

Application Module Algorithm Engineering Data

AM09-601

**Implementation
Application Module - 1**

***Application Module
Algorithm Engineering Data***

**AM09-601
Release 640
12/02**

Copyright, Trademarks, and Notices

© Copyright 1995 - 2002 by Honeywell Inc.

Revision 03 – December 6, 2002

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.	1-800-343-0228	Sales
	Industry Solutions	1-800-525-7439	Service
		1-800-822-7673	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
Industry Solutions
16404 North Black Canyon Highway
Phoenix, AZ 85053
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 provides a complete description of the functions of each of the 11 PV algorithms and 13 control algorithms for regulatory data points in **TotalPlant** Solution (TPS) system Application Modules (AMs). The purpose of this publication is to be used as a reference manual for process engineers who are configuring control strategies for TPS systems, using one or more Application Modules.

Table of Contents

1. INTRODUCTION

- 1.1 References
- 1.2 Terms And Notation Used In The Algorithm Descriptions

2. TWO TYPES OF ALGORITHMS—PV ALGORITHMS AND CONTROL ALGORITHMS

- 2.1 PV Algorithms and Control Algorithms
- 2.2 Rolling Average Algorithm
 - 2.2.1 Function
 - 2.2.2 Use
 - 2.2.3 Configuration
 - 2.2.4 Equations
 - 2.2.5 Errors
 - 2.2.6 Compatibility with previous AM Releases

3. DATA ACQUISITION (PV)

- 3.1 Type And Name
- 3.2 Function
- 3.3 Use
- 3.4 Options And Special Features
- 3.5 Equation
- 3.6 Migration

4. FLOW COMPENSATION (PV)

- 4.1 Type And Name
- 4.2 Function
- 4.3 Use
- 4.4 Options And Special Features
 - 4.4.1 Five Forms of Flow Compensation
 - 4.4.2 Restart or Point Activation
 - 4.4.3 Error Handling
 - 4.4.4 Special Notes
 - 4.4.5 Compensating for Assumed Design Conditions
- 4.5 Equations
- 4.6 Migration

5. GENERAL LINEARIZATION (PV)

- 5.1 Type And Name
- 5.2 Function
- 5.3 Use
- 5.4 Options And Special Features
 - 5.4.1 Restart or Point Activation
 - 5.4.2 Error Handling
 - 5.4.3 Changing Parameters through a Universal Station
 - 5.4.4 Parameter—Value Restrictions
 - 5.4.5 Extension of First and Last Segments
- 5.5 Equation
- 5.6 Migration

6. HIGH SELECTOR, LOW SELECTOR, AVERAGE (PV)

- 6.1 Type And Name
- 6.2 Function

Table of Contents

- 6.3 Use
- 6.4 Options And Special Features
 - 6.4.1 Forced Selection
 - 6.4.2 Error Handling
 - 6.4.3 Restart or Point Activation
- 6.5 Equations
- 6.6 Migration

7. TOTALIZER (PV)

- 7.1 Type And Name
- 7.2 Function
- 7.3 Use
- 7.4 Options And Special Features
 - 7.4.1 Typical Operation
 - 7.4.2 Time base and Engineering-Units Scaling
 - 7.4.3 Commands and States
 - 7.4.4 Near-Zero Cutoff
 - 7.4.5 Target-Value Flags
 - 7.4.6 Bad-Input and Warm-Restart Options
 - 7.4.7 Restart or Point Activation
 - 7.4.8 Scheduling
 - 7.4.9 Error Handling
- 7.5 Equations
- 7.6 Migration

8. MIDDLE-OF-THREE SELECTOR (PV)

- 8.1 Type And Name
- 8.2 Function
- 8.3 Use
- 8.4 Options And Special Features
 - 8.4.1 Normal Operation with Three Valid Inputs
 - 8.4.2 Error Handling
- 8.5 Equations
- 8.6 Migration

9. MULTIPLIER/DIVIDER (PV)

- 9.1 Type And Name
- 9.2 Function
- 9.3 Use
- 9.4 Options And Special Features
 - 9.4.1 Ensuring Adequate PV Range
 - 9.4.2 Error Handling
 - 9.4.3 Restart or Point Activation
- 9.5 Equations
- 9.6 Migration

10. SUMMER (PV)

- 10.1 Type And Name
- 10.2 Function
- 10.3 Use
- 10.4 Options And Special Features
 - 10.4.1 Ensuring Adequate PV Range

Table of Contents

- 10.4.2 Error Handling
- 10.4.3 Restart or Point Activation
- 10.5 Equations
- 10.6 Migration

11. SUM OF PRODUCTS (PV)

- 11.1 Type And Name
- 11.2 Function
- 11.3 Use
- 11.4 Options And Special Features
 - 11.4.1 Ensuring Adequate PV Range
 - 11.4.2 Error Handling
 - 11.4.3 Restart or Point Activation
- 11.5 Equations
- 11.6 Migration

12. VARIABLE DEAD TIME WITH LEAD-LAG COMPENSATION (PV)

- 12.1 Type And Name
- 12.2 Function
- 12.3 Use
- 12.4 Options And Special Features
 - 12.4.1 Four Combinations of Delay and Lead-Lag Compensation
 - 12.4.2 Dead-Time (Delay-Time) Calculation
 - 12.4.3 Changing Dead-Time (Delay-Time) Parameters
 - 12.4.4 Restrictions on Delay Time
 - 12.4.5 Time-Constant Recommendations
 - 12.4.6 Using Equation C or D for a Fixed Delay Time
 - 12.4.7 Restart or Point Activation
 - 12.4.8 Error Handling
- 12.5 Equations
- 12.6 Migration

13. CALCULATOR (PV)

- 13.1 Overview
- 13.2 Type And Name
- 13.3 Function
 - 13.3.1 Calculation and Arithmetic Functions Supported
- 13.4 Use
- 13.5 Options And Special Features
 - 13.5.1 Calculator Expression Errors
 - 13.5.2 Error Handling of Bad-Inputs and Uncertain Values
- 13.6 Equations

14. USER-WRITTEN CL BLOCK (PV)

- 14.1 Type And Name
- 14.2 Function
- 14.3 Use
- 14.4 Options And Special Features
 - 14.4.1 Initialization
 - 14.4.2 Restart
 - 14.4.3 Processing Schedule and Execution Time
 - 14.4.4 Parameters Used for Comparisons

Table of Contents

- 14.4.5 Error Handling
- 14.5 Equations
- 14.6 Migration

15. AUTO MANUAL STATION (CONTROL)

- 15.1 Type And Name
- 15.2 Function
- 15.3 Use
- 15.4 Options And Special Features
 - 15.4.1 "Bumpless" Returns to Cascade Operation
 - 15.4.2 Operating Modes
 - 15.4.3 Input Value Range
 - 15.4.4 Restart or Point Activation
 - 15.4.5 Error Handling
- 15.5 Equations
- 15.6 Initialization
- 15.7 Override Feedback Processing
- 15.8 Automan Parameters
- 15.9 Migration

16. INCREMENTAL SUMMER (CONTROL)

- 16.1 Type And Name
- 16.2 Function
- 16.3 Use
- 16.4 Options And Special Features
 - 16.4.1 Handling of Full Value, Floating PID Outputs
 - 16.4.2 Input Value Range
 - 16.4.3 Changes to Incremental Summer Output by User-Written Programs
 - 16.4.4 Override Control Strategy and Past-Value Updating
 - 16.4.5 Operating Modes
 - 16.4.6 Restart or Point Activation
 - 16.4.7 Error Handling
- 16.5 Equations
- 16.6 Initialization
- 16.7 Override Feedback Processing
- 16.8 Incrsum Parameters
- 16.9 Migration

17. LEAD-LAG (CONTROL)

- 17.1 Type And Name
- 17.2 Function
- 17.3 Use
- 17.4 Options And Special Features
 - 17.4.1 Operating Modes
 - 17.4.2 Eliminating a Lead or Lag Term
 - 17.4.3 Time-Constant Recommendations
 - 17.4.4 Restart or Point Activation
 - 17.4.5 SP Value Range
- 17.5 Equations
- 17.6 Initialization
- 17.7 Override Feedback Processing
- 17.8 Leadlag Parameters
- 17.9 Migration

Table of Contents

18. MULTIPLIER/DIVIDER (CONTROL)

- 18.1 Type And Name
- 18.2 Function
- 18.3 Use
- 18.4 Options And Special Features
 - 18.4.1 Operating Modes
 - 18.4.2 SP Value Range
 - 18.4.3 Restart or Point Activation
 - 18.4.4 Error Handling
- 18.5 Equations
- 18.6 Initialization
- 18.7 Override Feedback Processing
- 18.8 Muldiv Parameters
- 18.9 Migration

19. PID (CONTROL)

- 19.1 Type And Name
- 19.2 Function
- 19.3 Use
- 19.4 Options And Special Features
 - 19.4.1 Interactive and Noninteractive PID Forms
 - 19.4.2 Four Combinations of Control Terms
 - 19.4.3 Control by a Single Term
 - 19.4.4 Direct and Reverse Control Action
 - 19.4.5 PV Tracking
 - 19.4.6 Gain Options
 - 19.4.7 Windup Handling
 - 19.4.8 Suppression of Output "Kicks" When Switching to CAS Mode
 - 19.4.9 Initializing PID Output without Affecting Dynamics
 - 19.4.10 Restrictions on Some Values
 - 19.4.11 Ratio Control
 - 19.4.12 Operating Modes
 - 19.4.13 Restart or Point Activation
 - 19.4.14 Error Handling
- 19.5 Equations
- 19.6 Initialization
- 19.7 Override Feedback Processing
- 19.8 Pid Parameters
- 19.9 Migration

20. PID FEEDFORWARD (CONTROL)

- 20.1 Type And Name
- 20.2 Function
- 20.3 Use
- 20.4 Options And Special Features
 - 20.4.1 Add or Multiply Action
 - 20.4.2 Bypassing Feedforward Control Action
 - 20.4.3 Feedforward Signal Value Status
- 20.5 Equations
- 20.6 Initialization
- 20.7 Override Feedback Processing
- 20.8 Pidff Parameters

Table of Contents

20.9 Migration

21. PID WITH EXTERNAL RESET-FEEDBACK (CONTROL)

- 21.1 Type And Name
- 21.2 Function
- 21.3 Use
- 21.4 Options And Special Features
 - 21.4.1 Error Handling, RFB and TRFB Inputs
 - 21.4.2 Control Output Connections
- 21.5 Equations
- 21.6 Initialization
- 21.7 Override Feedback Processing
- 21.8 Piderfb Parameters
- 21.9 Migration

22. RATIO (CONTROL)

- 22.1 Type And Name
- 22.2 Function
- 22.3 Use
- 22.4 Options And Special Features
 - 22.4.1 Role of the Multiplier/Divider PV Algorithm
 - 22.4.2 Operating Mode
 - 22.4.3 Restart or Point Activation
 - 22.4.4 Error Handling
 - 22.4.5 SP Value Range
- 22.5 Equations
- 22.6 Initialization
- 22.7 Override Feedback Processing
- 22.8 Ratioctl Parameters
- 22.9 Migration

23. RAMP AND SOAK (CONTROL)

- 23.1 Type And Name
- 23.2 Function
- 23.3 Use
- 23.4 Options And Special Features
 - 23.4.1 Operational Features
 - 23.4.2 Changing Remaining Soak Time and Current Segment
 - 23.4.3 Guaranteed Soak Time
 - 23.4.4 Guaranteed Ramp Rate
 - 23.4.5 Mark Timer Functions
 - 23.4.6 Achieving Longer Sequences by Interconnecting RAMPSOAK Points
 - 23.4.7 How to Get Just One Sequence
 - 23.4.8 Changes of SP by Operators or User-Written Programs
 - 23.4.9 Notes on Ranges and Limits
 - 23.4.10 Restart or Point Activation
- 23.5 Equations
- 23.6 Initialization
- 23.7 Override Feedback Processing
- 23.8 Rampsoak Parameters
- 23.9 Migration

24. OVERRIDE SELECTOR (CONTROL)

Table of Contents

- 24.1 Type And Name
- 24.2 Function
- 24.3 Use
- 24.4 Options And Special Features
 - 24.4.1 Override and Bypass Options
 - 24.4.2 Restrictions
 - 24.4.3 Operating Modes
 - 24.4.4 Restart or Point Activation
 - 24.4.5 Error Handling
- 24.5 Equations
- 24.6 Initialization
- 24.7 Override Feedback Processing
 - 24.7.1 Override Feedback Initiation
 - 24.7.2 Override Feedback Propagation
- 24.8 Orsel Parameters
- 24.9 Migration

25. SUMMER (CONTROL)

- 25.1 Type And Name
- 25.2 Function
- 25.3 Use
- 25.4 Options And Special Features
 - 25.4.1 Single Input Sum and Four Input Sum
 - 25.4.3 Restart or Point Activation
 - 25.4.4 Error Handling
- 25.5 Equations
- 25.6 Initialization
- 25.7 Override Feedback Processing
- 25.8 Summer Parameters
- 25.9 Migration

26. SWITCH (CONTROL)

- 26.1 Type And Name
- 26.2 Function
- 26.3 Use
- 26.4 Options And Special Features
 - 26.4.1 Operator Control of Switch Position--Equation A
 - 26.4.2 Program Control of Switch Position—Equation B
 - 26.4.3 Tracking Option
 - 26.4.4 Operational Modes
 - 26.4.5 Restart or Point Activation
 - 26.4.6 Error Handling
 - 26.4.7 Input Value Range
- 26.5 Equations
- 26.6 Initialization
- 26.7 Override Feedback Processing
- 26.8 Switch Parameters
- 26.9 Migration

27. CL CONTROL ALGORITHM (CONTROL)

- 27.1 Type And Name
- 27.2 Function
- 27.3 Use

Table of Contents

27.4 Options And Special Features
27.4.1 Restart
27.4.2 Use of Key Control Subsystem Parameters
27.4.3 Error Handling
27.4.4 Processing Schedule and Execution Time
27.5 Equations
27.6 Initialization
27.7 Override Feedback Processing
27.8 CI Algorithm Parameters
27.9 Migration

INTRODUCTION

Section 1

This section provides

- *A list of reference publications.*
- *Definitions of the terms and notation used in this publication.*

1.1 REFERENCES

In addition to this publication, the following publications may be needed for the design of control strategies that use data points in Application Modules:

- *System Control Functions*, in the *Implementation/Startup & Reconfiguration - 1* binder; and *Application Module Control Functions*, in the *Implementation/Application Module - 1* binder. You should be familiar with the information in Section 3 and 4 in *System Control Functions* and with Section 3.1 in *Application Module Control Functions* before you refer to this manual and the publications listed below.
- *Process Manager Control Functions and Algorithms*, in the *Implementation/Process Manager - 2* binder.
- *Basic Controller Algorithm Engineering Data*, in the *Product Manual* binder in the BASIC System bookset.
- *Extended Controller Algorithm Engineering Data*, in the *Product Manual* binder in the BASIC System bookset.
- *Multifunction Controller Algorithm Engineering Data*, in the *Product Manual* binder in the BASIC System bookset.
- *Control Language/Application Module Overview*, in the *Implementation/Application Module - 2* binder.
- *Application Module Parameter Reference Dictionary*, in the *Implementation/Application Module - 1* binder.

1.2 TERMS AND NOTATION USED IN THE ALGORITHM DESCRIPTIONS

Some of the following terms have special meanings when used to describe the AM algorithms; others are usually meaningful to people familiar with computers and are defined here for those who may not be so familiar with them. In any case, the following terms have the meanings defined here when they are used in this publication.

Other terms used in this and other reference publications, such as "Active," "Inactive," "Configured," "Bad," "Uncertain," and "Normal," are defined in the *Application Module Control Functions* manual.

Algorithm	This is a procedure for calculating a result, solving a problem, or accomplishing some end. In TotalPlant Solution (TPS) systems, the algorithms do not actually reside in the data points that use them, but are retained in the memory of the process-connected boxes and modules that have them, and are used by the data points as the data points are processed; however, the algorithms operate as if they had continuous relationships with each other, and it is convenient to refer to them that way. For example, if algorithm A is said to be connected to algorithm B, you should understand that is true only as the data points using the algorithms are processed.
Bad Control	This is an indicator in the AM data points that use control algorithms. It is accessible to programs and to other data points, and may appear on Universal Station displays. When set, it indicates that something has happened that prevents the data point from properly processing the control algorithm.
Bad Control Return	This is an indicator similar to the Bad Control indicator that, when set, indicates that the Bad Control indicator has been reset and proper control is now possible.
Default	A default value is a value that the system uses if the process engineer elects not to configure a parameter. Default values are built into the system to eliminate the need for the engineer to enter values for parameters that are not used, or to let the system use a typical value.
Disposable	Control outputs from regulatory data points are sometimes referred to as "disposable" or "indisposable." To be disposable, the output must be connected to an input of at least one active secondary data point in CAscade mode, through a configured, active input; otherwise, the output is indisposable. "Indisposable" is a situation similar to the Initialization Manual condition in controllers (CB, EC, and MC) on the Data Hiway.

Indisposable	See Disposable.
Migration	This term means migration from SUPERVISORY/TOTAL or PMX Systems to modules, such as the Application Module. Adapting a control strategy that is in use on a SUPERVISORY/TOTAL or PMX System to modules on the Local Control Network is one form of migration. Most of the algorithm descriptions have a Migration discussion. Typically, these describe similar algorithms or features in the SUPERVISORY/TOTAL and PMX Systems.
Not a Number (NaN)	This is a value in a data point parameter that indicates that the parameter should contain a number, but does not. This could be because of an unavailable process input or because information from another data point is not available. Programs and other data points can access this indicator. A NaN data field on a Universal Station display appears as "-----" (six dashes).
Process-Connected	A process-connected box is a box on the Data Hiway or Universal Control Network that has electrical connections to the process instruments and control devices. Also, a process-connected data point (slot) is a point (slot) in such a box. The boxes include CB, EC, MC, DHP, and PIUs on the LCN as well as PM, APM and LMs on the UCN.
*	The asterisk is the symbol for multiplication commonly used in computer documentation. This symbol is used here because it may be used in other TPS system documentation or on Universal Station displays. $X*Y = X \ Y = XY$.
(())	$(X*Y(A + B)W*Z) = [X*Y(A + B)W*Z]$. In both cases, $A + B$ is calculated before the multiplication takes place. Such notation is common in computer documentation because of the limited number of parentheses and brackets in the printed character sets.

TWO TYPES OF ALGORITHMS—PV ALGORITHMS AND CONTROL ALGORITHMS Section 2

This section provides

- *Information about what to expect in the remaining sections of this publication. Those sections describe the AM's PV and control algorithms.*
- *A summary of the AM's processing sequence for regulatory data points.*
- *A description of the Rolling Average Algorithm option.*

2.1 PV ALGORITHMS AND CONTROL ALGORITHMS

Often, data points in the Application Module use both a PV algorithm and a control algorithm. If the data point is used for only data acquisition, it probably won't use a control algorithm.

As Figure 2-1 shows, PV algorithm execution takes place between the PV-input-processing step and the PV-filtering step in the regulatory data-point sequence. Control-algorithm execution takes place between deviation-alarm processing and control output processing. If a data point uses a PV algorithm and a PID control algorithm, the control algorithm usually uses the PV calculated by the PV algorithm.

Each of Sections 3 through 23 defines one algorithm. PV algorithms are described in Sections 3 through 12. Control algorithms are described in Sections 14 through 23.

Each of the algorithm descriptions has the same form and the same headings:

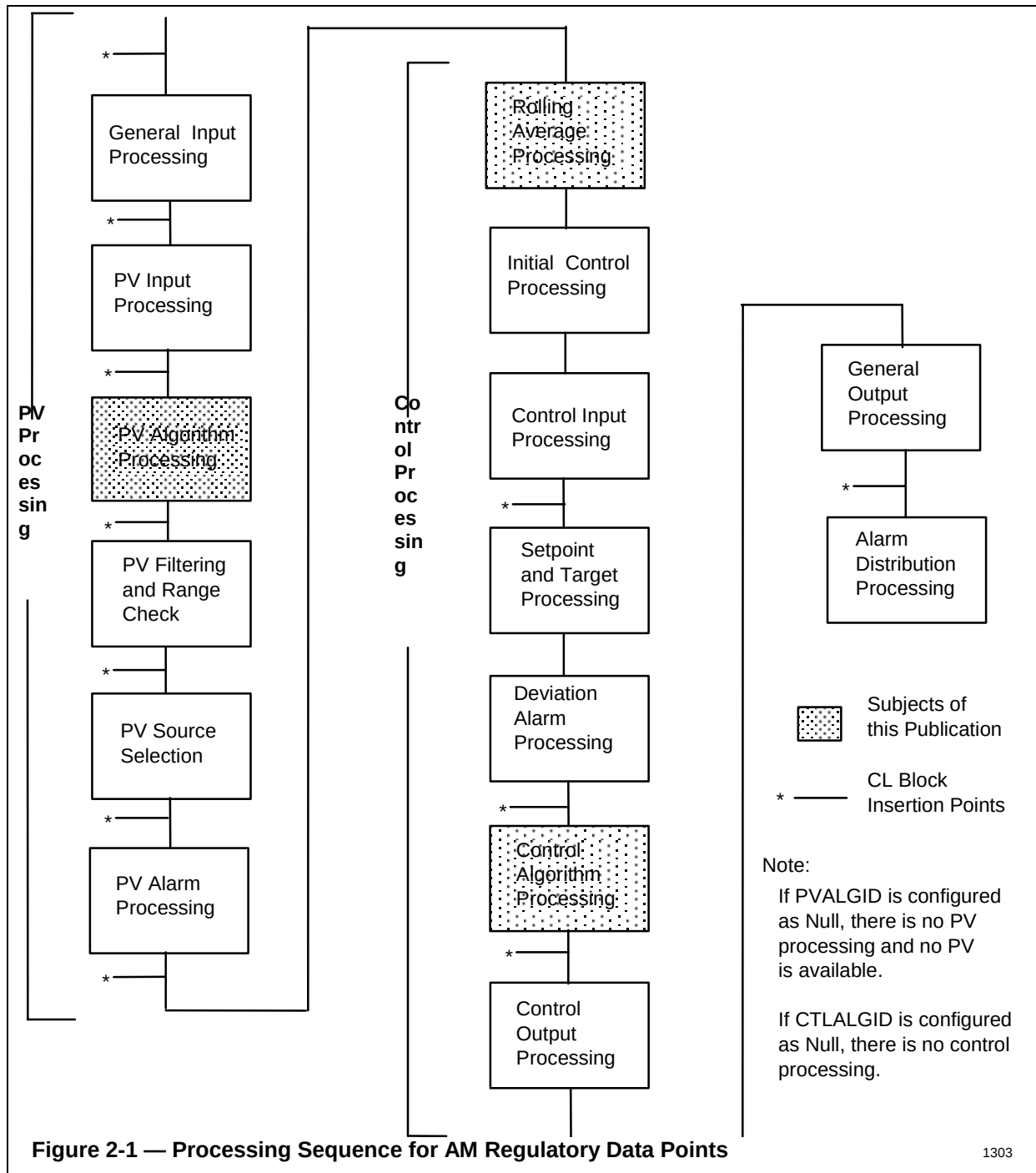
TYPE AND NAME
FUNCTION
USE
OPTIONS AND SPECIAL FEATURES
EQUATIONS
INITIALIZATION
OVERRIDE FEEDBACK PROCESSING
MIGRATION

Of these headings, Initialization and Override Feedback Processing do not apply to PV algorithms, so they are not included in Sections 3 through 13.

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 *Application Module Parameter Reference Dictionary*.

2.1

AM regulatory data points that are scheduled and have a PV algorithm that is not NULL are capable of calculating an average of the PV value (refer to Section 2.2 for the Rolling Average Algorithm option).



1303

2.2 ROLLING AVERAGE ALGORITHM

With R640, the Rolling Average algorithm option has been added to AM regulatory points. The following sections describe the function, use, configuration, equations, errors, and compatibility with previous AM releases.

2.2.1 Function

With the Rolling Average algorithm, the user can add rolling average calculations as an option to any scheduled AM regulatory point that has a real PV parameter. The Rolling Average algorithm parameters are exposed on the AM Regulatory point data-entry display when the PERIOD parameter is any value other than NOPERIOD and the PV algorithm (PVALGID) is not NULL.

2.2.2 Use

The Rolling Average calculates the parameter \$ROLLPV average of the process variable as follows:

$$\text{\$ROLLPV} = \frac{\text{Sum of (PVs_in_buffer)}}{\text{Number_of_ PVs_ in_buffer}}$$

The calculation is done in double-precision floating point when the AM processor supports double precision (K4/HMPU). Otherwise, the calculation is done in single precision floating point. The \$ROLLPV result is in single-precision floating-point format.

The \$ROLLPV parameter value is available to CL/AM and custom schematics. When the rolling average buffer is filled with samples, the oldest sample is removed from the average and replaced with a new PV sample.

2.2.2

The \$ROLLPV value is shown on the first page of the detail display as shown in Figure 2-2.

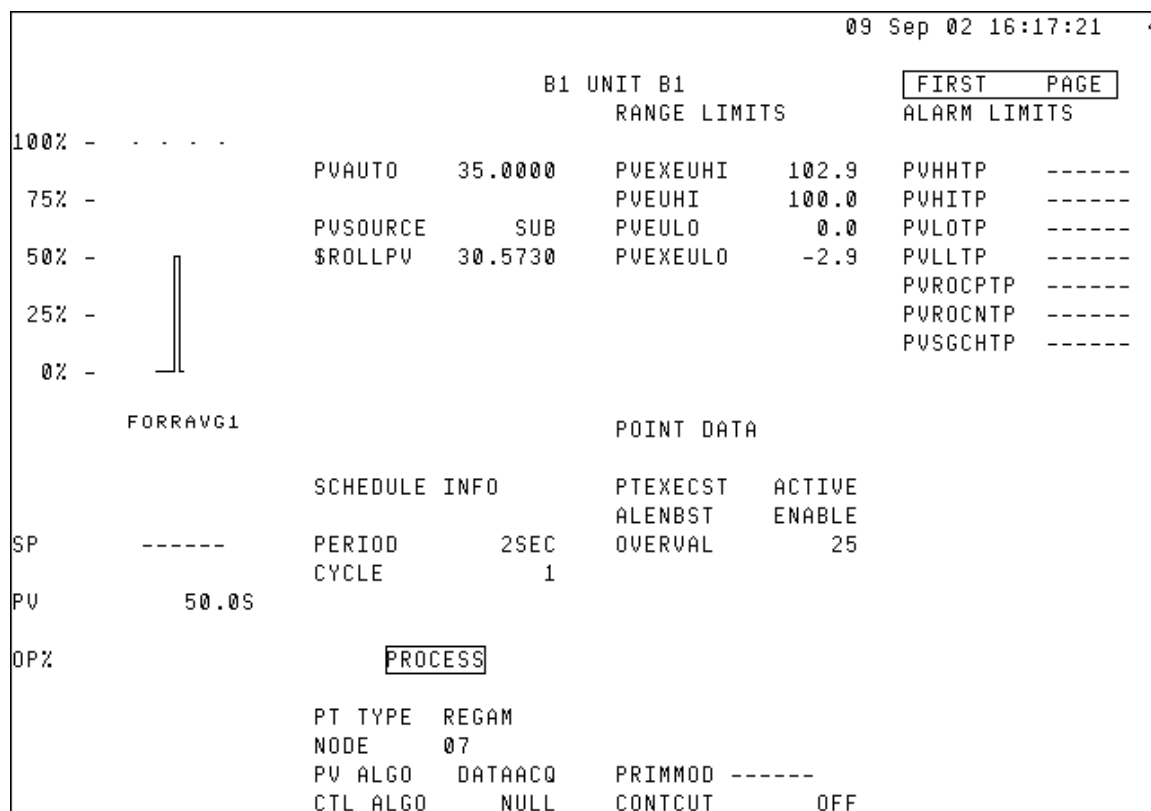


Figure 2-2 First page of Detail Display showing the \$ROLLPV parameter

2.2.2

Other parameters associated with the Rolling Average algorithm are \$ROLLAVG, \$ROLLMLT, \$ROLLOK, and \$ROLLBAD and are shown on the PV Algorithm page of the regulatory point's detail display. These parameters are described in the chart below. The \$ROLLPV value is repeated on this page.

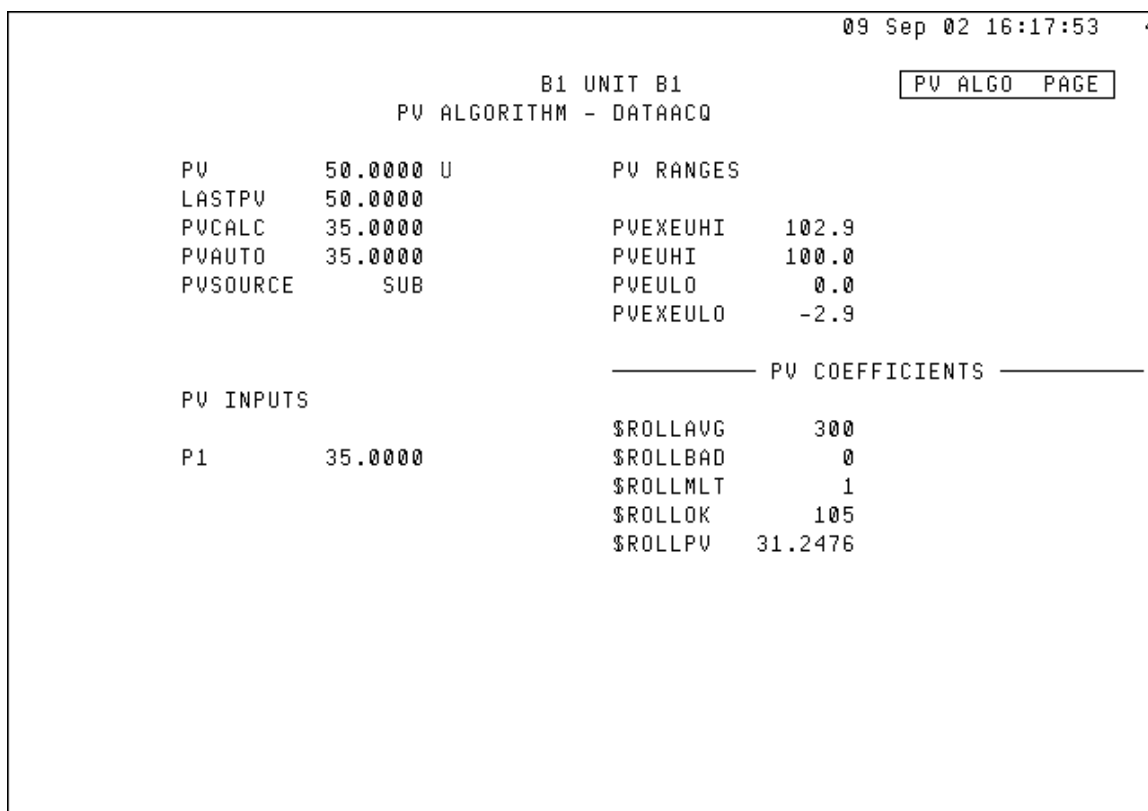


Figure 2-3 PV Algorithm Page of Detail Display

Point Name	Definition
\$ROLLAVG	Size of the rolling average PV sample buffer.
\$ROLLMLT	Multiple of the point execution PERIOD. It is the number of times the regulatory point processing executes before a PV is saved in the rolling average sample buffer.
\$ROLLOK	Total number of good PVs.
\$ROLLBAD	Total number of bad PVs.

2.2.2

When the rolling average is configured, the parameters \$ROLLRST, \$ROLLBBS, \$ROLLBUF, \$ROLLFBS, \$ROLLFRQ are added to the Configuration Data page of the Detail display as shown in Figure 2-4. The parameter \$ROLLRST can be edited from this page.

AMRC1		NETWORK VIEW <>		07 Sep 02 04:09:26		4
		B1 UNIT B1		CONFIG		PAGE
		CONFIGURATION				
		CONFIGURATION DATA				
SUPPIO	NOSUPPR	\$CDETAIL		BADPVPR	LOW	
CCINPT	NO	PVHHPR	LOW	BCLEALPR	LOW	
PVFORMAT	D1	PVHIPR	LOW	BCLFALPR	LOW	
PVFLTOPT	NONE	PVLOPR	LOW	CLEALMPR	LOW	
PVSRCOPT	ONLYAUTO	PVLLPR	LOW	CLFALMPR	LOW	
PVCLAMP	CLAMP	PVROCPPR	LOW	CNFERRPR	LOW	
PVALDB	ONE	PVROCNPR	LOW	\$AUXUNIT	--	
SPFORMAT	D1	PVSGCHPR	LOW	\$MPROD1		
EXTSWOPT	NONE	DEVHIPR	LOW	\$MPROD2		
MODEPERM	PERMIT	DEVLOPR	LOW	\$MPROD3		
SPOPT	NONE	ADVDEVPR	LOW	\$MPROD4		
RBOPT	NORATBI	\$SPTOL	0.00000			
INITTYPE	EXT	\$OPTOL	0.00000			
CTLEQN	EQA	\$ROLLBBS	HR			
CUTYPE	PERCENT	\$ROLLBUF	1			
PIDFORM	INTERACT	\$ROLLFBS	MIN			
CTLACTN	REVERSE	\$ROLLFRQ	1			
GAINOPT	LIN	\$ROLLRST	OFF			
PVTRACK	TRACK	PTDISCL	FULL			
ASSOCDSP		BADCTLPR	LOW			

Figure 2-4 Configuration Data Page of a Detail Display

Parameter	Definition
\$ROLLBUF	Size of the rolling average. This should be an integer value.
\$ROLLBBS	Unit of time in hours, minutes, or seconds for \$ROLLBUF
\$ROLLFRQ	Frequency of the rolling average. This should be an integer value.
\$ROLLFBS	Unit of time in hours, minutes, or seconds for \$ROLLFBS

When \$ROLLRST is set to ON, the count of samples in the rolling average buffer is reset to zero.

For a detailed description of the Rolling Average parameters refer to the *Application Module Parameter Reference Dictionary*.

2.2.3 Configuration

The Rolling Average algorithm is based on collecting the frequency and the buffer values from the user and then taking the average of the values to generate the rolling average. The user specifies the size of the rolling average buffer through parameter pair \$ROLLBUF & \$ROLLBBS, and the frequency of data collection through parameter pair \$ROLLFRQ & \$ROLLFBS, as shown in the data-entry display in Figure 2-5. The frequency must be an integer multiple of the point execution frequency (PERIOD).

AM-SCHEDULING DISPLAY					
POINT ASSIGNED TO IPP	(\$IPPASN)	☐ ON	☐ OFF		
POINT EXECUTION PERIOD	(PERIOD)	☐ 1SEC	☐ 2SEC	☒ 5SEC	☐ 10SEC
		☐ 15SEC	☐ 30SEC	☐ 1MIN	☐ 2MIN
		☐ 15MIN	☐ 25MIN	☐ 5MIN	☐ 10MIN
		☐ 15MIN	☐ 30MIN	☐ 1HR	☐ 8HR
		☐ 12HR	☐ 24HR	☐ NOPERIOD	
BEFORE/AFTER/CYCLE OPTION	(BEFAFT)	☒ NO	☐ BEFORE	☐ AFTER	☐ CYCLE
ROLLING AVERAGE BUFFER	(\$ROLLBUF)	1			
ROLLING AVERAGE BUFF BASE	(\$ROLLBBS)	1	☐ SEC	☐ MIN	☒ HR
ROLLING AVERAGE FREQUENCY	(\$ROLLFRQ)	1			
ROLLING AVERAGE FRQ BASE	(\$ROLLFBS)	1	☐ SEC	☒ MIN	☐ HR
F1=PED F3= F5=OVERWRITE F7=RECON F9 =WLK BACK F11=TAB F2=RECALL DISP F4= F6= F8=PED STATUS F10=WRITE F12=LOAD					

Figure 2-5 Parameter Entry Display showing Rolling Average Algorithm Parameters

2.2.4 Equations

The following are the equations for the \$ROLLAVG and \$ROLLMLT parameters.

$$\text{\$ROLLAVG} = \frac{\text{\$ROLLBUF\&\$ROLLBBS}}{\text{\$ROLLFRQ\&\$ROLLFBS}}$$

$$\text{\$ROLLMLT} = \frac{\text{\$ROLLFRQ\&\$ROLLFBS}}{\text{PERIOD}}$$

2.2.5 Errors

2.2.5.1 Point Build Error

Adding too many rolling averages during a point build results in a “data point too large” error message and the point load/reload is not done.

Although invalid rolling average parameter combinations result in successful point load completion, an error message describing the rolling average error is displayed and the rolling average parameters are left in a condition that will not calculate a rolling average. Correcting the error is required to calculate a rolling average. The invalid rolling average parameter combinations are as follows:

Error	Definitions
NULL PV ALGO ID	Null is specified as the PV algorithm
MUST SCHED PNT	AM regulatory point must be scheduled
BUFFER SPAN ERR	\$ROLLBUF or \$ROLLBBS value is illegal
FREQ SPAN ERR	\$ROLLFRQ or \$ROLLFBS value is illegal
BUFF/FREQ ERR	Size of sample buffer is illegal or buffer not an integer multiple of frequency. {e.g: frequency = 2MIN and buffer = 61 MIN is illegal (60 MIN buffer is o.k.)}
FREQ>BUFFER	Collection frequency is greater than total buffer span. {e.g.:frequency = 10 MIN and buffer = 5 MIN is illegal (50 MIN buffer is O.K.)}
FREQ/PERIOD ERR	Collection frequency is less than PERIOD or not an integer multiple of the PERIOD. {e.g.: frequency = 1 MIN and PERIOD = 2min is illegal (2 frequency would be O.K.)} {e.g.: frequency = 31 MIN and PERIOD = 15 min is illegal (30 frequency would be O.K.)}
BUFF MEMORY ERR	Insufficient user memory for rolling average sample buffer.

2.2.5.2 Run Time Error

Point process alarms are not generated from rolling average processing. A bad PV input to the rolling average algorithm is counted as a bad sample but not used in the calculation of the rolling average. A rolling average result (\$ROLLPV) will remain a bad value (NaN) until one good sample is collected.

The rolling average result (\$ROLLPV) will remain a bad value (NaN) when a rolling average point build error is not corrected and the AM regulatory point is activated. The number of rolling average samples (\$ROLLAVG) and the integer multiple of PERIOD parameter (\$ROLLMLT) will remain zero. When this happens the remainder of the AM regulatory point functions are unaffected and work normally.

2.2.5.3 Parameter Fetch Error

Reading the value of a rolling average parameter (\$ROLLPV, \$ROLLOK,...) when the sample buffer is not configured (\$ROLLBUF = 0) results in a parameter identifier error returned to the requestor.

2.2.5.4 Samples Over Limit Error

The AM Node is set to a WARNING status and a status notification message is generated when the number of running average samples processed (\$ROLLPRA) exceeds the rolling average trip point (\$ROLLATP). The node returns to OK status when the \$ROLLPRA value returns to be within the limits.

2.2.6 Compatibility with previous AM Releases

With R640, the Rolling Average algorithm function is “load & go”. No special migration procedure is required. The rolling average function is forward compatible because the function is an option on each AM regulatory point. Prior-release AM checkpoints (R630 and earlier) will default with the rolling averages feature ‘not active’.

If the rolling average function is not activated, the current release checkpoints (R640) are backward compatible. Prior releases of the AM (R630 and earlier) will not be affected with the rolling average feature being inactive.

If the rolling average feature is activated, the current-release checkpoints (R640) are not backward compatible. Prior releases of the AM (R630 and earlier) will see the point database with a data segment that cannot be understood and crash the AM when it is accessed.

2.2.6

Each AM regulatory point with the rolling average function enabled will increase the AM regulatory point size in user memory, and within the AM checkpoint volume. AM regulatory points that are currently at the maximum AM regulatory point size will not be able to add the rolling average function and will receive a point-build error.

Custom schematics on Universal Stations (R630 and earlier) will not have access to the rolling average values on the current AM (R640).

Universal Stations (R630 and earlier) that receive AM regulatory detail display page information (collection sets) will be able to display the value of rolling average parameters but will be unable to display the R640 parameter name correctly (SPARxxxx instead of \$ROLLPV).

Universal Stations (R640 and later) that build an AM regulatory point to an older AM (R630 and earlier) will receive a build-time error. The 'roll.sd' structure in release R640 prevents the newer US from building regulatory points in an older AM.

Parameter support on R630 and earlier AMs will return an error to requests for rolling average values.

To implement the rolling average function for existing AM regulatory points the user needs to modify the DEB IDs/EBs and add AM parameters (\$ROLLBUF, \$ROLLBBS, \$ROLLFRQ, \$ROLLFBS) to remain current.

DATA ACQUISITION (PV) Section 3

3.1 TYPE AND NAME

PV Algorithm: DATAACQ

3.2 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 3-1.

3.3 USE

The most common use of this algorithm is to provide a PV that has been through General Input Processing, PV Input Processing, PV Algorithm Processing, PV Filtering, and PV Source Selection (see Figure 2-1). 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. This algorithm could also be used to receive a calculated variable from a CL block, through a custom data segment.

The input to this algorithm and its output are in engineering units; for example, a temperature in °C from a thermocouple input to a Process Interface Unit (PIU).

3.4 OPTIONS AND SPECIAL FEATURES

This algorithm has no options or special features.

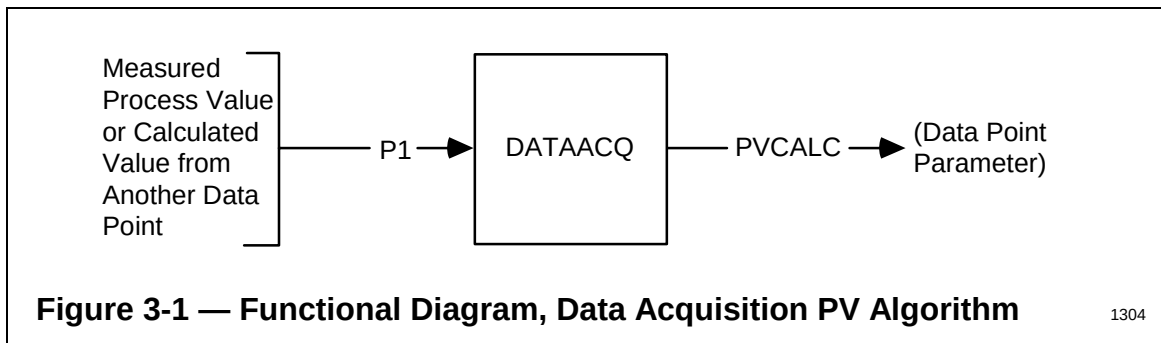
3.5 EQUATION

There is only one equation. The operation is simply the replacement of the data point's calculated PV (PVCALC) with the value of the input:

$$\text{PVCALC} = \text{P1}$$

Where P1 contains the first input value and PVCALC contains the value that becomes the PV when PVSOURCE = AUTO.

The parameters associated with this algorithm are P1, PVCALC, and P1STS. Refer to the *Application Module Parameter Reference Dictionary*.



3.6 MIGRATION

This Data Acquisition PV algorithm is like several similar PV algorithms in SUPERVISORY/TOTAL Systems and PMX Systems. The Hiway Gateway converts all process variables from the process-connected boxes, to engineering-units form, so only one Data Acquisition PV algorithm is needed in the AM.

Because the process-variable data supplied to SUPERVISORY/TOTAL and PMX Systems is not in such consistent form, several data-acquisition PV algorithms are used. Those algorithms are

- PV0 – Linear
- PV1 – Thermocouple Type J
- PV2 – Thermocouple Type K
- PV3 – Thermocouple Type T
- PV4 – Thermocouple Type S
- PV5 – Square Root Linearization
- PV6 – 100 Ohm RTD*
- PV7 – RH Radamatic*
- PV20 – Simple via Input Word
- PV21 – Simple via Input Source
- PV103 – 10 Ohm Copper RTD*
- PV104 – E Type Thermocouple Transmitter*
- PV105 – R Type Thermocouple Transmitter*

*Not available in PMX Systems.

FLOW COMPENSATION (PV)

Section 4

4.1 TYPE AND NAME

PV Algorithm: FLOWCOMP

4.2 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 4-1.

4.3 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 this data point, and the input value must be in engineering units. For process-connected inputs, the square root can be extracted in the process-connected box, and conversion to EUs can take place in the Hiway Gateway. See Figure 4-2.

The compensation is calculated from temperature, pressure, specific gravity, molecular weight, steam quality, or steam compressibility. Which of these is used depends on the type of compensation you choose. All of these inputs are obtained through PV input-connections.

4.4 OPTIONS AND SPECIAL FEATURES

4.4.1 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 4.4.5 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 a user-written program in an AM or in an upper-level processor connected to the Computer Gateway.

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.

4.4.2 Restart or Point Activation

On a cold, warm, or hot restart, or when this data point is activated, PVCALC is recalculated the next time the FLOWCOMP data point is processed.

4.4.3 Error Handling

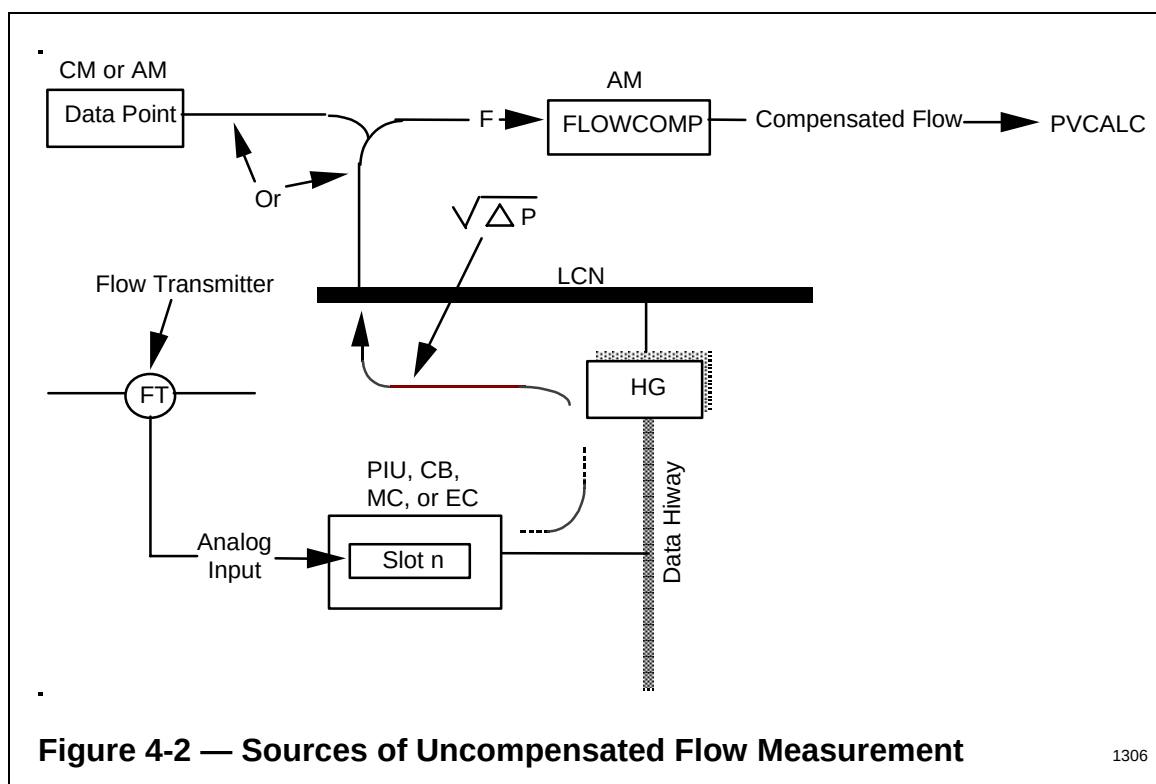
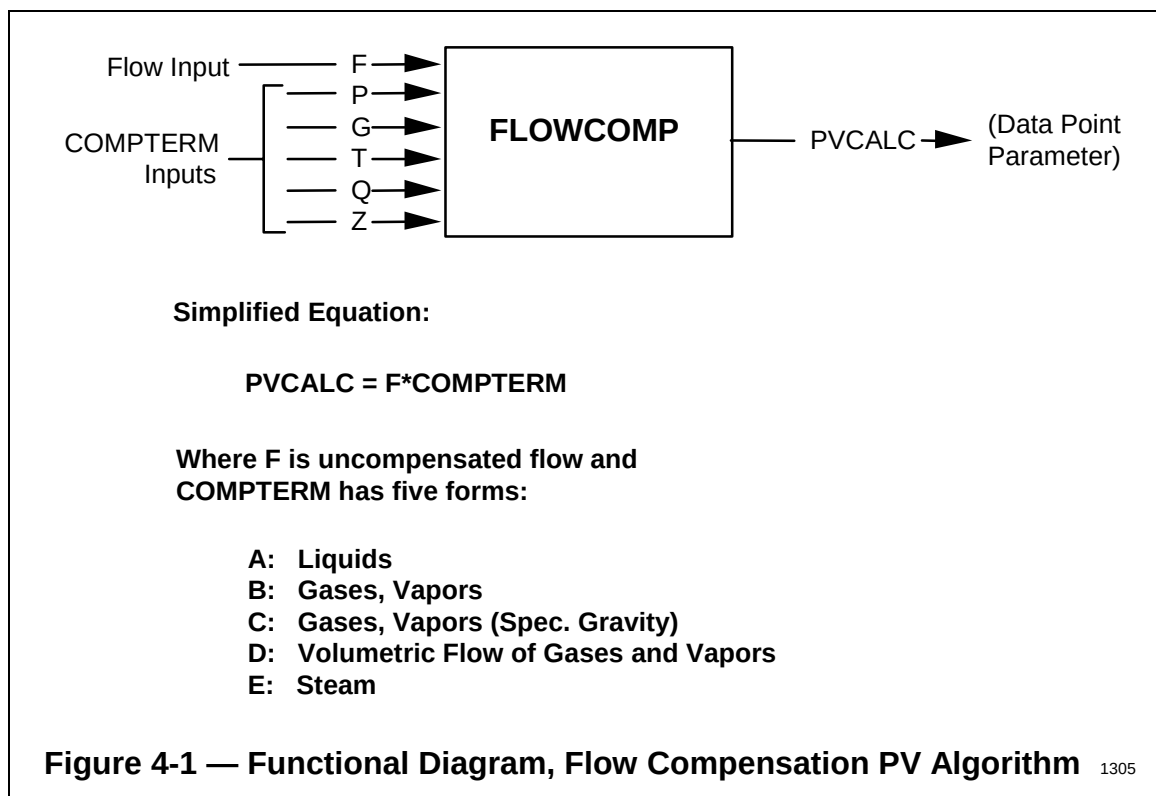
If the status of any of the input values is bad, PVCALC contains NaN and the PVAUTO status becomes Bad. If there are no bad inputs but the status of one or more of the inputs is "uncertain," the PVAUTO status becomes "uncertain."

4.4.4 Special Notes

Refer to 4.5 EQUATIONS for more detail on the inputs and parameters mentioned in these notes.

Zero Pressure Reference—Parameter P0 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.

Units of Measure—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 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 (omit the minus sign when you enter values in parameter T0).



Compensation Term Value—Typically, the COMPTERM value is near 1. It should never be zero or negative. The COMPL0 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 you expect.

Custody Transfer—Equation E can be used for "custody transfer" of gases or liquids. To do so, set parameter RX equals to one and specify the input connection to X to come from RX in this data point.

4.4.5 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 4.5) 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 a lab measures actual specific gravity, 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.

4.5 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

$$PVCALC = C * \frac{C1}{C2} * F * COMPTERM$$

Where:

PV CALC = The output of this algorithm. It is selected as the PV for this data point when the PV source is AUTOMATIC.

C = Scale factor. The default value is 1.0.

C1, C2 = Constants for correcting for assumed design conditions. See 4.4.5.
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:

$$\text{Equation A:COMPTERM} = \sqrt{\frac{G}{RG}} \quad (\text{Liquids})$$

$$\text{Equation B:COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0}} \quad (\text{Gases \& Vapors})$$

$$\text{Equation C:COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0} * \frac{G}{RG}} \quad (\text{Gases \& Vapors, w/Specific Gravity})$$

$$\text{Equation D:COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0} * \frac{RG}{G}} \quad (\text{Volumetric Flow of Gases \& Vapors})$$

$$\text{Equation E:COMPTERM} = \sqrt{\frac{P+P0}{RP} * \frac{RT}{T+T0} * \frac{X}{RX} * \frac{RQ}{Q}} \quad (\text{Steam})$$

4112

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 process engineer specifies the following parameters

RG = Reference specific gravity or reference molecular weight, in the same engineering units as G (Default value = 1.0).

RP = Reference pressure, in the same engineering units as P (Default value = 1.0).

RT = Reference temperature, in the same engineering units as T (default value = 1.0).

P0 = Zero reference for pressure, in the same engineering units as P. Typically 14.696 psia or 101.325 kPa. See 4.4.4. Default value = 0.

T0 = Zero reference for temperature, in the same engineering units as T. Typically -459.69°F or -273.15°C (omit the minus sign when entering a value in T0). See 4.4.4. Default value = 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 *Application Module Parameter Reference Dictionary*):

FSTS	QSTS	COMPTerm
GSTS	XSTS	PVCALC
PSTS	COMPLOLM	PVEQN
TSTS	COMPHILM	

4.6 MIGRATION

This Flow Compensation PV algorithm is like several similar PV algorithms in SUPERVISORY/TOTAL Systems and PMX Systems. The different forms of flow compensation are necessary in SUPERVISORY/TOTAL and PMX Systems to accommodate inputs in differing forms. The Hiway Gateway converts all process variables from the process-connected boxes, to engineering-units form, so only one Flow Compensation PV algorithm is needed in the AM.

Table 4-1 compares the AM Flow Compensation PV algorithm with the flow-compensation algorithms in SUPERVISORY/TOTAL and PMX Systems.

Table 4-1 — Comparison of SUPERVISORY/TOTAL and PMX Algorithms with FLOWCOMP

Nearest AM Equation	S/T-PMX Algorithm Number	S/T-PMX Equation in AM Terms	S/T-PMX Equations As Shown in S/T-PMX Publications
A	32	PV CALC = F*COMPTERM*C COMPTERM Variable: G	$PV = PV_S * S_P * V * K$ V-Term Variable: M
D	33	PV CALC = F*COMPTERM*C COMPTERM Variables: P, T, G	$PV = PV_S * S_P * V * K$ V-Term Variables: P, T, M
C	34	PV CALC = F*COMPTERM*C COMPTERM Variables: P, T, G	$PV = PV_S * S_P * V * K$ V-Term Variables: P, T, M
C	56	PV CALC = F*COMPTERM*C COMPTERM Variables: P, T, G (Square root of input flow, F, must already be extracted.)	$PV = K * V * \sqrt{\frac{X - D_L}{D_H - D_L}}$ V-Term Variables: P, T, G (The S/T-PMX algorithm extracts the square root of the uncompensated flow input.)
C	57	PV CALC = C*COMPTERM*F COMPTERM Variables: P, T, G (Square root of input flow, F, must already be extracted.)	$PV = K * V / PV_S$ V-Term Variables: P, T, M (The S/T-PMX algorithm extracts the square root of the uncompensated flow input.)
C	60	PV CALC = C*COMPTERM*F COMPTERM Variables: P, T, G	V-Term Variables: P, T, G
C	61	PV CALC = C*F*COMPTERM COMPTERM Variables: P, T, G	$PV = K * PV_S * \sqrt{G * \frac{P + K_P}{T + K_T}}$
A	132	PV CALC = F*COMPTERM*C COMPTERM Variable: G	$PV = PV_S * S_P * V * K$ V-Term Variable: M
$\frac{D}{C}$	$\frac{133}{134}$	PV CALC = F*COMPTERM*C COMPTERM Variables: P, T, G	$PV = PV_S * S_P * V * K$ V-Term Variables: P, T, M

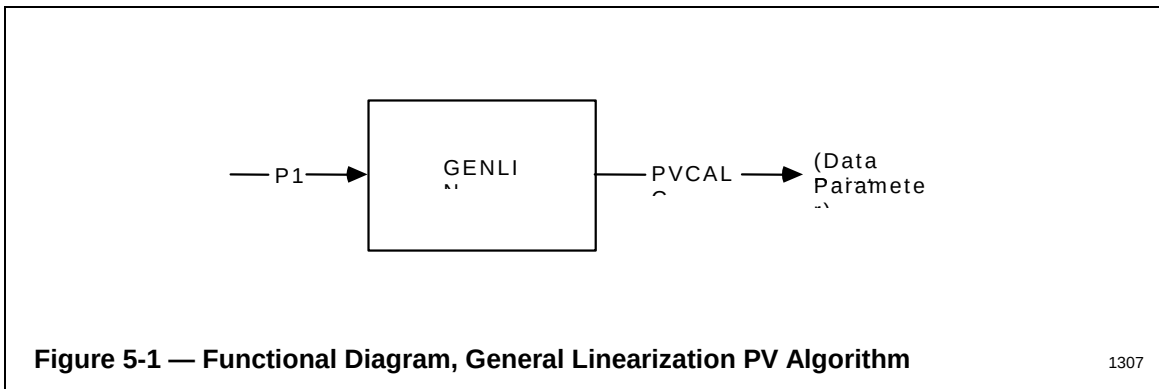
GENERAL LINEARIZATION (PV) Section 5

5.1 TYPE AND NAME

PV Algorithm: GENLIN

5.2 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 5-1 and 5-2.



5.3 USE

This algorithm is typically used to provide PVs in a linear range of 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 Hiway Gateway for standard temperature sensors and differential flow meters.

5.4 OPTIONS AND SPECIAL FEATURES

5.4.1 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.

5.4.2 Error Handling

If the status of the P1 input is "uncertain," the PVAUTO status becomes "uncertain."

If the status of the P1 input is bad or if any of the segment coordinates (IN_i or OUT_i) contains NaN, the PVAUTO-value status becomes bad.

If any of the segment coordinate values (IN_i or OUT_i) contains NaN, a configuration alarm is generated.

5.4.3 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.

5.4.4 Parameter—Value Restrictions

The input coordinate value parameters must be specified in ascending order from the smallest value to the largest.

5.4.5 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 IN_{segtot} (see 5.5), PVCALC is computed by assuming that the slope of the appropriate segment continues to the intersection point.

5.5 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),

$$\text{PVCALC} = \text{OUT}_i$$

- If P1 intersects the first segment (P1 < IN₁),

$$\text{PVCALC} = \frac{\text{OUT}_1 - \text{OUT}_0}{\text{IN}_1 - \text{IN}_0} * (\text{P1} - \text{IN}_0) + \text{OUT}_0$$

5.5

- 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],

$$PV\ CALC = \frac{OUT_{(i+1)} - OUT_i}{IN_{(i+1)} - IN_i} * (P1 - IN_i) + OUT_i$$

- 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.

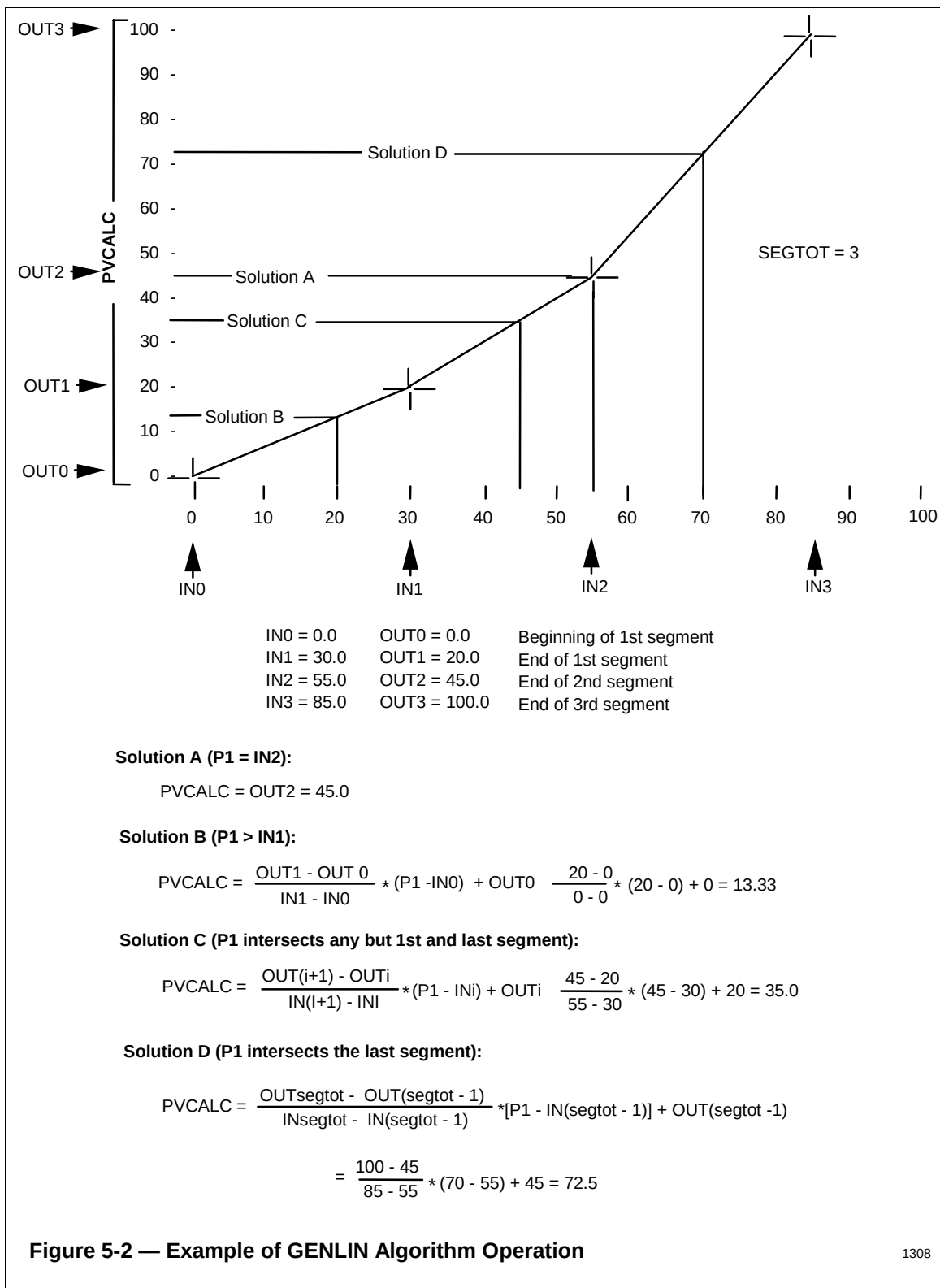
Other parameters associated with the GENLIN algorithm are as follows (refer to the *Application Module Parameter Reference Dictionary*):

P1STS PVCALC SEGTOT

5.6 MIGRATION

There are no similar algorithms in PMX and SUPERVISORY/TOTAL Systems.

The Extended Controller has a similar algorithm that offers up-to-eight segments.



1308

HIGH SELECTOR, LOW SELECTOR, AVERAGE (PV)

Section 6

6.1 TYPE AND NAME

PV Algorithm: HILOAVG

6.2 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-eight inputs. Valid inputs are those whose status is "Normal" or "Uncertain." When the input selection functions are used, the number of the input that is selected is contained in an accessible parameter (SELINP). See Figure 6-1.

6.3 USE

One example of the use of this algorithm is shown at the top of Figure 6-1. In this example, the high value-selector version of the algorithm is used to detect hot spots in a boiler or a reactor.

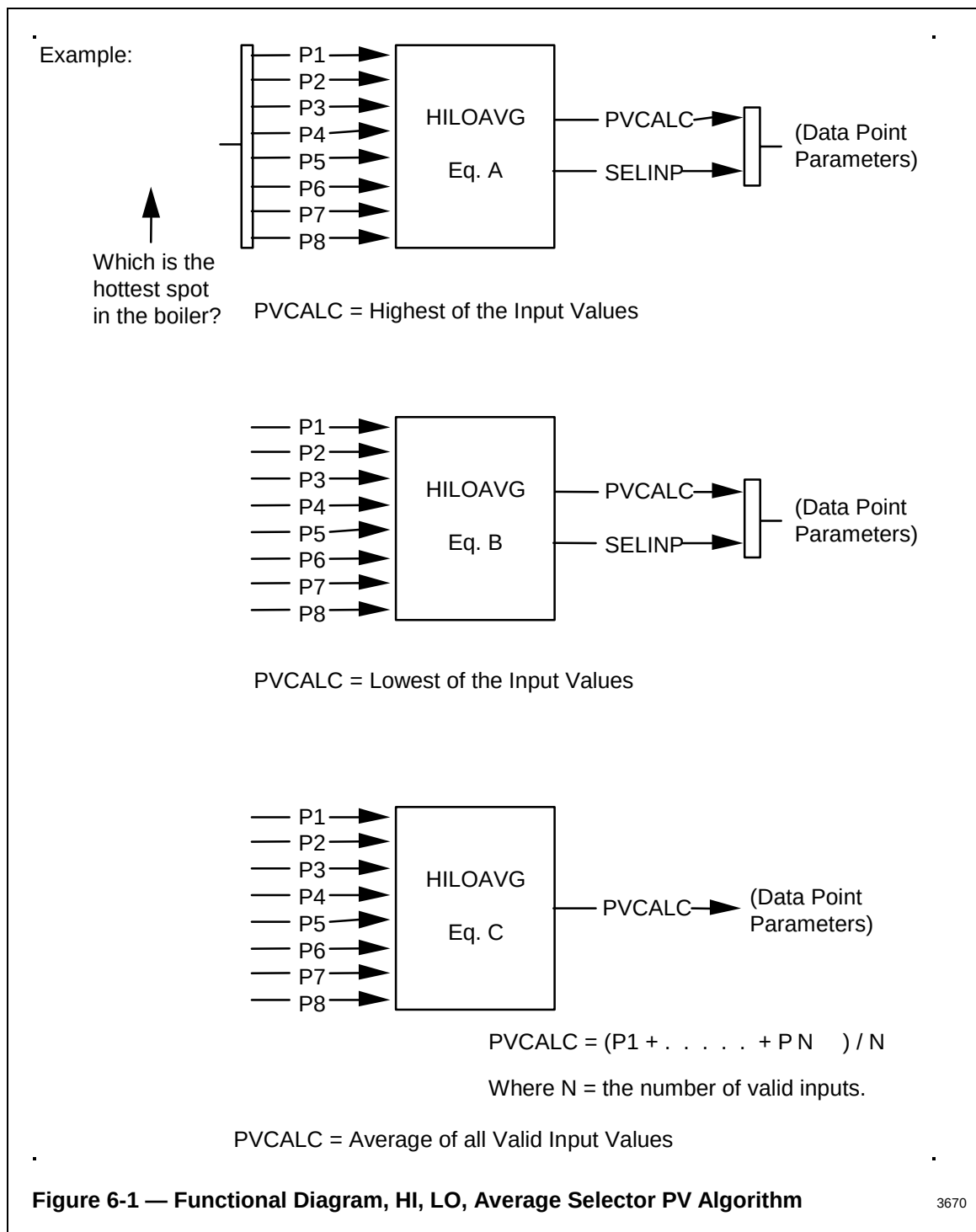
Either the high value-selector version or the low value-selector version can be used to detect production bottlenecks. For example, this algorithm might be used to notify the process operator that production is currently constrained by the speed of a gas compressor. One of the selector options might 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.

6.4 OPTIONS AND SPECIAL FEATURES

6.4.1 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.



3670

- If the FRCPERM parameter is configured as On, the forced-selection function is enabled and an operator, a user-written program, or a general input connection 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 SelectP8).

6.4.2 Error Handling

Except when forced selection is in effect (6.4.1), inputs with a bad status are ignored and they do not make the PVAUTO status 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 (PV status of good or uncertain) is less than the minimum number specified in parameter NMIN, PVCALC becomes NaN and the PVAUTO status is bad.

The value status of PVAUTO is changed to uncertain 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 the PVAUTO value-status 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.

6.4.3 Restart or Point Activation

On a cold, warm, or hot restart, or when this data point is activated, PVCALC is simply recalculated the next time this data point is processed.

6.5 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 *Application Module Parameter Reference Dictionary*):

NMIN	PVEQN
PnSTS	SELINP

6.6 MIGRATION

There are no similar algorithms in PMX and SUPERVISORY/TOTAL Systems.

TOTALIZER (PV) Section 7

7.1 TYPE AND NAME

PV Algorithm: TOTALIZR

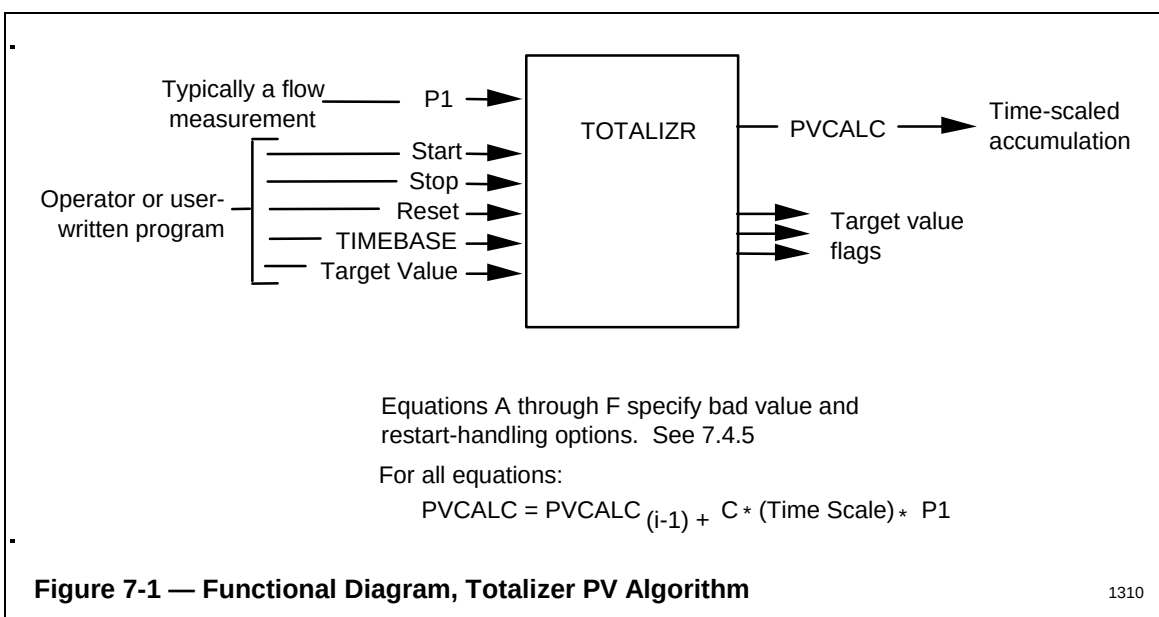
7.2 FUNCTION

This algorithm provides a time-scaled accumulation of a single-input value. The input value is typically a flow measurement. The time base can be seconds, minutes, or hours.

A data point that uses this algorithm cannot use a control algorithm.

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.



1310

7.3 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.

7.4 OPTIONS AND SPECIAL FEATURES

7.4.1 Typical Operation

The events in an operation that uses TOTALIZR might be as follows (see Figure 7-2):

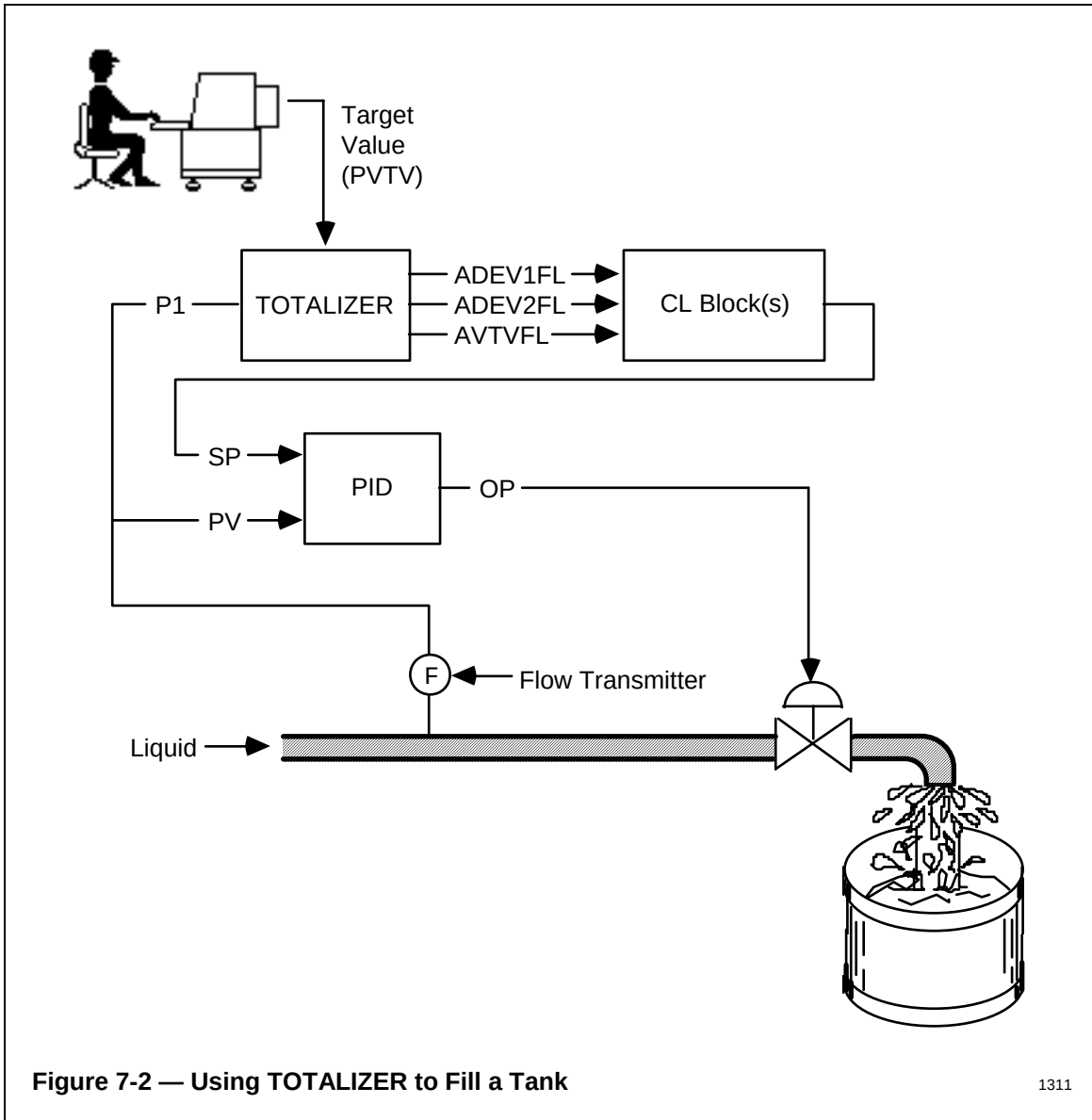
- The target value, which represents the desired total volume, is specified to the PVTV parameter in the TOTALIZR point, by an operator at a Universal Station or by a user-written program.
- An operator or a user-written program issues a RESET command to TOTALIZR point. This sets any accumulation value equal to RESETVAL.
- A START command is issued to the TOTALIZR point. A CL block inserted in the processing of one of the points uses the setpoint Target Value function (see 3.1.6.2 in AM Control Functions) in the PID point, to "ramp" the flow up to a steady rate.
- When the first "slowdown" or "near-target" flag (ADEV1FL) comes on, another CL block ramps the flow SP down to a lower value.
- When the second "slowdown" or "near-target" flag (ADEV2FL) comes on, the flow SP is lowered to a trickle.
- When the accumulation reaches the target value, filling is complete and the complete flag (AVTVFL) comes on. A CL block shuts the flow off. The TOTALIZR point's PV-high alarm can be configured to trip at this point, so the operator is notified that filling is complete.

7.4.2 Time base and Engineering-Units Scaling

The user specifies the timebase in seconds, minutes, or hours, in parameter TIMEBASE. This is the timebase in which the flow measurement is made. For example, liters per second.

7.4.3

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.



1311

7.4.3 Commands and States

Three 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 accumulation value to a user-specified value. This value is specified in parameter RSETVAL. If the accumulator is running, it continues from the reset value.

7.4.4 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 equal to or below CUTOFFLM, it is replaced by zero. You can eliminate this feature by specifying NaN in CUTOFFLM.

7.4.5 Target-Value Flags

The target value can be specified by an operator by storing it in PVTV. A user-written program can specify it by storing in AVTV. These parameters track each other. This feature can be disabled by storing NaN in AVTV. NaN cannot be stored by a CL program; it must be done by the Operator.

When the accumulated value in PVCALC exceeds 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. They are specified as deviations from AVTV. Each of them is associated with a flag:

AVDEV1FL trips when

$$\text{PVCALC} > \text{AVTV} - \text{AVDEV1TP}$$

AVDEV2FL trips when

$$\text{PVCALC} > \text{AVTV} - \text{AVDEV2TP}$$

When the accumulated value (PVAUTO) status is bad, AVTVFL, AVDEV1FL, and AVDEV2FL are all Off.

7.4.6 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 P1's value status goes bad, P1's value is replaced by zero and the accumulation continues with the PVAUTO status uncertain. When P1 is again good, PVAUTO remains uncertain until a reset command is received. No special action by the operator is required.
- **Use Last Good Value**—When the accumulator is running, if P1's value status goes bad, P1's value is replaced by the last good value and the accumulation continues with the PVAUTO status uncertain. When P1 is again good, PVAUTO remains uncertain until a reset command is received. No special action by the operator is required.
- **Set PVAUTO Status Bad and Stop**—When the accumulator is running, if P1's value status goes bad, the value in PVCALC becomes NaN, the PVAUTO status goes bad and the accumulator is stopped. If the PV source is AUTO, a bad-PV alarm is generated. When P1 is again normal, PVAUTO remains bad until the accumulator is started again. To restart the accumulation, the operator should estimate its value and use the reset command (see 7.4.3) 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 PVAUTO status goes to uncertain and remains so until a reset command is received.
- **Set PVAUTO Status Bad and Stop After a Warm Restart**—On a warm restart when the accumulator is running, the value in PVCALC becomes NaN, the PVAUTO status 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 P1-value status is ignored. If the accumulator is stopped on a warm restart, no special action by the operator is required.

7.4.7 Restart or Point Activation

When the TOTALIZR data point is activated or on a cold restart, the PVCALC value becomes NaN, PVAUTO status goes bad and the accumulator state is Stopped. If the PV source 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 7.4.6.

7.4.8 Scheduling

A data point that uses TOTALIZR must be scheduled after the point that supplies TOTALIZR's P1 input.

7.4.9 Error Handling

The PVAUTO value status is uncertain when

- The P1-value status is uncertain.
- The P1-value status is bad and "use zero" or "use last value" (Equations A, B, D, or E) is configured (see 7.4.6).
- The data point is in a warm restart and the continue option (Equations A, B, or C) is configured (see 7.4.6).

A reset command is needed to return the PVAUTO-value status to normal, provided the P1 status is normal.

PVCALC contains NaN and the PVAUTO-value status is bad when

- The P1-value status is bad and "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 the PVAUTO-value status to normal, provided the P1 status is normal.

7.5 EQUATIONS

You configure one of Equations A through F for a TOTALIZR data point; however, the equation specifies the operating bad-input and warm-restart options according to 7.4.6, and doesn't affect the accumulator calculation.

For all equations, when the accumulator is running, the accumulated value in PVALC is calculated as follows:

$$PVALC_{(i)} = PVALC_{(i-1)} + C * (Time-scale) * P1$$

Where

PVALC =	The output of this algorithm. It is selected as the PV for this data point when the PV source is AUTOMATIC.
PVALC _(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 Sec. TS if TIMEBASE contains Min. TS/60 if TIMEBASE contains Hrs.
TS =	The data point processing interval in minutes.
P1 =	The input value. Typically a flow rate.

Other parameters associated with the TOTALIZR algorithm are as follows (refer to the *Application Module Parameter Reference Dictionary*):

P1STS PVALC PVEQN

7.6 MIGRATION

There are some similarities between TOTALIZR and PV algorithm 31 in SUPERVISORY/TOTAL and PMX Systems. PV algorithm 31 in those systems does not accept start, stop, and reset commands. Its accumulation begins when the point is activated.

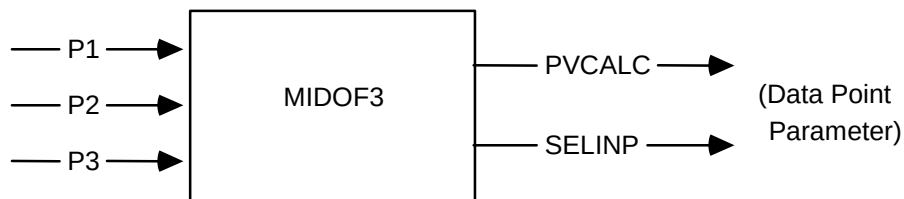
MIDDLE-OF-THREE SELECTOR (PV) Section 8

8.1 TYPE AND NAME

PV Algorithm: MIDOF3

8.2 FUNCTION

This algorithm provides a calculated PV (PVCALC) that is normally the middle value of three values from active PV-input connections. The PVAUTO status goes 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. See Figure 8-1.



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 8-1 — Functional Diagram, Middle-of-Three Selector PV Algorithm

1312

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.

8.3 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 eight input connections (see Section 6).

8.4 OPTIONS AND SPECIAL FEATURES

8.4.1 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.

8.4.2 Error Handling

The PVAUTO status becomes uncertain 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.

8.5 EQUATIONS

If three valid inputs are present, the equations have no meaning and the algorithm functions normally, as described under 8.4.1. 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, and B

PVCALC = the value of the valid input

SELINP = The selected input, SelectP1 through SelectP3

Equation C

PVCALC = the value of the valid input

SELINP = The selected input

- With three bad inputs

Equations A, B, and C

PVCALC = NaN

SELINP = None

Where:

PV CALC =	The output of this algorithm. It is selected as the PV for the data point when the PV source is AUTOMATIC.
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 PV CALC contains an average value, SELINP contains None.

Other parameters associated with the MIDOF3 algorithm are as follows (refer to the *Application Module Parameter Reference Dictionary*):

P1STS	P3STS
P2STS	PVEQN

8.6 MIGRATION

PV algorithm no. 54 in SUPERVISORY/TOTAL is similar to this algorithm with Equation B selected. There is a similar algorithm in the Extended Controller that selects the lower of two inputs if a third input is not available.

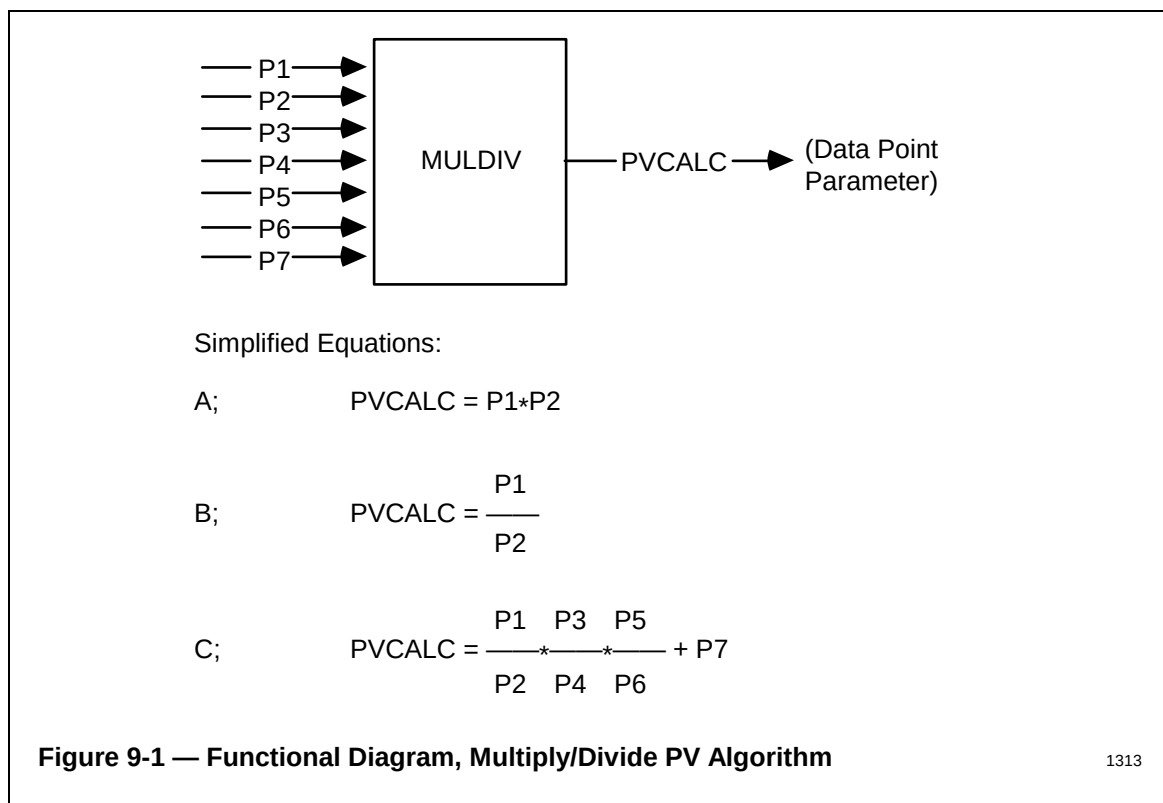
MULTIPLIER/DIVIDER (PV) Section 9

9.1 TYPE AND NAME

PV Algorithm: MULDIV

9.2 FUNCTION

This algorithm calculates a PV (PVCALC) that is either the product of two inputs (Equation A), a quotient of two inputs (Equation B), or the product of three quotients (Equation C). The products and quotients can be scaled and bias values can be added to them. See Figure 9-1.



9.3 USE

Some uses, in the approximate order of importance, are

- Scaling and biasing of process-connected inputs.
- Engineering-units conversions.
- Miscellaneous process calculations.

The following are some examples of use:

- Calculation of a scaled ratio between two flows

$$\text{PVCALC} = C \cdot P1 / P2 \quad \text{-- Equation B}$$

- Conversion of degrees API to specific gravity

$$\text{PVCALC} = 141.5 / (P2 + 131.5) \quad \text{-- Equation B}$$

- Conversion of the liquid level in an elevated vessel to pressure (D1 is the elevation of the vessel).

$$\text{PVCALC} = C \cdot (P1 + D1) \quad \text{-- Equation A}$$

- Scaling and biasing a bottleneck-detection input

$$\text{Severity} = \text{PVCALC} = C \cdot P1 + D1 \quad \text{-- Equation A}$$

Equation B or C is used with the Ratio Control Algorithm (see Section 19) to create a ratio-control data point whose setpoint is the desired ratio, and whose output is a setpoint to a flow controller. This algorithm provides the PV as a scaled ratio, therefore, it is a measure of the ratio actually attained.

9.4 OPTIONS AND SPECIAL FEATURES

9.4.1 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 very 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.

9.4.2 Error Handling

If there are no inputs with a bad status and the status of at least one input is uncertain, the PVAUTO-value status is uncertain.

If the status of at least one input is bad, the PVAUTO-value status becomes bad and PVCALC contains NaN.

9.4.3 Restart or Point Activation

On any type of restart or when this data point is activated, PVCALC is normally calculated.

9.5 EQUATIONS

You can select any one of three equations when configuring a data point that uses the Multiplier/Divider PV algorithm:

Equation A

$$\text{PVCALC} = C * (C1 * P1 + D1) * (C2 * P2 + D2) + D$$

Equation B

$$\text{PVCALC} = C * \frac{(C1 * P1 + D1)}{(C2 * P2 + D2)} + D$$

Equation C

$$\text{PVCALC} = C * \frac{(C1 * P1 + D1)}{(C2 * P2 + D2)} * \frac{(C3 * P3 + D3)}{(C4 * P4 + D4)} * \frac{(C5 * P5 + D5)}{(C6 * P6 + D6)} + (C7 * P7 + D7) + D$$

Where:

PVCALC = The output of this algorithm. It is selected as the PV for the data point when the PV source is AUTOMATIC.

P1 through P7 = The input values. The P1 default value is NaN. Default values for P2 through P6 are 1.0. For P7, the default value is 0.

C = The overall scale factor. The default value is 1.0.

D = The overall bias value. The default value is 0.

C1 through C7 = Scale factors for Pn inputs with the same number. The default value for each is 1.0.

D1 through D7 = Bias values for the scaled inputs with the same number. The default value for each is 0.

Other parameters associated with the MULDIV algorithms are as follows (refer to the *Application Module Parameter Reference Dictionary*):

PnSTS PVEQN

9.6 MIGRATION

The Multiply/Divide PV algorithm can be used to duplicate the functions of several PV algorithms in SUPERVISORY/TOTAL and PMX Systems. Because Application Module algorithms deal only with values in engineering units, just one algorithm can handle the functions of several algorithms in SUPERVISORY/TOTAL and PMX Systems. Table 9-1 compares the algorithms.

Table 9-1 — Comparison of SUPERVISORY/TOTAL and PMX PV Algorithms With MULDIVP

Nearest AM Eq	S-T/PMX Algo No.	S/T-PMX Equation in AM Terms	S/T-PMX Equation as shown in S/T-PMX Pubs
A	22	$PV_{CALC} = P1 * C$	$PV = PV_S * C$
A	23	$PV_{CALC} = P1 * C$	$PV = PV_{LINEAR} * C$
A	25	$PV_{CALC} = P1 * C$	$PV = PV_{SR} * C$
B	62*	$PV_{CALC} = \frac{C1}{P2 - D2}$	$PV = \frac{C1}{PV_S - C2}$
C	63*	$PV_{CALC} = \frac{C1}{P2} D$	$PV = \frac{C1}{PV_S} C2$
B	116/101	$PV_{CALC} = \frac{P1}{P2} * C$	$PV = \frac{F_{OUT}}{F_{IN}} * K$
*Not available on PMX Systems.			

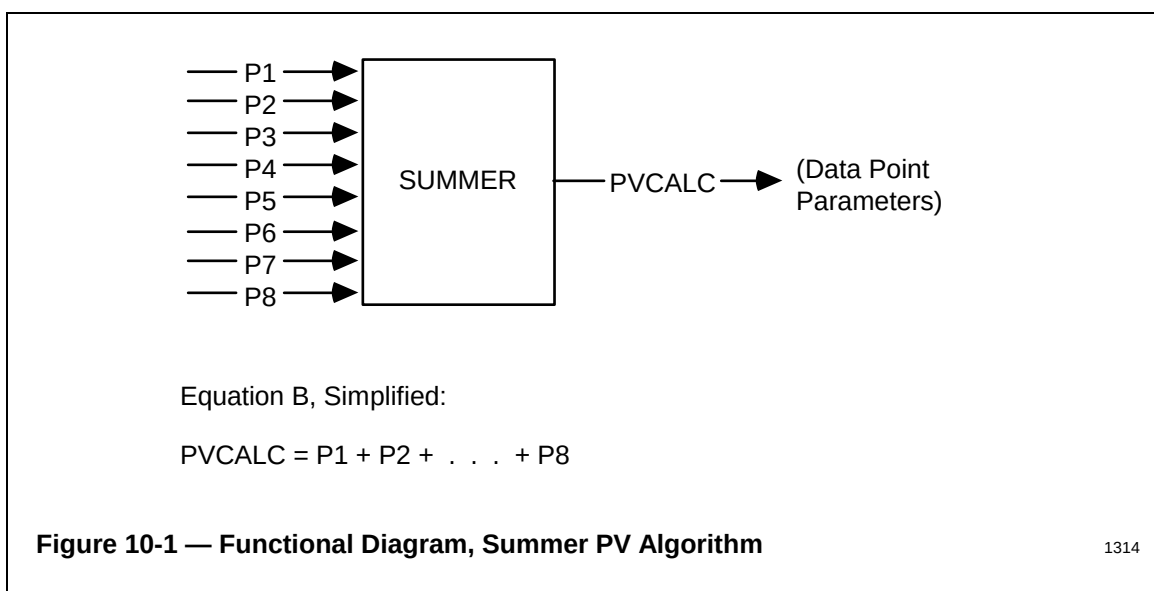
SUMMER (PV) Section 10

10.1 TYPE AND NAME

PV Algorithm: SUMMER

10.2 FUNCTION

This algorithm calculates a PV (PVCALC) that is the sum of up to eight 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 10-1.



10.3 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.

10.4 OPTIONS AND SPECIAL FEATURES

10.4.1 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 very 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.

10.4.2 Error Handling

If there are no inputs with a bad status and the status of at least one input is uncertain, the PVAUTO-value status is uncertain.

If the status of at least one input is bad, the PVAUTO-value status becomes bad and PVCALC contains NaN.

10.4.3 Restart or Point Activation

On any type of restart or when this data point is activated, PVCALC is normally calculated.

10.5 EQUATIONS

You can select one of two equations when you configure a data point that uses the Summer PV algorithm:

Equation A

$$\text{PVCALC} = C * P1 + D$$

Equation B

$$\text{PVCALC} = C * (C1 * P1 + C2 * P2 + . . . + Cn * Pn) + D$$

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 = 1.0.

C1 through Cn = The scale factors for P1 through Pn. Default = 1.0.

P1 through Pn = The PV input values. 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 as follows (refer to the *Application Module Parameter Reference Dictionary*):

N PnSTS PVEQN

10.6 MIGRATION

The Summer PV algorithm can accomplish the function of four similar algorithms in SUPERVISORY/TOTAL and PMX Systems. Table 10-1 compares those algorithms to this one.

Table 10-1 — Comparison of SUPERVISORY/TOTAL Algorithms With SUMMER

Nearest AM Equation	S-T/PMX Algorithm Number	S-T/PMX Equation in AM Terms	S-T/PMX Equation as shown in S/T-PMX Pubs.
B	26	$PV_{CALC} = P_1 + \dots + P_n$ n = 1 through 8	$PV = \sum_{i=1}^N J_1$ n = 1 through 15
B	43	$PV_{CALC} = C_1 * P_1 + C_2 * P_2$	$PV = \frac{A_1}{100} * F_1 + \frac{A_2}{100} * F_2$
B	120/103	$PV_{CALC} = C_1 * P_1 + \dots + C_n * P_n$ n = 1 through 8	$PV = \sum_{i=1}^N (J_1 * W_{F1})$ N = 1 through 14
B	115*	$PV_{CALC} = P_1 - P_2$	$PV = I_1 - I_2$
*Not available in PMX Systems.			

11146

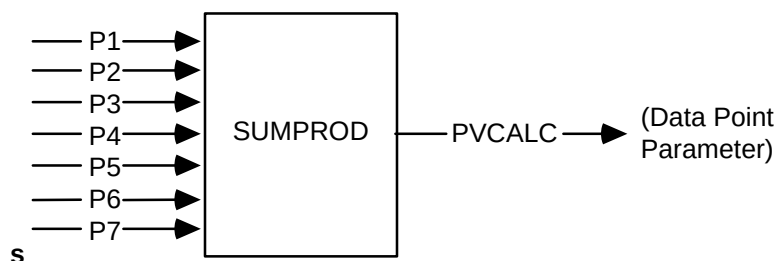
SUM OF PRODUCTS (PV) Section 11

11.1 TYPE AND NAME

PV Algorithm: SUMPROD

11.2 FUNCTION

This algorithm calculates a PV (PVCALC) that is either the sum of two 2-term products (Equation A) or the sum of two 3-term products (Equation B). The individual inputs and the whole calculation can be scaled, bias values can be added to the inputs, and a bias can be added to the whole calculation. See Figure 11-1.



Equation B, Simplified:

$$\text{PVCALC} = (\text{P1} \cdot \text{P2} \cdot \text{P3} + \text{P4} \cdot \text{P5} \cdot \text{P6}) + \text{P7}$$

Figure 11-1 — Functional Diagram, Sum of Products PV Algorithm

1315

11.3 USE

Heat-balance or mass-balance calculations can be made by using process-connected inputs received through the Hiway Gateway. Also, the inputs can be parameters from data points in the same AM or from other modules on the Local Control Network. A simple CL block could be inserted before PV-Input Processing (see Figure 2-1) to do calculations that result in a substitute for the raw PV value. This could allow this algorithm to be used for more sophisticated calculations, such as in thermodynamic equations.

11.4 OPTIONS AND SPECIAL FEATURES

11.4.1 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 very 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.

11.4.2 Error Handling

If there are no inputs with a bad status and the status of at least one input is uncertain, the PVAUTO-value status is uncertain.

If the status of at least one input is bad, the PVAUTO-value status becomes bad and PVCALC contains NaN.

11.4.3 Restart or Point Activation

On any type of restart or when this data point is activated, PVCALC is normally calculated.

11.5 EQUATIONS

You can select one of two equations when you configure a data point that uses the Sum of Products PV algorithm:

Equation A:

$$\text{PVCALC} = C * [(C1 * P1 + D1) * (C2 * P2 + D2) + (C3 * P3 + D3) * (C4 * P4 + D4)] + D$$

Equation B:

$$\text{PVCALC} = C * [(C1 * P1 + D1) * (C2 * P2 + D2) * (C3 * P3 + D3) + (C4 * P4 + D4) * (C5 * P5 + D5) * (C6 * P6 + D6)] + (C7 * P7 + D7) + D$$

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 through C7 = The scale factors for P1 through Pn. Default value = 1.0.

P1 through P7 = The PV input values. Default values are

P1 = NaN.

P2 and P3 = 1.0.

P4 through P7 = 0.

D = The overall bias. Default value = 0.

D1 through D7 = The bias for P1 through P7. Default value = 0.

Other parameters associated with the SUMPROD algorithm are as follows (refer to the *Application Module Parameter Reference Dictionary*):

PnSTS PVEQN

11.6 MIGRATION

There are no similar algorithms in SUPERVISORY/TOTAL Systems, nor in PMX Systems.

VARIABLE DEAD TIME WITH LEAD-LAG COMPENSATION (PV)

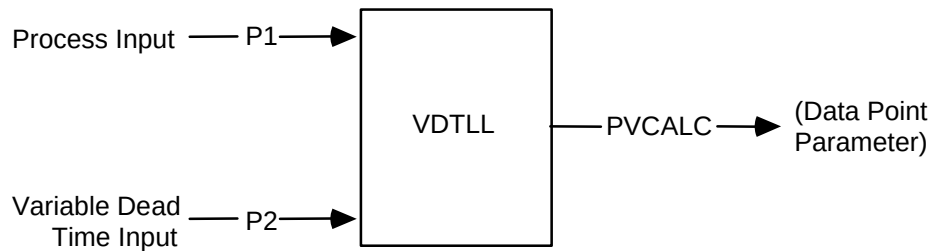
Section 12

12.1 TYPE AND NAME

PV Algorithm: VDTLL

12.2 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 12-1.



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

Figure 12-1 — Functional Diagram, Variable Dead Time with Lead Lag PV Algo

1316

12.3 USE

This algorithm is used for feedforward control and in process simulations. For additional use information, see Equations C and D, under 12.4.1.

This algorithm can be used as the PV algorithm in a data point that uses the PID Feedforward control algorithm. See Figure 19-2 in Section 19.

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.

12.4 OPTIONS AND SPECIAL FEATURES

12.4.1 Four Combinations of Delay and Lead-Lag Compensation

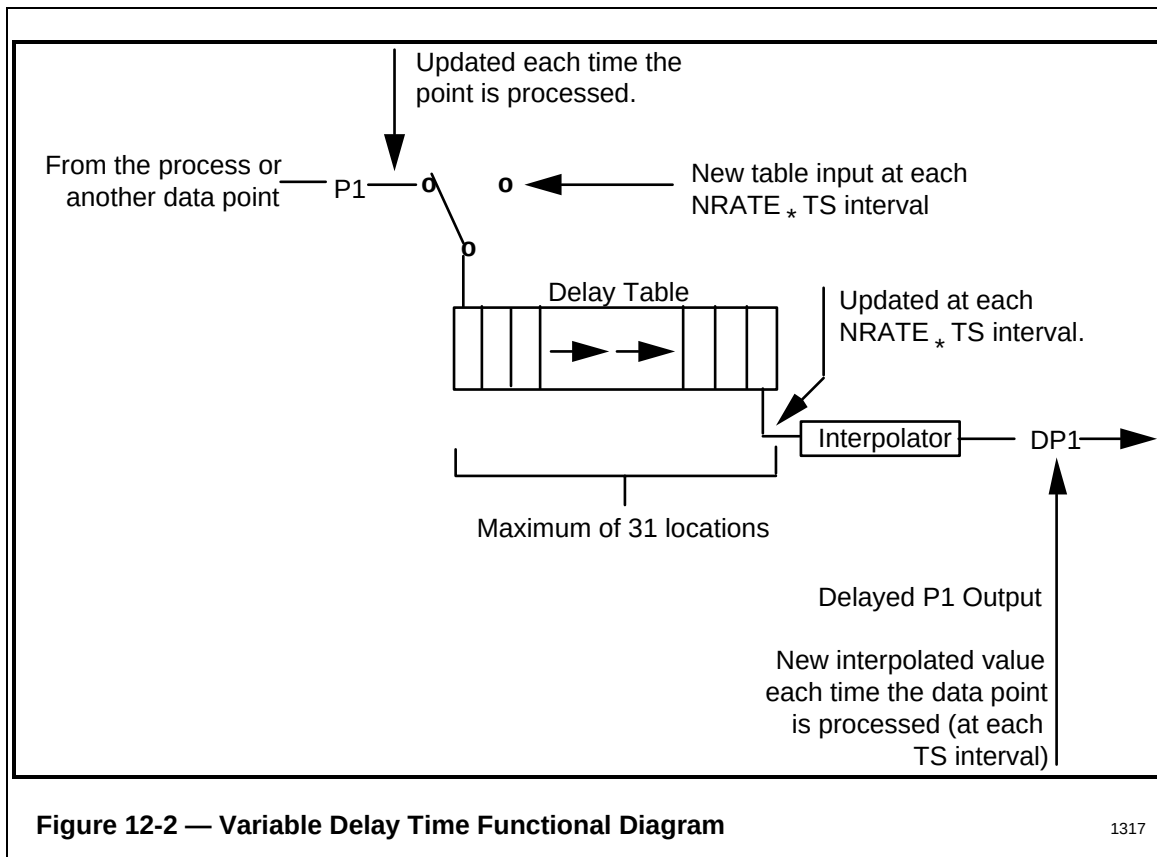
You select the combinations of delay, lead compensation, and lag compensation by selecting Equation A, Equation B, Equation C, or Equation D when configuring the data point. The equations function as follows:

- **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 TLD, 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 whose duration 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. 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 12.5.

Equation C can be used to produce a fixed delay time that can be changed while the data point is active; however, the resolution of the adjustments in delay time may be much less than is possible when using a true fixed delay, through Equation B.

- **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.



12.4.2 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 module'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 12-2.

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 half-minute (TS = 0.5 min.). 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 are provided below. 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:

$$\text{NRATE} = \text{TD} / (\text{TS} * 30) = 15 / (0.5 * 30) = 1$$

3. NLOC, the number of table locations to be used, is

$$\text{NLOC} = \text{TD} / (\text{TS} * \text{NRATE}) = 15 / (0.5 * 1) = 30 \text{ locations.}$$

4. The actual delay time is then recomputed as

$$\text{TD} = \text{NLOC} * \text{NRATE} * \text{TS} = 30 * 1 * 0.5 = 15 \text{ minutes.}$$

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*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 0.25 minutes.
2. Each time the data point is processed, a new variable delay time is calculated as

$$\text{TDNEW} = \text{C1} / (\text{C2} * \text{P2} + \text{D2}) + \text{D1.}$$

12.4.3

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.

$$TDNEW = 30.0 / (0.05 * 20.0 + 0) + 0 = 30 \text{ minutes.}$$

3. $NRATE = TDNEW / (TS * 30) = 30 / (0.25 * 30) = 4$

4. Actual delay time is calculated as:

$$TD = 30 * NRATE * TS = 30 * 4 * 0.25 = 30 \text{ minutes.}$$

Now suppose the P2 input changes to 23.4 units.

Second Variable Delay-Time Example

1. The value in TS is 0.25 minutes.

2. $TDNEW = C1 / (C2 * P2 + D2) + D1 = 30.0 / (0.05 * 23.4 + 0) + 0 = 25.641 \text{ minutes.}$

3. $NRATE = TDNEW / (TS * 30) = 25.641 / 0.25 * 30 = 3.419$

This is rounded to the nearest integer (not necessarily the next larger integer, as for a fixed delay time), so

$$NRATE = 3.$$

4. Actual delay time then is

$$TD = 30 * NRATE * TS = 30 * 3 * 0.25 = 22.5 \text{ minutes.}$$

In the Second Delay Time Example, the exact delay specified by the input was 25.641 minutes. If the actual delay is not accurate enough, the TS value can be set lower (by specifying a lower value for PERIOD) to achieve greater accuracy (resolution).

To eliminate excessive changes and jitter in the TD value that are caused by noise on the P2 input, a dead band of 10% of the minimum step in TD values is provided. If P2 doesn't change more than this value, the previous TD value is retained.

12.4.3 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.

12.4.4 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 30*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.

12.4.5 Time-Constant Recommendations

We recommend that the processing rate of a data point that uses this algorithm and Equation A or D must be a least ten times greater than the lead or lag break-point frequencies, 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 lead break-point frequency) be less than or equal to 10, so,

|TLD| should be equal to or less than 10*TLG1.

12.4.6 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.

12.4.7 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:

$$PVCALC = C * P1 + D$$

On a hot start, all calculations are resumed as if nothing had happened.

12.4.8 Error Handling

For Equation C and D, if neither input has a bad-value status, but one or both has an uncertain-value status, the PVAUTO-value status is uncertain. Equations A and B don't use the P2 input, so for them, the PVAUTO-value status is uncertain only if the P1-value status is uncertain.

For Equations C and D, if either input has a bad-value status, PVCALC becomes NaN and the PVAUTO-value status is bad. For Equations A and B, only a bad P1-value status causes PVCALC to contain NaN and the PVAUTO-value status to be bad.

When the input-value status is again normal or uncertain, the data point is initialized as for a cold start under 12.4.7, and the PVAUTO-value status becomes normal, or uncertain, as appropriate.

12.5 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

$$PVCALC(s) = \left[C \cdot \frac{1 + TLD \cdot s}{(1 + TLG1 \cdot s) \cdot (1 + TLG2 \cdot s)} \cdot P1(s) \right] + D$$

Equation B—Fixed Delay Time

$$DP1_t = P1_{t-TD}$$

$$PVCALC(s) = C \cdot DP1 + D$$

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:

$$DP1_t = P1_{t-TD}$$

$$TD = \frac{C1}{C2 \cdot P2 + D2} + D1$$

$$PVCALC(s) = C \cdot DP1 + D$$

Equation D—Variable Delay Time with Two Lag Compensations

$$TD = \frac{C1}{C2 \cdot P2 + D2} + D1$$

$$DP1_t = P1_{t-TD}$$

$$PVCALC(s) = [C \cdot \frac{1}{(1 + TLG1 \cdot s) \cdot (1 + TLG2 \cdot s)} \cdot DP1(s)] + D$$

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 = NaN.

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.

Another parameter associated with the VDTLL algorithm is as follows (refer to the *Application Module Parameter Reference Dictionary*):

PVEQN

12.6 MIGRATION

The only similar PV algorithms in SUPERVISORY/TOTAL Systems and PMX systems are delay algorithms 36 and 37. Equations B, C, and D can provide similar fixed delays. The resolution of the calculated delay is significantly better here, because the delay table in the SUPERVISORY/TOTAL and PMX Systems has a maximum of only 16 locations.

Control algorithm 20 in SUPERVISORY/TOTAL and PMX Systems provides lead-lag compensation similar to that of Equation A of this algorithm, except that algorithm 20 has only one lag-time constant.

CALCULATOR (PV) Section 13

13.1 OVERVIEW

CALCULTR is a new PV algorithm in the AM regulatory point. This new algorithm is a Release 530 enhancement. This algorithm is very similar in function to the APM regulatory point PV algorithm called “CALCULTR,” with two exceptions.

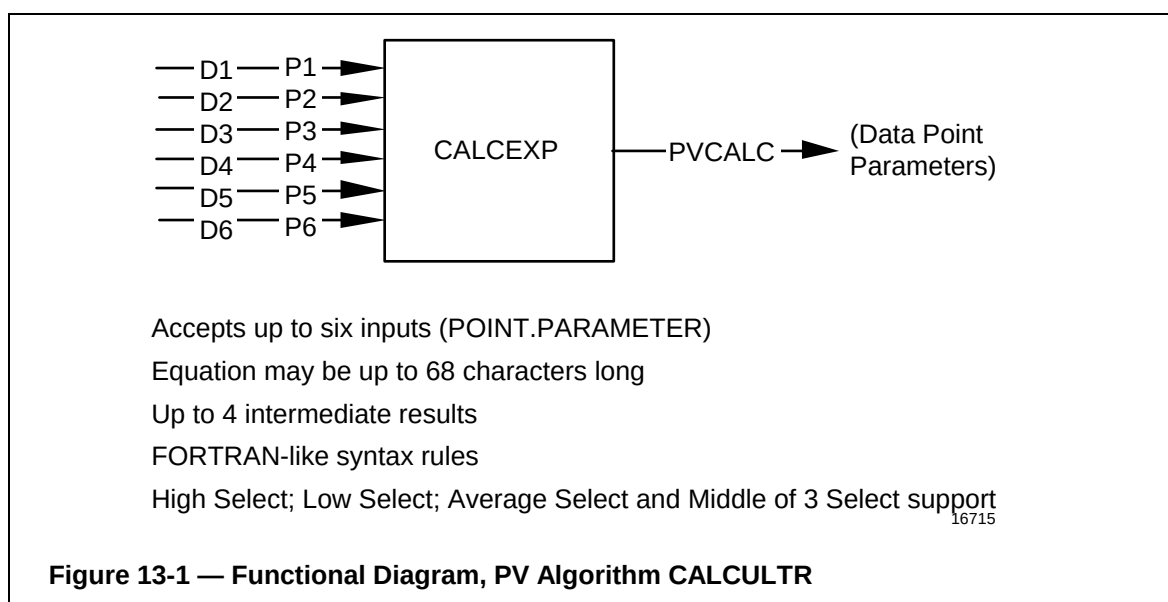
- The length of the new AM PV calculator algorithm expression is now expanded to 68 characters from the previous 40 of the APM expression.
- Also, an optional scaling factor is provided (D1-D6) for each of the six PV inputs (P1-P6) to the AM calculator expression.

13.2 TYPE AND NAME

PV Algorithm: CALCULTR

13.3 FUNCTION

The calculator algorithm allows the user to write an equation to compute the PV and up to four intermediate results can be calculated. The result from evaluating the expression is stored into PVCALC, which is then processed like any other calculated PV value (see the following figure).



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 sources (PISRC(n)) can be configured and stored into the destination parameters P1 . . . P6.

Up to six scaling factor parameters (D1-D6) are also provided, one for each of the input destination parameters, P1 . . . P6.

The following general guidelines apply.

- The equation can be up to 68 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.

13.3 .1 Calculation and Arithmetic Functions Supported

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 selector functions are supported:

Function	Associated Symbol
High Select	MAX
Low Select	MIN
Average Select	AVG
Middle of 3	MID3

Other statements:

Function	Associated Symbol
Equate	=
Separator	;

13.4 USE

The calculator algorithm can be used to perform any calculation or arithmetic function on up to six inputs, using up to four intermediate results.

13.5 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 . . . 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 C1-C4 are not expressions, then they can be used as configurable constants into the equation. C1-C4 can then be entered or modified by an operator through the detail displays or user programs.
- Parameters D1-D6 are scaling factors for each of the six PV Inputs, P1-P6. The default value for D1-D6 is 1.0.

13.5.1 Calculator Expression Errors

The calculator expression is compiled during the Data Owner Store into the algorithm expression string parameter, CALCEXP. In case of any syntax error(s) or if a parameter referenced in the expression is misspelled, the store is rejected and the error "CALCULATOR SYNTAX" is given to the user.

13.5.2 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.

13.6 EQUATIONS

The calculator equation can be up to 68 characters long. It is entered into the Parameter Entry Display (PED) in the port for the parameter CALCEXP. You can also change the equation from a schematic or User CL program, but the AM Regulatory point must be INACTIVE. 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
```


USER-WRITTEN CL BLOCK (PV) Section 14

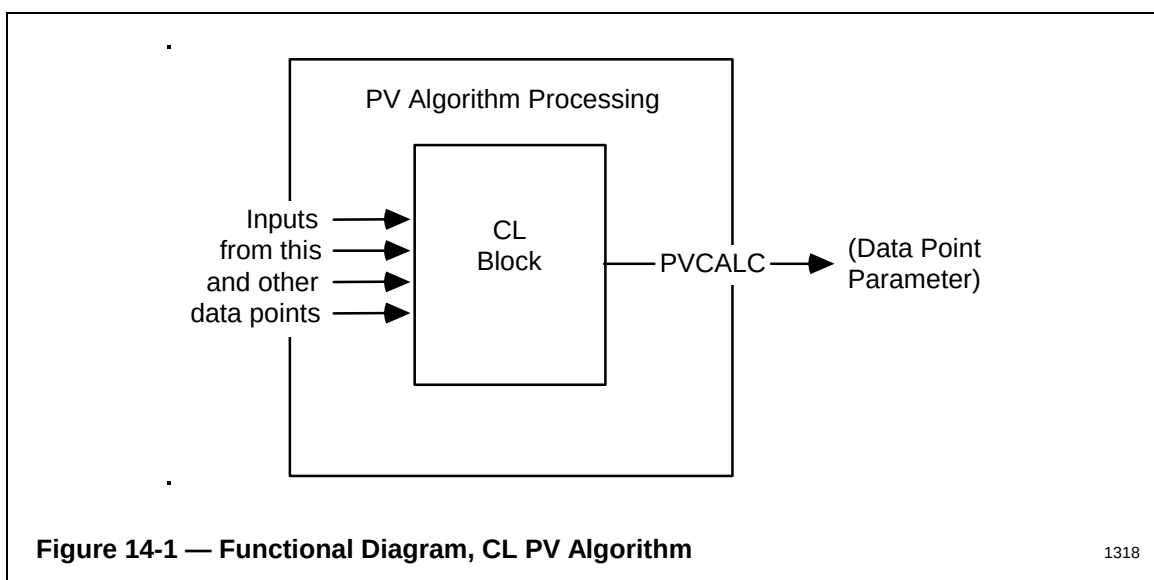
14.1 TYPE AND NAME

PV Algorithm: CL

14.2 FUNCTION

This algorithm is a user-written CL block that is like any other CL block except that it is inserted at the PV-algorithm insertion point in the processing sequence (see Figure 2-1) and is executed instead of one of the standard PV algorithms.

The CL block must calculate and store a value in PVCALC. Inputs to the CL block are usually acquired by direct references in CL, but they can also be acquired through general inputs to a Custom Data Segment (CDS) that is included in the data point. The value placed in PVCALC by the CL block is processed just as PVCALC is processed for any other data point that uses a PV algorithm.



14.3 USE

A CL block is used when normal point processing is appropriate but none of the standard PV algorithms will accomplish the desired function. The CL block can be bound to a single data point if only one point needs its functions or it can be written as a generic CL block and bound to several data points.

Here are two examples of the tasks that should be accomplished by the CL block:

- A Simple Operation—No Storage of Intermediate Values, No Dynamic Calculations
 1. Compute the algorithm output value and store it in PVCALC. This sets PVAUTOST to Normal or Bad, based on the value stored.
 2. If propagation of uncertain status is needed, check the status of the inputs used for the calculation to determine the status of the value in PVCALC (Normal, Uncertain, or Bad) and store that status in PVAUTOST.
- An Operation that Includes Storage of Intermediate Values, or Dynamic Calculations, or Both
 1. Compute the value and store it in PVCALC as follows:

If the status of the required input is Bad,

Set PVCALC to NaN (PVAUTO goes Bad automatically)

Else if (PVINIT = On) {*initialization is requested*}

OR (PVAUTO is NaN) {*this is a recovery from a bad PV*}

Compute PVCALC, using the CL block's initialization equation (PVAUTOST is automatically set to Normal).

Else

Compute PVCALC, using the CL block's normal equation.
 2. If propagation of uncertain status is needed, check the status of the inputs to be used for the calculation to determine the status of the value to be placed in PVCALC (Normal, Uncertain, or Bad) and store that status in PVAUTOST.

14.4 OPTIONS AND SPECIAL FEATURES

14.4.1 Initialization

No standard initialization is provided. The CL block can monitor the PVINIT parameter to determine whether to initialize or not.

14.4.2 Restart

The CL block can check the data point's RESTART parameter to determine what type of restart, if any, the point is going through, and then take appropriate action.

14.4.3 Processing Schedule and Execution Time

The CL block should not be used to perform long, complex operations because there may not be enough time in normal point processing to complete such operations. Points that use a CL PV algorithm should be scheduled at the longest reasonable interval, and should be assigned to the Slow Processor, if possible. If execution of the CL block takes too much time, it is aborted and an alarm is generated.

Lengthy, complex operations and calculations should be accomplished in Fortran or Pascal programs in a CM60.

14.4.4 Parameters Used for Comparisons

The value status of any parameters used for comparisons should be checked before the CL block does the comparison. A comparison done with a parameter that has a bad value causes the CL block to abort.

14.4.5 Error Handling

If a Bad value is used to determine the value to be stored in PVCALC, the PVCALC status, which is indicated by PVAUTOST, must be set to Bad. CLs Allow Bad Value function must be used to store a bad value in PVCALC.

14.5 EQUATIONS

The equation(s) used for a CL PV algorithm, if any, is a function of the CL block.

Another parameter associated with the CL PV algorithm is as follows (refer to the *Application Module Parameter Reference Dictionary*):

PVINIT

14.6 MIGRATION

PV algorithm 77 in SUPERVISORY/TOTAL and PMX systems has functions that are similar to the CL PV algorithm. The following table lists those functions.

Table 14-1 — Comparison of SUPERVISORY/TOTAL and PMX Algorithms with CL PV

Function	S/T	PMX	CL PV
Input access	Destination words or explicit	Explicit only	General inputs to Custom Data Segments or explicit references.
Accessible parameters	Defined list	Defined list	Same as for all user programs
Enumerations	No	No Yes	
Data type checking?	No	No Yes.	User defines data types.
Initialization	Yes	Yes	Yes

AUTO MANUAL STATION (CONTROL) Section 15

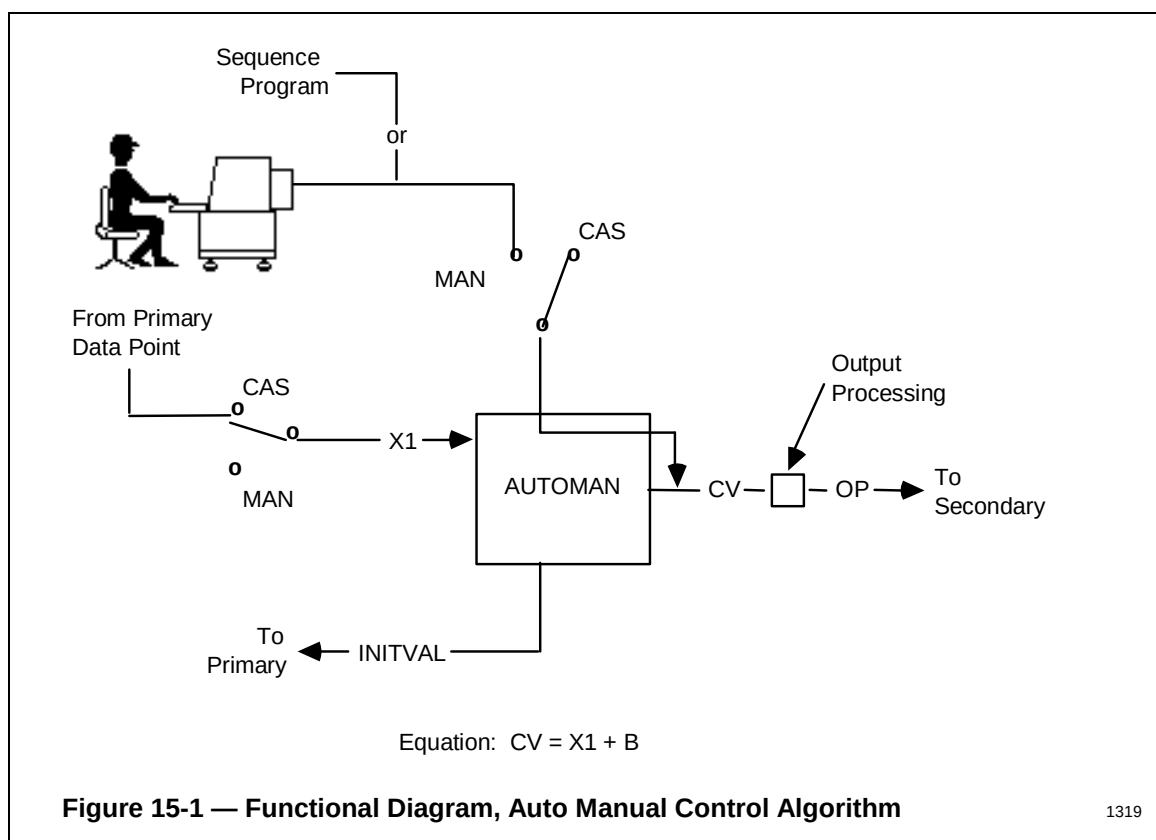
15.1 TYPE AND NAME

Control algorithm: AUTOMAN

15.2 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 the following figure.

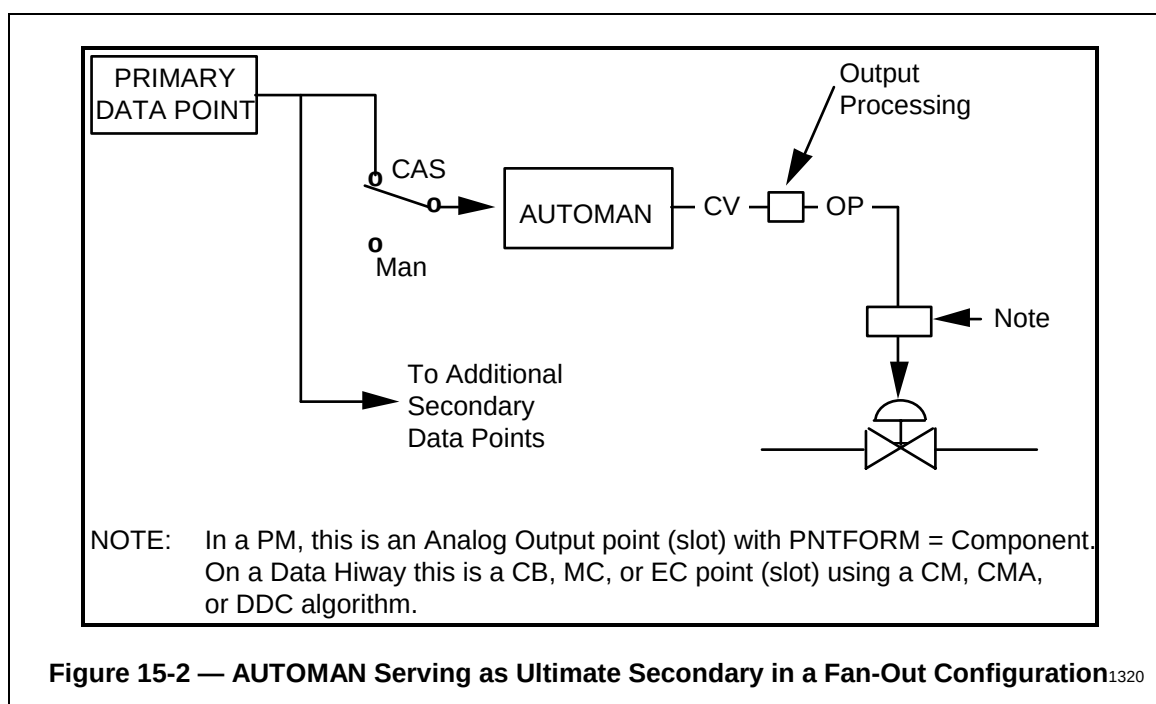
The algorithm is designed to provide "bumpless" returns to cascade operation, if it is configured for external initialization, even though its primary data point may not accept the initialization value from the AUTOMAN data point. When it is configured for internal initialization, during initialization, the bias value is back-calculated to provide an unchanged output when the data point returns to cascade operation.



15.3 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 a PM slot or a Data Hiway-based slot, as shown on the following figure.

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.



15.4 OPTIONS AND SPECIAL FEATURES

15.4.1 "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.

$$B = B0 + B1$$

- └ Internal adjustment-value storage. Usually B1 = 0.
- └ Internal storage for operator-entered or program-entered bias value.

15.4.1

When the user or a program enters a bias value in B while in cascade operation, that value goes to B0 and B1 is forced to zero. B1 is an internal parameter that cannot be accessed by a user.

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:

$$\text{INITVAL} = \text{CV} - \text{B0}$$

When cascade operation resumes, the value in B1 is calculated as follows:

$$\text{B1} = \text{INITVAL} - \text{OPprim}$$

└────────── The actual output value from the primary data point.

If the primary data point did accept AUTOMAN's initialization value, B1 turns out to contain zero. If the primary did not accept INITVAL, B1 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 B1 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-second. 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, configured for external initialization, is operating in CAS mode

$$\text{X1} = 6$$

$$\text{B} = 5; \text{B1} = 0, \text{B0} = 5$$

$$\text{CV} = \text{X1} + \text{B} = 11$$

$$\text{RATE1} = 2 \text{ units per minute}$$

The mode is changed to MANual

$$\text{INITVAL} = \text{CV} - \text{B0} = 11 - 5 = 6$$

└────────── The operator changed the output value.

$$\text{B} = 5; \text{B0} = 5, \text{B1} = 5$$

The primary does initialize

$$\text{OPprim} = 7$$

Go back to CAScade mode

$$x1 = /$$

$$B1 = 7 - 7 = 0$$

$$B = B0 + B1 = 5 + 0 = 5$$

$$CV = 7 - 5 = 12$$

└── Same value as when in MAN mode

Go to MAN mode again

$$x1 = 9$$

$$B1 = \text{INITVAL} - \text{OPprim} = 7 - 9 = -2$$

$$B = 5 - 2 = 3$$

$$CV = 9 + 3 = 12$$

└── Same as in MAN mode

This time, the primary doesn't accept INITVAL, and the output of the primary went to nine units

$$\text{OPprim} = 9$$

Go back to CAScade

$$x1 = 9$$

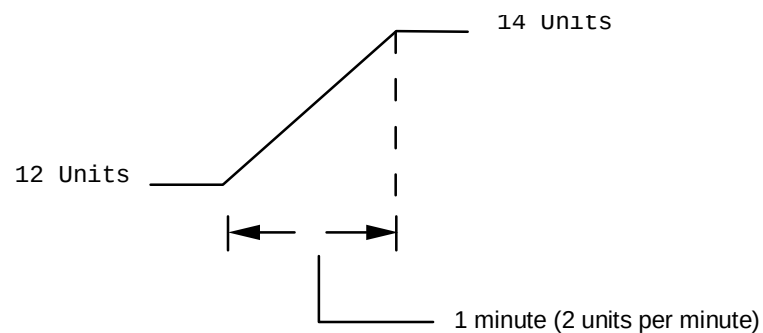
$$B1 = \text{INITVAL} - \text{OPprim} = 7 - 9 = -2$$

$$B = 5 - 2 = 3$$

$$CV = 9 + 3 = 12$$

└── Same as in MAN mode

Now, because B1 ramps from -2 units to zero, CV changes as follows:



15.4.2 Operating Modes

This algorithm operates only in MANual or CAScade modes.

15.4.3 Input Value Range

You must configure the X1 input range in XEULO and XEUHI.

15.4.4 Restart or Point Activation

On a cold or warm restart or when the data point is activated, initialization takes place as described under 15.6. On a hot restart, initialization does not occur.

15.4.5 Error Handling

If the X1 input has a bad-value status, 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, if configured for external initialization, an initialization request is sent to the primary.

15.5 EQUATIONS

This algorithm uses only one equation for normal operation:

$$CV = X1 + B$$

Where:

CV = Control-algorithm output in engineering units.

B = Bias value. Default = 0.

X1 = The initializable input.

15.6 INITIALIZATION

You can configure this algorithm for

- No initialization
- Internal initialization
- External initialization

If you choose no initialization, initialization requests are not sent to the primary data point and any initialization request from a secondary data point is ignored.

If you choose internal initialization, when the data point is initialized, the bias value is back calculated as follows:

$$B = CV - X1$$

If you choose external initialization, initialization takes place as described under 15.4.1

15.7 OVERRIDE FEEDBACK PROCESSING

When the data point's secondary uses an Override Selector algorithm (see Figure 15-2), the following functions take place:

- If this point is configured for external initialization and 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:

$$ORFB = ORFBSEC - B$$

└──────── Feedback value from the secondary

For more detail on override control see Section 23 of this publication, and 3.1.11 in the *AM Control Functions* manual.

15.8 AUTOMAN PARAMETERS

In addition to the parameters already mentioned, parameter X1STS is associated with the AUTOMAN algorithm (refer to the *Application Module Parameter Reference Dictionary*).

15.9 MIGRATION

The AUTOMAN algorithm is similar to control algorithm 25 in SUPERVISORY/TOTAL and PMX Systems. The equations compare as follows:

AM Terms	S/I-PMX Terms
$CV = X1 + B$	$C_N = X + B$

In SUPERVISORY/TOTAL and PMX Systems, algorithm 25 operates in AUTO and MAN modes, while the Auto Manual algorithm in Application Modules operates in CAS and MAN modes.

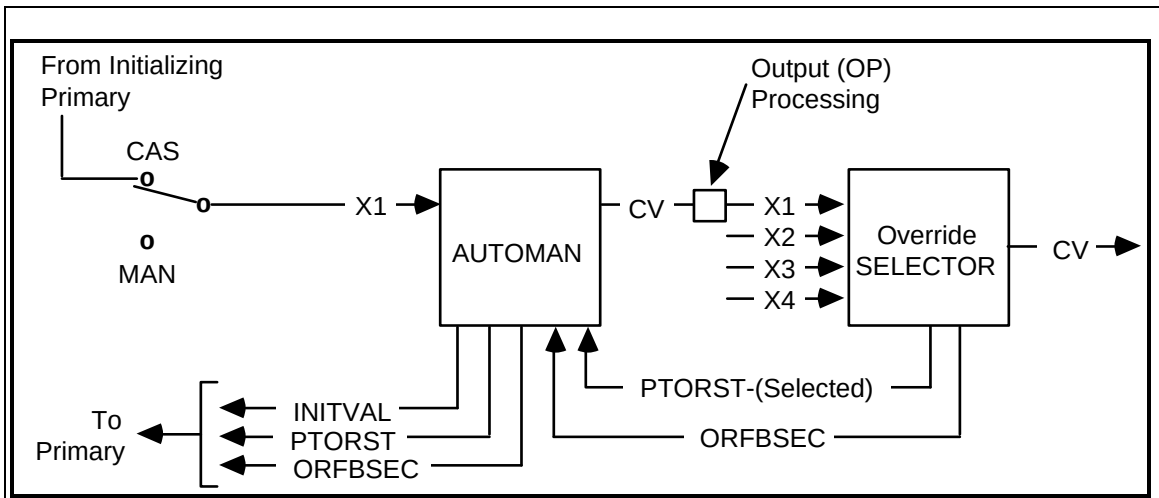


Figure 15-3 — Override Feedback Processing

1321

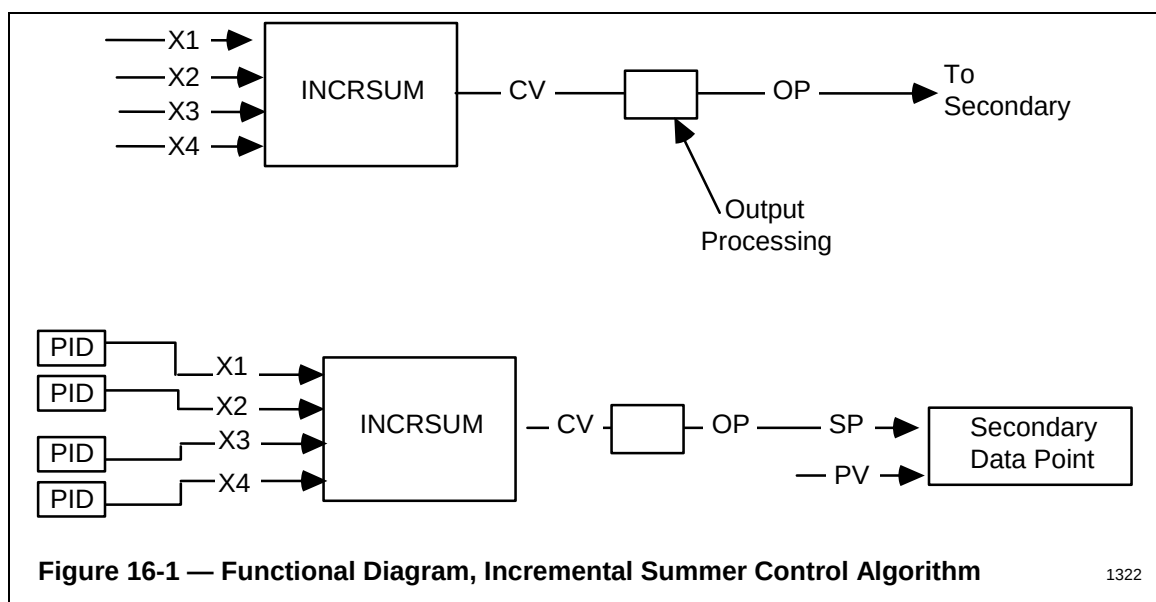
INCREMENTAL SUMMER (CONTROL) Section 16

16.1 TYPE AND NAME

Control Algorithm: INCRSUM

16.2 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 the following figure.



16.3 USE

This algorithm is typically used where more than one primary data point is used to manipulate the set point of the same secondary data point. The primaries usually use PID algorithms, and are connected to an INCRSUM data point, whose output is connected to the secondary, as shown here. This is sometimes referred to as a "fan-in connection."

16.4 OPTIONS AND SPECIAL FEATURES

16.4.1 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 process unit and the same AM.
- Each must be a normal PID (Section 18) or a PID Feedforward that uses additive feedforward (Section 19).

16.4.2 Input Value Range

You must configure the X1 input range in XEULO and XEUHI.

16.4.3 Changes to Incremental Summer Output by User-Written Programs

User-written CL programs 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 CAS mode 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 points are processed.

16.4.4 Override Control Strategy and Past-Value Updating

If the Incremental Summer's secondary is an Override Selector data point, the Incremental Summer's full-value output (CV) must be limited to a user-specified range beyond the feedback value received from the Selector point when the Incremental Summer is not selected. The feedback value is received in parameter ORFBSEC and you specify the overrange (or bias) in ORBIAS.

Because the Incremental Summer operates only on changes in value, each time it is processed it must store past values. These past values $[X_n(n-1)]$ are used to calculate the changes in values the next time the Incremental Summer point is processed.

The flow chart of Figure 16-2 shows how the override strategy and past-value updating operate. Also see 16.7.

16.4.5 Operating Modes

This algorithm operates only in MANual and CAScade modes. 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.

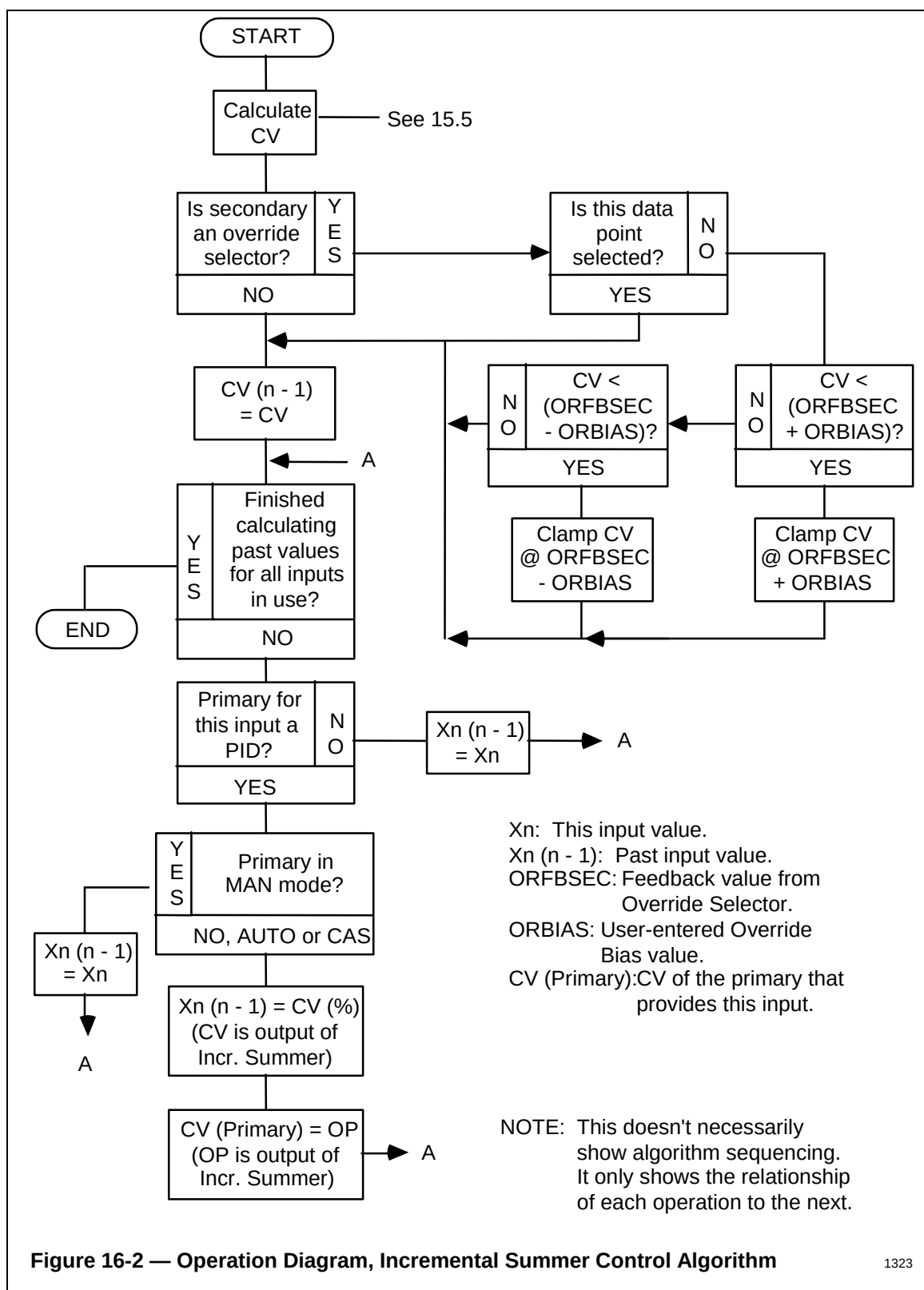
Note: The dynamic correction calculated by a PID primary appears as a change in that data point's output after it is processed again, and the Incremental Summer actually uses this change as its input. The full-value outputs of PIDs connected to the Incremental Summer have no meaning.

16.4.6 Restart or Point Activation

On a cold restart, a warm restart, or point activation, if the Incremental Summer is configured for initialization, it takes place as described under 16.6. On a hot restart, the inputs and past values become the present values and no initialization requests are sent to the primary data points.

16.4.7 Error Handling

If the status for an input value is bad, the input value is not used in the CV calculation and the point remains in the same mode. When a bad input returns to normal, that input is again used in the CV calculation without "bumping" the value.



16.5 EQUATIONS

CV is calculated as follows:

$$\begin{aligned}
 CV(n) = & CV(n-1) + K1 * [X1(n) - X1(n-1)] \\
 & + K2 * [X2(n) - X2(n-1)] \\
 & \vdots \\
 & + Km * [Xm(n) - Xm(n-1)]
 \end{aligned}$$

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 default = 1.0. K2 through K4 default to 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).

16.6 INITIALIZATION

You can configure this algorithm for

- Internal initialization.
- External initialization.

If you choose internal initialization, initialization requests are not sent to any of the primaries. In MAN mode, when the output is indisponible or when recovering from a bad CV value (NaN), initialization is as follows:

CV is back-calculated from OP

if the output is disposable, or

CV = the initialization value returned from the secondary data point, if the output is indisponible.

The past values are replaced as follows:

$$\begin{aligned} CV(n-1) &= CV \\ X1(n-1) &= X1 \\ &\cdot \\ &\cdot \\ &\cdot \\ X_m(n-1) &= X_m \end{aligned}$$

If you choose external 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 as shown above for internal initialization.
- The initialization value sent to the primary or primaries is equal to the value in the Incremental Summer's CV parameter.

16.7 OVERRIDE FEEDBACK PROCESSING

When this data point's secondary uses an Override Selector algorithm, the Incremental Summer doesn't send override-selection status or values to its primaries. It does, however, receive selected or not-selected status from the secondary, and its CV value is limited as described under 14.4.4 and on Figure 14-2.

16.8 INCRSUM PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the INCRSUM algorithm (refer to the *Application Module Parameter Reference Dictionary*):

INITTYPE	X3STS
X1STS	X4STS
X2STS	

16.9 MIGRATION

There are no similar algorithms in SUPERVISORY/TOTAL and PMX Systems. Algorithm 32, Incremental, in the Extended Controller is similar to the Incremental Summer control algorithm.

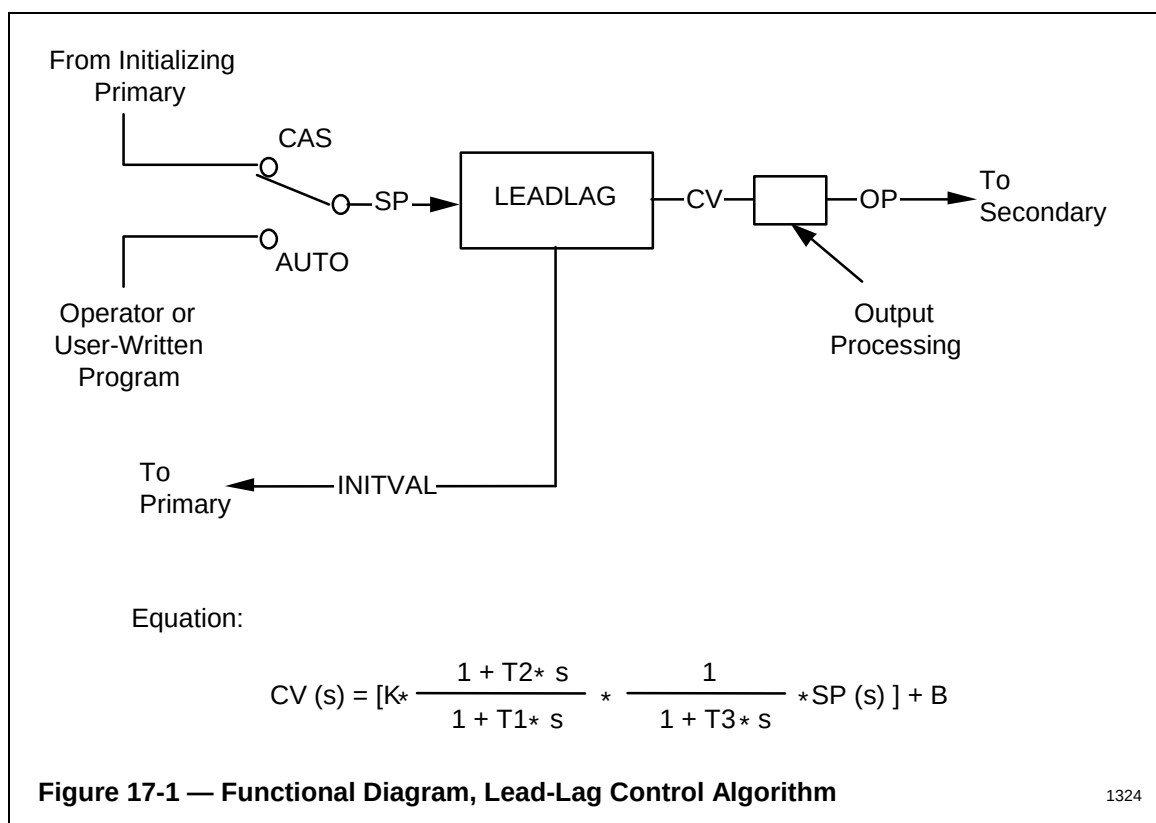
LEAD-LAG (CONTROL) Section 17

17.1 TYPE AND NAME

Control Algorithm: LEADLAG

17.2 FUNCTION

This algorithm provides dynamic lead and lag compensation to a feed forward signal. A scale factor can be applied to the input and a bias value can be added.



17.3 USE

This algorithm is used to provide dynamic lead and lag compensation to a feed forward variable from a process-connected input or from another data point. The algorithm's output is often connected to the feed forward input of a PID-with-Feed forward control algorithm. It can also be connected to an input of other types of data points such as an incremental Summer, a Multiplier/Divider control algorithm, a Summer control algorithm, or an Auto Manual Station algorithm. The algorithm can also be used to smooth out changes in the setpoint of another data point by an operator at a Universal Station.

17.4 OPTIONS AND SPECIAL FEATURES

17.4.1 Operating Modes

This algorithm operates in the following modes:

- AUTO
- MAN
- CAS

17.4.2 Eliminating a Lead or Lag Term

Setting the lead-time constant (T_2) to 0, eliminates the contribution of its term from the compensation. Likewise, a value of 0 for either of the lag-time constants (T_1 and T_3) eliminates their terms from the compensation.

17.4.3 Time-Constant Recommendations

We recommend that the processing rate of a data point that uses this algorithm be at least ten times greater than the lead or lag break-point frequencies, so

- $|T_2|$ should be equal to or greater than $10 \cdot TS$.
- T_1 should be equal to or greater than $2 \cdot TS$.
- T_3 should be equal to or greater than $2 \cdot TS$.

Because negative values for lead time are possible, its restrictions are specified in terms of its absolute value ($|T_2|$).

TS is the data-point processing-interval in minutes.

We recommend that the rate amplitude be less than or equal to ten, so

- $|T_2|$ should be less than or equal to $10 \cdot T_1$.

17.4.4 Restart or Point Activation

On a cold restart, warm restart, or when the point is activated, initialization is performed, as configured. No initialization occurs on a hot restart.

17.4.5 SP Value Range

You must configure the range of the SP value in SPEULO and SPEUHI.

17.5 EQUATIONS

Only one equation is available. It is as follows:

Equation A;

$$CV(s) = \left[K \frac{1 + T2*s}{1 + T1*s} * \frac{1}{1 + T3*s} * SP(s) \right] + B$$

Where:

CV = Normal control output in engineering units.

B = Bias value. Default value = 0.

K = Scale factor (gain). Default value = 1.0.

SP = Input value (initializable).

s = The Laplace operator.

T2 = Lead-time constant in minutes. Default = 0.

T1 = 1st lag-time constant in minutes. Default = 0.

T3 = 2nd lag-time constant in minutes. Default = 0.

17.6 INITIALIZATION

You can configure this algorithm for

- No initialization
- Internal initialization
- External initialization

If no initialization is chosen, normal processing continues if a condition should occur that would otherwise cause initialization.

If internal initialization is chosen, an initializing condition causes the bias value (B) to be calculated so that on a return to CAS or AUTO mode the output does not move. The initialization equation is

$$B = CV - K*SP$$

If external initialization is chosen, an initializing condition causes an initialization request to be sent to the primary, and the value of SP needed to maintain CV at the present value is calculated and passed to the primary in INITVAL. The initialization equation is

$$SP = \frac{CV - B}{K}$$

17.7 OVERRIDE FEEDBACK PROCESSING

This algorithm doesn't participate in override-feedback processing. It should not be used in an override-control strategy.

17.8 LEADLAG PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the LEADLAG algorithm (refer to the *Application Module Parameter Reference Dictionary*):

INITTYPE	SPEUHI
SPEULO	SPSTS

17.9 MIGRATION

Control algorithm 20 in SUPERVISORY/TOTAL and PMX Systems is similar to this algorithm, except that algorithm 20 has only one lag term. LEADLAG has normal initialization options, while algorithm 20 does prevent bumps when the input and output don't match, but the output decays toward the steady-state value.

MULTIPLIER/DIVIDER (CONTROL) Section 18

18.1 TYPE AND NAME

Control Algorithm: MULDIV

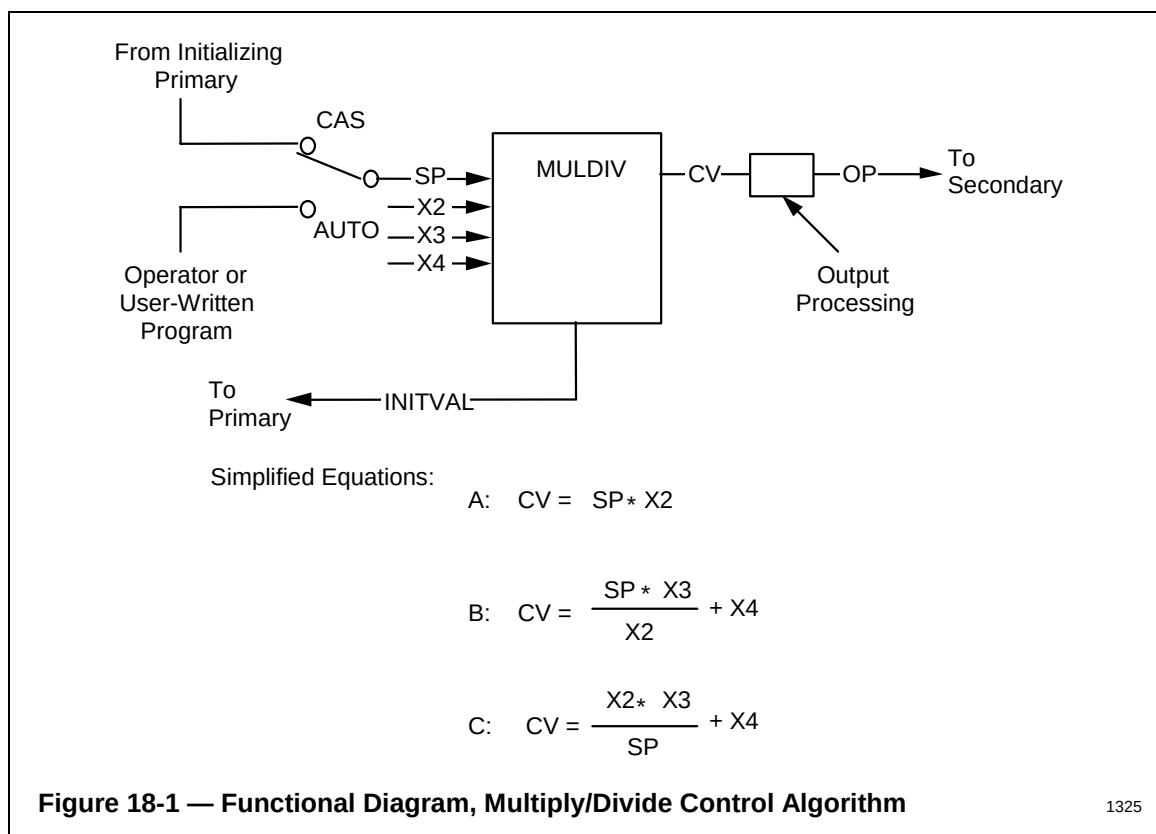
18.2 FUNCTION

This algorithm calculates a control output by multiplying two input variables and dividing the resulting product by a third input variable. Scale factors and bias can be applied to the input variables, and an overall scale factor and bias can be specified.

Three equations are provided. One provides only multiplication, and the other two provide multiplication and division.

One of the input variables, SP, is initializable. This variable appears in the numerator of one of the fractions and in the denominator of the other.

This algorithm is similar to PV algorithm MULDIV.



Input SP is initializable in all the equations. The only difference between equations B and C is that SP is in the numerator in B and it is in the denominator in equation C.

18.3 USE

This algorithm is normally part of a cascade-control strategy. Typically, its primary is a data point that is using a PID algorithm, and its secondary is another data point that is also using a PID algorithm. The Multiply/Divide algorithm can also provide an input to an Override Selector algorithm.

18.4 OPTIONS AND SPECIAL FEATURES

18.4.1 Operating Modes

This algorithm can operate in the following modes:

- MAN
- AUTO
- CAS

18.4.2 SP Value Range

You must configure the SP-value range in SPEULO and SPEUHI.

18.4.3 Restart or Point Activation

On a cold restart, a warm restart, and then the point is activated, initialization occurs, as configured. On a hot restart, no initialization occurs.

18.4.4 Error Handling

When any Xn input is bad, the CV value becomes bad (NaN) and this point continues in the same mode. When the bad input is again normal, CV is initialized, and if this point is configured for external initialization, an initialization request is sent to its primary.

18.5 EQUATIONS

You can select from one of three equations when you build a data point that uses this algorithm. The full equations are as follows:

Equation A

$$CV = [K * (K1 * SP + B1) * (K2 * X2 + B2)] + B$$

Equation B

$$CV = K * \frac{(K1 * SP + B1) * (K3 * X3 + B3)}{(K2 * X2 + B2)} + (K4 * X4 + B4) + B$$

Equation C

$$CV = K * \frac{(K2 * X2 + B2) * (K3 * X3 + B3)}{(K1 * SP + B1)} + (K4 * X4 + B4) + B$$

Where:

CV = Normal control-algorithm output in engineering units

K = Overall gain (scale factor). Default = 1.0.

K1, K2, K3, K4 = Gain (scale factor) constants. K1, K2, and K3 default to 1.0. K4 default = 0.

B = Overall bias. Default = 0.

B1, B2, B3, B4 = Bias constants. Default = 0.

SP = The initializable input value.

X2, X3, X4 = Control input values. Typically received through control input connections. X2 and X3 default to 1.0. X4 default = 0.

18.6 INITIALIZATION

You can configure the data point that is using the Multiply/Divide control algorithm for

- No initialization
- Internal initialization
- External initialization

If you select no initialization, initialization requests from secondaries are ignored and CV is as calculated by the normal equation (see 18.5).

If you select internal initialization, initialization requests are not sent to the primary data-point when an initializing condition occurs, but bias-value B is back-calculated as follows:

$$\text{Equation A} \quad B = CV - [K * (K1 * SP + B1) * (K2 * X2 + B2)]$$

$$\text{Equation B} \quad B = CV - [K * \frac{(K1 * SP + B1) * (K3 * X3 + B3)}{(K2 * X2 + B2)} + (K4 * X4 + B4)]$$

$$\text{Equation C} \quad B = CV - [K * \frac{(K2 * X2 + B2) * (K3 * X3 + B3)}{(K1 * SP + B1)} + (K4 * X4 + B4)]$$

If you select external initialization, an initializing condition causes INITVAL to be calculated as follows, and that value and an initialization request are sent to the primary data point:

$$\begin{aligned}
 \text{Equation A} \quad \text{INITVAL} &= \left[\frac{(CV - B)}{K*(K2*X2+B2)} - B1 \right] * \frac{1}{K1} \\
 \text{Equation B} \quad \text{INITVAL} &= \left[\frac{(CV - (K4*X4+B4) - B)*(K2*X2+B2)}{K*(K3*X3+B3)} - B1 \right] * \frac{1}{K1} \\
 \text{Equation C} \quad \text{INITVAL} &= \left[\frac{K*(K2*X2+B2) - B)*(K3*X3+B3)}{CV - (K4*X4+B4) - B} - B1 \right] * \frac{1}{K1}
 \end{aligned}$$

18.7 OVERRIDE FEEDBACK PROCESSING

If this data point's secondary is an Override Selector point (see Figure 18-2) and if this point is configured for initialization and 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. The feedback value is calculated as follows:

Equation A

$$\text{ORFB} = \left[\frac{\text{ORFBSEC} - B}{K*(K2*X2 + B2)} - B1 \right] * \frac{1}{K1}$$

Equation B

$$\text{ORFB} = \left[\frac{(\text{ORFBSEC} - (K4*X4 + B4) - B)*(K2*X2 + B2)}{K*(K3*X3 + B3)} - B1 \right] * \frac{1}{K1}$$

Equation C

$$\text{ORFB} = \left[\frac{K*(K2*X2 + B2)*(K3*X3 + B3)}{\{\text{ORFBSEC} - (K4*X4 + B4) - B\}} - B1 \right] * \frac{1}{K1}$$

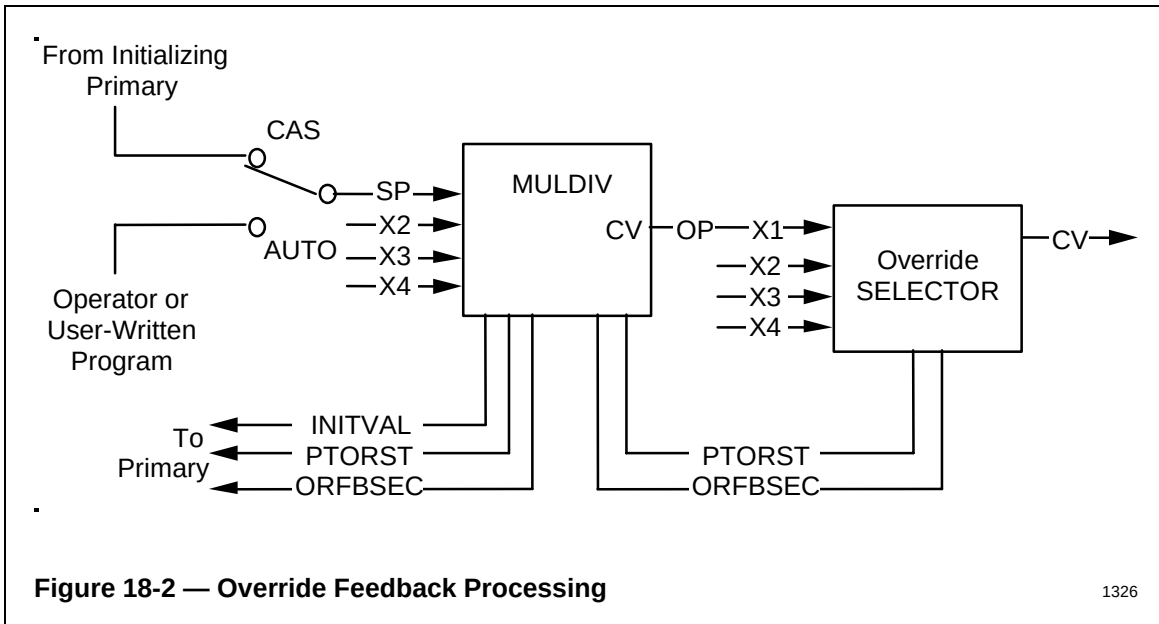
18.8 MULDIV PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the MULDIV algorithm. Refer to the *Application Module Parameter Reference Dictionary*.

CTLEQN	SPEUHI
INITTYPE	SPSTS
SPEULO	XnSTS

18.9 MIGRATION

Table 18-1 compares the AM Multiply/Divide control algorithm with similar algorithms in the former systems.



1326

Table 18-1 — Comparison of SUPERVISORY/TOTAL and PMX Algorithms with MULDIV

Nearest AM Equation	S-T/PMX Algorithm Number	S-T/PMX Equation in AM Terms	S-T/PMX Equation as shown in S-T/PMX Pubs.
A	31	$CV = K \cdot X2 \cdot SP + B$	$CN = KA \cdot X \cdot V + K2$
B	32	$CV = K \cdot SP / X2 + B$	$CN = KA \cdot V / X + K2$
A	52	$CV = X2 \cdot SP + B$	$CN = PV \cdot P + KA$
C	53	$CV = \frac{X2}{SP} + B$	$CN = \frac{PV}{P} + KA$
A	54	$CV = SP \cdot X2 + B$	$CN = R \cdot PVN + KA$

PID (CONTROL) Section 19

19.1 TYPE AND NAME

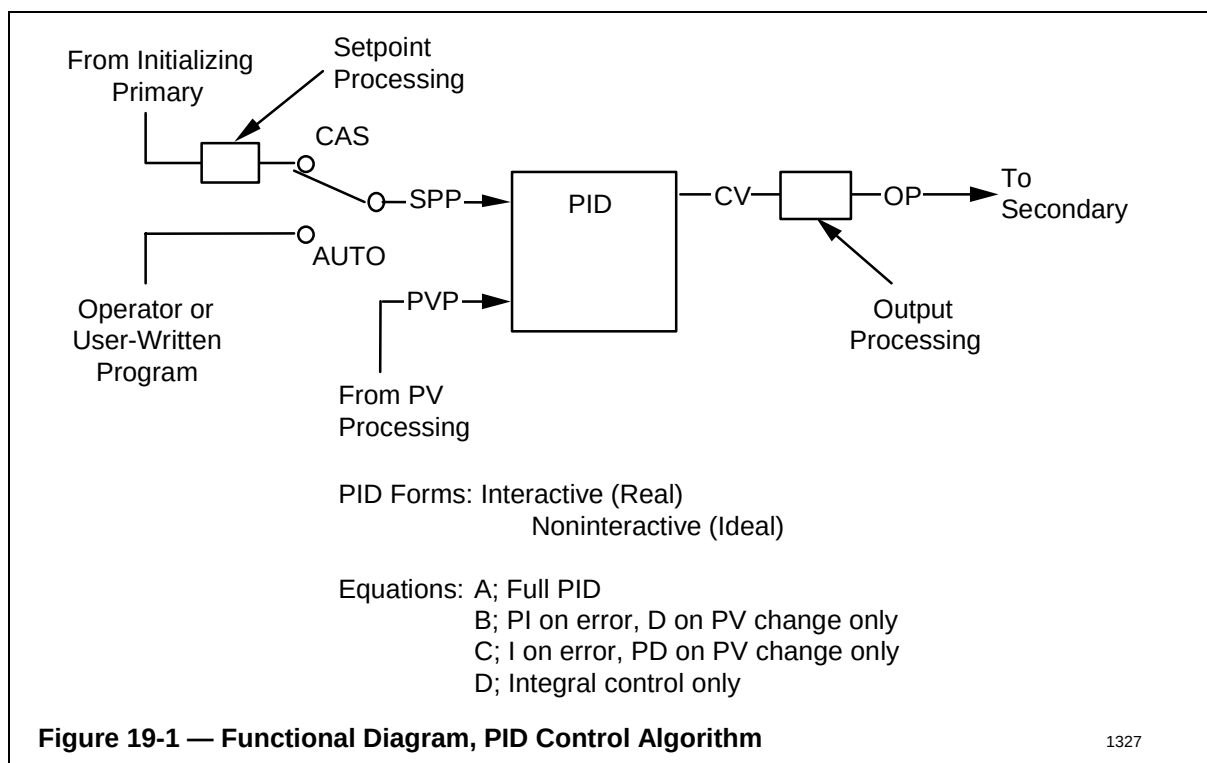
Control Algorithm: PID

19.2 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.

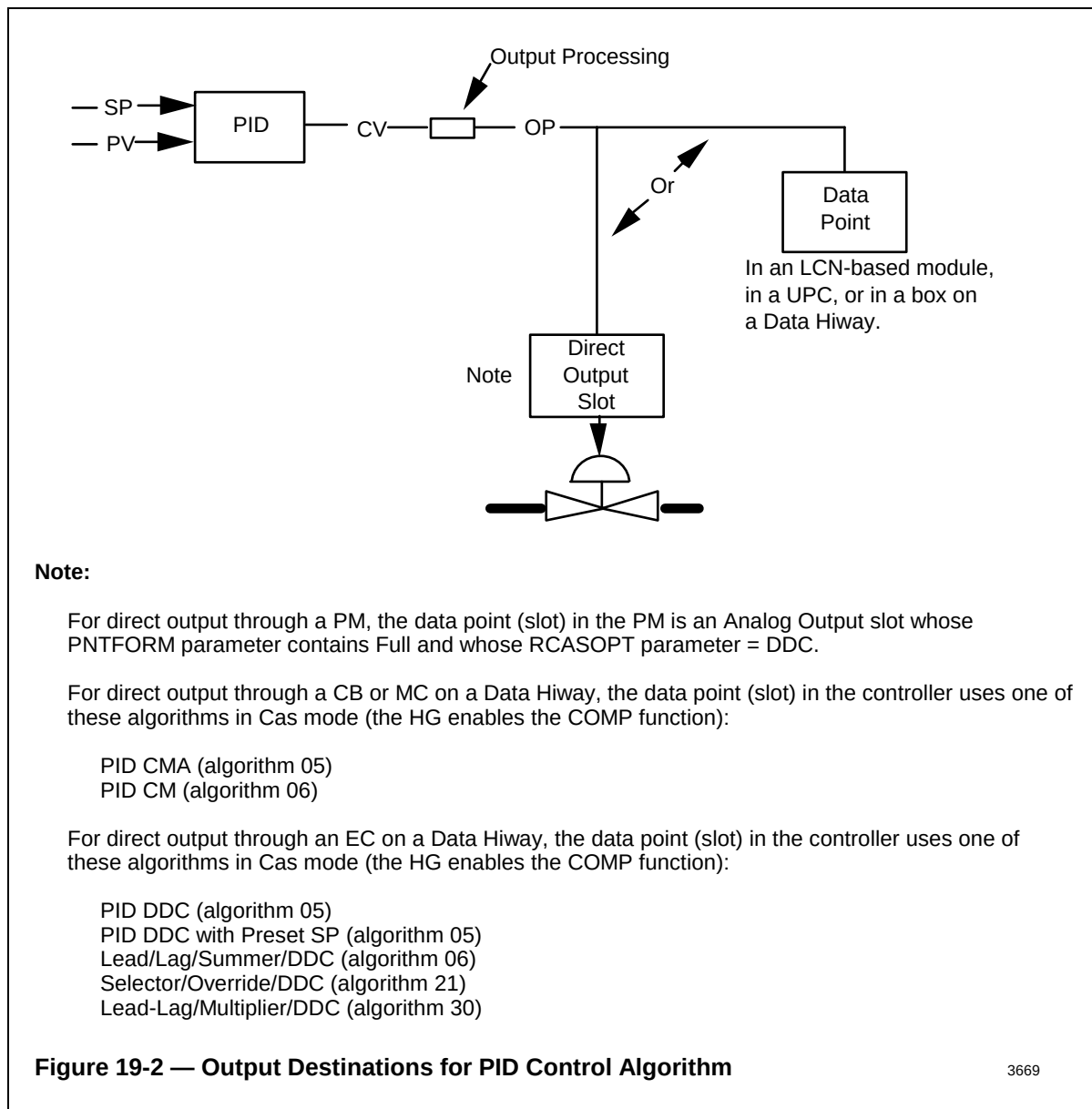
The output of this algorithm is normally "floating," because of the dynamics of the integral and derivative terms. Internally, the output is calculated as increments of output change, but the increments are accumulated to provide a full-value output, thus simplifying the techniques used to achieve "bumpless" 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.



19.3 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. See the following figure.



When the AM PID point is a primary for another data point in the same AM, another LCN module, in a PM, or in a box on a Data Hiway, its output is connected to the SP or the secondary point. If the AM point is directly controlling a valve, its output is connected to the output value (OP) of a slot in a PM, or to OP in one of the Data-Hiway-based points listed on Figure 19-2. The OP value determines the magnitude of the analog current supplied to the valve by the output slot's holding amplifier.

19.4 OPTIONS AND SPECIAL FEATURES

19.4.1 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 19.5). The poles and zeros must be real.
- **Noninteractive (Ideal) Form**—In this form, P, D, and I are added in the time domain. D is a pure derivative. This form is often called the digital-computer version of the PID controller.

19.4.2 Four Combinations of Control Terms

You select the combinations of proportional, integral, and derivative control terms by choosing Equation A, B, C, or D. The equations function as follows (also see 19.5):

- **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.

19.4.3 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 19.5). Setting both T1 and T2 to 0 results in only proportional control.

Use Equation D to achieve only integral control.

19.4.4 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 detailed 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

$$\text{SPP} = 50\%$$

$$\text{PVP} = 51\%$$

$$\text{Deviation} = \text{PVP} - \text{SPP} = 1\%$$

If PVP increases, the deviation (error) increases, so the output, CV, increases (see Equation A under 19.5).

The opposite occurs with reverse-control action: If the deviation increases, CV decreases.

19.4.5 PV Tracking

You can select PV tracking when configuring a data point that uses this algorithm. If you do, SP becomes equal to PV under any of the following conditions:

- The data point that uses this algorithm is in MAN mode.
- The output is indisposable.
- The first time the data point is processed after becoming active.
- On a cold or warm restart, if the data point is configured for external initialization.
- If parameter CTRLINIT contains On and this point is configured for external 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.

19.4.6 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 19.5) 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:

$$K = KLIN * KGAP, \text{ if } (SP - GAPLO) < PV < (SP + GAPHI)$$

or,

$$K = KLIN, \text{ if PV is outside the gap.}$$

Where:

KLIN = A linear-gain parameter, in percent-per-percent. The value of KLIN is tuned at a Universal Station. Default = 1.0.

KGAP = Gain-modification factor, specified by the user. Default = 1.0.

GAPLO = The bottom limit of the gap in the same engineering units as the PV.
Default = 0.

GAPHI = The upper limit of the gap in the same engineering units as the PV. 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:

$$K = KLIN * KNL$$

$$KNL = NLFM + (NLGAIN * |PVP - SPP| / 100.0)$$

If the resulting value in K exceeds 120.0, it is clamped at 120.0.

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 to 10.
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 (Section 5) 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 120.0, it is clamped at 120.0.

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.

It is possible to use this option for multiplicative-feed forward control, but the PID with Feed forward-control algorithm (Section 19) is a better choice because it provides a better operator interface and better recovery from a "bad" feed forward input.

19.4.7 Windup Handling

When the output of this algorithm reaches the user-specified integral or 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.

19.4.8 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.

19.4.9 Initializing PID Output without Affecting Dynamics

A user-written program in an Application Module (CL) or a Computing Module (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 connection is active 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 active 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 initializing its dynamic calculations, so the fuel switchover is quick and smooth.

19.4.10 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- and derivative-time constants be within the following ranges;

$$20.0 \cdot TS < T1 < 20,000 \cdot TS$$

$$T2 > 100.0 \cdot TS \text{ for the interactive form of the PID, and}$$

$$T2 > 10.0 \cdot TS \text{ for the noninteractive (ideal) form of the PID}$$

Where TS = the interval at which the data point is processed, in minutes.

If the values of T1 or T2 are not within the ranges suggested above, you should adjust them or TS to bring them within those ranges.

For the interactive form of the PID, if

$$T1 < 2.0 \cdot TS, \text{ and } T1 \text{ is clamped at } 2.0 \cdot TS$$

$$T2 < 10.0 \cdot TS$$

the respective time constant is treated as 0.

19.4.11 Ratio Control

Ratio control can be achieved by modifying the setpoint input to the PID algorithm by a ratio of some other process value, for example, a fuel-to-air ratio in furnace control (it can also be accomplished with the Ratio Control algorithm. See Section 20). When configuring a PID data point, you can select one of the following options for modifying the setpoint:

- No setpoint 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 modification does not occur because this option is intended to receive the process value to be modified by the ratio, only from another data point.

The "Auto" 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 do essentially the same functions as the PID Ratio, PID Auto Ratio, and PID Auto Bias algorithms in Basic Controllers, Multifunction Controllers, and Extended Controllers.

The parameters used for these options are RBOPTION, RATIO, BIAS, RTHILM, RTLOLM, BSHILM, and BSLOLM.

19.4.12 Operating Modes

The PID algorithm operates in the following modes:

- MAN
- AUTO
- CAS

19.4.13 Restart or Point Activation

On a cold restart, a warm restart, or when the data point is activated, initialization takes place as described under 19.6. On a hot restart, the PID dynamics are returned to a steady state.

19.4.14 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. If so configured, an initialization request and initialization value are sent to the primary data point.

19.5 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

$$CV(s) = K \left[\frac{1 + T1*s}{T1*s} * \frac{1 + T2*s}{1 + a*T2*s} * (PVP(s) - SPP(s)) \right]$$

Equation B—P and I act on error, D acts on PV

$$CV(s) = K \left[\frac{1 + T1*s}{T1*s} * \frac{1 + T2*s}{1 + a*T2*s} * PVP(s) - \frac{1 + T1*s}{T1*s} * SPP(s) \right]$$

Equation C—I acts on error, P and D act on PV

$$CV(s) = K \left[\frac{1 + T1*s}{T1*s} * \frac{1 + T2*s}{1 + a*T2*s} * PVP(s) - \frac{1}{T1*s} * SPP(s) \right]$$

Equation D—Integral control, only

$$CV(s) = \left[\frac{1}{T1*s} * (PVP(s) - SPP(s)) \right]$$

For the Noninteractive form:

Equation A—P, I, and D act on the error

$$CV(s) = K \left[\left(\frac{1 + T1*s}{T1*s} + T2*s \right) * (PVP(s) - SPP(s)) \right]$$

Equation B—P and I act on error, D acts on PV

$$CV(s) = K \left[\left(\frac{1 + T1*s}{T1*s} + T2*s \right) * PVP(s) - \frac{1 + T1*s}{T1*s} * SPP(s) \right]$$

Equation C—I acts on error, P and D act on PV

$$CV(s) = K \left[\left(\frac{1 + T1*s}{T1*s} + T2*s \right) * PVP(s) - \frac{1}{T1*s} * SPP(s) \right]$$

Equation D—Integral control, only

$$CV(s) = \frac{1}{T1 \cdot s} * (PVP(s) - SPP(s))$$

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 19.4.6.

PVP = The process variable in percent

s = The Laplace operator

SPP = The setpoint in percent

T1 = The integral time constant in minutes. See 19.4.10.

T2 = The derivative time constant in minutes. See 19.4.10.

19.6 INITIALIZATION

You can configure this algorithm for

- Internal initialization
- External initialization

Initialization occurs when the data point is in MAN mode or has just recovered from a bad CV, or when the output is indisposable.

If you choose internal initialization, no initialization request or value is sent to the primary data point. On initialization, if the output is disposable

$$CV = OP$$

└────────── The output after output processing.

If the output is indisposable, CV is initialized by request from the secondary.

In either case, the PID dynamics are returned to a steady state.

If external initialization is configured

and the mode is MAN, the output is indisposable, or this point has just recovered from a bad CV value,

CV = OP, if the output is disposable

CV is initialized by request from the secondary if the output is indisposable.

If the mode is MAN or AUTO, the output is indisposable, or this point has just recovered from a bad CV value,

an initialization request is sent to the primary. The initialization value sent to the primary is equal to this data point's setpoint, unless one of the ratio control options is chosen. In that case

$$\text{INITVAL} = (\text{SP} - \text{BIAS})/\text{RATIO}.$$

19.7 OVERRIDE FEEDBACK PROCESSING

When a PID point's secondary uses an Override Selector algorithm (see Figure 19-3), the following functions take place:

- When override feedback is propagated, override status is returned in PTORST to the PID point. The status will be 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, the PID point's CV is initialized in one of two ways. If the direct-control option is chosen, CV is initialized as follows:

$$\text{CV} = (\text{ORFBSEC} - \text{CVEULO})/(\text{CVEUHI} - \text{CVEULO}) * 100 + K * (\text{PVP} - \text{SPP})$$

If the reverse-control option is chosen, CV is initialized as follows:

$$\text{CV} = (\text{ORFBSEC} - \text{CVEULO})/(\text{CVEUHI} - \text{CVEULO}) * 100 - K * (\text{PVP} - \text{SPP})$$

In both examples above, the term $K * (\text{PV} - \text{SPP})$ is the offset value, and ORFBSEC is the override-feedback value (in %) sent to the PID from the secondary. If equation D is selected for the control algorithm (see 19.5), a value of $K = 1.0$ is used in the initialization calculations above (equation D does not use K).

Whether the PID point is selected or not, if it is in CAS mode and configured for external initialization, an override-feedback value is calculated as follows and sent to the primary:

$$ORFB = (PV - BIAS)/RATIO$$

The not connected/not selected/selected status received from the PID's secondary, is also sent on to the primary.

NOTE

For release 410 or later, if the offset value is in such a direction that it causes the nonselected PID to become selected, the offset value will be set to 0.0. The offset value is the term $K*(PV - SPP)$ in the CV initialization examples above.

See Section 22 in this publication and 3.1.11 in *Application Module Control Functions* for more information on override control.

19.8 PID PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the PID algorithm (refer to the *Application Module Parameter Reference Dictionary*):

CTLEQN INITTYPE PVTRACK DEV

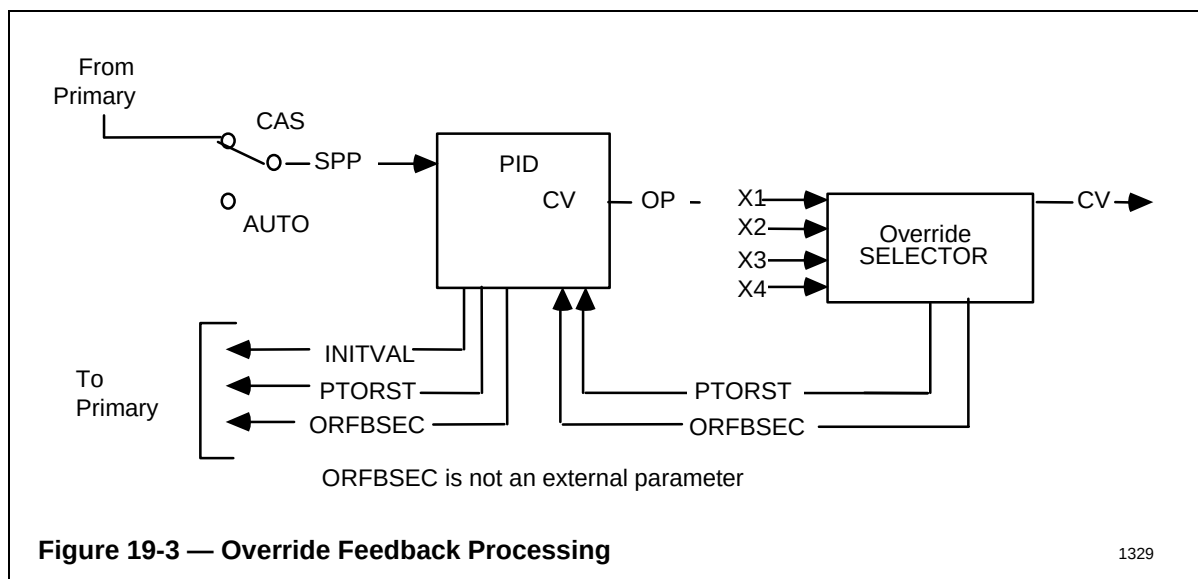
19.9 MIGRATION

The PID algorithm virtually duplicates the functions of 9 of the 12 PID algorithms in SUPERVISORY/TOTAL and PMX Systems. Only the interactive (real) form of the algorithms is available in SUPERVISORY/TOTAL and PMX Systems.

Most of the SUPERVISORY/TOTAL and PMX PID algorithms are actually implemented in Basic Controllers. Four of them can be implemented in the computer system or a Basic Controller. The following table shows how they compare with the PID in Application Modules.

Table 19-1 — Comparison of SUPERVISORY/TOTAL and PMX Algorithms with PID

S/T-PMX Algorithm	Where Implemented?	Nearest AM Equivalent
1 PID	CB or Computer	PID
2 PID Ratio	CB	PID*
3 PID Auto Ratio	CB	PID*
4 PID Auto Bias	CB	PID*
5 PID CMA	CB	PID
6 PID CM	CB	PID
7 PID SPC	CB	PID
10 PID Error Sq. on Gain	CB or Computer	PID
11 PID Error Sq. on Intg.	CB or Computer	PID (See Note)
12 PID Gap	CB or Computer	PID (w/Gap Gain)
* Using Ratio/Bias options in Setpoint Handling.		
Note: Error-squared-on-integral gain modification is not available in Application Modules. Nonlinear-gain modification can provide a similar function (see 17.4.6).		



1329

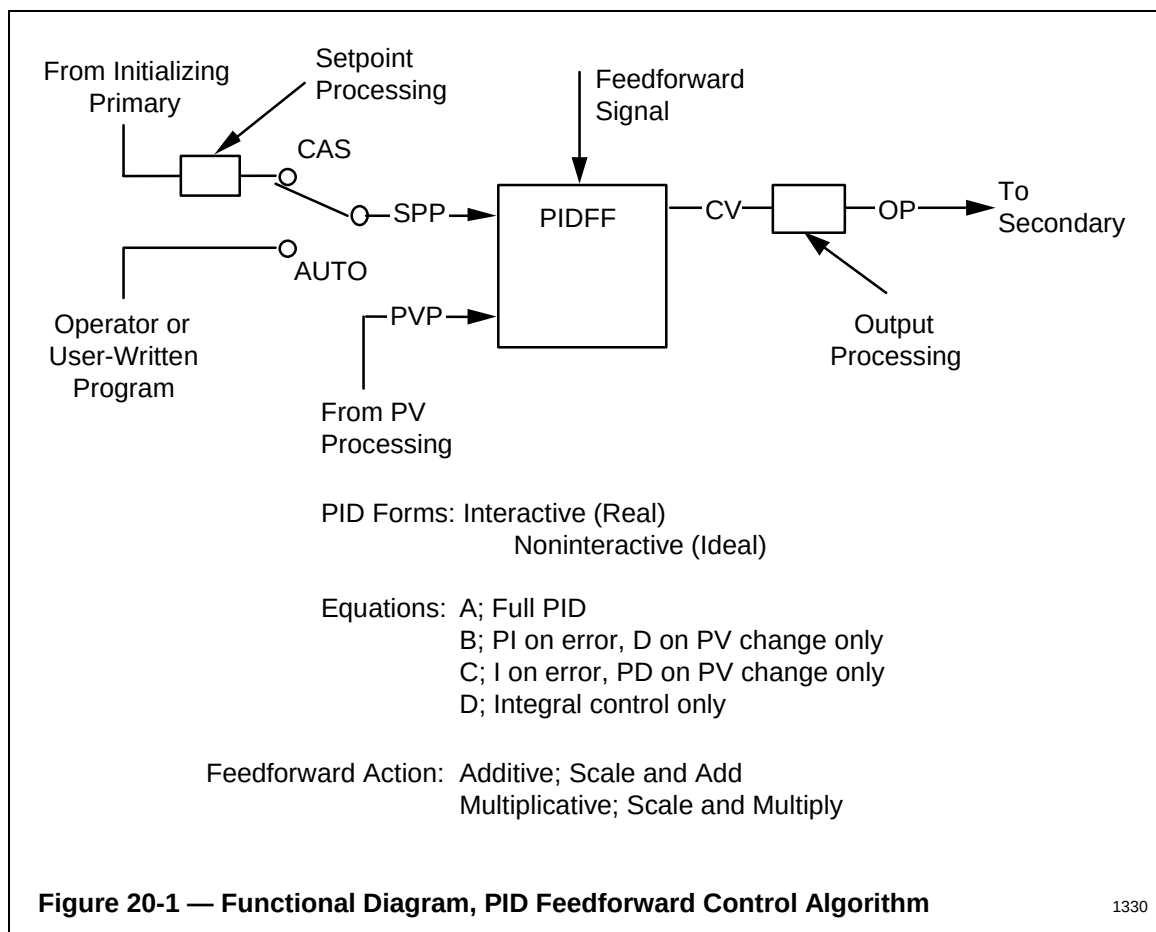
PID FEEDFORWARD (CONTROL) Section 20

20.1 TYPE AND NAME

Control Algorithm: PIDFF

20.2 FUNCTION

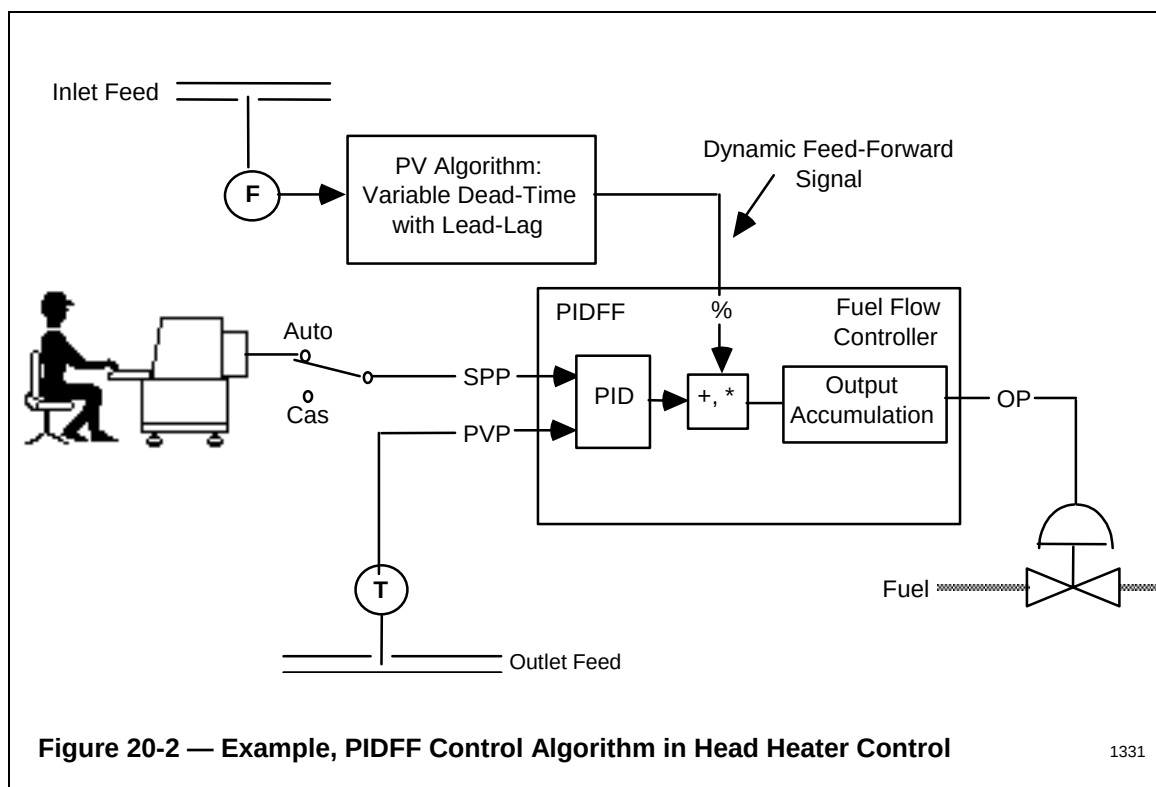
This algorithm operates as a 3-mode (proportional, integral, and derivative) controller. It is identical to the PID algorithm (Section 19), except that it accepts a feedforward signal to be added to, or multiplied by, the algorithm's incremental output, before the full-value output is accumulated. This algorithm lets you combine a feedforward signal with the PID output without using another data point or algorithm to do it.



20.3 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.

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 (Section 12) or the Lead-Lag Control algorithm (Section 15). Similar algorithms are available in Basic Controllers, Multifunction Controllers, and Extended Controllers. Figure 20-2 shows an example of such a strategy.



1331

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 incremental 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, in a heating application, if the feed rate is doubled, twice the amount of fuel might be required, which is the equivalent of a reduction of one-half in the process gain.

20.4 OPTIONS AND SPECIAL FEATURES

All of the following PID Control-algorithm options and special features apply to the PID Feedforward Algorithm:

- 19.4.1 Interactive and Noninteractive PID Forms
- 19.4.2 Four Combinations of Control Terms
- 19.4.3 Control By a Single Term
- 19.4.4 Direct and Reverse Control Action
- 19.4.5 PV Tracking
- 19.4.6 Gain Options
- 19.4.7 Windup Handling
- 19.4.8 Suppression of Output "Kicks" when Switching to CAS Mode
- 19.4.9 Initializing PID Output Without Affecting Dynamics, except where multiplication of the feedforward signal is configured
- 19.4.10 Restrictions on Some Values
- 19.4.11 Ratio Control
- 19.4.12 Operating Modes; MAN, AUTO, and CAS
- 19.4.13 Restart or Point Activation
- 19.4.14 Error Handling

In addition, the following apply to the feedforward action.

20.4.1 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.

20.4.2 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 input is received through a control-input connection, change the status of that connection to Inactive. To resume feedforward action, switch the connection status back to Active.
- If the feedforward signal comes from a PV algorithm, switch the PV source for the data point that is using the PV algorithm to MANUAL (if you do this and the PV is changed while the PV source is in manual, the feedforward signal is affected). To resume feedforward action, switch the PV source 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 MANUAL (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).

20.4.3 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.

20.5 EQUATIONS

You can select PID equations, just as described for the interactive form and the noninteractive form under 19.5.

In addition, the feedforward signal is applied to the incremental output of the PID computation, as follows:

- If additive action is configured

$$CV_n = CV_{n+1} + CVPID + KFF*(FF_n - FF_{n-1})$$

If the status of FF_n or FF_{n-1} is Bad, $CV = CVPID$.

- If multiplicative action is configured

$$CV_n = (CV_{n-1} + CVPID)*(KFF*FF_n + BFF)$$

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, $CVPID = CV/(KFF*FF_n + BFF)$.

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 0.1, it is clamped at 0.1.

Where:

CV = Full-value output in percent, PID combined with feedforward action.

CVPID = The incremental output of the PID computation. This is an internal parameter and is not available to displays nor to user-written programs.

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.

FF lgv = Last good value for the FF input (notation only, not a user-visible parameter).

KFF = Scale factor. Can be used to convert FF to percent as follows;

$$KFF = 100/(EUHI - EULO)$$

Where EUHI and EULO are the high and low limits of the engineering-units range. Default = 1.0.

n and n-1 = Notation to indicate the value this pass (n) and the preceding pass (n-1).

20.6 INITIALIZATION

Initialization is as described under 19.6

20.7 OVERRIDE FEEDBACK PROCESSING

Override-feedback processing is the same as described under 19.7, 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, the PID point's CV is initialized as follows:

$$CV = (ORFBSEC - CVEULO)/(CVEUHI - CVEULO)*100 + K*(KFF*FF + BFF)*(PVP - SPP)$$

if the direct-control option is chosen, or

$$CV = (ORFBSEC - CVEULO)/(CVEUHI - CVEULO)*100 - K*(KFF*FF + BFF)*(PVP - SPP)$$

if the reverse-control option is chosen.

ORFBSEC is the override-feedback value from the secondary.

20.8 PIDFF PARAMETERS

In addition to the parameters already mentioned, parameter FFSTS is associated with the PIDFF algorithm. Refer to the *Application Module Parameter Reference Dictionary*.

20.9 MIGRATION

No PID algorithms that combine a feedforward signal with the PID output are available in SUPERVISORY/TOTAL or PMX Systems. See 19.8.

PID WITH EXTERNAL RESET-FEEDBACK (CONTROL) Section 21

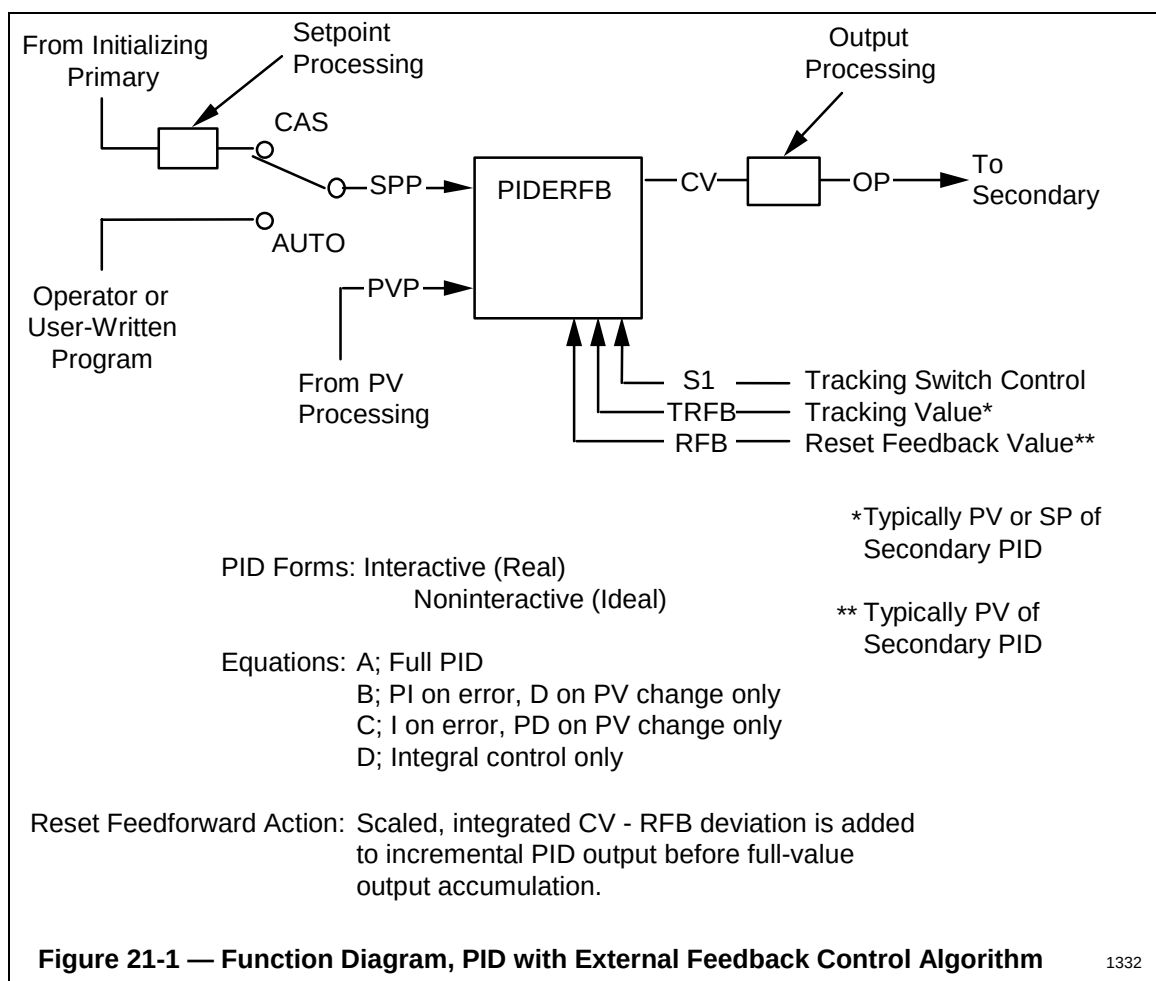
21.1 TYPE AND NAME

Control Algorithm: PIDERFB

21.2 FUNCTION

This algorithm operates as a 3-mode (proportional, integral, and derivative) controller. It is identical to the PID algorithm (Section 18), 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.



21.3 USE

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.

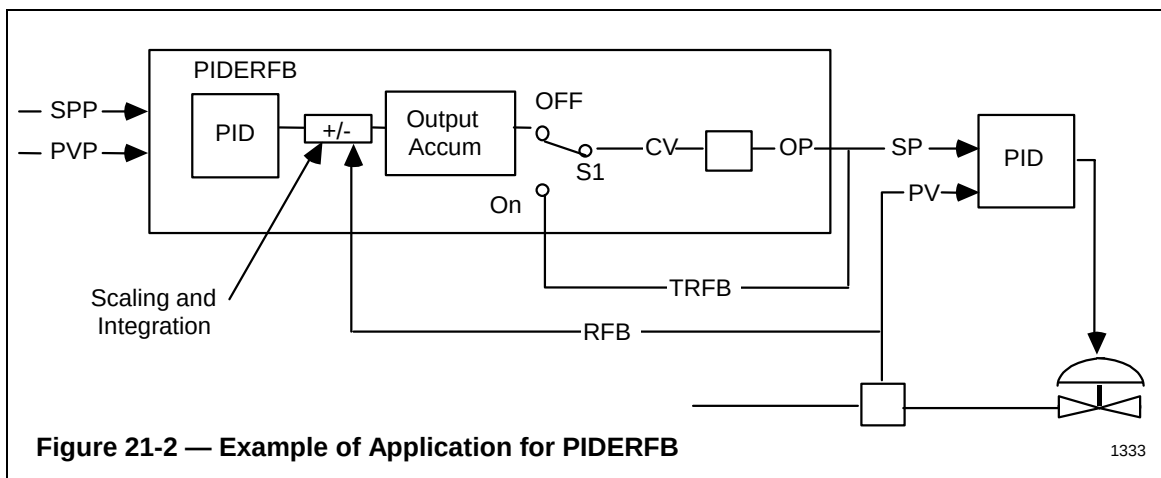
This algorithm also accepts a tracking value (TRFB) and a tracking switch-control signal (S1) from another data point, typically PV or SP of the the secondary PID data point that is receiving its setpoint from this data point. If the switch control is on, the CV value from this data point is replaced by the tracking value.

The RFB and TRFB values are usually received as control-input connections. S1 can be received from a general-input connection or from a CL block.

In a simple application, both the reset-feedback signal and the tracking value may come from the PV of the secondary data point. See the following figure.

If, for some reason, the secondary is not using the output of this data point, S1 is set to On, 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, and CV is then at the same value as the the controlled variable (PV), so there is no bump and normal control can resume.

If, for some reason, there is a sudden difference between the controlled variable and this point's CV value, the integration on the RFB signal smooths the output change.



21.4 OPTIONS AND SPECIAL FEATURES

All of the following PID Control-algorithm options and special features apply to the PID with External Reset Feedback algorithm:

- 18.4.1 Interactive and Noninteractive PID Forms
- 18.4.2 Four Combinations of Control Terms
- 18.4.3 Control By a Single Term
- 18.4.4 Direct and Reverse Control Action
- 18.4.5 PV Tracking
- 18.4.6 Gain Options
- 18.4.7 Windup Handling
- 18.4.8 Suppression of Output "Kicks" when Switching to CAS Mode
- 18.4.9 Initializing PID Output Without Affecting Dynamics
- 18.4.10 Restrictions on Some Values
- 18.4.11 Ratio Control
- 18.4.12 Operating Modes; MAN, AUTO, and CAS
- 18.4.13 Restart or Point Activation
- 18.4.14 Error Handling

In addition, the following applies to the PID with External Reset Feedback.

21.4.1 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 18.6) 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 18.6) and the dynamic terms are returned to a steady state. If so configured, an initialization request is sent to the primary data point.

21.4.2 Control Output Connections

Control output connections cannot be configured for this algorithm.

21.5 EQUATIONS

If the value in the S1 parameter is On,

$$CV = \frac{TFRB - CVEULO}{CVEUHI - CVEULO} * 100.0$$

For equations A, B, and C, if the S1 value is Off,

$$CVRFB(s) = K * \frac{K1}{T1 * s} * [rfb\%(s) - CV(s)]$$

For equation D, if the S1 value is Off,

$$CVRFB(s) = \frac{1}{T1 * s} * [rfb\%(s) - CV(s)]$$

$$CV = CVPID + CVRFB$$

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.

$$rfb\% = \frac{RFB - CVEULO}{CVEUHI - CVEULO} * 100$$

s = The Laplace operator.

S1 = The switch-control flag. Default = Off.

TFRB = The tracking value in percent. Default = NaN.

21.6 INITIALIZATION

Initialization is as described under 18.6.

21.7 OVERRIDE FEEDBACK PROCESSING

Override-feedback processing is as described under 18.7; however, use of PIDERFB in override strategies is not recommended.

21.8 PIDERFB PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the PIDERFB algorithm. Refer to the *Application Module Parameter Reference Dictionary*:

RFBSTS TRFBSTS

21.9 MIGRATION

There is no similar algorithm in SUPERVISORY/TOTAL Systems nor in PMX Systems. There is a similar algorithm, no. 13, in Extended Controllers.

RATIO (CONTROL) Section 22

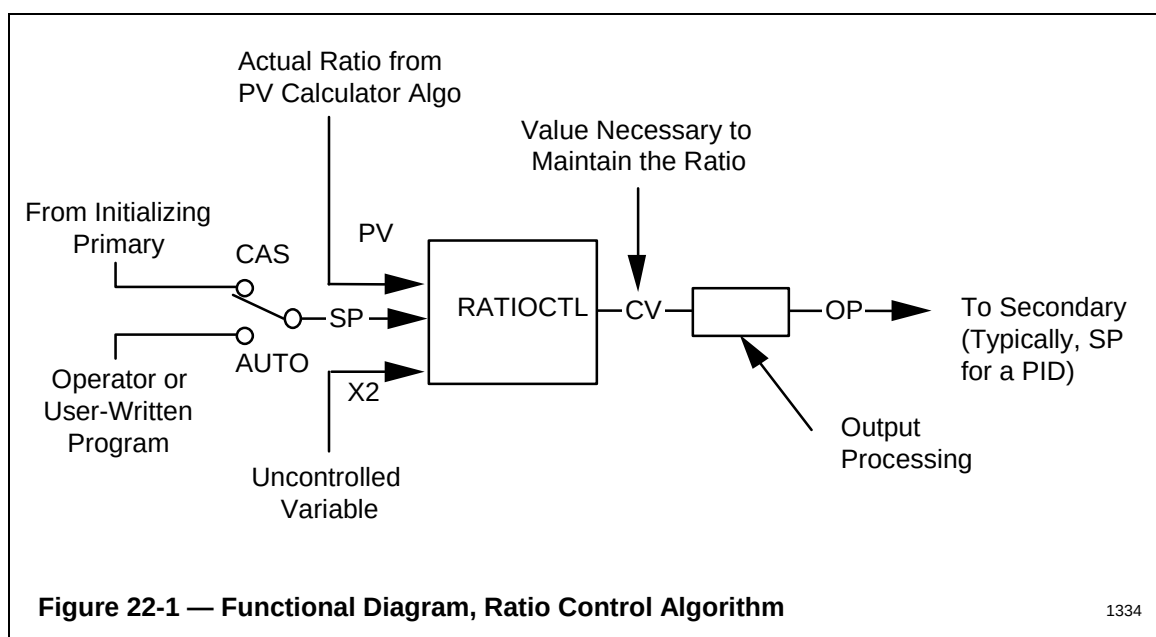
22.1 TYPE AND NAME

Control Algorithm: RATIOCTL

22.2 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 Equation B of the Multiplier/Divider PV algorithm (Section 9) to calculate the measured value of the ratio for displays and reports.

Ratio control can also be accomplished with the ratio-control options of the PID or PID Feedforward control algorithms (see 18.4.11). This Ratio-control algorithm, has several advantages, including the display of the actual ratio attained, as calculated by the Multiplier/Divider PV algorithm, and direct control of the ratio through the SP of the Ratio algorithm.



1334

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 ($C1 = K1$, $D1 = B1$, $K2 = C2$, and $B2 = D2$), 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 "wild" flow, multiplied by the ratio. In the example of Figure 22-2, the controlled flow would be $6.00 \times 2.00 = 12.00$ gallons each minute.

Note that the "wild" flow might not be wild at all, but may actually be a flow controlled by another controller. In any case, the controlled flow stabilizes at a value equal to the "wild" flow, multiplied by the desired ratio, as modified by any scale factors other than 1.0 or any bias values other than 0.

22.4 OPTIONS AND SPECIAL FEATURES

22.4.1 Role of the Multiplier/Divider PV Algorithm

Any data point that uses RATIOCTL should use the Multiplier/Divider PV algorithm (Section 9). Equation B of the Multiplier/Divider algorithm can be used. The uncontrolled ("wild") variable is connected to P2 and the variable controlled by the PID algorithm (see Figure 22-2) is connected to P1. The scale factors and bias values in the PV algorithm must have the same values as their counterparts in the Ratio control algorithm:

RATIOCTL		MULDIV
K1	=	C1
K2	=	C2
B1	=	D1
B2	=	D2

Thus, MULDIV can calculate the actual (measured) ratio attained, and when the PV source is AUTO, that value is available in the PV parameter of the data point for use on displays and reports.

22.4.2 Operating Mode

The RATIOCTL algorithm operates in the following modes:

- MAN
- AUTO
- CAS

22.4.3 Restart or Point Activation

On a cold or warm restart or when the RATIOCTL data point is activated, initialization takes place as described under 22.6. On a hot restart, normal operation resumes with no initialization.

22.4.4 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 22.6.

22.4.5 SP Value Range

You must configure the SP value range in SPEULO and SPEUHI.

22.5 EQUATIONS

The equations are as follows:

- Multiplier/Divider PV Algorithm

Equation B

$$PVCALC = \frac{(C1 \cdot P1 + D1)}{(C2 \cdot P2 + D2)}$$

(Other terms in Eq. B under 9.5 are not used.)

Where (see Figure 22-2)

PVCALC = The calculated, actual ratio achieved.

C1 = The P1 scale factor. Must equal K1 of the RATIOCTL algorithm.
Default value = 1.0.

C2 = The P2 scale factor. Must equal K2 of the RATIOCTL algorithm.
Default value = 1.0.

D1 = The bias value for C1*P1. Must equal B1 of the RATIOCTL algorithm. Default value = 0.

D2 = The bias value for C2*P2. Must equal B2 of 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 uncontrolled process variable. Source should be the same as the X2 input to RATIOCTL. If there is a PID-controller controlling this "wild" flow, the PV of that PID could be the source for P2 and X2.

- Ratio Control Algorithm

$$CV = \frac{SP * (K2 * X2 + B2) - B1}{K1}$$

Where (see Figure 22-2)

CV = The calculated output in engineering units

SP = The desired ratio input

X2 = The uncontrolled process variable. Source should be the same as P2 for MULDIV. If there is a PID-controller controlling this "wild" flow, the PV of that PID could be the source for X2 and P2. Default value = 0.

B1 = Bias constant. Should be the same value as D1 in the MULDIV PV algorithm. Default = 0.

B2 = Bias constant for the X2 input. Should be the same value as D2 in the MULDIV PV algorithm. Default = 0.

K1 = The ratio scale factor. Must equal C1 of the Multiplier/Divider PV algorithm.

K2 = The scale factor for X2. Must equal C2 of the MULTDIV algorithm.

22.6 INITIALIZATION

You can configure this algorithm for

- No initialization
- External initialization

If no initialization is configured, initialization requests from a secondary data point are ignored, no initialization requests are sent to a primary data point, and in CAS mode, CV is normally calculated.

If external initialization is configured, 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:

$$\text{INITVAL} = \frac{K1 \cdot CV + B1}{K2 \cdot X2 + B2}$$

22.7 OVERRIDE FEEDBACK PROCESSING

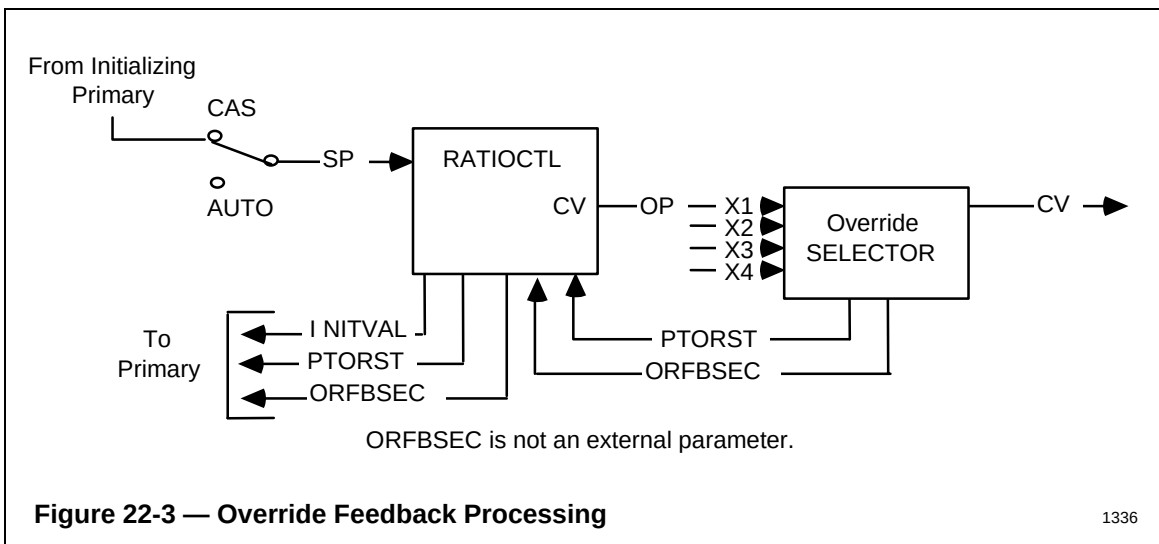
When the data point's secondary uses an Override Selector algorithm (see Figure 22-3), the following functions take place if the RATIOCTL algorithm is configured for external initialization and if it is in CASC mode:

- The override status is sent to the primary data point in PTORST.
- If the status in PTORST is not selected, a feedback value, calculated as follows, is sent to RATIOCTL's primary data point.

$$\text{ORFB} = \frac{K1 \cdot \text{ORFBSEC} + B1}{K2 \cdot X2 + B2}$$

Where ORFBSEC is the override-feedback value received from the secondary data point.

For more detail on override control, see Section 23 of this publication, and 3.1.11 in the *Application Module Control Functions Reference Manual*.



1336

22.8 RATIOCTL PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the RATIOCTL algorithm. Refer to the *Application Module Parameter Reference Dictionary*.

INITTYPE SPEUHI SPEULO

22.9 MIGRATION

Very similar algorithms are available in SUPERVISORY/TOTAL and PMX Systems. They are

Control Algorithm No. 100, Ratio; Control Algorithm No. 54, Ratio

$$C_N = F_{IN} * \text{RATIO} / K$$

PV Algorithm No. 116, Divide; PV Algorithm No. 101 Divide

$$PV = F_{OUT} / F_{IN} * K$$

RAMP AND SOAK (CONTROL) Section 23

23.1 TYPE AND NAME

Control Algorithm: RAMPSOAK

23.2 FUNCTION

This algorithm produces an output that consists of up-to-six alternate ramps and soak periods—a total of 12 segments. The output 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.

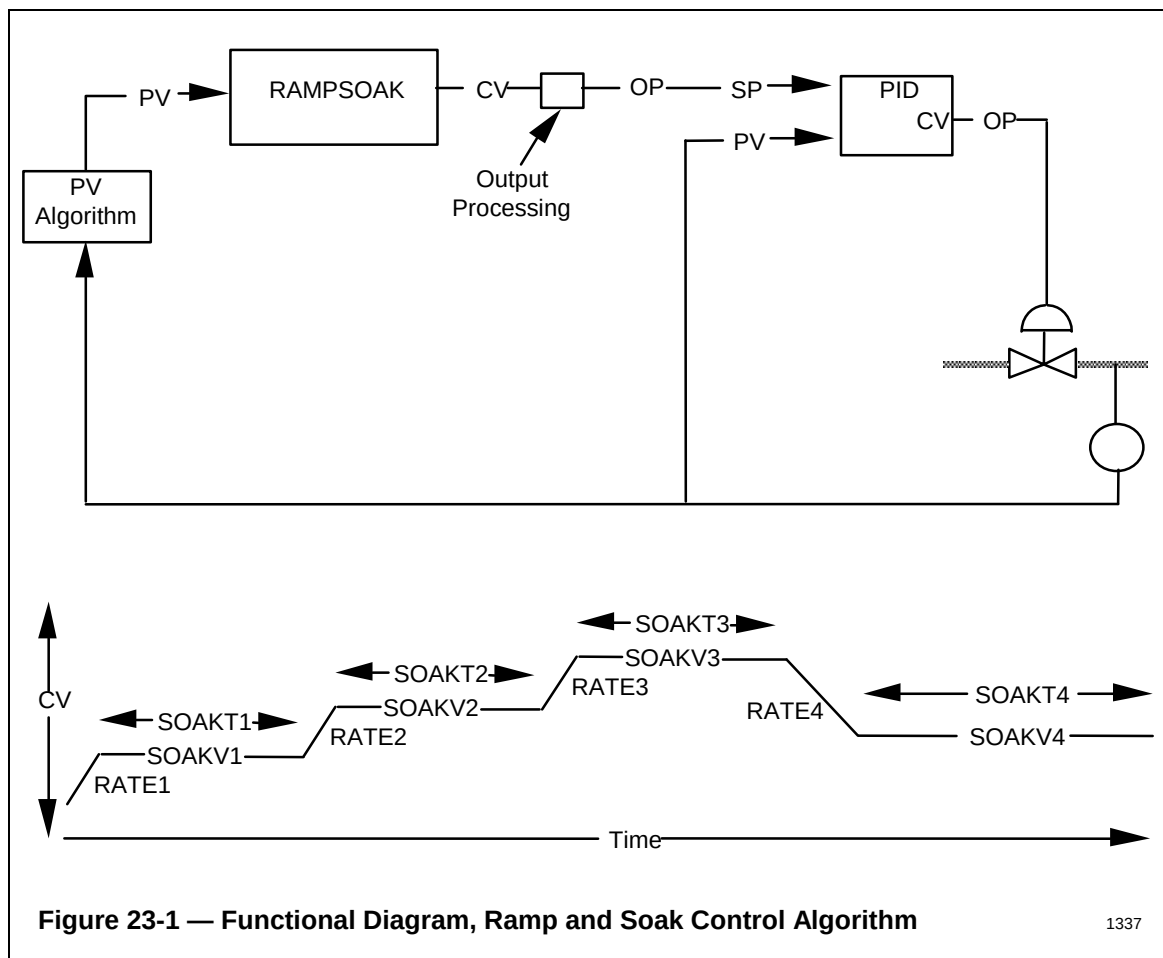


Figure 23-1 — Functional Diagram, Ramp and Soak Control Algorithm

1337

Once started the whole sequence of six ramps and six 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 MANual mode to freeze the sequence, and then return it to AUTO to continue the sequence.

23.3 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.

23.4 OPTIONS AND SPECIAL FEATURES

23.4.1 Operational Features

The operating modes establish the operating state of the RAMPSOAK algorithm as follows:

- MAN—The sequence is stopped and the timers are not running.
- AUTO—The sequence is running.
- CAS—The sequence and timers are reset.

Further functions in each mode are the following:

- MAN mode
 - 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 23.4.4),

CV changes at the ramp rate.

If CV should overshoot the next soak value, it is clamped at that value, and the remaining soak-time (REMSOAKT), the current-segment (CURSEGID), and the mark timers and flags (see 23.4.5) are updated.

23.4.2

- If the guaranteed ramp conditions are not OK, the mark timers (see 23.4.5) 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 23.4.3),
 - The soak timer doesn't start.
 - CV remains at its last value.
 - The mark timers (see 23.4.5) 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 23.4.5) 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.
- CAS Mode
 - CV is equal to SP, the current segment (CURSEGID) is Ramp1, the remaining soak time (REMSOAKT) is zero, and all timers are reset.

23.4.2 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 23.4.5), 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.

23.4.3 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. This check is also bypassed if you don't configure a PV algorithm (select the NULL PV algorithm).

The soak timer can also be kept from starting when HOLDCMD contains On. This allows you to use a general-input connection to HOLDCMD or a write to HOLDCMD by a CL block, to hold the soak timer until some other condition is met. Again, once the timer starts it continues, regardless of the deviation or the value in hold.

HOLDCMD also affects the guaranteed ramp function. See 23.4.4.

23.4.4 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

- $RATE_n > 0$

and

$$PV < (SP - MXRMPDEV)$$

- $RATE_n < 0$

and

$$PV > (SP + MXRMPDEV)$$

These checks are bypassed if MXRMPDEV contains NaN or if this data point doesn't use a PV algorithm (the NULL PV algorithm is configured).

Another condition that stops the ramp is HOLDCMD containing On. You can use a general-input connection to HOLDCMD or a write to HOLDCMD by a CL block 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 23.4.3.

23.4.5 Mark Timer Functions

Two flags are provided with the RAMPSOAK algorithm to indicate to other data points or to a CL block 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 is associated with three parameters that specify the segment, the time after the beginning of the segment, and the time from the beginning of the segment until the end time. These parameters are as follows:

Flag	Segment	Beginning Time	End Time
S1	S1SEGID	S1BGNTIM	S1ENDTIM
S2	S2SEGID	S2BGNTIM	S2ENDTIM

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.

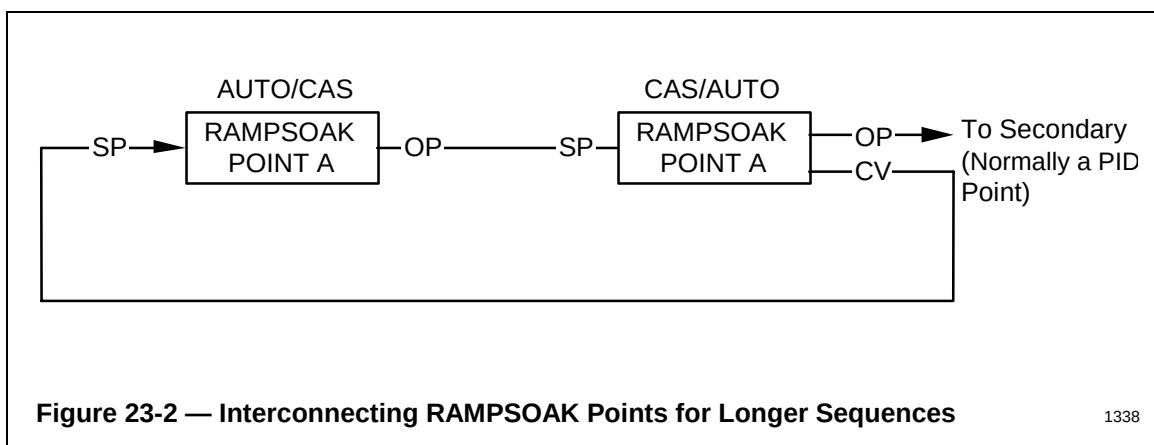
The following functions also take place:

- 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 stopped.
- The mark timers stop 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 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.

23.4.6 Achieving Longer Sequences by Interconnecting RAMPSOAK Points

A sequence of more than 12 ramp and soak segments can be attained by interconnecting RAMPSOAK points as shown on Figure 23-2.

OP from point A is sent to point B by a control-output connection from point A. The CV of point B is sent to SP of point A by a control-input connection in point A. The output (OP) of point B is sent to its secondary (usually a PID point) through a control-output connection. Control-output connections from point B to other secondaries can be made, but none should go to point A—point A gets point B's CV by a control-input connection.



Point B should be configured for external initialization and point A should be configured for internal initialization.

Only one of the RAMPSOAK points should be in AUTO mode at any time. The other should be in CAS mode. The point in AUTO mode goes through its sequence, and at the end of the last segment, it is switched to CAS mode and its timers are reset. At the same time, the other point is switched to AUTO and proceeds through its sequence.

The automatic mode switching can be achieved by configuring a mark flag (see 23.4.5) in each point, to go on just before the end of its sequence. The RAMPSOAK points exchange mark flags through general-input connections and the flags change the modes through external mode-switching. See Section 4 in the *System Control Functions* manual.

23.4.7 How to Get Just One Sequence

In normal operation, the sequence of ramp and soak segments repeats itself as long as the RAMPSOAK point is left in AUTO mode. An operator at a Universal Station can suspend a sequence by changing the mode to MAN, or he or she can reset the sequence by changing the mode to CAS.

A mark flag (see 23.4.5) can be used to switch the mode to MAN or to CAS at any time in a sequence, including at the end of a single sequence. The flag is used to change the mode through external mode-switching. See Section 4 in the *System Control Functions* manual.

23.4.8 Changes of SP by Operators or User-Written Programs

Universal Station operators and user-written programs can change the SP value with no error indication, but the next time the RAMPSOAK point is processed, SP again equals CV. SP tracks CV so that SP can be used for the guaranteed ramp and guaranteed soak functions. See 23.4.3 and 23.4.4.

23.4.9 Notes on Ranges and Limits

In MAN and AUTO modes, SP limits are ignored. To avoid confusion, and to have consistent operation in CAS mode, the SP limits should be configured as NaN.

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. If the RAMPSOAK point doesn't use a PV algorithm, you should configure its SP EU range to be the same as that of the SP of the secondary point.

The deviation limits (MXRMPDEV and MXSOKDEV) apply to all segments in the sequence. If you need different deviation limits in different segments, you can configure MXRMPDEV and MXSOKDEV as NaN and use a CL block and the HOLD flag (see 23.4.3 and 23.4.4) to take over the functions of these deviation limits.

23.4.10 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.

23.5 EQUATIONS

There are no configurable equations for the RAMPSOAK algorithm. The ramp and soak segments are specified in the following parameters (also see Figure 23-1):

- Number of ramp/soak segment-pairs in the sequence—NORSSEQ
Default = 2.

- Ramp Rates, EUs per minute—RATE1 through RATE6
Default = NaN
- Soak values—SOAKV1 through SOAKV6
Default = NaN
- Soak times, in minutes—SOAKT1 through SOAKT6
Default = 0

23.6 INITIALIZATION

You can configure this algorithm for

- Internal Initialization
- External Initialization

If internal initialization is configured, when an initializing condition occurs, no initialization request is sent to a primary data point and the RAMPSOAK point holds in the present state and resumes normal processing when initialization is finished.

If external initialization is configured, an initializing condition has the same effect on the RAMPSOAK point, except that it sends an initialization request to any primary point connected to this point's SP. See 23.4.6.

23.7 OVERRIDE FEEDBACK PROCESSING

This algorithm does not participate in override-feedback processing.

23.8 RAMPSOAK PARAMETERS

All of the significant parameters associated with the RAMPSOAK algorithm have been described. Refer to the *Application Module Parameter Reference Dictionary* for additional parameter information.

23.9 MIGRATION

There are similar ramp and soak algorithms in SUPERVISORY/TOTAL and PMX Systems and in Extended Controllers. They compare as shown in Table 23-1.

Table 23-1 — Comparison of SUPERVISORY/TOTAL and PMX Algorithms with RAMPSOAK

ITEM	EXT. CONTROLLER	S/T-PMX	AM
Algorithm No. 34	Control No. 70	RAMPSOAK	
Max. No. Segments (ramps plus soaks)	6	8	6
Modes MAN CAS AUTO COMP	Stop Ready Sequence N/A	Stop N/A CN = OP Sequence	Stop Reset Sequence N/A
Start Value	Init. or bump	Starts @ SP	Init. or bump
Guar. Ramp & Guar. Soak?	Yes	Yes	Yes
Override Strategy?	Yes	No	No

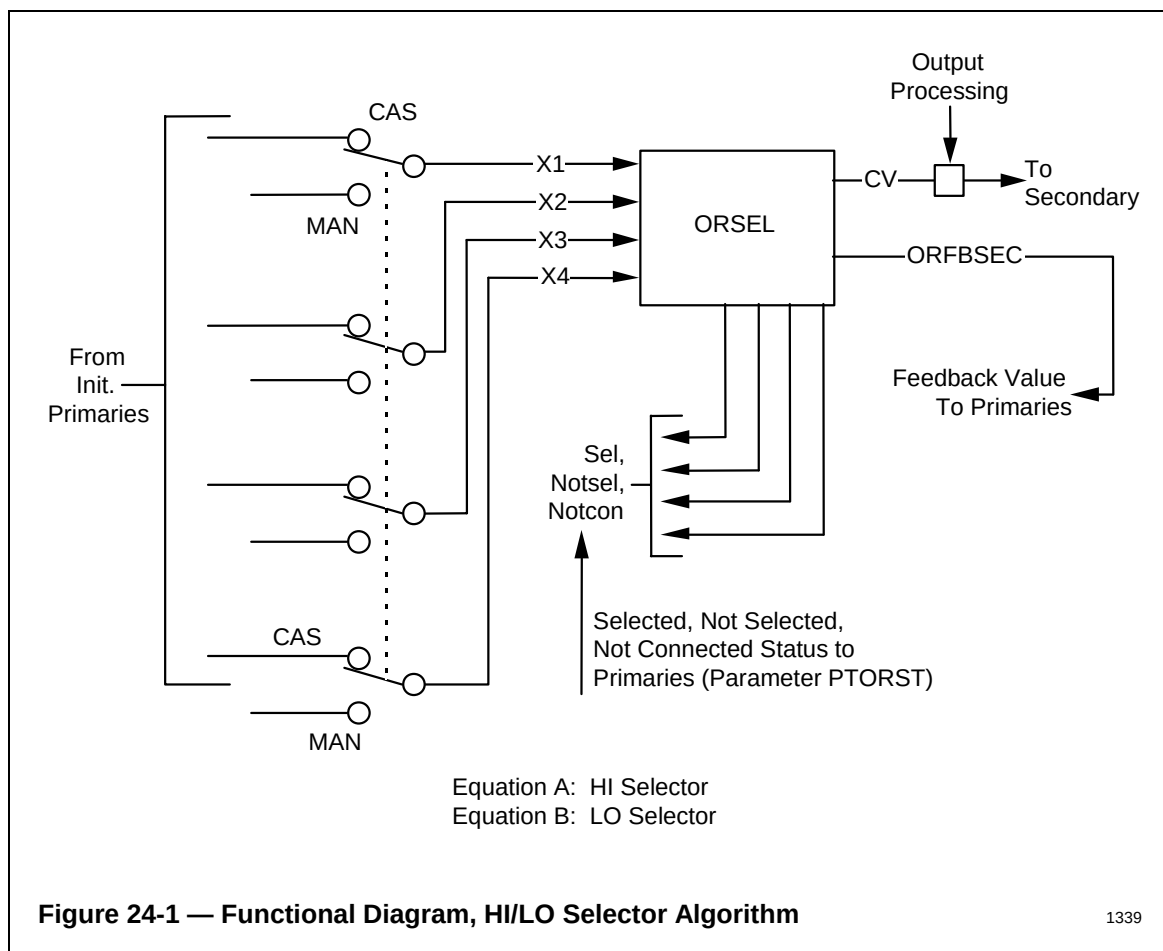
OVERRIDE SELECTOR (CONTROL) Section 24

24.1 TYPE AND NAME

Control Algorithm: ORSEL

24.2 FUNCTION

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. There can be up-to-four inputs, all of which are initializable. 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. If the override option is configured an operator can put the ORSEL point in a bypass state, where the first input is selected and all other inputs are initialized.

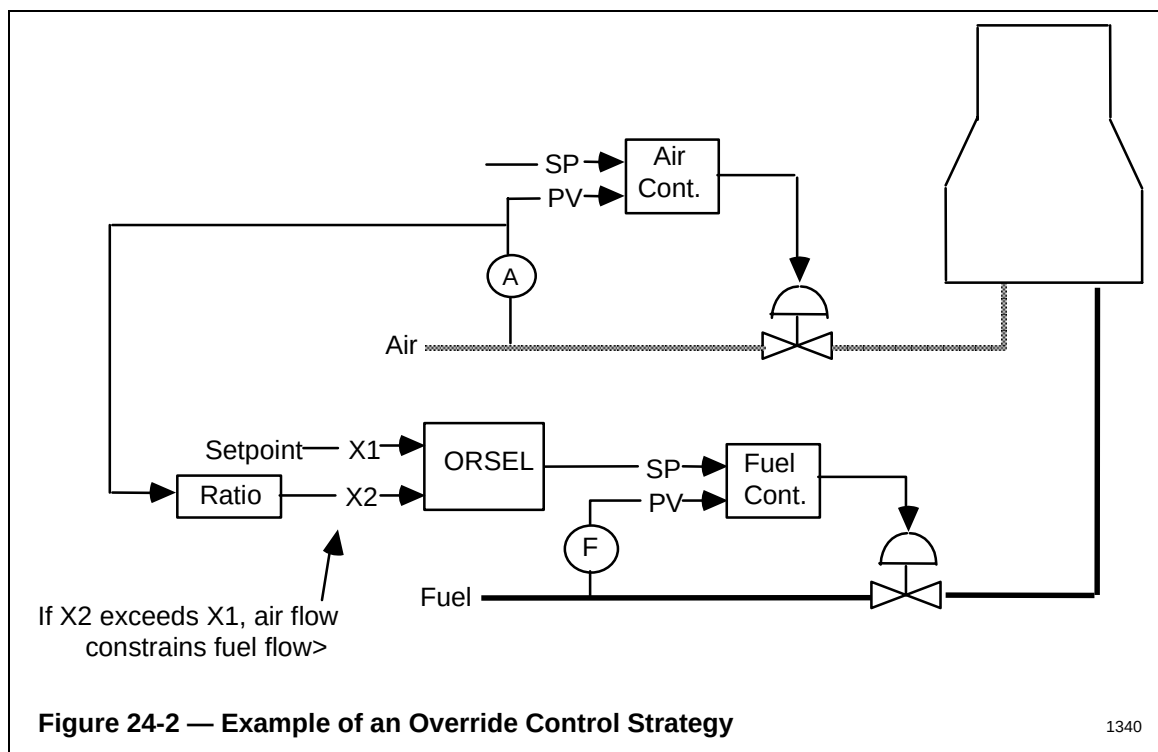


1339

24.3 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 24-2 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.



In a strategy like that of Figure 24-2, Equation A, the override option, and external initialization are configured. PID data points connected to nonselected 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 3.1.11 in *Application Module Control Functions*.

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 3.1.11 in *Application Module Control Functions*.

24.4 OPTIONS AND SPECIAL FEATURES**24.4.1 Override and Bypass Options**

If the override option is configured, external initialization must also be configured. In this configuration, PID points connected to non-selected inputs are prevented from "winding up" by forcing their outputs to track the override-feedback signal (ORFBSEC).

If the override option is configured, an operator at a Universal Station can force the X1 input to be selected by storing On in the BYPASS parameter. When BYPASS is On, X2, X3, and X4 are initialized to the value in X1. Also, when BYPASS is On, only the X1 input has to have a Normal-value status in order for normal override-feedback processing to take place.

An operator can turn BYPASS on and off at any time. When BYPASS is changed from Off to On, X2, X3, and X4 go into initialization. When it is changed from On to Off, X2, X3, and X4 come out of initialization.

24.4.2 Restrictions

The engineering-units range 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.

External initialization must be configured if the override option is configured.

24.4.3 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

24.4.4 Restart or Point Activation

On a cold or a warm restart, or on activating the data point, ORSEL operates normally unless it is configured for external initialization. If it is, CV is initialized to the value returned from the secondary and an initialization request is sent to all of the primaries.

On a hot restart, ORSEL resumes normal operation.

24.4.5 Error Handling

In CAsCade mode with BYPASS Off, if any input has a Bad-value status, CV's value is bad (NaN) (the mode doesn't change when the CV value goes bad). If BYPASS is On, only the X1 input needs to be Normal in order for CV to be calculated normally.

If a Bad input, which was causing CV to be Bad, returns to Normal, CV returns to normal and one of the following occurs:

- If no initialization is configured, CV is normally determined and the primaries are not initialized.
- If external initialization is configured, CV is made equal to OP and the primaries are initialized.

24.5 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

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.

24.6 INITIALIZATION

You can configure this algorithm for

- No initialization
- External initialization

If no initialization is configured, CV and SELXINP are normally calculated, and initialization requests from secondary points are ignored.

If external initialization is configured, the mode is CAS, the output is disposable, the override option is configured, and BYPASS is true; an initialization request from the secondary is passed on to the primaries connected to X2, X3, and X4 with an initialization value equal to CV.

SELECTX1 is placed in SELXINP.

If external initialization is configured and 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, an initialization request is passed on to all primaries with an initialization value equal to CV, and SELXINP contains None.

24.7 OVERRIDE FEEDBACK PROCESSING

24.7.1 Override Feedback Initiation

See Figure 24-1. If the override option is configured for the ORSEL point, external initialization must also be configured. Where both conditions are true, and when the ORSEL point is in CAscade 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:

$$\text{ORFBSEC} = \text{CV}$$

If, under the above conditions, BYPASS is On, the status sent to the primary connected to X1 is SEL, NOTCON is sent to all other primaries and they are all initialized.

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.

24.7.2 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 option and external initialization configured), and
- it is in CAS mode, and
- its output is disposable.

If so,

$$\text{ORFB} = \text{ORFBSEC}$$

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.

24.8 ORSEL PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the ORSEL algorithm. Refer to the *Application Module Parameter Reference Dictionary*.

CTLEQN INITTYPE OROPT XnSTS

24.9 MIGRATION

There are similar algorithms in PMX and SUPERVISORY/TOTAL systems; they are the Override Selectors in Basic Controllers, Multifunction Controllers, and Extended Controllers. The Override Selector algorithms in the controllers, and important differences from the AM HI/LO Selector algorithm are as follows:

- **Basic Controller and Multifunction Controller**

Algorithm No. 21, Override High Selector—Selects the highest of up-to-eight inputs. Operates in MAN, AUTO, and CAS modes. Operation is the same in AUTO and CAS.

Algorithm No. 22, Override Low Selector—Same as No. 21 but selects the lowest of up-to-eight inputs.

- **Extended Controller**

Algorithm No. 21, Selector/Override/DDC—Equation A selects the highest of up-to-six inputs and Equation B selects the lowest of up-to-six inputs. Initialization values are propagated to unselected input data points (slots) every other processing time. Operates in MAN, AUTO, and CAS modes. Also can accept output from the Data Hiway when COMP function is enabled (DDC output). (In systems with an LCN, operating modes are MAN, AUTO, CAS, and BCAS. In CAS mode, a module on the LCN, such as an AM, supplies OP. BCAS is the back-up-cascade mode that the algorithm can operate in if OP from the LCN is not available.)

SUMMER (CONTROL) Section 25

25.1 TYPE AND NAME

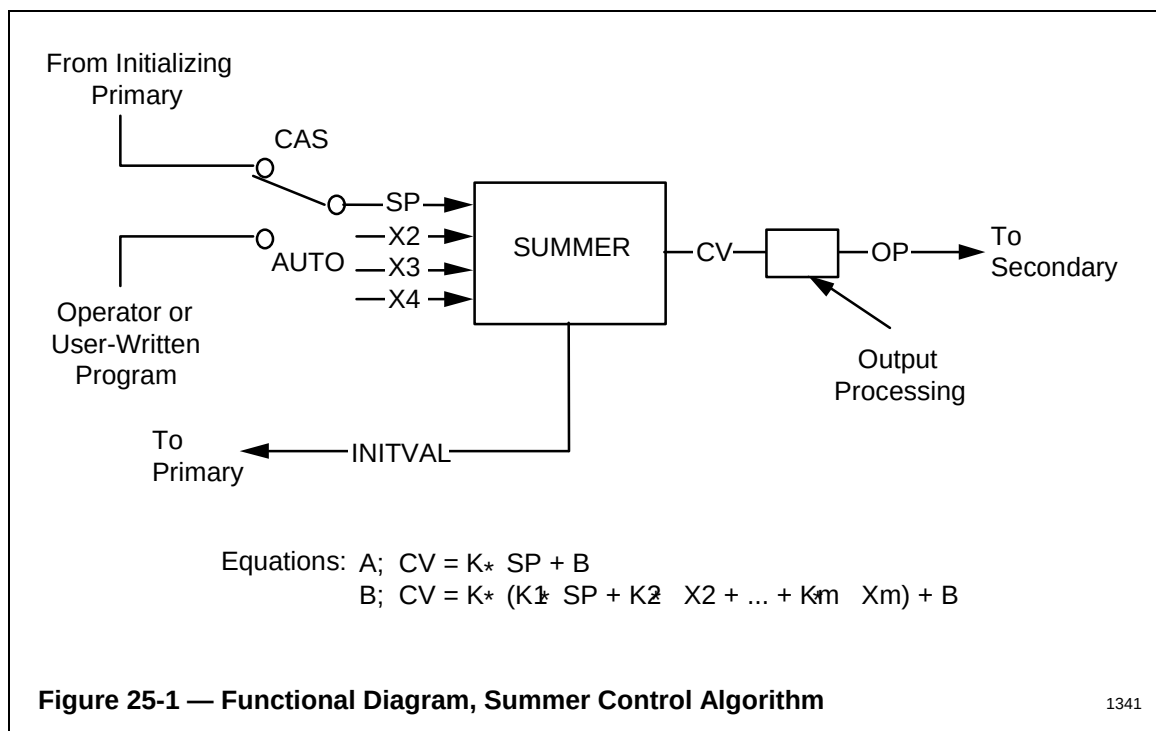
Control Algorithm: SUMMER

25.2 FUNCTION

This algorithm calculates an output value that is the scaled sum of up to three input variables. A bias value can be included in the sum.

Two equations are available. One adds a single scaled input to the bias value. The other adds up to four scaled inputs, multiplies the result by an overall scale factor, and adds the bias value.

This algorithm is similar to PV-algorithm SUMMER (Section 10).



25.3 USE

This algorithm is normally part of a cascade-control strategy. It can be used to calculate a sum of up to four control inputs. It can provide an input to an override-selector algorithm.

25.4 OPTIONS AND SPECIAL FEATURES

25.4.1 Single Input Sum and Four Input Sum

Equation A multiplies a single input by a scale factor and adds the result to the bias value. Equation B multiplies up to four inputs by individual scale factors, adds these products, multiplies the result by an overall scale factor and adds the bias value to the result.

In both equations, the SP input is initializable.

25.4.2 Operating Modes

This algorithm operates in the following modes:

- MAN
- AUTO
- CAS

25.4.3 Restart or Point Activation

For a cold or a warm restart or when this data point is reactivated, initialization takes place as described under 25.6. For a hot restart, normal operation resumes.

25.4.4 Error Handling

If any of the SP and X1 through Xm inputs has a bad-value status, the CV-value status goes bad, and the point continues in the present mode. When the bad input returns to normal, the CV status goes to normal, and if so configured, external initialization takes place (see 25.6).

25.4.5 Setpoint Value Range

You must configure SPEULO and SPEUHI to specify the SP value range.

25.5 EQUATIONS

You can choose one of two equations:

Equation A

$$CV = K \cdot SP + B$$

Equation B

$$CV = K \cdot (K1 \cdot SP + \dots + Km \cdot Xm) + B$$

Where

CV = The control output in engineering units

B = Bias value. Default = 0.

K = Overall scale factor. Default = 1.0.

K1 through Km = Scale factors for SP through the actual number of inputs used.
Default for each = 1.0.

M = No of inputs. Default = 2.

SP = The setpoint in engineering units. This input is initializable.

X2 through Xm = Control inputs X2 through the actual number of inputs used.
These are usually obtained through control-input connections.

25.6 INITIALIZATION

You can configure this algorithm for

- No initialization
- Internal initialization
- External initialization

25.6.1 Initialization Equations

When configured for no initialization, CV is normally calculated, initialization requests from secondaries are ignored, and initialization requests are not sent to primaries.

When configured for internal initialization, the bias (B) is "back-calculated":

$$\text{Equation A} \quad B = CV - K \cdot SP$$

$$\text{Equation B} \quad B = CV - K(K1 \cdot SP + \dots + Km \cdot Xm)$$

When configured for external initialization, the value of SP needed to maintain CV at the present value is calculated and passed as INITVAL to the primary with an initialization request:

$$\text{Equation A} \quad \text{INITVAL} = (CV - B) / K$$

$$\text{Equation B} \quad \text{INITVAL} = \frac{1}{K1} * \left[\frac{(CV - B)}{K} - (K2 \cdot X2 + \dots + Km \cdot Xm) \right]$$

25.7 OVERRIDE FEEDBACK PROCESSING

If this data point's secondary is an Override Selector point and if this point is configured for initialization and 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. The feedback value is calculated as follows:

Equation A

$$ORFB = \frac{ORFBSEC - B}{K}$$

Equation B

$$ORFB = \frac{1}{K1} * \left[\frac{ORFBSEC - B}{K} - K2 * X2 - . . . - Km * Xm \right]$$

25.8 SUMMER PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the SUMMER algorithm. Refer to the *Application Module Parameter Reference Dictionary*:

INITTYPE SPSTS XnSTS

25.9 MIGRATION

Three control algorithms in SUPERVISORY/TOTAL Systems have similar functions. Two of the three are also available in PMX Systems. Table 25-1 compares them.

Table 25-1 — Comparison of SUPERVISORY/TOTAL and PMX Algorithms

Nearest AM Equation	S-T/PMX Algorithm Number	S-T/PMX Equation in AM Terms	S-T/PMX Equation as shown in S/T-PMX Pubs.
B	30†	$CV = K1 * X1 + K2 * X2 + B$	$C_N = K1 * X + K_A * Y + K_2$
B	50	$CV = X1 + X2 + B$	$C_N = PV + P + K_A$
B	51	$CV = X1 + X2 + B$	$C_N = PV + P + K_A$
†Not available in PMX systems. This algorithm can be implemented in a Basic Controller or in the SUPERVISORY/TOTAL computer.			

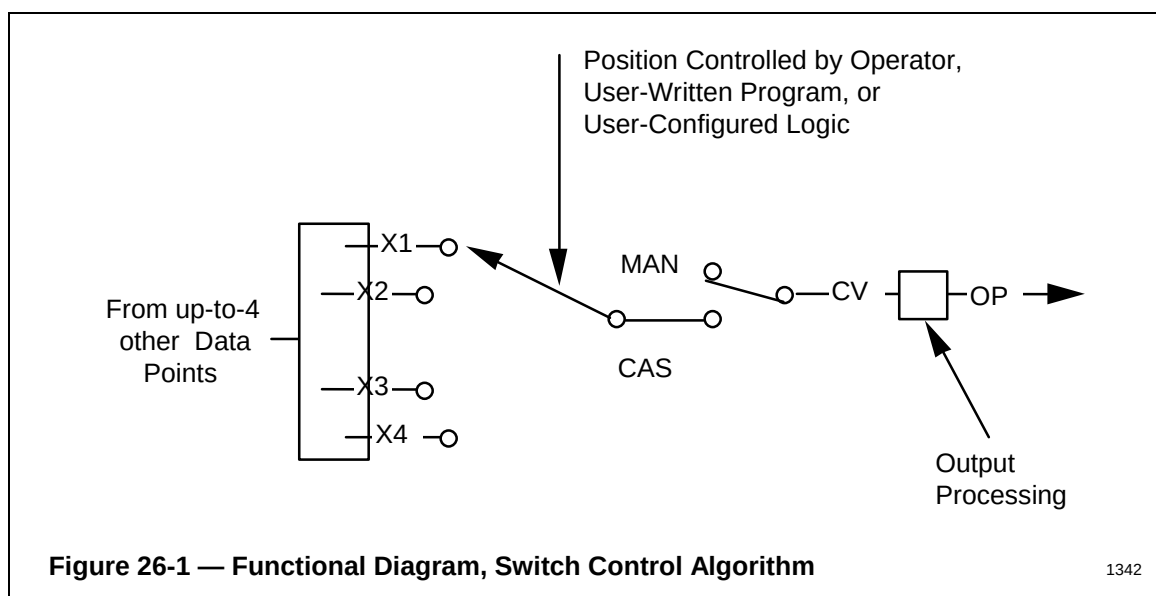
SWITCH (CONTROL) Section 26

26.1 TYPE AND NAME

Control Algorithm: SWITCH

26.2 FUNCTION

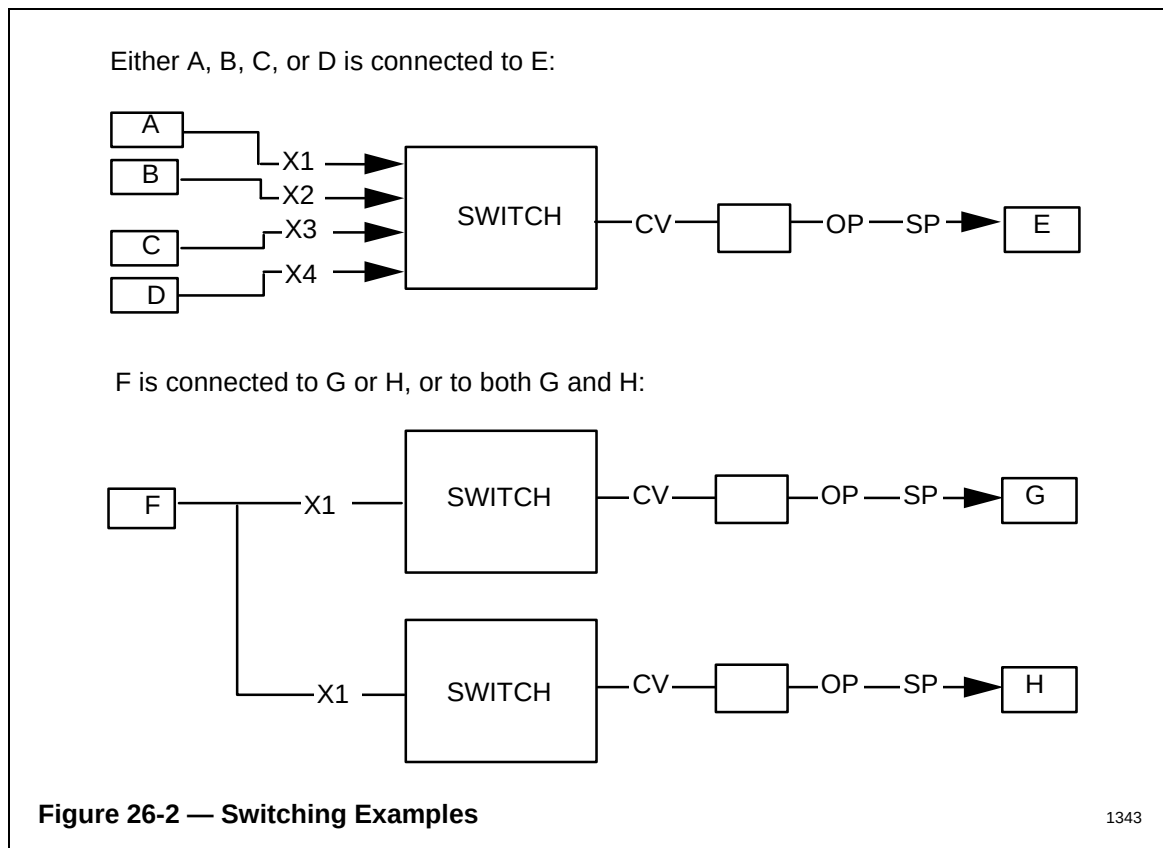
This algorithm operates as a single-pole, 4-position, rotary switch. An operator at a Universal Station, a user-written program, or user-configured logic can change the position of the switch, thereby selecting any one of the four inputs to be the control-algorithm output value, CV.



26.3 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 CL program can change the switch position, or you can configure logical parameters in another data point and use general-input connections from those parameters to control the switch position.

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 26-2 shows an example of each of these situations.



26.4 OPTIONS AND SPECIAL FEATURES

26.4.1 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 X_n input selected. The corresponding Si-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 these 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.

26.4.2 Program Control of Switch Position—Equation B

If Equation B is chosen, an operator cannot change the switch position, but a CL program or general-input connections from logic parameters can. The SELXINP parameter indicates the present switch position, but it cannot be directly changed by the program or input connections.

The switch position is changed by storing On and Off in the Si 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.

With Equation B, turning an Si indicator On does not turn the others off, as it does with Equation A.

26.4.3 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. (Bumpless mode switching is accomplished by the External Initialization option. See 26.6.)

Proper operation of the tracking option requires that External Initialization be configured so that the primaries connected to non-selected inputs can be initialized. Should one of them 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 external initialization from one of its other secondaries.)

When SWITCH is included in an override control strategy, the tracking option must be configured.

26.4.4 Operational Modes

The Switch-control algorithm operates in the following modes:

- MANual
- CAscade

26.4.5 Restart or Point Activation

On a cold or warm restart, or when the SWITCH data point is activated, initialization takes place, as configured. See 26.6.

On a hot restart, normal operation resumes with no initialization.

26.4.6 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, and if configured for external initialization, an initialization request is sent to the primary data point.

26.4.7 Input Value Range

You must configure the X-input range in XEULO and XEUHI.

26.5 EQUATIONS

Equations A and B:

$$CV = X_n$$

$$n = 1, 2, 3, \text{ or } 4$$

SELXINP = the selected input, which can range from SELECTX1 through SELECTX4.

S1 through S4 indicate the switch selection, as described under 26.4.1 or 26.4.2.

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.

26.6 INITIALIZATION

You can configure the SWITCH algorithm for

- No Initialization
- External Initialization

If you select no initialization, initialization requests from a secondary data point are ignored and initialization requests are not sent to the primaries.

If you select external 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 26.4.3), the non-selected primaries are continually initialized.

26.7 OVERRIDE FEEDBACK PROCESSING

If this data point's secondary is an Override Selector point and if this point is configured for initialization and 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.

26.8 SWITCH PARAMETERS

In addition to the parameters already mentioned, parameter TRACKING is associated with the SWITCH algorithm. Refer to the *Application Module Parameter Reference Dictionary*.

26.9 MIGRATION

No Switch algorithm is available in PMX Systems. SUPERVISORY/TOTAL Systems feature Switch control algorithm no. 26 and in Extended Controllers, algorithm no. 26 is a Switch algorithm. Some of the features of the AM Switch control algorithm, and the switch algorithms in SUPERVISORY/TOTAL Systems and the Extended Controller are compared in the following table.

Table 26-1 — Comparison of SUPERVISORY/TOTAL Algorithms with SWITCH

	APPL. MODULE	SUPERVISORY/TOTAL	EXT. CONT.
Algo. Name/Number	SWITCH	26	26
No. of Inputs	4	2	3
Modes	CAS, MAN	AUTO, MAN	AUTO, MAN
Selection	Operator, Program	By Mode Change	Operator, Computer
Override Processing?	Yes	No	Yes

CL CONTROL ALGORITHM (CONTROL) Section 27

27.1 TYPE AND NAME

Control Algorithm: CL

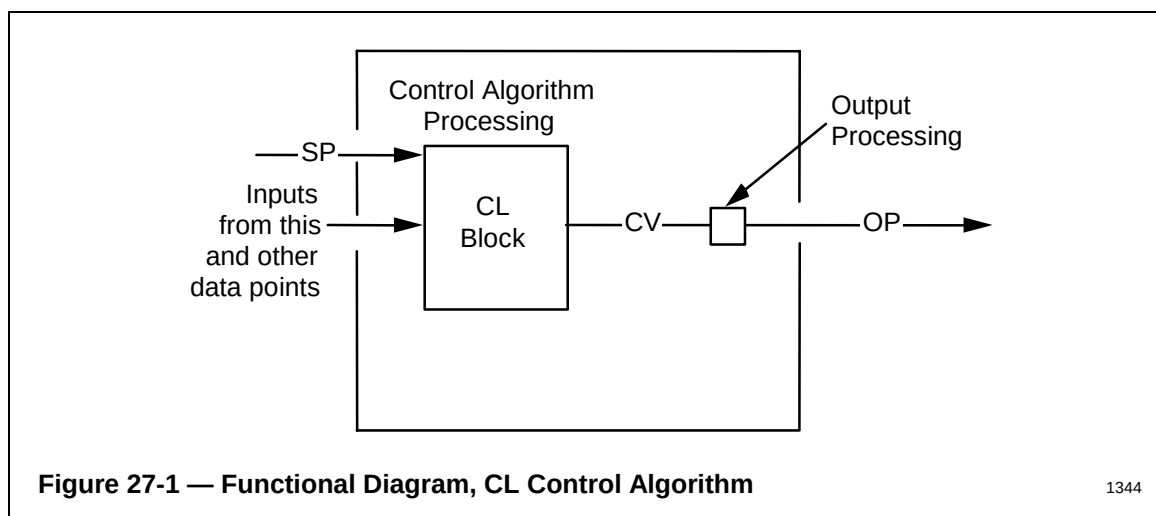
27.2 FUNCTION

This algorithm is a user-written CL block that is like any other CL block except that it is inserted at the control-algorithm insertion point in the processing sequence (see Fig. 2-1), and it is executed instead of a standard control algorithm.

The CL block must calculate and store a control-algorithm output value in CV. Inputs to the CL block are usually acquired by direct references in CL, but can be acquired through general inputs to a Custom Data Segment (CDS) that is included in the data point. Control inputs cannot be configured in the Parameter Entry Display (PED) when the CL Control Algorithm is specified. The value placed in CV by the CL block is processed just as CV is processed for any other data point that uses a control algorithm.

The CL block must also compute and store an anti-windup direction in ARWDI. Propagation of windup status to the primaries is automatic. See 3.1.10 in the *Application Module Control Functions* manual.

If this data point is part of an override strategy, the CL block must include appropriate override functions. See Section 23 in this publication and 3.1.11 in the *Application Module Control Functions* manual.



1344

27.3 USE

A CL block is used when normal point processing is appropriate but none of the standard control algorithms will accomplish the desired function. The CL block can be bound to a single data point, if only one needs its functions, or it can be written as a generic CL block and bound to several data points.

27.4 OPTIONS AND SPECIAL FEATURES

27.4.1 Restart

Normally, the CL block doesn't need to check the type of restart because the value in the PATHIND parameter (see 27.4.2) is affected by the type of restart; however, if needed, the CL block can base some specific actions on the content of the RESTART parameter.

27.4.2 Use of Key Control Subsystem Parameters

This data point provides the following standard control parameters, which should be used appropriately by the CL block:

- CVTYPE—Configured when the point is built to indicate the type of value in CV. It can be EngrUnit or Percent.
- ARWDI—Configured to define the directional relationship between the output and the setpoint. It contains Direct (SP increase causes CV increase) or Reverse (SP increase causes CV decrease). The CL block can change the value, if it needs to be changed, based on one or more of the inputs. Default value = Direct.
- INITTYPE—Configured to indicate the type of initialization, None, Int, or Ext. If Int is configured, the CL block should do its own internal initialization, such as adjusting bias to eliminate a bump to the process. If Ext is configured, the CL block must calculate the value in INITVAL that the primary is to provide to this point's SP. Configure None if no initialization is necessary.
- PATHIND—Path indicator, which contains a value derived by the control subsystem that indicates how the CL block should function on this processing pass. The values are as follows:

Fwd—Perform the normal forward calculation. SP is available as a standard input. Other inputs must be obtained by the CL block either by directly referring to them (as ABC100.PV) or by using general inputs to bring them into a custom data segment included in this point. If the value of a required input is bad, a bad-control alarm should be generated by storing Bad in CV (see 27.4.3), but the CL block should not abort.

Init—Initialize according to the type of initialization called for in INITTYPE. If INITTYPE contains Ext, calculate the SP from the CV and from the other inputs. CV has already been initialized.

OR—Participate in override-feedback propagation. Override feedback propagation occurs only on points that are “upstream” from a point using an override control algorithm. Also, it only occurs when the point’s mode is CAS and when INITTYPE contains Ext. Set the primary's PTORST to the value in your own PTORST and request a process special of the primary by placing OR in its PPSREQ parameter. In addition, if PTORST indicates NotSel, compute the feedback value for the primary, based on the value received in ORFBSEC and place that value in the ORFBSEC of the primary point.

Hold—Hold because the point is in MANual mode or the output is indisposible. CV should be computed as for a forward pass to continue proper handling of the bad-control situation.

NOTE

The HOLD and OR values are reserved words in CL. To use them in a CL program, place an apostrophe just before the word. For example

IF PATHIND = 'HOLD THEN EXIT

27.4.3 Error Handling

If an input used in calculating CV has a Bad-value status, the CL block should use the appropriate CL statement to declare the CV to be bad. Output processing then initiates "bad-control" handling.

27.4.4 Processing Schedule and Execution Time

The CL block should not be used to perform long, complex operations because there may not be enough time in normal point processing to complete such operations. Points that use a CL control algorithm should be scheduled at the longest reasonable interval, and if possible, should be assigned to the Slow Processor. If execution of the CL block takes too much time, it is aborted and an alarm is generated.

27.5 EQUATIONS

The equation(s) for a CL control algorithm, if any, is a function of the CL block.

27.6 INITIALIZATION

See 27.4.2.

27.7 OVERRIDE FEEDBACK PROCESSING

See 27.4.2.

27.8 CL ALGORITHM PARAMETERS

In addition to the parameters already mentioned, the following parameters are associated with the CL Control Algorithm. Refer to the *AM Parameter Reference Dictionary*.

SPEULO SPEUHI

27.9 MIGRATION

Control algorithm 77 in SUPERVISORY/TOTAL Systems and control algorithm 64 in PMX Systems have functions that are similar to the CL Control Algorithm. Table 27-1 compares functions.

Table 27-1 — Comparison of SUPERVISORY/TOTAL and PMX Algorithms with CL

	S/T	PMX	CL Control
Input access	Destination words or explicit	Explicit only	General inputs or explicit references (SP available)
Accessible parameters	Defined list	Defined list	Same as for all control algorithms
Enumerations	No	No	Yes
Data type checking?	No	No	Yes. User defines data types
Initialization	Yes	Yes	Yes

FAX Transmittal

FAX No.: (602) 313-4212

TO: Information Development

Total pages: _____
(including this page)

READER COMMENTS

Title of Document: **AM Algorithm Engineering Data**

Issue Date: **12/02**

Document Number: **AM09-601**

Comments: _____

Recommendations: _____

FROM:

Name: _____ **Date:** _____

Title: _____

Company: _____

Address: _____

City: _____ **State:** _____ **ZIP:** _____

Telephone: _____ **FAX:** _____

FOR ADDITIONAL ASSISTANCE:

Write	Call
Honeywell International Industry Solutions 16404, N. Black Canyon Highway Phoenix, AZ 85053	Technical Assistance Center (TAC) 1-800-822-7673 (48 contiguous states except Arizona) 602-313-5558 (Arizona) Or E-mail us at: pst@honeywell.com

Honeywell

Industry Solutions

16404 North Black Canyon Highway

Phoenix, AZ 85053-3095