Primmod point name is changed when alarms exist, the alarms are reissued for the new Primmod name.

Up to 10 Primmod points can be grouped into one annunciator group in the Area data base and Primmod points can be assigned to turn on the configurable button alarm indicators.

1.3.5 Auxiliary Units

Alarmable NIM points in Release 520 and later software have an Auxiliary Unit ($AUXUNIT) parameter. If this parameter is set to null (- -), alarms and messages on that point go to the primary unit. If a valid Unit ID is specified, alarms and messages from that point go to the Auxiliary Unit.

$Auxunit can be configured at build time or, with the proper keylevel, it can be initially assigned or changed by schematics, CL programs, the DEB alter parameters function, or from the configuration page of the Point Detail display. The keylevel necessary to change $AUXUNIT is configurable in the System Wide Values section of the Network Configuration File.

Alarms from a point where $AUXUNIT has been configured are accepted only by Universal Stations having an Area Database in which both the Primary Unit and the Auxiliary Unit are configured.

If an $AUXUNIT assignment is changed and there are outstanding alarms on the old unit, alarms are deleted from the old unit and added to the new unit.

1.4 RED TAGGING

A point can be "red tagged" to indicate that it requires maintenance. This is accomplished by setting the REDTAG parameter to On. Typically, the operator sets the output of the point to a desired safe value before putting on the red tag. Once red tagged, the mode, mode attribute, external mode switching state, and output cannot be changed. The point cannot be reconfigured or deleted if the red tag is On. Red tagging is supported for only the analog output, RegCtl (only if it is outputting to an AO or a PWM DO Processor), and digital composite point types. A point must be configured for the full point form in order to be red tagged.

The mode and the mode attribute of the point must be changed to Man and Oper, respectively, and the external mode switching state, if configured, must be Disabled before the point can be red tagged. To red tag a digital composite point, the output of the point must not be configured for the momentary state.

The Red Tag function prevents changes to the output on a point that has the REDTAG parameter set to ON, but it may still be possible to change the output connected to the field. For example, even if a Digital Composite point has REDTAG set to ON, a program/Logic Block can still change the state of the Digital Output.

1.5 REFERENCES

The following publications contain information related to the control functions described in this publication:

- **Control Functions in the System**—*System Control Functions* in the *Implementation/Startup & Reconfiguration* **-** *2* binder**.**

- **Control Functions in the CM50S**—Refer to the publications in the *Implementation/CM50S* binder.

- **Control Functions in the Application Module**—*Application Module Control Functions* in the *Implementation/Application Module - 1* binder.

- **Definitions of the APM Parameters**—*Advanced Process Manager Parameter Reference Dictionary* in the *Implementation/Advanced Process Manager - 2* binder.

- **Configuration, Redundancy, Failover information**—*Engineer's Reference Manual* in the *Implementation/Startup & Reconfiguration - 2* binder.

- **Control Language**—*Control Language/Advanced Process Manager Reference Manual* in the *Implementation/Advanced Process Manager* **-** *2* binder**.**

- **Command Processor Utilities**—*Command Processor Operation* in the *Implementation/Startup & Reconfiguration - 1* binder.

- **UDC 6000 Process Controller**—*PM/UDC 6000 Integration Manual* in the *Implementation/PM/APM/HPM Optional Devices* binder.

- **Performance and Implementation Information** — *APM Implementation Guidelines Manual* in the *Implementation/Advanced Process Manager - 1 binder.*

- **Smartline Transmitter Information** — *Smartline Transmitter Integration Manual* in the *Implementation/PM/APM/HPM Optional Devices binder.*

- **Serial Interface Communications**—*APM/HPM Serial Interface Options* in the *Implementation/PM/APM/HPM Optional Devices* binder**.**

I/O FUNCTIONS
Section 2

This section defines the I/O functions available in the APM analog and digital data points. The APM control functions are described beginning with Section 3. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary, which is in this binder.

2.1 OVERVIEW

The analog and digital data points in the I/O Processors (IOPs) perform input and output processing on all field I/O. These functions are performed independently of the control functions. The partitioning of processing requirements allows more efficient use of control processor capability and future integration of additional IOP types. The following I/O points are available in the APM, and are described in this section:

- Analog Input—High Level
- Analog Input—Low Level
- Smart Transmitter Interface*
- Analog Output
- Digital Input
- Digital Input, Sequence Of Events
- Digital Output
- Pulse Input
- Serial Interface

To support on-line field maintenance activities, Standby Manual units are recommended for the Process Manager AO and DO I/O Processors.

*Note that a Serial Device Interface (SDI) IOP is configured as a Smart Transmitter Interface (STI) IOP.

2.2 HIGH AND LOW LEVEL ANALOG INPUT POINT

The analog input point converts an analog PV signal received from a field sensor to engineering units for use by other data points in the APM, and by the rest of the **TotalPlant** Solution (TPS) system. To accomplish this function, the analog input point (shown in Figure 2-1) performs

- Analog-to digital conversion
- PV characterization
- Range Checking and PV filtering
- PV source selection
- Alarm detection

High level points are located in the High Level Analog Input (HLAI) IOP. One type of low level point is located in the Low Level Analog Input (LLAI) IOP. This type is generally used for control points. The other type is located in either the Low Level Multiplexer (LLMUX) or the Remote Hardened Multiplexer (RHMUX) IOP. This type is generally used for data acquisition points. The type of analog input point needed is based on the type of field sensor that is providing the input to the point and the characterization options selected by the user as listed in Table 2-1.

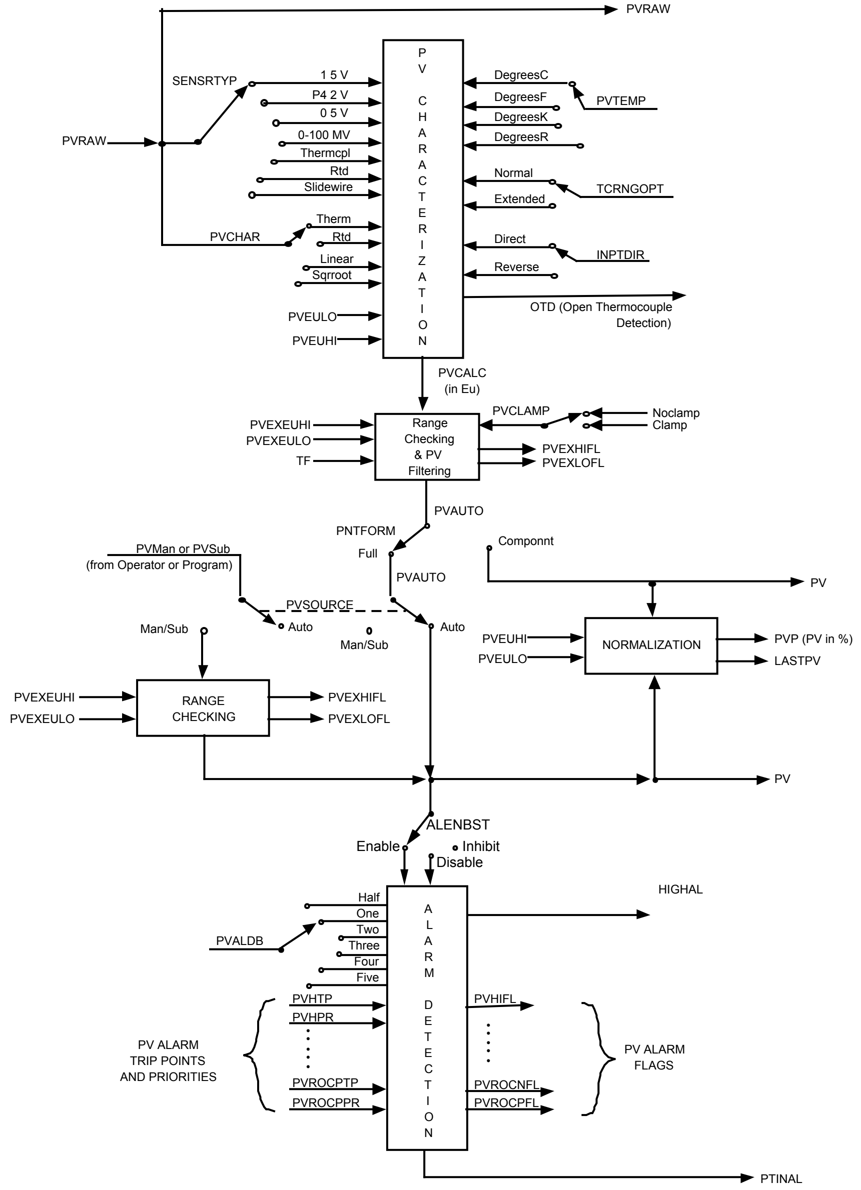


Figure 2-1 — Analog Input Point, Functional Diagram

2058

2.2.1 PV Characterization

The PV signal received from the field is characterized based on the entries that the user makes for the SENSRTYP, PVCHAR, PVTEMP, INPTDIR, and TCRNGOPT parameters as shown in Figure 2-1. The input PV signal is first converted to a raw PV signal (PVRAW) whose units can be %, ratio, millivolts, microvolts, or milliohms depending on the entry made for the SENSRTYP parameter.

The PVRAW signal is then converted to the engineering units. The engineering unit conversions that are performed in the HLAI, LLAI and LLMUX points are listed in the following chart, and described in the following paragraphs.

HLAI Point	**LLAI Point**	**LLMUX Point** (NOTE 1)
Linear	Linear	Linear
Square Root	Square Root	RTD (NOTE 2)
Thermal	Thermal (with reference junction compensation)	Thermal (with reference junction compensation)
Slidewire		

NOTES:

1. In general, LLMUX points apply to (include) points built against the LLMUX IOP as well as the RHMUX IOP. The RHMUX does not have a unique point type.
2. RTD is not supported by RHMUX IOP.

2.2.1.1 Linear Conversion

The PVRAW value is converted to a floating-point number. The output value of the linear conversion is PVCALC, which is calculated based on the raw input span (for slidewire and 0-100 mV sensor types only), and the engineering unit span. The state of the input direction parameter (INPTDIR) is taken into consideration during the calculation of PVCALC as follows:

For slidewire and 0-100 mV sensor types, when INPTDIR is Direct:

$$\mathrm{PVCALC} = \frac{(\mathrm{PVRAW} - \mathrm{PVRAWLO})}{(\mathrm{PVRAWHI} - \mathrm{PVRAWLO})} * (\mathrm{PVEUHI} - \mathrm{PVEULO}) + \mathrm{PVEULO}$$

For 0-5V, 0.4-2V, and 1-5V, sensor types, when INPTDIR is Direct:

$$\mathrm{PVCALC} = \frac{\mathrm{PVRAW}}{100} * (\mathrm{PVEUHI} - \mathrm{PVEULO}) + \mathrm{PVEULO}$$

For slidewire and 0-100 mV sensor types, when INPTDIR is Reverse:

$$\mathrm{PVCALC} = \mathrm{PVEUHI} - \frac{(\mathrm{PVRAW} - \mathrm{PVRAWLO})}{(\mathrm{PVRAWHI} - \mathrm{PVRAWLO})} * (\mathrm{PVEUHI} - \mathrm{PVEULO})$$

For 0-5V, 0.4-2V, and 1-5V, sensor types, when INPTDIR is Reverse:

$$\mathrm{PVCALC} = \mathrm{PVEUHI} - \frac{\mathrm{PVRAW}}{100} * (\mathrm{PVEUHI} - \mathrm{PVEULO})$$

Table 2-1 — HLAI, LLAI, and LLMUX Sensor Types and PV Characterization Options

Sensor Type (SENSRTYP)	AI Processor Type	PVCHAR Options	PV RAW[1]	PV CALC	Bad PV Detection[2]
0 to 5 Volts	HL & LL HL & LL HL HL	Linear Square Root Thermocouple RTD	%	EU	Range check on PVCALC
0.4 to 2 Volts	HL HL HL HL	Linear Square Root Thermocouple RTD	%	EU	Range check on PVCALC HLAI checks for open input
1 to 5 Volts	HL & LL HL & LL HL HL	Linear Square Root Thermocouple RTD	%	EU	Range check on PVCALC HLAI checks for open input
Slidewire	HL	Linear	Ratio	EU[3]	Bad slidewire source, range check on PVCALC
0 to 100 mV	LL, LLMUX RHMUX	Linear	millivolts	EU[3]	Range check on PVCALC
Thermocouple	LL, LLMUX, RHMUX	Thermocouple	μvolts	EU	Open thermocouple, and range check on PVCALC
RTD	LL, LLMUX	RTD	milliΩ	EU	Range check on PVCALC

LEGEND: EU = Engineering Units
HL = High Level Analog Input
LL = Low Level Analog Input
LLMUX = Low Level Analog Multiplexer
PVCALC = Calculated PV
PVCHAR = PV Characterization
PVRAW = PV received from field and converted to digital form by the A/D converter
RHMUX = Remote Hardened Analog Input Multiplexer

Notes:

1. PVRAW is the voltage signal at the APM Field Termination Assembly as a percentage of the voltage range for the sensor type. The exceptions are as follows
   a. For a thermocouple sensor type, PVRAW is in microvolts after reference junction compensation. If an open thermocouple is detected, PVRAW is set to NaN.
   b. For an RTD sensor type, PVRAW is in milliohms after lead-wire compensation. If an open RTD is detected, PVRAW is set to NaN.
   c. An external power source is used to excite the slidewire. The power source and the slidewire are connected to separate analog input points. One power source input can be used with several slidewire inputs.

   For a Slidewire type, PVRAW is the slidewire ratio (Vin/Vsrc)
   where: Vin is the FTA voltage input for this data point (slot)
   Vsrc is the FTA voltage source at the slidewire source slot specified by parameter SLWSRCID. If Vsrc is zero (fails the under-range check), PVRAW is set to "NaN."

   d. For a 0-100 millivolt sensor type, PVRAW is the FTA voltage input for the slot.
2. If the diagnostics determine that the A/D converter has failed, PVRAW of the slot is set to NaN.
3. The normal operating range for PVRAW is configured by the user (for a thermocouple 0% = PVRAWLO, 100% = PVRAWHI; for a slidewire, 0 = PVRAWLO, 1 = PVRAWHI).

2.2.1.2 Square-Root Conversion

The square-root calculation is applied to the PVRAW input such that 100% of span = 1.0. The square-rooted value is then converted to engineering units based on the configured PV engineering-unit range values. (For example, square root of 100% = 100%; square root of 50% = 70.71%.) The output value of the square-root conversion is PVCALC, which is calculated based on the state of the input direction parameter (INPTDIR) as follows:

If PVRAW ≥ 0.0 and INPTDIR is Direct:

$$\text{PVCALC} = \sqrt{\frac{\text{PVRAW}}{100}} * (\text{PVEUHI} - \text{PVEULO}) + \text{PVEULO}$$

If PVRAW < 0.0 and INPTDIR is Direct:

$$\text{PVCALC} = -\sqrt{\frac{\text{PVRAW}}{100}} * (\text{PVEUHI} - \text{PVEULO}) + \text{PVEULO}$$

If PVRAW ≥ 0.0 and INPTDIR is Reverse:

$$\text{PVCALC} = \text{PVEUHI} - \sqrt{\frac{-\ \text{PVRAW}}{100}} * (\text{PVEUHI} - \text{PVEULO})$$

If PVRAW < 0.0 and INPTDIR is Reverse:

$$\text{PVCALC} = \text{PVEUHI} - \sqrt{\frac{-\ \text{PVRAW}}{100}} * (\text{PVEUHI} - \text{PVEULO})$$

2.2.1.3 Thermal Conversion

Thermal linearization is performed on thermocouple and RTD input types, and is selectable by parameter PVCHAR. The following thermocouples are supported by the analog input point:

Btherm
Etherm
Jtherm
Ktherm
Rtherm
Rptherm
Stherm
Ttherm

The range of the thermocouple type used with the LLAI or LLMUX points (NOTE) can be increased by selecting Extended as the entry for the TCRNGOPT parameter. Refer to the APM PRD for the normal and extended ranges.

The LLAI and LLMUX points calculate the reference junction compensation from the measured reference junction output level. This value is stored and then later converted back to microvolts, with respect to 0 degrees C, for each thermocouple that is to be compensated. The cold-junction reference compensation (PVREFJN) parameter is expressed in microvolts for the specified thermocouple and is added to the microvolt value for PVRAW.

NOTE: In general, LLMUX points apply to (include) points built against the LLMUX IOP as well as the RHMUX IOP. The RHMUX does not have a unique point type.

The following 3-wire RTDs are supported by the analog input point:

PtDinRTD
PtJisRTD
NicklRTD
CopprRTD

For an RTD, the LLAI and LLMUX (NOTE) points calculate the lead-wire compensation and then subtract the value from PVRAW.

NOTE: RTD is not supported by RHMUX IOP.

The maximum allowable lead-wire resistance and intrinsic safety barrier resistance for the RTDs are listed in the following chart.

	Maximum Allowable Lead Resistance (Note 1)		Maximum Allowable Intrinsic Safety Barrier Resistance	
RTD Type	Entire Loop	Per Leg	Entire Loop	Per Leg
Pt: 100 Ω DIN Characterization	20 Ω	10 Ω	18 Ω	18 Ω
Pt: 100 Ω JIS Characterization	20 Ω	10 Ω	18 Ω	18 Ω
Ni: 120 Ω Edison Type 7 Characterization	20 Ω	10 Ω	18 Ω	18 Ω
Cu: 10 Ω SEER Standard Characterization	20 Ω	10 Ω	0.0 Ω	0.0 Ω

Note: Proper compensation for lead-wire resistance depends on the resistance being equal in each leg of the RTD. This includes resistance due to lead-wire resistance and intrinsic safety barriers. No provision is made to compensate for lead-wire resistance mismatch or intrinsic-safety-barrier resistance mismatch. Both the lead resistance and the intrinsic-safety-barrier resistance are allowed simultaneously when connected to an RTD in a Division 1 area.

2.2.2 PV Range Checking and Filtering

PV range checking ensures that the PVCALC output of PV characterization is within the limits defined by parameters PVEXEULO and PVEXEUHI. If either of the limits is violated, the output of the PVAUTO is set to NaN if clamping has not been specified. If clamping has been specified, the output of the PVAUTO is clamped to PVEXEUHI or PVEXEULO, except when PVRAW (and consequently, PVCALC) is NaN in which case PVAUTO will be NaN.

If the range-checked and filtered value is less than the value specified by the user-configured LOCUTOFF parameter, the final output called PVAUTO is forced to PVEULO.

First-order filtering is performed on PVCALC, as specified by the user through parameter TF (filter lag time).

2.2.3 PV Source Selection

The PVSOURCE parameter allows the user to select the source of the PV for this data point. As shown in Figure 2-1, the PV can be provided by the Range Checking and Filtering circuit (when PVSOURCE is Auto), it can be the manually entered PV (when PVSOURCE is Man), or it can come from a sequence program (when PVSOURCE is Sub).

In addition, the PV source option parameter (PVSRCOPT) determines whether it is permissible to change the PV source to a source other than Auto. PVSRCOPT has two states: OnlyAuto and All. The All state allows the PV to be manually entered for this data point.

2.2.4 Alarming

The analog input data point compares the PV to threshold values and records the alarms in the database of the data point. The alarms are then reported by the Advanced Process Manager Module (APMM). The parameters that are associated with alarming in the analog input point are as follows:

$OLDEVDS
ALENBST
BADPVFL
BADPVPR
CONTCUT
EIPPCODE
HIGHAL
HIGHALPR
PTINAL
PVALDB
PVALDBEU
PVEXHIFL
PVEXLOFL
PVHHFL
PVHHPR
PVHHTP
PVHIFL
PVHIPR
PVHITP
PVLLFL
PVLLPR
PVLLTP
PVLOFL
PVLOPR
PVLOTP
PVROCNFL
PVROCNPR
PVROCNTP
PVROCPFL
PVROCPPR
PVROCPTP

Refer to the *Advanced Process Manager Parameter Reference Dictionary* for the definitions of these parameters.

2.2.4.1 Disable Input Alarming When Using Open-Line Detection

With R640, disabling annunciation of BADPV alarms for IOP-resident AI points using open-line detection is performed using the $OLDEVDS parameter. This function is used for disabling the BADPV alarms generated by LLAI/LLMUX IOPs for open thermocouples or RTDs, when transmitters and positioners are physically disconnected for normal maintenance.

Setting $OLDEVDS from ENABLE to DISABLE removes the annunciation of BADPV alarms for the respective point as follows:

1. There is no audible console alarm.

2. No blinking LEDs on the console or keyword.

3. Schematic behavior based on events is suppressed. Schematic behavior based on parameters, e.g., BADPVFL, is not affected.

4. The event is removed from the alarm lists and alarm displays.

The initial value of $OLDEVDS can be ENABLE or DISABLE and can be changed between the two values at any time from a detail display or a schematic.

However, this function is not applicable for all AI points. For other types of analog inputs, the initial value of $OLDEVDS is NOOTDETC. When this value is set, it cannot be changed, even by the DEB. This ensures that analog inputs other than those having open-line detection will not have BADPV values removed from the alarm displays because the BADPVs are a result of other factors other than $OLDEVDS being set to DISABLE.

The NIM resident parameter $OLDEVDS can be configured through the DEB and the TPS Builder. This parameter is valid for both full and component point forms.

ATTENTION

- Only BADPV alarm events are disabled using this function.
- This function silences only the annunciation and does not affect any operations of the system.

2.3 SMART TRANSMITTER INTERFACE POINT

The Smart Transmitter Interface (STI) point provides an interface to Honeywell's advanced series of Smartline 3000 Smart Transmitters. The STI point can support the following Smartline Transmitter types:

- ST3000 Smart Pressure Transmitter for differential, gauge, and absolute pressure measurements
- STT3000 Smart Temperature Transmitter for temperature, millivolts, and ohms measurements, and
- MagneW 3000 Smart Magnetic Flow Transmitter for flow measurements

The STI points are located on the STI IOP. Each STI IOP has a maximum of 16 inputs, and it can communicate bidirectionally with up to 16 Smartline transmitters, regardless of the mix of transmitter types (pressure, temperature, or flow) using Honeywell’s digitally enhanced (D.E.) protocol.

The STIMV IOP supports all of the above and multi-PV Smartline Transmitter types such as the following:

- SCM3000 Smart Flow Transmitter (Coriolis method)
- Drexelbrook SLT Level Transmitter
- SMV 3000 Multivariable Pressure Transmitter
- SGC 3000 Gas Chromatograph

An STIMV IOP allows up to four multi-PV transmitters or a mix of multi-PV and single PV transmitter inputs that total no more than 16. A multi-PV transmitter is configured as if it were in n contiguous slots where n = the number of PVs expected. The STITAG parameter value for each contiguous slot must be identical. Refer to the *PM/APM Smartline Transmitter Integration Manual* for complete details.

The STI IOP and the Smartline Transmitters use bidirectional digital communication to allow the user to configure, view, and modify the transmitter database from the Universal Station. This digital protocol allows a more precise PV value to be transferred, thereby permitting more accurate control of the process. In addition, the transmitter can also send a secondary variable such as the transmitter temperature, cold junction temperature, or totalized value, depending on the transmitter type.

The transmitter database can be configured at the Universal Station and down-line loaded to the transmitter and the transmitter database can be up-line loaded to the STI IOP as required, when the STI point is in the inactive state.

During normal operation (when the STI point is in the active state), each time that the transmitter broadcasts the PV value to the STI IOP, it also sends the one byte of its database (depending on the selected DECONF mode) to the STI IOP. This allows the STI IOP to compare the stored database to the newly received database to check for database mismatches. If a mismatch is detected, the PV is set to NaN and the status is set to DBCHANGE. The user can easily correct the mismatch by down-line loading the database stored in the STI IOP.

All key transmitter parameters can be accessed from the Universal Station including

- Upper and Lower Range Values
- Damping
- PV type
- DE (Digital Enhanced communications) configuration variables
- Status of the Transmitter
- Transmitter's serial number and software revision number
- Transmitter's scratchpad

The user can access these variables through the point's Detail Display or custom-built schematics. In addition to the tag name assigned to the process point associated with the transmitter, the transmitter is also assigned a tag name and the access mechanism follows the TDC3000 parameter access mechanism. This allows all the LCN capabilities applicable to a data point to be also applicable to the Smartline Transmitters.

The STI IOP maintains a copy of the transmitter's database. When a transmitter failure occurs, the database can be down-loaded to the transmitter. This database save/restore feature can significantly reduce the downtime of a control loop by reducing the time in getting a replacement transmitter in operation. The transmitter database can also be saved to a History Module or removable media if a checkpoint request is initiated. This allows for centralized control of the transmitter database, which significantly minimizes the effort required to establish the transmitter database during startup or normal operation.

The STI IOP also allows the user to access the detailed status of a transmitter. The status is displayed at a Universal Station together with the scratchpad information that has been entered, including any maintenance notes.

Calibration of the transmitter can also be accomplished from the Universal Station. This function allows on-line adjustment of the transmitter's working ranges so that the reference points for a measurement are accurate.

In addition a Smart Field Communicator (SFC), which is a hand-held device, can also be physically connected to the appropriate FTA in the APM cabinet to communicate with Smartline Transmitters without disrupting the process, as required (refer to the appropriate *Operating Guide for Smart Field Communicators*).

2.3.1 Parameter Comparisons

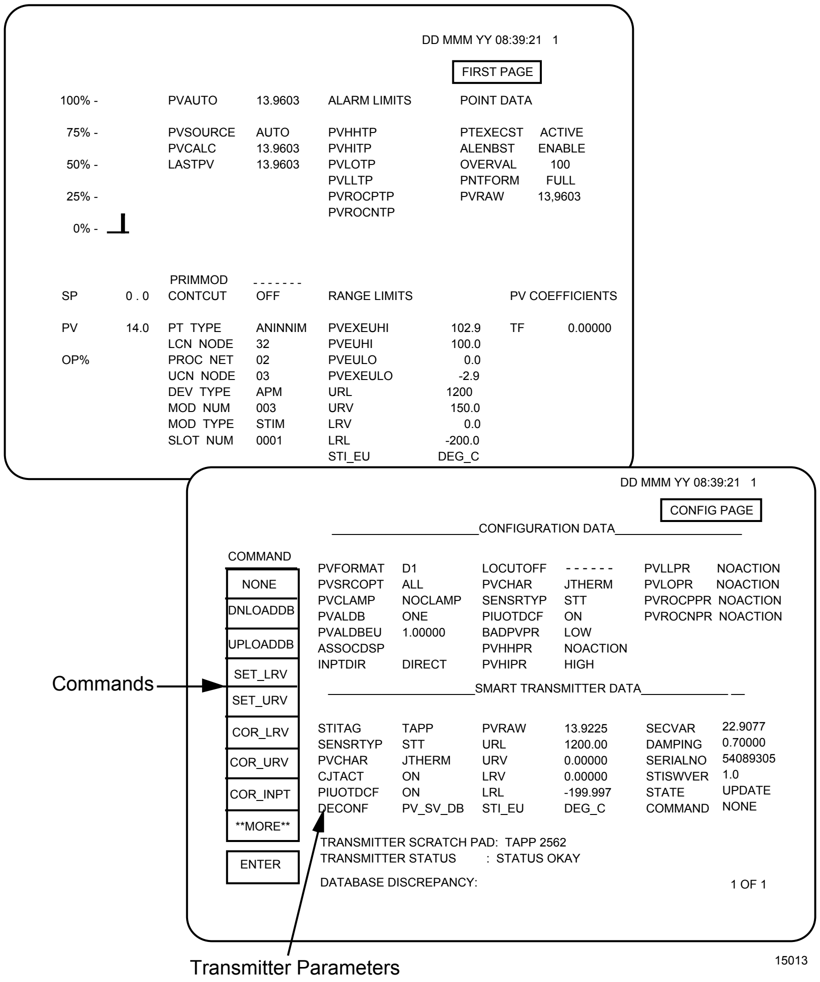
To configure the STI IOP point to operate with the appropriate Smartline Transmitter, it is necessary to know the parameter relationships between the STI IOP and the transmitter database. Table 2-2 lists and describes these relationships.

Table 2-2 — Smartline Transmitter/STI-IOP Parameter Comparison

Smart Transmitter Database Parameter	Corresponding STI IOP Parameter	Remarks
Upper Range Value & Lower Range Value	URV LRV	Define the operating range of the transmitter. These values correspond to the values for PVEUHI and PVEULO, respectively.
Upper Range Limit & Lower Range Limit	URL LRL	These parameters are the respective built-in maximum and minimum limits of the transmitter and they cannot be changed. These parameters are read-only parameters at the Universal Station. URL must be configured to match the URL value of the transmitter.
PV Damping	DAMPING	PV damping at the transmitter. Refer to paragraph 2.3.2.5.
Tag Identifier	STITAG	Transmitter identifier. This parameter is a read-only parameter at the Universal Station.
Software Version	STISWVER	Revision level of the software in the transmitter. This parameter is a read-only parameter at the Universal Station.
Serial Number	SERIALNO	Serial number (PROM) of the transmitter. This parameter is a read-only parameter at the Universal Station.
Secondary Variable	SECVAR	For a pressure transmitter, the secondary variable is the meter-body temperature of the transmitter. For a temperature transmitter, the secondary variable is the cold-junction temperature. For a flowmeter, the secondary variable is the totalized value. This parameter is a read-only parameter at the Universal Station.
Linear / Square Root Characterization	PVCHAR	Refer to Table 2-3.
Communication Mode	DECONF	Refer to paragraph 2.3.3.
Cold Junction Compensation Active	CJTACT	Applicable to STT 3000 only.

For more information on the STI IOP, refer to the appropriate Smartline transmitter publication and to the *PM/APM Smartline Transmitter* Integration Manual. For more information on STI IOP parameters, refer to the *Advanced Process Manager Parameter Reference Dictionary*.

A detail display of a typical STI point is shown in Figure 2-2. It also shows the location of the transmitter database parameters.



```
                                                         DD MMM YY 08:39:21   1

                                                              FIRST PAGE

100% -       PVAUTO       13.9603   ALARM LIMITS       POINT DATA

 75% -       PVSOURCE     AUTO      PVHHTP             PTEXECST    ACTIVE
             PVCALC       13.9603   PVHITP             ALENBST     ENABLE
 50% -       LASTPV       13.9603   PVLOTP             OVERVAL       100
                                    PVLLTP             PNTFORM      FULL
 25% -                              PVROCPTP           PVRAW       13,9603
                                    PVROCNTP
  0% -

              PRIMMOD     - - - - - - -
SP      0 . 0  CONTCUT     OFF       RANGE LIMITS                  PV COEFFICIENTS

PV      14.0   PT  TYPE    ANINNIM   PVEXEUHI          102.9     TF         0.00000
               LCN  NODE   32        PVEUHI            100.0
OP%            PROC  NET   02        PVEULO              0.0
               UCN  NODE   03        PVEXEULO           -2.9
               DEV  TYPE   APM       URL               1200
               MOD  NUM    003       URV               150.0
               MOD  TYPE   STIM      LRV                 0.0
               SLOT  NUM   0001      LRL              -200.0
                                     STI_EU            DEG_C
```

```
                                                         DD MMM YY 08:39:21   1

                                                                CONFIG PAGE
                     ______________________CONFIGURATION DATA____________________

 COMMAND
            PVFORMAT    D1          LOCUTOFF    - - - - - -    PVLLPR     NOACTION
  NONE      PVSRCOPT    ALL         PVCHAR      JTHERM         PVLOPR     NOACTION
DNLOADDB    PVCLAMP     NOCLAMP     SENSRTYP    STT            PVROCPPR   NOACTION
            PVALDB      ONE         PIUOTDCF    ON             PVROCNPR   NOACTION
UPLOADDB    PVALDBEU    1.00000     BADPVPR     LOW
            ASSOCDSP                PVHHPR      NOACTION
 SET_LRV    INPTDIR     DIRECT      PVHIPR      HIGH
 SET_URV             ______________________SMART TRANSMITTER DATA______________ __

 COR_LRV    STITAG      TAPP        PVRAW       13.9225        SECVAR     22.9077
 COR_URV    SENSRTYP    STT         URL         1200.00        DAMPING    0.70000
COR_INPT    PVCHAR      JTHERM      URV         0.00000        SERIALNO   54089305
**MORE**    CJTACT      ON          LRV         0.00000        STISWVER   1.0
            PIUOTDCF    ON          LRL         -199.997       STATE      UPDATE
  ENTER     DECONF      PV_SV_DB    STI_EU      DEG_C          COMMAND    NONE

            TRANSMITTER SCRATCH PAD:  TAPP 2562
            TRANSMITTER STATUS        :  STATUS OKAY

            DATABASE DISCREPANCY:                                      1 OF 1
```

Figure 2-2 — STI Point Detail Display

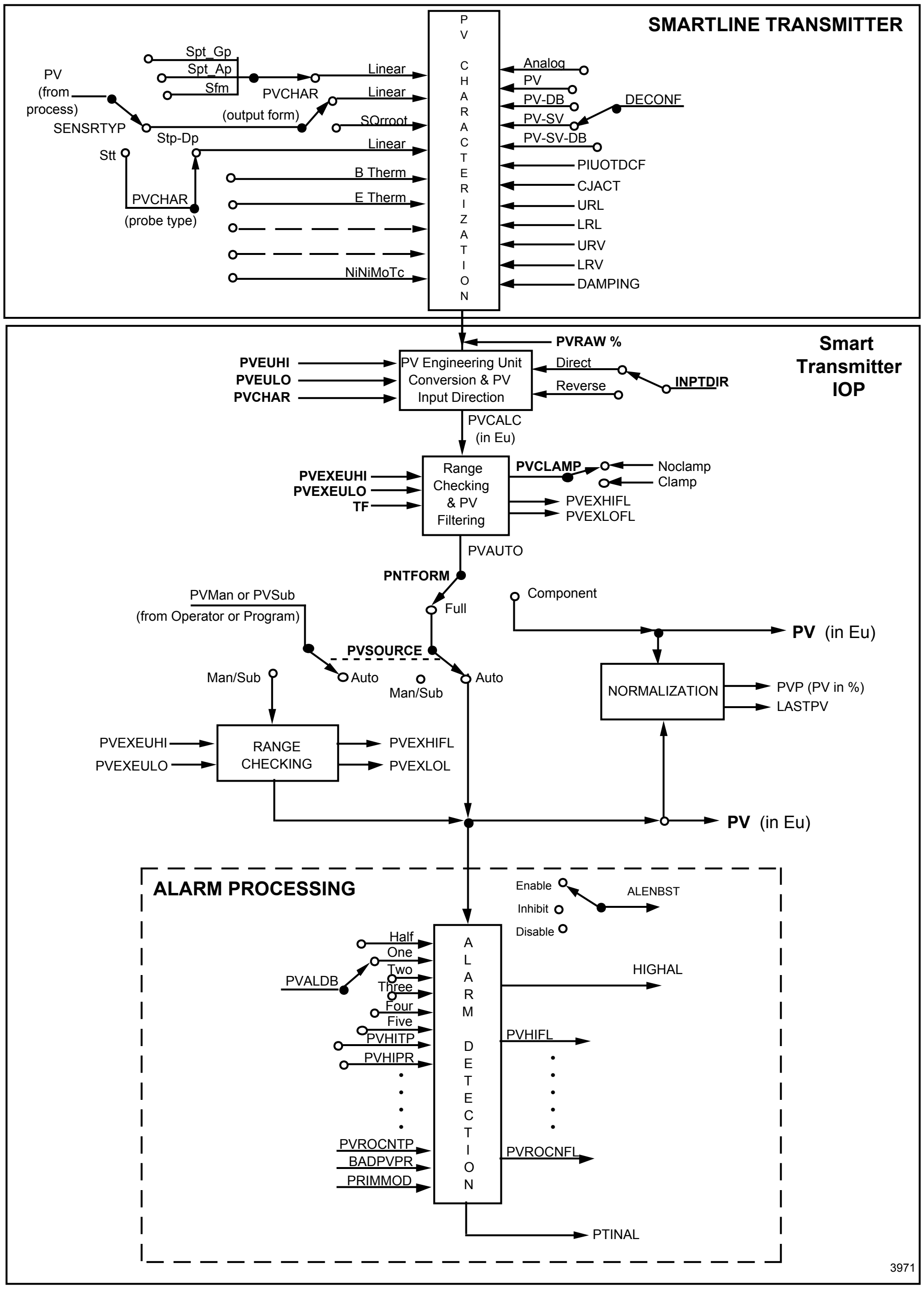


Figure 2-3— Smart Transmitter Input Point, Functional Diagram

3971

2.3.2 STI IOP Functions

A functional diagram of the STI IOP point is shown in Figure 2-3. The STI point performs the following functions:

- PV characterization
- Range Checking and PV filtering
- PV source selection
- Alarm detection

2.3.2.1 PV Characterization

The PV signal (PVRAW) received from the transmitter has been characterized by the transmitter in terms of linear or square-root characterization and damping. For the STT 3000, PVRAW is further characterized based on the entries that the user makes for the SENSRTYP, PVCHAR, and INPTDIR parameters as shown in Figure 2-3. Table 2-3 lists the PV characterization options available for the various transmitter (sensor) types.

2.3.2.2 Linear Conversion

If the entry for PVCHAR is Linear, the PVRAW input from the FTA is calculated as a proportion of the input span in percent, as determined from upper and lower range values URV and LRV. This proportion is then used in generating an identical proportion of the output span, as determined from PVEULO and PVEUHI shown in Figure 2-4. The URV and LRV values are the 100% and 0% values that correspond to the PVEUHI and PVEULO values, respectively.

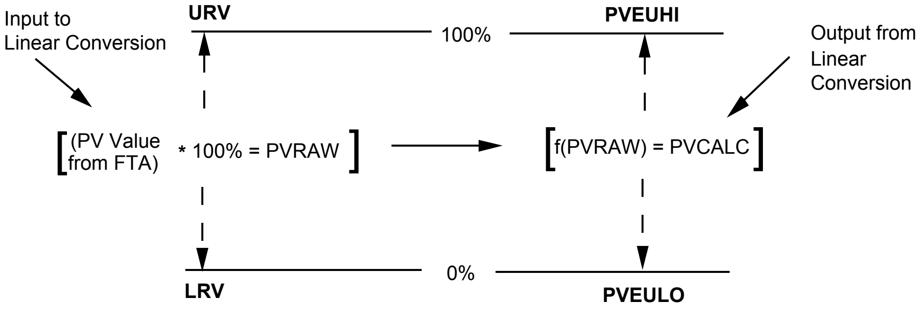


Figure 2-4 — Linear Conversion, STI IOP

Table 2-3 — STI Sensor Types and PV Characterization Options

Transmitter (Sensor) Type (SENSRTYP)	PVCHAR Options	PV RAW[1,3]	PV CALC	PV Detection[2]
Spt_Dp (Differential Pressure)	Linear	% in H_2O	EU	Range check on PVCALC
	Square Root	% in H_2O	EU	Range check on PVCALC
Spt_Gp (Gauge Pressure)	Linear	% in H_2O	EU	Range check on PVCALC
Spt_Ap (Absolute Pressure)	Linear	% in H_2O	EU	Range check on PVCALC
Stt (Temperature)[4]	Linear	% mV	EU	Range check on PVCALC
	Thermocouple	% °C	EU	Open thermocouple detection, and range check on PVCALC.
	RTD	% °C	EU	Range check on PVCALC
	RTD Ohms	% Ohms	EU	Range check on PVCALC
Sfm (Flow)	Linear	% m^3/Hr	EU	Range check on PVCALC

LEGEND: EU = Engineering Units
PVCALC = Calculated PV
PVCHAR = PV Characterization
PVRAW = PV received from transmitter and multiplied by 100 by the STI IOP

Notes:

1. PVRAW is a percentage of the configured range for the sensor type. For Multivariable transmitters, PVRAW Engineering Units are different for each PV slot.
2. If the transmitter gross status indicates Output mode or Bad, PVRAW of the STI point is set to NaN, and PVSTS is set to Bad
3. The normal operating range for PVRAW (0% = PVRAWLO, 100% = PVRAWHI) is configured by the user.
4. For the supported temperature ranges, refer to the definition of the PVCHAR parameter in the *Advanced Process Manager Parameter Reference Dictionary*.

2.3.2.3 Square-Root Conversion

If square root is selected, this function is performed by the smart transmitter in its computation of PVRAW. The value for PVCALC is then determined in the same manner as linear conversion. These conversion equations are provided below.

For INPTDIR = Direct:

$$\text{PVCALC} = \frac{\text{PVRAW}}{100}\,(\text{PVEUHI} - \text{PVEULO}) + \text{PVEULO}$$

If INPTDIR = Reverse:

$$\text{PVCALC} = \frac{\text{PVRAW}}{100}\,(\text{PVEULO} - \text{PVEUHI}) + \text{PVEUHI}$$

2.3.2.4 Thermal Conversion

Thermal linearization is available for the thermocouple and RTD inputs of the Stt (temperature) transmitter. Thermal linearization is selectable by parameter PVCHAR. The following thermocouples are supported for an STI point:

Btherm
Etherm
Jtherm
Ktherm
NiNiMoTC
Ntherm
RhRad
Rtherm
Stherm
Ttherm
W5W26TC
W3W25TC

The STI point calculates the reference junction compensation from the measured reference junction output level. This value is stored and then later converted back to millivolts, with respect to 0 degrees C, for each thermocouple that is to be compensated. The external cold-junction reference compensation (CJTACT) parameter is expressed in millivolts for the specified thermocouple and is added to the millivolt value for PVRAW.

The following 3-wire RTDs are supported by the STI point:

PtDinRTD (Pt100D)
PtJisRTD (Pt100J)
NicklRTD (Ni500)
Pt200RTD
Pt500RTD
Cu10RTD
Cu25RTD

For an RTD, the STI point calculates the lead-wire compensation and then subtracts the value from PVRAW.

2.3.2.5 PV Range Checking and Filtering

PV range checking ensures that the PVCALC output of PV characterization is within the limits defined by parameters PVEXEULO and PVEXEUHI. If either of the limits is violated, the output of the range check is set to BadPV if clamping has not been specified. If clamping has been specified, the output of the range check is clamped.

If the range-checked and filtered value is less than the value specified by the user-configured LOCUTOFF parameter, the final output called PVAUTO is forced to PVEULO.

PV filtering can be implemented at the STI IOP, or at the Smartline Transmitter. At the STI IOP, first-order filtering is performed on PVCALC, as specified by the user through parameter TF (filter lag time). At a transmitter, filtering is performed on the PV depending on the value entered for the DAMPING parameter using the SFC. The user should decide the type of filtering required based on the following guidelines:

- The DAMPING parameter allows for better control accuracy because more PV samples are used in calculating the filtered PV value at the transmitter.
- TF can be changed on-process from the Universal Station. To change the DAMPING value requires the point to be made inactive and requires the database to be down-line loaded to the transmitter after the change has been made.

For better control accuracy, the use of the DAMPING value is preferred over the TF value. The transmitter accepts only certain DAMPING values from the STI IOP, and the value received must first be converted to one of the predefined DAMPING values that reside in the transmitter. This conversion is accomplished automatically by the STI IOP by finding the DAMPING value that is nearest to the desired DAMPING value.

DAMPING values differ between the Smartline Transmitters. The valid DAMPING values for each transmitter type are contained in the following listing.

Valid DAMPING Values*		
Pressure (Spt)	**Temperature (Stt)**	**Flow (Sfm)**
0.0	0.0	0.0
0.16	0.3	0.5
0.32	0.7	1.0
0.48	1.5	2.0
1.00	3.1	3.0
2.0	6.3	4.0
4.0	12.7	5.0
8.0	25.5	10.0
16.0	51.1	50.0
32.0	102.3	100.0

* The values listed are the first-order filter time constants in seconds.

2.3.2.6 PV Source Selection

The PVSOURCE parameter allows the user to select the source of the PV for this data point. As shown in Figure 2-3, the PV can be provided by the Range Checking and Filtering circuit (when PVSOURCE is Auto), or it can be the manually entered PV (when PVSOURCE is Man or Sub).

In addition, the PV source option parameter (PVSRCOPT) determines whether it is permissible to change the PV source to a source other than Auto. PVSRCOPT has two states: OnlyAuto and All. The All state allows the PV to be manually entered for this data point.

2.3.2.7 Alarming

The STI point compares the PV to threshold values and records the alarms in the database of the data point. The alarms are then reported by the Advanced Process Manager Module (APMM). The parameters that are associated with alarming in the STI point are as follows:

ALENBST
BADPVFL
BADPVPR
PRIMMOD
PTINAL
PVALDB
PVALDBEU
PVEXHIFL
PVEXLOFL
PVHHFL
PVHHPR
PVHHTP
PVHIFL
PVHIPR
PVHITP
PVLLFL
PVLLPR
PVLLTP
PVLOFL
PVLOPR
PVLOTP
PVROCNFL
PVROCNPR
PVROCNTP
PVROCPFL
PVROCPPR
PVROCPTP

Refer to the *Advanced Process Manager Parameter Reference Dictionary* for the definitions of these parameters.

2.3.3 Smart Transmitter Communication Modes

All communications between the STI IOP and the Smartline transmitters are in bit-serial form using the Honeywell DE (digital enhanced) protocol. The communication mode for the transmitter is selectable through the DECONF (DE configuration mode) parameter, whose possible entries are as follows:

DECONF Entry	**Definition**
Analog	Not supported
Pv	Transmitter communicates only the PV (4-byte format)
Pv_Sv	Transmitter communicates the PV and the Secondary Variable (SV) (4-byte format)
Pv_Db*	Transmitter communicates the PV and the database (6-byte format)
Pv_Sv_Db*	Transmitter communicates the PV, SV, and the database (6-byte format)

* The use of these two modes is recommended because they offer database mismatch detection and on-process mismatch recovery.

2.3.4 Database Considerations

During normal operation, the STI point database and the corresponding transmitter database contain the same information. If under special conditions, the databases are not the same, a status message appears on the STI point's Detail Display to indicate a database mismatch. Three possible ways that a database mismatch can occur between the STI IOP database and the transmitter database are as follows:

- Smart Field Communicator (SFC) is used to change PV-related values in the transmitter.
- A write operation from the Universal Station to the STI IOP (during a checkpoint restore or point building).
- The same transmitter has been reinstalled after bench calibration, or after the transmitter electronic module has been replaced and the LRV/URV values have been modified.

The following parameters of the STI IOP database and the transmitter database are compared when the STI IOP checks for database mismatches:

CJTACT
DAMPING
DE_CONF
FREQ60/50
PIUOTDEN
PVCHAR
SENSRTYP
STITAG
URL
URV, LRV

If a database mismatch is detected, the first four parameter mismatches are displayed in the S1 status field of the STI point's Detail Display at the Universal Station. These mismatched parameters are preceded by the message "DATA BASE DISCREPANCY." The user can correct the mismatch by one of the following methods:

- Download the STI IOP database to the transmitter by issuing a DnLoadDb command through the COMMAND parameter when the point is in an Inactive state. If the loading is successful, the point STATE changes to OK when the point is made active.
- Upload the transmitter database to the STI IOP by issuing an UpLoadDb command through the COMMAND parameter when the point is in an Inactive state. If the loading is successful, the point STATE changes to OK when the point is made active.
- Correct the mismatched parameter using the SFC. If the parameter or parameters are updated by the SFC to the same value as in the STI IOP database, the state changes from Mismatch to OK when the next database transfer from the transmitter occurs.

If a mismatch occurred because the Smart Field Communicator (SFC) was used to change a parameter in the transmitter database, the data in the transmitter database may be correct. In this case, the STI IOP does not know which value to use and issues an "SFC MODIFIED XMTR DATABASE" message, which appears on the Detail Display. The user should wait until the next database transfer occurs from the transmitter before taking action. At that time, the specific change is displayed.

A database mismatch can also occur because the transmitter was miswired during installation. This can be fixed by correcting the transmitter wiring.

For the MagneW3000, the URL value displayed on the SFC is a factor of 10 higher than the actual value displayed at the Universal Station. The value displayed at the SFC is the scaled limit while the value displayed at the Universal Station represents raw data. Also, the STI IOP does not respond to changes made in the MagneW3000 database parameters for up to 3 minutes when the database changes are made using the Local Setting Card. This card is an option that is available with the MagneW3000.

2.3.5 Point States

The STATE parameter indicates the current status of the STI IOP and the transmitter. The various states are listed in Table 2-4.

Table 2-4 — STI IOP Point States

States	Description
OK	Normal state; indicates that the STI point and the transmitter are OK. Transmitter is updating the PV value at the STI point. STATE remains OK when the point is made inactive.
DBChange	Indicates that a database mismatch between the STI point and the transmitter has been detected. Transmitter is not updating the PV value at the STI point. STATE remains DBChange when the point is made inactive.
Loading	Indicates that database loading between the STI point and the transmitter is occurring.
Loadcomp	Indicates that the database transfer between the STI point and the transmitter has been successfully completed.
Loadfail	Indicates that the parameter transfer between the STI point and the transmitter has not been successfully completed.
Calib	Indicates that certain parameters are being calibrated at the transmitter by the STI point.
Calcomp	Indicates that the calibration has been successfully completed.
Calfail	Indicates that the calibration has not been successfully completed.

2.3.6 STI IOP Commands

The COMMAND parameter allows the engineer to load configuration parameters in the smart transmitter and to calibrate the transmitter. (The point must first be placed in the Inactive state through the PTEXECST parameter.) The enumerations of the COMMAND parameter are as follows:

COMMAND Selection	Description
DnLoadDB	Load the STI IOP database into the transmitter (down-line load).
UpLoadDb	Loads the transmitter database into the STI IOP (up-line load).
Set_LRV	Set the Lower Range Value
Set_URV	Set the Upper Range Value
Cor_LRV	Corrects the the Lower Range Value
Cor_URV	Corrects the Upper Range Value
Cor_Inpt	Correct the zero point
RstCor	Sets all input calibration parameters to their factory default values.
Null	A command has not been issued by the STI IOP.

The result of issuing a command to an STI IOP point is reflected in the STATE parameter for the point.

2.4 ANALOG OUTPUT POINT

The analog output point converts the output value (OP) to a 4-20 mA output signal for operating final control elements such as valves and actuators in the field. The OP parameter value can be controlled from a APM regulatory point, an AM regulatory point, the operator, or a user program, depending on the selected mode and the entries for the RCASOPT and PNTFORM parameters.

To convert the OP value to a 4-20 mA signal, the analog output point performs

- Direct/reverse Output Function
- Nonlinear Output Characterization

An option allows redundant Analog Output points (see section 3.6). Figure 2-5 is a functional diagram of the analog output point.

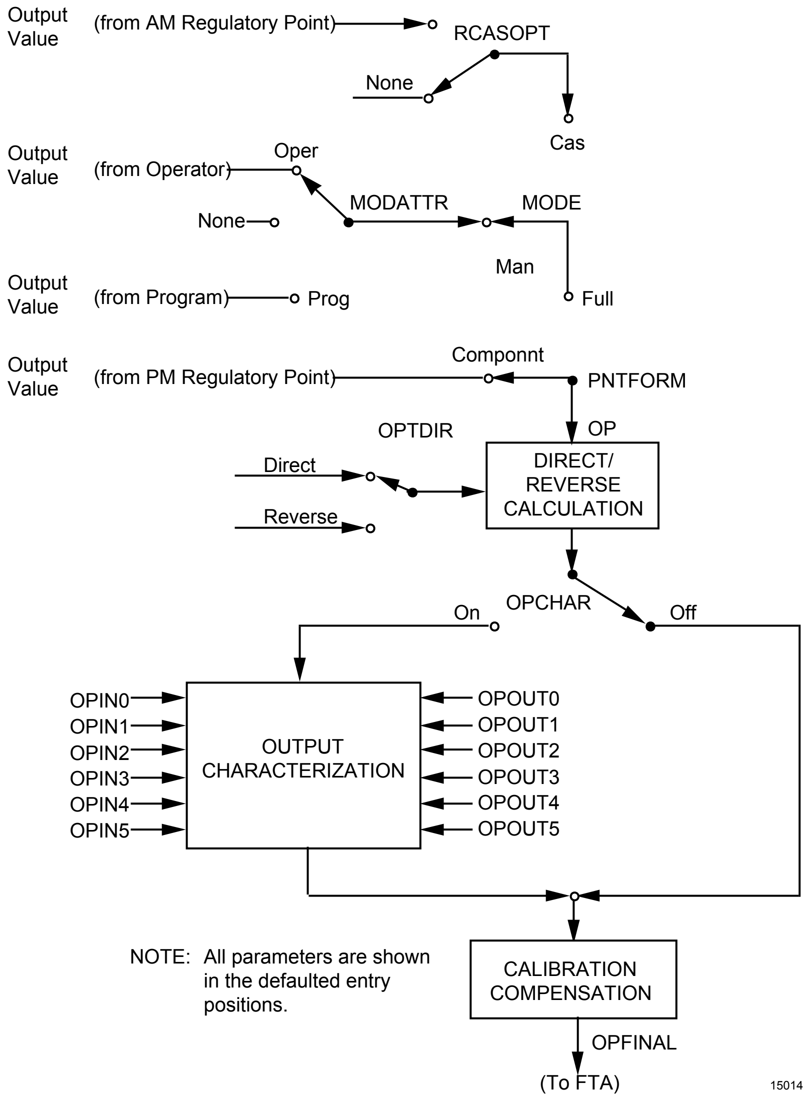
2.4.1 Direct/Reverse Output

Parameter OPTDIR allows the user to specify whether the output of the data point is direct acting (where 4 mA = 0%, and 20 mA = 100%) or reverse acting (where 4 mA = 100%, and 20 mA = 0%). The default mode is direct acting.

2.4.2 Output Characterization

Output characterization allows the user to specify an output transfer function, using configurable X-Y coordinates that provide five linear segments as shown in Figure 2-6. The length of each segment is variable according to the coordinates that can be entered as applicable constants for OPOUT1-4 and OPIN1-4 parameters, which are real numbers.

As shown in Figure 2-6, the end points of the curve are fixed at coordinates OPOUT0,OPIN0 (at -6.9%) and OPOUT5,OPIN5 (at 106.9%). These coordinates are fixed at these values to ensure that neither the characterization function nor its inverse can provide output values which are outside the -6.9% to 106.9% range.



NOTE: All parameters are shown in the defaulted entry positions.

Figure 2-5 — Analog Output Point, Functional Diagram

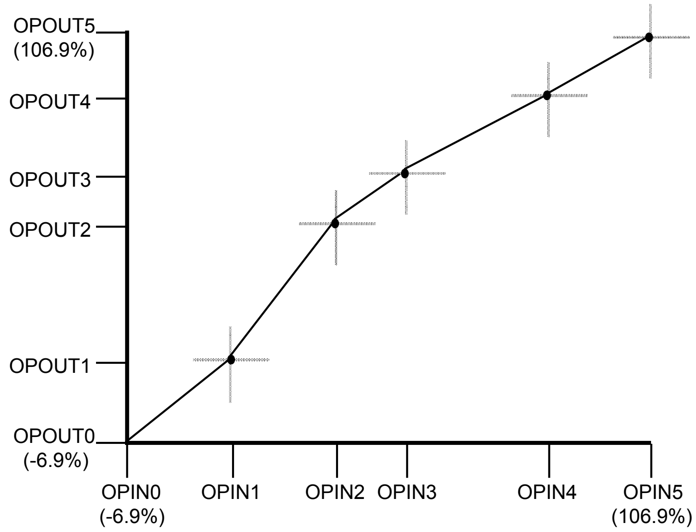


Figure 2-6 — Output Characterization for Analog Output Point 2071

Depending on the output value, the analog output point interpolates linearly between the two nearest values. The interpolated value becomes the output value OPFINAL.

Output characterization is an optional function that can be implemented by setting parameter OPCHAR to On. Refer to Figure 2-5.

2.4.3 Calibration Compensation

The final stage of output processing in the analog output point is calibration compensation. This is accomplished in the data point using internal offset and scale constants. The output value OPFINAL is then routed to the field through the appropriate FTA.

NOTE

Slot or module level soft failures can prevent a point (or points) from outputting to the field. The regulatory control point will initiate a "Bad Output" alarm (If configured) when any connection is broken. If all configured point connections to the field are broken, the regulatory control point driving that analog output slot goes into initialization.

2.4.4 Disable Output Alarming when Using Open-Line Detection

With R640, disabling annunciation of events for an IOP-resident AO point is performed using the $OLDEVDS parameter which disables the alarms associated with an open AO output line. This function allows you to disable the OUTPUTFL soft failure alarms generated by the AO IOPs when transmitters or positioners are physically disconnected for normal maintenance.

The open-line detect parameter has two states, ENABLE and DISABLE. By default, it will be in ENABLE state. Setting the parameter to DISABLE will disable the annunciation events. Only the OUTPUTFL soft failure alarm events are disabled.

The NIM resident parameter $OLDEVDS can be configured through DEB and the TPS Builder. This parameter is valid for both full and component point forms.

2.5 DIGITAL INPUT POINT

A digital input point converts a digital PVRAW signal received from the field to a PV that can be used by other data points in the APM and the system. A functional diagram of the digital input point is shown in Figure 2-7.

Bad PV Flag—Control strategies can test for a bad Digital Input PV. Parameter BADPVFL is set ON when—

- The PV source has been switched to Substituted, and the point is inactive or the module status is Idle.
- The PV source is AUTO and the PV is not being updated, because either the point is inactive, the module is idle, there is a slot soft failure, or the FTA is missing.

The digital input point is a single-input point that can be configured as a status input, a latched input, or for accumulation, as described in the following paragraphs.

2.5.1 Status Digital Input Point

For this digital input type, the PVAUTO value represents the state of the raw input signal after the direct/reverse conversion is performed. The status digital-input point can be configured for PV source selection, detection of off-normal alarms, and for reporting any PV state changes to the system. The status digital input point is selected by entering Status for the DITYPE parameter.

The current state of the PV input is represented on the Universal Station Group and Detail Displays as two boxes, as shown in Figure 2-8. The boxes are lighted or extinguished depending on the current state of PVRAW and the input direction as configured through the INPTDIR parameter, as shown in the chart in Figure 2-8. The current PV state is also available to be used as an input to logic slots, and other APM control functions.

2.5.1.1 PV Source Selection

The PV source parameter (PVSOURCE) option determines the source of the PV for a status input point. The source can be the PV input from the field (PVauto), the PV state entered by the operator (PVman), or it can be supplied by a user program (PVsub). PVSOURCE has no effect on the latched and accumulation options of the digital input point. If PVSOURCE is PVauto, PV tracks PVRAW.

2.5.1.2 Off-Normal Alarming

Enabling, Disabling, and Inhibiting Off-Normal Alarms—The ALENBST parameter allows the operator to enable (permit), disable, or inhibit the off-normal alarm. Disabling the alarm still allows the alarm to be listed on the Group and Detail displays. Inhibiting the alarm prevents the current PV state from being compared to the configured normal state.

Off-Normal Alarming & PV Change Reporting—Off-normal alarming can be selected for the digital input point through the ALMOPT parameter. An off-normal alarm is generated when the input PV state is different than the configured normal (desired) state for the point as specified by the PVNORMAL parameter. The priority of the off-normal alarm is determined through the OFFNRMPR parameter (ALPRIOR may also be used to maintain compatibility with schematics and CL programs designed for pre-R500 software).

Additionally, all PV state changes can be historized as determined by HM volume configuration.

Change Of State Reporting—Digital Input Status points (and Sequence of Events points) can be configured for Change Of State (COS) alarm reporting through the ALMOPT parameter. The alarm is generated when the input changes state in either direction. Alarm priority is determined through the OFFNRMPR/ALPRIOR parameters as before.

COS alarms are removed from the Alarm Summary display following acknowledgement. The Point does not remain in alarm so there is no Return-to-Normal. Point Detail or Group displays will never show a point in COS alarm.

Older digital input IOPs may need to have a new firmware chip for COS reporting. Check the IOP's detail display. For COS reporting the Digital Input IOP firmware revision must be 5.0 or later.

Note that when a point with COS reporting is changed from Inactive to Active, a COS alarm is generated if the PV = 1. There is no COS alarm if the PV = 0. The same alarming occurs if the point is active and the IOP is put into Run mode.

Alarm Delay—When off-normal alarming has been configured and an off-normal alarm is detected, the event is reported to the system. Further off-normal alarms for the same data point are not reported until the time delay (0 to 60 seconds) specified by the DLYTIME parameter expires. When the time delay expires, the time-delay function is disabled and the off-normal alarm for the data point can again be reported.

For Change of State alarms, when a PV state change occurs, a COS alarm is produced and the delay timer is started. When DLYTIME expires, two situations are possible—

- the PV is in the same state and future state changes are immediately alarmed.
- the PV is in the opposite state (it may have changed many times during the DLYTIME period) so a second COS alarm is produced and the timer starts again.

2.5.1.3 Event Reporting

The EVTOPT parameter for the status input allows the user to optionally specify the tag name (EIPPCODE) of a data point in the system that is to be notified when the PV changes state, and/or specify that a time stamp be added to the reported PV state change.

For a status input point, EVTOPT has the four possible entries: None, EIP, SOE, and EIPSOE. EIP specifies that the user supply the tag name of the data point in the system that is to receive the PV state change, while SOE specifies that a time stamp is added to the PV state change to establish a sequence of events.

Change of State Reporting can initiate Event Initiated Processing if the point is configured for both COS and EIP. Note that digital input points that have COS (or Off Normal alarming) and EVTOPT EIP configured will have EIP triggered twice. EIP processing associated with COS reporting is not recoverable for NIM/APM failovers. Full EIP recovery is available for DISOE points.

2.5.2 Latched Digital Input Point

To capture the occurrence of momentary digital inputs, such as from pushbuttons, requires the user to configure the digital input point as a latched digital input point. Configuring the point as a latched point is accomplished by entering Latched for the DITYPE parameter.

When configured as a latched input point, an input pulse that is on for a minimum of 40 milliseconds is latched true for 1.5 seconds. This ensures that any control function that needs to monitor this event will execute at least once during the time that the signal is latched on.

The current state of the latched PV input is represented on the Universal Station Group and Detail displays as two boxes, as shown in Figure 2-8. The boxes are lighted or extinguished depending on the current state of PVRAW and the input direction as configured through the INPTDIR parameter, as shown in the chart in Figure 2-8. The current PV state is also available to be used as an input to logic slots, and other APM control functions.

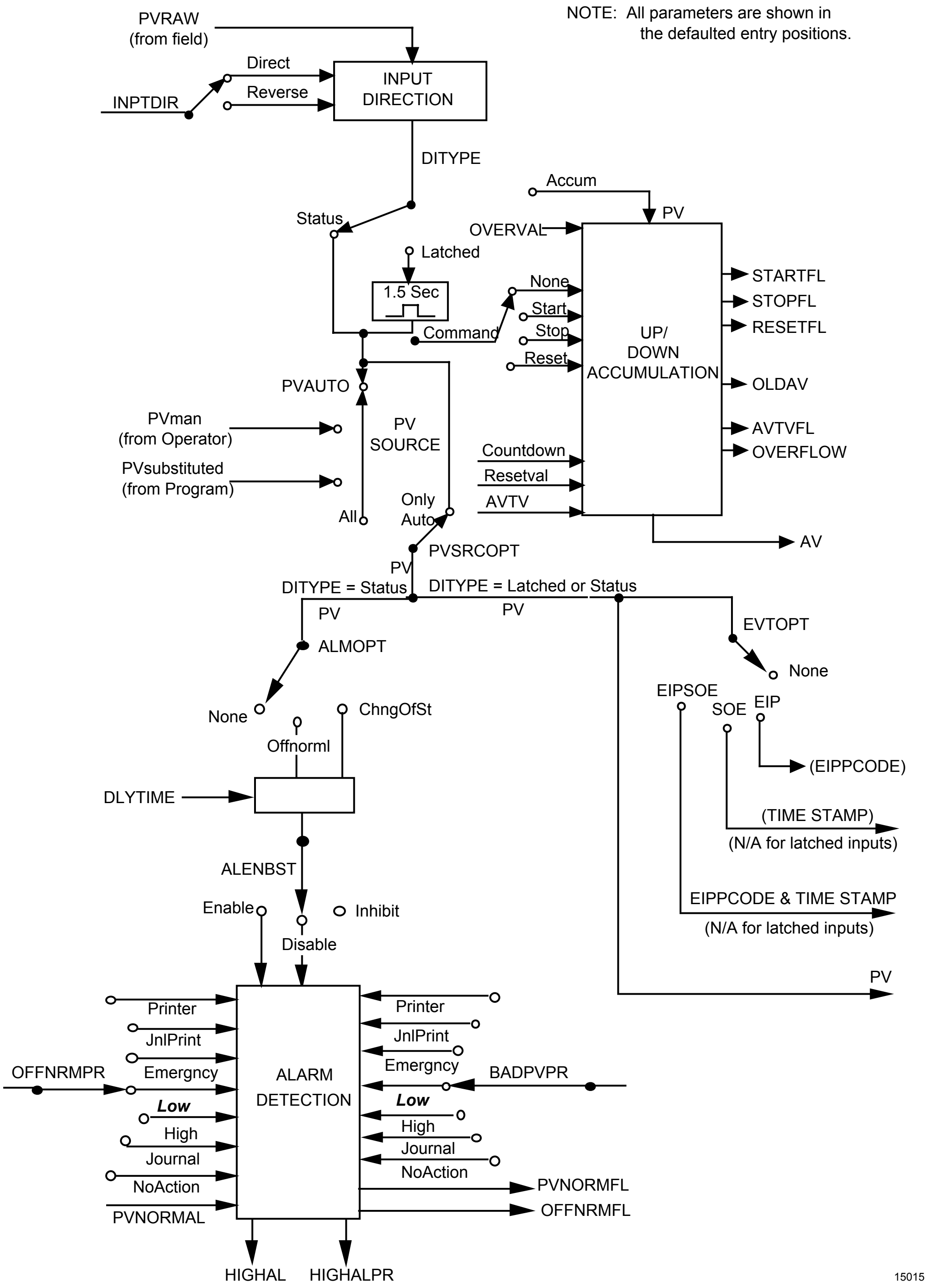
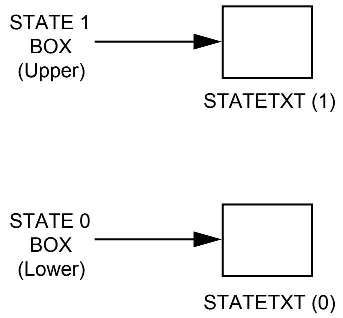


Figure 2-7 — Digital Input Point, Functional Diagram



PVRAW State	INPTDIR = Direct	INPTDIR = Reverse
ON	PV = ON. State 1 Box is lit. State 0 Box is extinguished.	PV = OFF. State 0 Box is lit. State 1 Box is extinguished.
OFF	PV = OFF. State 0 Box is lit. State 1 Box is extinguished.	PV = ON. State 1 Box is lit. State 0 Box is extinguished.

Figure 2-8 — Input States

2065

2.5.2.1 Event Reporting

The EVTOPT parameter for the latched input allows the user to optionally specify the tag name (EIPPCODE) of a data point in the system that is to be notified when the PV changes state. For a latched input point, EVTOPT has two possible entries: None and EIP.

2.5.3 Accumulation Digital Input Point

The accumulation type digital input point counts the transitions of the digital input pulses received from the field. The digital input point is configured for accumulation by entering Accum for the DITYPE parameter.

2.5.3.1 Up/Down Accumulation

The accumulation type digital-input point can be incremented or decremented depending on the entry made for the COUNTDWN parameter. If the entry for this parameter is Off, the point is to be incremented by the field transitions; if the entry is On, the point is decremented. The accumulated or decremented value is the value of parameter AV.

2.5.3.2 Accumulator Commands

The operator has the capability of issuing start, stop, and reset commands to the accumulator for starting, stopping, and resetting the accumulation. For resetting the accumulator, the operator can specify the value to which the accumulator is reset by using the RESETVAL parameter to enter the value. When the accumulator is reset, the old accumulated value is displayed using the OLDAV parameter.

2.5.3.3 Target Value

The operator can specify the accumulator target value through the AVTV parameter. When the accumulated value (PVAUTO) reaches the target value (PVTV), parameter AVTVFL is set to On. AVTVFL is set to Off when the accumulator is reset.

2.5.3.4 Overflow

If the accumulated value AV overflows the accumulator (AV > 999999), the OVERFLOW parameter is set to On. In addition, it is set to 0 when the accumulator is configured for counting up (incrementing); it is set to 32767 if it is configured for counting down (decrementing).

2.5.4 Sequence Of Events Digital Input Point

2.5.4.1 Description

Sequence of Events (SOE) points are used to report the order of occurrence of physical events. The SOE digital input IOPs can use the same type FTAs as digital input cards, but the best overall performance is obtained when using the high resolution 24 VDC Digital Input FTAs.

SOE events are recorded in a journal with a timestamp so that you can determine, for example, which event started an upset and the progression of events thereafter. The record includes the point ID, point descriptor, state text unit, and time of occurrence to one ms resolution. Refer to the *Engineer's Reference Manual* for a discussion of SOE Journal size requirements. For SOE Journal and Process Unit Journal Configuration requirements refer to the *Network Configuration Forms Instruction* manual.

SOE points can also be configured for Change of State Reporting (an alarm is generated when the input changes state in either direction). Refer to the discussion under Status Points.

Each sequence of events IOP card provides 32 status and/or latched type inputs.

2.5.4.2 Timestamping

Timestamps are based on the "wall clock" time entered in the LCN. Time is broadcast over the UCN every six seconds by the NIM with the lowest UCN address. To do this, the time synch parameter (TIMESYNC) must be enabled during NIM configuration and the NIM must contain an EPNI card. If there is a redundant NIM, its TIMESYNC parameter should also have been enabled in case the primary NIM fails.

The APMM receives time sync messages from the NIM and broadcasts time synch messages on its own I/O Link every two seconds to the Sequence of Events IOPs. The IOP time stamps an input state change with a code that is converted to, and reported as, wall clock time by the APMMs. Each IOP can store up to 20 seconds of state changes (for all 32 points) before older events are overwritten.

If an APMM fails, or if event collection by the APMM is stopped or delayed, or if the buffer in the IOP becomes full, the IOP rejects newer events (that is, it saves the older events).

If the DISOE IOP hardware detects loss of one or more physical events, then an SOE Lost event packet is passed in the System Status Journal (not the SOE Journal). This packet contains the time at which lost events were detected, and should be used in conjunction with the SOE journal to correlate the SOE events. SOE lost events are usually caused by a chattering input.

2.5.4.3 Specifications and Reporting

Resolution

The operator sees one millisecond resolution as reported in the SOE Journal. Event time is reported in hours, minutes, seconds, and milliseconds. The entries are further sorted at a submillisecond level that affects the order of entry but does not appear in the journal.

Reporting

Typical SOE Journal entries appear as follows:

```
Date (MM/DD/YY)
10:50:21:848 SE150501       ON
10:50:21:849 SE150501       OFF
10:50:21:850 SE150502       ON
10:50:21:851 SE150502       OFF
```

Table 2-5 shows the guaranteed Minimum Physical Event Separation (MPES) values using various Field Termination Assemblies for two digital state change events occurring between two points. In one case the events are on the same IOP. In the other case, assume that the events are in different APMMs on two separate UCNs, logical or physical, or in case of NIM failover. Assume that both points are using the same type of FTA. These specifications implicitly include DISOE IOP and/or APMM failovers, are valid for stable system operating conditions, and represents the guaranteed worst-case conditions.

Table 2-5 — Worst Case Resolution for Different System Configurations

	MPES	**MPES**
Digital Input FTA Type and Model #	Two Turn-On Events In Same IOP	Two Events (On/Off) On Different UCNs
High Resolution 24 Vdc (MU/C-TDID12/52)	3.0 ms	6.5 ms
Galvanic 24 Vd c (MU/C-GDID12/13/82/83)	6.6 ms	7.4 ms
Low Cost 24 Vdc (MU/C-TDID72)	10.6 ms	10.6 ms
120 Vac (MU/C-TDIA12/52)	44.1 ms	44.1 ms
240 Vac (MU/C-TDIA22/62)	44.1 ms	44.1ms

Figure 2-9 is a functional diagram of the SOE Digital Input point.

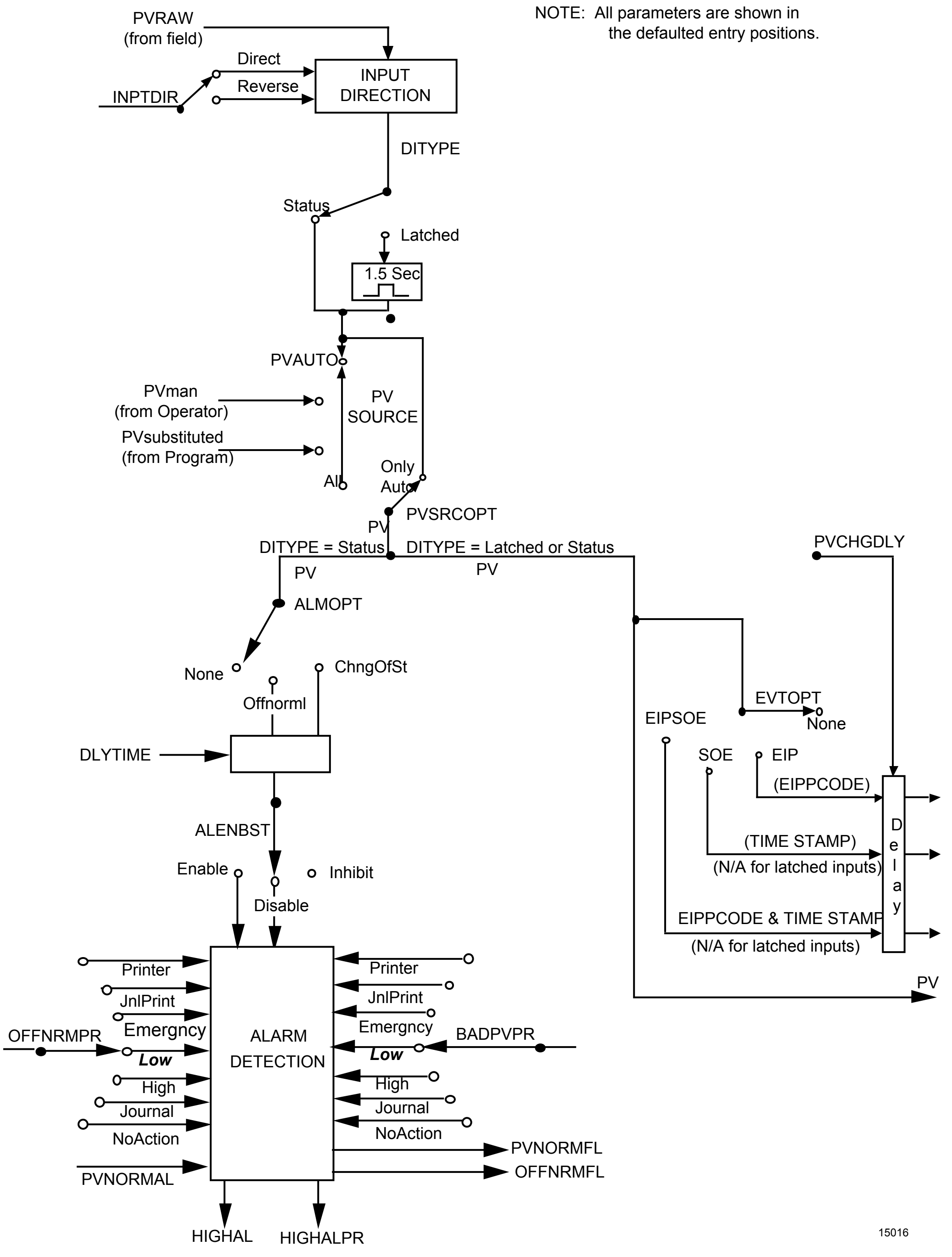


Figure 2-9 — SOE Digital Input Point, Functional Diagram

2.5.4.4 Configuration

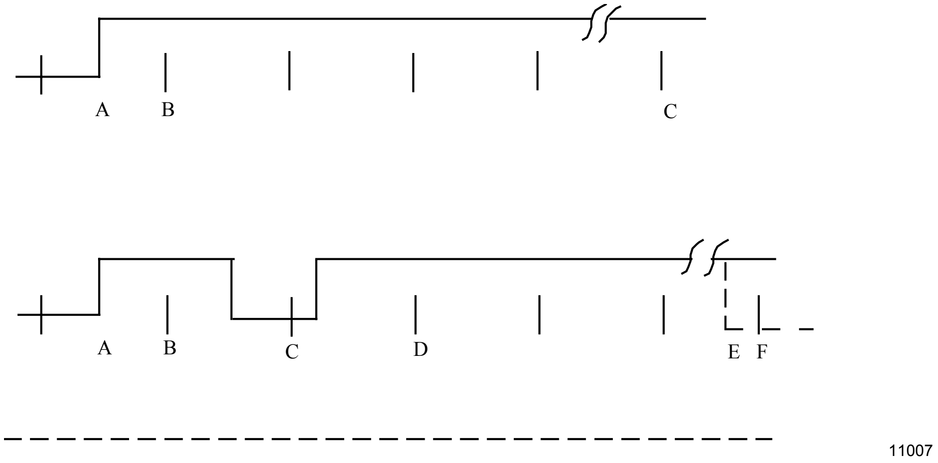
Digital input SOE IOP points are configured by selecting the DIGITAL INPUT target from the NIM Point Build Menu. The configuration considerations are similar to those described for the conventional digital input point, with the following additions:

PNTMODTY—For Point Module Type, select DISOE.

DITYPE—Choose STATUS as the Digital Input Type. Note that the SOE IOP can be used as a conventional digital input point if you choose Latched.

DEBOUNCE—The contact debounce time parameter specifies the time interval used to debounce an input from mechanical contacts of a field input source. It is defined as the length of time following an input state change during which the input must remain unchanged in the new state to declare it a valid event. DEBOUNCE has a range of 0 - 50 milliseconds in one millisecond increments. The default value of 10 ms should suffice for most contacts. If not, choose a value slightly longer than the manufacturer's specified contact bounce time.

The following drawings illustrate debounce operation—



This waveform represents the field input. Tick marks across the waveform indicate the 200 microsecond scan intervals of the DISOE IOP hardware. Assume that the input state changes at point A.

At point B, the state change is detected. At this point, the current time and old state are recorded. The debounce timer is started.

- if the input remains at a steady state until the debounce timer expires, then an event is generated with a timestamp corresponding to the time of detection (point B).
- if the input changes before the debounce timer expires (point c), then the change of state event detected at B is discarded, the timer is restarted and runs for the full debounce time—

  If a new input detected at D remains in a steady state until the debounce timer expires (point F), then an event is generated with a timestamp corresponding to the original time of detection (point D).

  If the input has returned to the old state (dotted line at E) when the debounce timer runs out (point F), no event is generated.

PVCHGDLY—The PV change delay parameter specifies the time of separation in seconds for reporting two consecutive PV change events from the same input source. It is intended to prevent repeated rapid reporting of PV change events (i.e., chattering). It can be configured over a 0–60 second range in one second increments. It applies to points configured as EIP, EIPSOE, or SOE only. Ideally, PVCHGDLY AND DLYTIME should have the same value.

When a PV state change is detected, the change is reported and the PV change delay timer is started.

If the PV does not change before the delay timer runs out, no further action is taken.

If the PV state changes more than once before the PV change delay timer runs out, only the last state change is noticed and timestamped but not yet reported. When the delay timer expires—

If the PV state changes more than once before the PV change delay timer runs out, only the second state change is noticed and timestamped but not yet reported. When the delay timer expires—

If the PV state is different from the state that started the timer, this event is reported.

If the PV state is the same as the original PV state that started the timer, no event is reported.

2.6 DIGITAL OUTPUT POINT

The digital output point provides a digital output to the field based on the origin of the input and the configured parameters. A functional diagram of the digital output point is shown in Figure 2-10. The digital output point does not have any modes.

There are two types of digital output points: pulse-width modulated (PWM) output and status output. Selection of the output type is accomplished through the DOTYPE parameter shown in Figure 2-10. The PWM type is used in combination with RegCtl algorithms to provide true proportional control. The status type output is the normal configuration for digital outputs that are linked to digital composite points. Actual output action can be status, latched or momentary, depending on the configuration of the digital composite point. The default for untagged component DO points is Status.

2.6.1 Pulse Width Modulated (PWM) Output Type

The pulse width modulated output type can receive its input from a APM regulatory point (that has been configured for a PID type algorithm) through a user-specified output connection. The length of the pulse is derived from the OP parameter provided by the regulatory point. Because OP is in percent, the percent value becomes the percent on-time for the pulse whose period (1 to 120 seconds) is specified by the PERIOD parameter, as shown in the timing diagram in Figure 2-10.

The output direction of the output signal can be configured to be direct or reverse acting by using the OPTDIR parameter, as shown in the timing diagram in Figure 2-10.

The pulse on-time for direct and reverse acting outputs is calculated as follows:

For direct action:

$$\text{Pulse On-Time} = \frac{\text{OP\% * PERIOD}}{100}$$

For reverse action:

$$\text{Pulse On-Time} = \frac{\text{100\% - OP\% * PERIOD}}{100}$$

If the value of OP is less than 0%, it is clamped to 0%; an OP with a value greater than 100% is clamped to 100%.

2.6.2 Status Output Type

The status output type can be controlled from a digital composite-point output, a logic slot output, or a RegCtl point (that has been configured for the PosProp algorithm) as determined by the output connection. The output latch function is obtained by linking digital composite-point output connections to the SO parameter. Pulsed operation (pulse-on or pulse-off) can be obtained by linking the output connections to the ONPULSE and OFFPULSE parameters, respectively.

The ONPULSE parameter sets SO to On for the specified duration, as shown in Figure 2-10. At the end of the pulse time, SO is set to Off. If ONPULSE is specified as 0.0, SO is immediately set to Off. This also applies to the OFFPULSE, except that the OFFPULSE sets SO to Off.

If SO is received from a logic slot, the SO output of the digital output point tracks the SO output provided by the logic slot.

2.6.3 Initialization Request Flag

When ON, this parameter indicates (for Status Output type points) that control strategies in the APM cannot manipulate the output. Parameter INITREQ is set ON when—

- a PWM type output is configured

- a Status Output type is configured and—

  the point is inactive

  the module is idle

  there is a soft failure such that the point is not working.

Digital Composite Points and Position Proportional Points—Note that Digital Composite points or Position Proportional control algorithms are automatically forced to initialize when outputting to a Digital Output point whose INITREQ is ON.

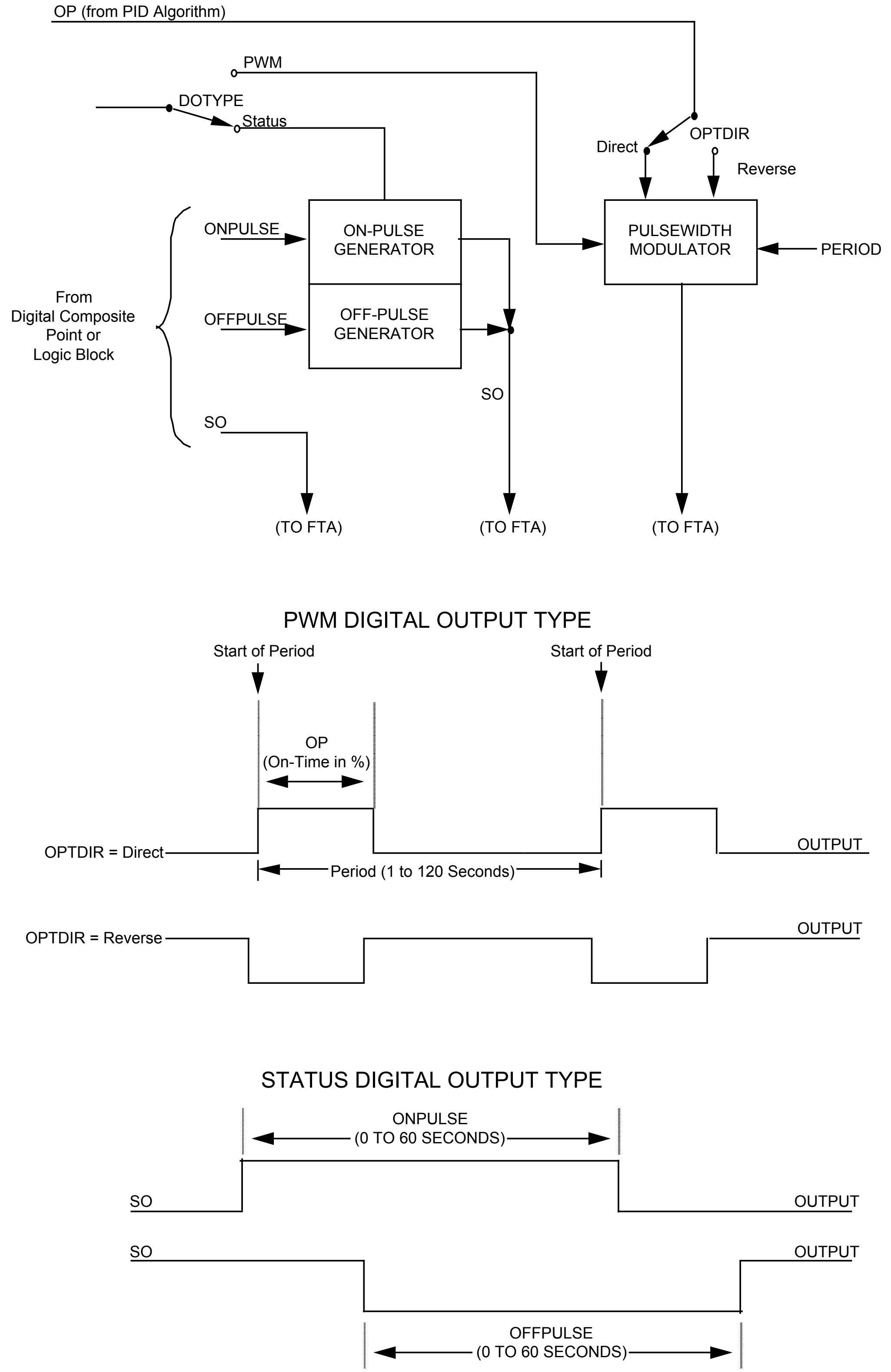


Figure 2-10 — Digital Output Point, Functional Diagram and Waveforms

2079

2.7 PULSE INPUT POINT

The Pulse Input IOP accepts variable frequencies from a variety of devices such as turbine meters, vortex type flow meters and positive displacement meters and converts input pulses to flow rate in engineering units. Each Pulse IOP module contains eight input points. The input frequency supported is up to 20 KHz per channel. Alarming and filtering are provided by the IOP. Pulse Input points are configured under the Analog Input point selection on the NIM Process Point Building Menu. Figure 2-11 is a functional drawing of the Pulse IOP.

2.7.1 Operation

For each point, the IOP is able to calculate flow rate in engineering units based on user selected scaling factors. The Pulse IOP module always provides two related functions simultaneously: totalizing and frequency counting.

2.7.1.1 Totalizing

Totalizing means the IOP maintains a 32-bit accumulation (AV) for each channel (updated every 20 ms). The least significant bits are provided by the hardware accumulator. The 32-bit entity continually rolls (i.e., there is no start, stop, or reset). The Pulse IOP only sets the parameter AV to zero when the point is inactive, the IOP is in Idle, or an error is detected. In the case of an error, status parameter AVSTS is set to BAD. Refer also to the Regulatory PV Totalizer algorithm description in subsection 7.7.7.

2.7.1.2 Frequency Counting

Frequency counting is expressed by parameter PVRAW.

$$\text{PVRAW} = \frac{\text{Delta\_AV}}{\text{Elapsed Time in Seconds}} = \text{Pulses Per Second}$$

The frequency of the input pulse train must be greater than 0.4 Hz for the calculated PV for that point to be correct. The AV is correct under all circumstances.

2.7.1.3 PVCALC

PVCALC is PVRAW after scaling (e.g., barrels per minute)—

$$\text{PVCALC} = \frac{\text{C1}}{\text{C2}} * \text{TIMEBASE} * \text{PVRAW}$$

C1 is an engineering units scale factor and is dimensionless, e.g., barrels per gallon.

TIMEBASE is a time scale factor. The choices are seconds, minutes, or hours.

C2 is a meter factor in pulses per engineering unit (e.g., pulses per gallon). If the instrument vendor instead supplies the factor K in engineering units per pulse (e.g., gallons per pulse), then C2 must represent the reciprocal of the factor K, i.e., C2 = 1/K.

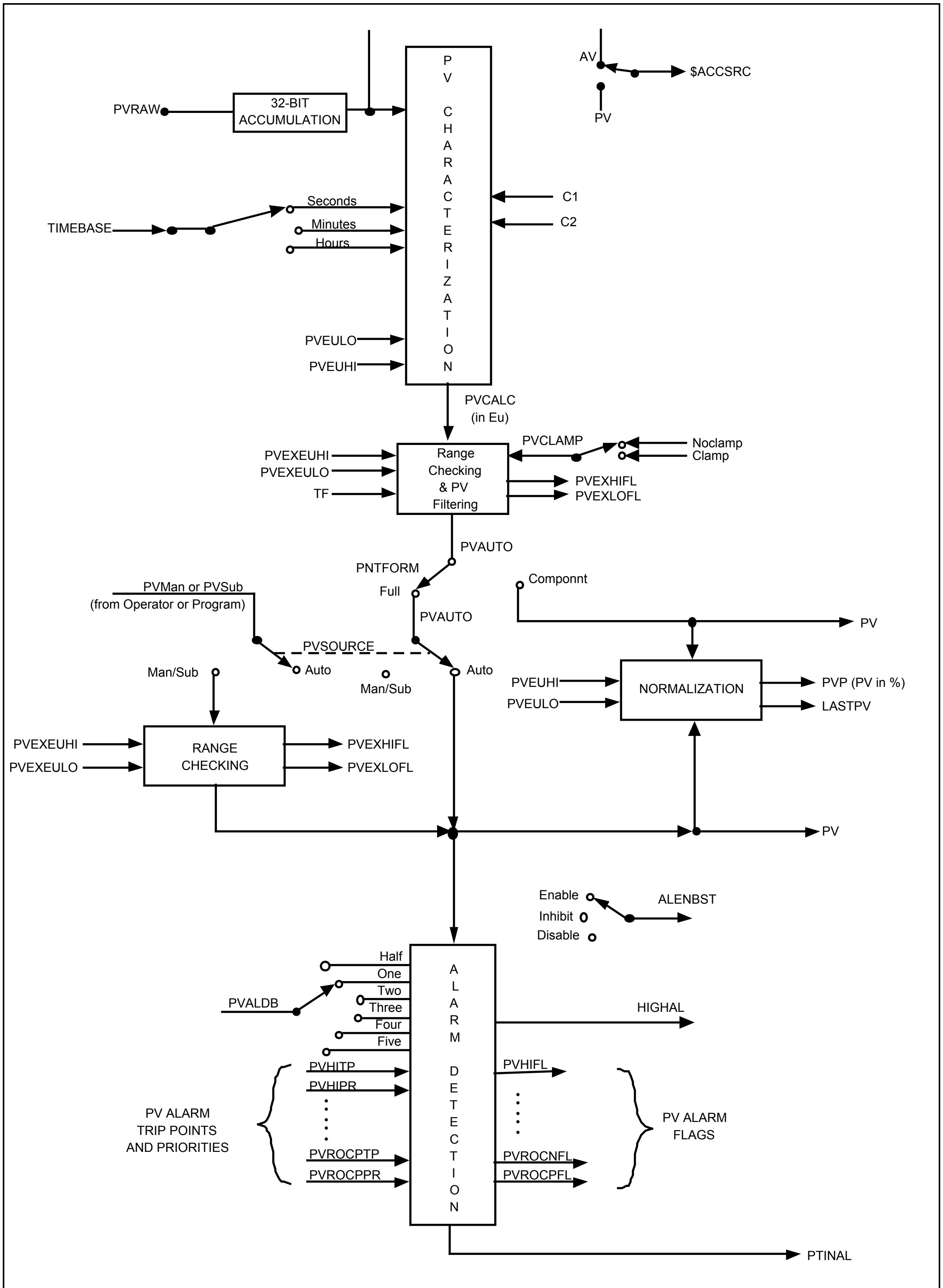


Figure 2-11 — Functional Diagram, Pulse Input IOP

2.7.1.4 Rate Value

The rate value is treated similar to an analog input PV and is supported by high/low alarms, Rate of Change (ROC) alarming, filtering, etc., on a half second processing interval (every four seconds for ROC only).

The Pulse IOP calculates the unsigned integer AVDELTHS every half second. AVDELTHS is the change in AV from the last half second and is intended for display purposes. The parameter's value is normally constant, but the operator will observe a change if the pulse input rate varies.

2.7.1.5 Pulse Period

The pulse period (1/PV) can be determined with CL.

2.8 SERIAL INTERFACE

The Serial Interface IOP provides bidirectional interfaces to various programmable logic controllers (PLC) and other Honeywell-approved serial devices that are compatible with one of several interfaces.

Figure 2-12 is a block drawing of the Serial Interface IOP and associated hardware.

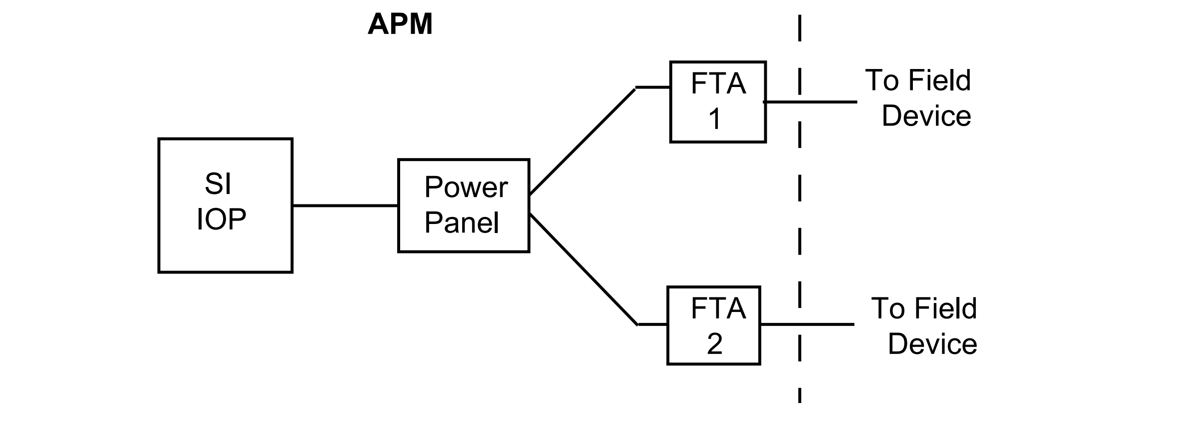


Figure 2-12 — Serial Interface Point Hardware, Block Diagram

Communications to the field device is by EIA RS-232D or EIA RS-485 standards*. RS-232D communication has a limit of 15 meters (50 feet) and RS-422/485 protocol is supported to 1.2 km (4000 feet). You can extend EIA RS-232D Communications between an FTA and a field device with appropriate MODEMs.

2.8.1 Operation

Each Serial Interface IOP connects to one or two FTA assemblies. There are 32 slots per Serial I/O card. 16 slots can be configured to FTA-1 and 16 slots can be configured to FTA-2. Plug-in modules adapt the FTAs for different communications protocols and applications. Qualified applications include—

Modbus Interface (Modicon Modbus-RTU protocol with extensions for real numbers and string support):
Modicon 984 programmable logic controller
Rosemount Micro-Motion Transmitter
Other Honeywell-approved Modbus compatible devices

Allen-Bradley Interface:
Allen-Bradley PLC-2, 3, and 5 programmable logic controllers

Other applications are being developed.

Each of the 32 Serial Interface slots can support up to 512 Flags, 16 Reals, 32 Integers, or 64 ASCII characters of contiguous data to/from a field device.

*Not all signals are supported

Each active SI slot must have a corresponding Array point. Data collected through the SI slot is available as local data for use by Digital Composite points, Device Control points, CL programs, etc. Array points are described in Section 10.

Flags and Numerics are fully supported as I/O connections; however, the Serial Interface is primarily intended as an input vehicle and it can efficiently import large quantities of data.

Additionally, the Serial Interface IOP firmware runs diagnostic routines. It monitors and reports any detected communications timeouts or errors between the IOP and FTAs.

2.8.2 Configuration

On the APMM Node Specific Configuration display, you must choose an APMM scan period (SCANPER). The APMM scan period specifies the scan period in seconds at which the APMM images the Serial Interface IOP database to the Array points. During Node Specific Configuration, you must also specify the IO Module Numbers that contain SI cards. Otherwise, no specific configuration is required for an SI slot, except what is entered for the associated Array point. Array points are described in Section 10 of this manual.

The maximum number of SI and associated Array points at the three configurable scan periods are—

Scan Period in Seconds	Maximum No. of Slots Scanned
1	80
.5	40
.25	20

Communication protocol in the FTA firmware is customized by Honeywell for specific uses. The SI IOP's Detail Status display shows the protocol, baud rate, and parity type configured for each FTA. If a fault exists, the configuration information is replaced with a failure message.

2.8.3 Checkpointing and Startup

Checkpointing—SI slot data is not checkpointed, but SI configuration is contained in the APMM Array point checkpoint. Configuration of related Array points is saved.

Startup—SI slot configuration data is automatically reloaded on every startup. When reloaded, the SI IOP Database Valid parameter is automatically set to Valid.

2.9 SERIAL DEVICE INTERFACE

The Serial Device Interface (different from the Serial Interface) uses a Serial Device Interface (SDI) module and a Companion Field Termination Assembly (FTA) designed to transfer serial data from and to specific external devices.

Communications with the device mimics a Smart Transmitter analog input point. The SDI module is configured as you would for a Smart Transmitter Interface Module (STIM).

Currently interfaces for the following devices have been developed:

- Manual/Auto Station
- Toledo Weigh Scale, Model 8142-2089
- Toledo Weigh Scale, Model 8142-2189
- UDC 6000 Process Controller

2.9.1 Serial Device Interface Description

Each SDI board can support two serial channels and each FTA can communicate in ANSI/EIA-232 (RS-232) or EIA-422/485. The EIA-232 interface connects to one serial device and the EIA-485 interface can connect to a multidrop network of up to eight devices.

Figure 2-13 illustrates the Serial Device Interface.

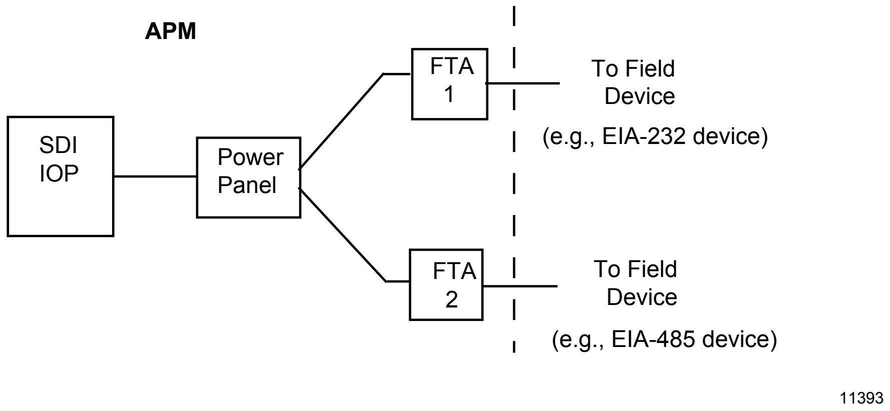


Figure 2-13 — The SDI Interface

2.9.1.1 Implementation and Control

Implementation —You can communicate directly with the SDI using CL or control algorithms. Other implementation methods are described in the *Process Manager Implementation Guidelines* or the *PM/UDC 6000 Integration manual* (see References).

Control—Operators typically control an SDI device from a custom schematic. You can build your own schematic or contract with Honeywell Engineering Services to build it.

2.9.2 Manual/Auto Station Interface

The Manual/Auto Station interfaces with the Process Manager through the Serial Device Interface FTA using EIA-485 protocol at 19.2 k baud. Up to four M/A Stations can be connected in multidrop to each of the two SDI FTAs. Each of eight slots in the SDI IOP can be configured for use with one Manual/Auto station. Slots 1–4 correspond to logical address for FTA one and slots 9–12 correspond to logical address 1–4 on FTA two (slot one = M/A Station 1 on FTA1, etc. and slot–9 = M/A Station 1 on FTA2 etc.).

Refer to the *Advanced Process Manager Implementation Guidelines* manual for additional information.

2.9.2.1 Communications

Invalid data—Both data received from the SDI IOP and data received from the Manual Auto Station are checked for reasonableness. Very small values may be rounded to .001. Very large (absolute) values are either rejected or stored as NaN.

Time out—A time out signal is sent from the SDI IOP to the FTAs every half second. Loss of this signal for 10 seconds or more halts all communications with the Manual/Auto Stations connected to that FTA.

Communication failure—When communications between the FTA and an M/A Station fails, the FTA makes two attempts to send the last message after the time out, then the point is put in bad input soft failure. As long as the slot remains active, there is an attempt to restore communications, but the M/A station is marked as not communicating.

Refer to the *Manual/Auto Station* manual for additional information on the Manual/Auto Station itself.

2.9.2.2 Implementation

The PV and OP process signals are directly connected to appropriate modules in the PM. Process and status signals are sent from the PM to the M/A station and operator changes to the Set Point at the M/A station are read into the PM as if those changes were at a Universal Station. Figure 2-14 illustrates a typical Manual Auto Station implementation.

Unless the PM Override flag is set, the PM responds to SP, OP, or mode change operations from the M/A Station's front panel. The PM Override function can inhibit operator functions attempted from the M/A station front panel that would affect SP, OP, or Mode.

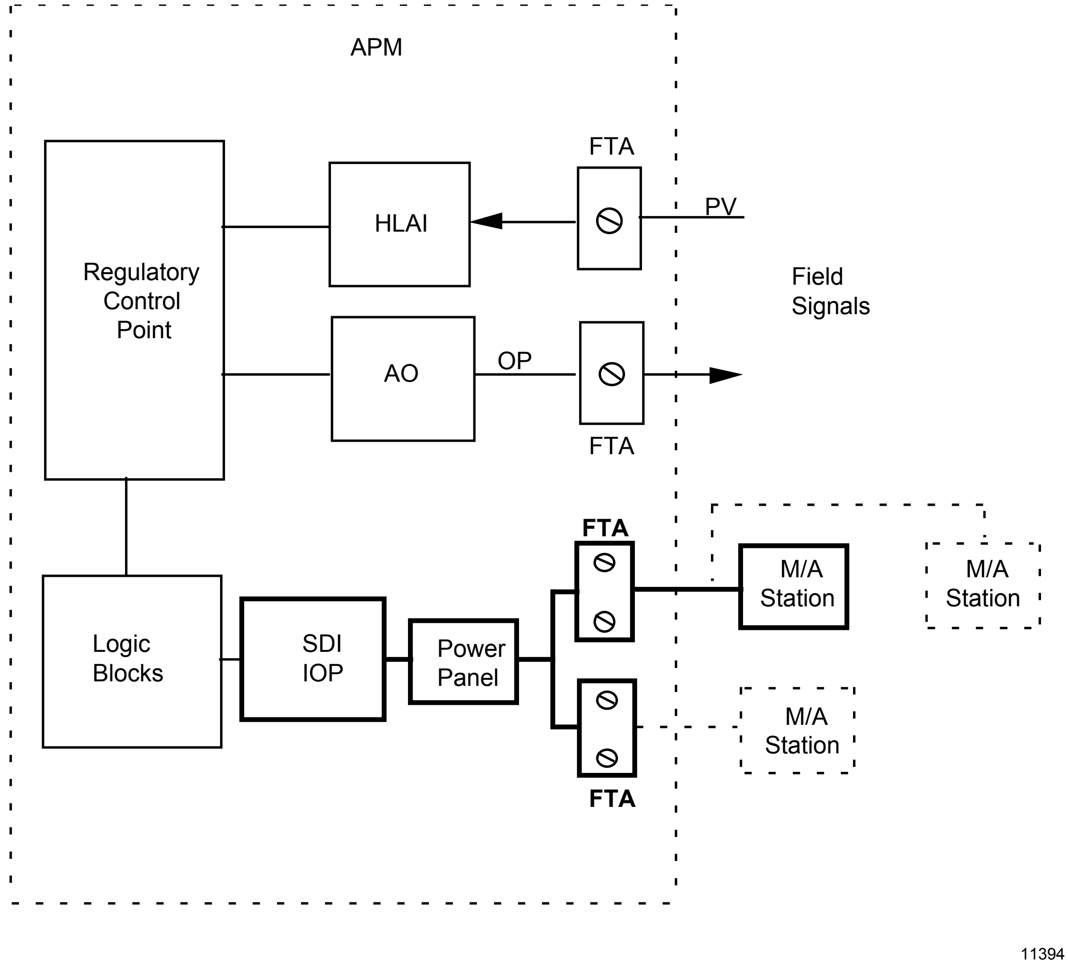


Figure 2-14 — M/A Station Implementation

Tables 2-6 through 2-8 show how to relate parameters seen at the Universal Station through the STI point and their meaning as Manual/Auto station parameters.

Table 2-6 — PV Related Parameters Sent to the Manual Auto Station

Parameter	Meaning as an STI Point Parameter	As an M/A Station Parameter	Comments
LRV	Low end of operating range for PVRAW	Low end of operating range for PVRAW	PV (in engineering units)
PVEUHI	PV high range in eng. units	PV high range in engineering units	Must be set to 100.0
PVEULO	PV low range in eng. units	PV low range in engineering units	Must be set to 0.0
PVEXEUHI	PV extended range high limit in engineering units	PV extended range high limit in eng. units	Must be set to 100.0
PVEXEULO	PV extended range low limit in engineering units	PV extended range low limit in engineering units	Must be set to 0.0

Table 2-7 — Control, Status and Other Parameters Sent to the Manual Auto Station

Parameter	Meaning as an STI Point Parameter	As an M/A Station Parameter

CJTACT	Internal cold junction reference active	OFF = Manual Mode
		ON = Auto Mode

DAMPING	Damping	M/A Station Alarms: 1 = Alarm 1 true 2 = Alarm 2 true 3 = Both Alarms true

PIUOTDCF	Open thermocouple detect enable	ON = In Override
		OFF = Not in Override

PTEXECST	Point execution state	Active = Point is active
		Inactive = Point is inactive

PVTV	PV target value	Setpoint in eng. units
STITAG	Transmitter tag name	M/A Station Tag Name
URL	Upper range limit of PV	OP in per cent
URV	Upper range value of PVRAW	Valve position in per cent

Table 2-8 — Parameters Sent from the Auto Manual Station

Parameter	Meaning as an STI Point Parameter	As an M/A Station Parameter
PVRAW	Raw PV input value	Setpoint (in eng. units) from M/A Station
SECVAR	Value of Secondary Variable	OP (in per cent) from the M/Auto Station)
SLOTSF	Slot soft failures	Soft failure due to bad input
STATE	Current point state	FTA Information
STI_EU	Smart Transmitter, engineering units INH2O MM_HG PSI KPA MPA	Change Requests: 0 = No change request 1 = SP change request 2 = OP change request 3 = Manual Mode request 4 = Auto Mode request
STISWVER	Smart Transmitter software revision level	FTA information: (sent from FTA) Slot 1 = M/A application Slot 2 = Software gen date Slot 3 = Software rev date

In addition to configuring the SDI IOP as an STI point, at least one, and preferably two logic points must be built to read data from each M/A station, perform necessary processing and then write the processed data, control, and status signals back to the M/A station.

The primary logic point handles the control functions and the following rules apply:

Parameter Change Requests (STI_EU)—an SP or OP change request takes on the enumeration value shown for approximately five seconds, after which it is reset to 0 (H2O) indicating no change request. Manual Mode change request and Auto Mode change requests are similar except their enumeration value is reset to 0 (H2O) as soon as the new mode is written to the M/A station (by means of CJTACT).

OP (SECVAR)—An OP request is read from the Manual/Auto Station, processed and written back to the M/A Station through the URL parameter.

SP (PVRAW)—An SP request from the Manual/Auto Station is processed and written back to the Manual/Auto Station using the PVTV parameter.

The second logic point is optional and is used to write back information such as Valve Position, Alarm state, and the Override flag using the parameters shown in the tables.

Refer to the *Advanced Process Manager Implementations Guidelines* (see References) for detailed information about implementing the above rules.

2.9.2.3 Operating Considerations

To modify any Regulatory Control Point parameter from the M/A station, the external Mode Switching Enable parameter ESWENBST must be set to ENABLE.

The Regulatory Control Point's Mode attribute parameter (MODATTR) must be set to PROG so that OP can be modified by a Logic point.

Only a one second scan period is supported for the M/Auto Station Logic points.

The PV High, PV Low, and PV Extended Range parameters must be configured with the values shown in Table 2-6 in order to transfer an SP change request to the Regulatory Control point's set point parameter SPP.

2.9.3 Toledo Weigh Scale Interface

Toledo Weigh Station models T8142-2089 and T8142-2189 interface with the Advanced Process Manager through the Serial Device Interface IOP using RS-232-D protocol at 9600 Baud. One Toledo Weigh Station can be connected to each of the two SDI FTAs. Slots one and nine in the SDI IOP can be configured as to communicate with the Toledo Weigh Station. The SDI IOP is implemented as a Smart Transmitter Interface Module (STIM).

2.9.3.1 Communications

Invalid data—The following parameters are checked for reasonableness: DAMPING, PVTV, URL, LRV, and DECONF.

Communication failure—When communications between an FTA and a Toledo Weigh Station fails, the FTA makes two additional attempts to send the last message, then the Toledo Weigh Station is marked as not communicating and PVRAW (the weight) is set to NaN.

To recover from most communication failures, switch the point execution status PTEXECST to inactive and back to active.

Scan Rate—Weight (PVRAW) and Flow Rate (SECVAR) are updated every 250 ms or faster.

Refer to the *Toledo Weigh Scale* manual for more information.

The following tables explain the parameters used to communicate with the Toledo Weigh Scale using an SDI interface configured as an STI module.

Table 2-9 — Parameters Sent from the Toledo Weigh Station

Parameter	Meaning as an STI Point Parameter	Meaning as a Weigh Scale Parameter
PVRAW	Raw PV input value	Weight (W)
SECVAR	Value of Secondary Variable	Flow Rate (R)
STI_EU	Smart Transmitter, type of eng. units used 0 = In_H2O 1 = MMHg 2 = PSI	Feed Status: Feed is Off (x) Feed is On (F) Feed is in alarm (A)
STISWVER	Smart Transmitter software revision level	FTA information: (sent from FTA) Slot 1 = Application Slot 2 = Software gen. date Slot 3 = Software rev. date

Table 2-10 — Parameters Sent to the Toledo Weigh Station

Parameter	Meaning as an STI Point Parameter	Meaning as a Weigh Scale Parameter

CJTACT	Internal cold junction reference active	Feed Control:
		On = Start Feeding/Feeding
		Off = Not Feeding

DAMPING	Damping	Weight Filter (F)*

PIUOTDCF	Open thermocouple detect enable	Mode:
		On = Ascending Mode (A)
		Off = Descending Mode (D)

PVTV	PV target value	Weight Setpoint* (S)
URL	Upper range limit of PV	Alarm Setpoint* (A)
LRV	Low end of operating range for PVRAW	Decimal Point Position (D) 1 = 1.0 2 = 0.1 3 = .01 4 = 0.001 5 = 0.0001
DECONF	Digitally Enhanced Configuration Mode	Parity Selection:** 0 = No Parity 1 = Odd Parity **2** = **Even Parity** 3 = No Parity 4 = No change
LRL	Lower Range Limit	Weight recorded when feed status switched from On to Off.

* Values are modified to 6 digits or less by moving the decimal point per LRV.
** Odd parity should not be used. Even parity is preferred.

CONTROL FUNCTIONS OVERVIEW
Section 3

This section provides an overview of the control functions available in the APM. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

3.1 CONTROL PERFORMANCE

The Advanced Process Manager is a high-performance device capable of an assured rate of 160 regulatory control loops per second. Users can customize their control configuration to meet the application requirements.

The parallel processing architecture of the Advanced Process Manager allows the control processing capability of the APM to be totally independent of other APM functions such as the number of I/O points built, data requests for APM data from the Network Interface Module and other UCN devices, and alarming functions; therefore, only two factors must be considered when configuring the control processing: **control slot type,** and the scheduled frequency of execution otherwise referred to as **scan rate**. These factors are described in the following paragraphs.

3.1.1 Control Slot Types

The following control slot types are resident in the Advanced Process Manager Module:

Slot Type	Max No. of Points (Slots)*	Applicable Parameter	Remarks
Digital Composite	512	NDCSLOT	Refer to Section 4
Device Control	160	NDEVSLOT	Refer to Section 11
Logic Slot	80	NLOGSLOT	Refer to Section 5
Process Module	160	NPMSLOT	Refer to Section 6
Regulatory PV	80	NPVSLOT	Refer to Section 7
Regulatory Control	160	NCTLSLOT	Refer to Section 8
Array	256	NARRSLOT	Refer to Section 10

*Subject to limitation of execution time.

The point mix is defined by specifying the number of slots of each type using the NDCSLOT, NLOGSLOT, NPMSLOT, NPVSLOT, NCTLSLOT, NDEVSLOT, and NARRSLOT parameters that are found on the UCN/PM Configuration Form. This configuration form allows the user to specify the particulars of the APM Box Data Point.

The slot numbers for each point type range from 1 to the user-specified maximum number listed in the above chart. For example, if NPVSLOT is set to 35, the RegPV points can be configured in any RegPV slot from slot 1 to slot 35. Similarly, if NLOGSLOT is set to 40, the logic points can be configured to run in any logic slot from slot 1 to slot 40. The point types and slot numbers are used for the initial configuration of a data point and for specifying the processing order.

3.1.2 Scan Frequency

Regulatory type (RegPV, RegCt1) and digital type (DigComp, DEVCTL, and Logic) slots can be configured for scanning at different rates as specified by the SCANRATE parameter and by the Fast Slots parameters. SCANRATE deals with the base scan rate for all points of a certain type. The Fast Slots parameters allow a specified number of regulatory or digital type slots to be processed at a quarter second rate without regard to the base scan rate for the rest of the group. These are APM Box Data Point parameters. Array points are not processed (require zero PUs) so they are not affected by the SCANRATE parameter.

3.1.2.1 Scan Rate

The scan rates indicate the number of times that all slots of that particular type (except Fast Slots) are scanned and processed each second. For example, a scheduled frequency of 1/4 sec for the regulatory slots indicates that all the regulatory slots in this APM will be scanned and processed four times each second. The scan rate has an impact on the number of slots that can be processed as described in paragraph 3.1.3.

The following chart contains the scheduled frequency of the respective entries for the SCANRATE parameter:

Entry for	**RegCtl & RegPV Slots**	**Logic, DigComp, DevCTL**	**ProcMod Slots**
SCANRATE Parameter	*Scheduled Freq.*	*Scheduled Freq.*	*Scheduled Freq.*
Reg1Log1	1 sec	1 sec	1 sec
Reg1Log2	1 sec	1/2 sec	1 sec
Reg1Log4	1 sec	1/4 sec	1 sec
Reg2Log2	1/2 sec	1/2 sec	1 sec
Reg2Log4	1/2 sec	1/4 sec	1 sec
Reg4Log4	1/4 sec	1/4 sec	1 sec

3.1.2.2 Fast Slots Parameters

Fast Slots parameters allow a specified number of the regulatory control, regulatory PV, digital composite, or logic slots to be processed at a quarter second rate without regard to the scan rate for the rest of the group. When fast slots are specified, they are the lower numbered slots for that type. For example, if there are 50 regulatory control slots and 15 are specified as Fast (NFASTCTL = 15), then slots 1-15 are scanned every quarter second, and slots 16-50 are scanned at the rate specified by the parameter SCANRATE.

The Fast Slots parameters are—

• NFASTCTL	Number of Fast Regulatory Control slots
• NFASTDC	Number of Fast Digital Composite slots
• NFASTPV	Number of Fast Regulatory PV slots
• NFASTLOG	Number of Fast Logic slots
• NFASTDEV	Number of Fast Device Control slots

If the SCANRATE parameter is configured for four times per second, then all of the points for those types are scanned at the quarter second rate and the number of fast slots is forced to the same as the total number of slots for that type. For example, if SCANRATE = REG1LOG4, then the number of Fast Logic slots parameter (NFASTLOG) are set equal to the number of Logic slots (NLGSLOT).

PERIOD—this parameter shows the processing period in seconds. Refer to the *Advanced Process Manager Parameter Reference Dictionary* for additional information.

3.1.2.3 Point Processing Executive

The point processing executive in the APMM runs at quarter-second (or 250 milliseconds) intervals or "cycles." (Each quarter second cycle is further divided into two sub cycles to provide a breakpoint for database synchronization.) Slot scheduling determines the assignment of slots to cycles and the ordering within a cycle. This operation is automatically performed at the time of transition from IDLE to RUN state, based on the following rules.

- There are four cycles per second. The slots executing at four times per second are processed on every cycle. The slots executing at two times per second are processed on every other cycle, and the slots executing once per second are processed on every fourth cycle.
- Each slot type has a different processing weight assigned to it. The slots processing slower than four times per second are distributed between the four cycles to equalize the processor loading (to whatever extent possible) of each cycle.

The processing order of the slots within a cycle is determined based on the point type as follows (from left to right):

RegPV → Fast RegPV → ProcMod → Logic → Fast Logic →
DevCtl → Fast DevCtl → DigComp → Fast DigComp →
RegCtl → Fast RegCtl →

Within each point type, the slots are processed in the descending order of the slot number (that is, from highest slot number, down to one).

3.1.3 Determining Processing and Memory Capacity

The processing power of the APM control processor is measured in terms of "Processing Units (PUs)." Each control processor has an assured rate of 160 PUs per second. Each slot type consumes a certain amount of PUs subject to the scheduled frequency.

The relationship between the scheduled frequencies and the Processing Units for the slot types are contained in the following chart. Please note that slots use more Processing Units at faster frequencies.

	PUs Per Second at Scheduled Frequencies		
Slot Type	**1/4 sec**	**1/2 sec**	**1 sec**
Digital Composite	0.4 PU	0.2 PU	0.1 PU
Device Control	4 PU	2 PU	1 PU
Logic	4 PU	2 PU	1 PU
Process Module	N/A	N/A	1 or 2 PUs
Regulatory PV	4 PU	2 PU	1 PU
Regulatory Control	4 PU	2PU	1 PU
Array	0 PU	0 PU	0 PU
String	0 PU	0 PU	0 PU
Timer	0 PU	0 PU	0 PU
Flag	0 PU	0 PU	0 PU
Numeric	0 PU	0 PU	0 PU

Process Module Points—When configuring each APM, you can choose how many processing units are allocated for its Process Module points.

- If parameter SEQPROC = **1_PU**, one PU is allocated per Process Module point per scan and up to 40 Process Module points can be processed each quarter-second cycle.

- If parameter SEQPROC = **2_PU**, two PUs are allocated per Process Module point per scan and up to 20 Process Module points can be processed each quarter-second cycle.

In batch applications where many small sequences are needed and you want all of them loaded at the same time, 1_PU is the appropriate choice for parameter SEQPROC.

Process module points in software releases R230 (and earlier) always used two processing units, therefore, parameter SEQPROC should be set to 2_PU if updating from R230 and earlier versions.

Memory Units—APMM capacity is measured in terms of Memory Units. For the point database and sequence programs, the maximum capacity is 15,000 Memory Units (MU) allocated as follows:

Slot Type	Memory Units Per Point	Slot Type	Memory Units Per Point
Digital Composite	5	Sequence Programs	1
Logic	15	Numerics	1/16*
Process Module	15	Flags	0
Regulatory PV	12	Device Control	30
Regulatory Control	13	Array	8
String	1/8*	Times	3/32*
		Timer	0

* 8 strings = 1 MU, 16 Numerics = 1 MU, 32 Times = 3 MU.

Any mix of the preceding slot types can be used, but the mix is subject to the following constraints:

1. Point mix used must not exceed 160 Processing Units or 15,000 Memory Units per APM.
2. Absolute number of each slot type per APM cannot exceed the maximum shown in the following chart (regardless of the available PUs):

Slot Type	Max. No.
Digital Composite	512
Device Control	160
Process Module	160
Regulatory PV	80
Regulatory Control	160
Logic	80
Array	256 (80 external max.)
Numeric	16,384
String	16,384
Time	4096

The following are two examples of possible configurations that show the PU calculations.

Example 1:

Slot Type	Slots	Freq	PUs/Slot	Total PUs/Slot Type
Regulatory Control	160	1 sec	1 PU	160 X1 = 160 PU
Digital Composite	- - - -	- - - -	- - - -	- - - - - - - - - -
Logic	- - - -	- - - -	- - - -	- - - - - - - - - -
Process Module	- - - -	- - - -	- - - -	- - - - - - - - - -
Regulatory PV	- - - -	- - - -	- - - -	- - - - - - - - - -

Total PUs for this APM = 160 PUs

Example 2:

Slot Type	Slots	Freq	PUs/Slot	Total PUs/Slot Type
Digital Composite	50	1/4 sec	0.4 PU	50 X 0.4 = 20 PUs
Logic	10	1/4 sec	4 PU	10 X 4 = 40 PUs
Process Module	10	1 sec	2 PU	10 X 2 = 20 PUs
Regulatory PV	10	1/2 sec	2 PU	10 X 2 = 20 PUs
Regulatory Control	30	1/2 sec	2 PU	30 X 2 = 60 PUs

Total PUs for this APM = 160 PUs

Invalid configurations are displayed as errors when the information is being loaded into the APM through the Universal Station.

3.1.3.1 Overrun Handling

The APM has the capability to handle most configurations of up to 160 PU without encountering an overrun. Overruns occur when a specific task cannot be completed in the allowed time. The following three kinds of overruns are possible in the APMM.

- Point Processing
- I/O Link Access
- UCN Access (Peer-to-Peer Communication)

Point Processing Overrun—A point processing overrun occurs when the slots that are scheduled to be processed during any subcycle cannot be finished within the allocated time of 125 msec. When this occurs, all the points are processed to completion (nothing is aborted), but the following subcycle is delayed until the next 125 msec subcycle. Also, the current hour point processing-overrun counter is incremented by 1.

The primary cause of point processing overruns are several long sequence programs that are processing on the same subcycle. If a point processing overrun occurs for four seconds in a row (on any subcycle within each second), a point processing-overload soft failure is generated. The soft failure is reset when an overrun does not occur for eight seconds.

For each subcycle, the current hour and previous hour point processing-overrun counters are maintained in the APMM and displayed on the UCN Detailed Status Displays to help track down the cause of the overrun.

I/O Link Access Overrun—This type of overrun occurs when parameter read or write access requests, from slots residing in the APMM to I/O Processors in the same APM, are not completed within a cycle time. This indicates that too many parameter access requests were attempted through the I/O link within the last cycle (250 msec). When this occurs, point processing is delayed by 125 msec and the I/O link overrun counter is incremented by 1. This continues until the requested data becomes available. I/O link accesses to some parameters are not counted as I/O link accesses, and accesses to them are considered as being "unlimited." These are the digital input point's PV (parameter PVFL) and BADPVFL, and the Digital Output point's SO, and INITREQ parameters.

The anticipated average number of I/O link accesses (from APMM points within the same APM) is 120 parameters for each 250 msec cycle. Access overruns can be expected when the number of I/O link accesses exceeds 200 parameters for each 250 msec cycle.
If an I/O link overrun occurs for four seconds in a row (on any cycle within each second), an I/O link access-overload soft failure is generated. The soft failure is reset when an overrun does not occur for eight seconds.

For each cycle, the current hour and previous hour I/O link access-overrun counters are maintained in the APMM and displayed on the UCN Detailed Status Displays to help track down the cause of the overrun.

UCN Access Overrun—This type of overrun occurs when parameter read or write access requests to other APMs (or other nodes) on the same UCN are not completed within the given time (0.75 seconds).

When this type of access overrun occurs, the UCN access-overrun counter is incremented by 1, but point processing is not delayed. The sequence programs that access data from other UCN devices remain suspended until a response becomes available (or a communication timeout occurs). For input connections at RegPV, RegCtl, and logic slots that require data from the UCN, the last fetched value is used until new data becomes available, or a time-out occurs.

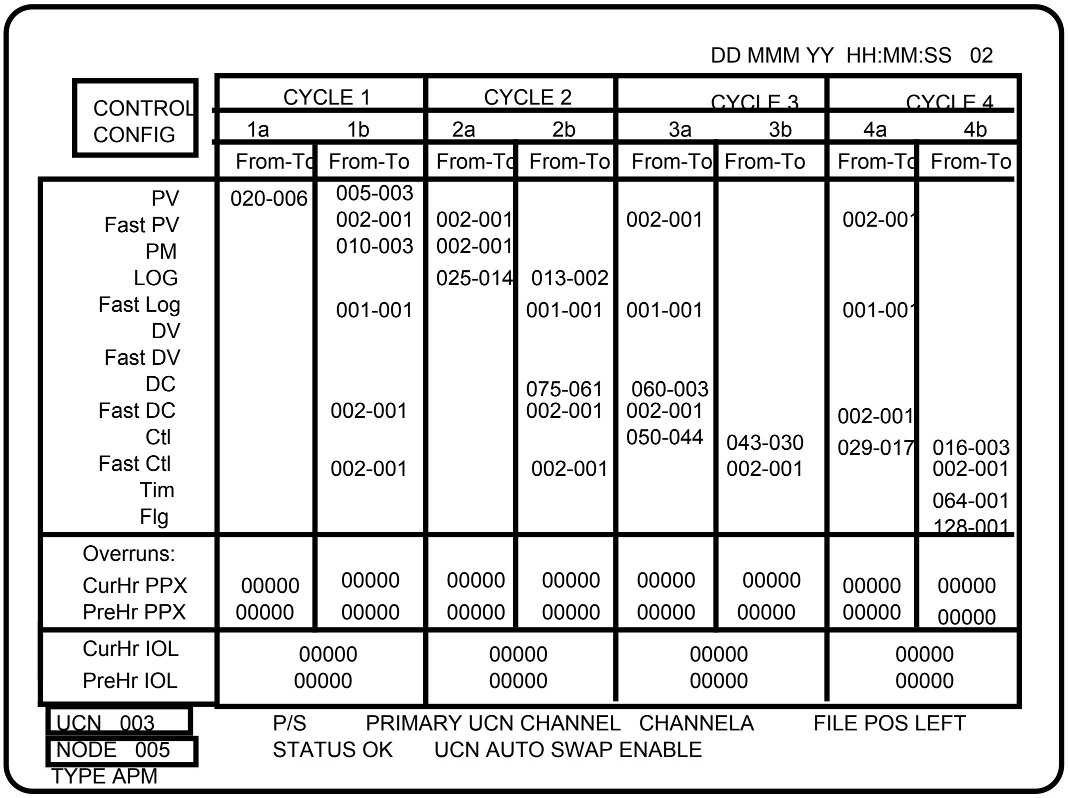
If the UCN access overrun occurs for 8 seconds in a row (on any cycle within each second), a UCN access-overload soft failure is generated. The soft failure is reset when an overrun does not occur for 16 seconds.

For each subcycle, the current hour and previous hour UCN access-overrun counters are maintained in the APMM and appear on the UCN Detailed Status Displays to help track down the cause of the overrun.

Peer-to-peer UCN Overruns and Soft Failures—Two consecutive peer-to-peer timeouts are required to count as a UCN overrun. More than one UCN overrun in 8 consecutive seconds causes a soft failure. The soft failure automatically clears when 16 consecutive seconds do not contain any UCN overruns.

Figure 3-1 illustrates the APM Scheduling Display—You may want to check this display for overruns, especially if your process module points contain a large number of statements per step (as long as all of the sequences in a subcycle average out to 20 statements per step, no overruns will occur).

This display can be accessed through the APM Status Display/Detail Display target; then under CONTROL CONFIGURATION, choose SCHEDULE INFO.



DD MMM YY HH:MM:SS 02

CONTROL CONFIG	CYCLE 1		CYCLE 2		CYCLE 3		CYCLE 4	
	1a	1b	2a	2b	3a	3b	4a	4b
	From-To	From-To	From-To	From-To	From-To	From-To	From-To	From-To
PV	020-006	005-003						
Fast PV		002-001	002-001		002-001		002-00	
PM		010-003	002-001					
LOG			025-014	013-002				
Fast Log		001-001		001-001	001-001		001-00	
DV								
Fast DV								
DC				075-061	060-003			
Fast DC		002-001		002-001	002-001		002-001	
Ctl					050-044	043-030	029-017	016-003
Fast Ctl		002-001		002-001		002-001		002-001
Tim								064-001
Flg								128-001
Overruns:								
CurHr PPX	00000	00000	00000	00000	00000	00000	00000	00000
PreHr PPX	00000	00000	00000	00000	00000	00000	00000	00000
CurHr IOL	00000		00000		00000		00000	
PreHr IOL	00000		00000		00000		00000	

UCN 003 P/S PRIMARY UCN CHANNEL CHANNELA FILE POS LEFT
NODE 005 STATUS OK UCN AUTO SWAP ENABLE
TYPE APM

Figure 3-1 — APM Scheduling Display

Along the left side, the type of point and execution order appears. The four 1/4 second cycles and their subcycles divide the display into columns. The numbers in each column represent the point slots. All of the points in a subcycle run consecutively in descending order (compare to the drawing in 3.1.2.3). Overruns (if any) for the point processing executive (PPX) and the I/O Link (IOL) appear across the bottom for the cycle or subcycle affected.

3.1.4 Performance Monitoring

Use the Toolkit displays to check various performance statistics. These displays show the number of nodes requesting and responding, and the percent of time the APM Communications and Controller CPUs are not busy. They also show the maximum, minimum, and average request/response times for transactions between peer nodes and other statistics. Section 10 of the *Engineer's Reference Manual* explains how to use the Toolkit displays.

A peer-to-peer communication efficiency statistic is displayed on the Control Configuration page of the APM Detail Display. Also refer to the *APM Implementation Guidelines* for other performance information.

3.2 DATABASE CONSIDERATIONS

The point mix and the scan rate are configured as a part of the APM Box Data Point. On the Universal Station, this is the Node Specific Configuration display. When the APM Box Data Point information is loaded into the APMM, the point mix is validated against the absolute maximum allowed for each point type and the available memory and the scan rate is validated against the available processing resources. The APMM box state must be IDLE to make any changes to the point mix and scan rate configuration.

If changes are not required in the point mix, the APM Box Data Point can be loaded while in the RUN state. For example, a new I/O Processor can be added without interrupting control by simply loading the APM box data configuration (with the new I/O Processor added) while in the RUN state.

CAUTION

When the point mix or the scan rate is changed, the database of all the slots is defaulted to a null configuration and the database requires reconfiguration.

Point building follows box Configuration. Each point is configured to run in a slot inside the APMM. The first time through a power-up cycle (APMM box state transition from ALIVE to IDLE), the database of each slot (included in the default point mix) is defaulted to a null configuration. The user can build or load a point configuration into each slot. When a point is deleted, the database of the slot containing the point is defaulted to the null configuration.

3.3 POINT EXECUTION STATE

Slot processing in the APMM is performed only when the APMM box state is RUN. Only those slots whose point execution state (PTEXECST) is Active are processed. The point's execution state must be Inactive in order to modify any of the configuration parameters (for example, engineering unit range, PV or control algorithm ID, input/output connections, etc.). When a point is made Inactive, all the calculated variables (like PV, CV, alarms, etc.) are set to their default values ("NaN" for real numbers and OFF for Logical). The point execution state cannot be changed back to Active until all the configuration parameters are specified.

3.3.1 Run-Idle-Run Transition

When the APMM is switched from RUN to IDLE and back to RUN, the affect on data points is shown by the following:

Table 3-1 — Effect of Run-Idle-Run Transition

	Effect of RUN to IDLE to RUN Transition
Logic Blocks	Execution halts, then resumes once box is running again.
CL Programs	Execution stops. Program needs to be restarted once PM is running again. Refer to parameter RSTROPT.
Timers	The timer stops and needs to be restarted once the box is running. Time accumulation starts from where it left off unless reset deliberately.
Other Point Types	Processing stops and outputs are held. Processing resumes when the APM is running again.

3.3.2 Warm/Cold Restart

The APM is started through the APM Status display. Warm/Cold startup refers to the behavior of algorithms under initialization. When going from the idle to run state, the operator can choose between warm and cold startup (for each APM). After a checkpoint restore, either a warm or cold startup is possible.

Cold startup — this usually requires operator intervention, that is, the operator must change some point states from manual to their normal operating mode. On a cold startup, all Regulatory Control points that output to the field go to manual mode. For Regulatory Control points that drive other Regulatory Control points (even in another APM), a cold start does not cause a change from the previous mode.

Warm startup — If you choose warm startup, Regulatory Control points remain in the previous mode and are back-initialized for a bumpless transfer to automatic control with their previous output value.

Powerup Restart — If RAM memory was not lost since the previous operation, a powerup restart is possible without reloading the program. In this case, the APM will return to its previous Idle/Run state. If the state is Run, all points behave as if a cold restart was performed.

Some points behave differently during startup regardless of the startup mode:

Regulatory PV Totalizer points — you can configure the point to continue or stop accumulation on restart (see parameter PVEQN).

Process Module point — you can configure sequence execution to restart from the beginning or remain off and be restarted by the operator (see parameter RSTROPT).

Regulatory Control Ramp Soak — Point is always set to Manual mode.

Table 3-2 summarizes the important status conditions for Warm/Cold restart functions.

Table 3-2 — Warm Cold Restart

Point Type	Parameter	Status After Power Off/On	Status After Cold Start	Status After Warm Start
Regulatory Control (except Ramp/Soak) Outputs to IOP	MODE	Manual	Manual	Last Mode. Back-initializes; goes on control
Regulatory Control (except Ramp/Soak) Outputs to another RegCtl point	MODE	Manual	Last Mode[1]	Last Mode. Back-initializes; goes on control
Regulatory Control **Ramp Soak**	MODE	Manual	Manual	Manual
Regulatory PV **Totalizer**[2] PVEQN = A, B, C	PV PVSTS STATE	Cont. Totalizing Uncertn Running	NaN Bad Stopped	Cont. Totalizing Uncertn Running
Regulatory PV **Totalizer** PVEQN = D, E, F	PVSTS STATE	Bad Stopped	Bad Stopped	Bad Stopped
Process Module RSTROPT = Restart	SEQEXEC PROCMOD	Run Norm	Run Norm	Run Norm
Process Module RSTROPT = Off	SEQEXEC PROCMOD	Loaded Off	Loaded Off	Loaded Off
Timer	PV STATE	Last Good Value[3] Stopped	Last Good Value[3] Stopped	Last Good Value[3] Stopped

[1] Control is halted by the final point in the cascade which always outputs to an IOP.

[2] Count may be inaccurate if rollover occurs in IOP.

[3] The last good value can come from PM RAM Memory if battery power was continuously available; otherwise, from the last checkpoint stored on the History Module.

3.4 PEER-TO-PEER COMMUNICATION

Peer-to-peer communication allows information to be passed between devices on the UCN and can be initiated from the Advanced Process Manager, the High-Performance Process Manager, the Process Manager, the Logic Manager, or the Safety Manager.

3.4.1 Implementation

The APM provides the following two convenient mechanisms for implementing peer-to-peer communications functions:

- Input/Output Connection
- CL Read/Write Statement

Both communication mechanisms offer location-independent access to data from another UCN device using the "Tagname.Parameter" notation.

Refer to Table 3-3 for a summary of peer-to-peer communications.

Characteristics and configuration limits for input/output connection for each access are described below.

- **Input Connections**—The RegPV, RegCtl, DevCtl, and Logic slots can use standard input connections to obtain data from other UCN devices. When such a connection is configured, it is scanned at the rate of twice per second, regardless of the scan rate of the slot in which it is configured. The data obtained is saved internally and used (more than once if the slot is running faster than twice a second) until the next time that it needs data. Peer-to-peer data is requested .75 seconds before it is needed. The data obtained from another UCN device can therefore be up to .75 second old by the time it is used. A maximum of 50 input connections can be configured for each APMM.

  We recommend that input connections be used for continuous peer access that requires a 1/2 second update (see Table 3-3).

- **Output Connections**—The RegCtl, DevCtl, and Logic slots can use standard output connections to provide outputs to other UCN devices. Output connections are processed when the slot is processed, thus outputs are updated four times per second, two times per second, or once per second depending on the point processing rate. The number of output connections per APMM is not limited.

  We recommend that output connections be used for regulatory-control cascades between UCN devices (see Table 3-3).

Table 3-3 — Peer-to-Peer Summary

Peer-to-Peer Mechanism	Configuration Limits (Per APM)	Handling of Failure/Recovery of Peer Node	Recommended Usage
Input Connections	50 values per 1/2 second	*Analog* —On failure, bad value (NaN) is substituted. *Digital* - Addressed by Logic Slot. On failure, ON, OFF, or last value is substituted based on user configuration. On recovery from failure, accessed values are then used.	Use for continuous (1/2 second) update of data required for RegPV, RegCtl, DevCTL, or Logic Slots.
Output Connections	Not limited	*Regulatory control cascade* — primary initializes. *Logic output* —store does not occur. On recovery from failure, regulatory control clears initialization and resumes control. Subsequent output changes are stored on transition.	Use for regulatory control cascades between nodes.
CL Read Statement	16 values per Process Module Data Point per second	If requested read transaction not completed, CL program branches to specified error location. Program fails when error option not used. On failure recovery, program behavior is based on user code, or requires operator interaction if failed.	Use for on-demand data required by CL programs. Use for continuous (1 second) update of data required for RegPV, RegCtl, DevCTL, or Logic Slots.
CL Write Statement	16 values per Process Module Data Point per second	Same as above.	Same as above Note: Use of Read or Write statements is based on user preference.

Characteristics and configuration limits for CL/APM read/write peer access are as follows:

- **CL/APM Program**—A user-written CL/APM sequence program executing in the APMM can use explicit read/write statements to read parameter values from other UCN devices and write parameter values to other UCN devices. Any time a read/write statement is executed, the sequence program is suspended until the following cycle when the response becomes available (normally one second).

  We recommend that the CL read/write technique be used for on-demand peer access, and for continuous peer access that requires a 1-second update.

3.4.2 Restrictions

Certain restrictions exist on the type of data that can be accessed through peer-to-peer communication. These restrictions vary with the point type. Table 3-4 shows the type of data that can be accessed for each point type or connection type.

Table 3-4 — Peer-to-Peer Communications

Connection Type	Allowed Data
RegCtl.CISRC	No restrictions
RegCtl.CODSTN	RegCtl.SP only
RegPV.PISRC	No restrictions
Logic.LISRC	No restrictions
Logic.LODSTN	No restrictions
DevCTL.LISRC	No restrictions
DevCTL.LODSTN	No restrictions
DigComp	No Peer-to-peer connections allowed
DevCtl	No Peer-to-peer connections allowed
ProcModl Read Statement	No restrictions
ProcModl Write Statement	No restrictions

In addition to the configuration constraints described above, there are performance-related limits on parameter access that affect the overall peer-to-peer throughput capability. Actual performance for each APM varies depending on specific system configuration; however, any APM should be capable of handling a total load of up to 400 parameters per second.

In addition to the above parameter throughput-related constraint, the number of devices concurrently communicating to a single Advanced Process Manager is another factor that influences performance.

The interaction of the above constraints must be considered primarily in applications in which an extreme communication load is continuous. In this type of application, the following configuration recommendations maximize peer-to-peer throughput capability:

- Adopt a single approach for all peer-peer communications; use either input/output connections or CL read/write statements. If a 1-second update is adequate and cascade interconnection is not required, the CL read/write approach offers increased capability (quantity/throughput) for peer data access.

- For CL read/write approach, consistently use same method (either CL reads or CL writes).

- For CL read/write approach, the Process Module Data Points that are continually affecting peer-to-peer communication should be grouped on the same 1/4-second cycle.

3.5 HARDWARE REFERENCE ADDRESSING

The APM allows the referencing of parameters of I/O points (AO,DI, DO) on a hardware basis within the same APM using the standard input/output hardware connection reference syntax (user-written sequence programs can reference DI points). This type of addressing does not affect the point count in the NIM and these "untagged" references can be quickly built, but there are several disadvantages. Untagged points are not easily visible in the system so you must use the Command Processor Find Names function to see the connections. Note that operating displays cannot include untagged points.

The use of untagged references is not recommended, however, there might be a reason to use them or some may be present in an existing system. If you use them keep good records.

The hardware reference address point format is

!MTmmSss.Parameter

where: MT is the IOP type such as AO, DI, DO, or PI,
mm is the number of the IOP card (1-40) in the card file,
ss is the input or output slot number on the specified IOP card.
Parameter is the parameter whose value is to be written to by an output connection, or a parameter whose value is to be read by an input connection.

The following are examples of hardware reference addresses:*

```
AO Processor Output ==> !AO12S03.OP
                        = parameter OP
                          of Slot #3
                          of AO Processor #12

DI Processor PV ==> !DI05S07.PVFL
                        = parameter PVFL
                          of Slot #7
                          of DI Processor #5

DO Processor (Status or Latched) Output ==> !DO15S12.SO
                        = parameter SO
                          of Slot #12
                          of DO Processor #15

DO Processor ON Pulse Command ==> !DO15S12.ONPULSE
                        = parameter ONPULSE
                          of Slot #12
                          of DO Processor #15

DO Processor OFF Pulse Command ==> !DO15S12.OFFPULSE
                        = parameter OFFPULSE
                          of Slot #12
                          of DO Processor #15
```

Standard status displays are available to show which points are associated with each hardware module.

*The AI address !AImmSss.Parameter is not supported because the analog input point does not have a useable default database.

3.5.1 Find Names Function

You can use the Find Names function to provide the node number, module type, point type, and slot number for all connections. This function is especially helpful to find the hardware reference points. Refer to the *Command Processor* manual for information on the Find Names utility.

3.6 REDUNDANCY

The APM can contain optional redundant APMM modules and certain types of optional redundant IOP modules (currently HLAI, STIM, and AO modules). During APM Node Specific configuration, the user specifies which modules are redundant and, for IOPs, the file/card locations of the redundant partners. Refer to Section 2 in the *Advanced Process Manager Implementation Guidelines* manual for more information.

If a redundant APMM module or synchronized redundant IOP module fails, switchover is automatic and transparent to the user (input and set output operations are unaffected). Data acquisition, alarming, and control strategies are automatically managed by the system.

The principal parameters that pertain to APM redundancy are—

ACTPRIM
FTACONN
IOMCARDA/B
IOMFILEA/B
IOREDOPT
PKGOPT
POSITION
SYNCHSTS
WITHBIAS

Refer to these parameters in the *Advanced Process Manager Parameter Reference Dictionary* for additional information.

3.7 POINT RESERVATION

To support batch applications, you need to be able to reserve certain points that represent shared devices for particular batches. The APM allows CL programs to reserve Regulatory Control, Regulatory PV, Logic, Digital Composite, Device Control, Process Module, and Array points for exclusive use.

Point reservation is accomplished using a 16-character string parameter (USERID). This string has special logic so that two programs cannot reserve the same point at the same time. This string can always be stored by the operator, but a program can only store a nonblank string when the USERID itself is currently blank. This should eliminate race conditions.

To keep programs from failing, storing an invalid string does not generate an error; it will just not do anything. The program should read the string back after the store to verify that it was accepted. To handle retries, a given value of the string can be stored any number of times. All blanks can be stored at any time to release the point. Programs must be careful not to release points that other programs or the operator have reserved. A special string starting with three or more dashes may be used to indicate that a point cannot be reserved and only the operator can clear this string.

3.8 STATUS MESSAGES

The Status Message function is available for RegCtl, RegPV, Logic, DigComp, DevCtl, ProcMod, and Array points. One table of messages is configurable for each NIM.

During UCN Node Configuration for the NIM, you can enter up to 15 Message Text Items, (MSGTXT(n)). The entry is an 8-character ASCII string. Message text item 0 defaults to NONE and is not configurable. The actual number of Message Text items that you can enter is determined by parameter NMSGTXT, the number of Message Text items.

On the point’s Detail display, the operator can select STSMSG and choose any message from the table of Message Text items that appears. That Message Text item then replaces the current contents of STSMSG and, if nothing of higher priority is currently displayed, appears as a comment at the lower left of the display. The Status Message also appears in place of the “Red Tag” message when Red Tag is on.

If a point is not in Red Tag, a CL program can store a message (in STSMSG). When Red Tag is on, only the engineer or supervisor can store the message.

DIGITAL COMPOSITE POINT
Section 4

This section describes the functions available in the digital composite point. The description of the output portion of the point is provided first, followed by the description of the input portion. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

4.1 DESCRIPTION

The digital composite point is a multi-input/multi-output point that provides an interface to discrete devices such as motors, pumps, solenoid valves, and motor-operated valves. This point provides built-in structures for handling interlocks, and supports display of the interlock conditions in group, detail, and graphic displays. In addition, the digital composite point can be used with logic slots and other digital points to implement complex interlock schemes.

Some of the features of the digital composite point are as follows:

- Input and output states of the data point are totally independent of each other and can be configured as required by the user's application. The inputs and outputs of the point can be linked to I/O points or to Boolean flags within the APM.
- Supports devices with two or three commandable states (motor-operated valves, reversing motors, etc.). Specific states can be defined as being momentary.
- Provides structured approach to handling interlock functions and includes indication to the operator of existence of interlock condition.

4.2 POINT STATES

4.2.1 Normal States

A digital composite point can have two or three normal states that allow the data point to interface with devices that have two or three operational states. The states of a digital composite point are as follows:

- State 1—this is the first active state.
- State 0—this is the inactive (middle) state. This is considered a safe state. If for any reason a valid output cannot be determined, the OP is forced to this state. See subsection 4.9 which discusses initialization manual.
- State 2—this is the second active state. (State 2 is applicable only when the entry for the NOSTATES parameter is 3, signifying that the data point has three states.)

These states can best be described by showing how the digital composite data point is depicted in a Universal Station group display, as shown in Figure 4-1. On the display, each state is represented by a separate box. The state 0 box is the middle box, the state 1 box is the upper box, and state 2 box is the lower box. (The state 2 box does not appear if the point has been configured for two states where parameter NOSTATES = 2). The state descriptor for each box is an 8-character descriptor and appears at the top of the box. In Figure 4-1, the descriptors are FORWARD (for state 1), STOP (for state 0), and REVERSE (for state 2), which are specified by parameters STATETXT(0)-STATETXT(2).

The lower portion of each box is split, with the left box being the indicator for the input PV state, and the right box being the commanded-output state indicator. The condition of the PV input that causes the left box to light is determined by parameters D2D1_00 - D2D1_11. The condition that causes the right box to light is determined by OP being commanded to that state.

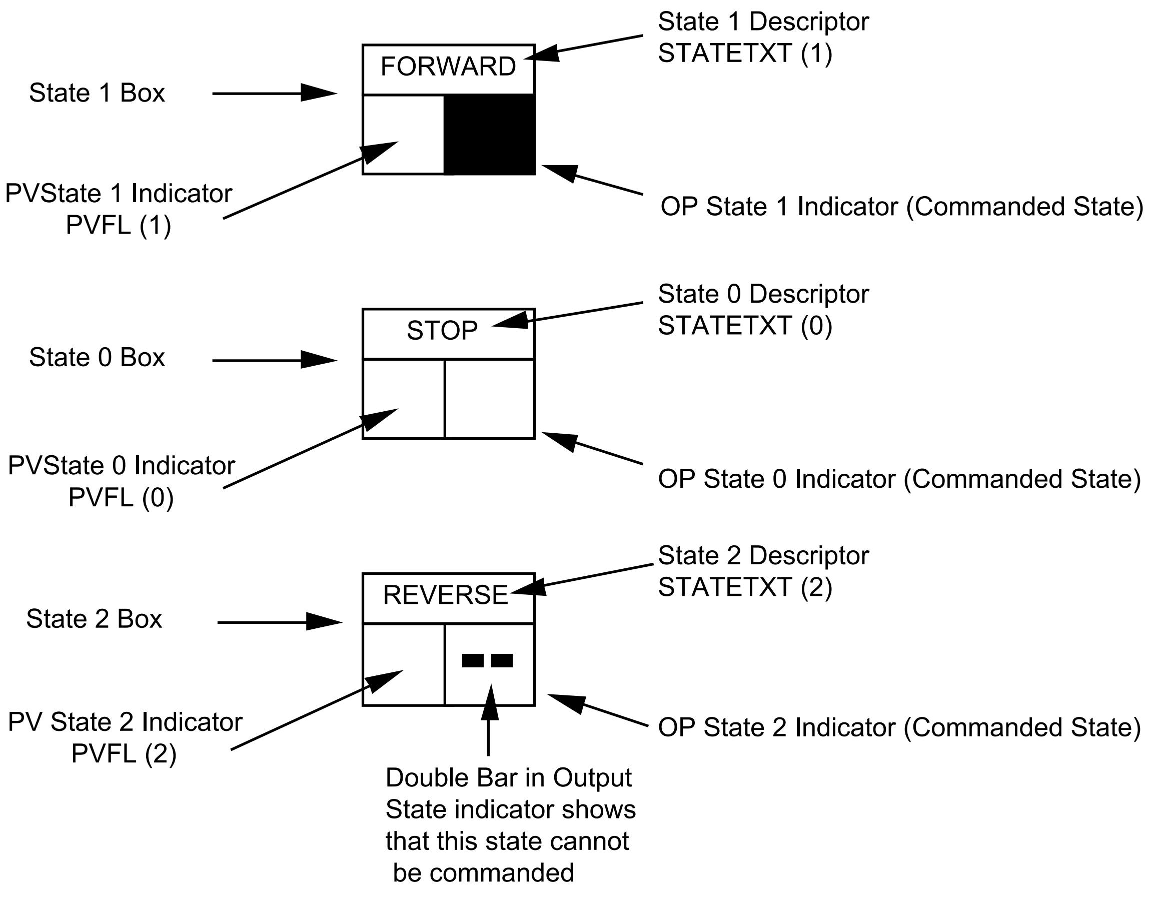


Figure 4-1 — Digital Composite Point Display

As an example, assume that the digital composite point shown in Figure 4-1 is connected through digital output points and digital input points to a motor control circuit and that the current state of the digital composite point is state 0 (STOP).

- In state 0, the PV indicator of the STOP box is lit, indicating that the motor is stopped because the proper PV input has been received from the field. All other indicators in the display are in the extinguished condition.

- When the operator touches the FORWARD box to place the motor in the forward running direction, the digital composite point output goes to State 1. Outputs from the digital composite point are provided as determined by the ST1_OPn parameter (where n=1..3). When the proper PV inputs are received from the field, the FORWARD box is lighted. This indicates to the operator that the motor is running. Also, the current state of the PV can be read by a logic slot or by a user program by accessing parameter PVFL(n) where n is 0, 1, or 2. If the PV is in State 0, PVFL(0) is on; otherwise, PVFL(0) is off. PVFL(1) and PV(FL2) operate in the same manner for States 1 and 2, respectively.

- Similarly, when the operator presses the STOP box, the operator is notified that the motor is stopped by the lighted PV indicator in the STOP box. The operator can then select the REVERSE box to start the motor running in the reverse direction.

4.2.2 Momentary States

The states of a digital composite point can be configured as being momentary by using the MOMSTATE parameter. The commanded states can be configured as momentary where the state acts like a doorbell (state remains active as long as a switch is pressed). The momentary states (MOMSTATE) parameter specifies which of the states are momentary as follows:

Entry	Description
Mom_1	State 1 is momentary. When released from this state, it jumps back to State 0.
Mom_0	State 0 is momentary. When released from this state, it jumps back to State 1. Mom_0 can be selected only if the NOSTATES parameter = 2.
Mom_2	State 2 is momentary. When released from this state, it jumps back to State 0. Mom_2 can be selected only if NOSTATES parameter = 3.
Mom_1_2	States 1 and 2 are momentary. When released from any one of these states, it back jumps to State 0. Mom_1_2 can be selected only if NOSTATES parameter = 3.

A state cannot be set to momentary if the MOMSTATE parameter = NONE. MOMSTATE is mutually exclusive with the Seal-In option or State Change option (STCHGOPT).

4.2.3 Moving/Bad States

The digital composite point has two standard states that represent conditions when the current state of the device is "bad" (indeterminate) or the current state is "moving" (from one state to another). The bad state can result when the PV input signals from the process are in an inconsistent state (e.g., for a valve, the limit switches indicating open and closed are both simultaneously on). The moving state is encountered when the device is in transition from one state to another (e.g., a slow moving valve is changing from the open state to the closed state). The moving/bad state descriptor is located below the State 2 box on the display. In the example described above, the descriptor assigned for the BADPVTXT parameter would be the word FAULTY and the descriptor assigned for the MOVPVTXT parameter would be the word MOVING. These descriptors are configured once for each APM box data point and then are used for all digital composite points in the same APM.

If the standard "bad/moving" default text is not desired, you can select the PV Text Option during point configuration, and enter two eight-character strings of your choice. While the words "bad/moving" are defined on a PM/APM box basis, your entries override them on a per point basis. Up to 15 unique bad/moving text descriptors can be defined on a UCN network basis.

4.3 MODE ATTRIBUTES

The digital composite point supports only the manual (MAN) mode with mode attributes (specified by parameter NMODATTR) of Operator and Program. If the mode attribute is Operator, only the Universal Station operator can provide the commanded output state, as shown in Figure 4-2. On the other hand, if the mode attribute is Program, only user programs (continuous or discontinuous) can provide the commanded output state. The operator mode attribute flag (OPRATRFL) is also provided for potential use by the interlocking logic.

The MODEPERM parameter is provided to optionally prevent the operator (as opposed to the supervisor or engineer) at the Universal Station from changing the mode attribute.

4.4 INTERLOCKS

Two kinds of interlocks, permissives and overrides, are provided in the output portion of a digital composite point. The states of the interlocks are typically controlled by output connections from logic slots that can write to the P0-P2 permissive interlock parameters, and I0-I2 override interlock parameters. In addition, a safety override interlock (SI0) that cannot be bypassed is provided.

4.4.1 Permissive Interlocks

The permissive interlock parameters P0-P2 shown in Figure 4-2 are controlled by logic slot outputs, and the permissive interlocks determine whether the operator and user programs are allowed to change the output of a digital composite point to a specific state. A permissive interlock is provided for each of the three states. The permissive interlocks themselves never cause the output to change.

For the commanded output to be changed to the desired state, the corresponding permissive interlock parameter must be set to ON. As an example, if a logic-slot output has set parameter P1 to OFF, the commanded output of the digital composite point cannot be set to State 1. The permissive interlock parameters P0, P1, and P2 are all defaulted to ON, thereby allowing permission for all the states. They must be individually set to OFF by the logic slots to prevent access to the corresponding commanded output state.

4.4.2 Override Interlocks

The override interlock parameters I0-I2 shown in Figure 4-2 are also controlled by the logic slot outputs and can force the commanded output to a specific state regardless of the condition of the permissive interlocks. The operator and user programs cannot change the output state when any override interlock is in the ON condition. An override interlock is provided for each of the three output states.

When parameter I0 is set to ON by a logic slot output, the commanded output state is forced to State 0 (regardless of the condition of parameters I1 and I2). When I0 is OFF and I1 is ON, the commanded output is forced to State 1 (regardless of parameter I2). Finally, when I0 and I1 are OFF and I2 is ON, the commanded output is forced to State 2.

Override interlock parameters I0-I2 are all defaulted to OFF, disabling all the override interlocks. They must be set to ON to force the output to go to any specific state. If the override interlock forces the output to go to a momentary state, it stays in that state as long as the interlock remains ON and then switches back to the original state when the override interlock is reset to Off.

When all the override interlocks are OFF, the last value of the commanded output is maintained (except for momentary state) until changed by the operator, a user program, or another override interlock. The interlock logic restores the output value to the output devices (Digital Output IOPs) whenever the interlock value changes, even if the state asserted by the interlock is already in effect.

The Override Alarm Option (OVRDALOP) allows an Optional alarm to be generated when an override occurs. You have the following configuration choices for the Override Alarm Option:

1. no alarm
2. return to normal when the override is cleared (auto return)
3. operator confirmation required after interlock is cleared (the operator must confirm the alarm before it is possible to command a new state).

The occurrence of a new override takes precedence over any previously active override (including the Safety Override Interlock, SIO) if that condition has returned to inactive. Therefore, to the operator, a lower priority confirmation appears to overwrite a higher priority override if that higher priority override has reset. This is sometimes called an **auto-confirm** of the previous override.

If a higher priority override has been reset, the descriptions and confirm prompt (if Cnfm_Rqd was configured) remain visible until the operator confirms the override from the Detail Display. The confirm prompt and alarm are then reset. If any other override occurs after a previous condition has reset, regardless of the previous priority, the new description and confirm prompt appear. This also applies to the Real Time Journal Display. If the confirmation required option was not selected for a particular class of overrides, the description will still override the previous one and the previous confirm prompt goes away, if the previous override condition has reset.

NOTE

For those upgrading from R400 to R510 and later systems, note the differences described in this section and make sure your strategy still works as intended.

In R410 and R510 or later systems, a lower priority interlock (such as I1) can become active if a higher priority interlock (such as I0) is no longer true but has not yet been confirmed. In R400 systems, Confirmation has to be done before the lower priority interlock is activated.

When the Digital Composite point is built, you can specify the tag name of the Logic point that is driving the interlocks as parameter LOGICSRC. The contents of this parameter then appear on the Group Display when the Digital Composite point is selected. After identifying the Logic point, an operator can call up the Detail Display for that point to find out about the interlocks.

You can predefine an eight character alarm descriptor for each override interlock, I0-I2. When the override interlock is activated, that string is displayed on the detail display. CL programs can write into these descriptors (InDESC) before asserting the interlock. The detail display text then explains why the program forced the interlock (e.g., TOO HOT in the Override Alarm Descriptor, OVRDDESC).

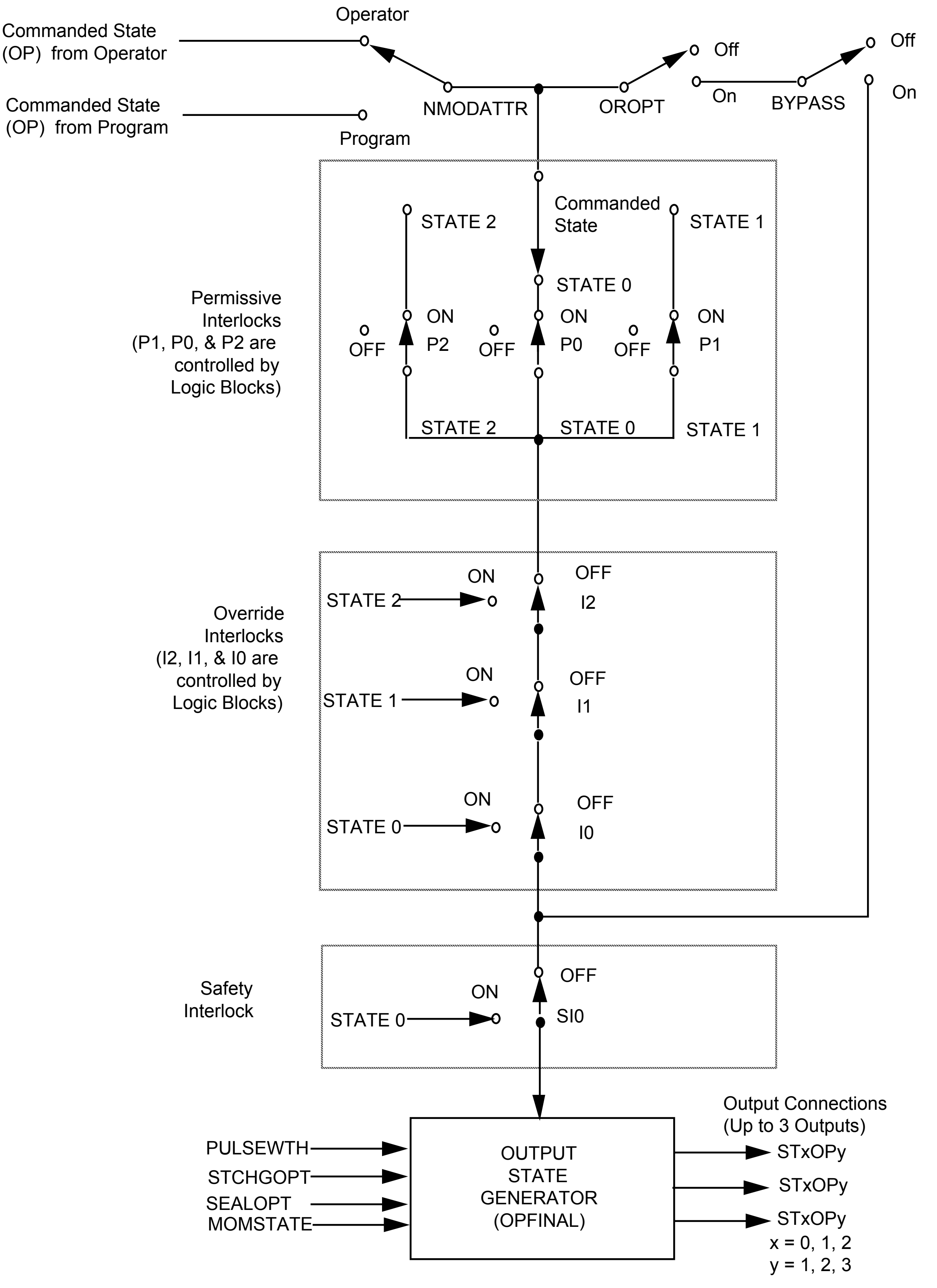


Figure 4-2 — Digital Output, Functional Diagram

2100

4.4.3 Configurable Interlock Bypass

The operator can have the ability to bypass the permissive and override interlocks for a digital composite point. To do so, the override parameter OROPT must be set to ON. The operator can then set (or reset) the parameter BYPASS from the Universal Station. Note that while BYPASS is ON, the point execution state parameter can't be changed. When the BYPASS parameter is reset, existing override interlocks (if any) take effect immediately. The group display shows whether interlocks are being bypassed. The organizational summary display lists all the points for which BYPASS = ON.

4.4.4 Safety Interlocks for State 0 (SI0)

The safety override interlock parameter SI0 behaves like the I0 interlock, but it cannot be bypassed even when BYPASS is configured and turned on (see Figure 4-2).

If a device has tripped as a result of a state 0 interlock, an interlock-trip alarm (SI0ALOPT) can result (if configured). The interlock-trip alarm has three options: None (no alarming), Auto Return or Alarm with Confirm Required. The alarms are described as follows:

- Auto Return— if a state 0 interlock causes the OP state to change, an interlock-trip alarm is generated and then automatically terminated when the interlock, that caused the alarm returns to OFF.

- Alarm with Confirm Required— if a state 0 interlock causes the OP state to change, an interlock-trip alarm is generated. The interlock that caused the alarm must return to OFF and the operator must reset the trip-confirm required flag OVRDCONF. The trip-confirm required flag can also be reset by REDTAG, BYPASS, or LOCALMAN, but the motor cannot be restarted by the operator, interlock, or other system functions until the interlock trip alarm is terminated.

- If SIO resets, a lower priority interlock can be asserted and confirm automatically as described in subsection 4.4.2.

You can predefine an 8-character alarm descriptor for SIO. When the safety override interlock is activated, that string is displayed on the detail display. CL programs can write into these descriptors (InDESC) before asserting the interlock. The detail display text then explains why the program forced the interlock (e.g., TOO HOT in the Override Alarm Descriptor, SIODESC). SIOALPR sets the interlock-trip alarm priority.

Trip Descriptor—When a device trips because of an interlock, the cause of the trip is available as an 8-character descriptor from the parameter OVRDDESC.

4.5 COMMANDED STATE (OP)

The command to go to a specific state results in outputs to the process that drive the actual state of the device to the commanded state. The commanded state is represented by the OP parameter in the digital composite point; the commanded state can be State 1 (ST1), State 0 (ST0), or State 2 (ST2). The OP parameter is available for configuration only if the number of digital output connections (NODOPTS) is configured to be greater than 0.

Up to nine Boolean parameters of the form STx_OPy (where x = 0, 1, or 2 for the state number and y = 1, 2, or 3 for the output number) allow the user to specify the state values that are to be stored by the output connections. For each of the three commanded states per output connection, the user must define the value (On or Off) of the state.

CL programs or Logic points can test the parameter INITREQ to see if they are allowed to command a certain state. If INITREQ(i) = ON, then CL programs or Logic points cannot change the output to STATE(i) (i = 0, 1, or 2).

The Status Output command parameter allows Logic points or CL programs (APM or AM) to manipulate outputs (subject to interlocks and the mode attribute). Logic or CL programs can set SOCMD(i) to ON to command State(i) where I = 0, 1, or 2. The same rules apply as storing the corresponding state to the OP parameter, i.e., the mode attribute must be PROGRAM, the corresponding permissive must be available, no override can be active, etc. The actual output (OP) is switched to the requested state only by an OFF to ON change of the corresponding SOCMD(i) flag. There is no priority scheme; the last one changing from OFF to ON controls the output.

OP can be commanded to state 1 or state 0 using the OPCMD parameter. When OPCMD is ON, OP is set to state 1. When OPCMD is OFF, OP is set to State 0. OPCMD can be used when three states are defined, but will only command OP to state 1 or state 0. Overrides take precedence over OPCMD; however, when the override is cleared, the OP will be commanded to the state determined by OPCMD. OPCMD should not be used in conjunction with SOCMD since OPCMD will always override the state set by SOCMD. Like SOCMD, OPCMD only affects OP when the mode attribute is PROGRAM.

CAUTION

Do not configure two different logic outputs to drive OPCMD and SOCMD of the same Digital Composite Point.

Seal-In Circuit—the seal-in circuit is a configurable option (SEALOPT). It is used to unlatch the digital output contacts on detection of the PV, not following the output command as happens on a device drop out (e.g., a stopped motor).

This condition is determined to be true any time the PV is not in state1 or state 2 and either of the following is present

- a command disagree alarm
- an uncommanded state change alarm

If the seal-in option is enabled, when the above condition is detected, the output destinations are set to the state corresponding to OP of state 0, but OP is not altered. The actual state commanded to the output destinations can be observed on the lower part of the group display as OPFINAL. OPFINAL is displayed in reverse video if it is different from OP. OPFINAL is set equal to OP on the next store to OP, thus clearing the seal condition. However, if OPCMD is used, it must be set OFF, then ON to clear the seal condition.

Note that the seal-in option is mutually exclusive with MOMSTATE.

State Change Option—A configurable option (STCHGOPT) allows break before make on output changes. You must configure state 0 as the break, or off, or inactive state. The number of states must equal 3.

When an output change from state 1 to state 2 or from state 2 to state 1 is commanded—

- OP changes to state 0 first.
- if the command disagree time out alarm is configured and a corresponding PV state is applicable, the alarm is held off until a successful feedback has been reached (i.e., PV = state 0).
- the digital composite point waits for a predefined pause time (PAUSETIM).
- OP is then changed to the requested value.

Example—if the break before make option is enabled, commanding a motor to change direction from forward to reverse causes the motor to stop before reversing.

Note that the CL state change statement waits until feedback for the second OP change is successfully received.

Note that the seal-in option is mutually exclusive with MOMSTATE.

4.6 DIGITAL OUTPUTS

The user can specify the latched or pulsed output type for the digital composite point through the configurable output connections.

4.6.1 Latched Output

Configuring the digital composite point for a latched output is accomplished by specifying any parameter other than ONPULSE or OFFPULSE.

4.6.2 Pulsed Output

Configuring the digital composite point for a pulsed output is accomplished through the output connections by specifying the ONPULSE or OFFPULSE parameters as the destination points for the outputs. The width of the pulse is configured at the digital composite data point through the PULSEWTH parameter; it allows the user to specify a pulse width from 0 to 60 seconds as a real number. The selected pulse width applies to all of the output connections.

4.7 OUTPUT CONNECTIONS

The destinations of the outputs (and the output types) from a digital composite point are specified by the user through the DODSTN(1)-DODSTN(3) parameters. The categories of destinations that can be specified are as follows:

- Digital Output Point—Latched Output
- Digital Output Point—Pulsed Output
- Logic Slot Flag
- APM Box Flag
- Process Module Slot Flag
- Device Control Slot Flag
- Array Point Flag (see also 10.3.5)

These destinations must be in the same APM box as the digital composite point that is being configured.

4.7.1 Digital Output Point—Latched Output

To assign the digital composite-point output to the latched output of a digital output point, the user can enter either of the following output connections for the respective DODSTN(n) output connection parameter:

Tagname.SO

or

!DOmmSss.SO

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

SO signifies the status output of the digital output point.

!DO signifies a digital output point. This is the start of the hardware-reference form of addressing the digital output.

mm is the IOP card number from 1-40 of the Digital Output IOP Card within the same APM.

the letter 'S' is a constant.

ss is the slot number in the range from 1-16 on the same Digital Output IOP Card.

4.7.2 Digital Output Point—Pulsed Output

Pulsed outputs can be either of two types: normally off pulsed-on, and normally on pulsed-off. To assign the digital composite-point output to the pulsed output of a digital output point, the user can enter the following information for the respective DODSTN(n) output connection parameter:

For a normally off pulsed-on output:

Tagname.ONPULSE

or

!DOmmSss.ONPULSE

where: Tag Name is the 8- or 16-character name assigned to the data point through the NAME parameter.

ONPULSE or OFFPULSE signifies the pulsed output of the digital output point.

For a normally on pulsed-off output:

Tagname.OFFPULSE

or

!DOmmSss.OFFPULSE

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

ONPULSE or OFFPULSE signifies the pulsed output of the digital output point.

Refer to paragraph 4.7.1 for descriptions of !DOmmSss

4.7.3 Logic Slot Flags

To assign the digital composite-point output to a logic-slot flag, the user can enter the following information for the respective DODSTN(n) output connection parameter:

Tagname.FL(nn)

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

FL signifies a logic-slot flag

nn is the flag number to which the output of the digital composite is sent. The flag number has a range of 7-12: flag numbers 1-6 have dedicated uses and cannot be used as destinations by a digital composite point.

4.7.4 APM Box PV Flags

To assign the digital composite-point output to an APM box PV flag in the same APM box, the user can enter the following information for the respective DODSTN(n) output connection parameter:

Tagname.PVFL or **!BOX.FL(nnnn)**

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

PVFL signifies the PV flag

!BOX specifies the same APM box in which the digital composite point resides.

nnnn is the flag number. Box flags from 1 to 16,384 are available.

4.7.5 Device Control Slot Flags

To assign the digital composite-point input to a Device Control slot flag, the user can enter the following information for the respective DISRC(n) input connection parameter:

Tagname.FL(nn)

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

FL signifies a flag

nn is the flag number to which the input of the digital composite is sent. The flag number has a range of 7-12: flag numbers 1-6 have dedicated uses and cannot be used as destinations by a Device Control point.

4.7.6 Array Point Flags

To assign the digital composite-point input to an Array Point flag, the user can enter the following information for the respective DISRC(n) input connection parameter:

Tagname.FL(nn)

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

FL signifies a flag

nn is the flag number to which the input of the digital composite is sent. The flag number has a range of 1-NFLAG, where NFLAG is the array point parameter that defines the number of flags in the Array point.

4.8 READ-BACK CHECK

Digital Composite and Device Control points can have digital output connections through an SI/Array point to a field device. The field device or its interface may interrupt or change the output and not provide any indication of the change.

If flag data is mapped back from the SI/Array point, a digital output read-back check determines the actual value of the output. After a new output state is stored to the digital output connections, the read-back check is delayed for a time period equal to the feedback-time parameter time (FBTIME) or 4 seconds, whichever is greater. This delay allows the SI/Array output enough time to reach even a slow responding field device before causing a Command Disagree alarm (see subsections 4.14 and 11.5).

If a discrepancy then exists between OPFINAL and OP, the operator is advised. If OPFINAL does not agree with any of the defined states, its state is displayed as NONE.

4.9 INITIALIZATION MANUAL

A digital composite point that has at least one output connected to a DO IOP is forced into the initialization manual condition (INITMAN is On) when one of the following conditions exists:

- The associated DO IOP has failed or been powered off (digital composite point cannot communicate with the DO IOP), the DO IOP is in an idle condition, or in standby manual, for the FTA is missing.
- The associated DO IOP has its initialization request (INITREQ) flag set. The flag may be set because the point is inactive, the point is not configured as a status output type, or one of the output diagnostics has failed.
- The digital composite point is inactive.
- The APMM is in an idle condition.

INITMAN is displayed at the Universal Station for the particular digital composite point to inform the user that one of the above conditions has occurred. When the condition is corrected, INITMAN is set to Off.

When INITMAN transitions from On to Off, the digital composite point provides an output value OP as follows:

- If override interlocks are active and not bypassed, the OP value corresponds to the highest priority override interlock.
- Otherwise, in cases where feedback is configured, the stored OP value tracks the PV state if the PV state is valid (that is, not bad or moving).
- Otherwise, (if the input is bad) the OP value is back-initialized from the output connections if there are no output connections to the ONPULSE/OFFPULSE parameters, and if a valid OP value can be constructed from the values of the output connections. For output connections to ONPULSE or OFFPULSE, OP is set to State 0.
- Otherwise, the OP is set to State0.

4.10 DIGITAL INPUTS

When configuring digital inputs of the digital composite data point, the user can specify the input connections, PV states, PV source and options, alarming, and change-of-state events. The input portion of a digital composite point can be configured only if the user has entered 1 or 2 for the number-of-digital-inputs parameter (NODINPTS). A functional diagram of the input portion of the digital composite point is shown in Figure 4-3.

4.11 INPUT CONNECTIONS

The inputs to a digital composite point are specified by the user through digital input-source parameters DISRC(1)-DISRC(2). Inputs to the digital composite point are designated as Input 1 and Input 2, and they can be obtained from any of the following sources:

- Digital Input Point—PV
- Digital Output Point—SO
- Logic Slot Output
- Logic Slot Flag
- APM Box Flag PV
- Process Module Slot Flag
- Array Slot Flag
- Device Control Slot Flag

These sources must be in the same APM box as the digital composite point that is being configured.

The status of Input 1 is represented by parameter D1; Input 2 is represented by parameter D2. Input 2 can be configured only when the entry for the number-of-digital-inputs parameter (NODINPTS) is 2.

4.11.1 Digital Input Point—PV

To assign the PV of a digital input point to the PV input of a digital composite point in the same APM box, the user can enter the following information for the respective DISRC(n) input connection parameter:

Tagname.PVFL
or
!DImmSss.PVFL

where: Tagname is the 8- or 16-character name of the point that will provide the PV.

- PVFL signifies the PV of the respective digital input point.
- !DI signifies a digital input point. This is the start of the hardware reference address.
- mm is the Digital Input IOP Card number from 1-40 within the same APM.
- the letter “S” is a constant.
- ss is the slot number in the same Digital Input IOP Card in the range from 1-32.

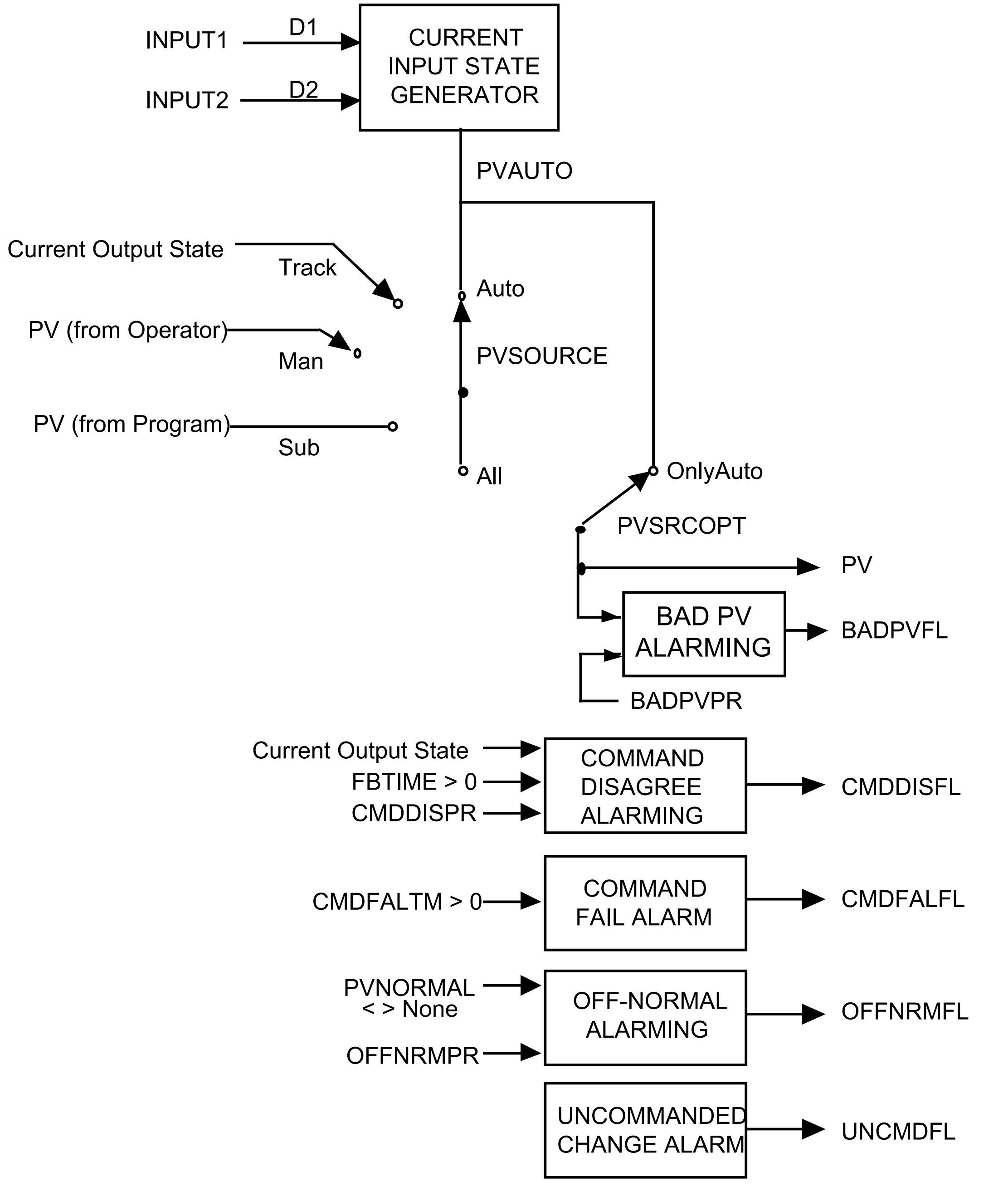


Figure 4-3 — Digital Input, Functional Diagram

15017

4.11.2 Logic Slot Output

To assign the logic slot output to the PV input of a digital composite point, the user can enter the following information for the respective DISRC(n) input connection parameter:

Tagname.SO(nn)

where: Tagname is the 8- or 16-character name assigned to the logic slot that is providing the output.

SO signifies the output of the logic block.

nn is the logic block number in the range from 1-24

4.11.3 Logic Slot Flags

To assign a logic slot flag to the input of a digital composite point, the user can enter the following information for the respective DISRC(n) input connection parameter:

Tagname.FL(nn)

where: Tagname is the 8- or 16-character name assigned to the logic slot that is providing the flag.

FL signifies the logic-slot flag

nn is the flag number in the range from 1-12.

4.11.4 APM Box PV Flags

To assign a APM box PV flag to the input of a the digital composite point, the user can enter the following information for the respective DISRC(n) input connection parameter:

Tagname.PVFL

or

!BOX.FL(nnnn)

where: Tagname is the 8- or 16-character name assigned to the box PV flag

PVFL signifies the PV flag

!BOX specifies the APM box.

FL(nnnn) is the flag number that has a range from 1 to 16384.

4.11.5 Device Control Slot Flags

To assign the digital composite-point output to a Device Control slot flag, the user can enter the following information for the respective DODSTN(n) output connection parameter:

Tagname.FL(nn)

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

FL signifies a flag

nn is the flag number to which the output of the digital composite is sent. The flag number has a range of 7-12: flag numbers 1-6 have dedicated uses and cannot be used as destinations by a Device Control point.

4.11.6 Array Point Flags

To assign the digital composite-point output to an Array Point flag, the user can enter the following information for the respective DODSTN(n) output connection parameter:

Tagname.FL(nn)

where: Tagname is the 8- or 16-character name assigned to the data point through the NAME parameter.

FL signifies a flag

nn is the flag number to which the output of the digital composite is sent. The flag number has a range of 1-NFLAG, where NFLAG is the array point parameter that defines the number of flags in the Array point.

4.11.7 Process Module Slot Flags

To assign a Process Module slot flag to the input of a digital composite point, enter the following information for the respective DISRC(n) input connection parameter:

Tagname.FL(nn)

where: Tagname is the 8- or 16-character name assigned to the Process Module flag

FL(nn) signifies the flag

4.12 CURRENT INPUT STATE (PV)

The flexibility of the digital composite point allows the user to assign the states of the PV for each possible combination of digital inputs, so that the states correspond to the different applications in which this point type can be used. The PV parameter represents the current state of the interfaced device and is derived from inputs D1 and D2 that can be feedback signals from the process. Separate parameters are used to configure a single-input point and a dual-input point.

NOTE

The inputs to a digital composite point are usually the PVs from digital input points. The digital input points should be configured as component points that force the input direction to be direct (as opposed to reverse). The actual direct/reverse action can be configured by assigning the appropriate PV state to the input as described in the following paragraphs.

For a single-input point, there is only one input parameter (D1). D1 has two possible PV states (PVstate0 and PVstate1) that can be assigned to either of the following input conditions:

D1 = 1 (D1 is ON; parameter D1_1)

D1 = 0 (D1 is OFF; parameter D1_0)

The user has to assign only PVstate0 or PVstate1 to parameter D_1; the system automatically assigns the other PV state to parameter D1_0.

For a dual-input point, there are two input parameters (D2 and D1) that together have four possible combinations of input values as follows:

D2D1 = 00 (D2 is OFF, D1 is OFF; parameter D2D1_00)

D2D1 = 01 (D2 is OFF, D1 is ON; parameter D2D1_01)

D2D1 = 10 (D2 is ON, D1 is OFF; parameter D2D1_10)

D2D1 = 11 (D1 is ON, D2 is ON; parameter D2D1_11)

To these four combinations of input values, the user can assign any four of the following five PV states:

Pvstate1

Pvstate0

Pvstate2

MovPV

BadPV

PVstates1, 0, and 2 cause the PV indicator to be lighted in the respective state box on the group display when the assigned D2D1 input conditions are satisfied. The MovPV and BadPV states cause the respective MOVPVTXT or BADPVTXT descriptor to appear below the state boxes on the group display.

4.13 PV Source

The PV source parameter (PVSOURCE) determines the source of the current PV state for the digital input portion of the digital composite data point. The possible sources of the current PV state are as follows:

Source	Description
Man (Manual)	Current PV state is provided by the operator from the Universal Station.
Auto (Automatic)	Current PV state is derived from Input 1 (D1) and Input 2 (D2).
Track	Current PV state is the commanded output state
Sub (Substituted)	Current PV state is provided by a user program

4.13.1 PV Source Option

During configuration, the user can specify the PV sources that can be used for this data point. Parameter PVSRCOPT allows the user to select the PV source as being only AUTO, or to select all the PV sources in the above listing as allowable sources of the PV. Note that the organizational summary display lists all the points for which PVSOURCE = TRACK.

4.14 ALARMING

The digital composite point can be configured to detect and report command disagree, command fault, uncommanded change, or off-normal alarms. The user has the option of specifying no alarming for the data point. Digital composite points also generate a bad PV alarm when any input is coming from a digital input point that has a bad PV flag (BADPVFL) status of ON.

During the command disagree timeout interval, determined by the feedback-time parameter, FBTIME, it is possible, in some instances, that the state defined as BADPV could occur in the PV inputs, but not reflect an actual failure of the inputs or associated hardware. To lessen operator confusion, these "state" BADPVs should be ignored. Therefore, only BADPVs resulting from detected input errors are declared by the point during the timeout interval. If a "state" BADPV occurs, the previous PV remains displayed.

These alarm options can be configured only if the digital composite point is configured to have inputs, or inputs and outputs. The BADPVPR parameter determines priority of the bad PV alarm.

4.14.1 Command Disagree, Command Fail, and Uncommanded Change

When the commanded-output state is changed and the actual input PVstate does not change accordingly within a predefined feedback time, a command disagree alarm is generated. The feedback time (1 to 1000 seconds) is specified by the FBTIME parameter and the timer starts whenever the OP value changes. A command disagree alarm is also generated if OP changes are caused by the interlocks.

This alarm condition returns to normal when the input PV state and the commanded-output state are the same. If the commanded state is momentary, no alarm is generated. For example, a motor may have two PV states (RUN and STOP), but there may be three commanded output states (RUN, STOP, and JOG) where JOG is defined as a momentary state. Command-disagree alarming is performed for only the RUN and STOP commanded output states.

The command fail alarm is similar to the command disagree alarm, but instead of waiting for the actual state to equal the commanded state, a check is made to verify that the PV changed from its original value to any other value within a configurable time interval. For slow devices, absence of this alarm provides feedback that the device responded to the command, even if it has not yet moved to its final position. The command fail alarm is enabled by entering an integer greater than zero for the command failure timeout parameter, CMDFALTM.

If a change does not occur in the commanded output state but the input PV state changes (and the PV is not bad), an uncommanded-change alarm is generated. This alarm condition returns to normal when the input PV state and the commanded state are the same. If the point state has been configured as being momentary, this type of alarm is not applicable. Alarm priority for the command disagree alarm, the command fail alarm, and the uncommanded change alarm is determined by the Command Disagree Priority (CMDDISPR) parameter.

4.14.2 Off-Normal

Detection of off-normal alarms is configured by selecting an FBTIME greater than zero. The normal state of the PV input is defined by the user through the PVNORMAL parameter.

When the PV input state is different than the state specified by the PVNORMAL parameter, the off-normal alarm is generated. The alarm condition returns to normal when the PV input state and the specified PV normal state are the same. When command disagree is configured, the off-normal alarm is inhibited if PVNORMAL = OP. This is to avoid two alarms at the same time and help isolate failures. You can set the priority of the off-normal alarm with the OFFNRMPR parameter.

4.15 CHANGE-OF-STATE EVENTS

Any transitions in the PV input state can be reported as events for journaling and for causing the event-initiated processing of points in the Application Module (AM). It is configured by entering EIP for the event-report-option parameter EVTOPT. The user must enter the tag name of the AM or CM point using the EIPPCODE parameter. If only journaling is required, the EIPPCODE parameter can be set to a null tag name.

4.16 LOCAL MANUAL INDICATION

Field devices that are interfaced by a digital composite point often have a local HAND/OFF/AUTO (often called HAND/OFF/REMOTE) switch. Unless this switch is in AUTO position, the control system (APM) may not have any control over that device. The user can optionally feedback the switch position into the APM to obtain some display indication for the Universal Station operator. This indication is provided by the word LOCALMAN appearing at the bottom of the digital composite point on a group display. Further, when in local manual, any changes to the output by the operator, and user programs, are prohibited. The override interlocks are still active when the local manual condition is on. When the local manual condition is on, both OP and OPFINAL follow the PV value (the actual state of the device).

To support HAND/OFF/AUTO switches, a Boolean flag called LOCALMAN is provided. The ON state indicates that the switch is not in AUTO position. The user can hard-wire the AUTO position of the HAND/OFF/AUTO switch through a digital input point. The state of the digital input state is then stored to the LOCALMAN flag from logic slot outputs or sequence programs.

4.17 MAINTENANCE STATISTICS

The Maintenance Statistics page of the Digital Composite Point Detail Display provides statistical information about the point. For example—

- accumulated number of transitions to each state
- date/time of most recent statistics reset
- accumulated time in each state
- date/time of most recent change to each state
- accumulated number of safety interlock overrides

Statistics are enabled by entries in the Maintenance Statistics section during point configuration. You can enter the maximum number of hours that you want to allow in each state and maximum number of transitions that you want to allow into each state. Even if either maximum is reached, the Digital Composite point does not provide an alarm, but a CL program can be written to monitor the statistics and take action.

4.17.1 Reset and Redtag

The above statistics are accumulated since the most recent reset. A program can reset the statistics with the RESETFL parameter anytime or they can be reset from the Maintenance Statistics page as explained below.

The Maintenance Statistics display contains a REDTAG target. The REDTAG and RESET targets work as follows:

> If you select the REDTAG target (and ENTER), the point’s REDTAG parameter is set to ON and a RESET target appears. Selecting the RESET target (and ENTER), resets the points maintenance statistic values. If you select REDTAG (and ENTER, the REDTAG condition clears and the RESET target disappears.

LOGIC POINT
Section 5

This section defines the functions available in the logic point. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

5.1 FUNCTIONS

The logic point provides a configurable mix of logic capability that, together with a digital composite point, provides the basis for integrated logic functions. The logic point is also fully integrated with the regulatory functions in the APM. A logic point consists of logic blocks, flags, numerics, user-defined generic descriptors, input connections, and output connections as shown in Figure 5-1.

5.2 CONFIGURING THE LOGIC POINT

The logic point can have up to 12 input connections, 24 logic blocks, and 12 output connections. The actual number of these is specified through the entry made for LOGMIX parameter as shown in the following chart:

LOGMIX Param. Entry	**No. of Input Connections**	**No. of Logic Blocks**	**No. of Output Connections**
12_24_4	12	24	4
12_16_8	12	16	8
12_8_12	12	8	12

5.2.1 Logic Input Connections

Inputs to the logic point are assigned by the user during configuration by using logic input connections. The user can specify up to 12 inputs that are assigned to logic-point inputs L(1) to L(12), as shown in Figure 5-1. The logic input connections can be specified using the "Tagname.Parameter" format, or the hardware reference address format by using the LISRC parameter.

The inputs to the logic point can be obtained from any of the following sources:

- Any Boolean, integer, enumeration, self-defining enumeration, or real parameter within this APMM or in another UCN node. Note that the integer, enumeration, or self-defining enumeration parameters are automatically converted to real values.
- The PV flag parameter (PVFL) of a DI point from a DI IOP in this APM.
- The Bad PV flag parameter (BADPVFL) of a DI point from a DI IOP in this APM.
- The Status Output parameter (SO) from a DO point in this APM.
- The Initialization Request flag (INITREQ) from a DO point in this APM.
- Any parameter other than the DI point's PVFL from the IOPs in this APM. Maximum of four such input connections can be configured (these follow the same kind of restrictions that CL/APM has for prefetches).

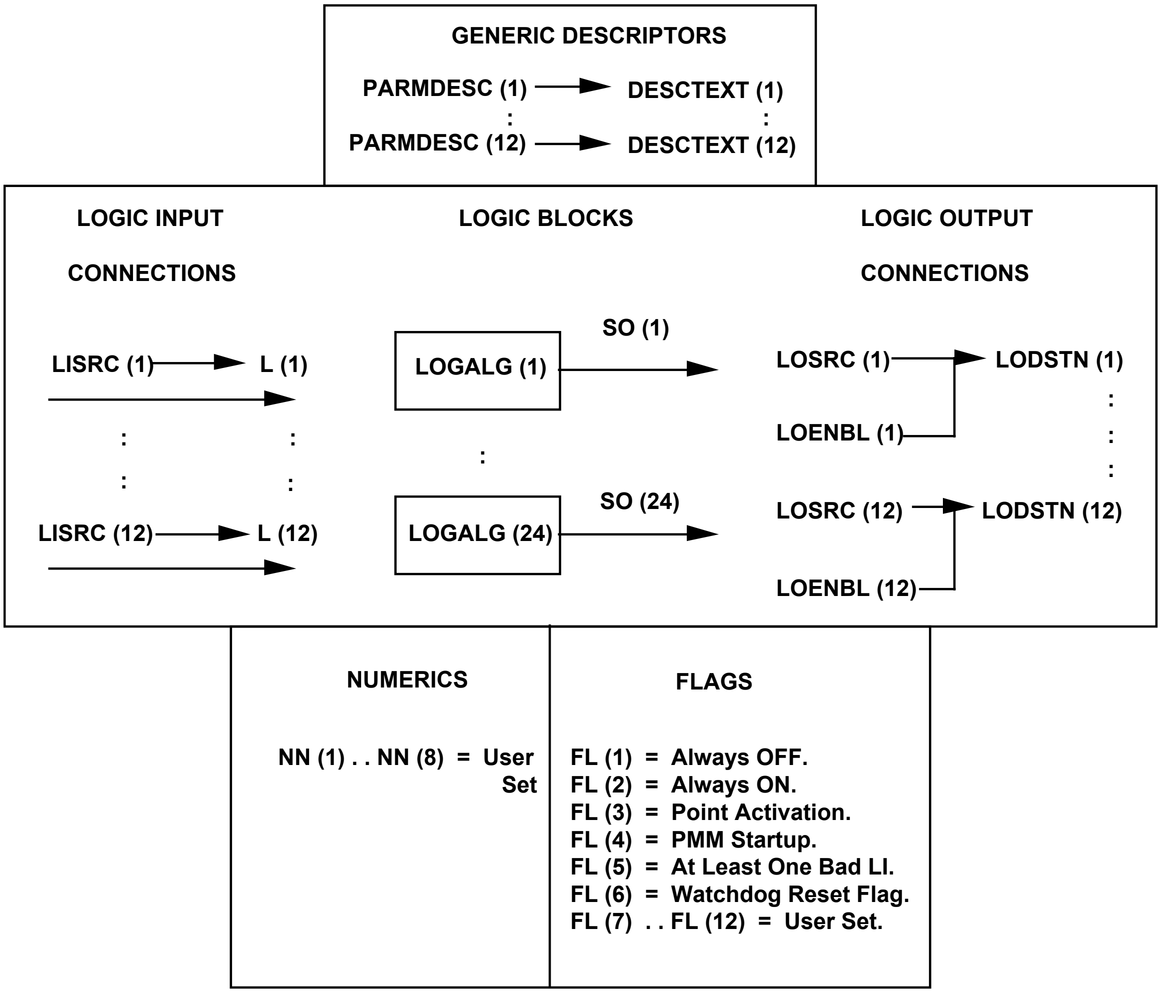


Figure 5-1 — Logic Point

5.2.1.1 Effect of Communication and Configuration Errors on Logic Input Connections

Because the logic point can be used for implementing safety interlocks, special handling is provided for inputs, especially Boolean inputs, that cannot be obtained because of communication or configuration errors.

For example, an input obtained from a failed DI Processor can result in a communication error. Alternatively, an input can be coming from a digital composite point in another APM (or PM) and the point mix in that APM can be changed such that the specified point no longer exists. This results in a configuration error. In order for the logic to be able to continue in spite of a configuration error, the following special features are provided:

- Bad Boolean Inputs—If a Boolean input is not successfully fetched, its value is defaulted based on the logic input bad handling option parameter (LIBADOPT) as follows:

  LIBADOPT = On The On state is substituted for the unsuccessful input.

  LIBADOPT = Off The Off state is substituted for the unsuccessful input.

  LIBADOPT = Hold The previous value (the last successfully fetched value) is substituted for the unsuccessful input. On startup, the previous value is defaulted to the Off state. If required, the startup condition can be monitored by checking the startup flag.

- Bad Real Inputs—If a real input cannot be successfully fetched, its value is defaulted to NaN. If required, special action can be taken in the logic by checking it with the "check for bad" logic algorithm.
- Input Bad Flag—When an error is encountered in fetching the value of any input, flag FL(5) is set to On. Note that flag FL(5) is not set to On if a real input is successfully fetched but has a value of NaN.
- Processing Order—The logic points defined for each APM are processed in decending order. Within each logic point, the inputs are calculated first, then the gates are processed, then the outputs are processed, each in ascending order.

5.2.2 Flags

Twelve flags, FL(1) to FL(12), are provided for each logic point. The states of flags FL(1) to FL(6) are controlled by the APM and cannot be changed by the user. Flags FL(7) to FL(12) are assigned by the user for controlling the path of the logic in the respective logic point.

The flags are described in the following listing:

Flag	Description
FL(1)	Always Off (but see 5.2.5 when used for logic output enable).
FL(2)	Always On.
FL(3)	On only if this logic point is processing for the first time after becoming active.
FL(4)	On only if this point is processing for the first time after the APMM box state has transitioned from Idle to Run, or the APMM has gone into the Run state following a power-up.
FL(5)	On if at least one of the logic input connections was unsuccessful in fetching the specified source parameter.
FL(6)	Used by the watchdog timer logic algorithm. It is set to On by the function or device being monitored to indicate that it is alive. The watchdog timer algorithm always sets this flag to Off. Flag FL(6) cannot be changed from the Universal Station.
FL(7)-FL(12)	Can be changed as determined by the user. They can be changed from the Universal Station, other logic points, or user programs.

Flags FL(1) to FL(5) cannot be changed by the user, but the states of these flags can be used as inputs to the logic points and can also be read by user programs.

5.2.3 Numerics

Eight numerics, NN(1) to NN(8), are provided with each logic point. The numerics can be used as reference values for the comparison logic algorithms, or they can be used as source parameters for the output connections when writing predefined analog constants to other points. The values of the numerics can be changed from the Universal Station, by other logic points, or by user programs. A bad numeric input typically has the value NaN.

5.2.4 Logic Blocks

The logic operations in a logic point are performed by the logic blocks. Up to 24 logic blocks can be configured by the user for each logic point. The actual number of logic blocks configured is determined by the logic mix described in paragraph 5.2. Each logic block consists of

- Up to 4 inputs (specified by S1, S2, S3, S4.)
- One logic algorithm (specified by LOGALG)
- One Boolean output (specified by SO)

The parameters associated with a logic block are always qualified with an array index corresponding to the number of that logic block. For example, LOGALG(4) represents the algorithm configured in the logic block number 4, whose output is SO(4), and the inputs may be S1(4), S2(4), etc.

On the first time through configuration, the outputs of all logic blocks are set to OFF; however, on a subsequent restart (for example: IDLE to RUN transition of the APMM state, or INACTIVE to ACTIVE transition of the point execution state), all the outputs are held at their previous values.

Processing Order—The logic points defined for each APM are processed in descending order. Within each logic point, the inputs are calculated first, then the gates are processed, then the outputs are processed, each in ascending order.

5.2.5 Output Connections

Logic output connections are used to write the values of local parameters of a respective logic point to the configured destinations. Up to 12 output connections can be configured for each logic point. The actual number of output connections is determined by the LOGMIX parameter described in paragraph 5.2. The destinations are specified by parameter LODSTN by using the "Tagname.Parameter" format or the hardware reference address format.

The logic output connection can write the selected local parameters of a logic point to any of the following destinations:

- Any Boolean, integer, enumeration, self-defining enumeration, or real parameter in this APMM, or another UCN node

- Any Boolean, integer, enumeration, self-defining enumeration, or real parameter in the IOPs in this APM. A maximum of eight such connections can be configured for each point. The remaining four output connections can be used to write to boolean or real parameters that reside in this APMM or in another UCN node.

The user must specify the local parameter within the logic point and the destination to which it is to be written. The local parameter to be written is specified as the logic output source (LOSRC(n), where n = 1 to 12). The source parameter for the output connection can be selected from any of the flags (specified as FL1, FL2, etc.), numerics (specified as N1, N2, etc.), external inputs (specified as L1, L2, etc.), or the output of any of the logic blocks (specified as SO1, SO2, etc.).

Associated with each output connection is a logic output enable flag, LOENBL(n). The parameter pointed to by LOENBL(n) must be ON for the corresponding output connection to write to the specified destination. If store-only-on-a-change is required, the appropriate condition to store can be determined by the CHDETECT logic algorithm and then used to manipulate the enable signal; or if the FL1 parameter is specified for the output enable and the output data type is Boolean, then output occurs on change only. The logic output enable flag can be obtained from any of the flags (specified as FL1, FL2, etc.), external inputs (specified as L1, L2, etc.), or the output of any of the logic blocks (specified as SO1, SO2, etc.).

5.2.6 Generic Descriptors

Up to 12 user-defined generic descriptors are provided with each logic point for identifying parameters of the logic point with custom names that are to be shown on Universal Station displays. Generic descriptors can be assigned to any of the following logic-point parameters:

- Flags (FL1, FL2, etc.)
- Numerics (N1, N2, etc.),
- External inputs (L1, L2, etc.),
- Outputs of any of the logic blocks (SO1, SO2, etc.).

Typically, the generic descriptors identify critical interlock signals, operator-adjusted parameters, etc.

The actual number of descriptors used is determined by the NODESC parameter for this logic point. For each descriptor, the parameter to which it is attached is defined by PRMDESC(n) parameter, and the corresponding 8-character descriptor is defined by GENDESC(n).

5.3 LOGIC BLOCK ALGORITHMS

Logic blocks perform logic functions. Each logic block can have up-to-four inputs and produce a single Boolean output, depending on the selected algorithm. The algorithm can require real or Boolean inputs. The real inputs can be obtained from any of the numerics (specified as NN1, NN2, etc.) or the external inputs (specified as L1, L2, etc.). The Boolean inputs can be any of the flags (specified as FL1, FL2, etc.), external inputs (specified as L1, L2, etc.), or the output of any of the logic blocks (specified as SO1, SO2, etc.).

When one of the external inputs (L(1), L(2), .. L(12)) is used as an input to an algorithm, the value fetched can be real or Boolean. If the algorithm requires a real input and the fetched value is Boolean, it is treated as NaN. If the algorithm requires a Boolean input and the fetched value is real, it is treated as OFF. In either case, the bad input flag, FL(5), is not affected.

The following logic block algorithms are supported:

- Null (NULL)
- AND Gate (AND)
- OR Gate (OR)
- NOT Gate (NOT)
- NAND Gate (NAND)
- NOR Gate (NOR)
- XOR Gate (XOR)
- Qualified OR Gate With 2 Inputs ON (QOR2)
- Qualified OR Gate With 3 Inputs ON (QOR3)
- Switch (SWITCH)

- Compare Equal With Deadband (EQ)
- Compare Not Equal With Deadband (NE)
- Compare Greater Than With Deadband (GT)
- Compare Greater Than or Equal With Deadband (GE)
- Compare Less Than With Deadband (LT)
- Compare Less Than or Equal With Deadband (LE)

- Check for Bad (CHECKBAD)

- Fixed Size Pulse (PULSE)
- Pulse With Maximum Time Limit (MAXPULSE)
- Pulse With Minimum Time Limit (MINPULSE)
- Delay (DELAY)
- On Delay (ONDLY)
- Off Delay (OFFDLY)

- Watchdog Timer (WATCHDOG)

- Flipflop (FLIPFLOP)

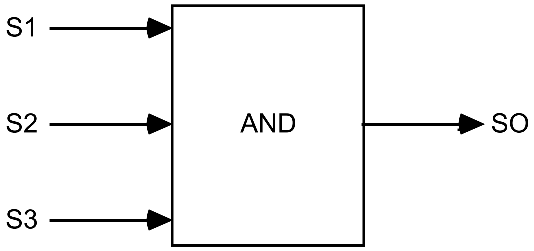
- Change Detect (CHDETECT)

5.3.1 Null (Null)

This logic algorithm provides an output (SO) that is always set to OFF.

5.3.2 AND Gate (AND)

This algorithm provides a 3-input AND gate, with each input (S1, S2, and S3) having the capability of being optionally inverted, as required. The output SO is determined as follows:

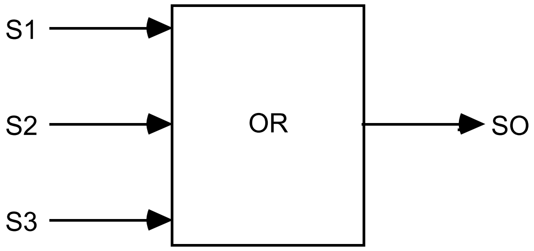


```
SO = (((S1REV = OFF) AND (S1 = ON)) OR ((S1REV = ON) AND (S1 = OFF)))
          AND
      (((S2REV = OFF) AND (S2 = ON)) OR ((S2REV = ON) AND (S2 = OFF)))
          AND
      (((S3REV = OFF) AND (S3 = ON)) OR ((S3REV = ON) AND (S3 = OFF)))
```

REV = indicates if the input signal is reverse acting.

5.3.3 OR Gate (OR)

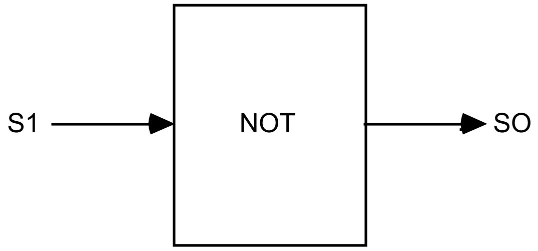
This logic algorithm provides a 3-input OR gate, with each input (S1, S2, and S3) having the capability of being optionally inverted, as required. The output SO is determined as follows:



```
SO = (((S1REV = OFF) AND (S1 = ON)) OR ((S1REV = ON) AND (S1 = OFF)))
          OR
      (((S2REV = OFF) AND (S2 = ON)) OR ((S2REV = ON) AND (S2 = OFF)))
          OR
      (((S3REV = OFF) AND (S3 = ON)) OR ((S3REV = ON) AND (S3 = OFF)))
```

5.3.4 NOT Gate (NOT)

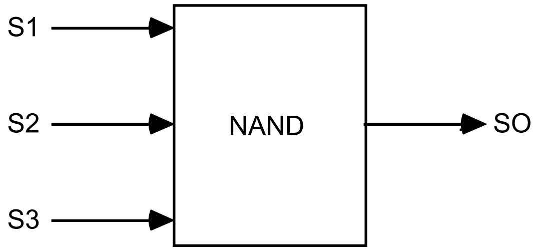
This algorithm provides the Boolean inversion (NOT) function. The output SO is the inversion of the input S1 as follows:



```
IF  (S1 = ON)  THEN  SO = OFF
ELSE  SO = ON
```

5.3.5 NAND Gate (NAND)

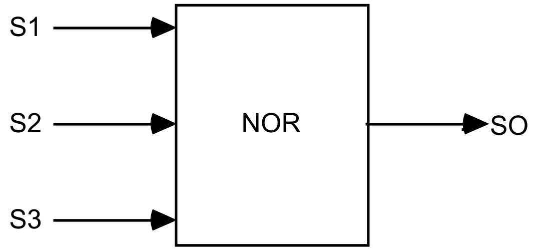
This algorithm provides a 3-input NAND gate, with each input (S1, S2, and S3) having the capability of being optionally inverted. The output SO is determined as follows:



```
SO = NOT ( (((S1REV = OFF) AND (S1 = ON)) OR ((S1REV = ON) AND (S1 = OFF)))
          AND
       (((S2REV = OFF) AND (S2 = ON)) OR ((S2REV = ON) AND (S2 = OFF)))
          AND
       (((S3REV = OFF) AND (S3 = ON)) OR ((S3REV = ON) AND (S3 = OFF))))
```

5.3.6 NOR Gate (NOR)

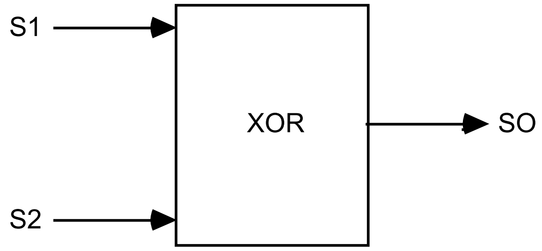
This logic algorithm provides a 3-input NOR gate, with each input (S1, S2, and S3) having the capability of being optionally inverted. The output SO is determined as follows:



```
SO = NOT ( (((S1REV = OFF) AND (S1 = ON)) OR ((S1REV = ON) AND (S1 = OFF)))
        OR
     (((S2REV = OFF) AND (S2 = ON)) OR ((S2REV = ON) AND (S2 = OFF)))
        OR
     ((S3REV = OFF) AND (S3 = ON)) OR ((S3REV = ON) AND (S3 = OFF))))
```

5.3.7 XOR Gate (XOR)

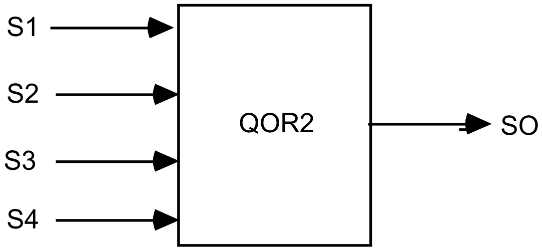
This algorithm provides a 2-input exclusive-OR gate. The output SO is determined as follows:



SO = NOT (S1 = S2)

5.3.8 Qualified OR Gate with 2 Inputs ON (QOR2)

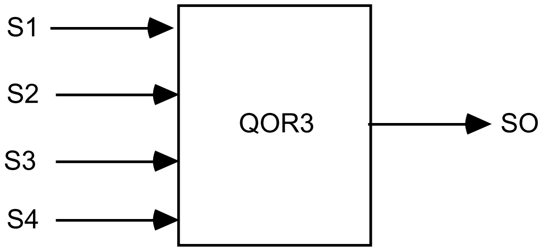
This algorithm provides a 4-input qualified-OR function that requires at least two inputs to be ON before output SO is set to ON. The output is determined as follows:



```
IF (at least 2 inputs out of S1, S2, S3, and S4 are ON) THEN
    SO = ON
ELSE
    SO = OFF
```

5.3.9 Qualified OR Gate with 3 Inputs ON (QOR3)

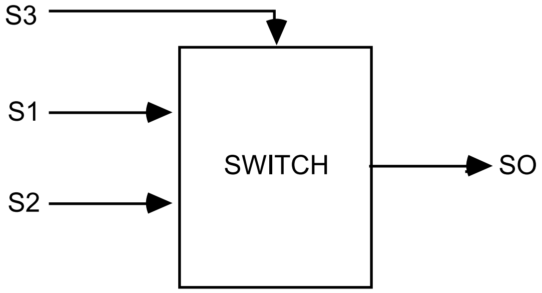
This algorithm provides a 4-input qualified-OR function that requires at least three inputs to be ON before output S0 is set to ON. The output is determined as follows:



```
IF (at least 3 inputs out of S1, S2, S3, and S4 are ON) THEN
    SO = ON
ELSE
    SO = OFF
```

5.3.10 Switch (SWITCH)

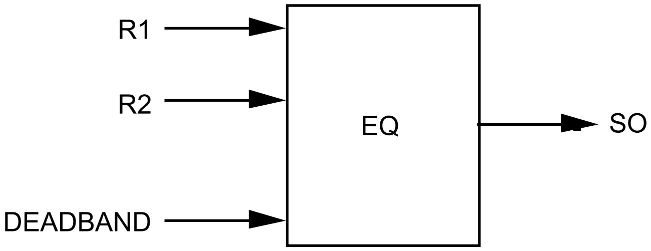
This algorithm provides a 2-input switch. Output SO is determined as follows:



```
IF (S3 = ON) THEN
    SO = S1
ELSE
    SO = S2
```

5.3.11 Compare Equal with Deadband (EQ)

This algorithm compares two real inputs (R1 and R2) for being "almost equal" or within a specified deadband. Output SO is determined as follows:

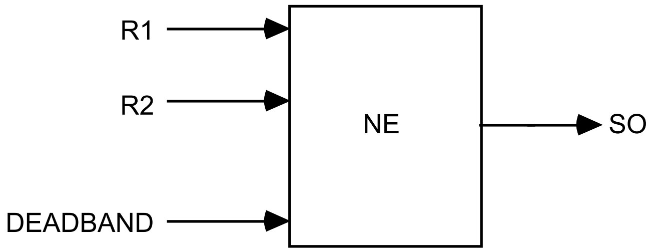


```
IF (ABS(R1 - R2) <= DEADBAND) THEN
    SO = ON
ELSE
    SO = OFF
```

If R1 and/or R2 inputs are "NaN", then SO is not changed.

5.3.12 Compare Not Equal with Deadband (NE)

This algorithm compares two real inputs for NOT being "almost equal," or being outside of a prespecified deadband. Output SO is determined as follows:

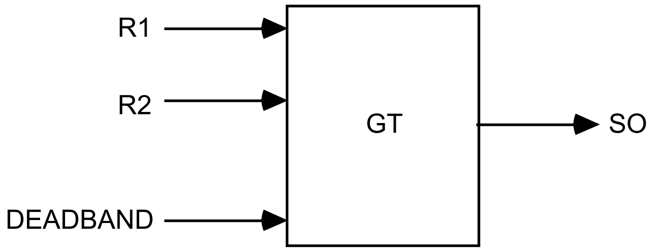


```
IF (ABS(R1 - R2) > DEADBAND) THEN
    SO = ON
ELSE
    SO = OFF
```

If R1 and/or R2 inputs are NaN, S0 is not changed.

5.3.13 Compare Greater Than with Deadband (GT)

This algorithm compares a real input (R1) for being greater than another real input (R2) with a predefined deadband. Output SO is determined as follows:

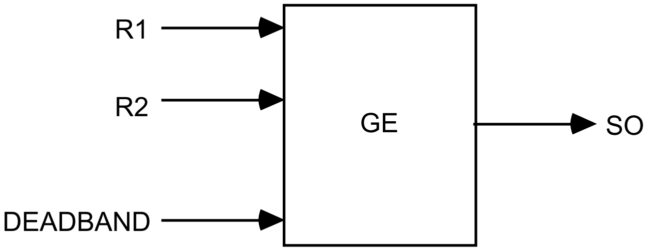


```
IF (R1 > R2) THEN
    SO = ON
ELSE IF (R1 <= (R2 - DEADBAND)) THEN
    SO = OFF
ELSE
    SO is not changed.
```

If R1 and/or R2 inputs are NaN, S0 is not changed.

5.3.14 Compare Greater Than or Equal with Deadband (GE)

This algorithm compares a real input (R1) for being greater than or equal to another real input (R2) with a specified deadband. Output SO is determined as follows:

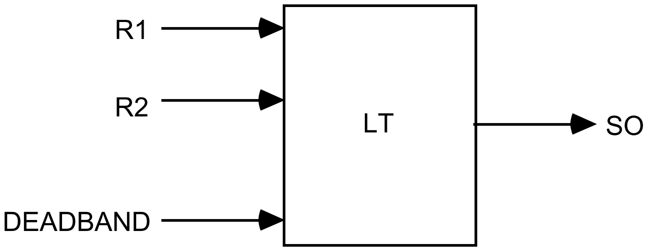


```
IF (R1 >= R2) THEN
     SO = ON
ELSE IF (R1 < (R2 - DEADBAND)) THEN
     SO = OFF
ELSE
     SO is not changed.
```

If R1 and/or R2 inputs are NaN, SO is not changed.

5.3.15 Compare Less Than with Deadband (LT)

This algorithm compares a real input (R1) for being less than another real input (R2) with a predefined deadband. Output SO is determined as follows:

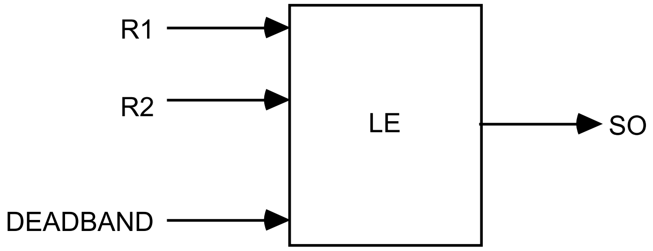


```
IF (R1 < R2) THEN
     SO = ON
ELSE IF (R1 >= (R2 + DEADBAND)) THEN
     SO = OFF
ELSE
     SO is not changed.
```

If R1 and/or R2 inputs are NaN, SO is not changed.

5.3.16 Compare Less Than or Equal with Deadband (LE)

This logic algorithm compares a real input (R1) for being less than or equal to another real input (R2) with a predefined deadband. Output SO is determined as follows:

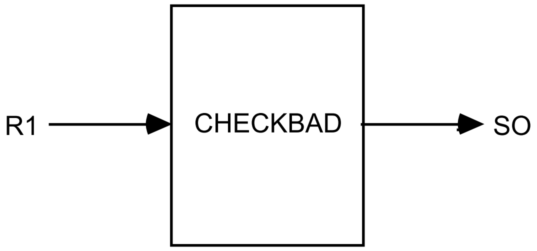


```
IF (R1 <= R2) THEN
    SO = ON
ELSE IF (R1 > (R2 + DEADBAND)) THEN
    SO = OFF
ELSE
    SO is not changed.
```

If R1 and/or R2 inputs are NaN, SO is not changed.

5.3.17 Check for Bad (CHECKBAD)

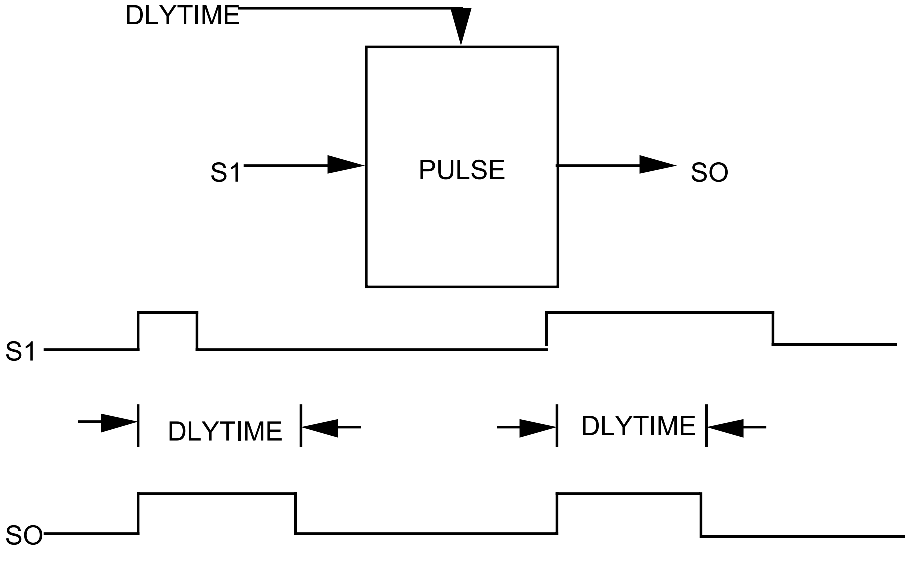
This logic algorithm checks a real input (R1) for being bad (equal to NaN). Output SO is determined as follows:



```
IF (R1 is "NaN") THEN
    SO = ON
ELSE
    SO = OFF
```

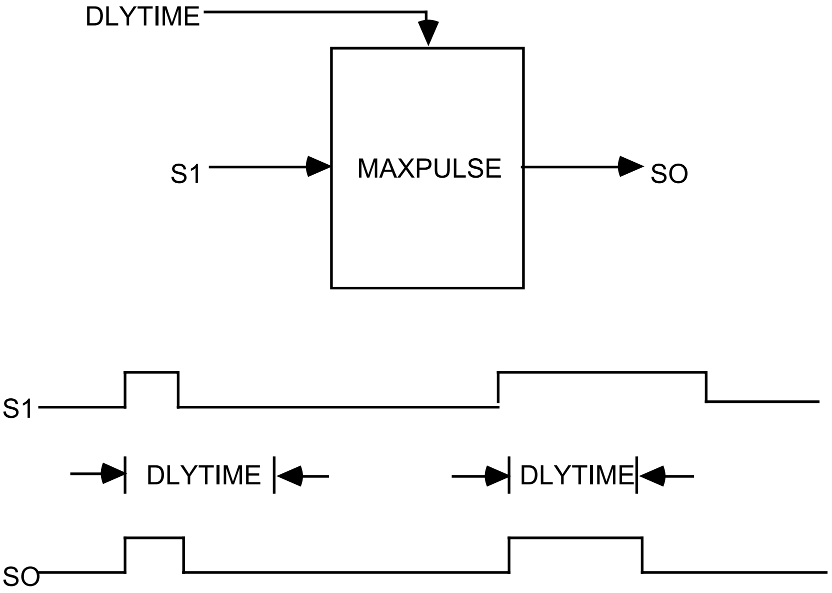
5.3.18 Fixed-Size Pulse (PULSE)

This logic algorithm provides a fixed-size output pulse at the SO output each time the S1 input transitions from the OFF state to the ON state. The output pulse width (in seconds) is specified by the DLYTIME parameter. If the delay time is less than or equal to one sample time (of the logic point), it is assumed to be equal to one sample time. Another output pulse cannot be generated until the generation of the preceding pulse has been completed.



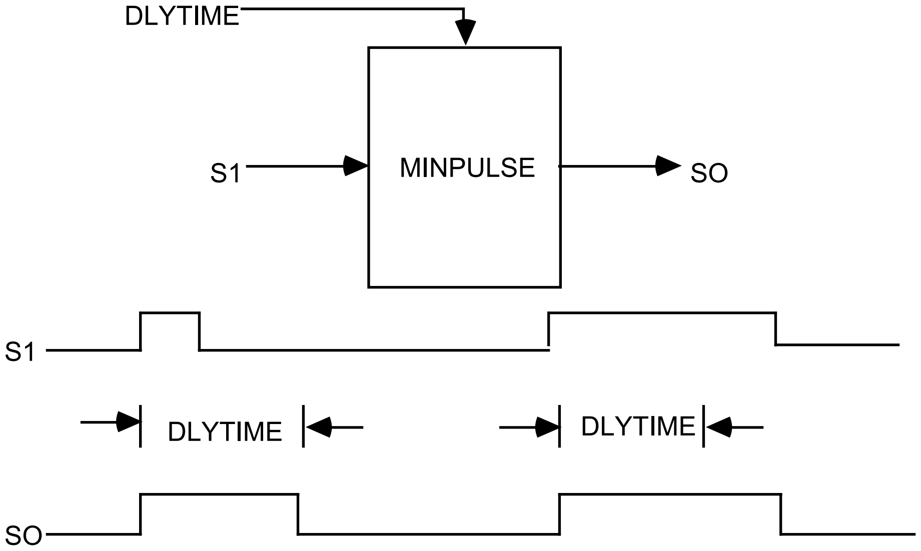
5.3.19 Pulse with Maximum Time Limit (MAXPULSE)

This logic algorithm provides a pulse at the SO output each time the S1 input transitions from the OFF to the ON state. If the input stays ON longer than a predefined time, the output pulse is terminated. The maximum output pulse width (in seconds) is specified by the DLYTIME parameter. If the specified output pulse width is less than or equal to one sample time (of the logic point), it is assumed to be equal to one sample time.



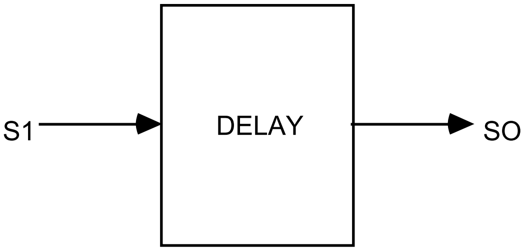
5.3.20 Pulse with Minimum Time Limit (MINPULSE)

This algorithm generates a pulse at the SO output each time the S1 input transitions from the OFF to the ON state. If the S1 input stays ON for an interval that is less than the specified time, the output pulse is extended until the timed interval is over. The minimum output pulse width (in seconds) is specified by the DLYTIME parameter. If the specified output pulse width is less than, or equal to, one sample time (of the logic point), it is assumed to be equal to one sample time.



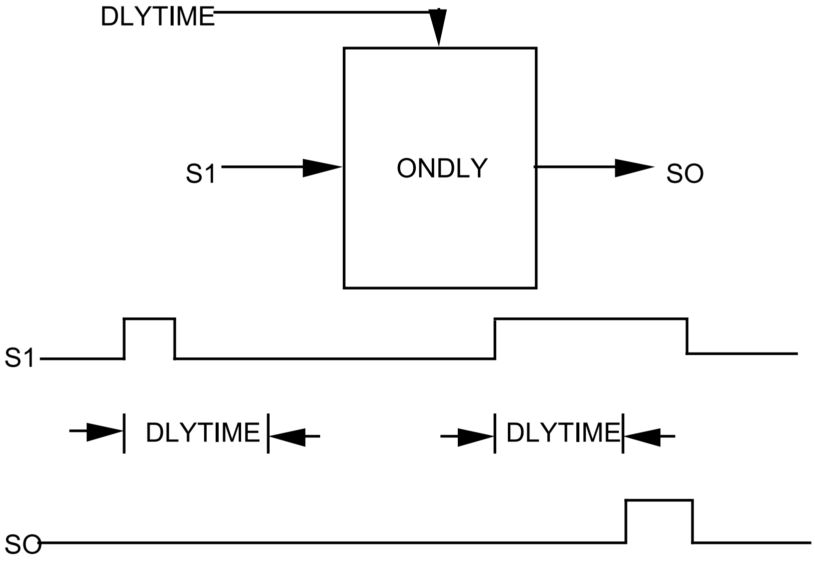
5.3.21 Delay (DELAY)

This logic algorithm delays the input signal at the S1 input by one sample time. The SO output always follows the input after one sample time delay.



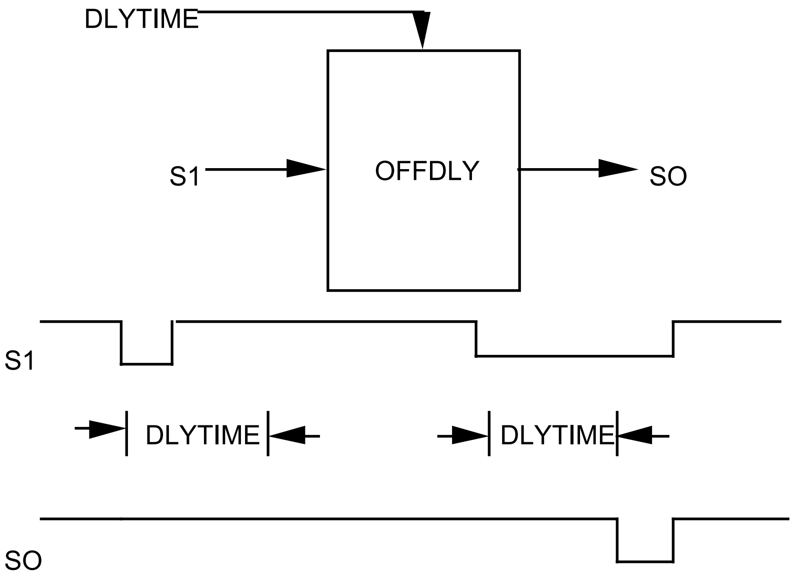
5.3.22 On Delay (ONDLY)

This logic algorithm delays the input signal supplied at the S1 input when the input signal is going from the OFF to the ON state. (There is no delay provided when the input changes from the ON to the OFF state.) When the input state changes from OFF to ON, an internal timer starts counting down the delay time specified by the DLYTIME parameter (in seconds). When it times out, the S1 input is monitored again, and if it is still ON, the SO output is set to ON. When the input state transitions to OFF, the SO output is set to OFF immediately, and the timer is shut off (if it is running). If the specified delay time is less than or equal to one sample time (of the logic point), it is assumed to be equal to one sample time.



5.3.23 Off Delay (OFFDLY)

This logic algorithm delays the input signal supplied at the S1 input when the input signal is going from the ON to the OFF state. (There is no delay provided when the input changes from the OFF to the ON state.) When the input state changes from ON to OFF, an internal timer starts counting down the delay time specified by the DLYTIME parameter (in seconds). When it times out, the S1 input is monitored again, and if it is still OFF, the SO output is set to OFF. When the input state transitions to ON, the SO output is set to ON immediately, and the timer is shut off (if it is running). If the specified delay is less than or equal to one sample time (of the logic point), it is assumed to be equal to one sample time.

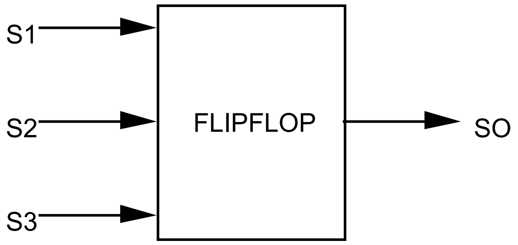


5.3.24 Watchdog Timer (WATCHDOG)

This logic algorithm provides a "time out" capability to monitor other system functions or remote devices. The function or device monitored must set the watchdog reset flag FL(6) (of this logic point) to ON within a time interval specified (in seconds) by the DLYTIME parameter, otherwise it is assumed to have failed, and the SO output of the algorithm is set to ON. If the specified delay time is less than or equal to one sample time (of the logic point), it is assumed to be equal to one sample time. When the watchdog timer algorithm runs and if the FL(6) flag is ON, the internal timer is set equal to DLYTIME, and FL(6) and the output SO are both set to OFF; however, if FL(6) is OFF, the internal timer is decremented, and if it becomes zero, the SO output is set to ON. Because this algorithm always uses the FL(6) flag of the logic point as the reset input, only one Watchdog Timer algorithm should be configured for each logic point.

5.3.25 Flip-Flop (FLIPFLOP)

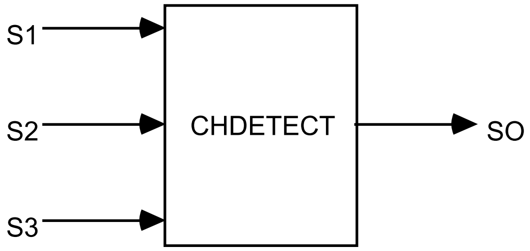
This algorithm provides the flip-flop function. The SO output is determined by the states of inputs S1 and S2 as follows:



S1	S2	SO
OFF	OFF	Not changed
ON	OFF	OFF
OFF	ON	ON
ON	ON	S3 input

5.3.26 Change Detect (CHDETECT)

This algorithm is used to detect changes in up-to-three inputs. The output SO is determined as follows:



```
SO = (S1 < > S1_last_time)
            OR
      (S2 < > S2_last_time)
          OR
      (S3 < > S3_last_time)
```

5.4 LOGIC BLOCK ALARMS

5.4.1 Configurable Alarms

Four custom alarms can be configured for each logic point. The alarm source can be any of the logic inputs (L1–L12), logic flags (FL1–FL12), logic gate outputs (SO1–SO24), or None.

When the alarm source is None, a CL program can force an alarm by writing ON to the alarm flag (C1FL–C4FL) and clear the alarm by writing OFF to the alarm flag.

Each of the four custom alarms permits an 8-character descriptor that appears on page one of the Detail Display, the Alarm Summery Display, and the Real Time Journal, when that alarm is active.

The alarm priority choices are: No Action, Journal, Low, High, Emergency, Journal Print, and Printer.

PROCESS MODULE DATA POINTS
Section 6

This section describes the functions available in the Process Module Data Point and the associated Box Flag, Box Numeric, and Box Timer Data Points. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

6.1 PURPOSE OF THE PROCESS MODULE DATA POINT

Control strategies frequently need the flexibility of user programs for continuous, batch, or hybrid applications. A Process Module Data Point is the resource for execution of user-created programs written in Honeywell's Control Language (CL/APM). CL/APM is a member of Honeywell's family of advanced process-engineer oriented languages and is a powerful tool for advanced continuous control, sequential control, and computations. CL programs are self-documenting, which is an important feature when future modifications of control strategies are anticipated.

Total statement capacity depends on configuration. The maximum number of CL/APM statements = 3(15000 - n) where n is the number of Memory Units used by data points (see subsection 3.1.3). Using the Universal Station or the Universal Work Station, individual programs can be easily modified and reloaded without affecting the execution of regulatory control, logic blocks, and other user programs.
programs.

The Process Module Data Point is the interface between the system and a CL/APM (sequence) program that resides in the Process Module Data Point. The Process Module Data Point shown in Figure 6-1 is the mechanism through which

- An operator, a CL/AM block, a user-written CM program, or another sequence program can start and stop a sequence program and monitor the current status of a sequence program.

- Sequence alarms (sequence errors, failures, and phase alarms) are reported to the system; an alarm priority can be assigned to sequence alarms.

- A sequence program can send information to an Advanced Control Point (ACP) in a Computing Module when implementing high-level control strategies.

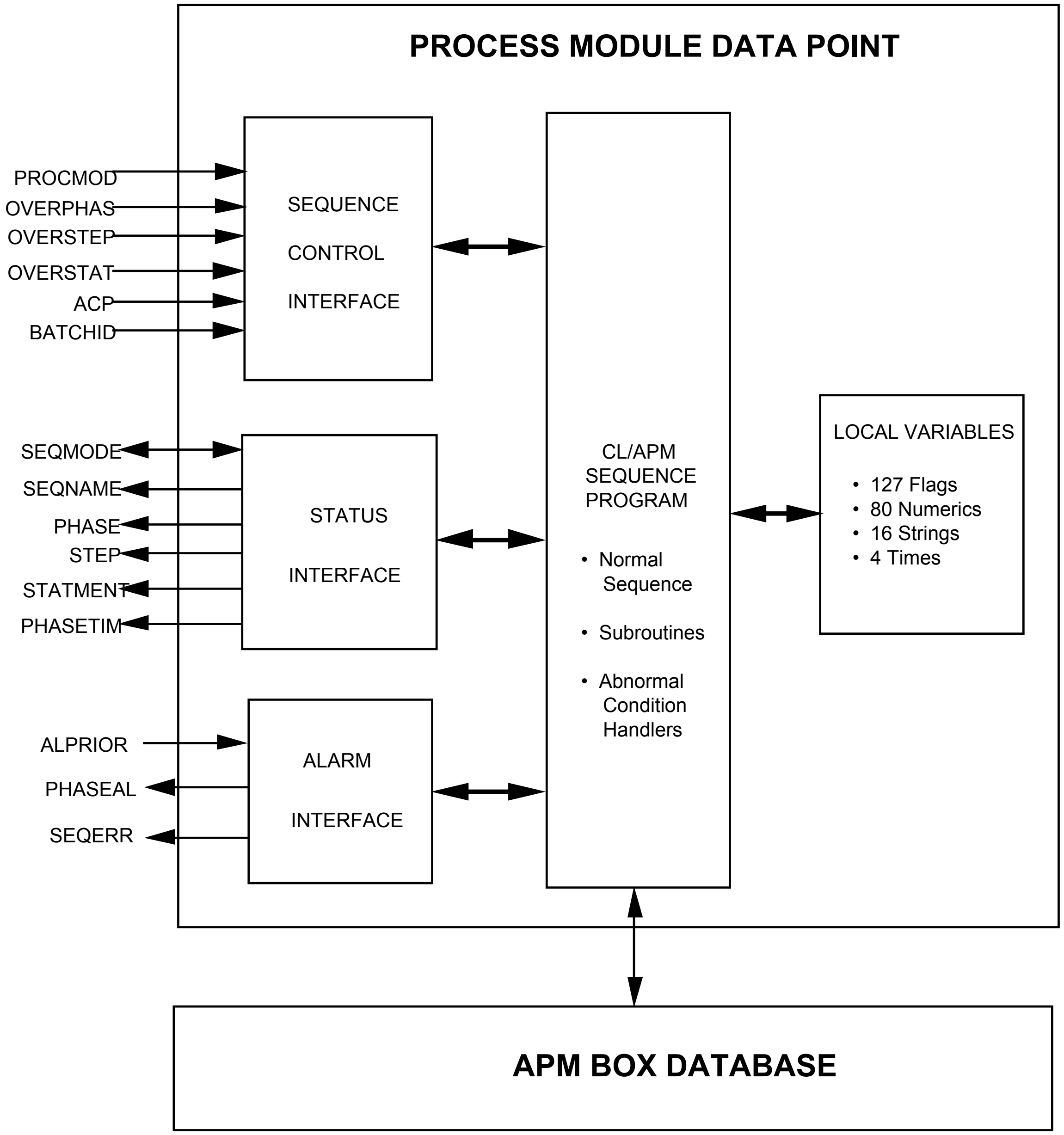


Figure 6-1 — Process Module Data Point, Functional Diagram

11012

6.1.1 CL/APM Sequence Program

To accomplish the desired sequence control, the sequence program can

- Monitor current process values such as temperatures, pressures, and flows and current states of devices such as valves and motors. The sequence program can access values from the local APM database, and *from any other node such as APM, HPM, PM, LM, or SM that is on the same UCN*. A sequence program can read the values of almost all parameters in the APM database except for character-string descriptors such as PTDESC (Point Descriptor), and input/output connections.

- Control the process by using the accessed values to calculate new values and write them to the appropriate parameters. The sequence program can write values into the local APM database and *into any other node such as APM, HPM, PM, LM, or SM that is on the same UCN*. A sequence program can write values to all parameters in the APM database except for calculated variables such as alarm flags and general configuration parameters such as PVALGID (PV Algorithm Identifier).

- Use the 127 local flag variables, the 80 local numeric variables, the 16 8-character string variables and the 4 time variables that reside in each Process Module Data Point. The string variables can alternatively be used as 8 16-character or 4 32-character or 2 64-character strings. It can also use the local flags, numerics, strings and times that reside in other Process Module Data Points. Additional flags, numerics, strings, time, and timer variables reside in the APM Box (see Section 9) and are available to all sequence programs in the same APM. CL programs cannot access APM Box variables with an index of greater than 4,095. Array points can be used to refer to those variables (see Section 10).

- Start another sequence program or force an abnormal condition in another Process Module Data Point as required.

- Issue predefined or computed messages to the operator at the Universal Station to inform the operator of current process conditions and to provide instructions. The operator can also override various portions of the sequence, as required.

NOTE

With R530, the compile time of a loaded CL/APM sequence program is provided. Time and Date stamp information is shown on the Detail Display of the UCN ProcMod point. To access this page, select the target CONFIG PARAMS from the first page of the ProcMod Point Detail Display.

6.1.2 Sequence Program/Unit Partitioning

The user's process is usually partitioned into process units where the operators have the responsibility to control one or more process units. To support this type of partitioning, the APM Process Module Data Points (sequence programs) can be partitioned on a process unit basis as shown in Figure 6-2. This is accomplished by using the following three elements:

- The set of CL/APM sequence programs that control individual process unit.

- The set of process monitoring and control elements that are used by the sequence program to control the operation of the process unit.

- The operating displays at the Universal Station such as the Module Summary Display (see Figure 6-3) which shows the current status of the process modules on a process unit basis, and the Module Group and Module Detail displays which allow the operator to start and stop the sequence programs, override various portions of the sequence programs, etc.

Wherever possible, process unit partitioning should be consistent with the inherent boundaries in the system, such as:

- The beginning and end points of major separable segments of a process.

- The physical separations or classifications of process equipment components.

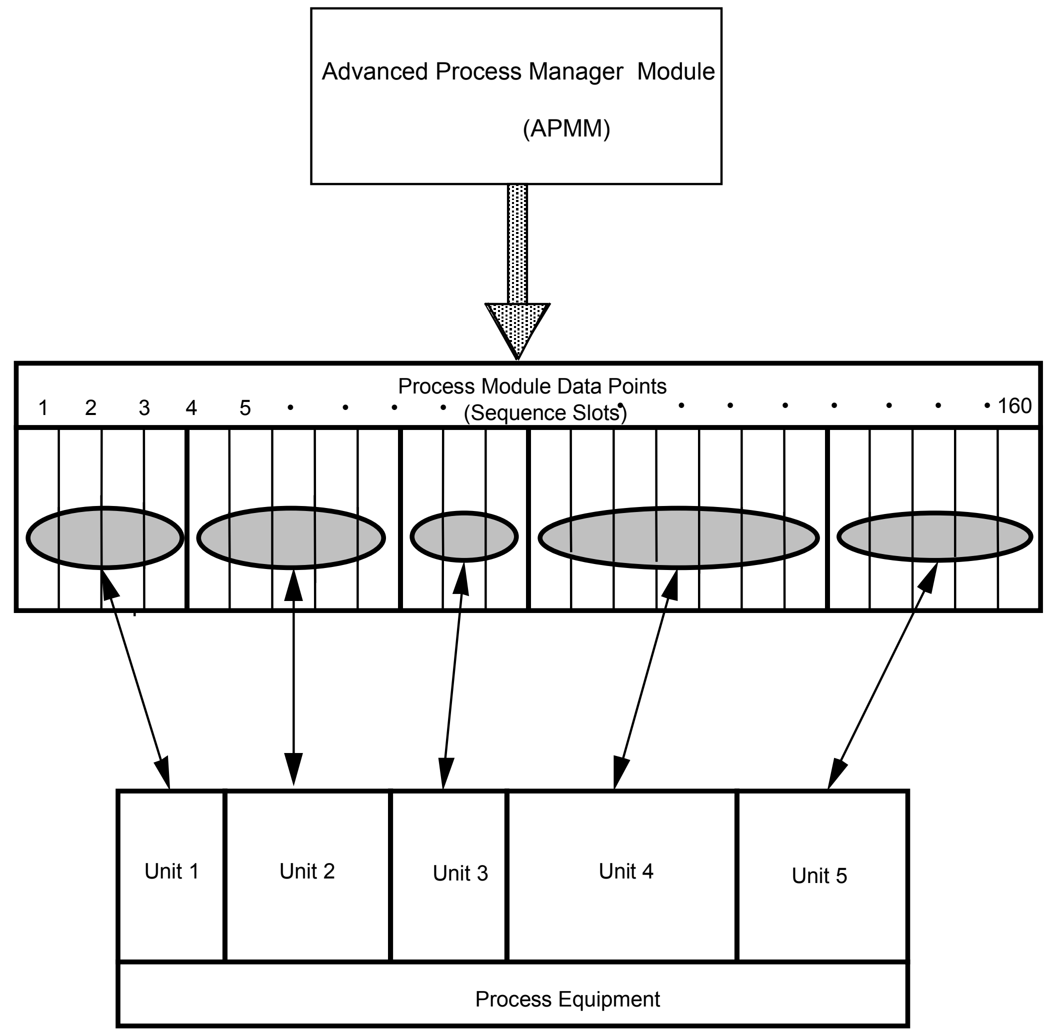
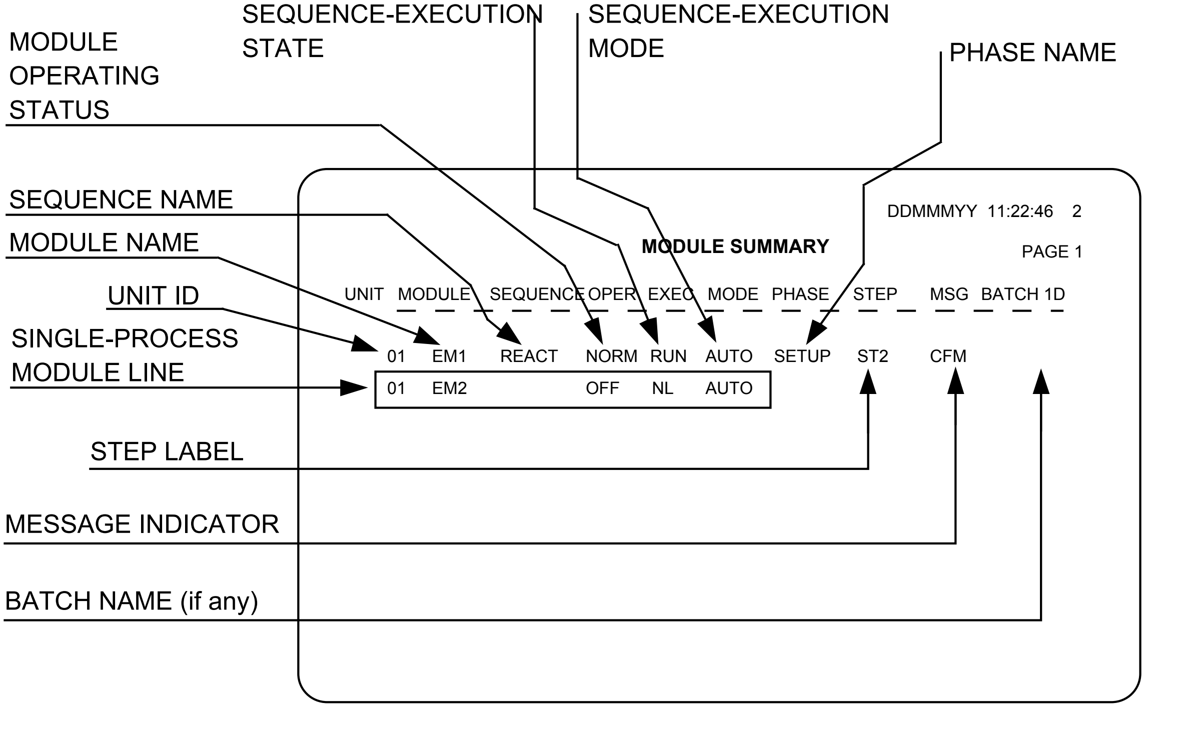


Figure 6-2 — Sequence Program Partitioning/Unit Partitioning

11013



```
                                          DDMMMYY  11:22:46   2
                       MODULE SUMMARY
                                                       PAGE 1
UNIT  MODULE  SEQUENCE OPER  EXEC  MODE  PHASE  STEP  MSG  BATCH 1D
 01    EM1    REACT    NORM  RUN   AUTO  SETUP  ST2   CFM
 01    EM2             OFF   NL    AUTO
```

Figure 6-3 — Module Summary Display

6.1.3 Writing the Sequence Programs

Sequence programs are written in CL/APM and compiled at a Universal Station. At compilation time, each sequence program is bound to a specific Process Module Data Point (i.e., a specific sequence slot) and is written to a separate object file. These files can be directly downline loaded through the NIM to the process module slots in the designated APM.

- To prepare CL/APM programs, refer to the *CL/APM Reference Manual* in the *Implementation/Advanced Process Manager - 2* Binder.

- To compile a CL/APM program (binding it to a process module point) and load it into the APM, refer to the *CL/APM Data Entry* publication in the *Implementation/Advanced Process Manager - 2* binder.

6.1.3.1 Sequence Library

The Sequence Library contains the names of the sequence programs. The Sequence Library is divided into three sections, with each section having a maximum of 1000 entries. (The section is defined by the LIBRYNUM parameter.) A section of the library can be viewed by reconstituting the point, using $N*xx*Lib*y* where *xx* is the UCN number on which the NIM resides and *y* is the section number of the library.

The library entries can be made automatically by the CL Compiler, or they can be entered manually by the user. If the manual entry method is to be used, the user should first enter the names on the NIM Library Configuration Form, AP88-540. The LIBRYTXT parameter defines the allowable character set for the library entries.

Library 1 is reserved for sequence names; Libraries 2 and 3 are reserved for all other names. If the CL Compiler completely fills up Libraries 2 and 3, the compiler will begin entering the overflow information into Library 1, beginning with index number 1,000 and working down towards index number 2.

The sequence library resides in the NIM, and it is applicable to all the APMs (or PMs or HPMs) on the same UCN.

6.1.3.2 Local Flags, Numerics, String and Time Variables

Each Process Module Data Point contains 127 local flags, 80 local numerics, 4 local time variables, and 16 local 8-character strings. The strings can alternatively be allocated as 8 16-character, 4 32-character, or 2 64-character strings. Flags are designated FL(1) to FL(127) and the numerics are designated NN(1) to NN(80). Times are designated TIME(1) to TIME(4) and Strings are designated STRn(mm) where n is the string length (8, 16, 32, 64) and mm is the specific string number. For example, STR8(15) for 8-character strings, STR16(7) for 16-character strings, etc.

The primary purpose of the flags and numerics is to provide storage for recipe data and intermediate results. Time variables are used to record date/time information, and strings contain text strings that name or describe things. These local variables can be accessed by the sequence executing in the same process module in which they reside, or they can be accessed by any other sequence in the same APM. The flags and numerics can also be accessed by all system functions, such as CL programs in the AM, Universal Station displays, etc. They can be accessed as local variables by CL/APM by using the AT clause. Refer to the *CL/APM Reference Manual*.

These local variables are accessed by the various system elements using the "Tagname.Parameter" convention where the Tagname is the name of the process module in which they reside, and the "Parameter" is the parameter name such as FL(15), NN(25), STR8(15), or Time(4). To store (write) into the local flags and numerics, the keylock position must equal or exceed the minimum access level specified in parameter SPLOCK.

Process module flags and numerics are set to the default values of OFF and NaN respectively, when the process module data point is deleted or when the APM database is initialized to default values. Time and string variables are initialized to spaces and 0 seconds, respectively. The APM database is set to default values on initial loading of the APM Box data point and when the SCANRATE parameter or point mix changes in subsequent loads of the APM Box data point. Process Module data point local flags and numerics are left unchanged during downloading of sequence programs or the process module data point itself to allow a record or history to exist between programs.

Flag, Numeric, String, Times, and Timer variables that reside in the APM box are a different set of variables from the local flags, numerics, strings, and times. These variables are available to the sequence programs in this APM and to the other sequence programs in the APMs (or PMs or HPMs) on the same UCN and, with some minor restrictions, to other system functions. They are described in Section 9 of this manual.

6.1.4 Sequence Program Structure

As shown in Figure 6-4, each sequence program consists of two main sections: the *data declaration section* and the *execution section.*

The *data declaration section* requires the user to provide optional CL/APM statements that relate user-defined names to APM flags, numerics, strings, and times that declare the numeric constants, and that name the data points used in the program.

The *execution section* consists of one-to-three parts:

- A *normal sequence* that is the main program. The normal sequence is the main program that provides control when conditions in the process are normal (e.g., the process is proceeding within acceptable limits and does not require special control actions). The main program is divided into "Phases," "Steps," and "Statements" as shown in Figure 6-4.

  Phase—A sequence program consists of a grouping of phases. A phase executes a major process function or marks a major process milestone, such as a charging phase or heat-up phase. Phase boundaries are key points of synchronization in the control program. A phase is identified in the sequence program by the PHASE statement, and consists of optional steps and statements.

  Step—A step executes a minor process function such as opening a valve with associated checks and verification, or checking a temperature. At least one step is executed each time the sequence is executed. A step is visible at a Universal Station as a process milestone. It is identified in the sequence program by a STEP statement, and consists of an ordered set of CL/APM statements. The step size is limited to a maximum of 255 statements.

  Statement—One CL/APM instruction. A statement performs an elementary action, such as commanding a valve to open, do arithmetic operations, fetch variables, etc.

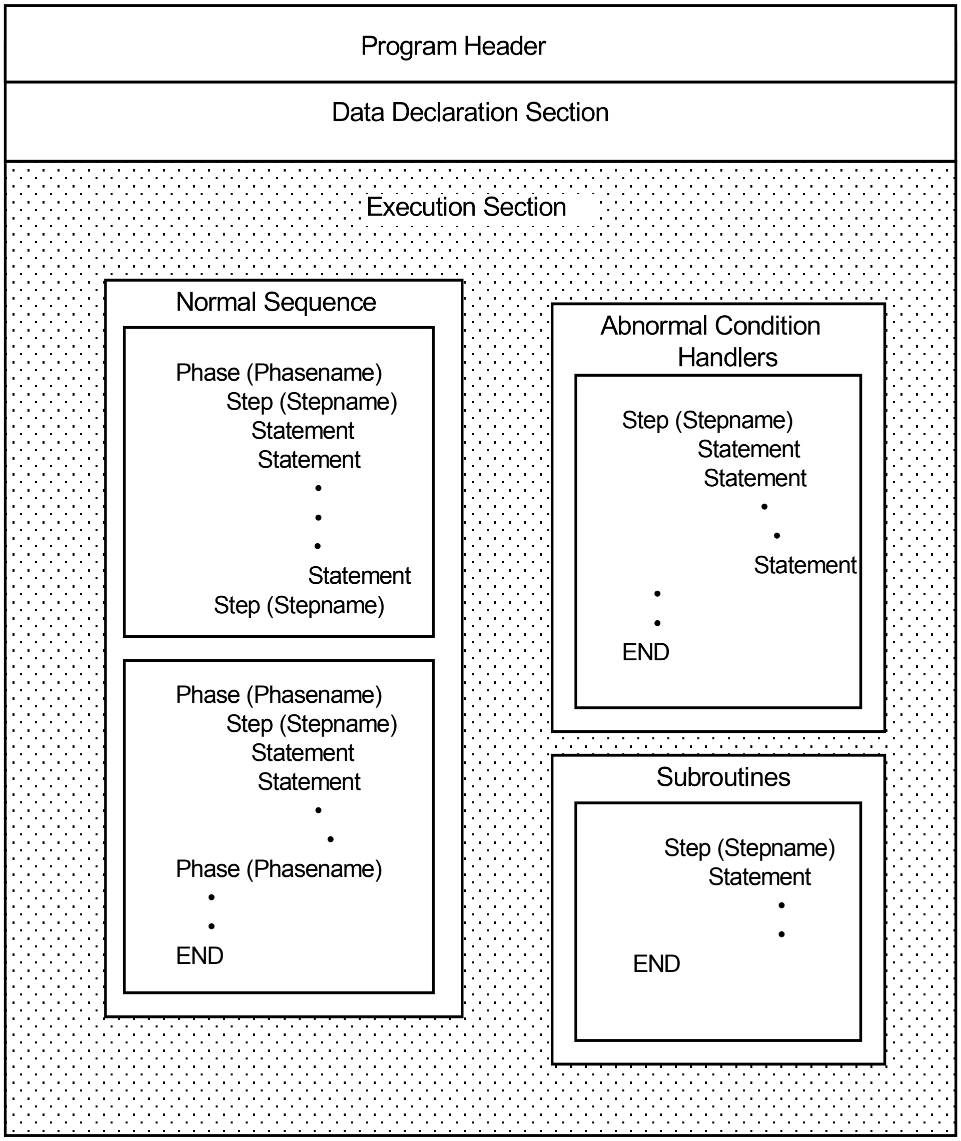


Figure 6-4 — Sequence Program Structure 2549

- *Subroutines*—Subroutines simplify the program structure and are used for executing repetitive functions, thereby reducing the programming effort and memory requirements. Subroutines can be called by the normal sequence, other subroutines, or abnormal condition handlers. Two levels of nesting are supported for both normal and abnormal sequences. A runtime error is generated if the nesting level is violated.

  Subroutines are specific to one sequence program, i.e., subroutines defined outside of the program and maintained in subroutine libraries are not supported in the Advanced Process Manager.

- *Abnormal-Condition Handlers*—Abnormal-condition handlers provide corrective action when abnormal conditions are encountered. The conditions can be detected when the SEQEXEC value is Run, Pause, Fail, Error, or End. When such a condition is detected, the normal sequence is suspended and the appropriate abnormal-condition handler begins to execute. The types of abnormal-condition handlers are as follows:

  - *Hold*—For a condition that requires a partial shutdown.

  - *Shutdown*—For a condition that requires a systematic shutdown.

  - *Emergency Shutdown*—For a condition that requires a complete, sudden shutdown.

  Abnormal-condition handlers are identified in the sequence program by the HANDLER statement, preceded by the name for the type of handler, and followed by the user-given handler name, such as in the following example:

  ```
  HOLD HANDLER cooldown
  ```

  Abnormal-condition handlers consist of steps and statements. Hold and Shutdown abnormal-condition handlers can be terminated by a restart section that is identified by the RESTART statement, followed by the handler's END statement. An Emergency Shutdown handler can be terminated by only the handler's END statement. Abnormal-condition handlers have priority over each other and over the normal sequences. The priorities are as follows:

Highest Priority –	Emergency Shutdown
	Shutdown
	Hold
Lowest Priority –	Normal Sequence

  Note that the priority of a Restart Section is the same as the abnormal-condition handler that contains the Restart Section.

  The abnormal condition handlers can be enabled or disabled anywhere within a Phase by using the ENB statement, which causes a suspend condition. The new conditions to be monitored take effect immediately. The abnormal condition can also be enabled or disabled by the phase header in the phase statement.

6.1.5 Sequence Execution

6.1.5.1 Normal Execution

Execution of the normal sequence can be initiated by one of the following:

- Universal Station operator from the Process Module Detail Display.
- Another sequence, executing in another process module by using the INITIATE statement. Generally, this other sequence is a master or supervisory sequence; that is, a sequence program that initiates, monitors, and controls other sequences.
- CL Block in an AM which changes the PROCMOD parameter value in the APM to Strt (Start).
- User-written program in a CM which changes the PROCMOD parameter value in the APM to Strt (Start).

To manipulate the PROCMOD parameter successfully, the value of PROCMOD must be Off, and the value of SEQEXEC must be Loaded. The keylock position must equal or exceed the minimum access level specified in parameter CTLLOCK. The value of PROCMOD can be changed to START from any keylock position regardless of the access level specified in CTLLOCK.

6.1.5.2 Execution of Abnormal-Condition Handlers

Enabled abnormal-condition handlers can be initiated by one of the following:

- Request from a Universal Station operator.
- Predefined process condition detected by the sequence program at every pre-emption point, such as before executing a step.
- Activation by a statement in the sequence (e.g., INITIATE HOLD).
- INITIATE statement from another sequence.
- CL block in an AM.
- User-written program in a CM.

6.1.6 Process Module Operating State

The process module operational state represents the operational condition of a process module. The module state is represented through the PROCMOD parameter and the allowable states are listed and described in Table 6-1. A state diagram is provided in Figure 6-5.

6.1.7 Sequence Execution Mode

The sequence execution mode defines the manner in which the sequence program is executed. The sequence execution mode is established through the SEQMODE parameter, and the allowable modes are listed and described in Table 6-2.

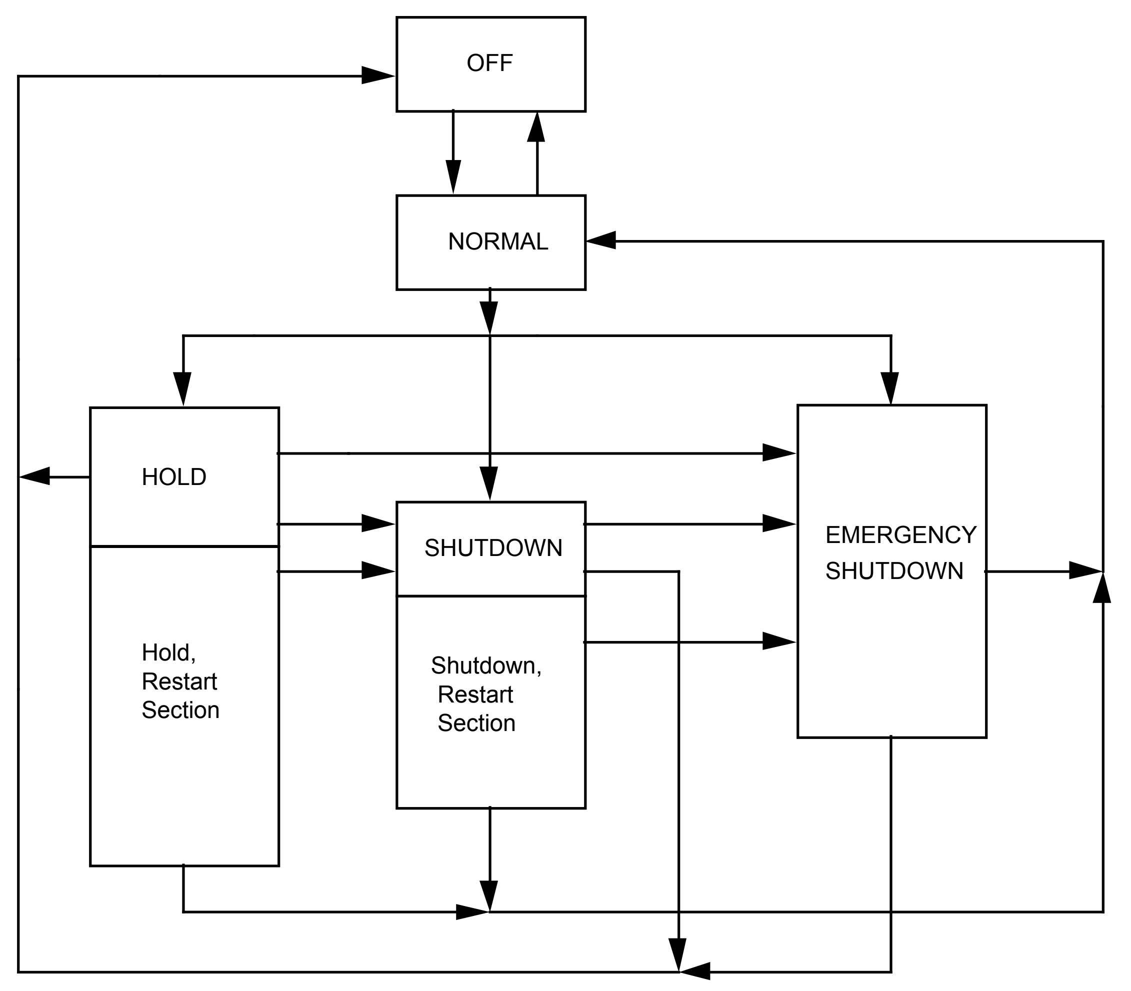


Figure 6-5 — Process Module Operating States

Table 6-1 — Process Module Operating States*

State	Description
Off	Process module slot is not being processed. A sequence program can be assigned to a process module but it cannot be executed. The sequence execution mode is not applicable to this module state.
Norm (Normal)	The normal sequence program is running in the process module slot.
Hold	Process module is executing the hold abnormal-condition handler. Hold sequence may be initiated by one of the items listed in paragraph 6.1.5.2.
Shdn (Shutdown)	Process module is executing the Shutdown abnormal-condition handler. Shutdown sequence can be initiated by one of the items listed in paragraph 6.1.5.2. Shutdown has priority over Hold.
Emsd (Emergency Shutdown)	Process module is executing the Emergency Shutdown abnormal-condition handler. Emergency Shutdown sequence can be initiated by one of the items listed in paragraph 6.1.5.2.

* Parameter PROCMOD also contains the Strt (Start) and Stop states, which are transitional states that can be controlled by one of the items listed in paragraph 6.1.5.2.

Table 6-2 — Sequence Execution Modes

Execution Mode	Description
Auto (Automatic)	Normal mode of operation. Sequence runs from beginning to end without operator intervention.
SemiAuto (Semi-Automatic)	Sequence stops at all PAUSE statements in the sequence. Sequence can be restarted by the operator.
SnglStep (Single-Step)	Normally used for debugging. Sequence is executed one step at a time. Sequence can be restarted by the operator.

6.1.8 Sequence Execution State

The sequence execution state represents the current state of the sequence program. These states are defined through parameter SEQEXEC and are listed and described in Table 6-3. Figure 6-6 shows the possible state transitions of the sequence.

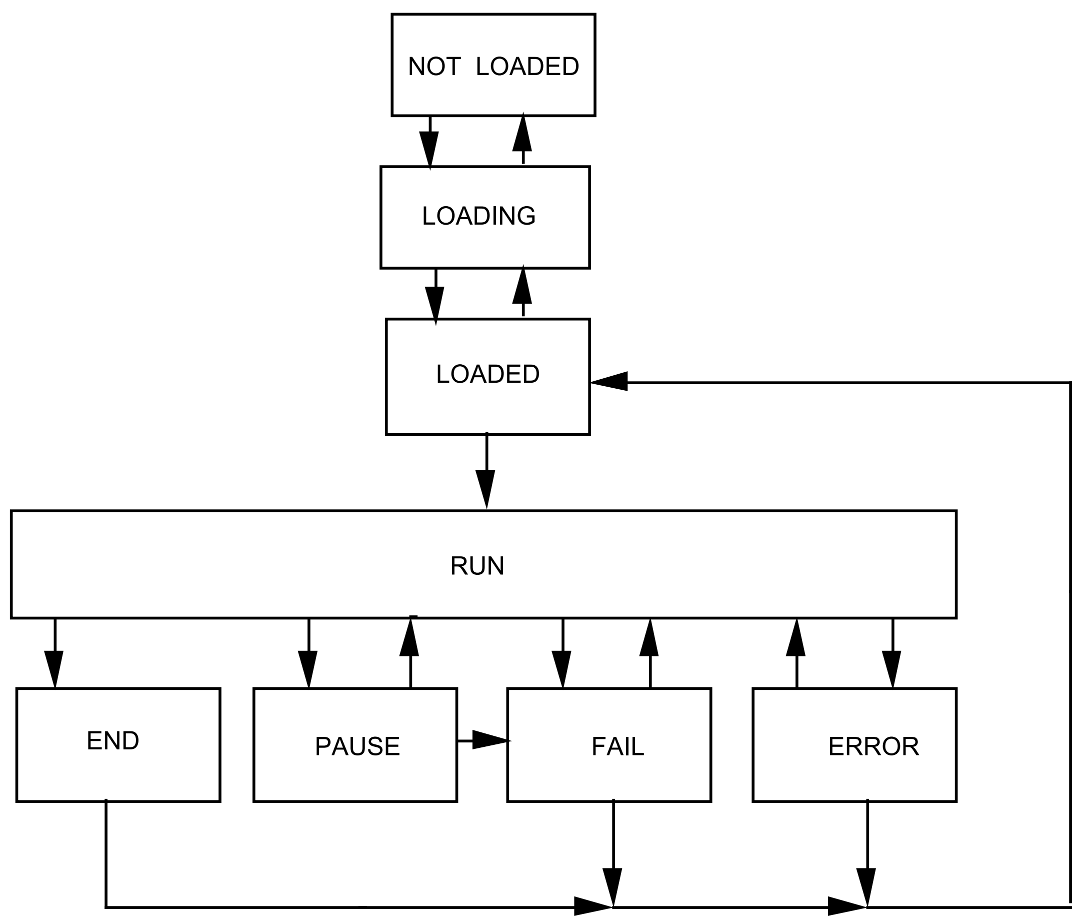


Figure 6-6 — Sequence Execution States

11022

Table 6-3 — Sequence Execution States

<table>
<tr><th>Execution State</th><th>Description</th></tr>
<tr><td>NL (Not Loaded)</td><td>Initial state in which sequence has not been assigned to a process module. This state is automatically entered from the DLL state when loading has been aborted.</td></tr>
<tr><td>DLL (Down-Line Loading)</td><td>Transient state during which the loading of the sequence into the process module slot is in progress.</td></tr>
<tr><td>Loaded</td><td>Sequence program has been loaded into the process module slot. This state is automatically entered from the DLL state when loading has been completed.</td></tr>
<tr><td>Run</td><td>Sequence program is running.</td></tr>
<tr><td>Pause</td><td>Execution of the sequence is suspended because of the execution of a PAUSE statement, or the completion of a step when in the single-step mode. Sequence execution can be resumed by the operator.</td></tr>
<tr><td>Fail</td><td>Sequence execution is suspended because of the detection of one of the failure conditions listed below. A system message that describes the failure condition is generated. Sequence execution can be resumed by the operator.
<table>
<tr><th><em>Failure Code</em></th><th><em>Definition</em></th></tr>
<tr><td>165</td><td>Sequence has been stopped by the operator.</td></tr>
<tr><td>166</td><td>Attempt to start an abnormal-condition handler that has not been enabled and no lower-priority abnormal-condition-handler is enabled. Possibly using INITIATE.
Suggestions: Use INITIATE with a When Error clause.</td></tr>
<tr><td>167</td><td>Not used.</td></tr>
<tr><td>168</td><td>Not used.</td></tr>
<tr><td>169</td><td>An attempt was made to start a sequence that was not loaded into the process module slot.</td></tr>
<tr><td>170</td><td>Communication error detected in READ/WRITE statement. Possible cause: a READ/WRITE statement without a When Error Path fails a bad read. Note that a post store reports its own failure code.
Suggestions:
• determine validity of the attempted fetch or store.
• check actual presence of point.</td></tr>
</table>
</td></tr>
</table>

(Continued)

Table 6-3 — Sequence Execution States (Continued)

Execution State	Description
Fail (continued)	
	171 Communication error detected during an I/O Link access. This error is generated for all post-store problems. Possible cause: • IO Link poststore failed. • DI scan PV status is bad. Suggestions: • check I/O Link for reasons for bad post-store, or bad PV in DI scan.
	172 Range error has been detected. Possible cause: • box timer set point value greater than 32000. • illegal stores to Digital Composite parameters.
	173 Store failure because of "rights" error. For example, an attempt was made to write to a parameter when the point was not in the proper mode. Possible cause: • Other points reject stores. • attempt to write to box timer set-point or time base when state is in RUN. • attempt to start timer when PV is greater than SP value. • attempt to INITIATE a lower priority handler, while in a higher priority handler. Suggestion: check state of point not accepting the store.
	174 An interlock error condition has been detected. (Not used).
	175 –255 Reserved for future use.
Error	Sequence execution is suspended because of the detection of one of the error conditions listed below. A system message that describes the error condition is generated. Sequence execution can be resumed by the operator .
	Error Code *Definition*
	101 Not used
	102 Array index error Possible cause: violated array index bound check. Suggestions: check array bounds defined in the LOCAL statement or in the subroutine header.

Table 6-3 — Sequence Execution States (Continued)

Execution State	Description	
Error (Continued)	*Error Code*	*Definition*
	103	Illegal generated code Possible cause: • could not find variable operator code in the reference list • invalid statement header • unsupported built-in function call. Suggestions: contact Honeywell Technical Assistance Center.
	104	Illegal generated code referencing a parameter or variable. Possible cause: • bad box timer parameter code • local numeric/flag index out of range • invalid time base. Suggestions: contact Honeywell Technical Assistance Center.
	105	Interpreter stack overflow Possible cause: Not enough stack depth. Suggestions: contact Honeywell Technical Assistance Center.
	106	GOTO destination error Possible cause: Invalid destination. Suggestions: contact Honeywell Technical Assistance Center.
	107	Key-level error Possible cause: • tried store to read only parameter • tried store with access level invalid • system error Suggestions: contact Honeywell Technical Assistance Center.
	108	Configuration mismatch error Possible cause: • may be system error Suggestions: contact Honeywell Technical Assistance Center.
	109	I/O Link prefetch overflow Possible cause: exceeded 12 I/O Link prefetches per step. Suggestions: insert a new STEP in the code if several IOL references are made within this step.
	110	Subroutine nesting level error Possible cause: nesting of subroutine calls is more than two levels deep. Suggestions: reduce nesting level of subroutine call to no more than two. Note that a subroutine called from a normal or abnormal program can call only one more level of subroutine.
	111	Illegal value error Possible cause: illegal value detected when doing READ/WRITE or SET. Suggestions: contact Honeywell Technical Assistance Center.

Table 6-3 — Sequence Execution States (Continued)

Execution State	Description	
Error (Continued)	*Error Code*	*Definition*
	112	FAIL statement was executed by the sequence Suggestions: • restart. • make necessary changes, then resume execution at the next sequence statement.
	113	I/O Link prefetch buffers full Possible cause: IOL prefetch buffers full. Tried to fetch IOL parameters eight times in a row, with preemption each time but was unsuccessful. Suggestions: • redistribute I/O Link fetches to another programming cycle. • Insert WAITS to slow collection down. • break into additional STEPS.
	114	I/O Link poststore buffers full. Possible cause: I/O Link poststore buffers full. Tried to store IOL parameters eight times in a row, with preemption each time butwas unsuccessful. Suggestions: • redistribute I/O Link stores to another programming cycle • Insert WAITS to slow collection down. • break into additional STEPS.
	115	UCN prefetch buffers full Possible cause: I/O Link prefetch buffers full. Tried to fetch UCN parameters sixteen times in a row, with preemption each time but was unsuccessful. Suggestions: • redistribute reads to another programming cycle. • Insert WAITS to slow collection down. • break into additional STEPS.
	116	UCN poststore buffers full. Possible cause: I/O Link poststore buffers full. Tried to store. UCN parameters sixteen times in a row, with preemption each. time but was unsuccessful. Suggestions: • redistribute writes to another programming cycle. • Insert WAITS to slow collection down. • break into additional STEPS.
	117 - 164	Reserved
End	Sequence execution has been completed as indicated by the execution of an END statement. This state is entered at the completion of a normal sequence or an abnormal condition handler (Hold, Shutdown, or Emergency Shutdown) sequence.	

6.1.9 Sequence Overrides

The process module displays at the Universal Station allow the user to perform phase, step, and statement overrides when the sequence execution state is Fail, Error, or Pause. An attempt to access the override function from anywhere other than the Universal Station results in an error code of 107 (key-level error). The overrides are not restricted by the value in CNTLOCK if the sequence execution state is Fail or Error.

- **Phase Override**—Allows the user to skip execution of one phase after another, in forward or reverse order. Phase override works only in the sequence for which it is invoked; it is implemented through the OVERPHAS parameter.

- **Step Override**—Allows the user to skip execution of one step after another, in forward or reverse order. Step override works only in the phase for which it is invoked; it is implemented through the OVERSTEP parameter.

- **Statement Override**—Allows user at a Universal Station to skip execution of one statement after another, in forward or reverse order. Statement override works only in the step for which it is invoked; it is implemented through the OVERSTAT parameter.

6.1.10 Sequence Alarms

Sequence alarms are generated when

- The process-module operational state is changed to Hold (HOLD), Shutdown (SHDN), or Emergency Shutdown (EMSD) as indicated by a change in the PROCMOD parameter. The PROCMOD parameter can be changed by entry into an abnormal condition handler or externally by operator, AM, etc. The PROCMOD parameter has a key level of Operator and can be written to from the LCN (by an operator, a CL/AM program, etc.), or by a CL/APM program.

- The sequence execution state is changed to Fail, Error, or End as indicated by a change in the SEQEXEC parameter. The SEQEXEC parameter has a key level of View only and cannot be changed from the LCN or by a CL/APM program.

- The phase timer has elapsed as determined by the PHASETIM parameter.

6.1.10.1 Sequence Alarm Priorities

The priority of the sequence alarm can be configured for each process module through the SEQPR parameter. The entry for this parameter can be Emergency, High, Low, JnlPrint, Printer, Journal, or NoAction.

6.1.11 Sequence Messages

6.1.11. 1 Sequence Status Messages

Sequence-status messages are issued when the:

- Process-module operational state is changed
- Sequence-execution mode is changed
- Sequence-execution state is changed
- Sequence begins a new phase.
- A sequence message is issued.

6.1.11.2 Operator Messages

Two types of programmed operator messages can be generated by the sequence program as listed below.

- **Message with feedback**—A SEND statement with the confirmation option (WAIT) causes the sequence to be suspended until the message has been confirmed by the user at the Universal Station.
- **Message without feedback**—This message is a 1-way communication from the sequence to a destination, such as the Universal Station.

These messages can be displayed at the Universal Station and/or logged on the printer, depending on how the user codes the message statement.

It is good practice to confirm any outstanding messages after a sequence failure before rerunning the sequence and before loading a new sequence.

6.1.12 Restart Option

The restart option determines how the sequence program is started following a warm restart. A Warm restart is considered as being a transition from Idle to Run, or a short power interruption that does not change the APM database. The following restart options are available:

- **Restart**—When the sequence-execution state is Run, the sequence program is started from the very beginning; otherwise, the process module operational state is changed to Off and the sequence-execution state is changed to Loaded (or Not Loaded if a sequence has not been loaded). If an abnormal handler is executing when the warm start occurs, execution begins from the beginning of the normal sequence and not from the abnormal handler. A sequence-state change event is issued to report the transition from Hold, Shutdown, or Emergency Shutdown to Normal.

- **Off**—The process module operational state is changed to Off and the sequence-execution state is changed to Loaded (or Not Loaded if a sequence has not been loaded).

REGULATORY PV POINT
Section 7

This section describes the functions available in the Regulatory PV (RegPV) point. The functions are described first and are followed by detailed descriptions of the algorithms. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

7.1 FEATURES

While standard I/O functions such as engineering unit conversion and alarming are handled directly by the I/O Processors, Regulatory PV (RegPV) points provide an easy-to-use configurable approach for implementing PV calculations and compensation functions. PV processing provides a menu of selectable algorithms such as mass flow, totalization, and variable dead-time compensation.

The RegPV data point supports the following PV algorithms:

Data Acquisition (DataAcq)

Flow Compensation (FlowComp)

Middle of 3 Selector (Mid0f3)

High Low Average Selector (HiLoAvg)

Summer (Summer)

Variable Dead Time with Lead Lag (VDTLDLag)

Totalizer (Totalizr)

General Linearization (GenLin)

Calculator (Calcultr)

Detailed descriptions of these algorithms can be found in this section beginning with paragraph 7.7.

Figure 7-1 is a functional diagram of the RegPV point.

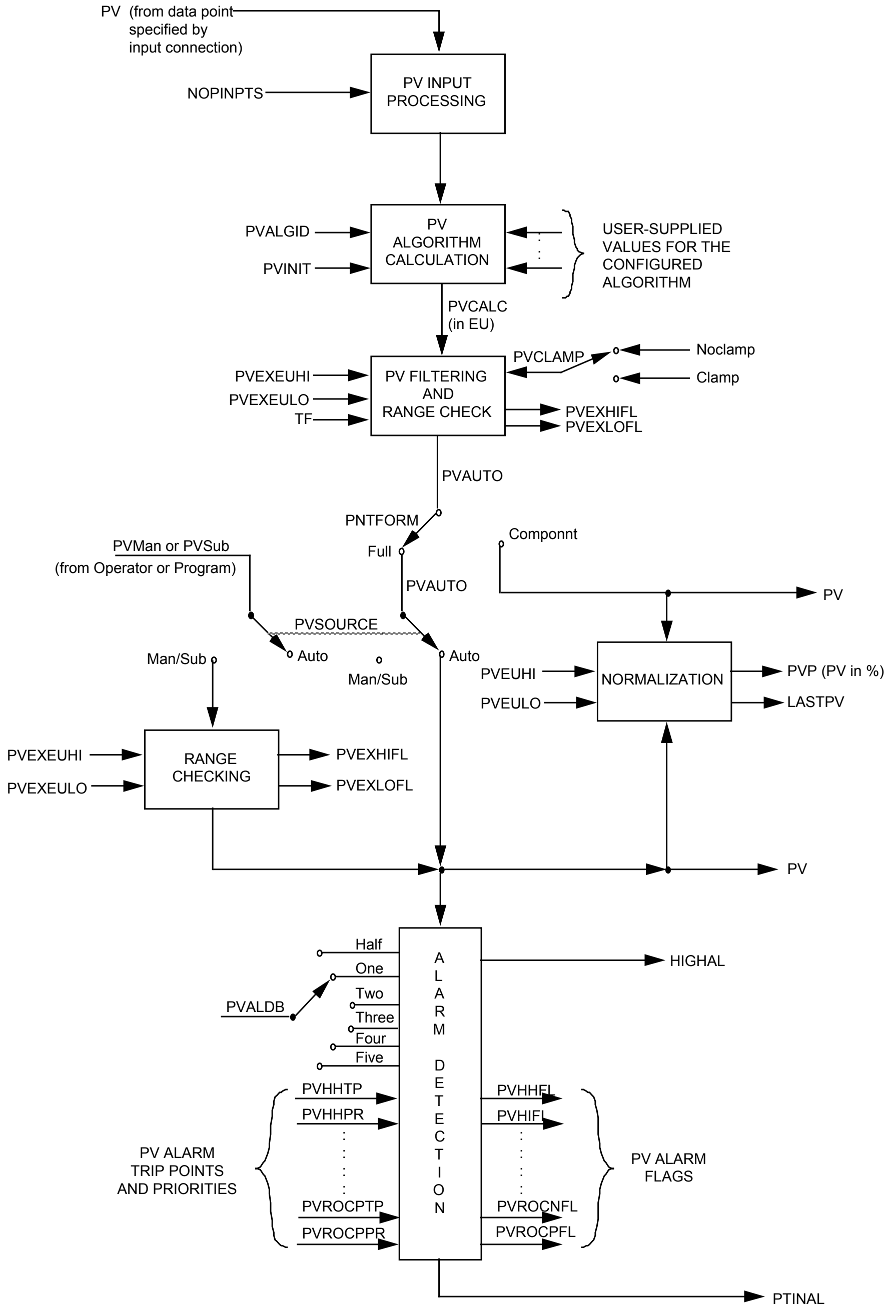


Fig. 7-1
UC09-200
#2080

Figure 7-1 — Regulatory PV Point, Functional Diagram

2080

7.2 PV INPUT CONNECTIONS

PV input connections are used to specify the source(s) for the inputs to PV algorithms. A maximum of six PV input connections can be configured for an algorithm, as required by the individual PV algorithm. The user can implement the input connections by using the "Tagname.Parameter" format.

When using the "Tagname.Parameter" format, the user must enter the tag name of the data point and the name of the parameter within that data point. The source parameter must contain a real number, integer, or Boolean quantity. The values provided by the input connection parameters are assigned to destination parameters within the RegPV point by the PIDSTN parameter. The destination parameters for these sources default to one for each input to the configured PV algorithm.

One input connection must be configured for each algorithm input, otherwise the point cannot be made active.

7.3 PV ALGORITHM CALCULATION

The configured PV algorithm accepts the values received through the PV input connections and produces the calculated PV value (PVCALC) plus its value status (PVAUTOST). The PV algorithms are described beginning with paragraph 7.7.

7.4 PV RANGE CHECK AND FILTERING

The PV range is configured in PVEULO and PVEUHI in a range from 0% to 100.0% of the engineering-units range. This is usually the normal operating range for the PV but it can extend into a configurable extended range, as defined by PVEXEULO and PVEXEUHI. The extended range is forced to be equal to or greater than the range defined by PVEULO and PVEUHI. PVCALC is generally constrained within PVEXEULO and PVEXEUHI, but the checks and constraints depend on the PV source, as follows:

- If **PVSOURCE** is **Auto**—(1) If PVAUTOST contains Bad, PVSTS is Bad and no PV-range checks are made. (2) If PVAUTO is outside the extended range and PV clamping (PVCLAMP) is not configured, the PV becomes a bad value and PVSTS is Bad; however, if PV clamping is configured, the PV becomes equal to the value of the violated range and PVSTS becomes Uncertn. In either case, the appropriate PV-range violation flag is set (PVEXLOFL or PVEXHIFL).
- If **PVSOURCE** is **Man**—Any value from the Universal Station that is outside the extended range is not accepted. PVSTS is already uncertain because the source is Man.
- If **PVSOURCE** contains **Sub**—If a program stores a PV value outside the extended range, the appropriate PVEXLOFL or PVEXHIFL flag is set. PVSTS is already uncertain because the source is Sub.

If clamping of the output is specified by the PVCLAMP parameter, PV filtering is performed before the range check. If the value entered for PV filtering through the TF parameter is other than the default value of 0.0, a single-lag filter is applied to the PVCALC value to remove noise. If a filter value is not specified, PVAUTO contains the same value as PVCALC.

Data-point parameter LASTPV always holds the last good value of the PV.

7.4.1 PV Value Status

The value in the PV value-status parameter, PVSTS, is determined as follows:

Normal PVSOURCE = Auto, PVAUTOST = Normal, and the PV value is within the range defined by PVEULO and PVEUHI.

Uncertn 1) PVSOURCE = Man or Sub and the PV value does not equal NaN (i.e., is a valid, real number), or,

2) PVSOURCE = Auto, and PVAUTOST = Uncertn. Note that PVAUTOST contains Uncertn if at least one of the required algorithm inputs is Uncertn and none of the required algorithm inputs is Bad, or,

3) PVSOURCE = Auto and the value in PVAUTO is outside the engineering-units range and is clamped.

Bad The PV value is NaN. This results from one of the following;

1) PVSOURCE = Auto and PVAUTO = NaN.

2) PVSOURCE = Auto, the value in PVAUTO is out-of-range, and has not been clamped.

3) PVSOURCE = Sub or Man and the PV is stored as NaN.

7.5 PV SOURCE SELECTION

The source of the PV can be PV processing, a Universal Station, or a user-written program, as shown in Figure 7-1. It is specified by the PVSOURCE parameter, whose value can be changed by an operator, a supervisor, an engineer, or a user-written program. The PVSOURCE parameter is configurable only if Full has been entered for the PNTFORM parameter. PVSOURCE can have one of the values:

- **Auto**—PV is received from PV processing through the PV filtering function. The value is in PVAUTO and its status is contained in PVAUTOST. During normal operation, the PV source is Auto, and the PV and its value status (PVSTS) become equal to PVAUTO and PVAUTOST, respectively, before PV range checks are made. When the PV source is changed from Auto to Man or Sub, the PV remains at the last value until it is changed by the operator (Man) or a program (Sub), so it doesn't "bump." In Man or Sub, the status in PVSTS is Uncertn. When the PV source is changed from Man or Sub to Auto, the PV immediately goes to the PVAUTO value. This might cause a bump in the value unless it is changed gradually to the value in PVAUTO before changing the source.
- **Man**—The PV is entered by an operator, supervisor, or engineer at a Universal Station.
- **Sub**—The PV is entered by a user-written program. A program can store a bad value in PV, and if it does, PVSTS goes Bad.

You can prevent PV source changes by configuring OnlyAuto in PVSRCOPT. This fixes the source as Auto and the parameter PVSOURCE is removed from the point. Configuring All in PVSRCOPT allows normal PV source selection.

7.6 PV ALARM DETECTION

The PV alarming function can be implemented only when Full has been entered for the PNTFORM parameter. The following types of alarms are detected during PV alarm processing:

PV High/Low
PV High-High/Low-Low
PV Rate-of-Change Positive/Negative
PV Significant Change
BadPV

PV source selection has no effect on alarm processing. For more detailed alarm information, refer to paragraph 4.3 in *System Control Functions*.

7.7 PV ALGORITHMS

The PV Algorithms in the APM are described in the sections that follow. Each of the PV algorithm descriptions has the same form and the same headings:

TYPE AND NAME

FUNCTION

USE

OPTIONS, EQUATIONS, AND SPECIAL FEATURES

Each of the algorithm descriptions mentions several parameters associated with the algorithm. The parameter names consist of CAPITAL letters. References to parameters not named in the descriptions are provided after the descriptions. Further information on all data-point parameters, including the data type, range, and access keys, is provided in the *Advanced Process Manager Parameter Reference Dictionary.*

7.7.1 Data Acquisition (DATAACQ)

7.7.1.1 Function

This algorithm normally accepts the input and places it, unchanged, in PVCALC. All of the other PV algorithms alter the input(s) in some way. See Figure 7-2.

7.7.1.2 Use

The most common use of this algorithm is to provide a PV that has been through PV Input Processing, PV Algorithm Processing, PV Filtering, and PV Source Selection (see Figure 7-1). The value in PVCALC is filtered, and becomes PV, if the PV source is Auto.

The input can be a measured process-variable, or the calculated PV or calculated output of another data point.

The input to this algorithm and its output are in engineering units.

7.7.1.3 Options and Special Features

This algorithm has neither options nor special features.

7.7.1.4 Equation

This algorithm has one equation form. The operation is simply the replacement of the data point's calculated PV (PVCALC) with the value of the input:

PVCALC = P1

Where P1 contains the first input value and PVCALC contains the value that becomes the PV when PVSOURCE = Auto.

The parameters associated with with this algorithm are P1, PVCALC, and P1STS. Refer to the *Advanced Process Manager Parameter Reference Dictionary.*

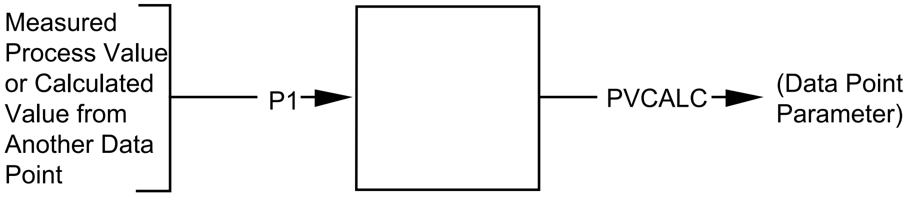


Figure 7-2 — Functional Diagram, Data Acquisition PV Algorithm

7.7.2 Flow Compensation (FLOWCOMP)

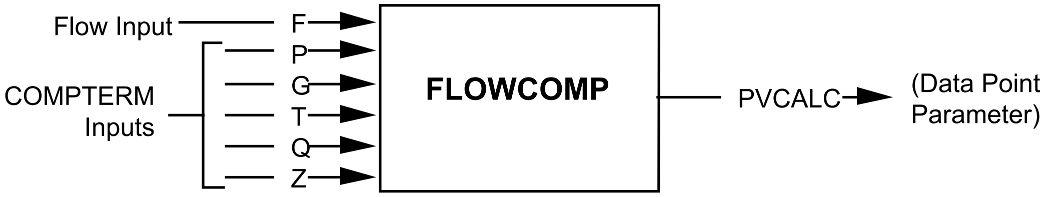
7.7.2.1 Function

This algorithm compensates a flow measurement for variations in temperature, absolute pressure, specific gravity, or molecular weight. The measured flow can be that of a gas, a vapor, or a liquid. An extended equation is provided for industrial steam-flow compensation, which includes factors that compensate for steam quality and compressibility. See Figure 7-3.

7.7.2.2 Use

The uncompensated-flow input is typically a square-rooted, differential pressure measurement. Other direct-flow measurements can also be used. The square root should be extracted before the input to the data point, and the input value must be in engineering units. For process-connected inputs, the square root can be extracted in the IOP; conversion to EUs also takes place in the IOP.

The compensation is calculated from temperature, pressure, specific gravity, molecular weight, steam quality, or steam compressibility. Choice of inputs depends on the type of the equation selected. All of these inputs are obtained through PV input-connections.



Simplified Equation:

PVCALC = F*COMPTERM

Where F is uncompensated flow and COMPTERM has five forms:

A: Liquids
B: Gases, Vapors
C: Gases, Vapors (Spec. Gravity)
D: Volumetric Flow of Gases and Vapors
E: Steam

Figure 7-3 — Functional Diagram, Flow Compensation PV Algorithm 1305

7.7.2.3 Options and Special Features

Five Forms of Flow Compensation

Parameter PVEQN specifies one of five different equations for this algorithm. The equation causes the compensation term (COMPTERM) to differ according to the application, as follows (see 7.7.2.4 for the actual equations):

Equation A

Primarily used for mass-flow or volumetric-flow compensation for liquids. Actual (measured or calculated) specific gravity is used as a compensation input.

Equation B

Primarily used for mass-flow compensation of gas or vapor flows. Actual absolute temperature and pressure are used as compensation inputs.

Equation C

Used for mass-flow compensation of gas or vapor flows. Actual specific gravity (measured or calculated), absolute temperature, and pressure are used as compensation inputs.

Equation D

Principally used for volumetric-flow compensation for gas or vapor flows. Actual temperature, pressure, and molecular weight are used as compensation inputs. The molecular weight can be calculated by the Calculator algorithm, or a user written program in the AM or Computing Module (CM50/CM60).

Equation E

Used for mass-flow compensation of steam flows in industrial applications. Actual temperature, pressure, specific gravity, steam compressibility, and steam quality are used as compensation inputs. This equation can also be used for "custody transfer" of gases or liquids.

Restart or Point Activation

On a cold or warm restart, or when this data point is activated, PVCALC is recalculated the next time the FLOWCOMP data point is processed.

Error Handling

If the status of any of the input values is bad, PVCALC contains NaN and the PVAUTOST becomes Bad. If there are no bad inputs but the status of one or more of the inputs is "uncertain," the PVAUTOST becomes Uncertn.

Special Notes

Refer to **7.7.2.4, Equations,** for more detail on the inputs and parameters mentioned in these notes.

Zero Pressure Reference—Parameter P0 (see definition of this parameter under 7.7.2.4) compensates for ambient atmospheric pressure. Most pressure sensors measure pressure relative to the atmospheric pressure. If the pressure measurement is actually absolute, P0 must be set to a value of zero. The usual zero reference is a value for sea level. If the pressure sensor is at a significantly different elevation than sea level, P0 should be set to a more appropriate value. For example, Denver, Colorado has an average atmospheric pressure of about 12.2 psia. Standard sea-level atmospheric pressure is 14.696 psia. P0 contains the absolute value of ambient atmospheric pressure.

Units of Measure—The absolute value of either U.S. Customary Units or SI (metric) units can be used. All inputs and parameters must be in engineering units of one system or the other. The typical value for P0 (see definition of this parameter under 7.7.2.4) in U.S. Customary Units is 14.696 psia and in SI units it is 101.325 kPa. The typical T0 value is 459.69°F in U.S. Customary Units and 273.15°C in SI units.

Compensation Term Value—Typically, the COMPTERM value is near 1. It should never be zero or negative. The COMPLO and COMPHI limits are used to prevent unrealistic values of COMPTERM caused by incorrect inputs. Should the calculated value of COMPTERM go beyond one of these limits, the value is held (clamped) at that limit. You should estimate the range of COMPTERM by considering the most extreme input-conditions you expect. Also, you should set the PV range for this data point, by considering the largest compensated-flow value expected.

Custody Transfer—Equation E can be used for "custody transfer" of gases or liquids. To do so, set parameter RX (see definition of this parameter under 7.7.2.4) equal to one and specify the input connection to X to come from RX in this data point.

Compensating for Assumed Design Conditions

Equation A can be used for either mass or volumetric compensation of liquid flows. The use depends on whether the measurement of uncompensated flow is a mass measurement or a volumetric measurement, and on the desired uncompensated-flow units. Here are three ways to use Equation A:

- **Converting an uncompensated mass-flow to compensated mass-flow**; C1 and C2 (see 7.7.2.4) are configured as 1.0.
- **Converting an uncompensated, standard volumetric-flow to compensated mass-flow**; C1 is configured to equal the design density, referenced to standard conditions. C2 is configured as 1.0.

- **Converting uncompensated, standard volumetric-flow to compensated, standard volumetric-flow**; If the variations in standard density caused by fluid-composition changes are significant, C2 is manipulated as follows:

  If the measured value of specific gravity at flow conditions is available, the actual specific gravity, referred to standard conditions, is calculated from that measurement by another data point and input to C2 through a general input connection. If actual specific gravity is measured by a lab, a numeric data-point could be used to hold the value and input to C1 through a general input connection. For the latter case, another data point uses the lab value to calculate specific gravity at flow conditions and the result is input G.

7.7.2.4 Equations

You configure PVEQN for data point that uses the Flow Compensation algorithm to specify one of five equations. The equations select the compensation term. The basic equation is

$$\mathrm{PVCALC} = \mathrm{C} * \frac{\mathrm{C1}}{\mathrm{C2}} * \mathrm{F} * \mathrm{COMPTERM}$$

Where:

PVCALC = The output of this algorithm. It is selected as the PV for this data point when the PVSOURCE is Auto.

C = Scale factor. The default value is 1.0.

C1, C2 = Constants for correcting for assumed design conditions. See **Compensating for Assumed Designed Conditions** under 7.7.2.3. Default value for each is 1.0.

F = The uncompensated flow input. A square-rooted, differential pressure input.

COMPTERM = The compensation term. This term differs in each of the five flow-compensation equations, A through E. Its value lies between the COMPLO and COMPHI limits, which are specified by the process engineer. If either limit parameter contains NaN, the corresponding limit check is not made.

The five forms of COMPTERM are as follows:

NOTE

The equations shown below are performed using absolute values for pressure and temperature. You must convert temperatures and pressures to absolute values in order to use this algorithm.

The COMPTERM computation depends on the value of the PVCHAR parameter. If PVCHAR = SqrRoot, then COMPTERM is as shown. If PVCHAR = Linear, then COMPTERM is determined without the square root function.

$$\text{Equation A: COMPTERM} = \sqrt{\frac{G}{RG}} \quad \text{(Liquids)}$$

$$\text{Equation B: COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0}} \quad \text{(Gases \& Vapors)}$$

$$\text{Equation C: COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0} * \frac{G}{RG}} \quad \text{(Gases \& Vapors w/Specific Gravity)}$$

$$\text{Equation D: COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0} * \frac{RG}{G}} \quad \text{(Volumetric Flow of Gases \& Vapors)}$$

$$\text{Equation E: COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0} * \frac{X}{RX} * \frac{RQ}{Q}} \quad \text{(Steam)}$$

Where the following (in engineering units) are received through input connections

G = Measured or calculated specific gravity or molecular weight.

P = Measured actual gage pressure.

T = Measured actual temperature.

X = Measured actual steam compressibility.

Q = Measured actual steam-quality factor.

And the following parameters are specified by the process engineer

RG = Design specific gravity or reference molecular weight, in the same engineering units as G (Default value = 1.0).

RP = Design pressure, converted to an absolute value (Default value = 1.0).

RQ = Design steam quality factor, in the same units as Q (Default value =1.0)

RT = Design temperature, converted to an absolute value (Default value = 1.0).

P0 = Factor to convert gauge pressure to an absolute value. Typically 14.696 psia or 101.325 kPa. Enter the absolute value of the number. **See 7.7.2.3 - Special Notes**. Default value = 0. If the measured pressure is already an absolute value, enter 0.

T0 = Factor to convert Celsius and Fahrenheit temperatures to an absolute value. Typically 459.69°F or 273.15°C (use the absolute value of the number when entering a value in T0). See **Special Notes**. Default value = 0. If the measured temperature is already an absolute value, enter 0.

RX = Reference steam compressibility, in the same engineering units as X. Default value = 1.0.

Other parameters associated with this algorithm are as follows (refer to the *Advanced Process Manager Parameter Reference Dictionary*):

COMPHILM
COMPLOLM
COMPTERM
FSTS
GSTS
PSTS
PVCALC
PVEQN
QSTS
TSTS
XSTS

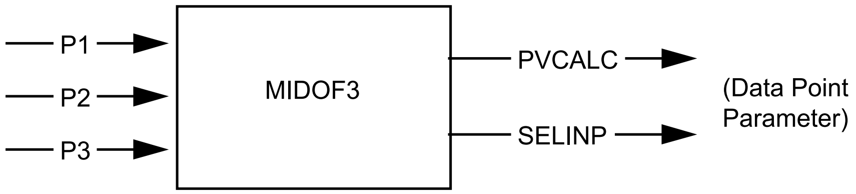
7.7.3 Middle of Three Selector (MIDOF3)

7.7.3.1 Function

This algorithm provides a calculated PV (PVCALC) that is normally the middle value of three values from the PV input connections. PVAUTOST is Bad, only if all three inputs to this algorithm are bad. If at least one input is valid (normal or uncertain), the algorithm provides a valid value in PVCALC.

When configured with only two inputs, this algorithm serves as a high/low selector or input averaging block. These functions are selected by choosing the corresponding control equation (A, B or C as shown below).

If only one valid input value is available, it is selected. If only two valid input values are available, the selected value can be the highest or the lowest, or the average of the two, as specified when you select the equation to be used by this algorithm. See Figure 7-4.



Normal Operation: PVCALC = Middle value of the three input values.

With only two valid inputs:

Equation A; PVCALC = Highest of the two inputs

Equation B; PVCALC = Lowest of the two inputs

Equation C; PVCALC = Average of the two inputs

With only one valid input: PVCALC = Value of the input

SELINP = The selected input, SelectP1 through SelectP3, except,
with only two valid inputs and Eq. C, SELINP contains None.

Figure 7-4 — Functional Diagram, Middle-of-Three Selector PV Algorithm 1312

7.7.3.2 Use

This algorithm is used to provide a reasonably secure PVCALC when inputs are available from three redundant inputs, one or more of which may occasionally fail or provide erratic values. The Low Selector, High Selector, Average PV algorithm provides a somewhat similar function with up-to-six input connections (see 7.7.4).

7.7.3.3 Options and Features

Normal Operation with Three Valid Inputs

Normal operation occurs if there are no inputs with a bad-value status. Inputs are treated as valid if their value status is either normal or uncertain.

If no two inputs have equal values,

```
PVCALC = the middle value of the three inputs, P1, P2 and P3
```

and

```
SELINP = the selected input, SelectP1 through SelectP3
```

If there are two inputs with equal values or if all three input values are equal,

```
PVCALC = the value for which there is at least one other equal
```

and

```
SELINP = the lowest-number input with and equal value, SelectP1
         through SelectP3.
```

Error Handling

PVAUTOST becomes Uncertn only when the selected input is uncertain, or for equation C, when one of the inputs used for averaging is uncertain.

The PVAUTO status is bad and PVCALC becomes NaN when the status of all three inputs is bad.

7.7.3.4 Equations

If three valid inputs are present, the equations have no meaning and the algorithm functions normally, as described under **Normal Operation with Three Valid Inputs.** The equations specify what the algorithm is to do if one or more inputs has a bad-value status. The equations function as follows:

- With one bad input

Equation A

```
PVCALC = Highest of the two input values

SELINP = The selected input, SelectP1 through SelectP3
```

Equation B

```
PVCALC = Lowest of the two input values

SELINP = The selected input, SelectP1 through SelectP3
```

Equation C

```
PVCALC = The average of the two input values
SELINP = None
```

- With two bad inputs

Equations A, B and C

```
PVCALC = the value of the valid input
SELINP = The selected input, SelectP1 through SelectP3
```

- With three bad inputs

Equations A, B, and C

```
PVCALC = NaN
SELINP = None
```

Where:

PVCALC =	The output of this algorithm. It is selected as the PV for the data point when the PVSOURCE is Auto.
P1, P2, and P3	The input values. The default value is NaN.
SELINP =	The selected input, SelectP1 through SelectP3. If no input is selected or if PVCALC contains an average value, SELINP contains None.

Other parameters associated with the MIDOF3 algorithm are as follows

P1STS P3STS P2STS PVEQN

Refer to the *Advanced Process Manager Parameter Reference Dictionary* for more information.

7.7.4 High Selector, Low Selector, Average (HILOAVG)

7.7.4.1 Function

This algorithm does one of the following:

- Selects the input with the highest value
- Selects the input with the lowest value
- Calculates the average value of all valid inputs

It can accept up to six inputs. Valid inputs are those with a status that is "Normal" or "Uncertain." When the input selection functions are used, the number of the input that is selected is contained in a accessible parameter (SELINP). See Figure 7-5.

7.7.4.2 Use

One example of the use of this algorithm is shown at the top of Figure 7-5. In this example, the high value-selector version of the algorithm is used to detect hot spots in a boiler or a reactor.

Either the high value-selector version or the low value-selector version can be used to detect production bottlenecks. For example, this algorithm might be used to notify the process operator that production is currently constrained by the speed of a gas compressor. One of the selector options might also be used to select the "safest" PV for control.

One use of the averaging option is in balancing furnace passes. In this application, the algorithm calculates the average of the outlet temperatures of the passes.

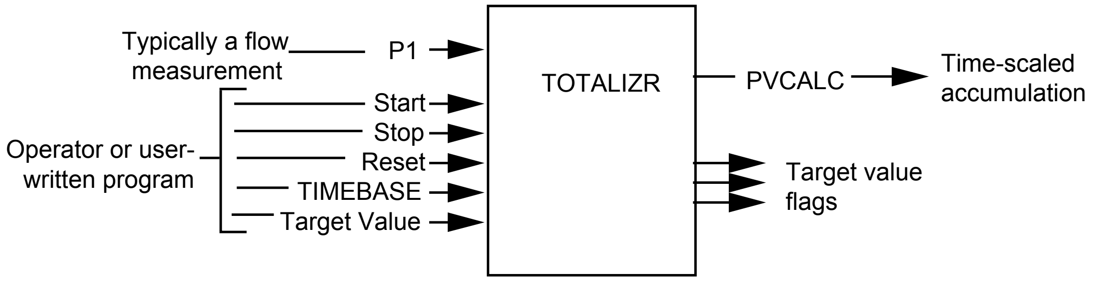
7.7.4.3 Options and Special Features

Forced Selection

The data point can be configured to allow the Universal Station operator, a user-written program, or a general-input connection to force selection of one of the inputs.

- If the FRCPERM parameter is configured as On, the forced-selection function is enabled and an operator or a user-written program can force the selection.
- IF FRCPERM is configured as Off, the forced-selection function is disabled.

The FSELIN parameter specifies the input to be selected, when selection is forced (SelectP1 through SelectP6).



Equations A through F specify bad value and restart-handling options. See 7.4.5

For all equations:

$$PVCALC = PVCALC_{(i-1)} + C * (Time\ Scale) * P1$$

Figure 7-5 — Functional Diagram, HI LO Average Selector PV Algorithm

1310

Error Handling

Except when forced selection is in effect (**Forced Selection**), inputs with a bad status are ignored and they do not make the PVAUTOST is Bad. For example, if the algorithm is configured as a 4-input high-selector and one of the inputs goes bad, the algorithm functions as a 3-input high-selector.

If the number of valid inputs (PVSTS is Normal or Uncertn) is less than the minimum number specified in parameter NMIN, PVCALC becomes NaN and the PVAUTOST is Bad.

PVAUTOST is changed to Uncertn under any of the following conditions:

- An input selection is forced and the status of that input is not bad (is normal or uncertain).
- Forced selection is not in effect, at least as many inputs as specified by NMIN are normal or uncertain, and the status of the selected one (Equation A or B) is uncertain.
- Equation C (averaging) is chosen, at least as many inputs as specified by NMIN are not bad (normal or uncertain), and the status of any of them is uncertain.

PVCALC becomes NaN and PVAUTOST becomes Bad under either of the following conditions:

- The selection of an input is forced and the status of that input is bad.
- Forced selection is not in effect, and there are fewer inputs with a status other than bad than are specified by NMIN.

Restart or Point Activation

On a cold or warm restart, or when this data point is activated, PVCALC is simply recalculated the next time this data point is processed.

7.7.4.4 Equations

Equation A selects the highest input value. Equation B selects the lowest input value. Equation C calculates the average of all valid inputs.

Equation A—High Selector

If FRCPERM and FORCE are both On,

```
PVCALC = the value of the input indicated by FSELIN and

SELINP = FSELIN
```

If either FRCPERM or FORCE is Off,

```
PVCALC = the highest valid input.

SELINP = the selected input, SelectP1 through SelectP8.
```

Equation B—Low Selector

If FRCPERM and FORCE are both On,

```
PVCALC = the value of the input indicated by FSELIN and

SELINP = FSELIN
```

If either FRCPERM or FORCE is Off,

```
PVCALC = the lowest valid input.

SELINP = the selected input, SelectP1 through SelectP8.
```

Equation C—Average

If FRCPERM and FORCE are both On,

```
PVCALC = the value of the input indicated by FSELIN and

SELINP = FSELIN
```

If either FRCPERM or FORCE is Off,

```
PVCALC = (Sum of the valid inputs)/N

SELINP = None
```

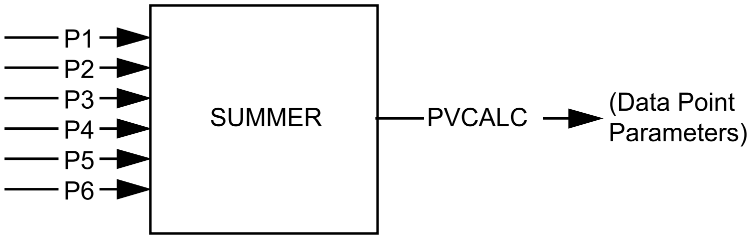
Other parameters associated with the HILOAVG algorithm are as follows (refer to the *Advanced Process Manager Parameter Reference Dictionary*):

NMIN PVEQN PnSTS SELINP

7.7.5 Summer (SUMMER)

7.7.5.1 Function

This algorithm calculates a PV (PVCALC) that is the sum of up-to-six input values. The input values can be scaled, the combined inputs can be scaled, and a bias value can be added to the result. See Figure 7-6.



EQUATION B, simplified:

PVCALC = P1 + P2 + . . . + P6

Figure 7-6 — Functional Diagram, Summer PV Algorithm

1314

7.7.5.2 Use

A typical use is the calculation of the rate at which a component of a raw product is entering a process unit, which is found by summing the proportion of the component in each of several input streams and multiplying by the stream flow rates. This algorithm can also be used to calculate a net heat loss by finding the difference between the heat inputs and heat outputs (the difference can be obtained by using a negative scale factor, for example, –1.0).

Other possible uses are mass-balance, heat-balance, and inventory calculations.

This equation can be used to scale and bias a single variable; see **7.7.5.4 Equation A**.

7.7.5.3 Options and Special Features

Ensuring Adequate PV Range

Because the input values can be either positive or negative, as can the scale factors and bias values, the results in PVCALC can have a broad range of values. You should evaluate the worst-case values you expect to be in use, to establish the PV range. When you configure the data point, be sure to specify a PV range adequate to cover all expected values.

Error Handling

If there are no inputs with a bad status and the status of at least one input is uncertain, PVAUTOST is Uncertn.

If the status of at least one input is bad, the PVAUTOST becomes Bad and PVCALC contains NaN.

Restart or Point Activation

On any type of restart or when this data point is activated, PVCALC is normally calculated.

7.7.5.4 Equations

You can select one of two equations when you configure a data point that uses the Summer PV algorithm:

Equation A

```
PVCALC = C * P1 + D
```

Equation B

```
PVCALC = C * ((C1 * P1) + (C2 * P2) + . . . + (Cn * Pn)) + D
            (Pn <= 6)
```

Where:

PVCALC = The output of this algorithm. It is selected as the PV for this data point when the PV source is AUTOmatic.

C = The overall scale factor. Default = 1.0.

C1 through Cn = The scale factors for P1 through Pn. Default = 1.0.

P1 through Pn = The PV input values. Equation B is limited to six inputs. Default for all values is NaN.

D = The overall bias. Default = 0.

n = The number of PV inputs used. Default = 2.

Other parameters associated with the SUMMER algorithm are: N, PnSTS and PVEQN. Refer to the *Advanced Process Manager Parameter Reference Dictionary* for more information.

7.7.6 Variable Dead Time with Lead-Lag Compensation (VDTLDLG)

7.7.6.1 Function

This algorithm provides a calculated PV (PVCALC) in which value changes may be delayed from the time that the corresponding change occurred in the P1 input. Dynamic lead-lag compensation to the PV can also be provided. Lag compensation is available in combination with the delay or with no delay. The delay time can be fixed or can be varied as the value of an input varies. See Figure 7-7.

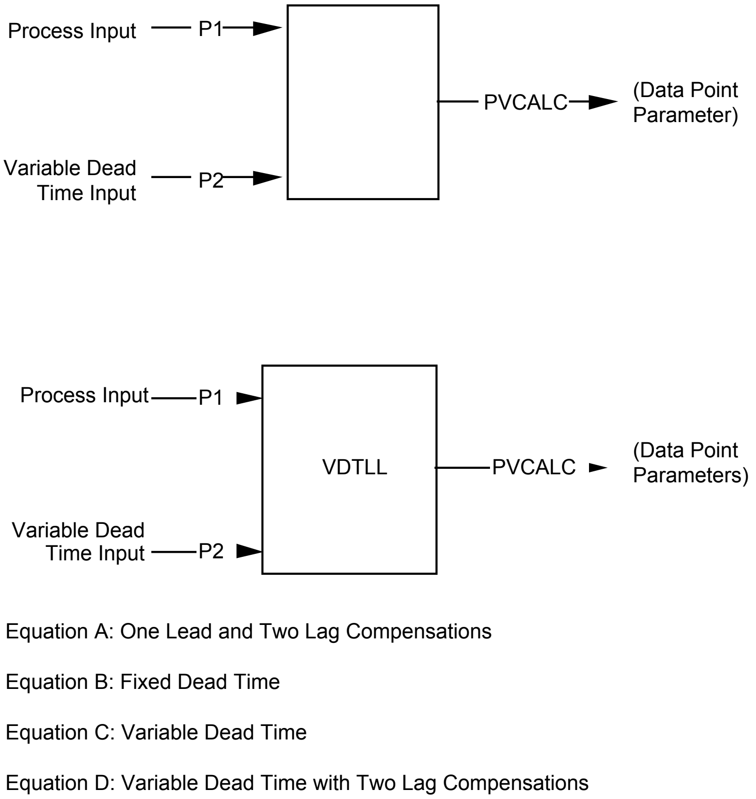

Process Input
P1
PVCALC
(Data Point Parameter)
Variable Dead Time Input
P2
Process Input
P1
VDTLL
PVCALC
(Data Point Parameters)
Variable Dead Time Input
P2
Equation A: One Lead and Two Lag Compensations
Equation B: Fixed Dead Time
Equation C: Variable Dead Time
Equation D: Variable Dead Time with Two Lag Compensations
1316

Figure 7-7 — Functional Diagram, Variable Dead time with Lead Lag PV Algo

7.7.6.2 Use

This algorithm is used for feedforward control and in process simulations. For additional use information, see Equations C and D, under 7.7.6.4.

This algorithm can be used as the PV algorithm in a data point that uses the PID Feedforward control algorithm (see 8.13.2), Figure 8-7.

In a typical feedforward application, the PV provided by this algorithm serves as the feedforward PV. An operator can "cut out" this feedforward component by switching the PVSOURCE to Man.

7.7.6.3 Options and Special Features

Four Combinations of Delay and Lead-Lag Compensation

You select the combinations of delay, lead compensation, and lag compensation by selecting Equation A, Equation B, Equation C, or Equation D when configuring the data point. The equations function as follows:

- **Equation A, Lead-Lag**—A change in the input value (P1) is subjected to one lead compensation and two lag compensations. If you specify a time constant of zero in the Lead Compensation time constant, TLD, Lag Compensation time constants, TLG1, or TLG2, the corresponding lead or lag compensation is suppressed. If you don't suppress the lead compensation, you must use at least one lag compensation.

- **Equation B, Fixed Dead Time**—A change in the input value (P1) is delayed by a user-specified time. This data point must be made inactive in order to change the dead-time value (TD).

- **Equation C, Variable Dead Time**—A change in the input value (P1) is delayed by a time period the duration of which varies as the inverse of P2-input value variations. The variable time period is determined by P2, the C1 and C2 scale factors, and bias values D1 and D2. The delay (or dead time) typically represents a delay in the process that depends on some variable in the process, such as flow, feed rate, or a conveyer-belt speed.

  Equations C and D have a cutoff feature that can simulate situations like a conveyer belt stopping. If the flow or speed value, represented by the P2 input, drops below a limit that you configure in the CUTOFFLM parameter, the value of the delayed P1 signal (DP1) goes to zero. (DP1 is not an external parameter.) When P2 again exceeds the CUTOFFLM value, DP1 resumes as a normal, delayed output. If you don't want this feature, configure CUTOFFLM as NaN. Note that DP1 is subject to scale factor C and bias value D. See 7.7.6.4.

  Equation C can be used to produce a fixed delay time that can be changed while the data point is active.

- **Equation D, Variable Dead Time with Two Lags**—A change in the input value (P1) is delayed as with Equation C and then receives lag compensation as specified by one or two time constants (TLG1, TLG2). This equation is useful for simulating a portion of a process that can be represented by a dead time and one or two lags. The cutoff feature applies as for Equation C.

Dead-Time (Delay-Time) Calculation

The delay of the input values is accomplished by a process that has the effect of shifting the values through a table in the APM's memory. Values are shifted from one location in the table to the next, at intervals calculated to provide the desired delay. This is illustrated in Figure 7-8.

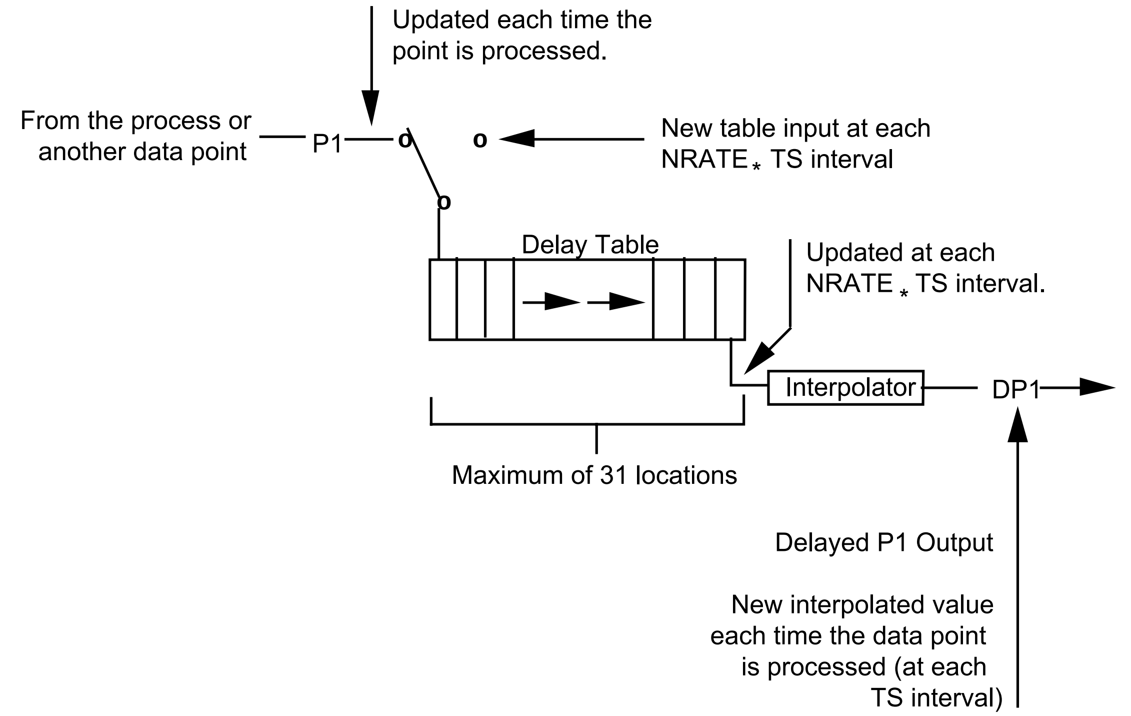

Updated each time the point is processed.
From the process or another data point
P1
o
o
o
New table input at each NRATE * TS interval
Delay Table
Updated at each NRATE * TS interval.
Interpolator
DP1
Maximum of 31 locations
Delayed P1 Output
New interpolated value each time the data point is processed (at each TS interval)
1317

Figure 7-8 — Variable Delay Time Functional Diagram

For an example of the delay-table operation, suppose that the P1-input value has been constant at 5.0 units for an hour. Assume that the specified delay time, TD, is 15 minutes and that the data point is processed every second; Time Sample (TS = 1/60) min. as determined by the SCANRATE parameter. At this time, the output of the interpolator is 5.0 units, all of the locations in the table contain a value of 5.0 units, and P1 contains a value of 5.0 units.

Now suppose that the input to P1 suddenly changes to 6.0 units. The interval at which new values are shifted through the table and the number of table locations in use have been set up so that it takes 15 minutes for the new value of 6.0 units to appear at the output of the interpolator.

Three sample calculations follow. The first shows how a fixed delay time is determined (Equation B), the second shows how a variable delay time is determined, and the third shows how a change in the P2 input changes the variable delay time.

Fixed Delay-Time Example

1. The value in TS is 0.5 minutes and TD has been specified as 15 minutes.

2. NRATE, the table shift-rate factor, is calculated as follows:

$$NRATE = \frac{TD}{(TS * 30)} = \frac{15}{(0.5 * 30)} = 1$$

3. NLOC, the number of table locations to be used, is

$$NLOC = \frac{TD}{(TS *NRATE)} = \frac{15}{(0.5 * 1)} = 30 \text{ locations}$$

4. The actual delay time is then recomputed as

$$TD = NLOC*NRATE*TS = 30*1*0.5 = 15 \text{ minutes}$$

If the calculated values of NRATE and NLOC had resulted in fractions, the results would have been rounded up to the nearest larger integer and the actual delay time would have been slightly more than specified.

Where the TD is less than 30 times TS, NRATE always has a value of 1. In such cases, the delayed output is a true, but delayed, representation of the corresponding input value.

Where the NRATE value is greater than 1, the output signal is interpolated to approximate the earlier change in the input, by using the last output value and the value in the last location in the table. This is an excellent approximation for typically smooth changes in process values.

Variable Delay-Time Example

1. The value in TS is $\frac{1}{60}$ minutes. NLOC = 30.

2. Each time the data point is processed, a new variable delay time is calculated as

   $$\text{TDNEW} = \frac{\text{C1}}{(\text{C2} * \text{P2}) + \text{D2}} + \text{D1}.$$

   Assume the P2 input is 20.0 units and its scale factor is 0.05. Scale factor C1 is 30.0. The biases, D1 and D2 both equal 0.

   $$\text{TDNEW} = \frac{30.0}{(0.05 * 20.0) + 0} + 0 = 30 \text{ minutes.}$$

3. $$\text{NRATE} = \frac{\text{TDNEW}}{(\text{TS} * \text{NLOC})} = \frac{30}{(\ \frac{1}{60} * 30)} = 60$$

   NLOC is configurable (range 2 - 30; in this example it is configured as 30)

4. Actual delay time is calculated as

   $$\text{TD} = \text{NLOC} * \text{NRATE} * \text{TS} = 30 * 60 * \frac{1}{60} = 30 \text{ minutes.}$$

   NLOC is configurable (range 2 - 30; in this example it is configured as 30)

   Now suppose the P2 input changes to 23.4 units.

Second Variable Delay-Time Example

1. The value in TS is $\frac{1}{60}$ minutes.

2. $\text{TDNEW} = \frac{\text{C1}}{(\text{C2} * \text{P2}) + \text{D2}} + \text{D1} = \frac{30.0}{(0.05 * 23.4) + 0} + 0 = 25.641$ minutes.

3. $$\text{NRATE} = \frac{\text{TDNEW}}{(\text{TS} * \text{NLOC})} = \frac{25.641}{\frac{1}{60} * 30} = 51.282$$

   This is rounded to the nearest integer (not necessarily the next larger integer, as for a fixed delay time), so
   NRATE = 51.

4. Actual delay time then is

$$TD = NLOC * NRATE * TS = 30 * 51 * \frac{1}{60} = 25.5 \text{ minutes.}$$

NLOC is configurable (range 2 - 30; in this example it is configured as 30)

In the Second Delay Time Example, the exact delay specified by the input was 25.641 minutes.

Changing Dead-Time (Delay-Time) Parameters

Variable delay-time parameters C1, C2, D1, and D2 in Equations C and D can be changed at a Universal Station while the data point is active. Note that the D1 value allows a supervisor or engineer to add a fixed delay time to the total variable delay time.

The C1 and C2 parameters are used to specify the time scale of the P2 value. D1 can be used to offset that scale. Note that the P2 value is inversely proportional to the variable time delay. Where P2 represents a flow rate or speed, when the flow or speed decreases, the time delay increases to simulate the effect of the reduced flow or speed.

Restrictions on Delay Time

The minimum fixed delay time (Equation B) is equal to TS, the processing interval in minutes. Delay values greater than 32,000*TS are rejected.

For Equations C and D the minimum step-change in the TD value is equal to NLOC*TS. This is also the value of the smallest dead time (delay time). If the TDNEW value is less than zero, it is clamped to zero. Also, if TDNEW exceeds 32,000*TS, it is clamped to 32,000*TS.

Time-Constant Recommendations

We recommend that the processing rate of a data point that uses this algorithm and Equation A or D must be at least ten times greater than the lead or lag break-point frequencies, so, divided by lead break-point frequency) be less than or equal to 10, so,

TLG1 should be equal to or greater than 2 * TS
TLG2 should be equal to or greater than 2 * TS
|TLD| should be equal to or greater than 10 * TS

Both positive and negative lead times can be specified, so it is the absolute value of TLD that must be equal to or greater than 10 * TS.

We recommend that the rate amplitude (lag break-point frequency divided by the lead break-point frequency) be less than or equal to 10 so,

|TLD| should be equal to or less than 10 * TLG1.

Using Equation C or D for a Fixed Delay Time

You can use these variable delay-time equations to attain a fixed delay time by setting the value of C1 to 0 and adjusting the value of D1 to get the desired delay value. This permits changes of delay time from a Universal Station while the data point is active, but the resolution may be much less than using a fixed delay time (Equation B), where the delay can be changed only by making the point inactive and then active again.

Restart or Point Activation

On a cold start, a warm start, and when the data point is activated, the lead-lag dynamics are set to the steady state, and all values in the delay table are set to the current value of the P1 input. PVCALC is calculated as follows:

```
PVCALC = C * P1 + D
```

Error Handling

For Equations C and D, if neither input has a bad-value status, but one or both has an uncertain-value status, PVAUTOST is Uncertn. Equations A and B don't use the P2 input, so for them, PVAUTOST is Uncertn only if the P1-value status (P1STS) is Uncertn.

For Equations C and D, if either input has a bad-value status, PVCALC becomes NaN and the PVAUTOST is Bad. For Equations A and B, when P1STS is Bad, it causes PVCALC to contain NaN and the PVAUTOST to be Bad.

When the input-value status is again normal or uncertain, the data point is initialized as for a cold start under **Restart or Point Activation**, and the PVAUTO-value status becomes normal, or uncertain, as appropriate.

7.7.6.4 Equations

You can select one of four equations when you configure a data point that uses the Variable Dead Time with Lead-Lag Compensation PV algorithm:

Equation A—Lead Compensation with Two Lag Compensations

$$\mathrm{PVCALC}(s) = \left[C * \frac{1 + \mathrm{TLD} * s}{(1 + \mathrm{TLG1} * s) * (1 + \mathrm{TLG2} * s)} * \mathrm{P1}(s)\right] + D$$

Equation B—Fixed Delay Time

$$\mathrm{DP1}_{t} = \mathrm{P1}_{t-\mathrm{TD}}$$

$$\mathrm{PVCALC}(s) = C * \mathrm{DP1} + D$$

Equation C—Variable Delay Time

If CUTOFFLM does not contain NaN and if P2 is less than CUTOFFLM, DP1 = 0. Otherwise, calculate DP1 as follows:

$$DP1_{t} = P1_{t-TD}$$

$$TD = \frac{C1}{C2 * P2 + D2}$$

$$PVCALC(s) = C * DP1 + D$$

Equation D—Variable Delay Time with Two Lag Compensations

$$TD = \frac{C1}{C2 * P2 + D2} + D1$$

$$DP1_{t} = P1_{t-TD}$$

$$PVCALC(s) = [C * \frac{1}{(1 + TLG1 * s) * (1 + TLG2 * s|)} * DP1(s)] + D$$

Where:

PVCALC = The output of this algorithm. It is selected as the PV for this data point when the PV source is AUTOmatic.

C = The overall scale factor. Default value = 1.0.

C1 = Scale factor, TDNEW denominator. Default value = 1.0.

C2 = Scale factor for P2. Default value = 1.0.

CUTOFFLM = Cutoff (zero-flow or zero-belt speed) limit. Default = 0.0.

D = Overall bias. Default value = 0.

DP1 = The delayed P1 value. Not accessible to Universal Stations nor to user-written programs.

D1 = Bias value for the variable delay time. Default value = 0.

D2 = Bias for P2. Default value = 0.
P1 = The input value to which the delay and lead-lag compensation are applied.

P2 = The input value that changes the variable delay when Equation C or Equation D is used.

s = The Laplace operator (notation only, not a parameter)

t = The present time (notation only, not a parameter)

t-TD = The present time minus the actual dead (delay) time (notation only, not a parameter).

TD = The fixed time delay in minutes for Equation B. The actual variable delay time in minutes for Equations C and D. Default = 0.

TDNEW = The calculated new (ideal) delay time in minutes for Equations C and D.

TLD = Lead-compensation time constant in minutes. 0 = no lead. Default = 0.

TLG1 = Lag-compensation time constant 1 in minutes. 0 = no lag. Default = 0.

TLG2 = Lag-compensation time constant 2 in minutes. 0 = no lag. Default = 0.

Other parameters associated with the VDTLDLG algorithm is as follows (refer to the *Advanced Process Manager Parameter Reference Dictionary*):

PVEQN NLOC (equations C and D)

7.7.7 Totalizer (TOTALIZR)

7.7.7.1 Function

This algorithm provides a time-scaled accumulation of an input value. The input value is typically a flow measurement. Either analog or pulse input can be selected through parameter ACCTYPE. The time-base can be seconds, minutes, or hours.

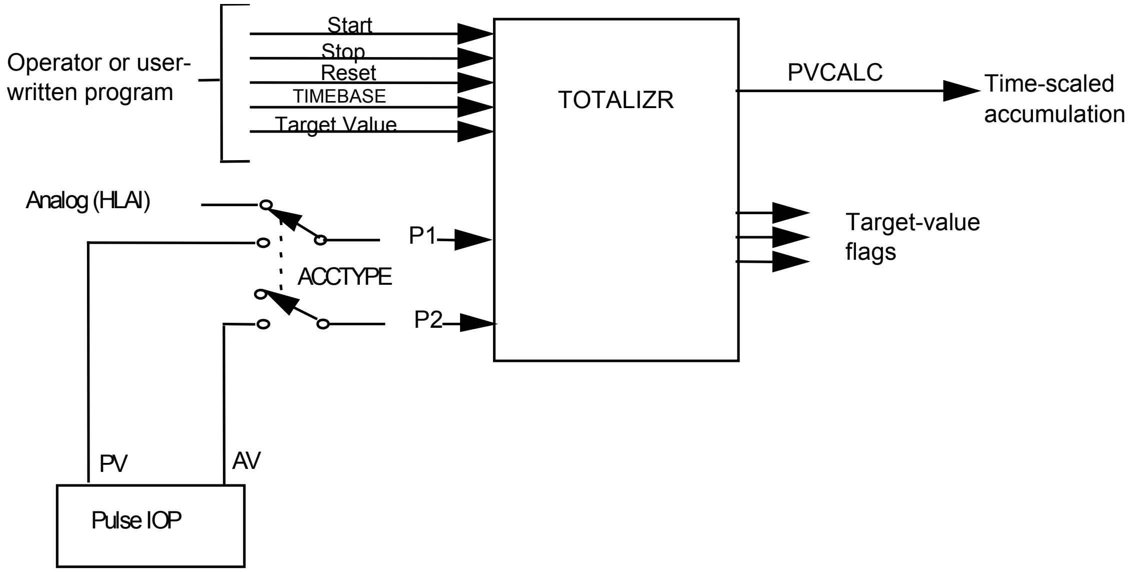


Figure 7-9 — Functional Diagram, Totalizer PV Algorithm

3369

The accumulation can be started, stopped, and reset by commands from a Universal Station operator or from a user-written program. An operator or user-written program can establish a target value for the accumulation. Status indicators are available to indicate that the accumulation is near the target value, nearer to the target value, and is complete (has reached or exceeded the target value).

For situations where the flow transmitter may not be precisely calibrated near the zero-flow value, a zero-flow cutoff feature is provided that avoids accumulating negative flow values. When the flow is below a user-specified cutoff value, the input value is clamped to zero.

7.7.7.2 Use

The Totalizer PV algorithm accumulates periodic measurements over time. It is principally used to accumulate total flows, or in applications such as the measurement of ingredients that are blended. The accumulated value can be used for control or just as process history.

An example of TOTALIZR's use in control is determining how full a tank is, so that the flow into the tank can be shut off before it overflows. In such an application, the P1 input to TOTALIZR would be the PV of PID-flow controller.

7.7.7.3 Options and Special Features

Typical Operation

The events in an operation that uses TOTALIZR might be as follows (see Figure 7-10):

- The target value, which represents the desired total volume, is specified to the AVTV parameter in the TOTALIZR point, by an operator at a Universal Station or by a user-written program.

NOTE

AVTV is displayed as the setpoint (SP).

- An operator or a user-written program issues a Reset command (using the COMMAND parameter) to the TOTALIZR point. This sets any accumulation value equal to RESETVAL.

- A Start command is issued to the TOTALIZR point. A logic slot or user program sets the setpoint to some value.

- When the first "slowdown" or "near-target" flag (ADEV1FL) comes on, it is read by logic or user program and reduces the setpoint.

- When the second "slowdown" or "near-target" flag (ADEV2FL) comes on, it is read by logic or user program and reduces the setpoint.

- When the accumulation reaches the target value (AVTV), filling is complete and the complete flag (AVTVFL) comes on. It is read by logic or user program and sets the setpoint = 0.

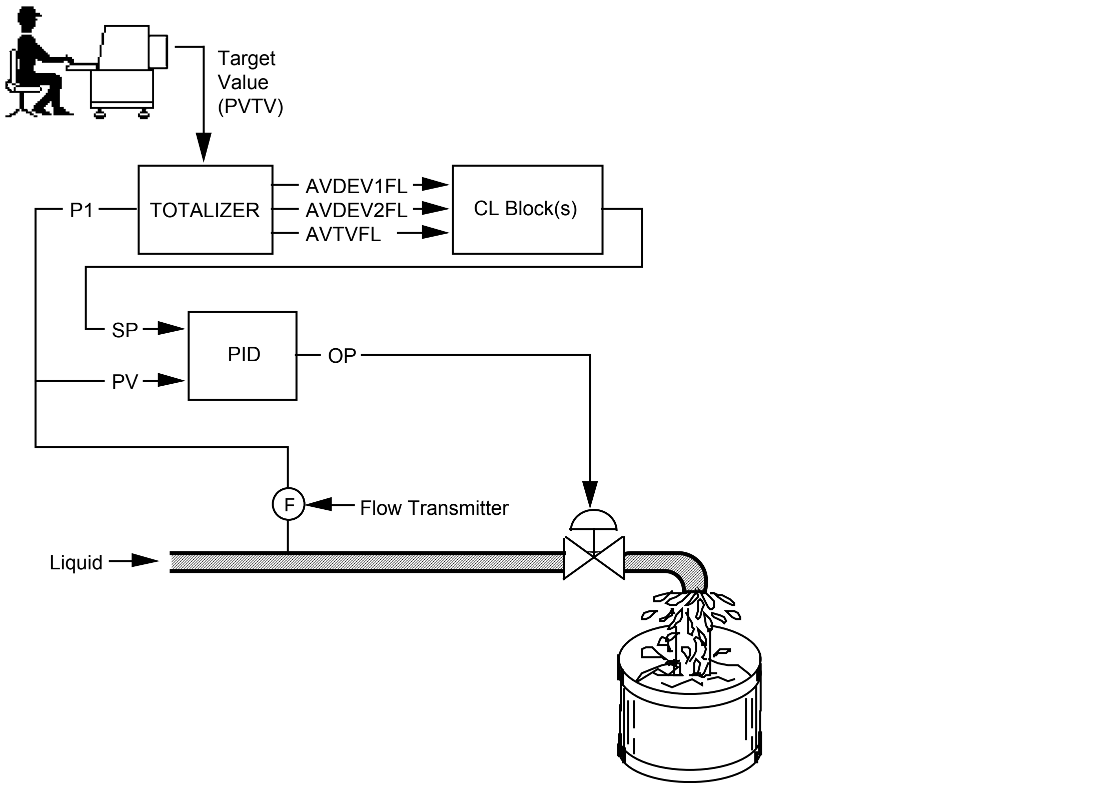


Figure 7-10 — Using TOTALIZR to Fill A Tank

1311

Time-Base and Engineering-Units Scaling

The user specifies the time base in seconds, minutes, or hours, in parameter TIMEBASE. This is the time base in which the flow measurement is made. For example, liters per second. This parameter only applies when ACCTYPE = ANALOG.

Scale factor, C, can be used to convert from one set of engineering units to another, for example, from gallons per minute to barrels per minute.

Commands, States and Command Flags

Commands can be issued to the data point that is using TOTALIZR from a Universal Station or by a user-written program. These commands are written in the TOTALIZR point's COMMAND parameter.

The commands are as follows:

- `None`—No action.

- `Start`—Start the accumulation. STATE changes to Running.

- `Stop`—Stop the accumulation. STATE changes to Stopped.

- `Reset`—Reset the accumulated value to a user-specified value. This value is specified in parameter RESETVAL. If the accumulator is running, it continues from the reset value.

Command Flags

In addition to the above (enumerated) commands, explicit command flags are provided to reset, start, or stop the totalizer.

- RESETFL — Off to On transition causes the totalizer to be reset. (See **Accumulated Value Before Reset** for additional information.)

- STARTFL — Off to On transition causes the STATE of totalizer to be Running.

- STOPFL — Off to On transition causes the STATE of totalizer to be changed to Stopped.

The above flags are program-access level, so they can be written to by a Logic Point or a user-written program.

Accumulated Value Before Reset

The reset command sets the PVCALC parameter equal to the reset value (RESETVAL). The value of PVCALC just prior to being reset is saved as the old accumulation value (OLDAV). This allows other system functions using the totalized value to be able to reset the totalizer without losing any "accumulation."

Range of Values, Clamping Option

The accumulated value has a normal range of PVEULO to PVEUHI, where PVEUHI defines the point where the bar graph is at the 100% level, and PVEULO is the point where the bar graph is at the 0% level.

This algorithm will continue to totalize past PVEUHI until it reaches the value of PVEXEUHI. When it reaches PVEXEUHI, the following occurs:

- If NoClamp was selected, the PV is set to BadPV and displayed as NAN.
- If Clamp was selected, the PV is flagged as uncertain and clamped to PVEXEUHI.

In either case, the algorithm continues to accumulate a value in PVCALC until it is reset, regardless of the selection of NoClamp/Clamp.

Using Scientific Notation

Values are accumulated as integers, but displayed as real numbers. You can enter values using scientific notation; for example, 1,000,000 can be entered as 1E6. If the accumulated value exceeds the limit of displayable characters, it will be displayed in scientific notation.

Near-Zero Cutoff

To prevent accumulation of negative flow values, where the flow transmitter may not be precisely calibrated near zero flow, you can specify a cutoff value in parameter CUTOFFLM. When the P1 value is below CUTOFFLM, it is replaced by zero. You can eliminate this feature by specifying NaN in CUTOFFLM.

Target-Value Flags

The target value can be specified by an operator or a user-written program that is stored in AVTV. This feature can be disabled by storing NaN in AVTV.

When the accumulated value in PVCALC is equal to or greater than AVTV, the target-value-reached flag, AVTVFL, goes to On, indicating that the accumulation is complete.

Even if the accumulator has stopped, this check is made on each processing pass.

You can specify two other trip points in AVDEV1TP and AVDEV2TP, as deviations from AVTV. Each of them is associated with a flag:

AVDEV1FL trips when

```
PVCALC ≥ AVTV - AVDEV1TP
```

AVDEV2FL trips when

```
PVCALC ≥ AVTV - AVDEV2TP
```

When the PVAUTOST of the accumulated value is Bad, AVTVFL, AVDEV1FL, and AVDEV2FL are all Off.

7.7.7 Bad-Input and Warm-Restart Options

You can configure equations A through F for this algorithm, but instead of specifying the calculation, they specify combinations of the following five options:

- **Use Zero**—When the accumulator is running, if the input status P1STS (or P2STS for Pulse IOP) is Bad, the input value is replaced by zero and the accumulation continues with a PVAUTOST of Uncertn.

  When the input status is Normal, PVAUTOST remains Uncertn until a reset command is received. No special action by the operator is required.

- **Use Last Good Value**—When the accumulator is running, if the input status is bad, the input value is replaced by the last good value and the accumulation continues with PVAUTOST as Uncertn. When the input status is Normal, PVAUTOST remains Uncertn until a Reset command is received. No special action by the operator is required.

- **Set PVAUTOST Bad and Stop**—When the accumulator is running and the input status is bad, the value in PVCALC becomes NaN, PVAUTOST goes Bad, and the accumulator is stopped. If the PVSOURCE is Auto, a bad-PV alarm is generated. When the input status is again Normal, PVAUTOST remains Bad until the accumulator is started again. To restart the accumulation, the operator should estimate its value and use the Reset command (see **Commands, States and Command Flags**) to establish that value, then use the Start command to restart the accumulation. The last accumulated value before the status went bad is in LASTPV.

- **Continue After a Warm Restart**—On a warm restart when the accumulator is running, the accumulation continues from the last PVCALC value. The PVAUTOST goes to uncertain (UNCERTN) and remains so until a Reset command is received.

  When the Pulse Input IOP is used (ACCTYPE = PULSE), accumulation continues with the accumulated value count (AV) obtained from the IOP. The totalizer assumes that exactly one rollover of the IOP count has occurred, if necessary. The PVAUTOST goes to UNCERTN and remains so until a Reset command is received.

- **Set PVAUTOST Bad and Stop After a Warm Restart**—On a warm restart when the accumulator is running, the value in PVCALC becomes NaN, PVAUTOST goes Bad, and the accumulation is stopped. The operator must intervene to restart the accumulator.

These options are selected as follows:

Equation	Bad Input Handling	Warm Restart
A	Use Zero	Continue
B	Use Last Good Value	Continue
C	Set Bad and Stop	Continue
D	Use zero	Set Bad and Stop
E	Use Last Good Value	Set Bad and Stop
F	Set Bad and Stop	Set Bad and Stop

If the accumulator is stopped, the input status is ignored. If the accumulator is stopped on a warm restart, no special action by the operator is required.

Restart or Point Activation

When the TOTALIZR data point is activated, the PVCALC value becomes NaN, PVAUTOST goes Bad and the accumulator state is Stopped. If the PVSOURCE is Auto, this causes a bad-PV alarm and the operator must re-establish normal operation.

The processing that takes place for a warm restart is described under **Bad Input and Warm-Restart Options**.

Error Handling

PVAUTOST is Uncertn when

- The input status (P1STS or P2STS) is Uncertn.

- The input status is Bad and the "use zero" or "use last value" (Equation A, B, D, or E) is configured (see **Bad-Input and Warm-Restart Options**).

- The data point is in a warm restart and the continue option (Equations A, B, or C) is configured (see **Bad-Input and Warm-Restart Options**).

A Reset command is needed to return PVAUTOST status to Normal, provided the input status is Normal.

PVCALC contains NaN and the PVAUTOST is bad when

- The input status is Bad and the "set bad and stop" (Equation C or F) is configured.

- The data point is in a warm restart and is configured for "set bad and stop" (Equations D, E, or F) is configured.

A Reset command is needed to return PVAUTOST to Normal, provided the input status is Normal.

7.7.7.4 Equations

Configure one of Equations A through F for a TOTALIZR data point equation specifying the operating bad-input and warm-restart options according to **Bad-Input and Warm-Restart Options.**

Analog Operation—

For all equations, when the accumulator is running, the accumulated value in PVCALC is calculated as follows:

$$\mathtt{PVCALC}_{(i)} = \mathtt{PVCALC}_{(i-1)} + \mathtt{C} * \mathtt{(TIME\text{-}SCALE)} * \mathtt{Pn}$$

Where

$\text{PVCALC}_{(i)}$ = The output of this algorithm from the current pass. It is selected as the PV for this data point when PVSOURCE is Auto.

$\text{PVCALC}_{(i-1)}$ = The accumulated value at the end of the last processing pass for this point.

C = The scale factor. Can be used to convert from eng. units to different eng. units. Default value = 1.0

(Time-scale) = TS*60 if TIMEBASE contains Seconds.
TS if TIMEBASE contains Minutes.
TS/60 if TIMEBASE contains Hours.

TS = The data-point processing interval in minutes.

Pn = The input value is a typical flow rate.

Pulse Input—See the discussion on pulse input operation.

7.7.7.5 Pulse Input Operation

If **pulse** is selected via parameter ACCTYPE, operation is the same, except input from the Pulse IOP is supported as described below (refer also to the Pulse IOP description in Section 2.7).

As Figure 7-9 indicates, P1/P2 input is from the Pulse IOP. Normally the P2 input (AV) is used (see PV/AV Selection). AV is the accumulated value from the pulse IOP. It is a 32 bit unsigned integer and used to determine volume. This method is more precise than if the Pulse IOP rate value is used. Error handling selected by Equations A through F reference the P2 input status.

The PV Totalizer algorithm contains a single user-configurable constant (C) —

PVCALC = C * DELTA_AV + PVCALC_LAST_SAMPLE

7.7.7.6 Configuration

When configuring the PV Totalizer algorithm for pulse operation, select ACCTYPE = PULSE. Then, define the pulse input source, P1SRC(1) to be the PV parameter of the PI IOP. The APMM software uses the entity specified for P1SRC(1) to establish the algorithm's second input (P2) except that it uses the parameter ID of **AV**.

7.7.7.6.1 PV/AV Selection

Either of the parameters AV or PV is available at the input of the Regulatory PV Totalizer algorithm. Normally AV is used, but only if all the following conditions are true—

The Pulse IOP is running
The Point is active
The IOP PV is not in lo cutoff
The IOP PV is not clamped
The IOP PV source is AUTO
There are no soft failures against the IOP slot

7.7.7.7 Parameters

Other parameters associated with the TOTALIZR algorithm are as follows (refer to the *Advanced Process Manager Parameter Reference Dictionary*):

P1STS (analog input)   P2STS (pulse input)   PVCALC   PVEQN
AVTV   ACCTYPE

7.7.8 General Linearization (GENLIN)

7.7.8.1 Function

This algorithm calculates a PV that is a function of the input. The function can be any that can be represented by up-to-12 continuous, linear segments. You specify the base value and slope of each segment. The input is compared with the input range of each segment and the output is set at the intersection of the input with the appropriate segment. See Figures 7-11 and 7-12.

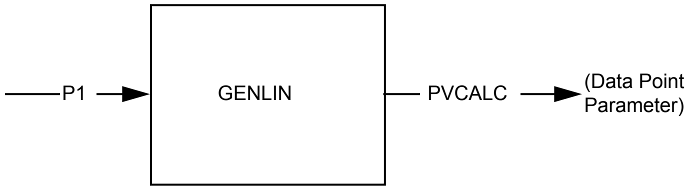


Figure 7-11 — Functional Diagram, General Linearization PV Algorithm

1307

7.7.8.2 Use

This algorithm is typically used to provide a linearized PV (in engineering units) for a sensor with a nonlinear characteristic. This algorithm can also be used to characterize functions of a single variable, such as heat transfer vs flow rate, or efficiency as a function of load. The algorithm is particularly useful when the relationship of the input to engineering units is empirically determined.

This algorithm supplements the standard linearization functions that are provided in the IOPs for standard temperature sensors and differential flow meters.

7.7.8.3 Options and Special Features

Restart or Point Activation

On a cold or warm restart, or when a data point using this algorithm is activated, PVCALC is recalculated the next time this data point is processed.

Error Handling

If P1STS is Uncertn, PVAUTOST status becomes Uncertn.

If P1STS is Bad or if any of the segment coordinates (IN_i or OUT_i) contains NaN, PVAUTOST becomes Bad.

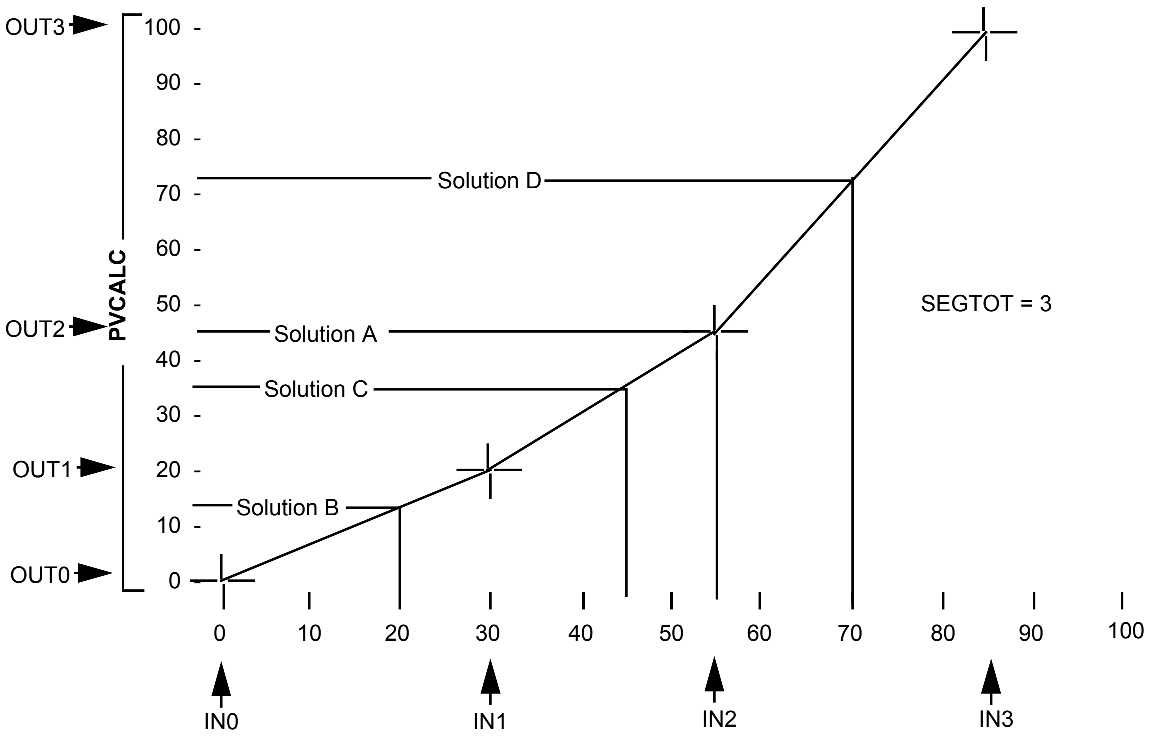
If any of the segment coordinate values (IN_i or OUT_i) contains NaN, a configuration alarm is generated.

Changing Parameters through a Universal Station

The SEGTOT, IN_i, and OUT_i parameters can be changed through a Universal Station only if the data point that uses the GenLin algorithm is made inactive.

Parameter—Value Restrictions

The input coordinate value parameters must be specified in ascending order from the smallest value to the largest.



IN0 = 0.0	OUT0 = 0.0	Beginning of 1st segment
IN1 = 30.0	OUT1 = 20.0	End of 1st segment
IN2 = 55.0	OUT2 = 45.0	End of 2nd segment
IN3 = 85.0	OUT3 = 100.0	End of 3rd segment

Solution A (P1 = IN2):

PVCALC = OUT2 = 45.0

Solution B (P1 > IN1):

$$\text{PVCALC} = \frac{\text{OUT1 - OUT 0}}{\text{IN1 - IN0}} * (\text{P1 -IN0}) + \text{OUT0} = \frac{20 - 0}{30 - 0} * (20 - 0) + 0 = 13.33$$

Solution C (P1 intersects any but 1st and last segment):

$$\text{PVCALC} = \frac{\text{OUT(i+1) - OUTi}}{\text{IN(I+1) - INI}} * (\text{P1 - INi}) + \text{OUTi} = \frac{45 - 20}{55 - 30} * (45 - 30) + 20 = 35.0$$

Solution D (P1 intersects the last segment):

$$\text{PVCALC} = \frac{\text{OUTsegtot - OUT(segtot - 1)}}{\text{INsegtot - IN(segtot - 1)}} * [\text{P1 - IN(segtot - 1)}] + \text{OUT(segtot -1)}$$

$$= \frac{100 - 45}{85 - 55} * (70 - 55) + 45 = 72.5$$

Figure 7-12 — Example of GENLIN Algorithm Operation

1308

Extension of First and Last Segments

The first and last segments are treated as if they indefinitely extended, so if P1 is less than IN0 or greater than INsegtot (see 7.7.8.4), PVCALC is computed by assuming that the slope of the appropriate segment continues to the intersection point.

7.7.8.4 Equation

Each time this algorithm is processed the input value P1 is compared with each segment, starting with the first and continuing until a segment is found that intersects with the input. When that segment is found, PVCALC is calculated as follows:

- If the P1 value is exactly equal to the input value at the beginning of any segment (P1 = IN_i, for i in a range from 0 to the value in SEGTOT),

$$PVCALC = OUT_i$$

- If P1 intersects the first segment ($P1 < IN_1$),

$$PVCALC = \frac{OUT_1 - OUT_0}{IN_1 - IN_0} * (P1 - IN_0) + OUT_0$$

- If P1 intersects any segment except the first one or the last one [$IN_i < P1 < IN_{(i+1)}$ for any i from 1 to segtot-2],

$$PVCALC = \frac{OUT_{(i+1)} - OUT_i}{IN_{(I+1)} - IN_1} * (P1 - IN_1) + OUT_1$$

- If P1 intersects the last segment [$P1 > IN_{(segtot-1)}$],

$$PVCALC = \frac{OUT_{segtot} - OUT(_{segtot-1})}{IN_{segtot} - IN_{(segtot-1)}} * [P1 - IN_{(segtot-1)}] + OUT_{(segtot-1)}$$

Where:

PVCALC = The output of this algorithm. It is selected as the PV for this data point when the PV source is AUTOmatic.

P1 = The input value.

$IN_{(i)}$ = Input value at the beginning of the intersecting segment.

$IN_{(i+1)}$ = Input value at the end of the intersecting segment.

$OUT_{(i)}$ = Output value at the beginning of the intersecting segment.

$OUT_{(i+1)}$ = Output value at the end of the intersecting segment.

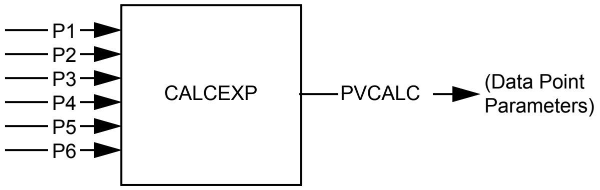
segtot = A subscript indicating the user-entered value in SEGTOT.

Other parameters associated with the GenLin algorithm are as follows (refer to the *Advanced Process Manager Parameter Reference Dictionary*).

P1STS PVCALC SEGTOT

7.7.9 Calculator (CALCULTR)

The calculator algorithm allows the user to write an equation to compute the PV and up to four intermediate results. The result from evaluating the expression is stored into PVCALC, which is then processed like any other PV algorithm. See Figure 7-13.



Accepts up to six inputs (POINT.PARAMETER).

Equation may be up to 40 characters long.

Up to 4 intermediate results

FORTRAN-like syntax rules

High Select; Low Select; Average Select and Middle of 3 Select support

Figure 7-13 — Functional Diagram, PV Algorithm CALCULTR 2097

7.7.9.1 Function

The equation is specified at the time of point building and is loaded from the DEB without additional steps such as compilation or linking. Up-to-six inputs can be configured and stored into the destination parameters P1, P2 . . . P6.

The following general guidelines apply.

- The equation can be up to 40 characters long.
- FORTRAN-like syntax rules apply.
- Up to 5 levels of nesting of expressions.
- Free format reals and mixed real and integer calculations permitted.
- Up to four intermediate results.
- The result of any expression that has no "equate" associated with it is stored into PVCALC.
- On point activation or warm restart PVCALC is initialized to the P1 input.

- The following operators are supported:

Operator	Associated Symbol
Divide	/
Multiply	*
Subtract	-
Add	+

- The following arithmetic functions are supported:

Function	Associated Symbol
Absolute	ABS
Square	SQR
Square Root	SQRT
Natural Logarithm	LN
Base_10 Logarithm	LOG
Exponent	EXP
Sine	SIN
Cosine	COS
Tangent	TAN
Arc-tangent	ATAN

- In addition, the following special functions are supported:

Function	Associated Symbol
High Select	MAX
Low Select	MIN
Average Select	AVG
Middle of 3	MID3

- Arithmetic assignment statement:

Equate	=

- Other

Separator	;

7.7.9.2 Use

This algorithm can be used to perform any calculation or arithmetic function on up to six inputs, using up to four intermediate results. Additionally is can be used as a selector algorithm as noted above (see also **7.7.9.3 Options and Special Features**).

7.7.9.3 Options and Special Features

- The user can configure up-to-six inputs, using the "Tagname.Parameter" format for PISCRC(n). The destination inputs can be assigned, using PIDSTN(n) for P1 through P6.

- Up to four intermediate results can be calculated, for example:

  ```
  C1=<expr_1>; C2=<expr_2>; C3 =<expr_3>; C4=<expr_4>;<expr_5>
  ```

  The result of expr_5 is stored in PVCALC (because it has no equate associated with it). There is no restriction on the order in which the sub equations and the expression for PVCALC are specified.

  If C1-C4 are expressions, they are recalculated every time the algorithm is processed.

- If it is necessary, or desirable, to key user configurable constants into the equation, C1-C4 can be useful. C1-C4 can then be entered or modified by an operator through the detail displays, or by logic slots, or user program.

- The equation can comprise up to 40 characters including the subequations.

- Can be loaded from the DEB without extra linking and compiling.

- FORTRAN syntax rules apply; up-to-five levels of nesting of expressions.

- HI, LO, and AVG functions may have any number of inputs including constants.

Error Handling of Bad-Inputs and Uncertain Values

If the calculated value of PVCALC is "bad," PVAUTOST is marked Bad. If the final value of PVCALC is a normal number, PVAUTOST is marked Uncertn if any input that is used in the calculation is uncertain or bad; otherwise it is set equal to Normal.

If the HI, LO, and AVG functions have bad inputs, they are ignored in the computation; if all inputs are bad, the result is marked bad. On MID3 function, if only one input is bad, the result is set equal to the average of the other two.

7.7.9.4 Equations

The equation can be up to 40 characters long. It is entered into the Parameter Entry Display in the port for the parameter CALCEXP. You can configure up to four intermediate expressions. The result of an expression not having an "equate" associated with it is stored in PVCALC.

Examples of use of this algorithm's equation:

```
(180.0/(.15*P1)) + (P2+P3*LOG(P4))

C1=P1*P2;C2=P4*MAX(0,C1,100);SQRT(C2*10)

MAX (MID3(P1,P2,P3), MID3(P4,P5,P6)

(P1 * P2)/C1
```

REGULATORY CONTROL POINT
Section 8

This section describes the functions available in the Regulatory Control point. The functions are described first, and then are followed by detailed descriptions of the algorithms. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

8.1 FUNCTIONS

Regulatory Control (RegCtl) points are used to perform standard control functions by executing the algorithms that have been configured. Each control algorithm includes a wide range of configurable options to allow implementation of complex control strategies by a simple menu-select process. For example, initialization and windup protection are inherently provided for all point interconnections. Also, the capability to ramp a setpoint (by operator entry of a target value and ramp time) is configurable. Predefined and custom displays are available to support these control strategies. These standard support functions significantly simplify the implementation and use of sophisticated multiloop control strategies. Increasingly advanced control strategies become practical as a result of the ease of APM configuration.

The RegCtl point supports the following algorithms:

PID (Pid)
PID with Feedforward (PidFf)
PID with External Reset Feedback (PidErfb)
Position Proportional (PosProp)
PID Position Proportional (PIDPosPr)
Ratio Control (RatioCtl)
Ramp Soak (RampSoaK)
Override Selector (ORSel)
Auto Manual (AutoMan)
Incremental Summer (IncrSum)
Switch (Switch)
Null (Null)

Detailed descriptions of these algorithms can be found in this section beginning with paragraph 8.13.

Figure 8-1 is a functional diagram of the RegCtl point.

8.2 CONTROL INPUT CONNECTIONS

Control input connections are typically used to assign "noninitializable" inputs to the configured control algorithm. They can be used to assign initializable inputs, but if initialization and windup protection are required, a control output connection from the primary point must be used.

The number of control input connections (NOCINPTS) and the destination parameter (CIDSTN) for each connection are defaulted based on the noninitializable inputs required by the control algorithm. (Refer to the individual control algorithm configuration form for the destination parameters within each algorithm.) The control input connection-source parameter CISRC allows the user to specify the source of the input using the "Tagname.Parameter" format.

The inputs can be obtained from any source parameter that is a real number, an integer, or a Boolean state. (Boolean states are treated as real numbers by the algorithm. The Off state is equal to a real number of 0.0; the on state is equal to 1.0). For all RegCtl algorithms except PIDERFB, the point's control output connections are also considered input connections because the RegCtl point reads the output status before writing to it. A RegCtl point can have a maximum of seven input and output connections that obtain inputs from I/O Processors in the same APM.

Generally, initializable inputs are stored with control output connections at the primary data point, and no input connections are required; however, in some cases it may be required to obtain initializable inputs with input connections (for example, when initialization and windup protection are not needed). To achieve this, the number of control inputs is allowed to be increased based on the initializable inputs of the control algorithm.

8.3 PV SOURCE SELECTION (Pid, PosProp, and RatioCtl Algorithms)

The source of the PV can be an analog input point, a PV algorithm, a Universal Station, or a user-written program. It is specified by the PVSOURCE parameter, whose value can be changed by an operator, a supervisor, an engineer, or a user-written program. The PVSOURCE parameter is configurable only if the data point has been configured as a full point. PVSOURCE can have one of the following values:

- **Auto**—PV is received from an analog input point or a PV algorithm. The value is in PVAUTO and its status is contained in PVAUTOST. During normal operation, the PV source is Auto, and the PV and its value status (PVSTS) become equal to PVAUTO and PVAUTOST, respectively, before PV range checks are made. When the PV source is changed from Auto to Man or Sub, the PV remains at the last value until it is changed by the operator (Man) or a program (Sub), so it doesn't "bump." In Man or Sub, the status in PVSTS is Uncertn. When the PV source is changed from Man or Sub to Auto, the PV immediately goes to the PVAUTO value. This might cause a bump in the value unless it is changed gradually to the value in PVAUTO before changing the source.

- **Man**—The PV is entered by an operator, supervisor, or engineer at a Universal Station.

- **Sub**—The PV is entered by a user-written program. A program can store a bad value in PV, and if it does, PVSTS becomes Bad.

You can prevent PV source changes by entering OnlyAuto for the PVSRCOPT parameter. This fixes the source as AUTO and the parameter PVSOURCE is removed from the point. Configuring All in PVSRCOPT allows normal PV source selection.

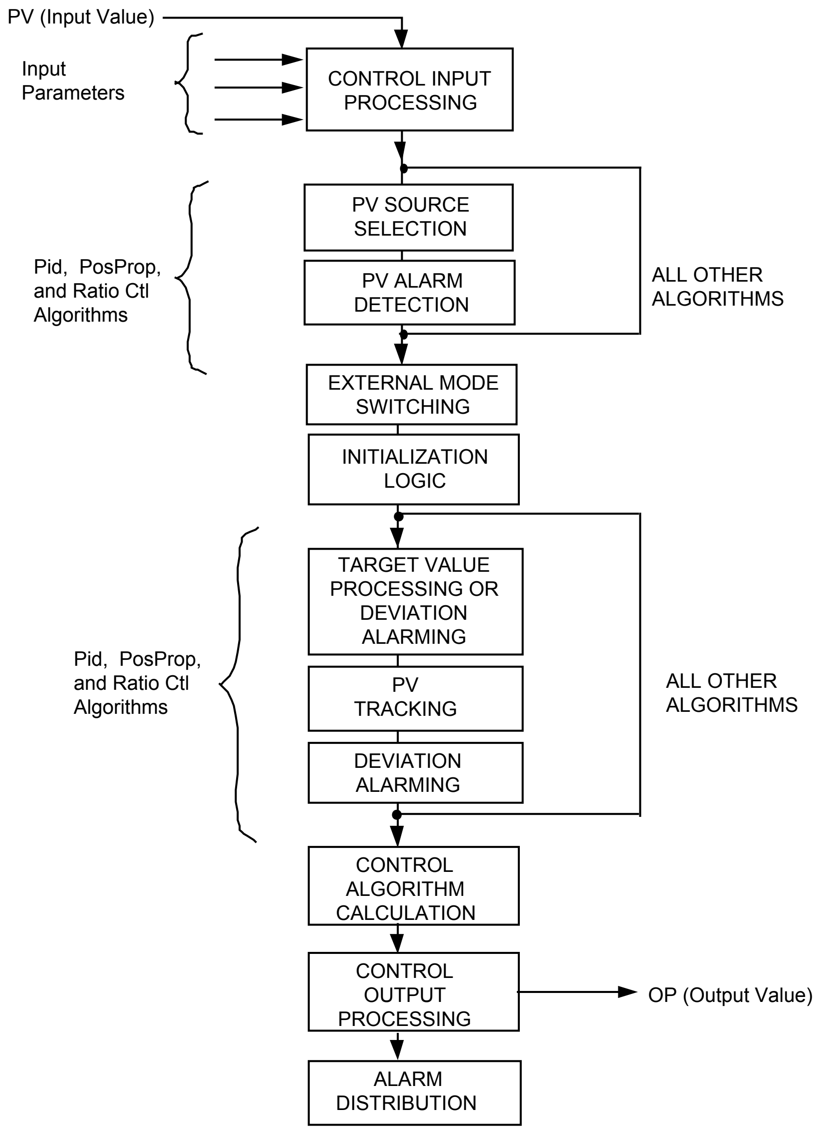


Figure 8-1 — Regulatory Control Point, Functional Diagram

8.4 MODES

The following operating modes are applicable to the RegCtl point (refer to Figure 8-2):

- **Manual (Man)**—provides the operator or the discontinuous program with direct control over the output value of the data point, regardless of any automatic control strategy.

- **Automatic (Auto)**—output value is computed by the configured RegCtl algorithm, and the setpoint comes from the local setpoint (LSP) location in the RegCtl point. An operator or a discontinuous program can change the setpoint value.

- **Cascade (Cas)**—data point receives its setpoint value from a primary data point.

- **Backup Cascade (Bcas)**—local cascade mode where the RegCtl point receives its setpoint from a primary data point even though the entry for the RCASOPT parameter is Spc, DdcRsp, or Rsp (where the AM provides the setpoint). In this way, should the AM or the NIM fail, the control strategy will shed to the local cascade mode.

Separate flags are provided to indicate if the current mode is manual (MANMODFL), auto (AUTMODFL), or if the mode attribute is Oper (OPRATRFL). The modes and mode attributes can be used in conjunction with the logic slots to implement interlocks.

8.4.1 Mode Attribute

The mode attribute denotes who has the authority to change certain parameters of a data point, and is established through parameter MODATTR. The mode attributes are as follows:

- **Operator**—Operator can supply the setpoint, output value, mode, ratio, and bias for a data point (operator-access level).

- **Program**—Program can supply the setpoint, output value, mode, ratio, and bias for a data point (program-access level).

- **None—**The mode does not have an attribute.

At a Universal Station, the mode attribute is displayed next to the mode of the data point. If the mode attribute is program, a "P-" is displayed to the left of MODE. If the mode attribute is Operator, blanks are displayed to the left of MODE.

An access level of Ccont (Continuous Control) allows supervisory control from an LCN module, such as the AM, to change the setpoint, output value, and mode of a data point.

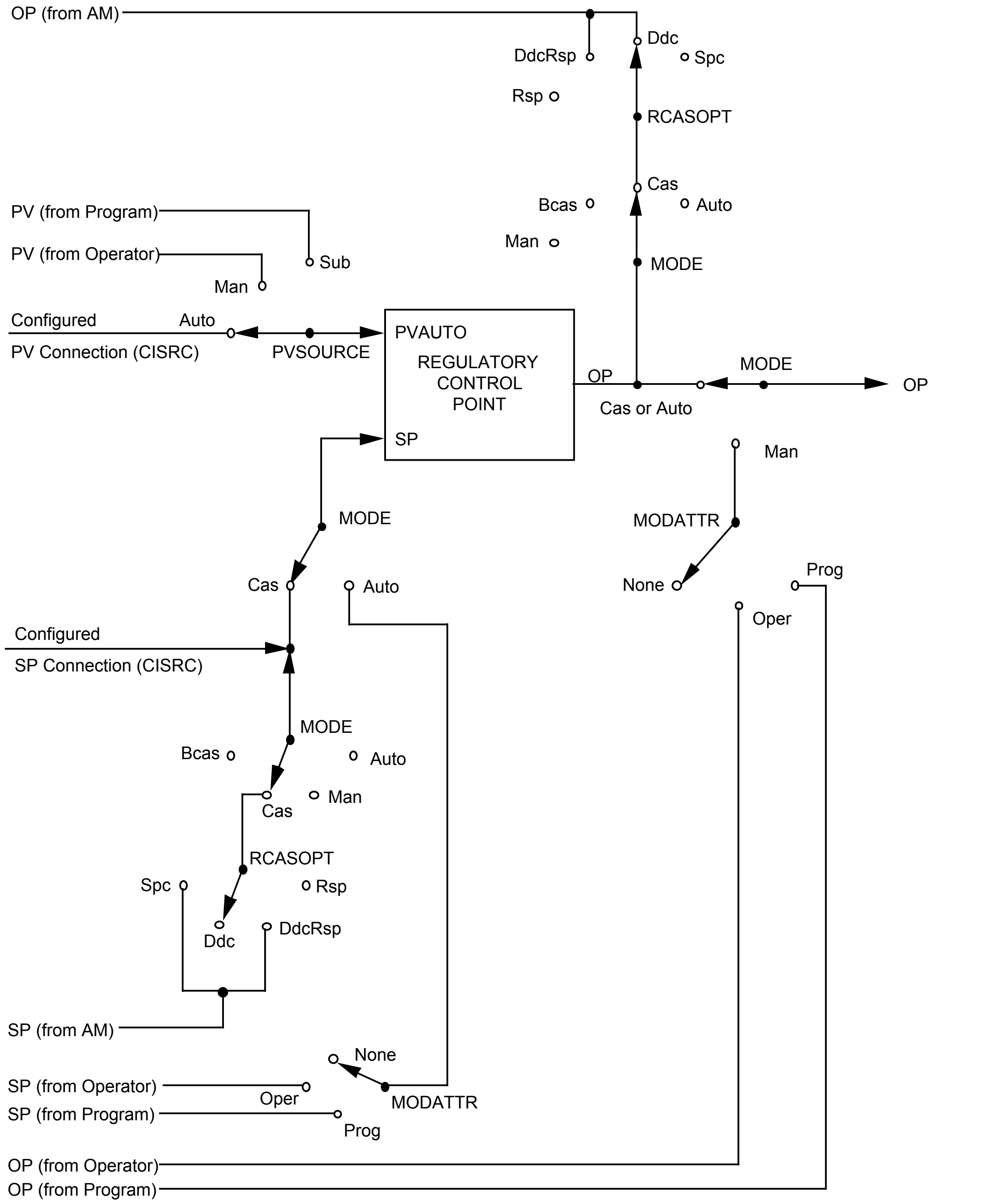


Figure 8-2 — Mode Structure

8.4.2 Normal Mode

The normal mode is the mode that is copied into the MODE parameter when the operator presses the NORM button at the Universal Station. Also at that time, the content of the NMODATTR parameter is copied into the MODATTR parameter. This then becomes the mode for the data point. The possible entries for the normal mode are None, Man, Auto, Cas, and Bcas.

8.4.3 Normal Mode Attribute

The normal mode attribute is the attribute that is copied into the MODATTR parameter when the NORM button is pressed. The possible entries for the NMODATTR parameter are Operator, Program, and None

8.4.4 Remote Cascade Handling (PID Algorithms)

The APM supports supervisory or DDC control for the PID algorithms from any remote device such as a computer (through the CG) or the AM. If the remote cascade connection is coming from a regulatory data point in the AM, it handles everything automatically; however, in all other cases (including the continuous CL programs in the AM directly writing to the APM), the user must ensure that the following conditions are handled:

- Remote device must use the continuous control access-level (Ccont) parameter for writing to the SP, OP, and MODE parameters.

- Remote device must also handle mode changes for closing the cascade.

- Initialization to ensure bumpless mode transfers.

- Windup protection.

8.4.4.1 Remote Cascade Options

The following remote cascade options are provided by the RCASOPT parameter:

- **Supervisory Control (Spc)**—This option is configured by setting RCASOPT to Spc. The AM control strategy writes to the setpoint of the RegCtl point, subject to the setpoint limits, when the RegCtl point is in Cas mode.

- **Direct Digital Control (Ddc)**—This option is configured by setting RCASOPT to Ddc. The AM control strategy writes to the OP output of the RegCtl point, when the RegCtl point is in Cas mode. The output limits are not applicable in this case. The SP cannot be written by the AM control strategy to the RegCtl point. If the control algorithm is a PID type and is configured for PV tracking, the SP is set equal to the PV when in Cas mode.

- **Direct Digital Control with Remote Setpoint (DdcRsp)**—This option is available for only PID-type algorithms. It is configured by setting RCASOPT to DdcRsp. The AM control strategy writes directly to the OP output of the RegCtl point when it is in Cas mode. The output limits are not applicable. The AM can also write to the SP of the RegCtl point subject to the setpoint limits. PV tracking is not performed in Cas mode. This option is used primarily when a single PID controller in the APM is used to back up a higher level control strategy running in the AM. The higher level control strategy writes to the OP and also provides the SP for the backup control strategy.

- **Remote Setpoint (Rsp)**—This option is available for only PID-type algorithms and is configured by setting RCASOPT to Rsp. The AM control strategy writes to the SP via an AM general output connection, subject to the setpoint limits, when the RegCtl point is in the Auto mode and it is being initialized by its secondary (i.e. INITMAN flag is On). PV tracking is not performed in Auto mode when INITMAN is On, if Rsp is selected.

The Rsp option is used primarily in a cascade control strategy where a RegCtl point is used to backup a higher level control strategy running in the AM, and the higher-level control strategy provides the SP for the backup control strategy. In the backup strategy, the primary PID would be configured for the Rsp option (allowing the AM to specify the backup SP), and the secondary would be typically configured for Ddc or Spc control.

8.4.4.2 Remote Cascade Request

When the RCASOPT parameter for the RegCtl point is configured for Spc, Ddc, or DdcRsp and the operator at the Universal Station or a discontinuous user program (e.g. a sequence program) tries to change the mode to Cas, the remote cascade request flag (CASREQ) is set to On, but the actual mode is left unchanged. The actual mode is changed to Cas only when

- The AM control strategy explicitly writes the mode to Cas (it should do this only after seeing a remote cascade request from the secondary), or

- If Spc is entered for the RCASOPT parameter, and the AM control strategy writes to the setpoint, or

- If Ddc or DdcRsp is entered for the RCASOPT parameter, and the AM control strategy writes to the output value OP of the RegCtl point.

After CASREQ has been set to On, any mode change causes CASREQ to be set to Off.

8.4.4.3 Remote Cascade Shed

The remote cascade shed mechanism in the APM allows the user to substitute the local backup control strategy in the APM for the AM control strategy if the AM or NIM fails.

When a RegCtl point that is configured for Spc, Ddc, or DdcRsp control is in Cas mode and the SP (if Spc) or the OP (if Ddc or DdcRsp) is not updated by the AM within a predefined time, the AM or the NIM is assumed to have failed. The backup control strategy is substituted by means of changing the mode to a preconfigured backup mode. The maximum time allowed between updates is specified by SHEDTIME (in seconds) and the backup mode is specified by SHEDMODE. Shedding to the backup mode can be disabled by setting SHEDTIME to 0.

When the mode is shed to the preconfigured shed mode, the remote cascade shed flag (RCASSHED) is set to On (mainly for indication at the Universal Station). Any subsequent mode changes automatically set the RCASSHED to Off. The remote cascade request flag is also set to On, which allows the AM to resume control at a later time without operator intervention.

8.4.4.4 Bad PV/Mode Shed

The Regulatory Control parameter BADCTLOP determines if the mode sheds to manual on detection of a bad PV (or bad CV for algorithms that do not have a meaningful PV). For the definition of a bad PV, refer to BADPVFL in the Parameter Reference Dictionary. This function does not apply to the RampSoak, IncrSum, or RatioCtl algorithms.

The BADCTLOP parameter is configured on a per point basis as shown in the following table:

Table 8-1 — Bad PV/Mode Shed

BADCTLOP Selection	Output	SHED To (Mode)	Mode Attribute	Ext. Mode Switching	Comments
No Shed	Held	Held	N/A	N/A	Point resumes control after initializing on PV recovery.
SHED HOLD	Held	Manual	Operator	Disabled	Point is not automatically returned to the previous mode before shed to manual. Operator is not allowed to return to the previous mode until the bad control condition clears.
SHED LOW	-6.9%				
SHED HIGH	106.9%				
SHEDSAFE	SafeOp				

8.5 SAFETY SHUTDOWN

This option allows the user to implement safety interlocks that effectively shut down a single control loop. When the shutdown flag (SHUTDOWN) is set to On by a user-written program, the mode and the mode attribute are changed to Man and Oper, respectively, and the OP output is set equal to a predefined safe output value (SAFEOP). Further, the external-mode-switching enable state (ESWENBST) is disabled, if it is currently enabled. As long as the shutdown flag is On, the MODE, MODATTR, ESWENBST, and OP parameter values cannot be changed. When the shutdown flag is set to Off, the control loop must be manually restarted.

A Logic Point or CL program must be used to reset the Safety Shutdown Flag (from ON to OFF).

If the point is already red tagged when the shutdown flag is turned On, the output value is not changed. Typically, the safe output value can be configured as 0% (if the valve is fully closed) or 100% (if valve fully open). In some cases it may be important to just hold the last value. This can be achieved by setting the SAFEOP parameter to NaN.

The shutdown flag cannot be initiated from the Universal Station.

8.6 EXTERNAL MODE SWITCHING

External mode switching (EMS) is typically used to establish mode interlocks or, under certain process conditions, to restrict the use of a mode that invokes a higher level of control. External mode switching is an option and can be configured by entering Ems for the EXTSWOPT parameter. Mode changes made through external mode switching have no effect on the mode attribute of the data point, or on the normal mode and normal mode attribute of the point. When a mode change is made, the last values of the normal mode and normal mode attribute parameters stay in effect.

Each regulatory data point has three parameters (ESWMAN, ESWAUTO, and ESWCAS) that are used as flags to indicate the mode to which the point has been switched.

When external mode switching is enabled by the operator or the program by setting the ESWENBST parameter to Enable, the operator and the program are prevented from changing the mode of the data point when the point is in the ESWMAN, ESWAUTO, or ESWCAS mode as indicated on Table 8-2; however, the operator or the program can disable external mode switching at any time by entering Disable for the ESWENBST parameter.

Table 8-2 — Mode-Switching Operation

		ESW		Parameters'	Can Mode Be
EXTSWOPT		**(Note 1)**		**Effective New Mode**	**Changed By**
= Ems	**MAN**	**AUTO**	**CAS**		**Operator or Program?**
No	X	X	X	Last Mode Requested	Yes
	F	F	F	Previous Mode (Note 2)	
Yes	T	X	X	Man	No
	F	T	X	Auto (Note 3)	
	F	F	T	Cas (Note 3)	

Notes :
1. MAN is ESWMAN, AUTO is ESWAUTO, and CAS is ESWCAS, T = True, F = False, and X = don't care.
2. Previous Mode is the mode when the point was last processed.
3. If the effective new mode is not legal for the control algorithm, the mode doesn't change.

8.7 SETPOINT HANDLING

Setpoint handling takes place only if the configured control algorithm requires a setpoint. Setpoint-handling functions are as follows:

- Setpoint (SP) limits
- SP Target-Value processing
- PV Tracking
- Deviation Alarming
- Advisory Deviation-alarm processing
- Ratio and bias options
- Setpoint Restrictions

8.7.1 Setpoint (SP) Limits (Pid, PosProp, PIDPosPr, and RatioCtl Algorithms)

Setpoint limit parameters SPHILM and SPLOLM prevent setpoint values from exceeding user-configured high and low limit values. These limits are configured in the same engineering units as the SP and must be within the SP range plus extensions. Crossover of setpoint limits is not allowed. The configured setpoint limits also apply to the advisory-target value.

Setpoint limits are observed in initialization calculations. If the limits are violated by the SP value, antireset windup-status propagation is invoked.

8.7.2 SP Target-Value Processing (Pid, PosProp, PIDPosPr, and RatioCtl Algorithms)

This option allows a Universal Station operator or a user-written program to "ramp" the setpoint from the current value to a new value over a period of time. The option is configured through the Data Entity Builder by entering TV in setpoint option parameter SPOPT. If an operator is to ramp the SP, the mode attribute in MODATTR must be Oper; and if a user-written program is to ramp the SP, the mode attribute must be Prog.

To use the SP target-value option, the operator

1. Enters the desired new SP value in SPTV.

2. Enters the ramp time (in minutes) in RAMPTIME.

NOTE

Step 1 or 2 causes the TVPROC parameter to go to Preset if the point is in the Auto mode, INITMAN is Off, and TVPROC is Off.

3. Enters Run in TVPROC.

4. The SP begins moving linearly toward the new value and the value in RAMPTIME decreases with time. When RAMPTIME = 0, SP reaches the new value and the status in TVPROC changes to Off.

TVPROC can be changed to the Run state only from the Preset state. While TVPROC contains either Preset or Run, SP high and low limits, and the SP high and low engineering-unit ranges can't be changed.

The following applies to TVPROC if it is in the Run state:

1. If MODE is changed to Man, state goes to Preset.
2. If MODE is changed to Cas, state goes to Off.
3. If INITMAN is true, state goes to Preset.
4. A store to SP forces the state to Off.
5. If SPTV > SP and the antireset-windup ARWNET parameter indicates that SP is wound Hi or HiLo, the SP stops changing. When ARWNET indicates that SP is no longer wound Hi or HiLo, SP ramping continues form the stop position. Note that when SP is ramping, ARWNET is not shown on the Group or Detail Displays. SP can normally be inferred from the output windup status. ARWNET can be accessed from a custom display.
6. If SPTV < SP and the output becomes wound Lo or HiLo, the SP stops changing.
7. If none of the above is true, the SP is ramped to SPTV.

8.7.3 PV Tracking (Pid Algorithms)

PV tracking is configured by entering Track for the PVTRACK parameter. During PV tracking, SP is set equal to PV whenever the cascade is broken by an operator or a program action, or the RegCtl point is a secondary (in a local cascade strategy) and the cascade is momentarily interrupted by a 1-shot initialization.

PV tracking occurs under the following conditions:

- Mode is Man.
- Mode is Cas and RCASOPT is Ddc.
- INITMAN is On, and either the mode is not Auto, or RCASOPT is not Rsp.
- RegCtl point is being processed for the first time after becoming active.
- RegCtl point is a secondary within a local cascade-control strategy (inside the same APM), and it is going through a 1-shot control initialization. One-shot control initialization occurs when
  - the control initialization-request flag CTRLINIT is On
  - the point is being processed for the first time after the APM state has changed to Run
  - just recovering from a bad PV
  - this slot has only one disposable secondary that just underwent 1-shot initialization.

Note that PV tracking (even if configured) is not done on return from a Bad PV.

8.7.4 Deviation and Advisory Deviation Alarming

8.7.4.1 Deviation Alarming (Pid, PosProp, PIDPosPr, RampSoak, and RatioCtl Algorithms)

Deviation (PV-SP) high and low alarms can be configured by using the alarm trip points provided by the DEVHITP and DEVLOTP parameters. In addition, alarm priorities can be established for these trip points by using the DEVHIPR and DEVLOPR parameters. Parameters DEVHIFL and DEVLOFL are used as flags to indicate that a deviation (high or low) alarm has been detected. This alarm returns to normal when the deviation (PV-SP) is less than or equal to the configured trip point minus a deadband equal to 10% of the trip point value.

8.7.4.2 Advisory Deviation Alarming (Pid, PosProp, PIDPosPr, and RatioCtl Algorithms)

This option allows an operator to manually change the SP to a predetermined value. The predetermined value is usually calculated by a user-written program that stores the value in advisory setpoint parameter ADVSP rather than storing it directly in SP. Advisory-deviation alarming is selected by entering Asp for setpoint option parameter SPOPT.

This alarm type is available if the RegCtl point has been configured as a full point. When this option is selected, an alarm is generated if the difference between the PV and the value in ADVSP is greater than the trip-point value in ADVDEVTP. This alarm returns to normal when the difference between the PV and ADVSP is less than or equal to the value in ADVDEVTP minus a deadband equal to 10% of the trip-point value.

The following conditions must be true if advisory-deviation alarming is to function:

- SPOPT = Asp
- ADVDEVTP < > NaN
- ASPROC = Enable
- PV alarm status in PVVALST < > Bad
- Alarm-enable status in ALENBST < > Inhibit

If the advisory-deviation alarm is present and the value of one of the above-listed configuration parameters is changed, the advisory-deviation alarm is reset.

If parameter ASPPROC = Disable, the value in ADVSP equals the value in SP.

8.7.5 Bad Output Alarm

The Regulatory Control point's Bad Output Alarm option (when enabled) generates a process alarm if one or more of the configured output connections to Analog Output or Digital Output points are broken. Broken means data cannot be pushed to a point. This alarm is cleared when all connections to the AO or DO points are good, or when the Bad Output Alarm Option is disabled.

8.7.6 Ratio and Bias Options (Pid Algorithms)

NOTE

If SP target value processing or advisory deviation alarming is configured for a RegCtl point, the ratio/bias options cannot be configured for the same data point.

The ratio and bias options are configured by entering one of the following values in ratio and bias option parameter RBOPT:

- **FixRatBi**—Fixed Ratio and Bias
- **AutoRat**—Auto Ratio and Bias
- **AutoBi**—Fixed Ratio and Auto Bias

If one of these options is configured, the SP is modified before being used by the PID algorithm as follows:

```
SP_Store_Value*RATIO + BIAS
```

Where SP_Store_Value is the setpoint before the modification.

You can configure limits for both the RATIO and the BIAS values in the following parameters:

- **RTHILM**—Ratio high limit
- **RTLOLM**—Ratio low limit
- **BSHILM**—Bias high limit
- **BSLOLM**—Bias low limit

In normal operation (in Cas mode and INITMAN is Off), all three options work alike. RATIO and BIAS can be changed by a Universal Station operator or by user-written programs (depending on whether MODATTR contains Oper or Prog). During initialization of this point, however, RATIO and BIAS can be changed only if they are not being initialized, based on the value in RBOPT, as follows:

- **FixRatBi**—The initialization value calculated for this point's primary is

  RINITVAL = (SP - BIAS)/RATIO

- **AutoRat**—RATIO is initialized as follows:

  RATIO = (SP - BIAS)/Store Value

If RATIO attempts to go outside one of its limits, it is clamped at the limit, and INITVAL is calculated as follows:

INITVAL = (SP - BIAS)/RATIO

- **AutoBi**—BIAS is initialized as follows:

  BIAS = SP - (Store_Value*RATIO)

If BIAS attempts to go outside one of its limits, it is clamped at the limit, and INITVAL is calculated as follows:

INITVAL = (SP - BIAS)/RATIO

8.7.6.1 Bias Limits

Two user-setable Hi and Lo limits on the SP-related bias parameter are provided. Whenever operator or program entries are outside the limits, the entries are clamped to the closest limit. Crossover of limits is inhibited.

You can configure high and low limits for the BIAS value that can be applied to the setpoint for PID algorithms. A Universal Station user with a Supervisor key can change these limits. An operator is prohibited from entering a value exceeding these limits. A user-written program is clamped to the exceeded limit. Crossover of these limits is prohibited.

8.7.6.2 Ratio Limits

You can configure high and low limits for the RATIO value that can be applied to the setpoint for Pid Algorithms. A Universal Station user with a Supervisor key can change these limits. An operator is not allowed to enter a value that exceeds these limits. If a user-written program attempts to store a value outside the limits, it is clamped to the limit. Crossover of these limits is prohibited.

8.7.7 SP Access Restrictions

The activities that can write a value to the SP are defined by Table 8-3.

8.8 SP/OP TOLERANCE CHECK

Release 530 provides a new function called "SP/OP Tolerance Check." This functionality has two new parameters called $SPTOL and $OPTOL that allow a SP (setpoint) and OP (output) tolerance value to be configured by the engineer.

Manually entered SP and OP values for the AM, HG, and NIM Regulatory Control points, OP values for HG and NIM Analog Output points, and OP values for HG Analog Composite points are checked against this new specified tolerance. If the tolerance is violated in either a plus or a minus direction from the current set value, the operator is alerted with a beep from the keyboard and a warning message. Operator confirmation is required before the value is stored.

The tolerance check is made from the Detail Display, Group Display, and in schematic actors RS_SYS, CHG_ZONE, and USER_CZ.

8.9 ALARM DETECTION

Alarms for a RegCtl point can be configured only when the point has been configured as a full point. The following alarm types are supported by the RegCtl points:

- PV High/Low*
- PV High-High/Low-Low*
- PV Rate-of-Change Positive/Negative*
- PV Significant Change*
- BadPV*
- Bad Control
- Deviation High/Low
- Advisory deviation

For detailed alarm information, refer to paragraph 4.3 in *System Control Functions*.

* Applicable to only Pid, PosProp, and RatioCtl algorithms.

8.10

8.10 CONTROL OUTPUT PROCESSING

The primary task of control output processing is to make the control algorithm calculation available to the rest of the system (displays, printers, CL programs, other data points, etc.) in percent or EUs, as needed. Control algorithms produce outputs in percent or in engineering units.

Control output processing performs the following functions:

- Provides the output value in percent and EUs for displays, printing, CL programs, and for interpoint communications.

- Processes control output connections that send the output to the secondary data point after conversion to EUs.

- Constrains the output to the configured limits.

- Generates and propagates windup status as a result of a violation an output limit.

The following parameters contain significant output-processing information. All of them are accessible for displays and printing and all can be accessed by a program written in CL.

CV = The result (calculated value) of calculation of the control algorithm, can be in percent or EU depending upon the control algorithm.

OP = The final control output, expressed as % of EU Range of the output.

OPEU = Final control output in Engineering Units.

CVEUHI = Output EU Range corresponding to 100% value of OP.

CVEULO = Output EU Range corresponding to 0% value of OP.

8.10.1 Initial Control Processing

During initial control processing, initialization data is fetched from the secondary points indicated by each control output connection.

The output engineering-unit range in CVEULO and CVEUHI is determined, based on the EU range of the secondary pointed to by the first active control output connection. If this connection has a communication or configuration error, the output range is set to bad and the point is aborted. If this happens, the CI connections and CO connections, scheduled for execution after initial control processing are not processed.

Table 8-3 — Setpoint Access Restrictions

MODE	BOX-IDLE or POINT INACTIVE or INITMAN	RCASOPT ATTRB	PID ALGO.				NON-PID ALGO.	
			PVTRK = ON		PVTRK = OFF			
			MODE ATTRB		MODE ATTRB	MODE ATTRB	MODE ATTRB	
			OPR	PRG	OPR	PRG	OPR	PRG
MAN	No	Any	I	I	Opr	Prg*	I	I
	Yes	Any	I	I	Opr	Prg*	I	I
AUTO	No	Any	Opr	Prg*	Opr	Prg*	Opr	Prg*
	Yes	None	I	I	Opr	Prg*	I	I
		SPC	I	I	Opr	Prg*	I	I
		DDC	I	I	Opr	Prg*	I	I
		DDCRSP	I	I	Opr	Prg*	——	——
		RSP	RC	RC	RC*	RC*	——	——
CAS	No	None	PC	PC	PC	PC	PC	PC
		SPC	RC	RC	RC	RC	RC	RC
		DDC	I	I	PC	PC	I	I
		DDCRSP	RC	RC	RC	RC	——	——
		RSP	PC	PC	PC	PC	——	——
	Yes	None	I	I	NCH	NCH	I	I
		SPC	I	I	NCH	NCH	I	I
		DDC	I	I	NCH	NCH	——	——
		DDCRSP	I	I	NCH	NCH	——	——
		RSP	I	I	NCH	NCH	——	——
BCAS	No	Any	PC	PC	PC	PC	PC	PC
	Yes	Any	I	I	NCH	NCH	I	I

NOTES: I => SP is being initialized, no change permitted.
Opr => Operator from group or detail display.
Prg => Discontinuous CL programs or sequence programs.
PC => APM cascade. Mainly the output connections from another control slot in this APM or another APM.
RC => Remote cascade. These include: continuous CL programs on LCN, output connections from regulatory points in the AM, or user computers on the LCN.
NCH=> No change permitted.
* => Target value processor state (TVPROC) must be off

Normal Computation

1. CV is calculated by the control algorithm.
2. OP is calculated from CV, with the units based on the range defined by CVEULO and CVEUHI. Note that the values in CVEULO and CVEUHI are determined by the engineering-units range of the secondary point to which the first active connection is made.
3. OP is checked for minimum output change, output rate-of-change, output high-limit, and output low-limit. If any of the limits is exceeded, the OP value is adjusted or clamped as applicable.
4. The windup status is set.
5. Each control output connection is processed as follows:

   If the control output connection is disposable,

   The corresponding value in the secondary point is made equal to OP, if the destination parameter is OP, otherwise it is made equal to OPEU.

   The OP or OPEU value is transferred to the secondary's destination parameter, subject to applicable limits on the destination parameter (e.g., SP limits).

In MANual Mode

1. OP is set (in percent) by an operator or by a user-written program, depending on the attribute.
2. Each control output connection receives normal processing, as described above. Remaining process is as defined under "Normal Computation."

When initializing because of a request from a secondary data-point

1. If all of the control output connections are indisposable, CV is initialized to its last value. When the first disposable control output connection becomes available, the CV is back-calculated (% to EU conversion if the destination parameter is OP) from the destination parameter of the secondary.
2. OP is calculated from CV, based on CVEULO and CVEUHI.

8.10.2 Output High and Low Limits

You can configure high- and low-limit values for the RegCtl point output using parameters OPHILM and OPLOLM . These limits are expressed as a percentage of the output range and the limit values can range from -6.9% to 106.9%. A Universal Station user with a Supervisory key can change these limits. Crossover of these limits is not permitted.

When an output limit is reached or exceeded, windup status is propagated up through cascade strategies.

8.10.3 Output Rate-of-Change Limits

You can configure a maximum rate of change in percent-per-minute for output values of RegCtl points using the OPROCLM parameter. The effect of this limit is to reduce excessive rates of change in the output, to the limit. The smallest limit can't be less than 0.1 percent per point-processing interval. The default value for the limit is NaN, which eliminates the limit check.

Typically, the output rate-of-change limit is used to match the slew rate of the final control element to the control dynamics.

Use caution when setting the value of OPROCLM. This value should be set before loop tuning has taken place. When done this way, tuning accommodates any slow down in response time caused by rate limiting. If OPROCLM is changed after a loop has been tuned, it is possible for poor loop dynamics or even instability to result.

Note that when a choice is available, rate limiting should be applied to PID algorithms (PID, PIDFF, and PIDERFB) rather than regulatory control algorithms of other types. The PID algorithms support special processing to prevent windup of the CV during rate limiting. Other algorithms cannot provide this special processing.

8.10.4 Output Minimum-Change Limit

You can configure a minimum output-change value for RegCtl points using parameter OPMCHLM. This value is a percentage of the output-value range. If the absolute difference between the output value at one processing pass and the next doesn't equal or exceed the minimum change, the earlier value is maintained. A Universal Station user with a Supervisory key can change the minimum output-change value.

The default minimum output-change value is NaN, which eliminates the minimum change check.

This feature is used to minimize "wear and tear" on the final control device.

8.10.5 Output Limiting in Manual Mode

No output limits, output rate-of-change or output minimum-change limits are observed while in manual mode. An indication is given to the operator if a manually entered or program-entered output value exceeds the output limits. Output values written by CL programs are clamped to the HI and LO output limits.

8.10.6 Control Output Connections

Control output connections are used to establish initializable cascade connections between the output of the RegCtl point output and other points. Control output connections are accomplished through parameter CODSTN. To assign the RegCtl output to the parameter of another data point, the user can choose from one of the following two output connection conventions (Tagname.Parameter or hardware reference address):

Tagname.Parameter

or

!MTmmSss.Parameter

where: MT is the IOP type (AO or DO)

mm is the IOP number in the APM file, from 1 to 40

ss is the slot number from 1 to 8 for analog outputs, and from 1 to 16 for digital outputs

Parameter is the parameter in the AO or DO point to which this output value is to be written.

Up to four output connections can be configured for a RegCtl point. Initialization and wind-up protection are supported for multioutput configurations.

The RegCtl data point can write the output value to only the following destination parameters:

- SP, RATIO, X1, X2, X3, and X4 parameters of another RegCtl data point in the same APM. When a control output connection is made to the RATIO parameter, the ratio high and low limits on the secondary are used as its engineering unit range. Also the secondary should be configured for a ratio bias option of auto ratio and placed in Program Cascade mode for the initialization to work correctly.
- X1 parameter of a RegCtl data point in another APM or HPM on the same UCN.
- SP parameter of a RegCtl point in another APM, PM, or HPM on the same UCN.

NOTE

An output connection of this type counts as one input and one output for the purpose of limiting the number of input/output connections going across APMs.

- OP parameter of an analog output point in the same APM. The component form of the output connection can be used to connect to an AO point that has been configured as a component point. The following restrictions apply:
  - The total number of control input and output connections fetching data from the I/O Processors must not exceed seven (prefetch limit updated in R600).
  - The analog output slot must be configured as a component point.
  - The output destinations cannot be a mix of IOPs and RegCtl points.
- OP parameter of a digital output point that has been configured for pulse-width modulation in the same APM. The following restrictions apply:
  - The total number of control input and output connections fetching data from the I/O Processors must not exceed seven (prefetch limit updated in R600).
- Any addressable parameter of any accessible slot if the configured control algorithm is PidERFB.

CAUTION

Up to four output connections are permitted from a Regulatory Control point but the control output destinations must be all IOPs (AOs or pulse width modulated DOs) or all Regulatory Control inputs. The outputs cannot be a mix of IOPs and Regulatory Control inputs.

8.10.7 Output High/Low Alarms

With R510 and later software, an Output High or Output Low alarm is set when a Regulatory Control point's output value (OP) exceeds the configured high or low alarm limit. The alarm is removed when the value of OP returns toward normal past a configurable deadband.

The following parameters deal with the Output Alarms and are configurable for each point:

OPHITP/OPLOTP contain the Output High/Low Alarm trip points

OPHIPR/OPLOPR contain the Output High/Low Alarm priorities

OPHAFL/OPLAFL contain the Output High/Low Alarm flags

OPALDB contains a deadband for the Output Alarms

The output alarms are available only for Regulatory Control points configured as full points and are disabled if the trip points are not specified.

8.11 INITIALIZATION

Initialization provides meaningful initial values in the data point parameters before processing is started or restarted. Separate mechanisms are provided to initialize PV-related parameters and control-related parameters.

8.11.1 PV Initialization

The principal purpose of PV initialization is to set up starting values the first time the point is processed or the first time it is processed after recovering from a BadPV value status. PV initialization is useful for only functions involved in history collection or for dynamically varying values. Stated another way, PV initialization is required where the new value depends on the previous value.

The following events cause PV initialization:

- Point's execution state is changed to Active.
- APM undergoes a warm or cold restart.
- PVAUTOST recovers from Bad value status.

PV initialization consists of the following functions:

- PVCALC is calculated from the PV inputs, using the steady-state portion of the equation. For calculations that don't involve time, the normal equation is used.
- PVAUTO is made equal to PVCALC. If PV filtering is configured, the filter dynamics are initialized to steady state.
- No other PV processing is affected by PV initialization.

8.11.2 Control Initialization

Control initialization allows normal control strategies to be re-established after they have been interrupted without "bumps" in the output to the process, and without the need for manual balancing of values to avoid such bumps.

Control initialization compensates for changes that may have occurred since the normal control strategy was last operating. For example, a Universal Station operator might have taken over control of the output to the process, so that it now has a value that is different than normal processing would calculate. The initialization procedures automatically readjust either the bias value in the data point(s) or an input to the data point(s) so that when normal control is reestablished, the output to the process does not move or "bump."

For the control algorithms, the new value is back-calculated for an input that absorbs any output change. This value and an initialization request are sent to the primary data point that provides the input. Thus, the primary absorbs the change and it must take similar action with its own primary, if it has one, so that the whole strategy can absorb the change.

By configuring a control output connection from one point (primary) to an initializable input to another point (secondary) an initialization path is created. It is along this initialization path that a value is transferred for use by a primary to absorb external process-upsets that may have occurred at the secondary.

Two or more active paths from a single primary to multiple secondaries are referred to as "fan out" connections. Where there are two or more control output connections from a primary to two or more secondaries and all of these outputs are indisposable, the primary goes into the initialization state.

The value that is to be protected from a bump (the value to back-calculate from) is obtained at the point's output or at the secondary's initializable input. When "fan out" connections to more than one output are used, the Regulatory Control output is initialized from the first disposable secondary.

General Mechanism—At each point-processing pass, the following information is retrieved from all secondaries to which the data point has control output connections. The data that is retrieved to support initialization is

- Initialization Request
- Initialization Value

An initialization request from the secondary causes the control output connection on the primary to go to an "output indisposable" state, a condition where a newly generated output to the secondary has no effect on the secondary. A control output connection also has "output indisposable" status if an error has been detected when the initialization request and initialization value should have been received.

When all connections from a primary are in the "output-indisposable" state, the primary is forced to the initialized state. This causes the primary to set its initialization request. In turn, this may cause the control output connection of a preceding primary to assume the "output indisposable" state and may force that primary to be initialized as well. In this manner, the initialization state is propagated upstream to all interconnected primaries.

During the initialization state, each point remains in that state until a disposable output connection is found. For points with one control output connection, the primary sets its output value to the initialization value received from its secondary while in the initialized state. For points with more than one control output connection, on the first processing pass when a disposable output connection is available, the primary sets its output value to the initialization value received from its secondary.

When initialization is being performed for a control strategy, each data point's output is readjusted by back-calculating an initializable input or by an internal (bias) adjustment. When each point resumes normal calculations, the new output and the input value at its secondary are balanced and no bump occurs.

Indisposable outputs are not the only factor for triggering initialization. It is possible to force a back-calculation by writing to a point's output while it is in MANual mode, or a user program or a logic slot can set the control initialization flag in a point. For these cases, the point does not go to the initialization state but its primary may, because the point sent an initialization request to the primary.

Special handling is provided for connections to local I/O (e.g., Analog Outputs or Digital Pulse Width Modulated Outputs). In the event that communication with the local output module is lost, the mode automatically goes to MANual to permit the operator to take control. When communication with the module is restored, back initialization to the output value occurs automatically. The operator can then return the strategy to the desired mode. This handling is triggered by loss of communications on the I/O link, failure of the I/O processor, or an I/O processor-detected power outage.

Conditions that Cause Control Initialization—The need to initialize a data point is indicated by external upsets that directly affect the point or it is indicated by an initialization request from a secondary data point.

Control initialization is caused by any of the following:

- A user program or logic slot has requested initialization (see "Initialization Forced by a Program," below).
- The point is active for the first time (an inactive to active transition).
- The point is executing the first time after a warm APM restart.
- All control output connections were indisposable and now one or more output connections is disposable.

A control output connection is indisposable when

- A secondary has made an initialization request, or
- A communication/configuration error has been detected in retrieving an initialization request and initialization value from a secondary.

The following are the reasons why a secondary data point sends an initialization request to its primary data point:

- The secondary isn't in Cas mode, or
- The secondary is inactive.
- The initializable input to the secondary (the destination of the primary's control output connection) is not selected for a secondary that uses the Switch algorithm (configured for PV tracking) or the input is being bypassed by the Override Selector algorithm, or
- The secondary is in the initialization state.

Initialization of a Point in the Same APM—For a primary and a secondary that are both in the same APM, an enhancement is provided. This enhancement presents the operator with an immediate indication of initialization when a cascade strategy is opened or closed.

Initialization Forced by a Program or Logic Slot—A user-written program or a logic slot can cause a data point to initialize by setting the data point's control initialization-request flag (CTRLINIT). The next time the point is processed, it initializes. CTRLINIT is cleared when the processing pass is complete. An initialization request is sent to its primary, thus propagating initialization up through the control strategy.

Limit Checking During Control Initialization—Limit checks apply during initialization except for output rate-of-change limits.

How Initialization is Indicated at Universal Stations—When a data point that is initializing is displayed, the display indicates that it is initializing. For example, on a group display or a detail display, INIT appears in the point-status field that is just below the mode indicator.

8.12 WINDUP PROTECTION

PID algorithms are protected from windup caused by reset action. Windup status parameters are maintained that pass the status "upstream" to the primary points along the initialization path. Each PID algorithm checks its output windup status and takes appropriate action to prevent reset windup. These functions are standard and require no configuration by the user.

8.12.1 Windup Status

The following parameters contain windup-status information:

- **ARWOP**—Output (OP) windup status
- **ARWNET**—Windup status for SP or another initializable input.

When this point's ARWOP contains something other than Normal, integral control in the windup direction stops. Integral action in the other direction and P and D action continue.

For the remainder of the control subsystem, the windup status serves only as a warning, and not as a constraint. For example if the status in ARWNET is Lo, lowering SP won't have an immediate effect on the output of the final secondary; however, SP can be lowered if the SP low-limit has not yet been reached.

The values in the windup-status parameters indicate whether raising or lowering the associated parameter value will affect the output of the final secondary, as it should.

The values for parameters ARWNET and ARWOP are as follows:

- **Normal**—Free to move in either direction
- **Hi**—Free to move only in the lower direction
- **Lo**—Free to move only in the upper direction
- **HiLo**—Not free to move in any direction

8.12.2 Status Propagation

Windup status is propagated to ARWOP and ARWNET of the same point, and then from ARWNET of the secondary point to the primary point, and so on.

Propagation from secondary to primary is instantaneous if both are in the same process unit and the same APM. Otherwise, it takes place on the next processing pass for each point.

8.13 OVERRIDE CONTROL

Override control strategies are primarily used for multivariable constraint control. In such a strategy, control of a single process variable is based on one of two or more PVs. These multiple PVs are referred to as "constraints." A selector algorithm selects the PV to be used for control. When an different PV is selected, the former PV has been overridden—the new PV has constrained the others.

Consider the following examples:

- In typical **boiler-control strategy**, the constraint on the fuel flow (SP of the fuel-flow PID controller) can be that it cannot exceed the actual air flow, multiplied by a ratio (which may be computed by the O_2 controller). In this case, the fuel flow is the primary PV to be controlled and the actual air-flow PV is the "constraint."
- In some **heating applications** it may be desirable to control the temperature of the feed (the primary PV) as well as possible, without ever letting the temperature of the hottest part of the pot (or the heat exchanger) exceed a safe limit. In this case, the safe limit on the pot temperature is the "constraint."
- In an **oven temperature-control** application, the temperatures can be measured at several places in the oven, and it may be desirable to control all these PVs by controlling one valve. One PID controller can be used for each PV, and the PID controller representing the PV that is farthest from its SP can be allowed to control the valve. In this case, there is no primary PV, because all the PVs have the same importance. Each PID controller represents a constraint on the other PID controllers.

Most often the objective is to achieve the best possible control of a PV without violating any of the constraints. The manipulated variable is driven by the output of an Override Selector algorithm that selects the highest or the lowest of up to four inputs. A PID that is in a cascade strategy (but is not selected) is prevented from winding up with the help of override initialization. In the rest of this section, the term "O/R selector" is used to mean an Override Selector control algorithm that is configured for external initialization. Refer to the detailed description of the Override Selector algorithm in this section.

8.13.1 O/R Status and Feedback

Regulatory data points contain the following parameters to support O/R strategies:

PTORST—This parameter contains the O/R status of the point as follows:

NotCon The point is not connected to an O/R selector. Strictly, it means that this point is not on an initializable path to an O/R Selector or it is now being initialized. PTORST defaults to this value.

Sel The point is a part of an O/R control strategy and is now selected

NotSel The point is part of an O/R control strategy and is not selected by the O/R selector.

When the point is returned from inactive to active status, when it undergoes a cold start, or when it is initialized, the status in PTORST becomes NotCon.

The following two parameters apply only to the Override Selector Control algorithm:

OROPT— When on, the feedback value is propagated to nonselected primaries of the override selector algorithm.

When Off, the feedback value is not propagated. However, nonselected primaries are kept from winding up by propagating a windup status opposite of the OrSel equation (Hi-Lo) in addition to the OrSel point windup status (ARWNET) to the nonselected primaries. This windup status propagation prohibits the non-selected primaries of the OrSel from winding up in the direction opposite of the OrSel selection equation.

OROFFSET — When On, PID output is initialized to: Feedback Value + Gain * Error. When Off, PID output is initialized to only the Feedback value.

8.13.2 Processing in an Override Control Strategy

Figure 8-3 is an example of an O/R strategy. The override portion of the strategy includes the Override Selector point and all points "upstream" from it. Here is how O/R processing works:

- There must be at least one PID controller in the O/R strategy. In Figure 8-3 the points named TAG-A, TAG-B, TAG-C, TAG-D, and TAG-E constitute the O/R strategy.
- All points upstream of the O/R selector are processed on normal cycles (highest slot index to lowest). In the example, the points can be processed in TAG-D, TAG-A, TAG-B, TAG-C order. Their PV and control algorithms are executed normally.
- The next point to be processed is the O/R selector (TAG-E in the example). It selects one input. Assume input X2 is selected.
- The O/R selector then propagates appropriate O/R status to each one of its own initializing primaries. It also propagates the O/R-feedback value to the nonselected, initializing primaries. In the example, TAG-E propagates O/R status of Sel to TAG-C because input X2 is selected, and O/R-status Notsel to TAG-B. Further, TAG-E propagates the O/R-feedback value to the nonselected, initializing primary, TAG-B. TAG-D does not receive the O/R status nor the feedback value because it is not an initializing primary.
- Each primary (provided it is in Cas mode), in turn, propagates O/R status to its own primaries (if any). It also propagates O/R feedback upstream, if it is not selected. In the example, TAG-B would propagate Notsel and an O/R feedback value to TAG-A.
- The propagation upstream continues until there are no more primaries. The output of any PID in a cascade chain, connected to a nonselected input of the O/R selector is initialized to override-feedback value, plus gain times deviation (PV-SP) if OROFFSET is On. If OROFFSET is Off, it is initialized to the override-feedback value. Because TAG–A contains a PID algorithm and is not selected, it undergoes O/R initialization.

The whole cycle is repeated.

- All points downstream of the O/R selector are processed at their specified interval. In the example, TAG-F is processed after TAG-E.

NOTE

The override status (Feedback Value) can be propagated to a maximum of five primaries.

If there are multiple O/R selectors in a strategy

O/R-feedback propagation is initiated by only the most downstream O/R selector. In the example, if TAG-F were also an O/R selector, the O/R strategy would consist of points TAG-A, TAG-B, TAG-C, TAG-D, TAG-E, and TAG-F. O/R propagation would be initiated by TAG-F and not TAG-E, as before.

Initialization in an Override Strategy

When a cascade is broken in an O/R strategy, initialization propagation supersedes O/R propagation. In the example, if point TAG-B is placed in Man mode, it doesn't propagate O/R status or an O/R-feedback value to TAG-A. TAG-A's O/R status then is Notcon.

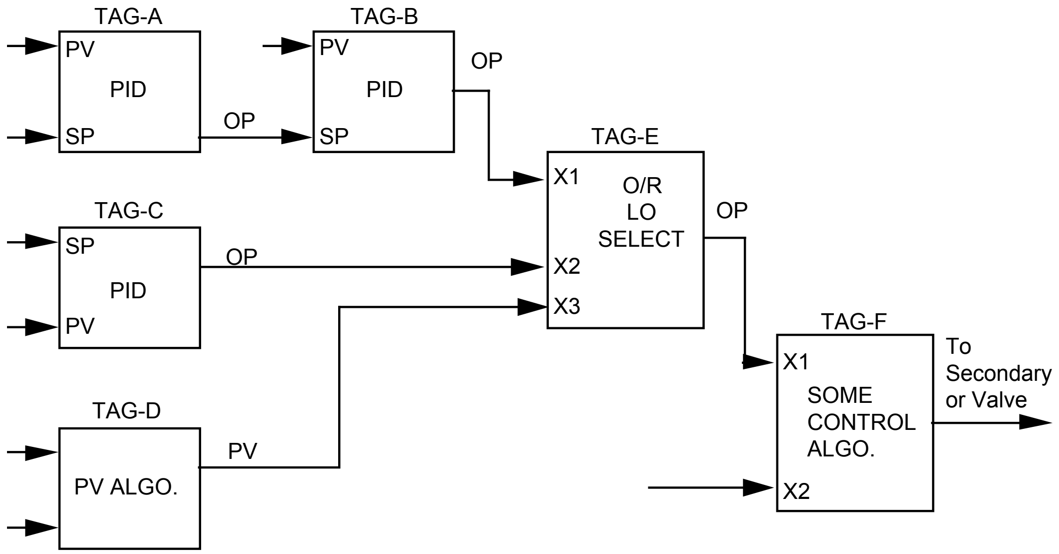


Figure 8-3 — Example of an Override Control Strategy 2070

8.13.3 Guidelines for Using Override Control

You should follow these guidelines to configure properly functioning O/R strategies:

- **Proportional and Derivative Action on PIDs**—While PIDs in an O/R scheme can be configured with proportional and derivative action on SP, use of these actions should be carefully considered because undesired results may occur, such as momentary oscillation caused by "kicks" in the error.
- **Boundaries**—The entire O/R strategy must be within the same APM. The number of primaries for initiating the O/R strategy must be ≤ 5.
- **Fanout**—No fanout control output connections are allowed in an O/R strategy. All primaries upstream from the O/R-selector point can have only one control output connection.

8.14 CONTROL ALGORITHMS

8.14.1 Proportional, Integral, Derivative (PID)

8.14.1.1 Function

This algorithm operates as a 3-mode (proportional, integral, and derivative) controller. You can choose one of two forms of this algorithm: the interactive (or real) form and the noninteractive (or ideal) form.

The output of this algorithm is normally "floating," because of the dynamics of the integral and derivative terms. Internally, the output is calculated as increments of output change, but the increments are accumulated to provide a full-value output, thus simplifying the techniques used to achieve "bumpless" outputs when modes or tuning constants are changed.

The algorithm operates to reduce error in the control loop to zero. Error is represented by the difference between the process variable in percent (PVP) and the setpoint in percent (SPP). The control-algorithm output value (CV) is also calculated as a percentage of the configured engineering-units range for the data point that uses this algorithm.

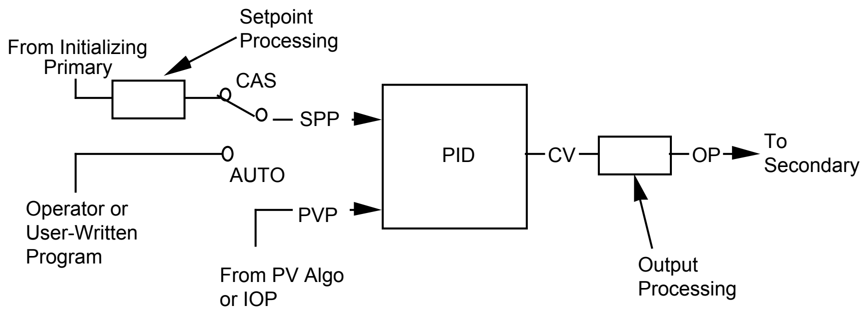
8.14.1.2 Use

The PID algorithm is used as a controller that either directly moves a control device (valve) in the process, or provides an input to another data point.

This algorithm requires only one input connection. The default for number of input connections is 1; however, it can be increased to 2, allowing the SP to be fetched with an input connection. When the SP is fetched, the normal operating mode of the point is usually affected.

When the APM's PID point is a primary for another data point in the same APM or another APM (or PM) on the same UCN, its output is connected to the SP of the other data point (via Tagname.Parameter). If the APM's PID point is directly controlling a valve, its output is connected to the output of a Analog or Digital IOP (through "Tagname.Parameter" or the hardware reference address !AOmmSss.OP or !DOmmSss.OP where mm is the IOP card number in the APM, and ss is the slot number of the output on the IOP card).

If the APM point is a secondary for another data point, it can be configured to receive an input from another source. Sources include another data point on the same UCN, an AM data point, or a point properly configured in another CM with proper access level. If the remote cascade connection is coming from a regulatory data point in the AM, it handles everything automatically.



PID Forms: Interactive (Real)
Noninteractive (Ideal)

Equations:
A: Full PID
B: PI on error, D on PV change only
C: I on error, PD on PV change only
D: Integral control only

Feedforward Action: Additive, Scale and Add
Multiplicative; Scale and Multiply

Figure 8-4 — Functional Diagram, PID Control Algorithm

1327

In all other cases (including the continuous CL programs in the AM directly writing to the APM), the user must explicitly take care to assure

- The remote device must use "continuous_control" access level for stores to SP, OP, and MODE parameters.
- It must also handle mode changes for closing the cascade. See 8.4.4, Remote Cascade Requests, for additional information.
- Initialization for bumpless mode transfers.
- Windup protection.

This algorithm supports all remote cascade options. PV source selection is supported if the point is configured "full." See 7.5, PV Source Selection, for details.

For additional information on restrictions on types and numbers of control output connections, see 8.10.6, Control Output Processing.

8.14.1.3 Options and Special Features

8.14.1.3.1 Interactive and Noninteractive PID Forms

During configuration, select one of these two forms. They differ as follows:

- Interactive (Real) Form—This form emulates traditional pneumatic-PID controllers. The P, I, and D terms are calculated as the sum of P and I, multiplied by D. D interacts in the time domain with the P and I terms. An advantage of this form is that the poles (lags) and zeros (leads) can be easily placed (see the equations under 8.14.1.4). The poles and zeros must be real.

- Noninteractive (Ideal) Form—In this form, P, I, and D are added in the time domain. D is added as a damped derivative to limit peak amplitude. This form is often called the digital-computer version of the PID controller.

8.14.1.3.2 Engineering Unit Ranges

The PV engineering unit range (PVEULO and PVEUHI) must be specified. The setpoint engineering unit range (SPEULO and SPEUHI) always follows the PV range. The output engineering unit range (CVEULO and CVEUHI) is derived from the secondary if the number of output connections is greater than 0, otherwise it must be specified.

8.14.1.3.3 Four Combinations of Control Terms

You select the combinations of proportion, integral, and derivative control terms by choosing Equation A, B, C, or D. The equations function as follows (also see 8.14.1.4):

- **Equation A**—all three terms (P, I, and D) act on the error (PV - SP).

- **Equation B**—The proportional and integral terms act on error (PV - SP) and the derivative acts on PV changes. This equation is used to eliminate derivative spikes in control action that occur with quick changes in the setpoint.

- **Equation C**—The integral term acts on error (PV - SP) and the proportion and derivative terms act on PV changes. This equation provides the smoothest and slowest response to setpoint changes.

- **Equation D**—This equation provides only integral control.

8.14.1.3.4 Control By a Single Term

When you use equation A, B, or C, the integral or derivative terms can be eliminated by setting their time constants to 0 (see 8.14.1.4). Setting both T1 and T2 to 0 results in only proportional control.

Use Equation D to achieve only integral control.

8.14.1.3.5 Direct and Reverse Control Action

When configuring a data point that uses the PID algorithm, you can select direct-control action or reverse-control action. You can also change the control action through the detail display if you have an engineer's key, or a user-written program can change the control action. The control action can be changed at the Universal Station or by a program, only while the data point is in Man mode. The attribute must be appropriate (Oper or Prog) for the change to be accepted.

Changing the control action effectively changes the sign of the gain. With direct action, an increase in PV increases output; with reverse action, an increase in PV decreases output.

As an example, with direct-control action, assume

SPP = 50%

PVP = 51%

Deviation = PVP - SPP = 1%

If PVP increases, the deviation (error) increases, so the output, CV, increases (see Equation A under 8.14.1.4).

The opposite occurs with reverse-control action: If the deviation increases, CV decreases.

8.14.1.3.6 PV Tracking

PV tracking is configured by setting PVTRACK to Track. If configured, SP is set equal to PV when the cascade is broken by an operator, a program action, or when this data point is a secondary in a local cascade strategy and the cascade is momentarily interrupted by one-shot initialization; that is, when the following conditions exist

- The data point that uses this algorithm is in Man mode.
- The mode of this data point is Cas and RCASOPT = Ddc
- INITMAN = On, and either the mode is not Auto, or RCASOPT is not Rsp
- The first time the data point is processed after becoming active.
- This slot is a secondary within a local (inside the same APM) and is going through 1-shot control initialization.

NOTE

The 1-shot control initialization occurs when

- the control initialization request flag (CTRLINIT) is On
- this slot is being processed for the first time after the APM state has changed to RUN (OK)
- this slot has just recovered from a bad PV
- this slot has only one disposable secondary that just underwent 1-shot initialization

PV tracking is typically chosen when the data point is a secondary in a cascade control strategy, because it allows the PID to resume control with no error, after the point has been in Man mode or is initialized.

PV tracking can also be used when the data point is the ultimate primary point. In such a case, a startup procedure could be used where the point is started in Man mode and the valve manually adjusted to bring the PV close to the desired value, and the data point would then be switched to Auto.

PV tracking (even if configured) is not done on return from a Bad PV.

8.14.1.3.7 Gain Options

When configuring a data point that uses the PID algorithm, and equations A, B, or C, you can choose any of the following four gain options:

- Linear Gain—This is the most commonly used gain option. The gain, K, used in the chosen equation (see 8.14.1.4) is set by the user. The default value for K is 1.

- Gap Gain Modification—This option is used to reduce the sensitivity of the control action when the PV is in a narrow band (gap) around the setpoint. The size of this band is specified by the user. K, as used in the chosen equation is derived as follows:

  $$K = KLIN * KGAP, \text{ if } (SP - GAPLO) \leq PV \leq (SP + GAPHI)$$

  or,

  ```
  K = KLIN, if PV is outside the gap.
  ```

  Where:

  KLIN = A linear-gain parameter, in percent-per-percent. The range of KLIN is 0.0 to 240.0, and the value is tuned at a Universal Station. Default = 1.0
  KGAP = Gain-modification factor, specified by the user. The range of KGAP is 0.0 to 1.0. Default = 1.0.
  GAPLO = The bottom limit of the gap in the same engineering units as the PV. GAPLO can be any value ≥ 0.0. Default = 0.
  GAPHI = The upper limit of the gap in the same engineering units as the PV. GAPHI can be any value ≥ 0.0. Default = 0.

- Nonlinear Gain Modification—This option provides control action proportional to the square of the error, rather than the error itself. The gain, K, used by the chosen equation, is derived as follows:

  $$K = KLIN * KNL$$

  $$KNL = NLFM + \frac{NLGAIN * (|PVP - SPP|)}{100}$$

  If the resulting value in K exceeds 240.0, it is clamped at 240.0.

  Where:

  KLIN = Same as for gap gain
  KNL = Nonlinear-gain modifier
  NLFM = Nonlinear-gain form. 0 or 1, as specified by the user. Default = 1.

For the ideal form of the PID, nonlinear gain does not act on the derivative component.

NLGAIN = Nonlinear gain, specified by the user. Value ranges from 0.0 to 240.0. Default = 0.
PVP = PV in percent
SPP = SP in percent

- External Gain Modification—The gain, K, used by the chosen equation, is modified by an input value that can be from the process, from a PV calculated from a process input by a PV algorithm, or from a user-written program.

  The main use of this option is to compensate for nonlinear-process gain. The user can tune the PID gain independently of the operating point of the process. For example, in controlling the level in a tank whose cross section is not constant, the gain could be modified to compensate for the nonlinear rate of level change that is caused by the changing shape of the tank. The General Linearization PV algorithm (subsection 7.9.8) could be used to compute the inverse of the level-change characteristic, and the resulting PV could be used to modify the level-control gain.

  K is derived as follows:

  ```
  K = KLIN * KEXT
  ```

  If the resulting value in K exceeds 240.0, it is clamped at 240.0.

  Where:

  KLIN = Same as for linear gain
  KEXT = The external gain-modification factor. It can be entered by a user-written program, or it can be a general input from another data point. KEXT must be a positive number. Default = 1.0.

  It is possible to use this option for multiplicative-feedforward control, but the PID with Feedforward-control algorithm (subsection 8.14.2) is a better choice because it provides a better operator interface and better recovery from a "bad" feedforward input.

8.14.1.3.8 Windup Handling

When the output of this algorithm reaches the user-specified output limits, or reaches the setpoint limits of the data point's secondary, or when a woundup-status indication is received from the secondary, the PID algorithm stops calculating the integral term but the calculation of the proportional and derivative terms continues.

This is the same way that windup conditions are handled in Basic Controllers, Multifunction Controllers, and Extended Controllers.

8.14.1.3.9 Suppression of Output "Kicks" When Switching to Cas Mode

Without this suppression feature, the first setpoint change after switching from Man or Auto to Cas mode could cause a sudden move (kick) in the output because of the proportional or derivative terms. This "kick" occurs when, for some reason, the primary data point's output is not initialized, and an abrupt change in the setpoint occurs when Cas mode resumes.

To suppress this "kick," the proportional and derivative terms are not calculated the first time the PID data point is processed after changing to Cas mode.

This feature is especially useful when the PID point is one of two or more secondaries of its primary data point. When this data point is changed to Cas mode, even if the primary is not initialized, the output of this data point does not bump the first time it is processed.

8.14.1.3.10 Initializing PID Output Without Affecting Dynamics

A logic slot or a user-written program in the APM, AM, or a CM (FORTRAN or Pascal) can store a value in the CV parameter of the PID data point, even while the algorithm is doing its normal, incremental PID calculation. This may change the full-value output of the data point, but it has no effect on the continuing incrementation or decrementation of the output. It is possible, therefore, for a user-written program to initialize a PID output without affecting the dynamics of the PID calculations, and without initializing the output of the primary data point.

As an example of the usefulness of this feature, consider a single PID that is controlling temperature by controlling the flow of either gas or oil. This PID's output is connected to both flow controllers, but only one secondary is in cascade at any time. When a change from one fuel to the other is made, the user-written program initializes the output of the temperature-controller PID by storing a new, full-value output in CV. The cascade connection is switched from one to the other, and the dynamic compensation of the flow of the new fuel proceeds. The value stored in CV is the setpoint of the new secondary in percent (SPP). Through this technique, the full-value output of the primary has been initialized without affecting its dynamic calculations, so the fuel switchover is quick and smooth.

8.14.1.3.11 Restrictions on Some Values

The following are restrictions on some of the values used with this algorithm:

- The engineering units range that you specify for the PV also applies to the SP.
- For best performance, we recommend that the integral (T1) and derivative (T2) time constants be within the following ranges:

```
T1 > 20.0 * TS

T2 > 100.0 * TS for the interactive form of the PID, and

T2 > 10.0 * TS for the noninteractive (ideal) form of the PID
```

Where TS = the interval at which the data point is processed, in minutes.

For the interactive form of the PID,

```
If T1 > 0 but < 2.0 * TS, it is clamped to 2.0 * TS

If T2 > 0.but < 10.0 * TS, it is clamped at 0.0
```

8.14.1.3.12 Bias Options

Ratio control can be achieved by modifying the setpoint input to the PID algorithm by a RATIO of some other process point that is stored to through a control output connection, for example, a fuel-to-air ratio in furnace control (it can also be accomplished with the Ratio Control algorithm. See 8.14.6). When configuring a PID data point, you can select one of the following options for modifying the setpoint through the RBOPT parameter:

- No ratio or bias
- Fixed ratio and bias
- Auto ratio (fixed bias)
- Auto bias (fixed ratio)

If you select one of the ratio and bias options, configured or operator-entered ratio and bias values are used to modify the setpoint (by multiplying it by the ratio and adding the bias value) only while the data point is in Cas mode. In Auto mode, the ratio calculation does not occur because this option is intended to receive the process value to be modified by the ratio, only from another data point (which can happen only in the Cas mode).

The "Auto ratio and Auto bias" options adjust the ratio or bias while the data point is in Auto or Man modes, or is undergoing initialization, so that when it returns to Cas mode, the new SP won't "bump" the process. The adjustment is as follows:

- For Auto ratio, the operator can change only bias, and ratio is calculated to maintain the same setpoint when the mode is changed to Cas. The operator can change the ratio in Cas mode.
- For Auto bias, the operator can change only ratio, and bias is calculated to maintain the same setpoint when the mode is changed to Cas. The operator can change the bias in Cas mode.

Modification of the setpoint by a ratio and a bias is actually handled by setpoint processing rather than by the PID algorithm. It is applied to only PID setpoints.

These options allow this one algorithm to perform PID Ratio, PID Auto Ratio, and PID Auto Bias functions.

The parameters used for these options are RBOPT, RATIO, BIAS, RTHILM, RTLOLM, BSHILM, and BSLOLM.

8.14.1.3.13 Operating Modes

The PID algorithm operates in the following modes:

- Man
- Auto
- Cas
- Bcas — If RCASOPT = Spc or Ddc or DdcRsp

8.14.1.3.14 Remote Cascade Options

All Remote Cascade Options are supported: Spc, Ddc, DdcRsp and Rsp.

8.14.1.3.15 Restart or Point Activation

On a warm restart, or when the data point is activated, initialization takes place as described under 8.14.1.5.

8.14.1.3.16 Error Handling

If the status of the PV value goes bad, the CV value is changed to bad (NaN) and the data point remains in the current mode. When the PV-value status returns to normal, the CV value is initialized and the PID dynamics are returned to a steady state. An initialization request and initialization value are sent to the primary data point.

8.14.1.4 Equations

You can select one of four equations when you configure a data point that uses the PID control algorithm. Equations A through D differ in the interactive and noninteractive forms of the algorithm.

For the Interactive form:

Equation A — P, I, and D act on the error

$$CV_S = K * [\frac{1 + T1 * s}{T1 * s} * \frac{1 + T2 * s}{1 + a * T2 * s} * (PVP_S - SPP_S)]$$

Equation B — P and I act on error, D acts on PV

$$CV_S = K * [\frac{1 + T1 * s}{T1 * s} * \frac{1 + T2 * s}{1 + a * T2 * s} * PVP_S - \frac{1 + T1 * s}{T1 * s} * SPP_S]$$

Equation C — I acts on error, P and D act on PV

$$CV_S = K * [\frac{1 + T1 * s}{T1 * s} * \frac{1 + T2 * s}{1 + a * T2 * s} * PVP_S - \frac{1}{T1 * s} * SPP_S]$$

Equation D — Integral control, only

$$CV_S = \frac{1}{T1 * s} * (PVP_S - SPP_S)$$

For the Noninteractive form:

Equation A — P, I, and D act on the error

$$CV_S = K * [(\frac{1 + T1 * s}{T1 * s} + T2 * s) * (PVP_S - SPP_S)]$$

Equation B — P and I act on error, D acts on PV

$$CV_S = K * [(\frac{1 + T1 * s}{T1 * s} + T2 * s) * PVP_S - \frac{1 + T1 * s}{T1 * s} * SPP_S)]$$

Equation C — I acts on error, P and D act on PV

$$CV_S = K * [(\frac{1 + T1 * s}{T1 * s} + T2 * s) * PVP_S - \frac{1}{T1 * s} * SPP_S]$$

Equation D — Integral control, only

$$CV_S = \frac{1}{T1 * S} * (PVP_S - SPP_S)$$

Where:

CV = Output of the PID algorithm, full value in percent

a = A constant equal to 0.1. 1/a is the high-frequency gain or rate amplitude.

K = Gain. See 8.14.1.3.7.

PVP = The process variable in percent

s = The Laplace operator

SPP = The setpoint in percent

T1 = The integral time constant in minutes per repeat for interactive form. **See 8.14.1.3.11, Restrictions on and Recommendations for Some Values.**

T2 = The derivative time constant in minutes. **See 8.14.1.3.11, Restrictions on and Recommendations for Some Values.**

8.14.1.5 Initialization

When the output destination of the primary is to the secondary, initialization occurs as follows:

- When the control initialization request flag (CTRLINIT) is on
- When this slot is being processed for the first time after the APM state (APMMSTS) has changed to Run or OK
- When this slot is just recovering from a Bad PV
- When this slot has only one disposable secondary that just underwent 1-shot initialization
- When the output is indisposable (when the mode of the secondary slot is not in Cas)

8.14.1.6 Override Feedback Processing

When a PID point's secondary uses an Override Selector algorithm (see Figure 8.5), the following functions take place:

- When override feedback is propagated, override status is returned in PTORST to the PID point. The status is one of these.

  Not Connected
  Selected
  Not Selected

- When the PID point is processed, it does the following.

  If the status returned is Not Connected, there is no action.

  If the status returned is Not Selected and if the PID point's mode is Auto or Cas and CV is not equal to NaN, the PID point's CV is initialized as follows:

  IF OROFFSET = On, CV = ORFBSEC* + GAIN ∗ ERROR

  ```
  CV = ORFBSEC + K * (PVP – SPP)
  ```

  where K * (PVP - SPP) = offset value if the direct-control option is chosen, or

  ```
  CV = ORFBSEC – K * (PVP – SPP)
  ```

  if the reverse-control option is chosen.

  IF OROFFSET = Off

  ```
  CV = ORFBSEC
  ```

  where ORFBSEC is the override-feedback value (in %) sent to the PID from the secondary.

  When the PID point is not selected, if it is in Cas mode an override-feedback value is calculated as follows and sent to the primary:

  If RBOPT is not equal to NoRatBi

  $$\text{ORFB} = \frac{\text{PV - BIAS}}{\text{RATIO}}$$

  otherwise, ORFB = PV

Note that if equation D is selected, K = 1 in these equations.

*ORFBSEC is not an external parameter

The not connected/not selected/selected status received from the PID's secondary, is also sent on to the primary.

NOTE

If the offset value is in such a direction that it causes the nonselected PID to become selected, the offset values will be set to 0.0.

See 8.13, Override Control, for more information on override control.

8.14.1.7 PID Parameters

In addition to the parameters already mentioned, the following parameters are associated with the PID algorithm (refer to the *Advanced Process Manager Parameter Reference Dictionary*):

CTLEQN	PVTRACK	DEV	DELCV

Deviation (DEV) and incremental output (DELCV) are accessible parameters.

8.14.1.8 Control Output Connections

Control outputs can be to the following destinations:

- SP, RATIO, X1, X2, X3, and X4 parameters of another control slot in the same APM box.*

  When output connections are made to RATIO, the secondary mode should be Cas and MODATTR should be PROGRAM and RBOPT should be AutoRat.

- SP of a control slot in another APM (or HPM or PM) box on the same Universal Control Network.*

- X1 input of a control slot in another APM (or HPM) box on the same UCN.

  Note that such a connection counts as one output and one input for the purpose of limiting the number of UCN connections.

When output connections are not to any of the above, they can be to

- OP parameter of an analog output slot inside an AO IOP in the same APM box*. AO point must be configured as a component point.

- OP parameter of a digital output slot (configured for DOTYPE = PWM) in the same APM box.*

- Any addressable parameter of any accessible slot if the control algorithm is PID with external reset feedback.

Output does not go to a device on the Data Hiway (MC, EC, CB).

* See 8.10.6, Control Output Connections, for restrictions and additional information.

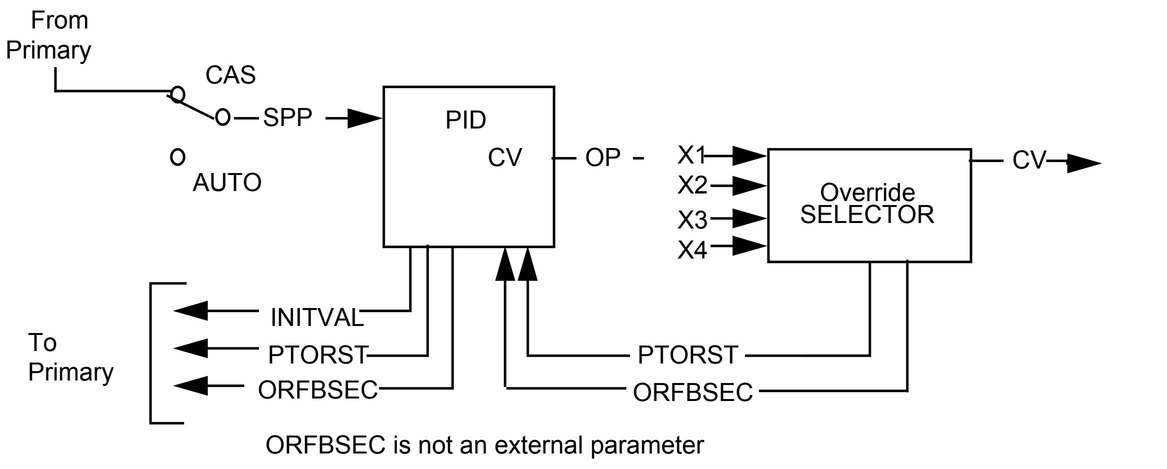


Figure 8-5 — Override Feedback Processing

8.14.2 PID with Feedforward (PIDFF)

8.14.2.1 Function

This algorithm operates as a 3-mode (proportional, integral, and derivative) controller. It is identical to the PID algorithm (8.14.1), except that it accepts a feedforward signal to be added to, or multiplied by, the algorithm's incremental output, before the full-value output is accumulated. This algorithm lets you combine a feedforward signal with the PID output without using another data point or algorithm to do it.

8.14.2.2 Use

The use of the PID Feedforward Control algorithm is the same as the PID algorithm, except that this algorithm can accept a dynamic feedforward signal from the process, or a value that is representative of some condition in the process, to be combined with the PID's incremental output before the full-value output is accumulated.

This algorithm requires two input connections (from PV processing and FF inputs). The inputs can be increased to three, allowing the SP to be fetched with an input connection.

The feedforward signal can be obtained from an analog-input point, and it is often subjected to dead-time compensation or lead-lag compensation before being connected to the FF input of this algorithm. That compensation can be provided by algorithms such as the Variable Dead-Time with Lead-Lag Compensation PV algorithm (7.7.6). Figure 8-7 shows an example of such a strategy.

If additive-feedforward action is chosen, the feedforward signal is multiplied by a user-specified scale factor (KF) and added to the incremental output of the PID computation. This scale factor might be used to convert an engineering-units input to a percentage.

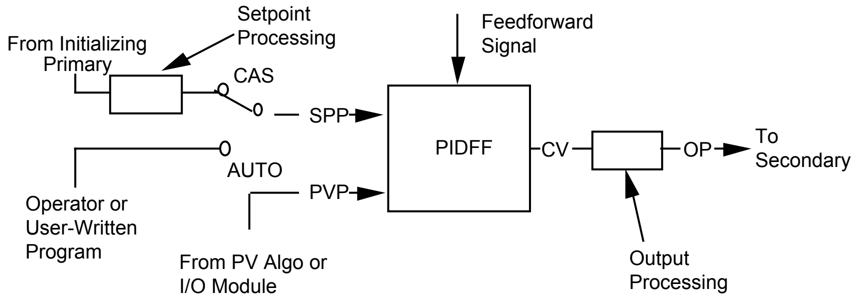
If multiplicative feedforward action is chosen, the feedforward signal is multiplied by the scale factor (KF) and then multiplied by the full value output of the PID computation. This action is typically used to compensate for variations in process gain that are caused by changes in throughput. For example, if the feed rate is doubled in a heating application, twice the amount of fuel might be required, which is the equivalent to doubling the process gain.

This algorithm supports all remote cascade options and PV source selection (if the point is configured as a full point).

8.14.2.3 Options and Special Features

All of the following PID Control-algorithm options and special features apply to the PID Feedforward Algorithm:

- 8.14.1.3.1 Interactive and Noninteractive PID Forms
- 8.14.1.3.2 Engineering Unit Ranges
- 8.14.1.3.3 Four Combinations of Control Terms
- 8.14.1.3.4 Control By a Single Term
- 8.14.1.3.5 Direct and Reverse Control Action
- 8.14.1.3.6 PV Tracking
- 8.14.1.3.7 Gain Options
- 8.14.1.3.8 Windup Handling
- 8.14.1.3.9 Suppression of Output "Kicks" when Switching to Cas Mode
- 8.14.1.3.10 Initializing PID Output Without Affecting Dynamics, except where multiplication of the feedforward signal is configured
- 8.14.1.3.11 Restrictions on Some Values
- 8.14.1.3.12 Ratio Control
- 8.14.1.3.13 Operating Modes; Man, Auto, and Cas
- 8.14.1.3.14 Remote Cascade Options
- 8.14.1.3.15 Restart or Point Activation
- 8.14.1.3.16 Error Handling



PID Forms: Interactive (Real)
Noninteractive (Ideal)

Equations:

A: Full PID
B: PI on error, D on PV change only
C: I on error, PD on PV change only
D: Integral control only

Feedforward Action: Additive, Scale and Add
Multiplicative; Scale and Multiply

1330

Figure 8-6 — Functional Diagram, PID Feedforward Control Algorithm

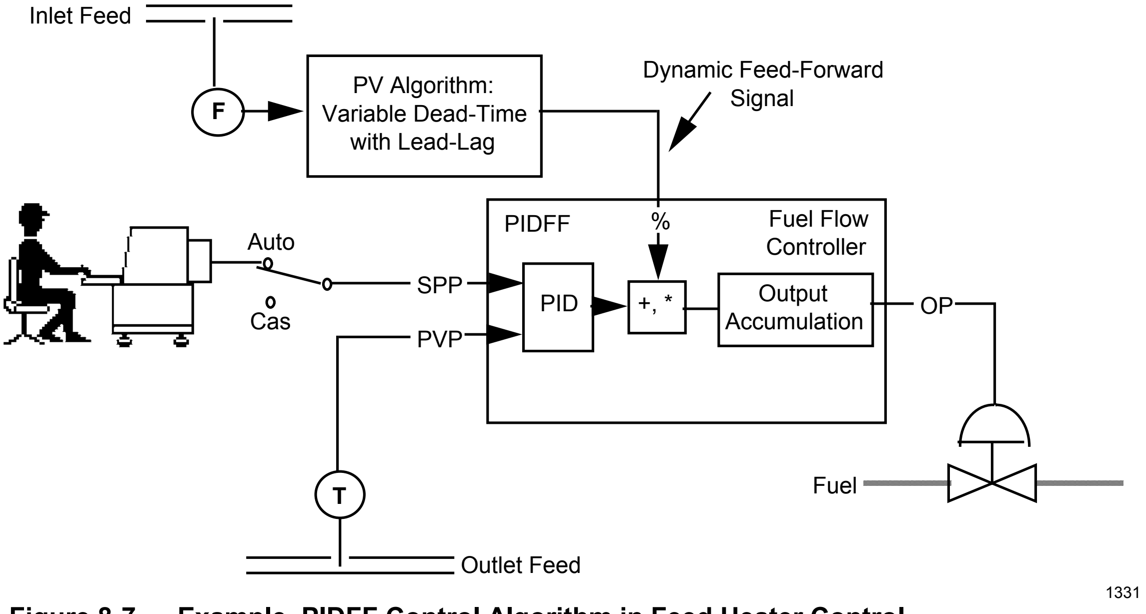


Figure 8-7 — Example, PIDFF Control Algorithm in Feed Heater Control

8.14.2.3.1 Add or Multiply Action

Parameter FFOPT is configured to specify whether the feedforward signal is to be added to the incremental PID output or multiplied by it.

8.14.2.3.2 Bypassing Feedforward Control Action

An operator at a Universal Station or a user-written program can bypass the feedforward action by one of the following:

- If the feedforward signal comes from a PV algorithm, switch the PV source for the data point that is using the PV algorithm to Man (if you do this and the PV is changed while the PVSOURCE is Man, the feedforward signal is affected). To resume feedforward action, switch the PVSOURCE back to Auto.

- If the feedforward signal comes from a control algorithm, switch the mode of the data point that is using the control algorithm to Man (if you do this and the output (OP) is changed while the source point is in manual, the feedforward signal is affected). To resume feedforward action, switch back to Normal mode (Auto or Cas).

8.14.2.3.3 Feedforward Signal Value Status

If the value status for the feedforward signal goes bad, the feedforward component of the output value is frozen at the last good value and normal PID processing continues.

When the value status of the feedforward signal returns to normal, normal feedforward action resumes. This does not cause a bump in the output because any change from the last good value is internally absorbed and the PID dynamics are not affected. The floating, full-value output continues as if there were no feedforward change, but the contribution of the feedforward action continues from that time.

8.14.2.3.4 Operating Modes

The PIDFF algorithm operates in the following modes:

- Man
- Auto
- Cas
- Bcas — If RCASOPT = Spc or Ddc or DdcRsp

8.14.2.3.5 Remote Cascade Options

All Remote Cascade Options are supported: Spc, Ddc, DdcRsp and Rsp.

8.14.2.4 Equations

You can select PID equations, just as described for the interactive form and the non-interactive form under 8.14.1.4.

In addition, the feedforward signal is applied to the incremental output of the PID computation, as follows:

- If additive action is configured

  $CV_n = CV_{n-1} + DELCV + KFF * (FF_n - FF_{n-1})$

  DELCV = Incremental output

  If the status of FF_n or FF_{n-1} is Bad,

  $CV_n = CV_{n-1} + DELCV.$

- If multiplicative action is configured

  CVPID = CVPID + DELCV

  $CV = CVPID * (KFF * FF_n + BFF)$

  When multiplicative action is configured, CV is a read-only parameter for CL programs.

  If the status of FF_n is Bad,

  CV = CVPID * (KFF * FF lgv + BFF),

  where FF lgv = last good value of FF.

  If FF_n is OK but the status of FF_{n-1} is bad,

  $$CVPID = \frac{CV}{(KFF * FF_n + BFF)}$$

  Note that the back calculation of CVPID keeps CV unchanged, and thus, prevents a bump.

  If the result of $(KFF * FF_n + BFF)$ is less than 0.01, it is clamped at 0.01.

  Where:

  CV = Full-value output in percent, PID combined with feedforward action

  CVPID = The full value output before the multiplicative term. This is an internal parameter and is not available to displays nor to user-written programs.

  DELCV = The incremental output of PID computation. Default = N/A.

  BFF = Bias value for multiplicative action. Default = 0.

FF = The feedforward input signal, from a control-input connection. Normally from a parameter with a percentage value. Default = N/A

FF lgv = Last good value for the FF input (notation only, not a user-visible parameter).

KFF = Scale factor applied to FF. Normally the source parameter is in units of percent. However, if it isn't, KFF can be used as an EU to percent conversion factor. In this case, KFF is set to:

$$\frac{100}{\text{EUHI - EULO}}$$

Where EUHI and EULO correspond to the EU range of the source parameter.

n and n-1 = Notation to indicate the value this pass (n) and the preceding pass (n-1).

8.14.2.5 Initialization

Initialization is as described under 8.14.1.5.

8.14.2.6 Override Feedback Processing

Override-feedback processing is the same as described under 8.14.1.6, except that, if multiplicative action is configured, a feedforward term is added to the output calculation, as follows:

If the status returned is not selected and if the PID point's mode is Auto or Cas and CV is not equal to NaN, the PID point's CV is initialized as follows:

If OROFFSET = On

```
CV = ORFBSEC + K * (KFF *  FF + BFF) * (PVP - SPP)
```

if the direct-control option is chosen, or

```
CV = ORFBSEC - K * (KFF *  FF + BFF) * (PVP - SPP)
```

if the reverse-control option is chosen.

ORFBSEC* is the override-feedback value from the secondary in percent

If OROFFSET = Off

```
CV = ORFBSEC + (KFF *  FF + BFF)
```

If direct-control option is chosen, or

```
CV = ORFBSEC - (KFF *  FF + BFF)
```

if reverse-control option is configured.

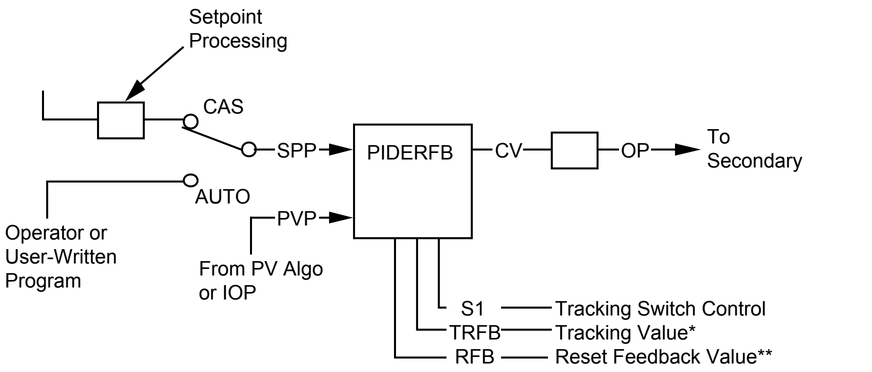
*ORFBSEC is not an external parameter

8.14.3 PID with External Reset-Feedback (PIDERFB)

8.14.3.1 Function

This algorithm operates as a 3-mode (proportional, integral, and derivative) controller. It is identical to the PID algorithm (8.14.1), except that it accepts a reset feedback signal to be combined with this algorithm's incremental output, before the full-value output is accumulated. It also accepts a tracking-value signal.

The intent of this algorithm is to prevent windup when it has a secondary data point, typically a PID point, that may or may not be responding to the output of this data point.



* Typically PV or SP of Secondary PID

** Typically PV of Secondary PID

PID Forms: Interactive (Real)
Noninteractive (Ideal)

Equations: A: Full PID
B: PI on error, D on PV change only
C: I on error, PD on PV change only
D: Integral control only

Reset Feedback Action: Scaled, integrated CV - RFB deviation is added to incremental PID output before full-value output accumulation.

4250

Figure 8-8 — Functional Diagram, PID with External Feedback Control Algorithm

8.14.3.2 Uses

The use of the PID with External Reset-Feedback algorithm is the same as the PID algorithm, except that this algorithm can accept a reset-feedback signal (RFB) from another data point, typically the PV of the secondary PID data point that is receiving its setpoint from this data point.

This algorithm requires three input connections:

- From PV Processing (PVAUTO)
- RFB (Reset Feedback Value) — typically from the PV of another data point which is receiving its setpoint from this data point
- TRFB (Tracking Value) — typically the PV or SP of another data point that is receiving its setpoint from this data point.

Output destination can be to a secondary data point in the APM or to any desired destination using control output connection (that is, any addressable parameter of any accessible slot).

The number of inputs can be increased to four, allowing the SP to be fetched with an input connection.

The tracking switch-control signal (S1) is usually stored with an output connection from a Logic slot (or a program). If the switch control is On, the CV value from this data point is replaced by the tracking value. If, for some reason, the secondary is not using the output of this data point, S1 can be set to On by logic external to the PIDErfb, which causes this point's CV to track the secondary's PV. When the secondary begins to accept OP from this point for control, S1 is set to Off (by the external logic), and CV is then at the same value as the controlled variable (PV), so there is no bump and normal control can resume.

In a simple application, both the reset-feedback signal and the tracking value may come from the PV of the secondary data-point. See Figure 8-9.

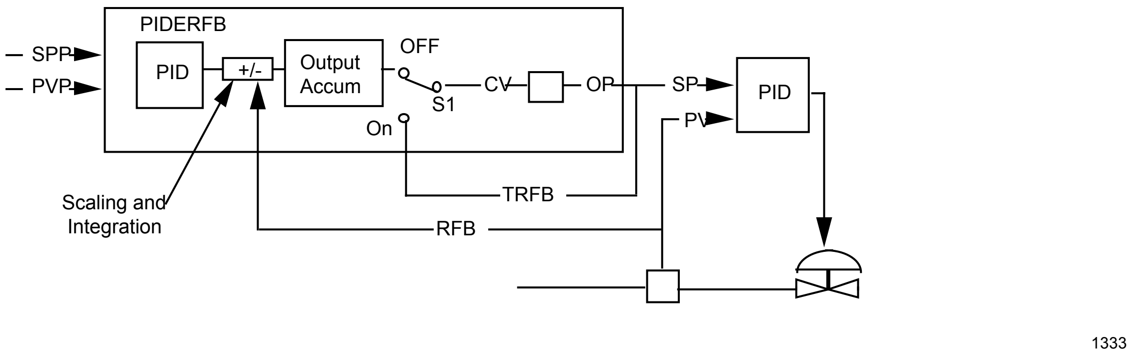


Figure 8-9 — Example of Application for PIDERFB

8.14.3.2.1 Use of Hand/Auto Stations

The following functions will be supported by the APM to integrate a third-party hand/auto station:

- OUTPUT Ability to manipulate the output of the hand/auto station (when in Auto position) from the PID, and to initialize the PID output to allow the hand/auto station to switch bumplessly to auto position.
- MODE Indication that the hand/auto station is in "hand" (or Local Manual) position at the group display of the PID driving the output.
- PV and SP Indication of PV and SP at the hand/auto station, and optionally the ability to raise or lower the PID SP from the hand/auto station. PV and SP may be from the PID driving the output, or from a different PID. For example, in a single loop case, the PV, SP and output all come from the same PID, but in case of a temperature flow cascade, the PV and SP come from the temperature controller, and the output is driven by the flow controller.

In order for the interface to be generic (to allow support for hand/auto stations from many different vendors), all signals are wired into the APM through analog and digital IOPs. This requires a total of two AI slots, one AO slot, and two DI slots in addition to the requirements if no hand/auto station support is desired.

8.14.3.2.2 Interface Requirements for Hand/Auto Station

The following I/O signals must be supported by the third-party hand/auto station:

- PV_IN: Analog input used to display the PV value on the hand/auto station. The signal from the analog transmitter is wired into this input as well as to the APM through an AI IOP.

- CV_IN: Analog input from the controller. It represents the control value position requested by the APM's PID.

- CV_OUT Analog output. This output drives the control valve. When the hand/auto station is in auto position, CV_OUT should track the CV_IN input; when in local manual position, it can be manipulated by the raise/lower keys.

  This output is also fed back to the APM through an AI IOP for initialization of the PID output. The corresponding AI FTA must not be grounded (the zero-ohm resistor to ground at the FTA should be removed). Also the output current drive capability of the hand/auto station must be sufficient to handle an additional 250 ohm resistor used to convert 4-20 milliamps to 1-5 volts.

- MAN_STS: Digital output indicating that the hand/auto station is in local manual position.

- SP_IN: Analog input used to display the SP value on the hand/auto station. The setpoint of the APM's PID controller is connected to this input through an AO IOP.

  NOTE: This is not required if SP raise/lower capability is not to be supported.

- SP_OUT: Analog output generated by the hand/auto station to change the SP value of the APM's SP.

- SP_CHNG: Digital output requesting the APM to accept a new SP (equal to SP_OUT) from the hand/auto station.

  NOTE: This is not required if SP raise/lower capability is not supported.

8.14.3.3 Options and Special Features

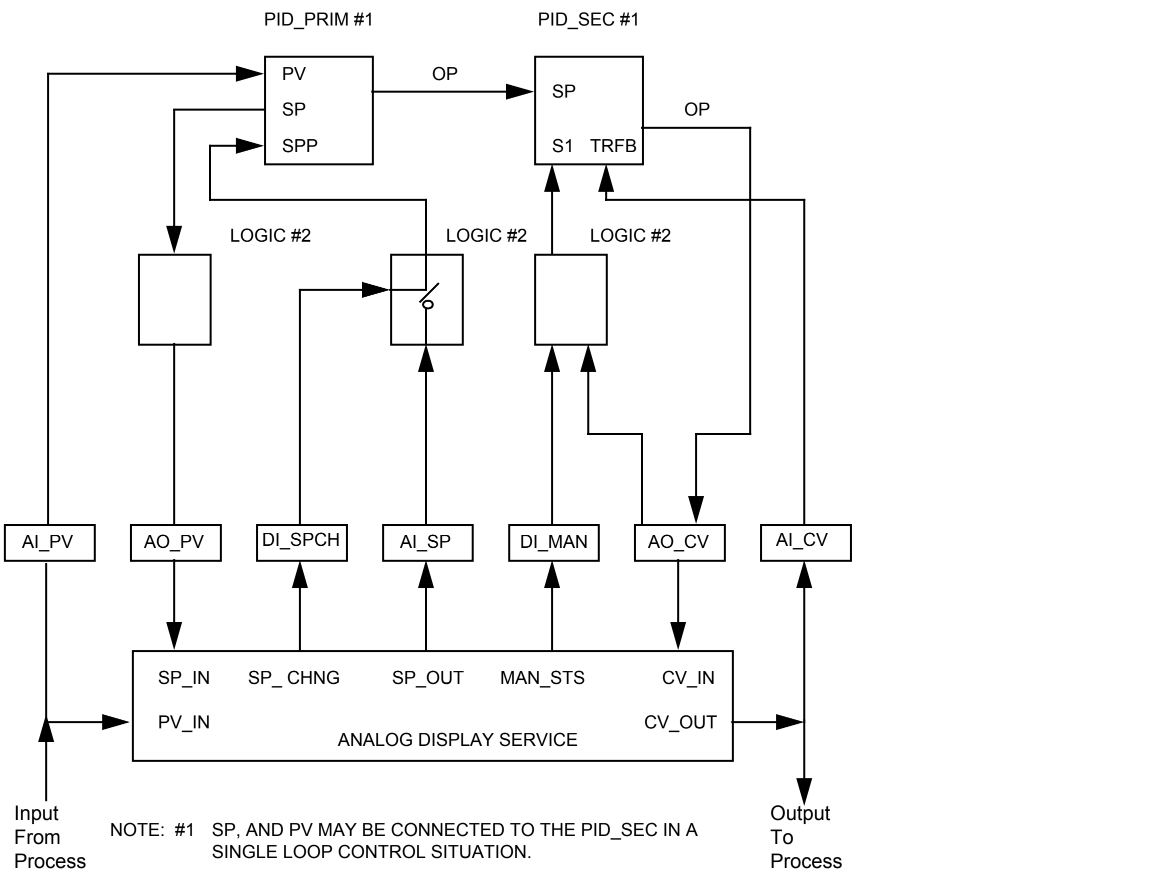
All of the following PID Control-algorithm options and special features apply to the PID with External Reset Feedback algorithm:

• 8.14.1.3.1	Interactive and Noninteractive PID Forms
• 8.14.1.3.3	Four Combinations of Control Terms
• 8.14.1.3.4	Control By a Single Term
• 8.14.1.3.5	Direct and Reverse Control Action
• 8.14.1.3.6	PV Tracking
• 8.14.1.3.7	Gain Options
• 8.14.1.3.9	Suppression of Output "Kicks" when Switching to Cas Mode
• 8.14.1.3.11	Restrictions on Some Values
• 8.14.1.3.12	Ratio Control
• 8.14.1.3.13	Operating Modes; Man, Auto, Cas and Bcas
• 8.14.1.3.14	Remote Cascade Options
• 8.14.1.3.15	Restart or Point Activation
• 8.14.1.3.16	Error Handling

8.14.3.3.1 Error Handling, RFB and TRFB Inputs

If S1 is Off, and the reset-feedback input has a bad value, the data-point mode doesn't change and the CV value goes bad (NaN). When the RFB input is again good, the CV value is initialized (see 8.14.1.5) and the dynamic terms are returned to a steady state. If configured for external initialization, an initialization request is sent to the primary data point.

If S1 is On, and the tracking-value input has a bad value, the data-point mode doesn't change and the CV value goes bad (NaN). When the TRFB input is again good, the CV value is initialized (see 8.14.1.5) and the dynamic terms are returned to a steady state. If so configured, an initialization request is sent to the primary data point.



NOTE: #1 SP, AND PV MAY BE CONNECTED TO THE PID_SEC IN A SINGLE LOOP CONTROL SITUATION.

#2 SEE THE NEXT PAGE FOR DETAILS OF THE LOGIC SLOT CONFIGURATION.

2083

Figure 8-10 — Configuration and Wiring Diagram Hand/Auto Station

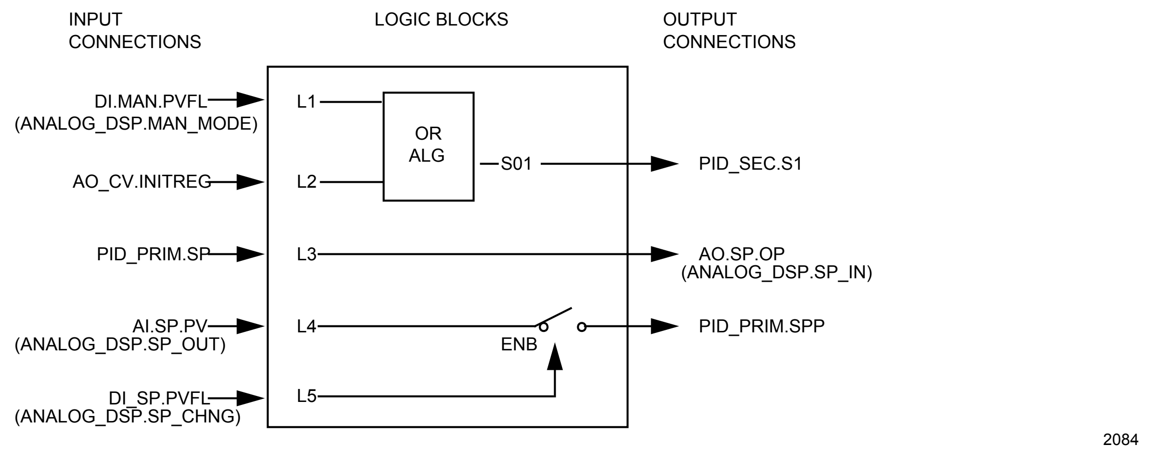


Figure 8-11 — Logic Point Configuration Hand/Auto Station

8.14.3.3.2 Output Connections

The output of this algorithm can be to any desired destination using control output connections.

NOTE

Initialization and windup protection normally associated with control output connections are not performed with this algorithm. That is, initialization occurs only when S1, the tracking switch-control signal, is true.

Output connections store the value of the OP after converting to engineering units.

See 8.10.6, Control Output Connections, for restrictions on the number of output connections and destinations for this algorithm.

8.14.3.3.3 Engineering Unit Ranges

PV Engineering unit range (PVEULO and PVEUHI) and the output engineering unit range (CVEULO and CVEUHI) must be specified. The SP engineering unit range (SPEULO and SPEUHI) always follows the PV range.

8.14.3.3.4 Operating Modes

The PIDERFB algorithm operates in the following modes:

- Man
- Auto
- Cas
- Bcas — If RCASOPT = Spc or Ddc or DdcRsp

8.14.3.3.5 Remote Cascade Options

All Remote Cascade Options are supported: Spc, Ddc, DdcRsp and Rsp.

8.14.3.4 Equations

If the value in the S1 parameter is On,

$$\mathrm{CV} = \frac{\mathrm{TRFB - CVEULO}}{\mathrm{CVEUHI - CVEULO}} * 100$$

For equations A, B, and C, if the S1 value is Off,

$$\mathrm{CVRFB}(s) = \mathrm{K} * \frac{\mathrm{K1}}{\mathrm{T1} * s} * (\mathrm{rfb}(s) - \mathrm{CV}(s))$$

For equation D, if the S1 value is Off,

$$\mathrm{CVRFB}(s) = \mathrm{K} * \frac{1}{\mathrm{T1} * s} * (\mathrm{rfb}(s) - \mathrm{CV}(s))$$

$$\mathrm{CV = CVPID + CVRFB}$$

Where:

CV = Full-value output in percent, PID combined with CVRFB.

CVPID = The incremental output of the PID computation. This is an internal parameter and is not available to displays nor to user-written programs.

CVRFB = The scaled, integrated deviation of RFB from CV. This is an internal parameter and is not available to displays nor to user-written programs.

K = Gain

K1 = External, reset-feedback gain

RFB = The external, reset-feedback signal in engineering units. Default = NaN.

rfb = $\frac{\mathrm{RFB - CVEULO}}{\mathrm{CVEUHI - CVEULO}} * 100$

s = The Laplace operator.

S1 = The switch-control flag. Default = Off.

TRFB = The tracking value in engineering units. Default = NaN.

T1 = The integral term in minutes per repeat.

8.14.3.5 Initialization

Initialization normally associated with output connections is not performed with this algorithm. Output connections store the value of the OP (after converting to engineering units based on CVEULO and CVEUHI) to the secondary.

8.14.3.6 Override Feedback Processing

Override-feedback processing is as described under 8.14.1.6; however, use of PIDERFB in override strategies is not recommended.

8.14.4 Position Proportional Controller (POSPROP)

8.14.4.1 Function

This algorithm manipulates two digital outputs, raise and lower, to drive the PV toward the SP. The setpoint is typically the desired position of a valve and the PV is the actual feedback from the process.

Digital outputs are pulsed at a time interval specified by the CYCLETIME parameter and the pulse width is proportional to the error signal.

This algorithm requires only one input. The number of inputs is defaulted to one; however, the SP can be fetched with an input connection. PV Source selection is supported if the point is configured as a full point. The Advisory (ASP) and Target Value (TV) options of SPOPT (Setpoint Option) are supported.

The Raise/Lower digital outputs are stored to the destinations specified by the parameters RAISDSTN and LOWRDSTN, respectively. The only valid output destinations are to pulsed (but not PWM type) digital outputs. The pulse can be ONPULSE or OFFPULSE.

See Figure 8-12.

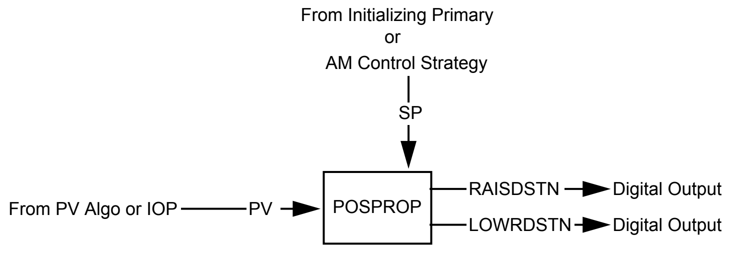


Figure 8-12 — Position Proportional Algorithm Functional Diagram

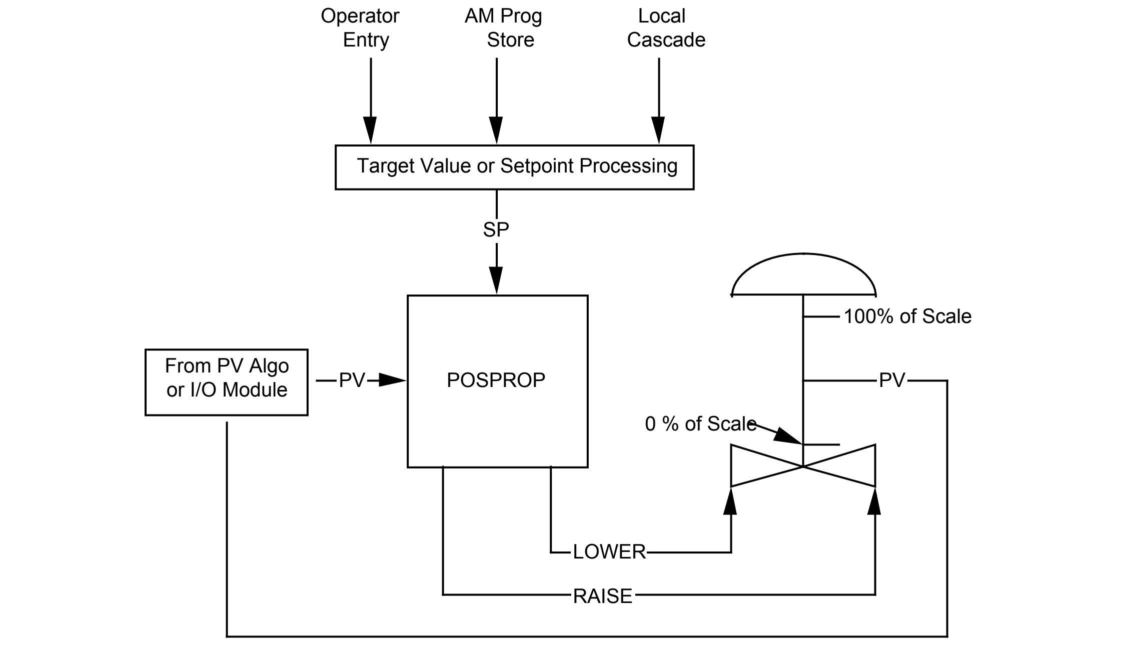


Figure 8-13 — Example of Position Proportion in Step Valve Control Strategy 2086

8.14.4.2 Use

This algorithm is used to pulse two digital outputs to drive the process variable toward the setpoint. This algorithm would typically be used to step a valve open or closed, to raise or lower a rotary device, or to move plates of a pulp mill refiner together or apart.

8.14.4.3 Options and Special Features

8.14.4.3.1 Modes

All four modes (Man, Auto, Cas, and Bcas) apply. When in Auto, Cas or Bcas modes, the normal computation is performed. When in Man mode, the output pulses are issued on operator demand.

8.14.4.3.2 Remote Cascade Options

Only the SPC option is supported.

8.14.4.3.3 Output Manipulation in Manual Mode

In Manual mode, the output can be manipulated from the group and detail displays using the Raise/Lower keys. At the time of configuration, the engineer must define "manual output pulse time (MANOPTIM)." When the operator presses the Raise key (single up-arrow), raise output pulse of width equal to MANOPTIM is issued. When the operator presses the fast raise key (double up-arrow), raise output pulse to width equal to 10 times MANOPTIM is issued. Lower output pulses can be similarly generated.

8.14.4.3.4 Windup Feedback Limit Switches

Output high and low flags can be set to indicate the status of the limit switches representing the valve position. When OPHIFL is set, the Raise output pulses are not generated. When OPLOFL is set, Lower output pulses are inhibited.

OPHIFL and OPLOFL are usually set by output connections on Logic slot based on limit switch feedback (brought into the system through digital inputs) from the process.

NOTE

In manual mode, the operator can manipulate the output Raise/Lower pulses regardless of the status of the OPHIFL and the OPLOFL.

8.14.4.3.5 Raise/Lower Output Destinations

The Raise/Lower digital outputs are stored to the destinations specified by RAISDSTN and LOWRDSTN, respectively. The only valid outputs are to ONPULSE or OFFPULSE of digital outputs configured as Status type.

8.14.4.3.6 Setpoint Options

Both options, advisory setpoint (ASP) and target value (TV), are supported.

8.14.4.3.7 PV Source Selection

PV source selection and PV alarm reporting are supported.

8.14.4.3.8 Engineering Unit Ranges

The PV Engineering unit ranges (PVEULO and PVEUHI) must be specified. The setpoint engineering unit range (SPEULO and SPEUHI) always follows the PV range. The output engineering unit range (CVEULO and CVEUHI) does not apply.

8.14.4.3.9 Analog Output Related Parameters

The standard control slot parameters that are normally related to the analog output value do not apply because analog outputs are not valid destinations for outputs from this algorithm. The OPHIFL and OPLOFL are explained under 8.14.4.3.4. Other parameters associated with analog outputs cannot be fetched or stored by any system function. Internally, they are defaulted such that they do not have any effect on the system functions.

PARAMETER NAME	**INTERNALLY DEFAULTED TO**
CV	NaN
CVEUHI	100.0
CVEULO	0.0
OP	-6.9% of Full Scale
OPEU	N/A
OPHILM	105.0%
OPLOLM	-5.0%
OPROCLM	NaN
OPMCHLM	0.0
SAFEOP	N/A
NOCOPTS	1
INITMAN	Off

8.14.4.4 Equations

Only one equation is available. The equation is calculated as follows:

The raise and lower pulses are generated at a rate specified by the cycle time. At the beginning of each cycle time, the pulse width is calculated and the digital outputs are pulsed as follows:

```
If CYCLE_TIMER expired:
  IF (PVP < (SPP - DEADBAND)) AND (OPHIFL = Off) THEN
    Issue a raise pulse of width equal to:
      $RAISETIM = K * \frac{(SPP - PVP)}{RAISRATE} + DEADTIME$
  ELSE IF (PVP > (SPP + DEADBAND)) AND (OPLOFL = Off) THEN
    Issue a lower pulse of width equal to:
      $LOWERTIM = K * \frac{(PVP - SPP)}{LOWRRATE} + (RT * DEADTIME)$
  ELSE Reset the cycle timer to the beginning.
```

Where: PVP = PV in percent

SPP = SP in percent

DEADBAND = the error deadband in percent (Range = 0.0 to 100.0; Default = 5.0)

OPHIFL = High output windup flag

OPLOFL = Low output windup flag

CYCLETIM = the cycle time in seconds. Range: 0.25 to 1000.0; Default = 10.0

RAISRATE = the raise stroke rate in percent per second. Default = 100.0% per sec
Range: greater than 0.0

LOWRRATE = the lower stroke rate in percent per second.
Range: greater than 0.0. Default = 100.0% per second

RAISETIM = the raise pulse time in seconds. Default = N/A.
It is clamped to the lower of MAXPULSE or CYCLETIM.
If smaller than MINPULSE, no pulse is issued.

LOWERTIM = the lower pulse time in seconds. Default = N/A
It is clamped to the lower of MAXPULSE or CYCLETIM.
If smaller than RP*MINPULSE, no pulse is issued.

K = Gain constant (Range = 0.0 to 10.0; Default = 1.0)

RP = Pulse ratio (Default = 1)

RT = Deadtime ratio (Default = 1)

DEADTIME = Additional pulse time required to overcome friction in the motor when it begins to move or change direction. It is added to the computed pulse time except when the pulse issued last cycle time was in the same direction (as the pulse this time) and the pulse width was equal to CYCLETIM. Default = 0.0

MAXPULSE = Maximum pulse time limit. If computed pulse time is greater than this, it is clamped to the lower of MAXPULSE or CYCLETIM.
Default = 60 seconds

MINPULSE = Minimum pulse time limit for raise pulse. If computed pulse is smaller than this, no pulse is issued. Default = 0.0 seconds.

8.14.4.5 Initialization

The Raise and Lower outputs are both set to OFF (or their normal states) and the cycle is restarted when forward calculation is resumed after initialization.

The SP is set equal to the PV (subject to the setpoint limits) when any of the following conditions exist:

- The mode is Man.
- The slot is being processed for the first time after becoming active.
- This slot is a secondary within a local (inside the same APM box) cascade control strategy, and it is going through one-shot control initialization.

NOTE

The one-shot control initialization occurs:

- when the control initialization request flag (CTRLINIT) is On,
- when this slot is being processed for the first time after the APM state has been changed to RUN (Ok),
- when just recovering from a bad PV.

The position proportional control algorithm is forced to initialize when outputting to a digital output point that has its INITREQ parameter = ON.

8.14.5 PID Position Proportional Controller (PIDPOSPR)

8.14.5.1 Function

This algorithm can be viewed as a normal PID algorithm joined in cascade with a PosProp algorithm such that the PosProp part uses the del_cv of the normal PID as its PV to generate raise and lower pulses. The function of the PID part of the algorithm is the same as that of the normal PID algorithm except that PidPosPr does not support OP (and OP related parameters OPEU, OPHILM, etc.) and CV (and CVEUHI, CVEULO). The end part of the algorithm behaves exactly as a POSPROP algorithm. Figure 8-14 demonstrates the concept.

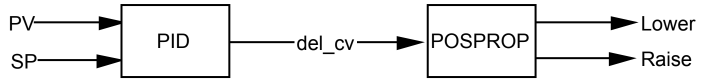


Figure 8-14 — PID Position Proportional Algorithm Functional Diagram 11016

8.14.5.2 Use

This algorithm is typically used in place of the POSPROP algorithm to operate a motor driven valve without position feedback.

8.14.5.3 Options and Special Features

8.14.5.3.1 Modes

All four modes (Man, Auto, Cas, and Bcas) apply. When in Auto, Cas, or Bcas modes, the normal computation is performed. When in Man mode, the output pulses are issued on operator demand.

8.14.5.3.2 Remote Cascade Options

Only the SPC option is supported.

8.14.5.3.3 Output Manipulation in Manual Mode

In Manual mode, the output can be manipulated from the group and detail displays using the Raise/Lower keys. At the time of configuration, the engineer must define "manual output pulse time (MANOPTIM)." When the operator presses the Raise key (single up-arrow), raise output pulse of width equal to MANOPTIM is issued. When the operator presses the fast raise key (double up-arrow), raise output pulse to width equal to 10 times MANOPTIM is issued. Lower output pulses can be similarly generated.

8.14.5.3.4 Windup Feedback Limit Switches

Output high and low flags can be fetched using the input connection parameters Output High and low Flag Input Source (OPHISRC) and (OPLOSRC), respectively. These flags serve the same function as the similar flags for POSPROP.

8.14.5.3.5 Local Manual Support

The Local Manual state is supported and it can be fetched with an input connection using the input parameter Local Manual Source (LMSRC).

8.14.5.4 Equations

Only one equation is available. The equation is calculated as follows:

```
If CYCLE_TIMER expired:
  IF ((ACCUMULATED_DEL_CV > DEADBND) AND (OPHIFL = OFF))
    Issue a raise pulse of width equal to:
      $\text{RAISETIM} = \text{K1} * \frac{\text{ACCUMULATED—DEL—CV}}{\text{RAISRATE}} + \text{DEADTIME}$
  ELSE IF (ACCUMULATED_DEL_CV < ( - DEADBAND)) AND (OPLOFL = OFF)
   THEN
    Issue a lower pulse of width equal to:
      $\text{LOWERTIM} = \text{K1} * \frac{(\text{-ACCUMULATED—DEL—CV})}{\text{LOWRRATE}} + (\text{RT} * \text{DEADTIME})$
  ELSE Reset the cycle timer to the beginning.
```

Where: ACCUMULATED_DEL_CY = the accumulations of del_cv (of the PID part) during this cycle.

K1 = the Gain constant for the POSPROP part of the algorithm. It is renamed K1 (was K for the POSPROP algorithm) to avoid confusing it with the PID gain constant K.

DEADBAND = the error deadband in percent (range = 0.0 to 100.0; default = 5.0)

OPHIFL = High output windup flag

OPLOFL = Low output windup flag

RAISRATE = the raise stroke rate in percent per second. Default = 100.0% per sec
Range: greater than 0.0

LOWRRATE = the lower stroke rate in percent per second.
Range: greater than 0.0. Default = 100.0% per second

RAISETIM = the raise pulse time in seconds. Default = N/A.
It is clamped to the lower of MAXPULSE or CYCLETIM.
If smaller than MINPULSE, no pulse is issued.

LOWERTIM = the lower pulse time in seconds. Default = N/A
It is clamped to the lower of MAXPULSE or CYCLETIM.
If smaller than RP*MINPULSE, no pulse is issued.

RT = Deadtime ratio (Default = 1)

DEADTIME = Additional pulse time required to overcome friction in the motor when it begins to move or change direction. It is added to the computed pulse time except when the pulse issued last cycle time was in the same direction (as the pulse this time) and the pulse width was equal to CYCLETIM. Default = 0.0

8.14.5.5 Analog Output Related Parameters

Parameters related to the analog output value do not apply and are defaulted to the same values as those similar parameters in the PosProp algorithm.

8.14.5.6 Safety Shutdown

The Shutdown flag, when turned on, causes the mode to go to MAN with OPR attribute and the output is derived based on the value of a new parameter SAFOPCMD as follows:

When:

SAFOPCMD = Idle No more output pulses are issued.

SAFOPCMD = Raise Raise pulses are issued until the PV is ≥ EUHI, or OPHIFL is ON. If the PV is bad, the test for PV ≥ EUHI is ignored.

SAFOPCMD = Lower Lower pulses are issued until the PV is ≤ EUHI, or OPLOFL is ON. If the PV is bad, the test for PV ≤ EUHI is ignored.

8.14.5.7 Bad PV/Mode Shed

For PidPosPr (and PosProp) algorithms, Shed Hold, Shed Low, High, and Shed Safe are interpreted as follows:

BADCTLOP	Action Taken for PIDPosPr (and PosProp)
No Shed	IDLE, No more output pulses are issued.
Shed Hold	IDLE
Shed High	Raise, Raise pulses are issued until PV ≥ EUHI or OPHIFL is ON. If the PV is bad, the test for PV ≥ EUHI is ignored.
Shed Low	Lower, Lower pulses are issued until the PV ≤ EUHI or OPLOFL is ON. If the PV is bad, the test for PV ≤ EUHI is ignored.
Shed Safe	Handled like Shutdown based on the value of SAFOPCMD.

8.14.6 Ratio Control (RATIOCTL)

8.14.6.1 Function

This algorithm calculates a setpoint for a PID algorithm that is the desired ratio of a controlled variable to an uncontrolled variable. The value of the controlled variable is maintained at a specified ratio of the value of the uncontrolled variable. The data point that uses this algorithm usually uses Calculator PV algorithm to calculate the measured value of the ratio for displays and reports.

Ratio control can also be accomplished with the ratio-bias options of the PID or PID Feedforward control algorithms (see 8.14.1, 8.14.2). This Ratio-control algorithm has several advantages, including the display of the actual ratio attained as calculated by the Calculator PV algorithm and direct control of the ratio through the SP of the Ratio algorithm.

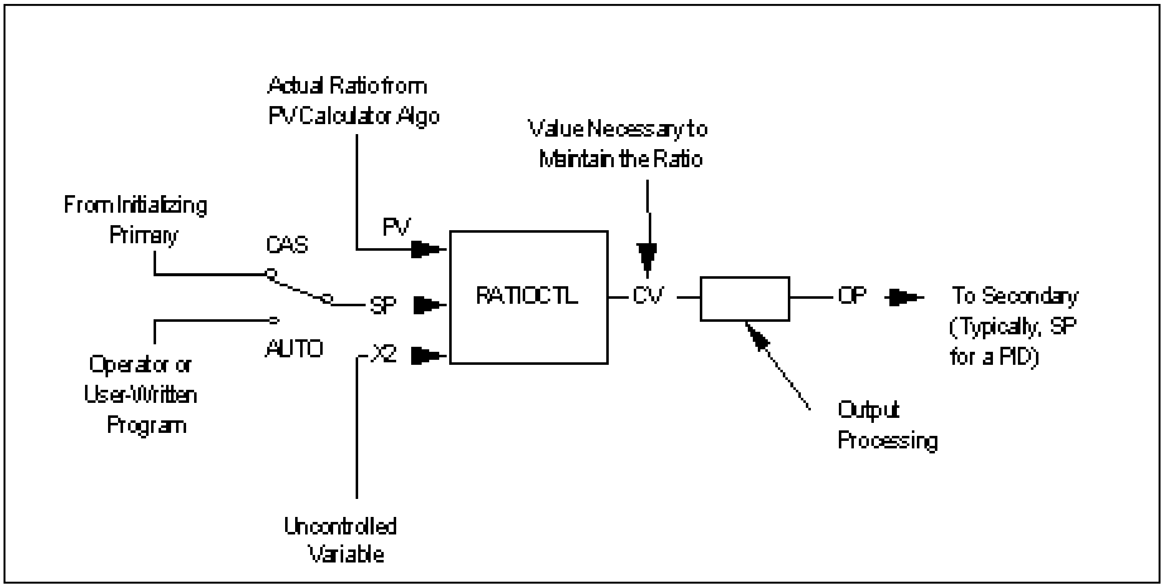


Figure 8-15 — Functional Diagram, Ratio Control Algorithm 1334

8.14.6.2 Use

This algorithm is typically used in the control of the flow of a gas or fluid, as a ratio of an another flow. For example, in a furnace, the air supply might be controlled as a ratio of the fuel supply. If more heat is required to maintain combustion efficiency, the fuel flow is increased and the air flow can be increased as a ratio of the fuel-flow increase.

Figure 8-16 shows an example of such an application. In this example, the data point that uses the Ratio-control algorithm also uses the Calculator PV algorithm to calculate the actual ratio achieved, for display or printing. The Calculator PV should use the filtered value (X2FILT) of the uncontrolled variable.

To evaluate this example, see the equations under 8.14.6.4 and you will note that the same scale factor, 0.7, is used for P1 in the PV algorithm and for K1 in the Ratio-control algorithm. The resulting scaled ratio between the "other" flow and the controlled flow is 2.00/0.7 = 2.857, so if the "other" flow is 6.00 gallons per minute, the controlled flow must be 6.00*2.857 = 17.143 gallons per minute.

The 0.7 scale factor is used for C1 and K1 in the example to illustrate that the same scale factors and bias values must be used with the PV algorithm and the Ratio-control algorithm (K1 = C1, K2 = C2, B1 = C3, and B2 = C4), so that the actual ratio calculated by the PV algorithm will be the same as the desired ratio (2.00) when the loop is stable. If the scale factor in C1 and K1 were 1.0, the controlled flow would stabilize at the "other" flow, multiplied by the ratio. In the example of Figure 8.15, the controlled flow would be 6.00*2.00 = 12.00 gallons each minute. In any case, the controlled flow stabilizes at a value equal to the "other" flow, multiplied by the desired ratio, as modified by any scale factors other than 1.0 or any bias values other than 0.

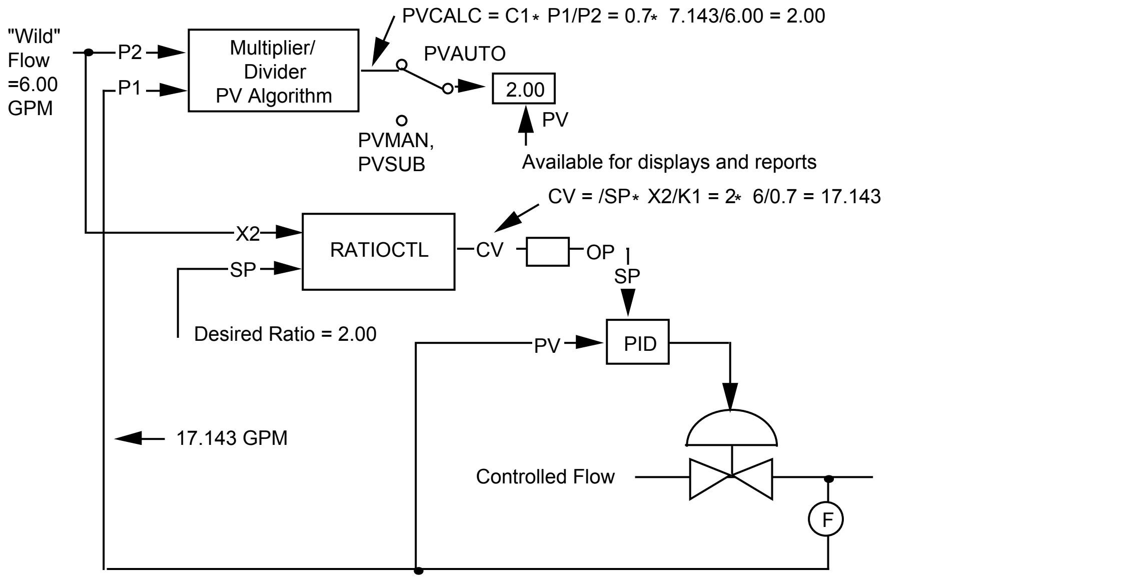

PVCALC = C1* P1/P2 = 0.7* 7.143/6.00 = 2.00
"Wild" Flow =6.00 GPM
P2
P1
Multiplier/ Divider PV Algorithm
PVAUTO
2.00
PV
PVMAN, PVSUB
Available for displays and reports
CV = /SP* X2/K1 = 2* 6/0.7 = 17.143
X2
SP
RATIOCTL
CV
OP
SP
Desired Ratio = 2.00
PV
PID
17.143 GPM
Controlled Flow
F

Figure 8-16 — Ratio Control Example

8.14.6.3 Options and Special Features

8.14.6.3.1 Control Input Connections

This algorithm requires two input connections, for PV processing and X2. The number of input connections is defaulted to 2; however, it can be increased to 3, allowing the SP to be fetched with an input connection.

8.14.6.3.2 Engineering Unit Ranges

The PV engineering unit range (PVEULO and PVEUHI) must be specified. The setpoint engineering unit range (SPEULO and SPEUHIU) always follows the PV range. The output engineering unit range (CVEULO and CVEUHI) is derived from the secondary if the number of output connections is greater than 0; otherwise, it must be specified.

8.14.6.3.3 Role of the Calculator PV Algorithm

Any data point that uses RATIOCTL should use the Calculator PV algorithm (7.7.9). X2FILT, the filtered value of the uncontrolled variable X2 is connected to P2 and the variable controlled by the PID algorithm (see Figure 8-16) is connected to P1. The scale factors and bias values in the Calculator PV algorithm must have the same values as their counterparts in the Ratio Control algorithm:

RATIOCTL		CALCULATOR
K1	=	C1
K2	=	C2
B1	=	C3
B2	=	C4

Thus, Calculator can calculate the actual (measured) ratio attained, and when the PVSOURCE is Auto, that value is available in the PV parameter of the data point for use on displays and reports.

8.14.6.3.4 Operating Modes

The RATIOCTL algorithm operates in the following modes:

- Man
- Auto
- Cas
- Bcas—if RCASOPT = Spc

8.14.6.3.5 Remote Cascade Options

Only the Spc option is supported.

8.14.6.3.6 Restart or Point Activation

On a cold or warm restart or when the RATIOCTL data point is activated, initialization takes place as described under 8.14.6.5.

8.14.6.3.7 Error Handling

If the value status of the X2 input is bad, the CV value is changed to bad (NaN). The data point remains in the same mode. When the X2 input again has normal status, initialization takes place as described under 8.14.6.5.

8.14.6.3.8 Operator Entered Bias (BO)

In R600 or later, the Algorithm includes an operator-entered bias. This Bias parameter value is added after all other calculations to obtain the CV for print. For equivalent operation when migrating from PRE R600 Releases set BO = 0.0.

8.14.6.3.9 Initialization Ramping Bias

The bias B is made up of two components, BO and BI, where BO is the operator entered bias and BI is the internal bias. An optional initialization method using an internal ramping bias is available. Parameter RATE1 specifies the decay rate for the Internal Bias, BI. The value of BI is set during the algorithms initialization and decays to zero at a rate defined by RATE1.

Compatibility—If RATE1 is set to NaN, the initialization ramping value is set to 0.0 and the initialization value for the primary is determined by back calculation compatible with PRE R500 functions. When migrating from R4xx or R5xx to R600, the fixed bias term BO should be defaulted to 0.0.

When RATE1 = NaN

$$\texttt{INITVAL} = \frac{K1*(CV - BO) + BI}{K2*X2FILT + B2}$$

If RATE1 is set to a non-zero value, and the Ratio Control block is in Cascade mode, the SP value is back calculated. During initialization, an internal bias that is equal to the difference between the old CV value and the new CV value is added to the output calculation to produce a bumpless output Internal Bias (BI) is calculated as follows:

$$\texttt{BI} = \mathrm{CV}_{\mathrm{old}} - \frac{SP*(K2*X2FILT + X2) - BI}{K1} \ \texttt{-BO}$$

where $\mathrm{CV}_{\mathrm{old}}$ is the last calculated CV value before initialization.

If the primary accepts the initialization value (INITVAL), then BI turns out to be zero. If the primary does not accept the initialization value, then BI turns out to be non-zero. If not zero, the internal bias (BI) decays at the rate specified by RATE1. RATE1 is specified in engineering units per minute. If RATE1 = 0, then BI does not decay but remains fixed.

If the Ratio Control block operates in Auto mode during initialization when the mode changes from manual to auto, the internal bias (BI) is added to the output. BI decays at the rate specified by RATE1 as described for Cascade mode. The default value of RATE1 = NaN.

8.14.6.3.10 X2 Filter

At the X2 input to the Ratio Control block, the value of the uncontrolled variable is filtered by a single lag filter. Parameter X2TF determines The X2 filter time (0 - 60 minutes) and X2FILT is the filtered value of X2. The X2FILT value appears on the point detail display. This filter is active only if the point is in the AUTO or CASC mode.

Compatibility—When X2TF is set for 0.0 minutes lag time, X2 is not filtered as was the case for the algorithm prior to Release 500.

The default value of X2TF = 0.0.

8.14.6.4 Equations

The equations are as follows:

- Calculator PV Algorithm

$$\text{PVCALC} = \frac{(C1 * P1 + C3)}{(C2 * P2 + C4)}$$

Where (see Figure 8.15)

PVCALC = The calculated, actual ratio achieved.
C1 = P1 scale factor. Must equal K1 of the RATIOCTL algorithm Default value = 1.0.
C2 = P2 scale factor. Must equal K2 of the RATIOCTL algorithm Default value = 1.0.
C3 = Bias constant P1 input. Should be the same value as B1in the RATIOCTL algorithm. Default value = 0.
C4 = Bias constant P2 input. Should be the same value as B2 in the RATIOCTL algorithm. Default value = 0.
P1 = The controlled process variable. Source should be the same as the PV of the PID controller that is RATIOCTL's secondary.
P2 = The filtered value (X2FILT) of the uncontrolled process variable. If there is a PID-controller controlling this other flow, the PV of that PID could be the source for P2 and X2.

- Ratio Control Algorithm

$$\text{CV} = \frac{SP * (K2 * X2FILT + B2) - BI)}{K1} + \text{B}$$

Where (see Figure 8-16)

B = The overall bias value (= BO + BI)
BI = Internal bias
BO = Fixed output bias
CV = The calculated output in engineering units
SP = The desired ratio input
X2FILT = The filtered value of the uncontrolled process variable X2.

BI = Internal Bias.

B1 = Bias constant. Should be the same value as C3 in the CALCULTR PV algorithm. Default = 0.

B2 = Bias constant for the X2 input. Should be the same value as C4 in the CALCULTR PV algorithm. Default = 0.

K1 = The ratio scale factor. Must equal C1 of the CALCULTR PV algorithm.

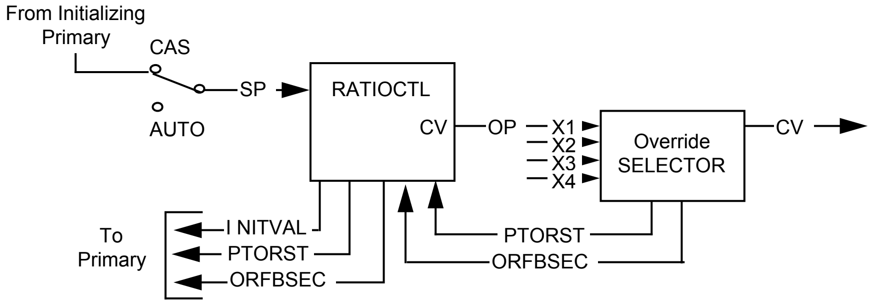
K2 = The scale factor for X2. Must equal C2 of the CALCULTR algorithm.

8.14.6.5 Initialization

When the data point is initialized, an initialization request is sent to the primary, and the initialization value to be applied by the primary to the SP input is calculated and sent to the primary as follows:

$$\texttt{INITVAL} = \frac{K1*(CV - BO) + BI}{K2*X2FILT + B2}$$

Refer to 8.14.6.3.8 Ramping Bias for a more complete discussion.



ORFBSEC is not an external parameter.

Figure 8-17 — Override Feedback Processing

1336

8.14.6.6 Override Feedback Processing

When the data point's secondary uses an Override Selector algorithm (see Figure 8-17), the following functions take place if the RATIOCTL algorithm if it is in Cas mode:

- The override status is sent to the primary data point in PTORST (Point Override Status).
- If the status in PTORST is not selected, a feedback value, calculated as follows, is sent to RATIOCTL's primary data point.

$$\text{ORFB} = \frac{K1*(ORFBSEC - B) + BI}{K2*X2FILT + B2}$$

Where ORFBSEC is the override-feedback value received from the secondary data point.

For more detail on override control, see 8.13 of this publication.

8.14.7 Ramp and Soak (RAMPSOAK)

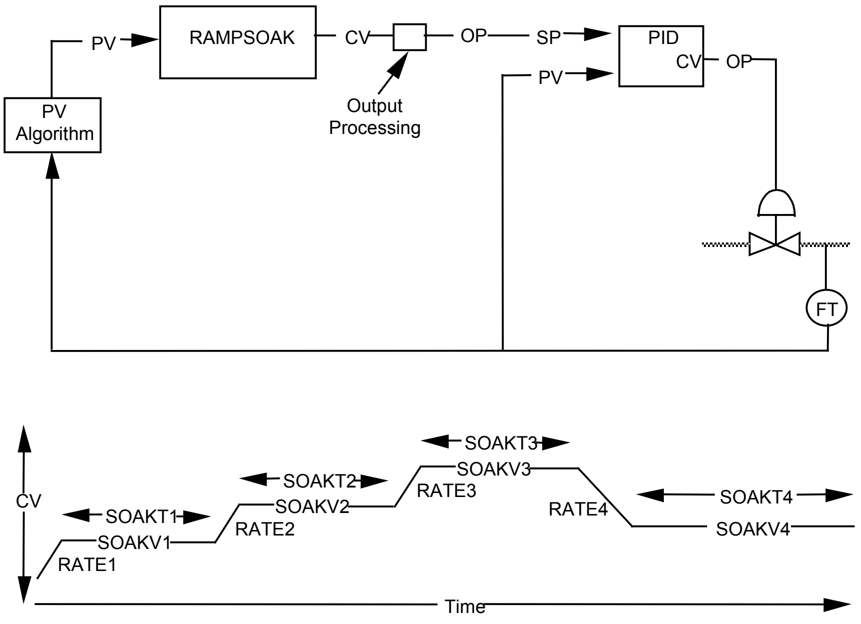


Figure 8-18 — Functional Diagram of Ramp and Soak Control Algorithm

1337

8.14.7.1 Function

This algorithm is typically used as a setpoint programmer. It produces an output that consists of up to 12 alternate ramps and soak periods—a total of 24 segments. Twelve pairs should be enough to cover most applications commonly encountered; however, if longer ramp and soak sequences are required, four such algorithms can be configured to push to a control switch algorithm. Then, using logic, the switch could select one of the ramp and soak algorithms. See 8.14.7.3.8.

When not used in connection with the switch algorithm, output from this algorithm is usually used as the setpoint for a secondary data point that uses a PID algorithm to control a process variable, according to the ramps and soak periods. The PV of a data point that uses RAMPSOAK is normally the PV of the PID point.

8.14.7.2 Use

RAMPSOAK is principally used for automatic temperature cycling in furnaces and ovens. It can also be used for automatic startup of units, and for simple batch-sequence control where the batch sequence is part of a process that is otherwise a continuous process.

8.14.7.3 Options and Special Features

8.14.7.3.1 Single or Cyclic Sequencing

Once started, the configured sequence of ramps and soak periods repeats itself, if it is not stopped by an operator or by a user-written program. A Universal Station operator can put the point in Man mode to freeze the sequence, and then return it to Auto to continue the sequence.

A configuration parameter called cycle option (CYCLEOPT = Single or Cyclic) is provided that optionally permits the cycle to stop after completing a single cycle. This works as follows:

- CYCLEOPT = Single

  When the mode is changed from Man to Auto, the algorithm cycles through the configured rampsoak sequence until the last soak segment is completed. At this point, the mode is switched back to Man and the current segment ID is set to RAMP1 (the first segment). Another sequence can now be executed by simply switching the mode back to Auto.

- CYCLEOPT = Cyclic

  When the mode is changed from Man to Auto, the algorithm cycles through the configured rampsoak sequence until the last soak segment is completed. At this point, the current segment ID is set to RAMP1 (the first segment) and the whole cycle is repeated.

8.14.7.3.2 Operational Modes

The operating modes establish the operating state of the RAMPSOAK algorithm as follows:

- Man—The sequence is stopped and the ramp/soak timers are not running.
- Auto—The sequence is running.

Further functions in each mode are the following:

- Man mode
  - The Ramp and Soak actions are suspended.

    This allows the user to enter starting output value for the Ramp/Soak profile and to change the value during profile.
  - The timers are stopped and hold the last value.
  - The value in CV is replaced by the OP value (after converting to EUs).
  - SP = CV (SP doesn't affect the output but can be seen at Universal Stations and user-written programs).
- Auto Mode
  - If the current segment is a ramp, and if the guaranteed ramp conditions are OK (see 8.14.7.3.6),

    CV changes at the ramp rate.

    If CV should overshoot the next soak value it is clamped at that value, the remaining soak-time (REMSOAKT), the current-segment (CURSEGID), and the mark timers and flags (see 8.14.7.3.7) are updated.
  - If the guaranteed ramp conditions are not OK, the mark timers (see 8.14.7.3.7) are stopped.
  - If the current segment is a soak, and if the point just changed from Man to Auto, or just started the soak segment, and the guaranteed soak time conditions are not OK (see 8.14.7.3.5),

    The soak timer doesn't start.

    CV remains at its last value.

    The mark timers (see 8.14.7.3.7) are stopped.

- If the guaranteed soak-time conditions are OK,

  The soak timer begins to run or continues to run.

  CV holds at its last value and the mark timers and flags (see 8.14.7.3.7) are updated.

  The remaining soak time (REMSOAKT) is adjusted.

  If the soak timer times out, the current segment (CURSEGID) becomes the next ramp segment.

- In any case, in Auto mode, SP is equal to CV. SP doesn't affect the output but can be seen at Universal Stations and by user-written programs.

> **NOTE**
>
> If the polarity of the ramp rate is opposite to the soak target value, the output jumps to the soak target value. That is, the ramp segment for this ramp/soak cycle is omitted and the next segment is invoked.

8.14.7.3.3 Remote Cascade Options

Remote cascade options are not supported.

8.14.7.3.4 Changing Remaining Soak Time and Current Segment

When the RAMPSOAK point is in Man mode, an operator at a Universal Station can change the remaining soak time (REMSOAKT) if the current segment is a soak.

Also, when the point is in Man mode, an operator can change the current segment (CURSEGID).

When the mode is returned to Auto the sequence continues, as modified by these changes. If the segment was changed, the sequence resumes with the new segment, which can be a ramp or a soak.

Because changes to these parameters don't change the mark functions (see 8.14.7.3.7), except if CURSEGID is a lower segment than the mark segment (SnSEGID), operators should not be allowed to change REMSOAKT or CURSEGID when the mark functions are configured.

8.14.7.3.5 Guaranteed Soak-Time

This feature guarantees that the PV is at the proper soak value before the soak-time measurement begins.

If, when a soak segment begins or is resumed by switching from Man to Auto, the PV is not within a user-specified deviation (MXSOKDEV) from the SP value (SP always equals CV), the soak timer doesn't start. When the deviation is within the MXSOKDEV value, the timer is started and continues, even if the deviation again exceeds MXSOKDEV.

Because the PV could be above or below SP, it is the absolute value of the deviation that is checked against MXSOKDEV.

To bypass this check you can change MXSOKDEV to NaN.

The soak timer can also be kept from starting when HOLDCMD is On. This allows you to use a Logic Slot to set HOLDCMD to hold the soak timer until some other condition is met.

HOLDCMD also affects the guaranteed ramp function. See 8.14.7.3.6.

8.14.7.3.6 Guaranteed Ramp Rate

This feature guarantees that the PV keeps up with the desired value indicated by SP (SP always tracks CV).

You can specify a maximum ramp-deviation value in MXRMPDEV. There are two conditions that cause the ramp to stop to wait for the PV to catch up with SP. They are

- RATEn > 0

  and

  PV < (SP - MXRMPDEV)

- RATEn < 0

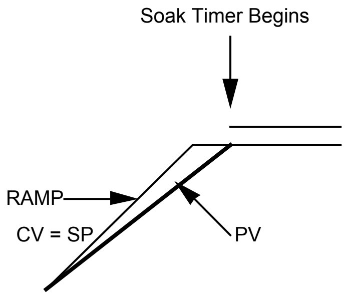
  and

  PV > (SP + MXRMPDEV)

These checks are bypassed if MXRMPDEV contains NaN.

Another condition that stops the ramp is HOLDCMD containing On. You can use a logic slot to set HOLDCMD to stop or hold the ramp until some condition that you specify is met.

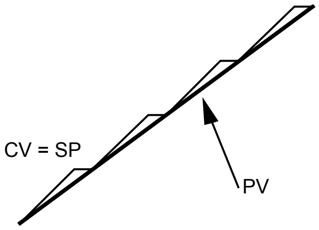
The content of HOLDCMD also affects the guaranteed soak-time feature. See 8.14.7.3.5.



By assuring that PV = proper soak value, the user is assured of, for example, the proper wait time between phases of a warm up or that temperature is held for the proper length of time during heat treating.

Figure 8-19 — Wait Time Between Phases (Ramp/Soak Algorithm)

11017



By ensuring that the ramping "waits" for the process to catch up, we never have a large error caused by setpoint change.

Figure 8-20 — Ramping Waits for Process

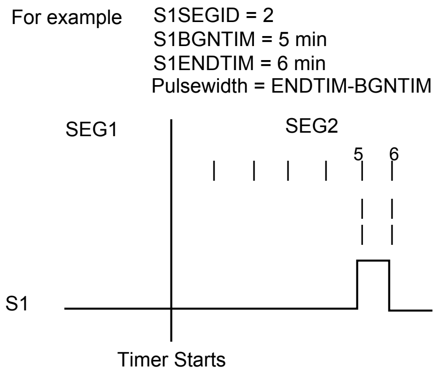
11018

8.14.7.3.7 Mark Timer Functions

Two flags are provided with the RAMPSOAK algorithm to indicate to other data points that a specified time has elapsed from the beginning of a specified ramp segment or soak segment. These mark-timer flags are S1 and S2.

Each of these flags has three associated parameters that specify the segment in which the flag is set On, the time counted from the beginning of the segment, and the time from the beginning of the segment until the end time (when the flag is set Off). These parameters are as follows:

Flag	Segment	Beginning Time	End Time
S1	S1SEGID	S1BGNTIM	S1ENDTIM
S2	S2SEGID	S2BGNTIM	S2ENDTIM



NOTE

- S1BGNTIM must be less than or equal to S1ENDTIM. (Same is true for S2 parameters.)
- S1ENDTIM is not required to terminate the marker within the segment specified by S1SEGID. For example, if Segment 2 is only 5.5 minutes long, then the marker will remain on until 6 minutes after the start of Segment 2 (which is 0.5 minutes into Segment 3).

Figure 8-21 — Ramp Soak Mark Timers

11019

The S1 or S2 flag is turned on at the number of minutes after the specified segment begins, as is specified in SnBGNTIM. The corresponding flag is turned Off at the number of minutes after the specified segment begins, as specified in SnENDTIM.

The following functions also take place:

- At the end of the last segment in the sequence, the S1 and S2 flags are turned Off and the timers are reset.
- When a ramp or a soak segment is held up by the guaranteed-ramp or the guaranteed-soak functions, the mark timers are suspended.
- The mark timers suspend when the data point is in Man mode and the S1 and S2 flags are unchanged.
- If the remaining soak time (REMSOAKT) is changed (in Man mode), the mark timers are not affected.
- If the current segment (CURSEGID) is changed (in Man mode) to a segment that is earlier than or equal to a segment specified by SnSEGID, the corresponding mark flag goes to Off and its timers are reset. If a later segment is specified in CURSEGID, the flags and timers are not affected.

8.14.7.3.8 Achieving Longer Sequences by Interconnecting RAMPSOAK Points

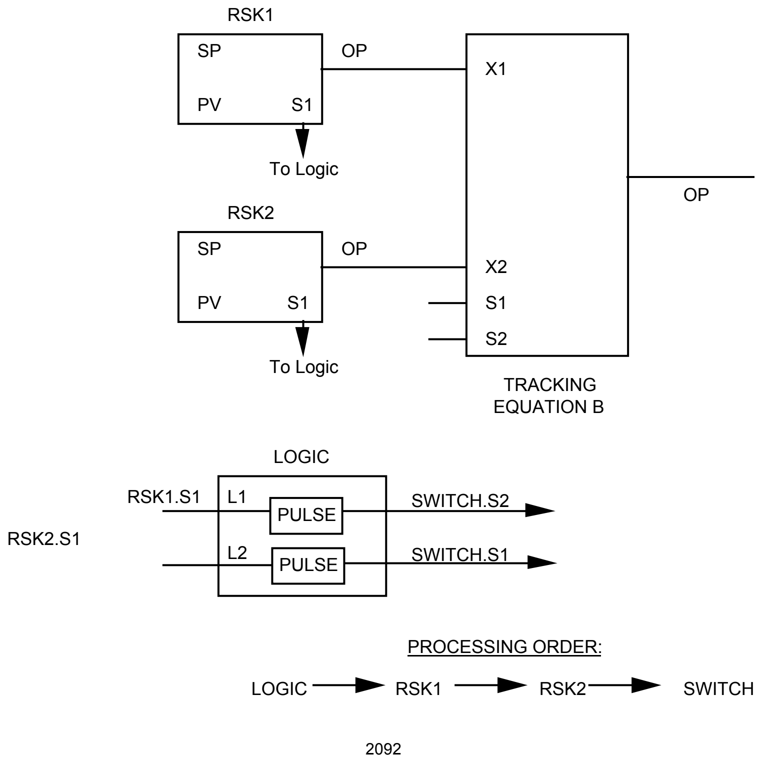


Figure 8-22 — Use of Logic Slot and Switch to Achieve Additional RampSoak Segments

A sequence of more than 12 ramp and soak segments can be attained by interconnecting RAMPSOAK points. A Logic Slot can be used with a Switch Algorithm (see 8.14.10) to select any of the four RampSoak Algorithm data points.

Longer sequences can be accomplished by setting the S1 Mark Flag at the end of the last soak segment (S1 must be on for at least one sample time).

A Logic point reads the S1 Mark Flag and, using a pulse Logic Block (Pulse width can be one sample time), sets the corresponding S input on the Switch Algorithm. This causes the next RampSoak to become selected and all nonselected RampSoaks are INIT. The Switch Algorithm should be configured for tracking and Equation B.

8.14.7.3.9 Engineering Unit Ranges; Setpoint , Output and Deviation Limits

The PV engineering unit range (PVEULO and PVEUHI) and the output engineering unit range (CVEULO and CVEUHI) must be specified. The setpoint engineering unit range (SPEULO and SPEUHI) always follows the PV range.

If the RAMPSOAK data point uses a PV algorithm, the setpoint EU range is the same as the PV EU range, and can't be differently configured. You should configure this point's PV EU range to be the same as the SP EU range of the secondary point that is receiving this point's output.

The setpoint and output limits do not apply. Setpoint limits are not available at all, and the output limits are defaulted to -6.9% to 106.9% and cannot be changed.

The deviation limits (MXRMPDEV and MXSOKDEV) apply to all segments in the sequence.

8.14.7.3.10 Restart or Point Activation

On a cold restart, a warm restart, or when the data point is activated, the mode goes to Man and the CV value is NaN. All timers are reset, and the current segment ID is made equal to the first ramp segment.

No special action occurs on a hot restart and the operation continues from where it was.

8.14.7.3.11 Control Input Connections

The ramp soak control algorithm requires only one control input connection (for PV). The number of control input connections is fixed at one.

In a typical application, the output of this algorithm provides the SP of a secondary controller (which is usually a PID control algorithm). The PV of the secondary controller is then fetched with a control input connection into the PV of the RAMPSOAK. If used in some different way such that the PV is not available from the secondary, the PV of the RAMPSOAK should be fetched from its own SP parameter.

8.14.7.4 Equations

There are no configurable equations for the RAMPSOAK algorithm. The ramp and soak segments are specified in the following parameters (also see Figure 8.18):

- Number of ramp/soak segment-pairs in the sequence—NORSSEQ
  Default = 2
- Ramp Rates, EUs per minute—RATE1 through RATE12
  Default = NaN
- Soak values—SOAKV1 through SOAKV12
  Default = NaN
- Soak times, in minutes—SOAKT1 through SOAKT12
  Default = 0

8.14.7.5 Override Feedback Processing

This algorithm does not participate in override-feedback processing.

8.14.7.6 RAMPSOAK Parameters

All of the significant parameters associated with the RAMPSOAK algorithm have been described. Refer to the *Advanced Process Manager Parameter Reference Dictionary* for additional parameter information.

8.14.8 Auto Manual (AUTOMAN)

8.14.8.1 Function

In cascade mode, this algorithm calculates a control output that is equal to the input value plus a bias value. The bias value is normally provided by a Universal Station operator. In manual mode, the output is controlled by a Universal Station operator or a user-written program. See Figure 8-23.

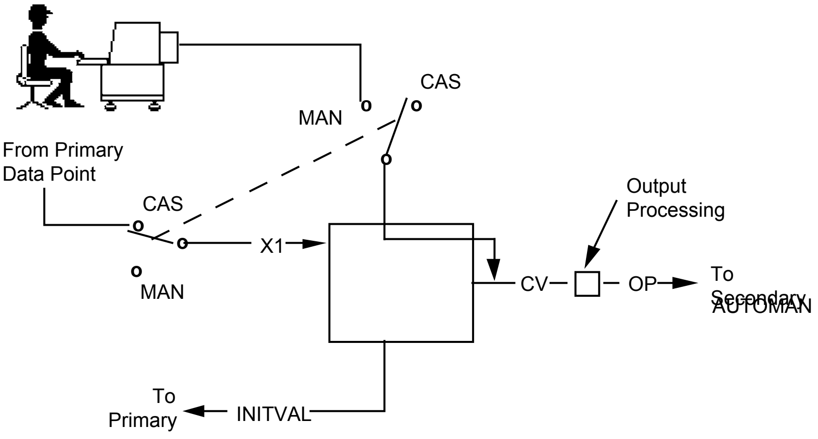
Equation A is designed to provide "bumpless" returns to cascade operation, even though its primary data point may not accept the initialization value from the AUTOMAN data point. Equation B provides automatic balancing of the biases between several auto manual stations and "bumpless" closing of cascades, with ramping of the initialization component.

8.14.8.2 Use

The AUTOMAN control algorithm typically serves as the ultimate secondary data point in a cascade-control strategy. As such it directly drives the control element (valve) through an analog output slot (a slot using a PID algorithm). See Figure 8-24.

This algorithm is very useful for a secondary data point that is one of two or more secondaries of the same primary data point (a "fan-out" configuration). In such a configuration, AUTOMAN can provide a "bumpless" output even if its primary doesn't accept AUTOMAN's initialization request, but accepts one from one of its other secondaries. This might happen because the AUTOMAN point is temporarily out of the cascade because it is inactive or because it is in Man mode.

For an example of output from a PID controller to two auto manual stations, see the example under Equation B (8.14.8.4).



CV = X1 +B

Where :

X1 = Input (usually from a Primary PID controller)

B = BO + BI

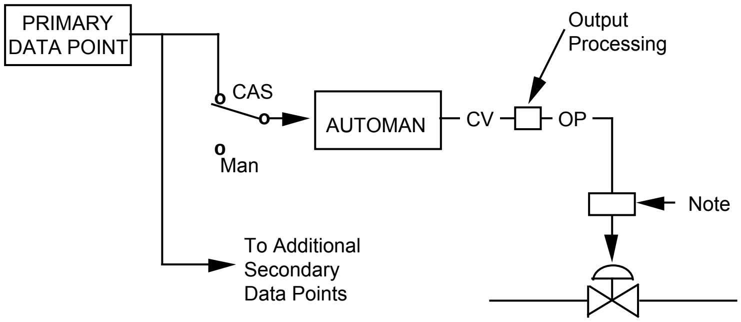
B = Overall Bias, consisting of:

Equation A:

BO = Operator-entered bias = K * X2 for Equation B

BI = Initialization Bias Component

Figure 8-23 — Functional Diagram, Auto Manual Control Algorithm 1319



NOTE: In a PM, this is an Analog Output point (slot) with PNTFORM = Component. On a DataHiway this is a CB, MC, or EC point (slot) using a CM, CMA, or DDC algorithm.

Figure 8-24 — AUTOMAN Serving as Ultimate Secondary in a Fan-Out Configuration 1320

8.14.8.3 Options and Special Features

8.14.8.3.1 "Bumpless" Returns to Cascade Operation

To support the use of this algorithm as one of the secondaries in a fan-out configuration, special handling of the bias value is provided. When the data point is configured for external initialization, the bias value, B, consists of two components.

```
B = B0 + BI
      |    |
      |    └── Internal adjustment-value storage.  Usually BI = 0.
      └─────── Internal storage for operator-entered or program-entered bias value
```

When the user or a program enters a bias value in B while in cascade operation, that value goes to B0 and BI is forced to zero. BI is an internal parameter that cannot be accessed by a user. B cannot be entered in Equation B.

When the cascade connection is broken (in Man mode, initialization-manual input, etc.) a value is calculated for the primary to initialize to, as follows:

```
INITVAL = CV - B0
```

When cascade operation resumes, the value in BI is calculated as follows:

```
BI = INITVAL - OPprim
               └──The actual output value from the primary data point (X1 input)
```

If the primary data point did accept AUTOMAN's initialization value, BI turns out to contain zero. If the primary did not accept INITVAL, BI contains a value that causes CV to contain the value it had just before the cascade closed. In either case, CV does not "bump."

If BI has a value other than zero, that value ramps to zero at a rate specified by the user in the RATE1 parameter. RATE1 is specified in engineering units-per-minute or NaN. If RATE1 is set to NaN, then the internal bias instantaneously changes to 0 and may cause a bump in the output. If a positive value is entered in RATE1, the value of BI ramps to zero at a rate specified in the RATE1 parameter. Thus, while the output doesn't "bump" it does ramp to the new value called for by the input from the primary.

For example, assume that the AUTOMAN data point is operating in Cas mode

```
X1 = 6

B = 5; BI = 0, BO = 5

CV = X1 + B = 11

RATE1 = 2 units per minute
```

The mode is changed to Man

```
INITVAL = CV - BO = 12 - 5 = 7
                    └── The operator changed the output value
B = 5; BO = 5, BI = 0
```

The primary does initialize

```
OPprim = 7
```

Go back to Cas mode

```
X1 = 7

BI = CV - BO - X1       (12 - 5 - 7 = 0)

B = B0 + BI = 5 + 0 = 5

CV = 7 + 5 = 12
             └── Same value as when in MAN mode
```

Go to Man mode again

```
CV = 12  (the operator didn't change it)

INITVAL = CV - B = 12 - 5 = 7

B = 5; BI = 0, B0 = 5
```

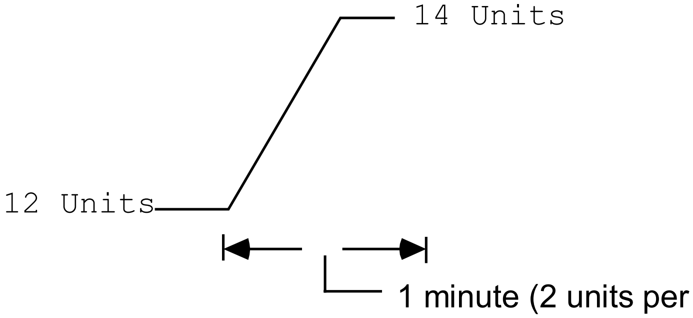
This time, the primary doesn't accept INITVAL, and the output of the primary went to nine units

```
OPprim = 9
```

Go back to Cas

```
X1 = 9

BI = INITVAL - OPprim = 7 - 9 = -2

B = 5 - 2 = 3

CV = 9 + 3 = 12
```

Now, because BI ramps from -2 units to zero, CV changes as follows:



8.14.8.3.2 Operating Modes

- Man
- Cas
- Bcas if RCASOPT = Ddc

8.14.8.3.3 Remote Cascade Options

Only the Ddc option is supported.

8.14.8.3.4 Engineering Unit Ranges

You must configure the X1 input range in XEULO and XEUHI. The output engineering unit ranges (CVEULO and CVEUHI) always follow the X input ranges.

8.14.8.3.5 Restart or Point Activation

On a cold or warm restart or when the data point is activated, initialization takes place as described under 8.14.8.3.1. On a hot restart, initialization does not occur.

8.14.8.3.6 Error Handling

If the X1 input has a bad-value status (or Equation B and X2 is bad), the CV value is bad but the data point remains in the same mode. When the bad input returns to normal, CV is recalculated and an initialization request is sent to the primary.

8.14.8.4 Equations

This algorithm has two forms:

Equation A

```
CV = X1 + B
```

Where B = B0 + BI

X1 is the input that is typically pushed from the primary PID controller and B is the overall bias, which consists of the operator set bias component (BO) and the initialization bias component (BI).

This form of the equation requires NO input connection. The number of input connections is defaulted to 0; however, it can be increased to 1, allowing the X1 input to be fetched with an input connection (X1 fetched by connection number 1).

Equation B

```
CV = X1 + B
```

Where:

CV	=	Control algorithm output in Engineering Units
X1	=	Initializable input pushed from primary
B	=	Bias Value, and
B	=	(K ∗ X2) + BI
X2	=	input connection fetched with input connection #1
BI	=	Initialization Bias Component

X1 is the input that is typically pushed from the primary PID controller and B is the overall bias, which consists of the automatic bias adjustment (K ∗ X2) and the initialization bias component (BI). X2 is fetched with an input connection (#1) from a calculated input. Bumpless closing of cascades is provided the same as in Equation A (with ramping of the initialization component). The overall Bias B is a read-only parameter and cannot be stored to.

Equation B requires only one input connection — X2. The number of input connections is defaulted to one; however, it can be increased to two, allowing the X1 input to be fetched with input connection #2.

Consider the following example:

If a PID controller outputs to two auto manual stations, the operator may want the total required offset between the outputs of the two auto manual stations to be distributed equally above and below the PID's output. The operator can store the bias (= half the required offset) to the first auto manual station (configured for Equation A). The second auto manual must be configured with Equation B, have K of **-1.0**, and get its X2 input from the **B0** parameter of the first one. Thus, whenever the operator changes B0 of the first auto manual point, the bias of the second point tracks automatically.

Alternately, the strategy could be configured so that the bias value is held in a numeric. Each auto manual point would then fetch the numeric value into its X2 input. In this configuration, bias balancing is achieved by setting K on the two auto manual stations to +0.5 and -0.5, respectively.

8.14.8.5 Initialization

Refer to 8.14.8.3.1 for information on when initialization can occur. Initialization value = CV - B.

8.14.8.6 Override Feedback Processing

When the data point's secondary uses an Override Selector algorithm (see Figure 8-25), the following functions take place:

- If this point is in Cas mode, Override Status PTORST is returned to this point through special processing. If that status is "not selected," an override-feedback value is calculated as follows, and it is passed on to the AUTOMAN data point's primary:

```
ORFB = ORFBSEC - B
```

For more details on override control see subsection 8.13 of this publication.

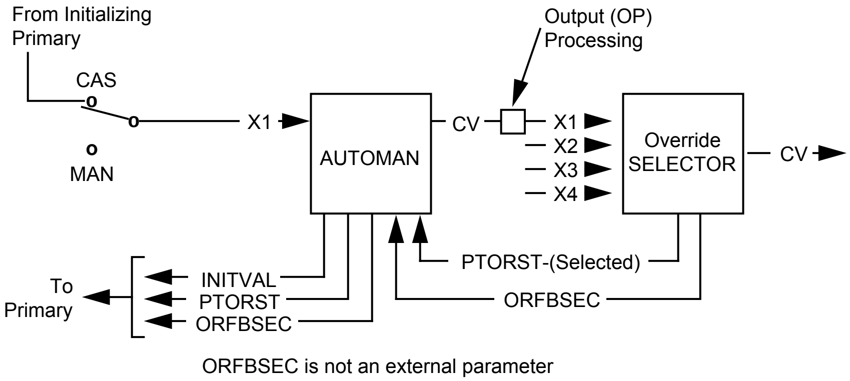


Figure 8-25 — Override Feedback Processing

3370

8.14.9 Incremental Summer (INCRSUM)

8.14.9.1 Function

This algorithm calculates the sum of the incremental changes in up to four input values. The output is obtained by adding the sum of the changes in all inputs, after each input is multiplied by a scale factor. See Figure 8-26.

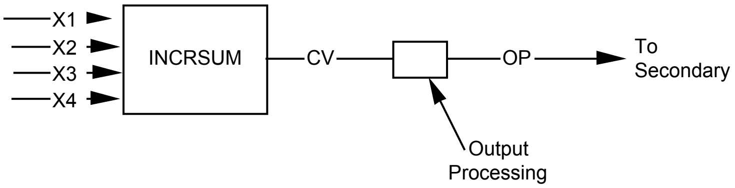


Figure 8-26 — Functional Diagram, Incremental Summer Control Algorithm

1322

8.14.9.2 Use

This algorithm is typically used where more than one primary data point is used to manipulate the setpoint of the same secondary data point. The primaries usually use PID algorithms, and are connected to an INCRSUM data point. The output from the INCRSUM data point is connected to the secondary, as shown here. This is sometimes referred to as a "fan-in connection."

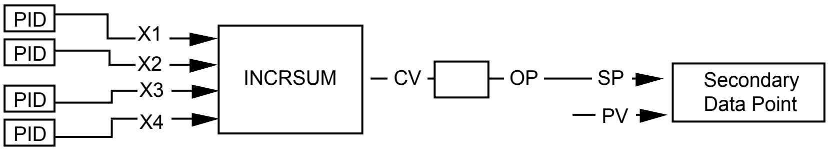


Figure 8-27 — Fan-in Connection

2093

8.14.9.3 Options and Special Features

8.14.9.3.1 Control Input Connections

This algorithm requires NO input connections. The number of input connections is defaulted to 2, but it can be increased to up to four. For each input connection, the user must also specify the destination parameter, X1, X2, X3 or X4.

8.14.9.3.2 Handling of Full Value, Floating PID Outputs

Special handling of the outputs of PID data points is necessary in the Incremental Summer, because it is the dynamic operation of the PID that is significant—the full-value output is not significant. As the process variables change, PID outputs have no direct relation to the input, therefore they are said to be "floating." The incremental summer responds only to changes in the PID outputs, and calculates a full-value output to be applied to the control device (valve) in the process or to the secondary data point.

While the PID algorithms handle SPs, PVs, and outputs as percentages of the configured range, the Incremental Summer performs its calculations in engineering units.

To prevent a primary PID point in Auto or Cas mode from winding-up beyond its own output limits, every time the Incremental Summer point is processed it changes the PID's output value to its own CV value after converting to percent. This does not affect the dynamic changes in the PID outputs.

If a primary PID is in Man mode, the PID's CV is not changed by the Incremental Summer.

The following restrictions apply to the PID points that are the Incremental Summer point's primaries.

- Each must have only one control-output connection.
- Each must reside in the same APM box.
- Each must be a normal PID (8.14.1) or a PID Feedforward.

8.14.9.3.3 Engineering Unit Ranges

You must configure the X-input ranges in XEULO and XEUHI. The output engineering unit ranges (CVEULO and CVEUHI) always follow the X input ranges.

8.14.9.3.4 Changes to Incremental Summer Output by User-Written Programs

User-written CL programs in the AM or user-written programs in a Computing Module can directly store new values in the CV parameter of the Incremental Summer's data point while the Incremental Summer point is in BCas mode, or if in CAS mode and RCASOPT is not = DDC, and normally operating. This causes a shift in the CV value but the dynamic changes continue, because the Incremental Summer dynamically increments or decrements CV each time the point is processed.

8.14.9.3.5 Override Control Strategy

If the Incremental Summer's secondary is an Override Selector data point, and its Override Status (PTORST) is not selected, the CV of the INCRSUM data point is set equal to the override feedback value. During the next pass of forward calculation, the incremental summer computes its CV in the normal fashion.

This is very useful for cases where several feedforward signals are added to the output of a PID controller (the addition done by the incremental summer algorithm) and the overall result is fed to an override selector. Because the output of the PID controller tracks the output of the incremental summer, which in turn tracks the output of the override selector when not selected, the output of the PID controller is prevented from winding up when not selected by the override.

8.14.9.3.6 Operating Modes

This algorithm operates only in Man and Cas modes, and in Bcas if RCASOPT = Ddc. Because the output of each PID primary tracks the OP output value of the INCRSUM data point, switching this point from Man to Cas does not "bump" the process.

Note: The dynamic correction calculated by a PID primary appears as a change in that data point's output after it is processed again and the Incremental Summer actually uses this change as its input. The full-value outputs of PIDs connected to the Incremental Summer have no meaning.

8.14.9.3.7 Remote Cascade Options

Only the Ddc option is supported.

8.14.9.3.8 Restart or Point Activation

On a warm restart, or point activation, initialization takes place as described under 8.14.9.5.

8.14.9.3.9 Bad Control Handling

When an input becomes bad (= NaN), it is not used in the calculation. No bad control alarm is generated. When the bad input returns to a normal value, its use in the output computation is resumed in the CV calculation without "bumping" the value. This feature allows adding multiple feedforward signals to the output of a PID controller.

NOTE

If a single feedforward input is needed, the PID feedforward control algorithm should be used.

8.14.9.4 Equations

CV is calculated as follows:

```
CV(n) = CV(n-1) + K1*[X1(n) - X1(n-1)]

                + K2*[X2 - X2(n)(n-1)]
                        .

                        .

                        .
                + Km*[Xm(n) - Xm(n-1)]
```

Where:

CV(n) = Current full value of the output of this algorithm in engineering units.

CV(n-1) = Past full value from the data point (value from the last time the data point was processed).

m = The number of inputs actually used (m = 1 through 4).

K1 through Km = User-specified scale factors (gains). K1 through K4 default to 1.0.

X1(n) through Xm(n) = Current values of each X input in use.

X1(n-1) through Xm(n-1) = Past value of each X input (value from the last time the data point was processed)

8.14.9.5 Initialization

Initialization occurs in Man mode when the output is indisposable, or when recovering from a bad CV value (NaN), as follows:

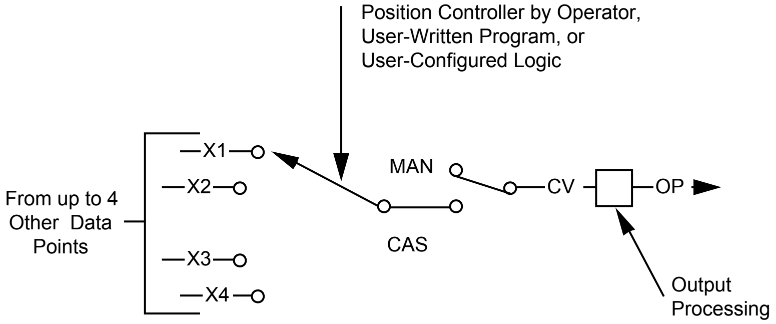
- An initialization request is sent to all primary data points (up-to-four).
- The Incremental Summer's CV value and past values are initialized:

```
CV(n-1) = CV

X1(n-1) = X1
     .

     .

     .

Xm(n-1) = Xm
```

8.14.10 Switch (SWITCH)

8.14.10.1 Function

This algorithm operates as a single-pole, 4-position rotary switch. An operator at a Universal Station, a user-written program, or user-configured logic can change the position of the switch, thereby selecting any one of the four inputs to be the control-algorithm output value, CV.



Equation A: Operator Controls Switch Position

Equation B: Program or Logic Controls Switch Position, Operator can store to SELXINP if all S1-S4 flags are OFF

Figure 8-28 — Functional Diagram, Switch Control Algorithm

1342

8.14.10.2 Use

The SWITCH control algorithm is used to allow the operator at a Universal Station to alter control strategies by selecting any of four inputs to be passed on to the output, if Equation A is chosen. If Equation B is chosen, a Logic Slot can change the switch position, or the operator can store to SELXINP if all four input selection flags (S1, S2, S3, and S4) are Off.

You can use SWITCH to select inputs from differing sources and to pass them on to a single destination or you can use more than one SWITCH data point to switch a single source to differing destinations. Figure 8-29 shows an example of each of these situations.

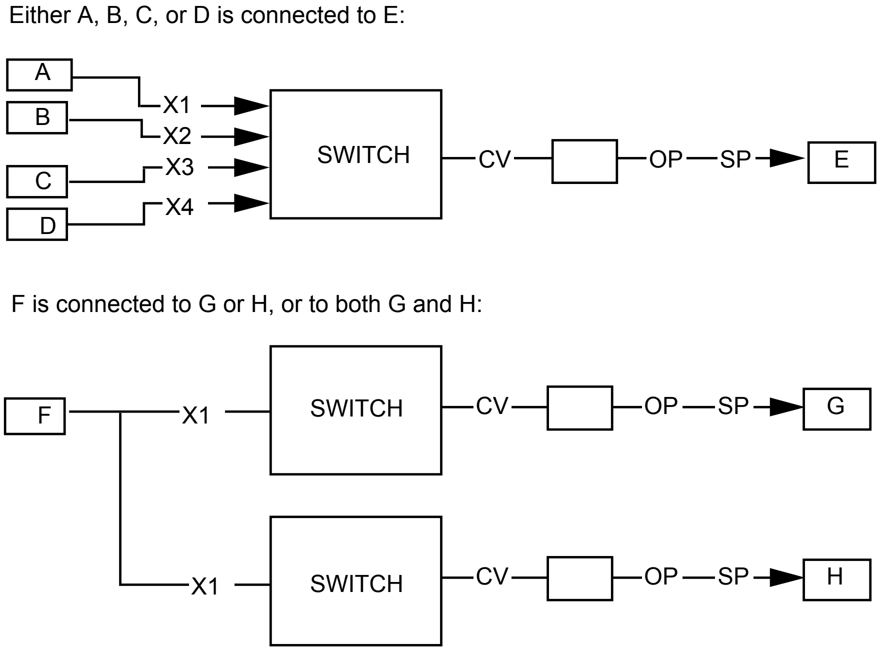

Either A, B, C, or D is connected to E:
A
B
C
D
X1
X2
X3
X4
SWITCH
CV
OP
SP
E
F is connected to G or H, or to both G and H:
F
X1
SWITCH
CV
OP
SP
G
X1
SWITCH
CV
OP
SP
H

Figure 8-29 — Switching Examples

1343

8.14.10.3 Options and Special Features

8.14.10.3.1 Control Input Connections

This algorithm requires NO input connections. The number of input connections is defaulted to 0; however, it can be increased to up to 4. For each input connection, the user must also specify the destination parameter as X1, X2, X3 or X4.

8.14.10.3.2 Engineering Unit Ranges

The X-input engineering unit ranges (XEULO and XEUHI) must be specified for X-inputs. The output engineering unit ranges (CVEULO and CVEUHI) always follow the X-input ranges.

8.14.10.3.3 Operator Control of Switch Position--Equation A

If Equation A is chosen, an operator at a Universal Station can change the switch position in one of two ways:

- By altering the value in SWITCH data-point parameter SELXINP—The value in SELXINP specifies the Xn input selected. The corresponding S1, S2, S3, or S4 switch indicator goes On and the other three switch indicators go Off.
- By changing the desired switch indicator from Off to On—When one of the S1, S2, S3, or S4 indicators is changed to On, all others go Off. SELXINP then indicates the position selected.

The second method is very useful when operating with custom displays. The displays can be built to allow the operator to see the positions and strategies selected unless S1 through S4 are all OFF.

8.14.10.3.4 Control of Switch Positions–Equation B

Equation B is primarily intended for automatic switching between inputs with the help of logic (or user programs). In the APM, an operator is allowed to change the switch position, by storing to the SELXINP parameter when configured for Equation B, if all input selection flags (S1, S2, S3 or S4) are Off.

With Equation B, turning an S1, S2, S3, or S4 indicator On does not turn the others Off, as it does with Equation A; therefore, the switch position is changed by storing On and Off in the S1-S4 parameters as follows:

S1	**S2**	**S3**	**S4**	**Xn**	**SELXINP**
On	-	-	-	X1	SELECTX1
Off	On	-	-	X2	SELECTX2
Off	Off	On	-	X3	SELECTX3
Off	Off	Off	On	X4	SELECTX4

Where "-" means On or Off does not affect the switch position.

S1 has the highest priority, S4 has the lowest priority.

8.14.10.3.5 Tracking Option

You can configure the SWITCH algorithm for the tracking option, which causes non-selected inputs to track the selected input value. This allows the switch position to be changed without "bumping" the output.

When tracking option is configured, the primaries connected to non-selected inputs can be initialized. Should one of the primaries not accept the initialization value from the SWITCH data point, the output may bump when that input is selected. (A primary might not accept an initialization value because it has more than one secondary and accepts initialization from one of its other secondaries.)

NOTE

For the tracking option to work, the Switch input must come from an APM primary Regulatory Control point. The output destination from the RegCtl point must specify the Switch point (that is, the Switch input must be pushed from the RegCtl point).

When SWITCH is included in an override control strategy, the tracking option must be configured.

8.14.10.3.6 Operational Modes

The Switch-control algorithm operates in the following modes:

- Man
- Cas
- Bcas if RCASOPT = Ddc

8.14.10.3.7 Remote Cascade Options

Only the Ddc option is supported.

8.14.10.3.8 Restart or Point Activation

On a cold or warm restart, or when the SWITCH data point is activated, initialization takes place, as configured. See 8.14.10.5.

On a hot restart, normal operation resumes with no initialization.

8.14.10.3.9 Error Handling

If a selected input has a bad-value status, the CV value goes bad (NaN), but the operating mode does not change. When the status of the selected input is again good, CV is recalculated, an initialization request is sent to the primary data point.

8.14.10.4 Equations

Equations A and B:

```
CV = Xn
```

n = 1, 2, 3, or 4

SELXINP = The selected input, which can range from SelectX1 through SelectX4.

S1 through S4 indicate the switch selection, as described under 8.14.10.3.3 or 8.14.10.3.4.

Where:

CV = The control output value in engineering units.
SELXINP = The selected-input. Default = SelectX1.
S1 through S4 = Switch indicators
M = The number of inputs configured. Default = 2.

NOTE

See 8.14.10.3.4 for special information regarding Equation B.

8.14.10.5 Initialization

When an initializing condition occurs, an initialization request is sent to the selected primary and the initialization value is the present CV value.

If the tracking option is configured (see 8.14.10.3.5), the non-selected primaries are continually initialized.

8.14.10.6 Override Feedback Processing

If this data point's secondary is an Override Selector point and if this point is in Cas mode, when override-feedback processing takes place, override status and an override value are passed to this point's primary. The status is in parameter PTORST.

If PTORST indicates not selected, the value passed to the selected primary in ORFB is equal to the value received from the secondary in ORFBSEC.

8.14.10.7 Switch Parameters

In addition to the parameters already mentioned, parameter TRACKING is associated with the SWITCH algorithm. Refer to the *Advanced Process Manager Parameter Reference Dictionary.*

8.14.11 Override Selector (ORSEL)

8.14.11.1 Function

The Override Selector Algorithm is used with up-to-four PID inputs, all of which are initializable. The input with the highest value or the input with the lowest value is selected and passed on to the output of this data point. The algorithm can operate as a simple selector or an override option can be configured that prevents PID points in an override-control strategy from winding up. Refer to Figure 8-29.

If the override option is configured (OROPT = On), an operator can put the ORSEL point in a bypass state and any of the inputs can be bypassed; that is, not used in the algorithm calculation. Four logical parameters are provided to select the input to be bypassed. These parameters are BYPASSX1, BYPASSX2, etc. There is an additional parameter, BYPASS. When BYPASS = On, the input Xn for which BYPASSXn is On is initialized (equal to the selected input) and not used in the calculation of the output. When all four inputs are bypassed, the output is maintained at the last values.

All bypass parameters can be stored by the operator, or any system function, but only the BYPASS parameter is available for changing at the group display level.

8.14.11.2 Use

This algorithm can be used, without the override option, as a simple selector that selects either the highest or the lowest of the connected and active inputs. With the override option, it is used for override-control strategies where a process variable is measured and normally controlled, but where another variable is selected to constrain the controlled variable, under a specified condition. This is often referred to as "multivariable-constraint control."

Figure 8-31 illustrates an override strategy. The X1 input to the ORSEL point is normally selected and applied as the setpoint to the fuel-flow controller. If the value of the air flow multiplied by some ratio exceeds the fuel-flow setpoint, the air flow constrains the fuel flow.

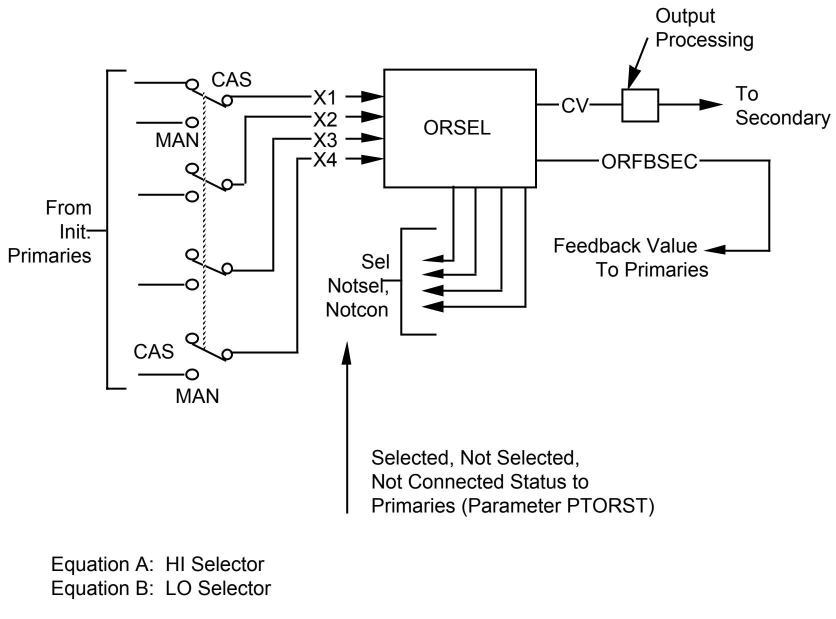


Figure 8-30 — Functional Diagram, ORSEL Algorithm

2094

In a strategy like that of Figure 8-31, Equation A, the override option is configured. PID data points connected to non-selected inputs are prevented from "winding up" by forcing their outputs to track the override feedback signal (ORFBSEC). For more detail on such strategies, refer to 8.13.

The simple selector (override option not configured) can be set up to initialize one input, but not all inputs, by using control-input connections for the inputs that are not to be initialized, and by using a control-output connection from the point that is connected to the input to be initialized.

NOTE

There are some important guidelines that must be observed when configuring an override-control strategy. See 8.13 and the configuration information below.

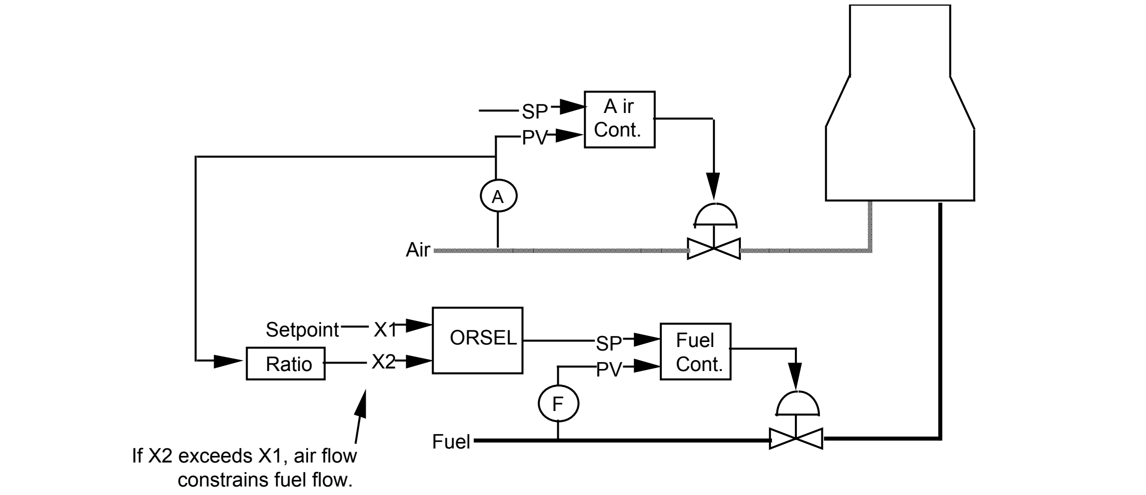


Figure 8-31 — Example of an Override Control Strategy

8.14.11.3 Options and Special Features

8.14.11.3.1 Control Input Connections

This algorithm requires NO input connections. The number of input connections is defaulted to 0; however, it can be increased to up to 4. For each input connection, a destination parameter X1, X2, X3 or X4 must be configured.

8.14.11.3.2 Override and Bypass Options

If the override option is configured, PID points connected to non-selected inputs are prevented from "winding up" by forcing their outputs to track the override-feedback signal (ORFBSEC).

Override Offset

A configuration parameter called Override Offset (OROFFSET = On) is provided to control the behavior of any PIDs connected to the non-selected inputs. If Override Offset is configured, the output of the PID whose override status is not selected is set equal to the override feedback value, plus gain times error (after converting to percent). If the override offset is not configured, the output of the PID whose override status is not selected is set equal to the override feedback value (after converting to percent).

Bypass Options and Parameters

When configured as an override selector, any of the inputs X1, X2, X3, and X4 can be individually bypassed (not used in the algorithm calculation). Four logical parameters are provided to select the input to be bypassed. These parameters are BYPASSX1, BYPASSX2, BYPASSX3, and BYPASSX4. An additional parameter, BYPASS is provided. When BYPASS is On, the input Xn for which BYPASSXn is On is initialized (set equal to the selected input) and not used in the calculation of the output. When all four inputs are bypassed, the output is maintained at the last value.

These parameters are available only when the override option is configured (OROPT = On).

All bypass parameters can be stored by the operator, or any system function, but only BYPASS is available for changing at the group display level.

NOTE: When a selected input is bypassed, all parameters go through 1-shot initialization.

8.14.11.3.3 Engineering Unit Ranges

The engineering-units ranges for the X1 through X4 inputs must be configured in parameters XEUHI and XEULO. These parameters contain the high and low values for the range, which is the same for all four inputs. The output engineering unit ranges (CVEULO and CVEUHI) always follow the X-input range.

8.14.11.3.4 Operating Modes

Because a data point that uses ORSEL is always a secondary to at least one other data point, this algorithm operates only in the following modes:

- Cas
- Man
- Bcas if RCASOPT = Ddc

8.14.11.3.5 Remote Cascade Options

Only the Ddc option is supported.

8.14.11.3.6 Restart or Point Activation

On a warm restart, or on activating the data point, the CV is initialized to the value returned from the secondary and an initialization request is sent to all of the primaries.

8.14.11.3.7 Error Handling

In Cascade mode with BYPASS Off, if any input has a Bad-Value status, CV's value is bad (NaN) and the mode doesn't change when the CV value goes bad. Also, in Cascade mode with BYPASS On, any non-bypassed input (BYPASSXn=Off) that has a Bad-Value status causes CV to go bad (NaN).

If a bad input, that was causing CV to be Bad, returns to Normal, CV returns to normal, CV is made equal to OP, and all bypassed primaries are initialized (see subsection 8.14.11.5).

8.14.11.4 Equations

There are two equation choices:

Equation A—Select the higher of the connected, active inputs.

```
CV = highest of X1 through Xm
```

Equation B—Select the lower of the connected, active inputs.

```
CV = lowest of X1 through Xm
```

Where

CV = The control-algorithm output in engineering units

X1 through X4 = The four available inputs

M = The number of inputs configured. Default = 2.

For either equation

SELXINP = The selected input: SelectX1 through SelectX4. If more than one input has the highest (EqA) or the lowest (EqB) value, the lower-numbered input is selected, e.g., if X2 and X3 have exactly the same highest value (EqA) SELXINP contains SelectX2.

8.14.11.5 Initialization

Initialization requests from secondary points are ignored as long as the ORSEL output is disposable.

If the ORSEL is initialized because it is in Manual mode, the output is indisposable, or it has just returned from bad control status, CV is made equal to INITVAL from the secondary, and SELXINP is set to None. When the ORSEL is in Manual mode, an initialization request is passed on to all primaries with an initialization value equal to CV. When the ORSEL is initializing because the output is indisposable or it is returning from bad control, an initialization request is passed on to all primaries whose input are bypassed (BYPASS = On and BYPASSXn = On) with an initialization value equal to CV.

8.14.11.6 Override Feedback Processing

8.14.11.6.1 Override-Feedback Initiation

If the override option is configured for the ORSEL point and the ORSEL point is in Cas mode and not initializing, it propagates override-feedback information to its primary points and on "upstream."

When BYPASS is Off, the appropriate NotCon, Sel, NotSel status is given to ORSEL's primaries in PTORST, and the override-feedback value that is passed to the primaries is calculated as follows:

```
ORFBSEC = CV
```

Where ORFBSEC is the feedback value from the secondary. (ORFBSEC is an internal parameter)

If, under the above conditions, BYPASS is On, the status sent to the primary connected to the nonbypassed selected input is Sel, NotCon is sent to the bypassed primaries and they are all initialized. NotSel is set to the nonbypassed non-selected primaries.

If there is more than one Override-Feedback data point in a strategy, only the one nearest the final control element (the "most downstream" point) initiates override feedback.

8.14.11.6.2 Override-Feedback Propagation

Override-feedback propagation is the passing of status and feedback values, from the initiating Override Feedback Selector, "upstream" through one or more other data points.

If a "downstream" Override Feedback Selector requests status and value propagation, an "upstream" Override Feedback Selector" propagates the value and status "upstream," only if

- it is configured as an Override Selector (override configured), and
- it is in Cas mode, and
- its output is disposable.

If so,

```
If PTORST ( Point Override Status) = SEL (selected), ORFB = CV
  else
ORFB = ORFBSEC
```

Where ORFBSEC is the feedback value from the secondary.

And, PTORST status sent to the selected primary is the same as that received from the secondary. If BYPASS is false, the status to all other primaries is NotSel. If BYPASS is true, non-selected inputs are initialized, so the status sent to them is NotCon.

8.14.11.7 ORSEL Parameters

In addition to the parameters already mentioned, the following parameters are associated with the ORSEL algorithm. Refer to the *Advanced Process Manager Parameter Reference Dictionary*.

BYPASS

BYPASSX1

BYPASSX2

BYPASSX3

BYPASSX4

CTLEQN

OROFFSET

OROPT

8.14.11.8 Processing Order

Processing order is always important, but especially for the ORSEL strategy. ORSEL functions will not work properly if point processing order is incorrect.

APM BOX FLAG, NUMERIC, STRING, TIME, AND TIMER GLOBAL VARIABLES
Section 9

This section describes the global variables available in the Advanced Process Manager. They are the Flags, Numerics, Timers, Times, and String variables. Additional information about the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

9.1 GLOBAL VARIABLES

The APM provides the following global variables

- 16,384 Flag variables
- Up to 16,384 Numeric variables
- Up to 16,384 8-character String variables
- Up to 4096 Time variables
- 64 Timer variables

The full 16,384 flags and 64 timer variables are always present. Numerics, strings, and times are configured by APM Node Specific Configuration parameters such as NSTRING (number of strings). Configuration is determined by considering the Memory Units needed for your system (see Section 3). Box global variables do not require any PUs.
Some specific types of points (e.g., Process Module) provide similar local variables.

9.2 BOX FLAG DATA POINT

A Box Flag Data Point is a 2-state (On and Off) point that is used for storing a Boolean value. The value can be supplied by the operator, by the sequence program, by an output connection from another APM point, by any APM (or PM) box on the same UCN, or by a node on the LCN. There are 16384 Flag Data Points (slots) available in an APM box. The first 2047 flag points are tagable and can be configured as Full or Component points. The first 4095 Box Flags are accessible from the LCN. The remaining flags must be accessed through an array point.

Flag data points are not scheduled and only off-normal alarms are processed. Their states change when they are accessed by other functions such as an operator or a user-written program.

Flag points do not require a tag name. They can be accessed by !BOX.fl(i) or $NMxxNyy.fl(i) where xx is the network number and yy is the APM node number. After configuration, flag variables are initialized to Off. A functional diagram of the Flag Data Point is shown in Figure 9-1.

As shown in this figure, the input to the flag point is provided by parameter PVFL, which can be On or Off. PVFL will set the flag-point PV state to the same state as PVFL. The PV is then available as an output from the flag point and its current state can be accessed by other points in the APM and in the system.

PVFL will also light the STATE1 and STATE0 boxes on the Universal Station Displays, depending on its state. If PVFL is On, the STATE1 (upper) box will be lighted; if PVFL is Off, the STATE0 (lower) box will be lighted.

9.2.1 Alarming

The first 128 flag points (slots) can be configured for off-normal alarming. An alarm will be generated when the PV of the flag point is changed from STATE0 (Off) to STATE1 (On).

Alarming is available only if the flag point has been configured as a Full point.

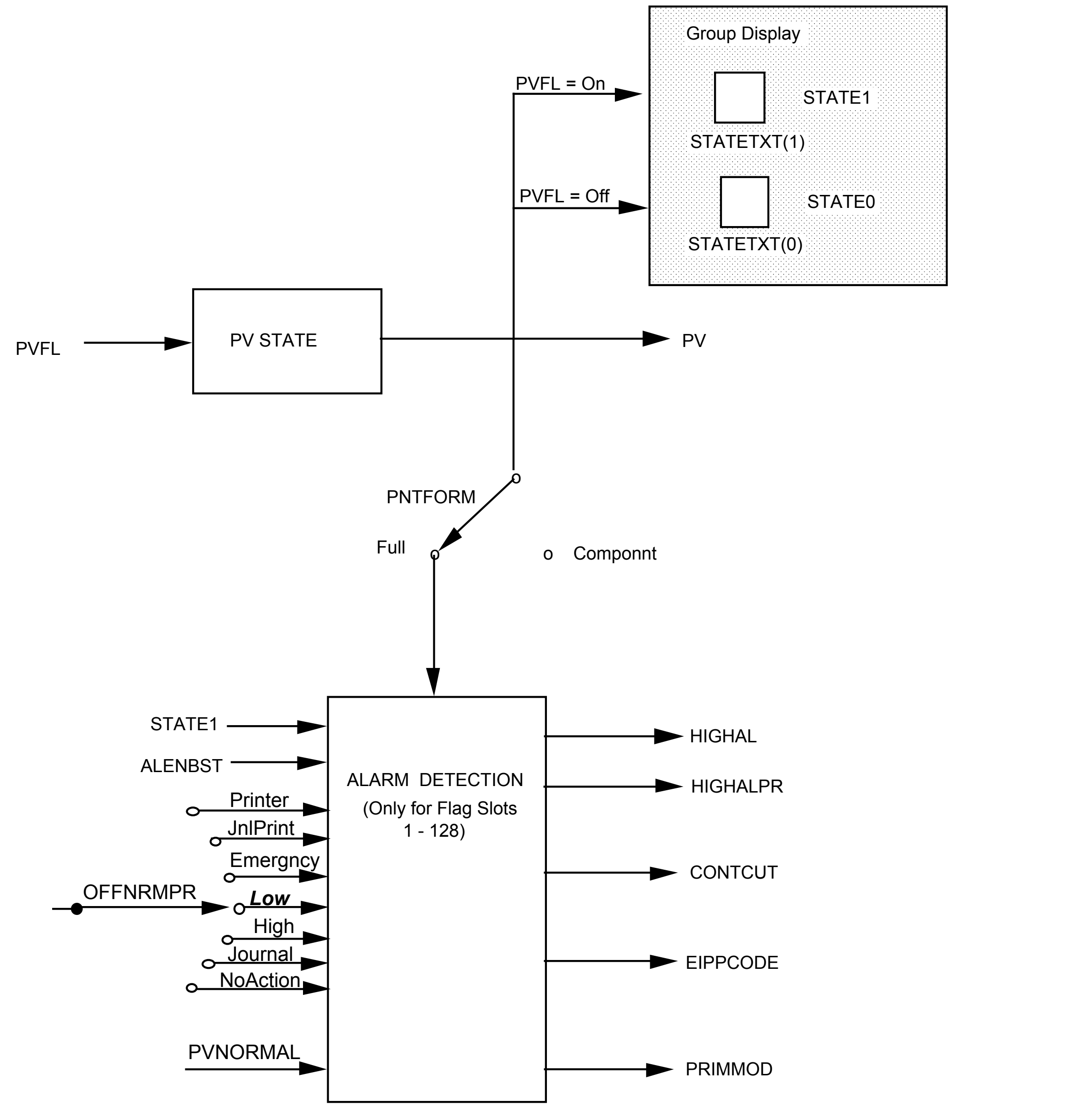


Figure 9-1 — Box Flag Data Point, Functional Diagram

New

9.3 BOX NUMERIC DATA POINT

Numeric Data Points are used to store numeric values that can be used for batch/recipe operations, or they can be used as a scratch pad to store the intermediate results of calculations. The values in a numeric point are real numbers that have been entered by the operator, or by a sequence program, or other system elements. Box Numeric Data Points are full points and cannot be configured as component points.

Up to 16,384 Numeric slots can be configured as part of each APM box. Numeric slots do not require a tag name. They can be accessed by !BOX.nn(i) or $NMxxNyy.nn(i) where xx is the network number and yy is the APM node number. After configuration, numeric variables are initialized to NaN. An Array point must be used to access index numbers greater that 4095 over the LCN.

These points are accessible to the sequences in the same APM box, to any APM (or HPM or PM) box on the same UCN, and to any node on the LCN. The first 2047 Numeric points are tagable and can be configured as named data points. The first 4095 Box Numerics are accessible from the LCN. The remaining Numeric variables can be accessed through an array point.

Numeric data points are not scheduled and are not processed. Their parameter values change when they are accessed by a system activity, such as by an operator or a sequence program.

9.4 BOX TIMER DATA POINT

The Box Timer Data Point allows the operator and the sequence program to time the process events, as required. This type of data point keeps track of the elapsed time after the timer has been started and provides an indication when the elapsed time has reached the predefined limit. Box Timer Data Points are component points and cannot be configured as full points.

There are 64 Timer Data Points in each APM Box, and like box numeric and flag points, timer data points do not require a tag name. These points are accessible to the sequences in the same APM box, to any APM (or HPM or PM) box on the same UCN, and to any node on the LCN. A functional diagram of the Timer Data Point is shown in Figure 9-2.

To use a Timer Data Point, an operator at a Universal Station or the sequence program loads a preset time value (the length of time that the timer is to run in seconds or minutes) into parameter SP. Once the preset value is entered, the timer is started by setting parameter COMMAND to Start. (The timer can also be stopped, reset, or reset and restarted through the COMMAND parameter.)

The time value in PV starts at zero and increments toward the preset time value when the timer is processed. (The timer is processed once each second.) The RV parameter indicates the time remaining until the timer reaches its limit (SP - PV). When PV = SP, the status parameter SO is set to On to indicate that the time limit has been reached. The values in PV and SP can range from 0 to 32000 seconds or minutes, as configured.

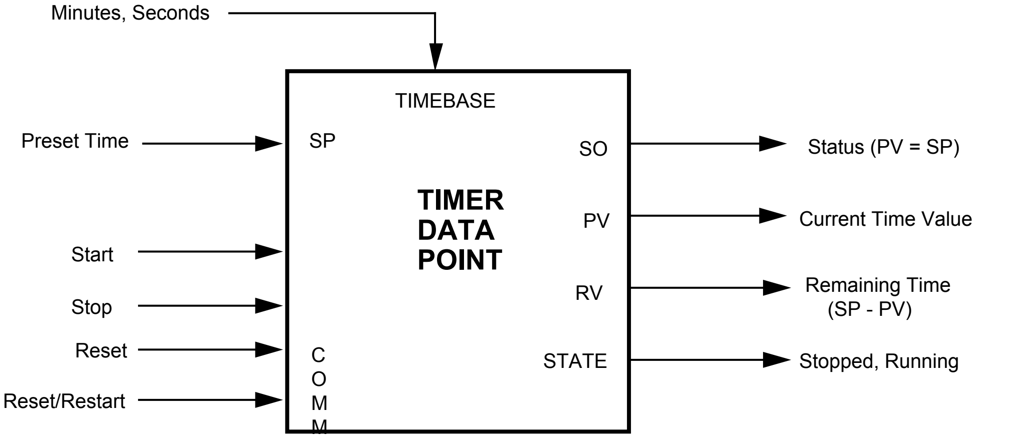


Figure 9-2 — Box Timer Data Point, Functional Diagram

9.5 BOX TIME VARIABLES

Time variables can be used to store date/time information such as the date machinery was put in service, when the machinery was serviced last, when the machinery was last used, etc.

The APM box provides up to 4096 global Time variables. The maximum limit is determined by the Node Specific Configuration parameter NTIME (number of times). Time variables are not configurable with a tag name. They can be accessed by !BOX.TIME(i) or $NMxxNyy.TIME(i) where xx is the network number and yy is the APM node number. The first 4095 Box Times are accessible from the LCN. An array point must be used to access number 4096. After configuration, time variables are initialized to 0.

9.6 BOX STRING VARIABLES

String variables can be used to store descriptive text data such as the brand name or model of machinery, who to contact for service, etc.

The APM box provides up to 16,384 8-character String variables. The upper limit is determined by the Node Specific Configuration (box) parameter NSTRING (number of strings). The first 4095 Box Strings are accessible from the LCN. An array point must be used to address strings with an index of greater than 4095.

String variables are not configurable with a tag name. They can be accessed by !BOX.STRn(i) where n is the string length or $NMxxNyy.STRn(i) where xx is the network number and yy is the APM node number. After configuration, string variables are initialized to spaces.

ARRAY POINTS
Section 10

This section describes the Array point available in the Advanced Process Manager. Additional information about the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

10.1 ARRAY POINT OVERVIEW

Array points provide access to two sources of data—

- APM box global variables
- external data from/to Serial Interface devices

In the first case, an Array point can define a section of the box variables as its own data (see Figure 10-1). Access to single variables over the LCN is limited by the upper index (4095), but using an Array point you can access upper Flag, Numeric, String, and Time variables. The Array point can access and fetch string data in lengths of 8, 16, 32, or 64 characters.

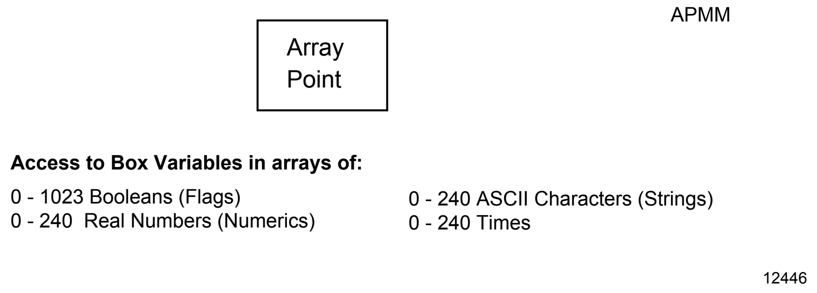
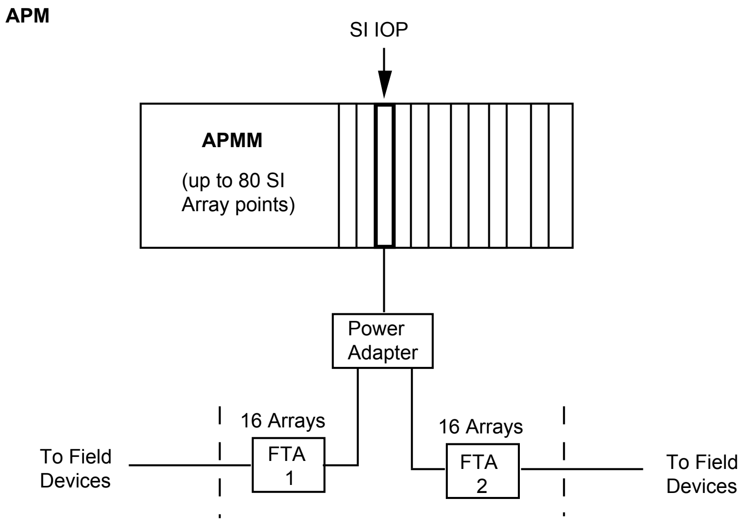

APMM
Array Point
Access to Box Variables in arrays of:
0 - 1023 Booleans (Flags)
0 - 240 Real Numbers (Numerics)
0 - 240 ASCII Characters (Strings)
0 - 240 Times
12446

Figure 10-1 — Array Point Access to Box Global Variables

In the second case, large quantities of data from a Serial Interface (SI) IOP can be scanned and imaged (read), and used as if it were local data to the Array point (see Figure 10-2).

Writes through the Serial Interface are also permitted. The Serial Interface IOP module is described in Section 2 of this manual, and its use with an Array point is discussed later in this section.



Each Array point can contain up to 512 Booleans (Flags), or 16 Reals (Numerics) or 32 Integers (Numerics), or 64 byte-sized Integers (Numerics), or 64 ASCII Characters (Strings)

Figure 10-2 — Serial Interface to an Array Point

12447

10.1.1 Node Specific Configuration Requirements

Array point configuration begins with APM Node Specific Configuration. You can allocate up to 256 array points through the parameter NARRSLOT. Other Node Specific Configuration parameters that affect the Array point are—

- NNUMERIC - the number of Box Numerics (in multiples of 16) that you want available.

- NSTRING - the number of Box Strings that you want available (in multiples of 16). Note that Box Strings are only available as 8-character strings.

- NTIME - the number of Box Times (in multiples of 32) that you want available.

- SCANPER - if you intend to access Serial Interface data, this parameter specifies the period at which the APMM scans SI data and maps it to the Array points. The APMM can scan SI IOP data at three rates and this determines the maximum number of Array points with SI connections that you can have:

	Maximum Number of
SCANPER	SI/Array Points
1 Second	80
0.5 Second	40
0.25 Second	20

Finally, you must select SI as the Module Type on the IO Module Configuration pages for each SI IOP board that you wish to implement.

Note that 16,384 box flags are always available. The number of variables of each particular global box variable data type available to an Array point is further limited by configuration parameters for the Array point.

10.1.2 Array Parameter Names

Data elements in the Array point are referred to by the Array point tagname, type of variable and index number (i). For example ARR01.FL(9). The complete list follows:

	Parameter
Variable	Reference
Flags	FL(i)
Numerics	NN(i)
Strings	STRn(i)
Times	TIME(i)

In the case of strings, n refers to the string length, 8, 16, 32, or 64. For example STR32(4).

10.2 ARRAY POINT USE WITH APM BOX GLOBAL VARIABLES

10.2.1 Configuration

When the External Data Option parameter (EXTDATA) on the Array point configuration screen form is set to NONE, the Array point parameters provide indirect access to flag, numeric, string, and time APM box global variables.

Note that an Array point is just a convenient way of referring to related data; the actual data comes from the APM box global variables (in this case). When accessing one of these variables by the Array point tag name, the APM translates the request to a specific APM box global variable.

When accessing the APM box global variables, each Array point allows the following configuration:

	Number of Elements	**Array Point Starting Index**	**Range of Starting Index**	**Range of Array Size**
Flags	1–16,384	FLSTIX	1–16,384	0–1023
Numerics	1–NNUMERIC	NNSTIX	1 to NNUMERIC	0–240
Strings	1–NSTRING	STRSTIX	1 to NSTRING	0–240
Times	1–NTIME	TIMESTIX	1 to NTIME	0–240

Refer to the *APM Parameter Reference Dictionary* for a complete description of the above parameters.

String length—The Array point parameter STRLEN specifies string length as 8, 16, 32, or 64 characters and this determines the format in which strings are presented on the Array point Detail Display. Regardless of the value of STRLEN, Array point string data can still be referenced using the STR8, STR16, STR32, or STR64 formats.

Figure 10-3 shows how longer character string lengths can be fetched from the 8-character format APM box global string variables (provided that the access requested is within the configured section). For example, if the Array point string starting index is set to 301 and adequate APM box string variables are configured (in 8-character format), a request for the 64 character Array point parameter STR64(2) will fetch Box.STR8(309) through Box.STR8(316). If only part of a longer character string can be been fetched, the available portion is fetched. An array index error results when none of the requested string is available.

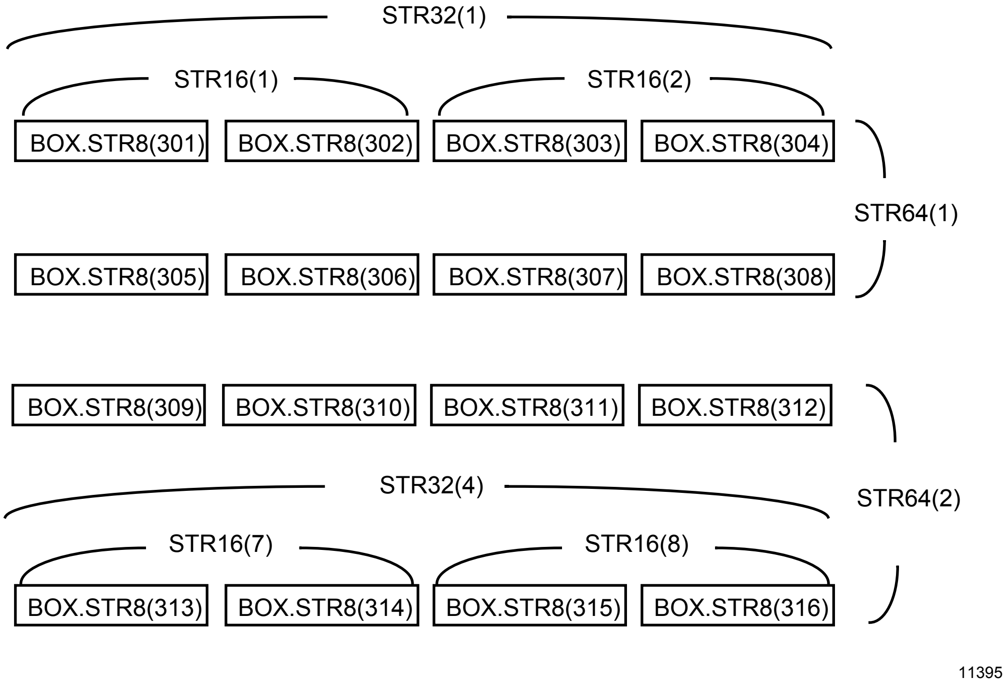


Figure 10-3 — String Array with Starting Index Set to 301

Descriptors—You can enter 4 descriptors up to 64 characters long, 1 for each type of variable. These descriptors are typically used to describe the array variables. They appear on the Array point Detail Display. Refer to the *Process Operations Manual* for examples.

Access Lock—Parameter SPLOCK determines store access to the variables. Configuration choices are Operator, Supervisor, Engineer, or Program.

10.2.2 Array Point Example

Array point ARR100 is configured to represent the ingredients necessary to make product X. The parameters are configured as follows:

Parameter	Meaning	Example Entry
PTDESC	point descriptor	"Product X ingredients"
FLSTIX	flag starting index	0
NFLAG	number of flags in array	0
NNSTIX	numerics starting index	2001
NNUMERIC	number of numerics in array	75
STRLEN	string length	16
STRSTIX	strings starting index	1001
NSTRING	number of strings in array	75
TIMESTIX	times starting index	0
NTIME	number of times in array	0
SPLOCK	set point lock	Operator
NNDESC	numeric descriptor	"Product X Ingredient Amounts"
STRDESC	strings descriptor	"Product X Ingredient Descriptions"

The 75 numerics list the amounts of all possible ingredients. The 75 strings describe each ingredient. The descriptor entries (for NNDESC and STRDESC in this case) subsequently appear on the point Detail Display and describe the ingredients. Refer to the *Process Operations Manual* for examples of the Array point Detail Display.

Additional Array points could be configured to store other attributes of each of the 75 ingredients, such as specific gravities, etc.

After loading the Array point, the parameters AR100.STR16(1 to 75) can be used to refer to the ingredients for Product X. The parameters AR100.NN(1 to 75) can be used to refer to the amount of each ingredient for Product X.

On the Detail Display, array variables appear in a table numbered from 1 through the end of the array. Note that by choosing the starting index to end in 1 in the above examples, the box variables and array variables correspond more conveniently. For example, ARR100.NN(70) is mapped to box variable NN(2070) and appears on the Detail Display as N0070.

10.3 SERIAL INTERFACE TO ARRAY POINTS

When the Array point's EXTDATA parameter is set to IO_FL, IO_NN, or IO_STR (flags, numerics, or strings, respectively), communications is through the Serial Interface (SI) and the point is often referred to as an SI Array point. A maximum of 80 Array points can interface with SI IOP modules depending on the box parameter SCANPER (see subsection 10.1.1).

The Array point accesses external data from the serial device through its Flag, String, and Numeric parameters. For each Array point, only one type of variable (Strings, Flags, or Numerics) can be used for Serial Interface communications. Non-selected parameters still refer to the APM Box variables. For example, if the EXDATA selection is IO_FL, the number and starting index for flags refers to Serial Interface data through the selected FTA, while numeric, string, and times references from this array point are to the APM box variables.

Figure 10-4 illustrates communications between the Array point and the serial IO subsystem. The plug-in FTA module adapts I/O requirements for the specific serial interface.

The SI IOP module has 32 slots available. Up to 16 slots can communicate through FTA number 1, and up to 16 slots can communicate through FTA number 2. An array point automatically communicates through any available slot on the SI IOP connected to the FTA specified during Array point configuration. The Serial Input IOP module is described in Section 2 of this manual and detailed configuration information is provided in the *APM Serial Interface Options* manual.

10.3.1 Accessing SI Array Data

Serial Interface IOP data mapped to an Array point is accessed through the Array point Numeric(NN(i)), Flag (FL(i)), and String (STRn(i)) parameters.

Because the APMM regularly scans data from the Serial Interface, read access to SI Array data does not require an IOL prefetch cycle (that is, CL programs are not delayed while data is being fetched). See also 10.3.5.

10.3.2 SI Array Point Configuration

The following paragraphs describe some of the parameters and considerations that are needed for SI Array point communications. Refer to the *APM Serial Interface Options* manual for a detailed explanation.

10.3.2.1 External Data Parameters

When the Array point is configured to obtain input from a serial interface (EXTDATA = IO_FL, IO_NN, or IO_STR), additional parameters appear on the configuration screen form. These are:

• the SI IOP module number	IOPNUM
• the FTA number, 1 or 2	FTANUM
• the serial link device address	DEVADDR
• FTA scan priority, low or high	SCANPRI

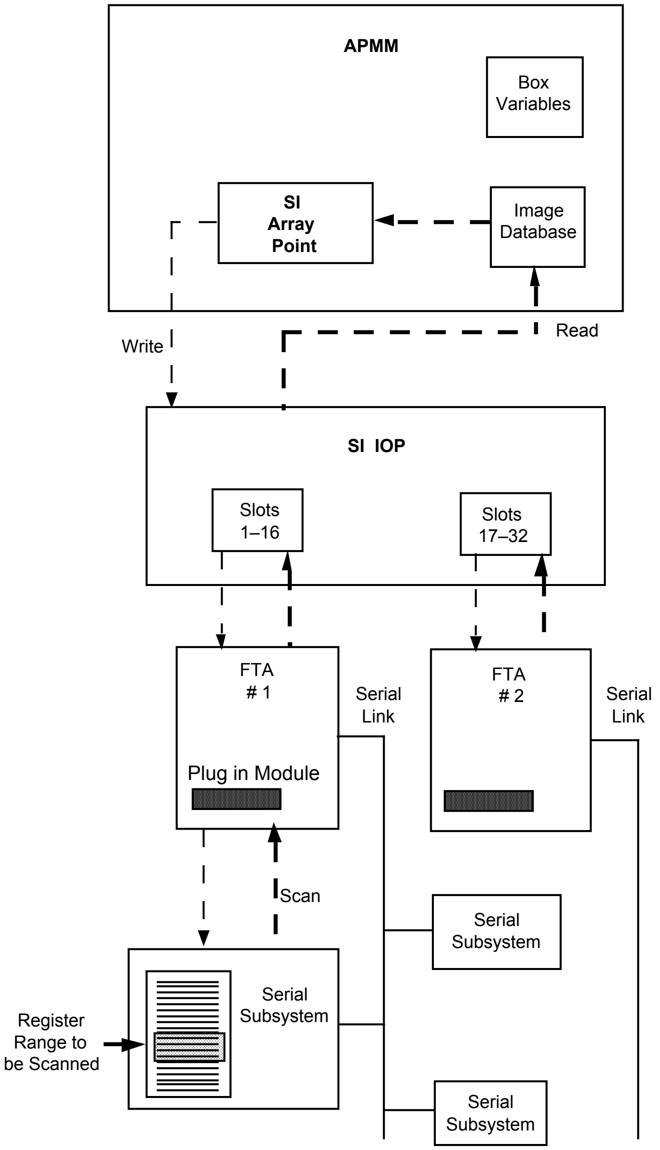


Figure 10-4 — Signal Path Between SI Subsystems and Array Point

11396

10.3.2.2 Modbus and Allen-Bradley Interface Parameters

An SI Array point typically interfaces with an AEG Modicon Generic Controller, a Generic Modbus RTU Controller, or an Allen-Bradley programmable logic controller. Other devices that use the Modbus RTU protocol may communicate to an SI Array point. Your Honeywell representative can provide a list of certified devices.

AUXDATA Parameters—These parameters are for use with the Generic Modbus interface. They should be set to NaN (dashes) when communicating with an Allen-Bradley PLC. Likewise, when the Modbus interface is being used, the AB_DATA parameters should be set to NaN.

Parameter	Specifies
AUXDATA1	Modbus Coil Keep Alive Address
AUXDATA2	Wait time before message retry
AUXDATA3	EIA protocol and modem control
AUXDATA4	Baud rate and Parity

AB_DATA parameters—These parameters are for use with the Allen-Bradley interface. They should be set to NaN (dashes) when communicating with a Modbus. Likewise, when the Allen-Bradley interface is being used, set AUXDATA parameters to NaN.

The following table is only a general overview of the AB_Data parameters. Each is discussed in detail in the APM *Serial Interface Options* manual and there are many variations depending on the PLC family.

Parameter	Specifies
AB_DATA1	The PLC family
AB_DATA2	The PLC file number
AB_DATA3	The Data type
AB_DATA4	The Scan rate

10.3.2.3. Array Type/Size and Starting Index

Most of the remaining SI Array parameters considerations deal with a starting index and array size (number of flags, numerics, or strings).

When used with the Serial Interface, the starting index parameters: FLSTIX (Flag), NNSTIX (Numeric), or STRSTIX (String) refer to a register address in the serial device. This register is scanned over the appropriate range as set by the array size.

The concept of array type, size, and starting index is the same as described for mapping to APM box variables, but for Serial Interface data, the limits are as follows:

Array Type	Array Size	Starting Index Range
Flag	0 -512	0 - 99,999
Numeric (reals)	0 - 16	0 - 99,999
Numeric (integers)	0 - 32	0 - 99,999
Strings	0 -8*	0 - 99,999

*64 characters maximum configured any way from eight 8-character strings to one 64-character string.

10.3.3 Status and Error Checking

Status reporting—Several Array point parameters are provided for status reporting. Error information appears on the Array Point Detail displays and the SI IOP Detail Status display, or can be tested with CL. The following parameters should be checked periodically:

- BADPVFL provides an overall database status (ON = bad). The overall status is bad when there are communication problems or the SI module is in Idle. When the overall status is bad, an attempt to access numeric, flag, or string data results in a bad access status. Consequently, CL programs should check BADPVFL or INITREQ before using SI data.
- INITREQ, when set to ON, indicates that the write to the SI module cannot be completed.
- ERRCODE provides an 8-character error string. Refer to the *Advanced Process Manager Parameter Reference Dictionary* for complete information.

Note that BADPVFL and INITREQ are always OFF if not using SI data (EXTDATA = None).

10.3.4 Read-Back Check for Device control/Digital Composite Points

Device Control or Digital Composite points can have digital output connections through an SI/Array point to a field subsystem. The field device or its interface may interrupt or change the output and not provide any indication of the change.

If flag data is mapped back from the SI/Array point, a digital output read-back check determines the actual value of the output. After a new output state is stored to the digital output connections, the read-back check is delayed for a time period equal to the feedback-time parameter (FBTIME) or 4 seconds, whichever is greater. This delay allows the SI/Array output enough time to reach even a slow responding field device before causing a Command Disagree alarm (see subsections 4.5 and 11.5).
If a discrepancy then exists between OPFINAL and OP, the operator is advised. If OPFINAL does not agree with any of the defined states, its state is displayed as NONE.

10.3.5 CL Notes

All Array point data (flags, numerics, etc.) can be accessed by a CL program (without impacting IOL prefetch limitations). Refer to the *Control Language/Advanced Process Manager Reference* manual for additional information.

The CL/APM compiler issues a warning when external Array point data (data read from a Serial Interface) is used in a LOCAL declaration. This is done because the data is not truly local, but resides in the external device connected to the Serial Interface IOP.

DEVICE CONTROL POINT
Section 11

This section describes the functions available in the device control point. Definitions of the parameters mentioned in this section can be found in the Advanced Process Manager Parameter Reference Dictionary.

11.1 FEATURES

The Device Control (DevCtl) point provides a way to manipulate a device, typically a motor, and a way to view the strategy through a single point. In addition, this point helps the operator to graphically trace the source of an interlock condition.

You can allocate up to 160 Device Control points (NDEVSLOT) or combinations of standard and fast DevCtl (NFASTDEV) points during Node Specific Configuration.

Some of the features of the device control point are as follows:

- Permits a custom logic design (within the strategies provided by the device control point) of an interlocked motor control strategy.
- Allows configuration of a seal-in circuit function.
- Allows configuration of a state change function.
- Accommodates single- and dual-speed single direction motors, reversible single-speed motors and motor-operated valves.

11.2 GENERAL DESCRIPTION

Figure 11-1 is a block diagram of the Device Control Point. It can be thought of as a combination Digital Composite Point, Logic Point, and Regulatory PV Data Acquisition point. The left side of this drawing provides a concept of the input processing capabilities, while the right side of the drawing illustrates the output concept.

The following paragraphs briefly describe the Device Control Point, then each part is explained in detail on the pages that follow.

Digital Composite Section—Two digital inputs, a Local Manual input, and the Digital Output(s) make up the Digital Composite part of the point as shown in the upper part of Figure 11-1. All of the properties of the Digital Composite point described in Section 4 apply to this part of the Device Control point.

Logic Section—The left side of Figure 11-1 shows how up to 12 logic inputs are brought into the point. The gates that follow allow the inputs to be inverted, delayed, compared, etc. The resulting signals can drive other gates, interlocks, permissives, or output commands. Two logic outputs can be configured to output most real or logical data from the point.

Regulatory PV Section—An analog input referred to as the secondary variable (SECVAR) is provided. It is typically used to monitor motor current. You can configure alarms to indicate when this input exceeds specified limits and you can accumulate statistics on levels and durations for this input.

Processing order—The DevCtl point processing order should be regarded as: the regulatory PV section first, then the logic section, and finally the digital composite section.

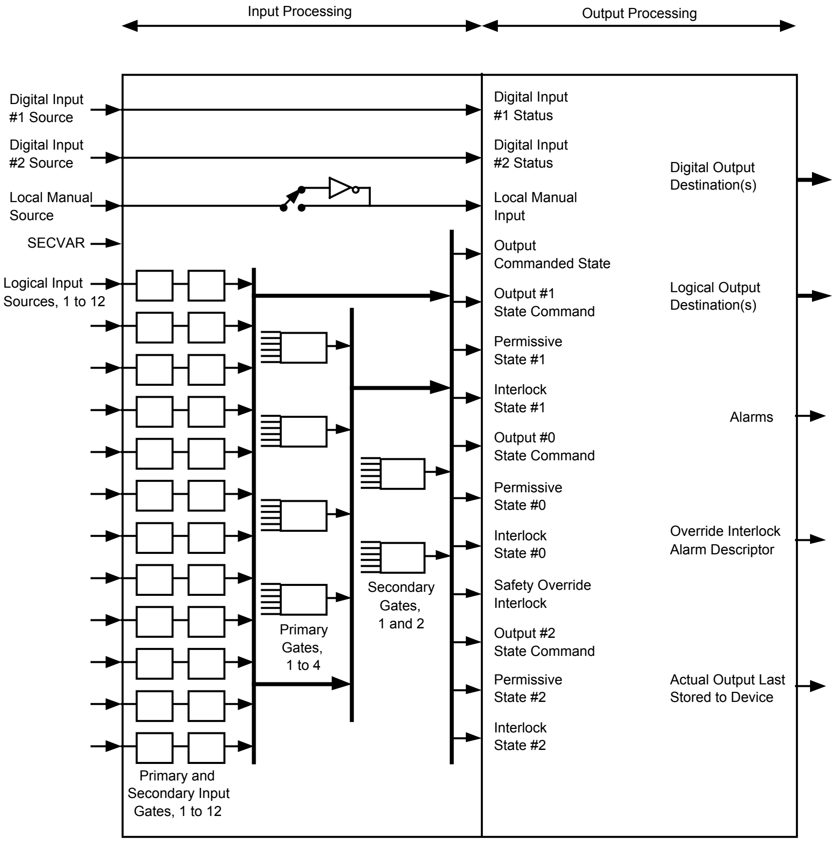


Figure 11-1 — Block Diagram of Device Control Point

11397

11.3 INPUT CONNECTIONS

11.3.1 Digital Inputs

Up to two digital inputs connections can be allocated with NODINPTS (number of Digital Inputs). The input source is defined by DISRC(1) and DISRC(2), which are thereafter referred to as D1 and D2. The input source(s) can be specified as a tag name**.**parameter or a hardware reference address. Allowable sources are—

- Digital Input Points —PV
- Digital Output Points —SO
- Logic Slot gate outputs
- PM Box Flags —PV
- Logic Slot Flags
- ProcMod Flags
- DevCtl Flags
- Array Point Flags

These sources must be in the same APM box as the Device Control point. The PV state is calculated from the inputs the same way as for a Digital Composite Point.

Other configuration for this section such as box color, etc. is the same as for a Digital Composite Point as described in Section 4. You can specify the PV states, options, and alarming.

11.3.2 Logic Inputs

Figure 11-2 presents a simpler way of looking at just the Logic Inputs, their associated gates, how the outputs can be routed, and possible destinations.

Looking at the logic gates in Figure 11-2, you should note the concept of Primary or Secondary *Input* Gates and Primary or Secondary gates. This terminology is used throughout the following discussions and on the point configuration screens.

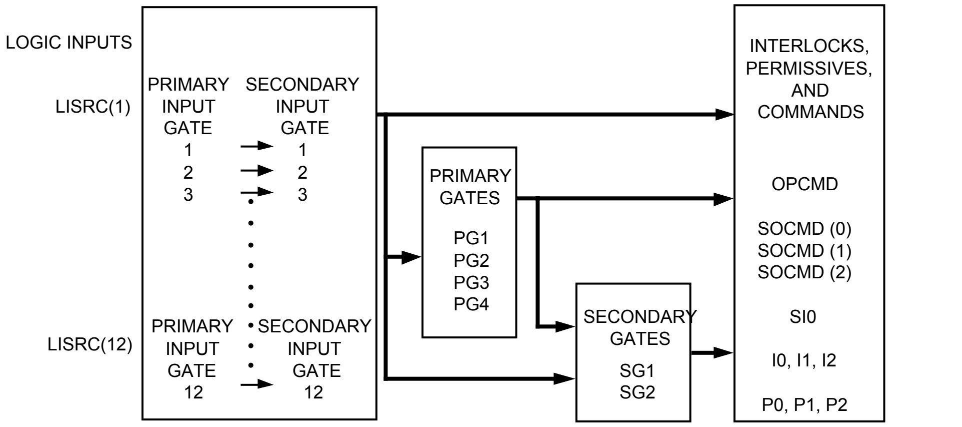


Figure 11-2 — Logic Gates and Destinations

You can designate up to 12 Logical inputs (NOLINPTS) and enter an 8-character ASCII descriptor for each input (LIDESC(n) where n is 1–12). The ASCII descriptor can be changed by a CL program or a person with Engineer access level.

Each input source is specified by Logic input source parameter LISRC(n) and you can use—

- any boolean, integer, enumeration, self-defining enumeration, or real parameter from within the APM, or another UCN node
- the PV flag or BADPVFL for a Digital Input point from within the APM
- the SO or INITREQ parameters for a Digital Output point from within the APM
- any IOP parameter (up to a maximum of six).

Logic inputs (LISRC(1) – LISRC(12) are referenced within the configuration pages as L1 – L12.

The parameter LIBADOPT allows substitution of Off, On, or the last good value (Hold) in case a logic input goes bad.

NOTE

Every logic input that you specify must eventually be used to drive some output, permissive, or interlock.

11.3.2.1 Primary and Secondary Logic Input Gates

Algorithms—Figure 11-3 shows the various choices for the Primary Input Gate algorithm (parameter PIALGID(n)) and the Secondary Input Gate algorithm (parameter SIALGID(n)). The choices are discussed in the following paragraphs.

Primary Input Gate Algorithms—

Null means the input is passed unchanged to the output.

Comparison algorithms for the Primary Input Gate such as GT, GE, etc compare the input to parameter PINN(n) where n is the gate number. PINN(n) contains a real number entry of your choice.

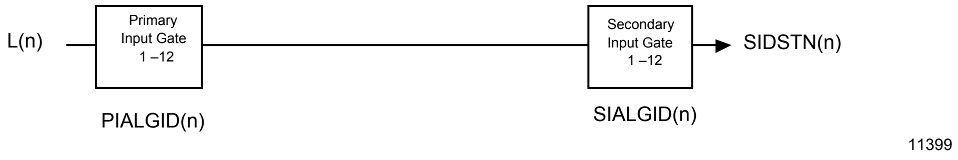
Algorithms ending in 2 such as GT2, GE2, etc., compare the input to a primary input source specified by PISRC(n) where n is the gate number. This allows you to compare the configured input to one of the logic inputs (L1 – L12). Gate(n) output goes true when the comparison succeeds.

In_Set algorithm compares the gate input with a table of ten numerics (NNINSET(n)) and the output goes true if the input equals any value in the set. You must enter the ten numeric values which can range from 0 to 32,767.

Primary Input gates that use comparison algorithms can have a deadband value (PIDEADBD(n) for each gate—

For GT, GT2, GE, GE2, LT, LT2, LE, or LE2 gates, the deadband only applies on a true to false comparison. For example, assume that one of these gates is configured for L1 (input) greater than 50 and the deadband value = 5. The gate output goes true when L1 is greater than 50, but does not go false unless L1 falls below 45.

For EQ, EQ2, NE, and NE2 gates, the deadband value defines the range of comparison. For example, an EQ gate is configured with input numeric PINN(n) = 6 and deadband parameter PIDEADBD(n) =2. The output is true when the input source is between 4 and 8.



(Algorithms)

NULL
INVERT
GT - Greater than PINN(n)
GE - Greater than or equal to PINN(n)
LT - Less than PINN(n)
LE - Less than or equal to PINN(n)
EQ - Equal to PINN(n)
NE - Not equal to PINN(n)
GT2 - Greater than PISRC(n)
GE2 - Greater than or equal PISRC(n)
LT2 - Less than PISRC(n)
LE2- Less than or equal PISRC(n)
EQ2 - Equal to PISRC(n)
NE2 - Not equal to PISRC(n)
IN_SET

(Algorithms)

NULL
DLY - Delay
ONDLY - On Delay
OFFDLY - Off Delay
PULSE
MAXPULSE
MINPULSE

Figure 11-3 — Primary and Secondary Logic Input Gate Algorithms

Secondary Input Gate Algorithms—These include various types of delays. Your entry (0 – 8000 seconds) in parameter SIDLYTIM(n) determines the delay (per gate). These functions are briefly described here and fully in the Logic Point description (Section 5).

Delay—The Delay algorithm causes a one cycle delay of the input value (at the execution rate of the point). Parameter SIDLTYM does not apply to this gate.

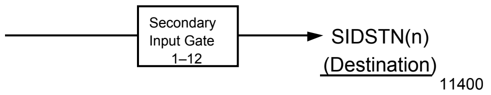
On Delay—On Delay starts counting when the input switches from Off to On. If the input is still On when the time runs out, the output is set to On. When the input signal switches Off, the output is set to Off immediately and the timer (if running) is stopped.

Off Delay —Off Delay starts counting when the input switches from On to Off. If the input is still Off when the timer runs out, the output is set to Off. When the input signal switches On, the output is set to On immediately and the timer (if running) is stopped.

PULSE —Pulse provides a fixed pulse output when the input switches from Off to On. The pulse width is specified by SIDLYTIM(n). Another output pulse cannot be generated until the preceding pulse has completed.

- **MAXPULSE**—MAXPULSE provides a pulse output when the input switches from Off to On. If the input switches Off before the specified time, the output is also set to Off immediately. If the input stays On longer than the timer period, the output pulse shuts Off at the end of the timer period.
- **MINPULSE—**MINPULSE provides a pulse output when the input switches from Off to On. If the input switches Off before the specified timer period, the output is extended until the period is over. If the input stays On longer than the timer period, the output pulse follows the input pulse.

Secondary Input Gate Destinations—the output destination from each Secondary Input Gate is specified by SIDSTN(n). The choices are shown in Figure 11-4.



- None (gate not used)
- Interlocks SI0, I0, I1, or I2
- Permissive Interlocks P0, P1, or P2
- OPCMD, SOCMD0, SOCMD1, or SOCMD2
- Primary Gates PG1, PG2, PG3, or PG4.
- Secondary Gates SG1 or SG2

Figure 11-4 — Secondary Input Gate Destination Choices

NOTE

Every Secondary Input Gate that you configure must have an output destination specified.

11.3.2.2 Primary and Secondary Logic Gates

You can configure up to four Primary Gates (NOPGATE) and up to two Secondary Gates (NOSGATE). Each gate can have up to six inputs. The inputs are determined by specifying them as destinations from other gates. It isn’t necessary to use these gates if you do not need the functions they provide.

Figure 11-5 illustrates the Primary and Secondary Gate algorithm choices.

Primary Gate Inputs—Inputs to the Primary Gates can only come from Secondary Input Gates. Up to six inputs are available.

Secondary Gate Inputs—Inputs to the Secondary Gates can come from any of the 12 Secondary Input Gates or any of the four Primary Gates.

Primary and Secondary Gate Algorithms—both the Primary and Secondary Gates have identical algorithms. Algorithms beginning with a P have a pulsed output when the gate is enabled. Pulse width is specified by parameter PGPLSWTH(n) for Primary Gates or SGPLSWTH(n) for Secondary Gates. Pulse width can range from 0–8000 seconds.

Primary Gate 1–4
PGDSTN(n)
PGALGID(n)
Secondary Gate 1– 2
SGDSTN(n)
SGALGID(n)
11401

(algorithms)

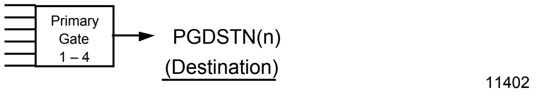
AND	PAND
OR	POR
NAND	PNAND
NOR	PNOR
XOR	PXOR

(algorithms)

AND	PAND
OR	POR
NAND	PNAND
NOR	PNOR
XOR	PXOR

Figure 11-5 — Primary and Secondary Logic Gate Algorithms

Primary Gate Destinations—Parameter PGDSTN(n) directs outputs from Primary Gate n to any one of the destinations shown in Figure 11-6.



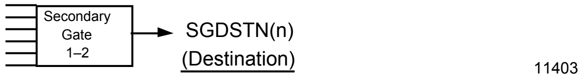
- None (gate is unused)
- Interlocks SI0, I0, I1, or I2
- Permissive Interlocks P0, P1, or P2
- OPCMD, SOCMD0, SOCMD1, or SOCMD2
- Secondary Gates SG1 or SG2

Figure 11-6 — Primary Gate Destinations

NOTE

Every Primary and Secondary Gate that you configure must have an output destination specified.

Secondary Gate Destinations—Parameter SGDSTN(n) directs outputs from Secondary Gate n to any one of the destinations shown in Figure 11-7.



- None (gate is unused)
- Interlocks SI0, I0, I1, or I2
- Permissive Interlocks P0, P1, or P2
- OPCMD
- SOCMD0, SOCMD1, OR SOCMD2

Figure 11-7 — Secondary Gate Destinations

11.3.2.3 Detail Display Presentation

A graphic representation of the logic connections is presented on the point's detail display. Figure 11-8 is typical of the Override Interlock section of this display. The Logic Input descriptors you choose (LIDESC(n)) appear instead of the numbered inputs shown in the example. Heavy lines indicate power flow and allow the operator to determine which input is driving an interlock such as shown in Figure 11-8 where Input _7 is driving the override interlock. Refer to the *Process Operations Manual* for additional information.

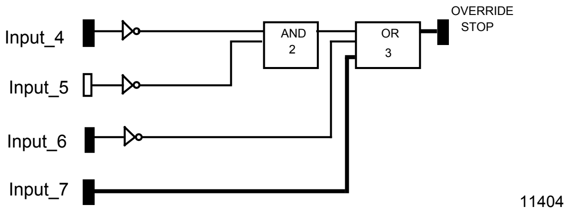


Figure 11-8 — Detail Display of Logic Configuration

11.3.3 Secondary Variable Input

This Regulatory PV section of the DevCtl point provides an analog input for a Secondary Variable (SECVAR). The Secondary Variable input connection is specified by parameter SVSRC as a tag name**.**parameter or hardware address reference. Typically this input is used to monitor motor current, flow rate, valve position, etc.

During configuration you can specify

- descriptors–(SVDESC and SVEUDESC)
- the high/low engineering unit range–(SVEUHI/SVEULO)
- the target value (setpoint)–(SVTV)
- the high and high high trip point–(SVHITP and SVHHTP)

You can also specify the alarm priorities, an alarm deadband, and the secondary variable alarm mask time. Mask time (MASKTIM) specifies the time from 1–1000 seconds that alarms are inhibited after an output change.

Several historical items related to the Secondary Variable can be accumulated and reported on the Maintenance Statistics Display section of the point's Detail Display. These are—

- the peak value of the secondary variable on the last run.
- the duration of the first peak above the full load high trip limit.
- the time that the secondary variable is continuously greater than its high trip limit.

Subsection 11.6 describes the maintenance statistics in more detail.

Figure 11-9 shows how the Secondary Variable appears on the point detail display. The thick bar represents the actual value. The lower vertical dash represents the target value (SVTV). The Secondary Variable high alarm priority trip point in percent (SVHITPP) and high high alarm trip point in percent (SVHHTPP) are represented by the other two small vertical dashes.

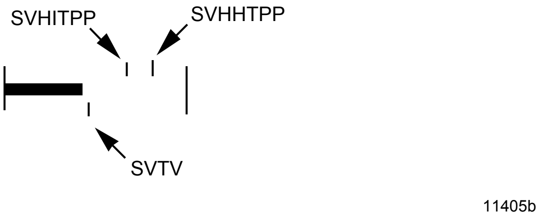


Figure 11-9 — Secondary Variable Representation on the Point Detail Display

The colors of SVHITPP and SVHHTPP are described in Table 11-1.

Table 11-1 — SVHITP and SVHHTP Colors

Network Config. Red Color Alarm Priority	SVHITP Color			SVHHTP Color		
Two Alarm Colors	**SVHIPR =** **Low**	**Hi**	**Emg.**	**SVHHPR =** **Low**	**Hi**	**Emg.**
Low	Red	Red	Red	Red	Red	Red
High	Yel	Red	Red	Yel	Red	Red
Emergency	Yel	Yel	Red	Yel	Yel	Red
Three Selectable Alarm Colors	Depends on color selected in NCF			Depends on color selected in NCF		

When SVHITPP is configured = SVHHTPP, a single vertical dash represents both values. Its color is determined in the same way as for SVHHTPP.

The Red Color Alarm Priority is configured during the Console Data part of Network Configuration.

The bar graph color for the Secondary Variable in percent (SVP) depends on the Secondary Variable High alarm Flag, the High High alarm Flag, and the Secondary Variable High and High High alarm priorities.

If the two-color alarm option is selected in the Network Configuration File (NCF), SVP color is described in Table 11-2.

Table 11-2 — SVP Bar Graph Color

	Condition			Result
1	If SVHIFL and SVHHFL are both FALSE			SVP color is Cyan
2	If SVHIFL is TRUE, then the following table determines when SVP color is RED.			
	Network Configuration Red Color Alarm Priority =	SVHIPR = Low	Hi	Emg.
	Low	Red	Red	Red
	High	N/A	Red	Red
	Emergency	N/A	N/A	Red
3	If SVHHFL is TRUE, then the following table determines when SVP color is RED.			
	Network Configuration Red Color Alarm Priority =	SVHHPR = Low	Hi	Emg.
	Low	Red	Red	Red
	High	N/A	Red	Red
	Emergency	N/A	N/A	Red
4	For all other conditions			SVP color is Yellow

If the three color alarm option was selected in the NCF, the SVP color is based on the color choices in the NCF.

If the three color alarm option was selected in the NCF, the SVP color is based on the color choices in the NCF.

11.3.4 Local Manual Input

Parameter LMSRC allows you to designate a logical input for the local manual signal. The source can be specified as a tag name**.**parameter or as a hardware reference address. Parameter LMREV permits the input to be inverted.

When the LOCALMAN flag is true, it indicates that the device’s state is being controlled directly by external hardware. The Device Control point’s output tracks the PV (the actual state of the device). When the device is taken out of Local/Manual, the output of the point matches the current state of the device being controlled.

11.4 OUTPUT CONNECTIONS

11.4.1 Digital Outputs

The number of digital outputs (up to three) is determined by parameter NODOPTS. State Outputs destinations for STxOPy (where x = 0, 1, 2 and y = 1, 2, 3) are assigned by parameter DODSTN(n) where n = 1–3 (depending on the number of outputs configured).

Acceptable destinations are—

- Digital Output Point–Latched Output
- Logic Slot Flag
- Process Module Slot Flag
- Array Point Flag (see also 10.3.4)
- Digital Output Point–Pulsed Output
- APM Box Flag
- Device Control Slot Flag

Destinations are specified as a tag name**.**parameter or a hardware address and they must be in the same APM as the Device Control point. If a hardware address is used, it must have one of the following forms—

- !BOX.FL(nnnn)   where nnnn = 1 to 16,384
- !DOmmSss.S0

11.4.1.1 Output Commands

There are two ways to command the output:

- when MODATTR = Operator, the operator can command it
- when MODATTR = Program, a Logic point or CL program can command it

OPCMD controls State 0 and State 1. SOCMD(n) controls State 0, State 1, and State 2.

Inputs to OPCMD or SOCMD can come from any of the following logic gates:

- the Primary Gates
- the Secondary Gates
- the Secondary Input Gates

Inputs to OPCMD or SOCMD are made by specifying them as destinations of the above logic gates. Both OPCMD or SOCMD can be configured to drive the output, but you should use only one of these on a given point. When interlock logic does not drive OPCMD or SOCMD, they can be written from outside (for example from a Logic point, a CL Program, etc.)

CAUTION

Do not configure two different logic outputs to drive OPCMD and SOCMD of the same Device Control Point.

The state of OP is calculated from signals generated by the primary and secondary gates through OPCMD or SOCMD, as well as LOCALMAN and REDTAG.

OP can be commanded to state 1 or state 0 using the OPCMD parameter. When OPCMD is commanded ON, OP is set to state 1. When OPCMD is commanded OFF, OP is set to State 0. OPCMD can be used when three states are defined, but will only command OP to state 1 or state 0.

SOCMD(i) provides a command for each state (i = 0, 1, or 2). Unlike OPCMD, the output state is commanded when SOCMD(i) goes from OFF to ON. For example, if SOCMD(1) is commanded ON, State 1 goes true. Then if SOCMD(0) is commanded ON, State 0 is commanded. It does not matter that the logic for SOCMD(1) is still commanding an ON condition. In order to return to State 1, SOCMD(1) must command an OFF, and then an ON condition.

11.4.1.2 Interlocks, Permissives, and Overrides

The ON command is sent to the output only if the permissives and interlocks are in a state to permit the output. If the ON state is allowed, ON becomes the Output Final OPFINAL output. OP and OPFINAL may be different; for example, if the seal in option is activated.

Seal-in —If the seal-in circuit is configured, it will unlatch the digital output contacts on detection of the PV not following the output command as happens on a device drop out (for example, a stopped motor). This condition is true any time the PV is not in state 1 or state 2 and either a command disagree alarm or an uncommanded state change alarm is present.

If the seal in option is enabled when the above condition is detected, the output destinations are set to the state corresponding to OP of state 0, but OP is not altered. The actual state commanded to the output destinations can be observed on the lower part of the group display as OPFINAL. OPFINAL appears in reverse video when not equal to OP.

Figure 11-10 illustrates the interlock system associated with the Device Control Point's digital outputs. This illustration is described thoroughly in Section 4 for the Digital Composite point and is reproduced here for convenience.

Inputs—Inputs to the interlocks can come from the output of —

- any of the Secondary Input Gates
- any of the Primary Gates
- either of the Secondary Gates.

Inputs to the interlocks can also be written to from outside if not driven by any interlock logic (like OPCMD and SOCMD(n). Parameter LOGICSRC allows you to enter the name of an external point (if any) that is controlling interlocks and permissives. The entry then appears on the Detail Display.

Break Before Make Option—When OP is commanded from State 1 to State 2 (for example, from forward to reverse), or from State 2 to State 1, OP will first go to State 0 (for example, stop) and remain there for a configured pause time before going to the commanded state.

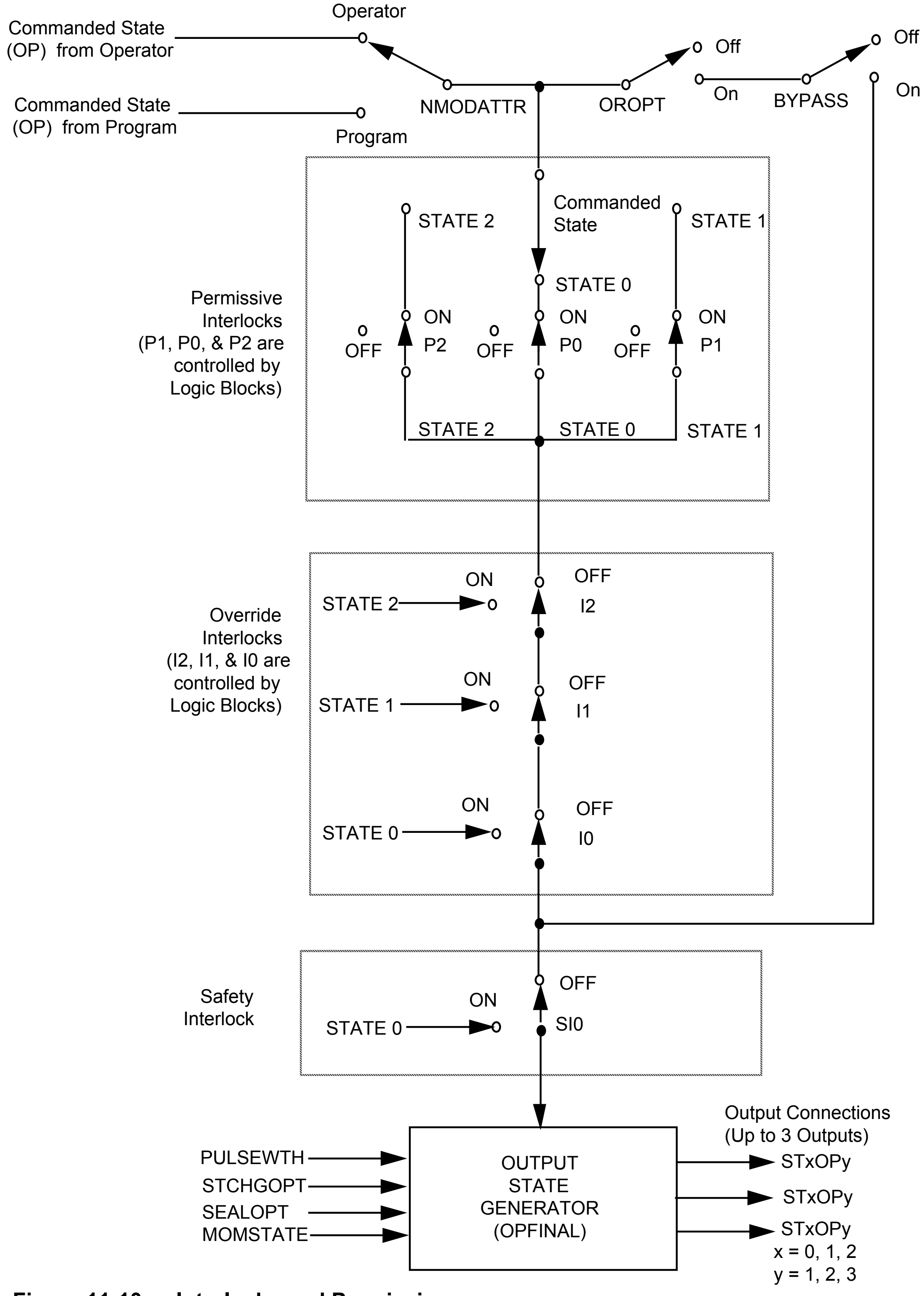


Figure 11-10 — Interlocks and Permissive

11.4.1.3 Digital Outputs on the Detail Display

Figure 11-11 shows how the commanded digital output appears on the detail display for a two state and for a three state output.

The right half of any state box (when filled) indicates the commanded state. The left half of any state box (when filled) indicates the PV has changed to that state.

Two dashes in a state box means that operator commands for that state are not permitted. This can be caused because an interlock is active, when a program is controlling the point, or the permissive for the state is OFF.

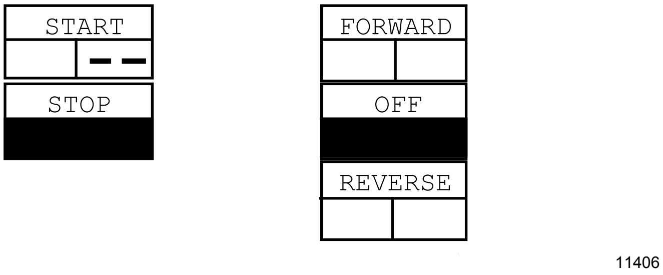


Figure 11-11 — Output Indicators on the Point Detail Display

11.4.2 Logic Output Connections

The Device Control point can have up to two (NOLOPTS) Logic Output connections.

Figure 11-12 illustrates the Logic Output connection strategy. LOSRC is the logic source, LOENBL is the flag that enables the gate, and LODSTN is the output destination.

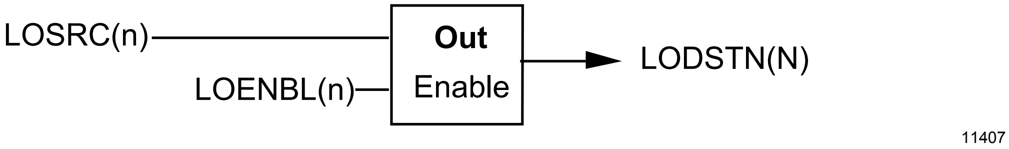


Figure 11-12 — Logic Output Connections

LOSRC(n), the logic source can come from—

L1–L12	the logic inputs
FL1–F12	the local flags (you can set FL7–FL12 Off or On)
NN1–NN8	the local numerics (you can enter the values)
D1 or D2	the digital inputs
P0, P1, or P2	permissive interlocks
SI0, I0, I1, or I2	Safety and other Status Overrides
PISO1–PISO12	Primary Input Gate output value
SISO1–SISO12	Secondary Input Gate status output
PGSO1–PGSO4	Primary Gate status output
SGSO1–SGSO2	Secondary Gate status output
PINN1–PINN12	Primary input comparison numerics
SECVAR	the secondary variable

LOENBL(n), the output enable signal can come from—

L1–L12	the logic inputs
FL1–F12	the local flags
D1 or D2	the digital inputs
P0, P1, or P2	permissive interlocks
SI0, I0, I1, or I2	Safety and other Interlock Overrides
PIS01–PISO12	Primary Input Gate output value
SISO1–SISO12	Secondary Input Gate status output
PGSO1–PGSO4	Primary Gate status output
SGSO1–SGSO2	Secondary Gate status output

LODSTN(n), the destination point**.**parameter to which the output is pushed can be either an ASCII tag name or the hardware address of a point. The destination parameter can be any type parameter (a conversion is made internally to write the correct data type to the destination). If a hardware address is used it must be of the form !BOX, !AOmmSss, or !DOmmSss, where mm is an Analog output or Digital Output IOP module number and ss is the slot number within that module.

11.5 ALARMS

The Device Control Point provides the following alarms:

BADPVFL, Bad process variable alarm
BADSVFL, Bad secondary variable alarm
CMDDISFL, Command disagree alarm
CMDFALFL, Command fail alarm
OFFNRMFL, PV off normal state alarm
OVRDI0FL, OVRDI1FL, OVRDI2FL, Override interlock alarms
OVRDSIFL, Safety Override alarm flag
SVHHFL, Secondary variable high high alarm
SVHIFL, Secondary variable high alarm
UNCMDFL, Uncommanded state alarm

BADPVFL indicates that a digital input has detected a bad PV value.

CMDDISFL, the command disagree flag indicates that the field device did not go to the commanded state within the allowed time.

CMDFALFL, the command fail alarm (also called the crack timer) is generated if the PV doesn't change within the time specified by the command fail timeout parameter (CMDFALTM) after the OP is commanded to a new state. This alarm is useful for motor driven valves with states of OPEN, INBETWEN, and CLOSED. Alarm priority for the command fail alarm is determined by the Command Disagree Priority (CMDDISPR) parameter.

OFFNRMFL indicates that an Off Normal alarm has been detected.

The Secondary Variable high and high high alarms indicate the analog input SECVAR has exceeded the configured limits and BADSVFL indicates the input value is NaN.

UNCMDFL, the uncommanded state alarm indicates that a field device changed state without being command to do so.

The override interlock alarms and the safety override alarm indicate that an override is active. Also refer to the Override descriptor description below.

Override Interlock Alarm Descriptor—When an interlock is tripped, the Device Control point traces the cause of the interrupt and presents the logic descriptor LIDESC(n) for that input on the Detail Display just above the MAN/P-MAN indicator.

11.6 MAINTENANCE STATISTICS

If the Maintenance Support Option is enabled (MAINTOPT = ON), various maintenance statistics are calculated and displayed on the maintenance page of the point's Detail Display. These statistics are available as standard parameters so they can be accessed by CL programs, schematics, etc., and can be saved on a history module.

A program can reset the maintenance statistics at any time by setting the point's reset flag parameter (RESETFL) to ON. The operator can only reset the statistics when the point is in REDTAG.

During configuration, you can enter a maximum time allowed. The parameter MAXTIMnH, n = 0, 1 or 2 sets the maximum time in hours. The maximum number of transitions allowed into each state since the last reset is configured with parameter MAXTRANn (where n = 0, 1 or 2). There is no alarming if the values are exceeded, but CL programs can compare maximum specified values with accumulated values.

The Maintenance Statistics are presented in a plain english format. In addition, the display contains a REDTAG target.

If you select the REDTAG target and press ENTER, the point's REDTAG parameter is set to ON and a RESET target appears. Selecting the RESET target (and pressing ENTER) resets the points maintenance statistic values. If you select REDTAG, (and press ENTER) the REDTAG condition clears and the RESET target disappears.

Index

Topic **Section Heading**

- Accumulation, Digital Input 2.5.3
  - Accumulator Commands 2.5.3.2
  - Overflow 2.5.3.4
  - Target Value 2.5.3.3
  - Up/Down Accumulation 2.5.3.1
- Alarms, General
  - Cut Out 1.3.3
  - Enabling, Disabling, and Inhibiting 1.3.2
  - Priorities 1.3.1
  - Primmod points 1.3.4
  - Regulatory Control Output High/Low Alarms 8.10.7
- Algorithms
  - Auto Manual (AUTOMAN) 8.14.8
  - Calculator (CALCULTR) 7.7.9
  - Data Acquisition (DATACQ) 7.7.1
  - Flow Compensation (FLOWCOMP) 7.7.2
  - General Linearization (GENLIN) 7.7.8
  - High Selector, Low Selector, Average (HILOAVG) 7.7.4
  - Incremental Summer (INCRSUM) 8.14.9
  - Middle of Three Selector (MIDOF3) 7.7.3
  - Override Selector (ORSEL) 8.14.11
  - PID with External Reset -Feedback (PIDERFB) 8.14.3
  - PID with Feedforward (PIDFF) 8.14.2
  - Proportional, Integral, Derivative (PID) 8.14.1
  - Ramp and Soak (RAMPSOAK) 8.14.7
  - Ratio Control (RATIOCTL) 8.14.6
  - Summer (SUMMER) 7.7.5
  - Switch (SWITCH) 8.14.10
  - Totalizer (TOTALIZR) 7.7.7
  - Variable Dead Time with Lead-Lag Com (VDTLDLG) 7.7.6
- Allen-Bradley PLC Interface Through SI Array Point 10.3
  - Report By Exception 10.3.5.5
  - Diagnostic Procedures 10.3.5.7
- Analog Inputs, High Level and Low Level 2.2
  - Alarming 2.2.4
  - Overview 2.1
  - PV Characterization 2.2.1
    - Characterization Options Table 2-1
    - Linear Conversion 2.2.1.1
    - Sensor Types Table 2-1
    - Square-Root Conversion 2.2.1.2
    - Thermal Conversion 2.2.1.3
  - PV Range Checking and Filtering 2.2.2
  - PV Source Selection 2.2.3
- Analog Inputs, Smart Transmitter Interface 2.3
  - Alarming 2.3.2.7
  - Commands 2.3.6
  - Communication Modes 2.3.3
    - Considerations 2.3.4
    - Correcting Database Mismatches 2.3.4
    - Component Point Usage Example Figure 1-2

Index

Topic	Section Heading
Database Considerations, STI	2.3.4
Database Mismatches, STI	2.3.4
Detail Display	Figure 2-2
Functions	2.3.2
Number of Points Per STI Card	1.1.1
Parameter Comparisons	2.3.1
Point States	2.3.5
PV Characterization	2.3.2.1
PV Characterization Options	Table 2-3
Linear	2.3.2.2
Square Root	2.3.2.3
Thermal	2.3.2.4
PV Range Checking and Filtering	2.3.2.5
PV Source Selection	2.3.2.6
Smart Field Communicator	2.3, 2.3.4
Status,	
STI	2.3.5
Transmitter	2.3.4
Transmitter Types Supported	2.3
Use of Damping vs. TF	2.3.2.5
Valid Damping Values	2.3.2.5
Analog Outputs	2.4
Calibration Compensation	2.4.3
Direct/Reverse Output	2.4.1
Output Characterization	2.4.2
Associated Display	1.2.2
Array Points	10
Node Specific Configuration Requirements	10.1.1
Array Parameter Names	10.1.2
Array Point Use with APM Box Global Variables	10.2
Array Point Example	10.2.2
Serial Interface to Array Point	10.3
Status and Error Checking	10.3.3
Modbus and Allen-Bradley PLC Interface parameters	10.3.2.2
Read-Back Checks	10.3.4
CL Notes	10.3.5
Auto Manual Station — *see also Manual/Auto Station*	
Auto Manual (AUTOMAN) Control Algorithm	8.14.8
Bumpless Returns to Cascade Operation	8.14.8.3.1
Engineering Unit Ranges	8.14.8.3.4
Equations	8.14.8.4
Error Handling	8.14.8.3.6
Function	8.14.8.1
Functional Diagram	Figure 8-23
Initialization	8.14.8.5
Modes, Operating	8.14.8.3.2
Options and Special Features	8.14.8.3
Override Feedback Processing	8.14.8.6
Diagram of	Figure 8-25
Restart or Point Activation	8.14.8.3.5
Use	8.14.8.2
Auxilliary Units	1.3.5
Bad PV/Mode Shed	8.4.4.4
Bias Option	8.7.6

Index

Topic	Section Heading
Box, APM	
Alarm Enable Status	1.3.2
Alarm Priorities	1.3.1
Alarming	1.3
AO Data Point	2.4
Box Flag Data Point	9.2
Box Numeric Data Point	9.3
Box Timer Data Point	9.4
Common Functions	1.2
Control Functions	1.1.2, Section 3
Control Performance	3.1
Database Considerations	3.2
Determining Processing Capacity	3.1.3
DI Data Point	2.5
Dig Comp Data Point	Section 4
DO Data Point	2.6
Flag Data Point	9.2
Global Variables	9.1
Hardware Reference Addressing	3.5
HLAI Data Point	2.2
I/O Functions	1.1.1, Section 2
LLAI Data Point	2.2
LLMUX Data Point	2.2
Logic Slot	Section 5
Numeric, Box Data Point	9.3
Peer-to-Peer Communication	3.4
Point Execution State	3.3
Process Module Data Point	6.1
Pulse Input Data Point	2.7
Red Tagging	1.4
RegCtl Data Point	Section 8
RegPV Data Point	Section 7
RHMUX Data Point	2.2
Scan Frequency	3.1.2
Scan Rate	3.1.2.1
STI Data Point	2.3
Time Variables	9.5
Timer, Box Data Point	9.4
Calculator Algorithm	7.7.9
Arithmetic Functions	7.7.9.1
Assignment Statement	7.7.9.1
Equation	7.7.9.4
Error Handing of Bad-Inputs and Uncertain Values	7.7.9.3
Function	7.7.9.1
Functional Diagram	Figure 7-13
Operators	7.7.9.1
Options and Special Features	7.7.9.3
Separator	7.7.9.1
Use	7.7.9.1

Index

Topic — **Section Heading**

- CALCULTR 7.7.9
- Changing Dead Time (Delay-Time) Parameters *(see also Variable Dead Time with Lead Lag)* 7.7.6.3
- Characterization (HLAI, LLAI, LLMUX, and RHMUX)
  - Input (PV) 2.2.1
    - Linear Conversion 2.2.1.1
    - Range Checking and Filtering 2.2.2
    - Source Selection 2.2.3
    - Square Root 2.2.1.2
    - Thermal (RTD and Thermocouple) 2.2.1.3
- Characterization (STI)
    - Input (PV) 2.3.2.1
    - Linear Conversion 2.3.2.2
    - Range Checking and Filtering 2.3.2.5
    - Source Selection 2.3.2.6
    - Square Root 2.3.2.3
    - Thermal (RTD and Thermocouple) 2.3.2.4
- CL/APM Sequence Program 6.1.1
- Cold/Warm Restart 3.3.2
- Command Disagree Alarm 4.5, 4.14, 11.4.1, 11.5
- Communications, Peer-to-Peer 3.4
- Connections
  - Hardware Reference Addressing 3.5
  - Input Connections
    - To Digital Composite 4.11
      - From Digital Input PV 4.11.1
      - From Logic Slot Output 4.11.2
      - From Logic Slot Flag 4.11.3
      - From APM Box PV Flag 4.11.4
    - To Logic Slot 5.2.1
    - To Regulatory PV Data Point 7.2
    - To Regulatory Control Point 8.2
  - Output Connections
    - From Digital Composite
      - Latched 4.7.1
      - Pulsed 4.7.2
      - To Logic Slot Flag 4.7.3
      - To APM Box PV Flag 4.7.4
    - From Logic Slot 5.2.5
    - From Regulatory Control Slot 8.9.6
- Contact Cutout 1.3.3
- Control Algorithms 8.14
- Control Functions — *see Control Slots*
- Control Language — *See "Process Module Data Point"*
- Control Slots
  - Connections — *see "Connections"*
  - Determining Processing Capacity 3.1.3
  - Digital Composite 4.1
  - Functions 1.1.2
  - Logic Slot 5.1
  - Overruns 3.1.3.1

Index

Topic	Section Heading
Overview	3.1
Peer-to-Peer Communication	3.4
Point Form	1.2.1
Regulatory Control	8.1
Regulatory PV	7.1
Scan Rates	3.1.2
Types	3.1.1
Data Acquisition Algorithm (DATACQ) PV Algorithm	7.7.1
Function	7.7.1.1
Functional Diagram of	Figure 7-2
Equations (forms of)	7.7.1.4
Options and Special Features	7.7.1.3
Use	7.7.1.2
DATACQ	7.7.1
Dead-Time (Delay-Time) Calculation *(see also Variable Dead Time with Lead Lag)*	7.7.6.3
DEVICE CONTROL POINT	11, 3.1.1
Input Connections	11.3
Digital Inputs	11.3.1
Logic Inputs	11.3.2
Secondary Variable Input	11.3.3
Local Manual Input	11.3.4
Output Connections	11.4
Digital Outputs	11.4.1
Logic Output Connections	11.4.2
Read-Back Check	10.3.4
Alarms	11.5
Maintenance Statistics	11.6
Device Reservation	3.7
Digital Composite — *see Section 4*	
Alarms	4.14
Command Disagree	4.14.1
Off-Normal	4.14.2
Uncommanded Change	4.14.1
Array Point Flags	4.7.6, 4.11.6
Change-Of-State Events	4.15
Commanded State	4.5
Connections	
Input	4.11
Output	4.7
Configuring the Input State	4.12
Device Control Point Flags	4.7.5, 4.11.5
Initialization Manual	4.9
Inputs	4.10
Interlocks	4.4
Latched Output	4.6.1
Local Manual Indication	4.16
Maintenance Statistics	4.17
Reset/Redtag	4.17.1
Mode Attributes	4.3
Momentary States	4.2.2

Index

Topic **Section Heading**

  - Moving/Bad States 4.2.3
  - Normal States 4.2.1
  - Outputs 4.6
  - Override Interlocks 4.4.2
  - Permissive Interlocks 4.4.1
  - Pulsed Output 4.6.2
  - PV Source 4.13
  - Read-Back Check 4.8
  - States 4.2
- Digital Input
  - Accumulation Digital Input 2.5.3
    - Accumulator Commands 2.5.3.2
    - Overflow 2.5.3.4
    - Target Value 2.5.3.3
    - Up/Down Accumulation 2.5.3.1
  - Latched Digital Input Point 2.5.2
    - Event Reporting 2.5.2.1
  - Status Digital Input 2.5.1
    - Event Reporting 2.5.1.3
    - Off-Normal Alarming 2.5.1.2
    - PV Source Selection 2.5.1.1
- Digital Output 2.6
  - Pulse Width Modulated Output Type 2.6.1
  - Status Output Type 2.6.2
- External Mode Switching 8.6
- Fast Slots 3.1.2.2
- Field Inputs
  - *see "Inputs, Field" or "Analog Input" or "Digital Input"*
- Field Outputs
  - *see "Outputs, Field" or "Analog Output " or "Digital Output"*
- Fixed Delay Time *(see also Variable Dead Time with Lead Lag)* 7.7.6.3
- Flags
  - Logic Point 5.2.2
  - APM Box 9.2
    - Alarming 9.2.1
  - Process Module 6.1.3.2
- FLOWCOMP 7.7.2
  - Flow Compensation PV Algorithm (FLOWCOMP) 7.7.2
  - Compensating for Assumed Design Conditions 7.7.2.3
  - Equations (forms of) 7.7.2.4
  - Error Handling 7.7.2.3
  - Function 7.7.2.1
  - Options and Special Features 7.7.2.3
  - Parameters 7.7.2.4
  - Restart or Point Activation 7.7.2.3
  - Special Notes 7.7.2.3
  - Use 7.7.2.2
- General Linearization PV Algorithm (GENLIN) 7.7.8
  - Changing Parameters through a Universal Station 7.7.8.3
  - Equation 7.7.8.4
  - Error Handling 7.7.8.3

Index

Topic — **Section Heading**

  - Example of Operation — Figure 7-12
  - Extension of First and Last Segments — 7.7.8.3
  - Function — 7.7.8.1
  - Options and Special Features — 7.7.8.3
  - Parameter — Value Restrictions — 7.7.8.3
  - Restart or Point Activation — 7.7.8.3
  - Use — 7.7.8.2
- Generic Modbus Interface to Array Point — 10.3
- GENLIN — 7.7.8
- Global Variables — 9.1
- Hardware Reference Addressing — 3.5
- High Selector, Low Selector, Average PV Algorithm (HILOAVG) — 7.7.4
  - Equations — 7.7.4.4
  - Error Handling — 7.7.4.3
  - Forced Selection — 7.7.4.3
  - Function — 7.7.4.1
  - Functional Diagram — Figure 7-5
  - Options and Special Features — 7.7.4.3
  - Restart or Point Activation — 7.7.4.3
  - Use — 7.7.4.2
- HILOAVG — 7.7.4
- Idle-Run Transition — 3.3.1
- Inputs, Field
  - Analog
    - High Level — 2.2
    - Low Level — 2.2
  - Digital — 2.5
    - Accumulation — 2.5.3
    - Latched — 2.5.2
    - Status — 2.5.1
  - Smart Transmitter Interface — 2.3
  - Incremental Summer (INCRSUM) Control Algorithm — 8.14.9
  - Bad Control Handling — 8.14.9.3.9
  - Changes to Output by User-Written Programs — 8.14.9.3.4
  - Control Input Connections — 8.14.9.3.1
  - Engineering Unit Ranges — 8.14.9.3.3
  - Equations — 8.14.9.4
  - Full Value, Floating PID Outputs, handling of — 8.14.9.3.2
  - Function — 8.14.9.1
  - Functional Diagram — Figure 8-26
  - Initialization — 8.14.9.5
  - Modes, Operating — 8.14.9.3.6
  - Options and Special Features — 8.14.9.3
  - Override Control Strategy — 8.14.9.3.5
  - Remote Cascade Options — 8.14.9.3.7
  - Restart or Point Activation — 8.14.9.3.8
  - Use — 8.14.9.2

Index

Topic **Section Heading**

- Interlocks 4.4
  - Overrides 4.4.2
  - Permissives 4.4.1
  - Types, 4.4
- I/O Functions — *see "Inputs, Field" or "Outputs, Field"*
- I/O Points — *see "Inputs, Field"or "Outputs, Field*"
- LLMUX Data Point 2.2
- Logic Point
  - Algorithms 5.3
    - AND Gate 5.3.2
    - Check for Bad 5.3.17
    - Compare Equal with Deadband 5.3.11
    - Compare Greater Than or Equal with Deadband 5.3.14
    - Compare Greater Than with Deadband 5.3.13
    - Compare Less Than or Equal with Deadband 5.3.16
    - Compare Less Than with Deadband 5.3.15
    - Compare Not Equal with Deadband 5.3.12
    - Delay 5.3.21
    - Fixed Size Pulse 5.3.18
    - Flipflop 5.3.25
    - NAND Gate 5.3.5
    - NOR Gate 5.3.6
    - NOT Gate 5.3.4
    - Null 5.3.1
    - Off Delay 5.3.23
    - On Delay 5.3.22
    - OR Gate 5.3.3
    - Pulse with Maximum Time Limit 5.3.19
    - Pulse with Minimum Time Limit 5.3.20
    - Qualified OR Gate with 2 Inputs On 5.3.8
    - Qualified OR Gate with 3 Inputs On 5.3.9
    - Switch 5.3.10
    - Watchdog Timer 5.3.24
    - XOR Gate 5.3.7
  - Configuring the Logic Point 5.2
  - Connections
    - Input 5.2.1
    - Communication and Configuration Errors 5.2.1.1
    - Output 5.2.5
  - Descriptors, Generic 5.2.6
  - Logic Mix 5.2
  - Flags 5.2.2
  - Numerics 5.2.3
- Low-Level Multiplexer Data Point 2.2
- Manual/Auto Station, SDI Interface 2.9.2
- Memory Units (MU) 3.1.3
- Message Text, see Status Messages
- Middle of Three Selector PV Algorithm (MIDOF3) 7.7.3
  - Equations (forms of) 7.7.3.4
  - Error Handling 7.7.3.3
  - Function 7.7.3.1

Index

Topic	Section Heading
Functional Diagram	Figure 7-4
Normal Operation with 3 Valid Inputs	7.7.3.3
Options and Special Features	7.7.3.3
Use	7.7.3.2
MIDOF3	7.7.3
Modbus Interface Through SI Array Point	10.3
Motor Control Point, see Device Control Point	10.3.4.2NIM
Maximum number of points	1.2.1
Numerics	
Logic Point	5.2.3
APM Box	9.3
Process Module Data Point	6.1
Out of Service, Points	1.4
Output/Setpoint (SP/OP) Tolerance Check	8.8
Outputs, Analog	
Calibration Compensation	2.4.3
Direct/Reverse	2.4.1
Output Characterization	2.4.2
Outputs, Digital	
Pulse Width Modulated (PWM)	2.6.1
Status	2.6.2
Outputs, Digital Composite	
Commanded State	4.6
Latched	4.6.1
Pulsed	4.6.2
Override Selector (ORSEL) Control Algorithm	8.14.11
Override and Bypass Options	8.14.11.3.2
Control Input Connections	8.14.11.3.1
Engineering Unit Ranges	8.14.11.3.3
Equations	8.14.11.4
Error Handling	8.14.11.3.7
Function	8.14.11.1
Functional Diagram	Figure 8-30
Example of an Override Control Strategy	Figure 8-31
Initialization	8.14.11.5
Modes, Operating	8.14.11.3.4
Options and Special Features	8.14.11.3
Override Control	8.13
Override-Feedback Initiation	8.14.11.6.1
Override Offset	8.14.11.3.2
Remote Cascade Options	8.14.11.3.5
Restart or Point Activation	8.14.11.3.6
Use	8.14.11.2
Peer-to-Peer Communications	3.4

Index

Topic **Section Heading**

PID Control Algorithm — *see Proportional, Integral, Derivative*
PID with External Reset-Feedback (PIDERFB) Algorithm 8.14.3
  Engineering Unit Ranges 8.14.3.3.3
  Equations 8.14.3.4
  Error Handling, RFB and TRFB Inputs 8.14.3.3.1
  Function 8.14.3.1
  Functional Diagram Figure 8-8
  Hand Auto Stations
    Configuration and Wiring Diagram Figure 8-10
    Logic Point Configuration Figure 8-11
    Requirements for Interface 8.14.3.2.2
  Hand Auto Stations, use of 8.14.3.2.1
  Initialization 8.14.3.5
  Initialization and Windup Protection (Note) 8.14.3.3.2
  Operating Modes 8.14.3.3.4
  Output Connections 8.14.3.3.2
  Override Feedback Processing 8.14.3.6
  Remote Cascade Options 8.14.3.3.5
  Uses 8.14.3.2
PID with Feedforward (PIDFF) Control Algorithm 8.14.2
  Add or Multiply Action 8.14.2.3.1
  Bypassing Feedforward Control Action 8.14.2.3.2
  Equations 8.14.2.4
  Example, in Feed Heater Control Figure 8-7
  Feedforward Signal Value Status 8.14.2.3.3
  Function 8.14.2.1
  Functional Diagram Figure 8-6
  Initialization 8.14.2.5
  Modes, Operating 8.14.2.3.4
  Override Feedback Processing 8.14.2.6
  Options and Special Features 8.14.2.3
  Remote Cascade Options 8.14.2.3.5
  Use 8.14.2.2
PID Position Proportional Controller (PIDPOSPROP) Algorithm 8.14.5
  Analog Output Related Parameters 8.14.5.5
  Bad PV/Mode Shed 8.14.5.7
  Equations 8.14.5.4
  Functional Diagram Figure 8-14
  Options and Special Features 8.14.5.3
  Safety Shutdown 8.14.5.6
  Use 8.14.5.2
Point Reservation 3.7
Point Types
  Analog Input, High Level and Low Level 2.2
  Analog Input, Smart Transmitter Interface 2.3
  Array Points 10.1
  Digital Input 2.5
  Digital Inputs for Sequence Of Events 2.9
  Digital Output 2.6
  Digital Composite 4.1

Index

Topic — **Section Heading**

  - Flag, Box 9.2
  - Full/Component Points 1.2.1
  - Logic Point 5.1
  - Numeric, Box 9.3
  - Primary Module (PRIMMOD) point 1.3.4
  - Process Module 6.1
  - Pulse Input 2.7
  - Regulatory Control 8.1
  - Regulatory PV 7.1
  - Sequence Of Events (see Digital Inputs)
  - Smart Transmitter Interface 2.3
  - Strings, Box 9.6
  - Timer, Box 9.4.4
  - Times, Box 9.5
- Points, Maximum Number of 1.2.1
- Position Proportional Controller (POSPROP) Algorithm 8.14.4
  - Analog Output Related Parameters 8.14.4.3.9
  - Engineering Unit Ranges 8.14.4.3.8
  - Equations 8.14.4.4
  - Example of in Step Valve Control Strategy Figure 8-13
  - Function 8.14.4.1
  - Initialization 8.14.4.5
  - Modes 8.14.4.3.1
  - One-Shot Control Initialization (Note) 8.14.4.5
  - Options and Special Features 8.14.4.3
  - Output Manipulation in Manual Mode 8.14.4.3.3
  - PV Source Selection 8.14.4.3.7
  - Raise/Lower Output Destinations 8.14.4.3.5
  - Remote Cascade Options 8.14.4.3.2
  - Setpoint Options 8.14.4.3.6
  - Use 8.14.4.2
  - Windup Feedback Limit Switches 8.14.4.3.4
- Primary Module (PRIMMOD) Point 1.3.4
- Processing Units (PUs) 3.1.3
- Process Module Data Point 6.1
  - CL/APM Sequence Program 6.1.1
  - Local Flags, Numerics, Times, and Strings 6.1.1, 6.1.3.2
  - Process Module Operating State 6.1.6
  - Restart Option 6.1.12
  - Sequence Alarms 6.1.10
  - Sequence Execution 6.1.5
  - Sequence Execution Mode 6.1.7
  - Sequence Execution State 6.1.8
  - Sequence Library 6.1.3.1
  - Sequence Messages 6.1.11
  - Sequence Overrides 6.1.9
  - Sequence Program/Unit Partitioning 6.1.2
  - Sequence Program Structure 6.1.4
  - Writing the Sequence Programs 6.1.3

Index

Topic	Section Heading
Proportional, Integral, Derivative (PID) Control Algorithm	8.14.1
Bias Options	8.14.1.3.12
Control by a Single Term	8.14.1.3.4
Control Output Connections	8.14.1.7
Direct and Reverse Control Action	8.14.1.3.5
Engineering Unit Ranges	8.14.1.3.2
Equations	8.14.1.4
Error Handling	8.14.1.3.16
Four Combinations of Control Terms	8.14.1.3.3
Function	8.14.1.1
Gain Options	8.14.1.3.7
Initialization	8.14.1.5
Initializing PID Output without Affecting Dynamics	8.14.1.3.10
Interactive and Noninteractive PID Forms	8.14.1.3.1
Operating Modes	8.14.1.3.13
Options and Special Features	8.14.1.3
Override Feedback Processing	8.14.1.6
Parameters	8.14.1.7
PV Tracking	8.14.1.3.6
Remote Cascade Options	8.14.1.3.14
Restart or Point Activation	8.14.1.3.15
Restrictions on Some Values	8.14.1.3.11
Suppression of Output "Kicks" When Switching to CAS Mode	8.14.1.3.9
Use	8.14.1.2
Windup Handling	8.14.1.3.8
Pulse Input Point	2.7
PV Algorithms	7.7
Ramp and Soak (RAMPSOAK) Control Algorithm	8.14.7
Changing Remaining Soak Time and Current Segment	8.14.7.3.4
Control Input Connections	8.14.7.3.11
Engineering Unit Ranges	8.14.7.3.9
Equations	8.14.7.4
Function	8.14.7.1
Functional Diagram	Figure 8-18
Guaranteed Ramp Rate	8.14.7.3.6
Guaranteed Soak Time	8.14.7.3.5
Limits of Deviation, Output and Setpoint	8.14.7.3.9
Longer Sequences, Achieving by Interconnecting	8.14.7.3.8
Use of Logic Point in Longer Sequence	Figure 8-22
Mark Timer Functions	8.14.7.3.7
Pictured	Figure 8-21
Operational Modes	8.14.7.3.2
Options and Special Features	8.14.7.3
Override Feedback Processing	8.14.7.5
Parameters (for RAMPSOAK0)	8.14.7.6
Ramping Waits for Process	Figure 8-20
Remote Cascade Options	8.14.7.3.3

Index

Topic	Section Heading
Restart or Point Activation	8.14.7.3.10
Single or Cyclic Sequencing	8.14.7.3.1
Use	8.14.7.2
Wait Time between Phases	Figure 8-19
Ratio Control (RATIOCTL) Control Algorithm	8.14.6
Calculator PV Algorithm, Role of	8.14.6.3.3
Control Input Connections	8.14.6.3.1
Engineering Unit Ranges	8.14.6.3.2
Equations	8.14.6.4
Error Handling	8.14.6.3.7
Example of	Figure 8-17
Function	8.14.6.1
Functional Diagram	Figure 8-15
Initialization	8.14.6.5
Modes, Operating	8.14.6.3.4
Options and Special Features	8.14.6.3
Override Feedback Processing	8.14.6.6
Functional Diagram of	Figure 8-17
Remote Cascade Options	8.14.6.3.5
Restart or Point Activation	8.14.6.3.6
Use	8.14.6.2
Read-Back Check, DevCtl or DigCmp points	4.8, 10.3.4
Red Tagging	1.4
Redundancy	3.6
Regulatory Control Data Point	
Alarming	
Advisory Deviation Alarming	8.7.4
Bad Output Alarm	8.7.5
Detection	8.9
Deviation Alarming	8.7.4
Output High/Low Alarms	8.10.7
Bias Option	8.7.6
Connections	
Input	8.2
Output	8.10.6
Control Output Processing	8.10
Initialization	8.10
PV	8.11.1
Control	8.11.2
Limits	
Bias Limits	8.7.6.1
Output High and Low	8.10.2
Output Rate-of-Change	8.10.3
Output Minimum Change	8.10.4
Output Limiting in Manual Mode	8.10.5
Ratio Limits	8.7.6.2
Setpoint	8.7.1
Modes	8.4
External Mode Switching	8.6
Mode Attribute	8.4.1
Normal Mode Attribute	8.4.3

Index

Topic **Section Heading**

Normal Mode 8.4.2
Remote Cascade Handling 8.4.4
Remote Cascade Options 8.4.4.1
Remote Cascade Shed 8.4.4.3
Remote Cascade Request 8.4.4.2
Override Control 8.13
Guideline for use of 8.13.3
Processing 8.13.2
Status and Feedback 8.13.1
PV
Source Selection 8.3
Tracking 8.7.3
Ratio Option 8.7.6
Setpoint
Access Restrictions 8.7.7
Handling 8.7
Limits 8.7.1
Target Value Processing 8.7.2
Shutdown, Safety 8.5
Windup
Protection 8.12
Status 8.12.1
Status Propagation 8.12.2
Regulatory PV Data Point
Input Connections 7.2
PV
Alarm Detection 7.6
Range Check and Filtering 7.4
Source Selection 7.5
Value Status 7.4.1
Restart, Warm Cold 3.9
Restrictions on Delay Time *(see also Variable Dead Time with Lead Lag)* 7.7.6.3
Reservation, Point 3.7
RHMuX Data Point 2.2
SDI *see Serial Device Interface*
Sensor Types, Field Table 2-1
Sequence Of Events Digital Input Point 2.5.4
Sequence Program, CL/APM 6.1.1
Alarms 6.1.10
Execution 6.1.5
Execution Mode 6.1.7
Execution State 6.1.8
Library 6.1.3.1
Messages 6.1.11
Overrides 6.1.9
Process Module Data Point 6.1
Program Structure 6.1.4
Restart Option 6.1.12
Unit Partitioning 6.1.2
Writing the 6.1.3

Index

Topic **Section Heading**

- Serial Device Interface 2.9
  - Description 2.9.1
  - Manual/Auto Station Interface 2.9.2
  - Toledo Weigh Scale Interface 2.9.3
- Serial Interface 2.8
  - Operation 2.8.1
  - Configuration 2.8.2
  - Checkpointing and Startup 2.8.3
  - Serial Interface to Array Points 10.3
    - Configuration 10.3
    - Generic Modbus Interface 10.3.2.2
    - Allen-Bradley PLC Interface 10.3.2.2
  - SI Array Point 10.3
- Setpoint/Output (SP/OP) Tolerance Check 8.8
- Smart Transmitter Interface Point 2.3
  - Alarming 2.3.2.7
  - Commands 2.3.6
  - Communication Modes 2.3.3
  - Component Point Usage Example Figure 1-2
  - Database
    - Considerations 2.3.4
    - Mismatches 2.3.4
  - Detail Display Figure 2-2
  - Functions 2.3.2
  - Linear Conversion 2.3.2.2
  - Number of Points per STI Card 1.1.1
  - Point States 2.3.5
  - PV Characterization 2.3.2.1
  - PV Characterization Options Table 2-3
  - PV Range Checking and Filtering 2.3.2.5
  - PV Source Selection 2.3.2.6
  - Smart Field Communicator 2.3, 2.3.4
  - Square Root Conversion 2.3.2.3
  - Status
    - STI 2.3.5
    - Transmitter 2.3.4
  - Thermal Conversion 2.3.2.4
  - Transmitter/STI IOP Parameter Comparisons 2.3.1
  - Transmitter Types Supported 2.3
  - Use of Damping vs. TF 2.3.2.5
  - Valid Damping Values 2.3.2.5
- States, Digital
  - Current Input (PV) 4.12
  - Normal 4.2.1
  - Momentary 4.2.2
  - Moving/Bad 4.2.3
- Status Messages 3.8
- Strings, Box 9.6

Index

Topic	Section Heading
SUMMER	7.7.5
Summer Algorithm (SUMMER) PV Algorithm	7.7.5
Ensuring Adequate PV Range	7.7.5.3
Equations	7.7.5.4
Error Handling	7.7.5.3
Function	7.7.5.1
Functional Diagram	Figure 7-6
Options and Special Features	7.7.5.3
Restart or Point Activation	7.7.5.3
Use7.7.5.2	
Switch (SWITCH) Control Algorithm	8.14.10
Control Input Connections	8.14.10.3.1
Engineering Unit Ranges	8.14.10.3.2
Equations	8.14.10.4
Error Handling	8.14.10.3.9
Examples	Figure 8-29
Function	8.14.10.1
Functional Diagram	Figure 8-28
Initialization	8.14.10.5
Modes, Operational	8.14.10.3.6
Options and Special Features	8.14.10.3
Override Feedback Processing	8.14.10.6
Parameters	8.14.10.7
Positions of Switch	
Control of by Operator (Equation A)	8.14.10.3.3
Control of with Equation B	8.14.10.3.4
Remote Cascade Options	8.14.10.3.7
Restart or Point Activation	8.14.10.3.8
Tracking Option	8.14.10.3.5
Use	8.14.10.2
Time-Constant Recommendations *(see also Variable Dead Time with Lead Lag)*	7.7.6.3
Timer, Box	9.4
Times, Box	9.5
Toledo Weigh Scale Interface	2.9.3
Totalizer Algorithm (TOTALIZR)	7.7.7
Accumulated Value before Reset	7.7.7.3
Bad-Input and Warm-Restart Options	7.7.7.3
Command Flags	7.7.7.3
Commands, States and Command Flags	7.7.7.3
Diagram of Use	Figure 7-9
Equations	7.7.7.4
Error Handling	7.7.7.3
Function	7.7.7.1
Near-Zero Cutoff	7.7.7.3
Restart or Point Activation	7.7.7.3
Special Features	7.7.7.3
Target-Value Flags	7.7.7.3
Time Base and Engineering Units Scaling	7.7.7.3
Typical Operation	7.7.7.3
Use	7.7.7.2

Index

Topic — **Section Heading**

TOTALIZR 7.7.7
USERID 3.7
Variable Dead Time with Lead-Lag Compensation Algorithm (VDTLDLG) 7.7.6
  Changing Dead-Time (Delay-Time) Parameters 7.7.6.3
  Dead Time (Delay-Time) Calculation 7.7.6.3
  Equations (forms of) 7.7.6.4
  Error Handling 7.7.6.3
  Four Combinations of Delay and Lead-Lag Compensation 7.7.6.3
  Function 7.7.6.1
  Functional Diagram Figure 7-7
  Options and Special Features 7.7.6.3
  Restrictions on Delay Time 7.7.6.3
  Time Constant Recommendations 7.7.6.3
  Use 7.7.6.2
  Variable-Delay Time Example 7.7.6.3
VDTLDLG 7.7.6
Warm Cold Restart 3.3.2
Windup 8.14.1.3.8

FAX Transmittal **FAX No.: (602) 313-4212**

Reader Comments

Title of Document: **APM Control Functions and Algorithms**

Document Number: **AP09-600** Issue Date: **12/03**

Comments: ______________________________

____

____

____

____

____

____

Recommendations: ______________________________

____

____

____

____

____

FROM:

Name: ______________________ **Date:** ____________

Title: ______________________________

Company: ______________________________

Address: ______________________________

City: ______________ **State:** ________ **ZIP:** ________

Telephone: ______________ **FAX:** ______________

FOR ADDITIONAL ASSISTANCE:

WRITE	CALL
Honeywell International Process Solutions 2500 West Union Hills Drive Phoenix, AZ 85027	**Technical Assistance Center (TAC)** 1-800-343-0228 (48 contiguous states except Arizona) Or **E-mail to**: pst@honeywell.com

Honeywell

Process Solutions

2500 West Union Hills Drive

Phoenix, AZ 85027