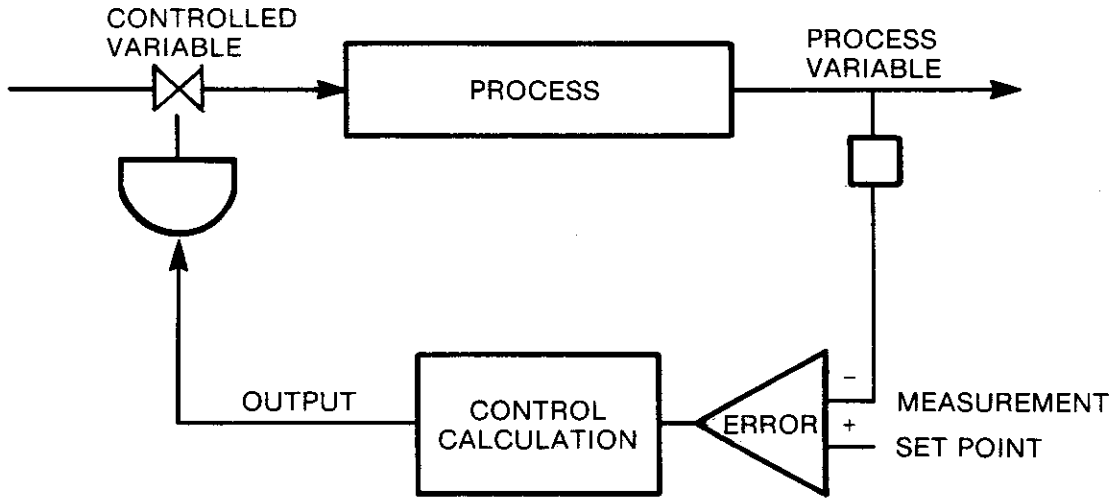
PROPORTIONAL — INTEGRAL — DERIVATIVE

Typical Analog Control Loop



The basic function of analog loop control is to balance the amount of energy or material being supplied to a process against the amount of energy or material consumed by the process.

This is done by comparing a measurement, termed the "Process Variable", with the desired control point termed the "Setpoint". The difference between the measurement and setpoint is termed the "error".

This error calculation is fed into a control calculation (algorithm) which calculates the control value (output) necessary to set the controlled variable so that the measurement = the set point.

The control algorithm provides outputs based on three types of control:

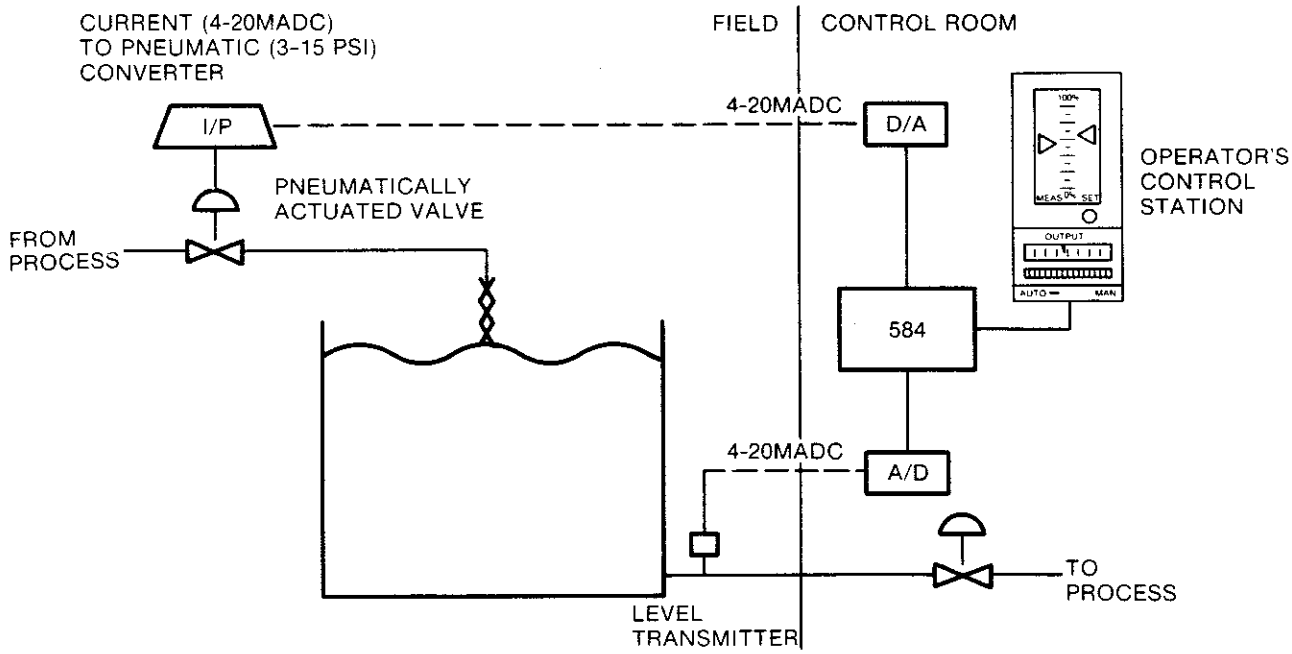
Proportional (P) — Provides an output which is proportional to the error.

Integral (I) — An addition to the proportional output based on how long the measurement has been away from set point.

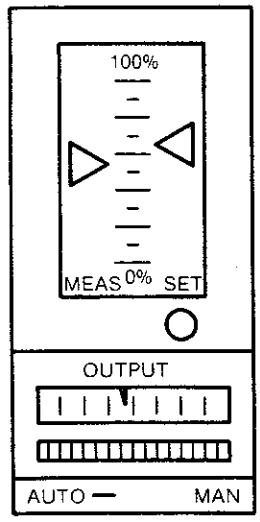
Derivative (D) — An addition to the proportional output based on how fast the measurement is moving away from set point.

PROPORTIONAL — INTEGRAL — DERIVATIVE

Typical Process Loop Diagram



The above figure depicts the typical instrumentation found in a tank level control application. The measurement signal is provided by the level transmitter. This is the process variable. The output signal drives the intake valve, which is the controlled variable. The set point is entered by the operator via the control station. A typical analog control station is depicted below.



Control Stations Provide:
- Measurement indication
- Set point adjustment and indication
- Output adjustment (manual only) and indication
- Auto/Manual Switching

PID

Proportional Control Action

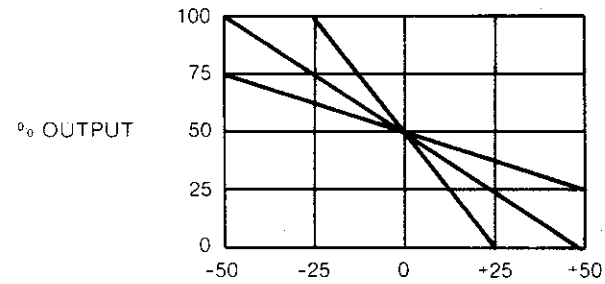
"Proportional" means that the output of the controller is some multiple (gain) of the error (measurement — set).

$$\text{OUTPUT} = \text{GAIN} \times \text{ERROR}$$

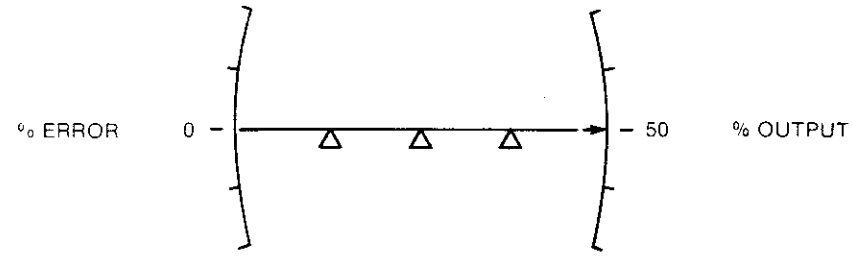
This multiple is termed the "Gain" of the controller. Some analog controllers have a gain adjustment, however, most have a "proportional band" adjustment. Gain, expressed as "percent proportional band" follows:

$$\text{GAIN} = \frac{100}{\%\ \text{PROPORTIONAL BAND}}$$

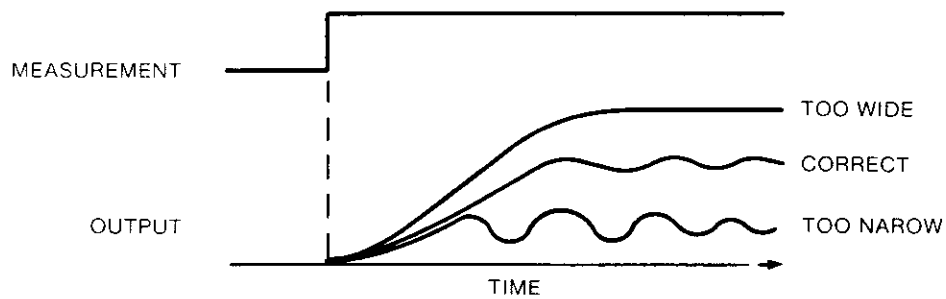
As the proportional band approaches zero, the gain of the controller approaches infinity (e.g. a light switch has a zero proportional band). If the proportional band = 100, the gain of the controller = 1.



PROPORTIONAL DIAGRAM



Proportional Controller Response for Three Proportional Band Settings



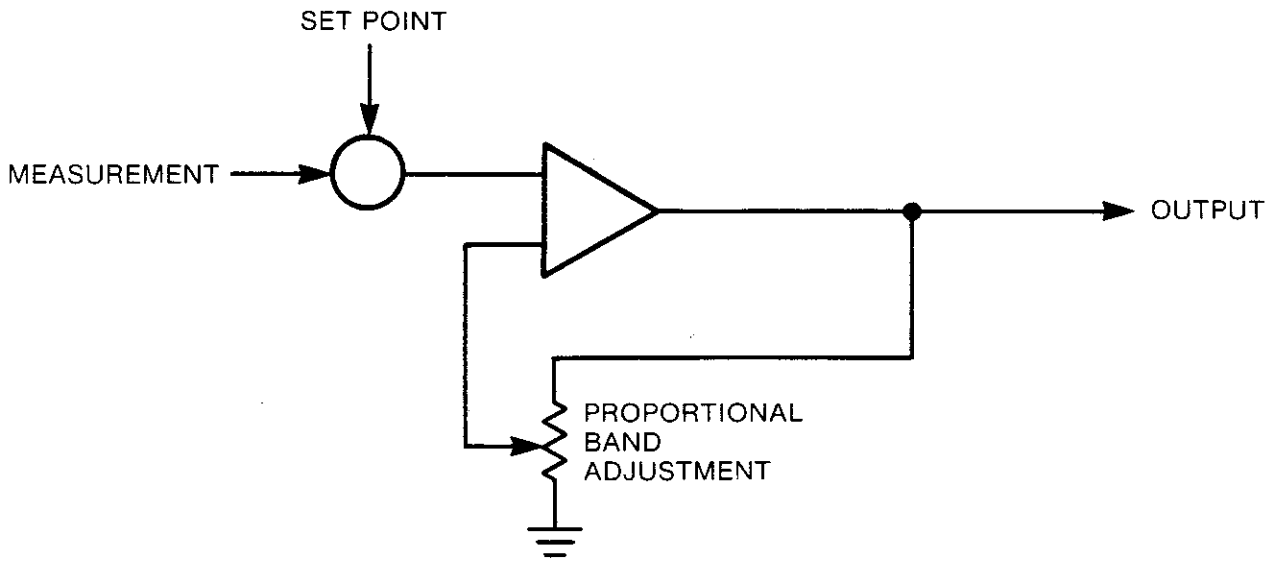
EXAMPLES OF THREE PROPORTIONAL BAND SETTINGS

Proportional Controller Action Formula

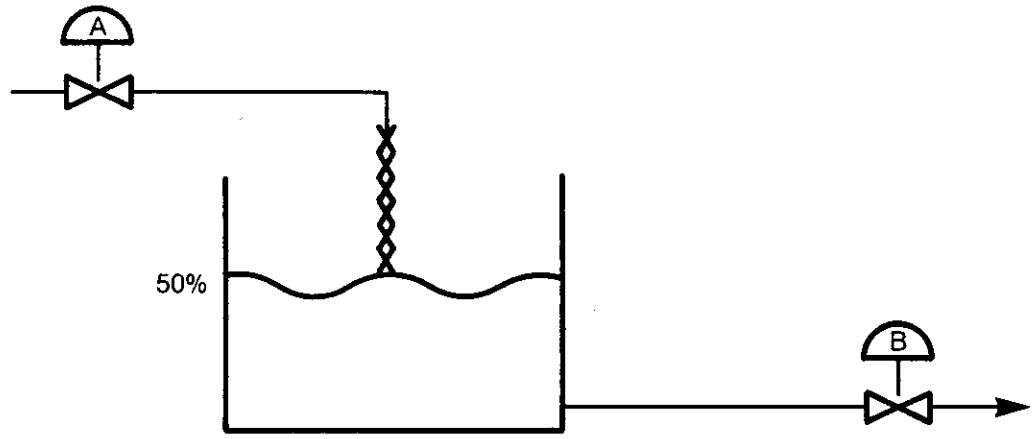
$$m = \frac{100}{P} e + b$$

where m = Output
P = Proportional Band
e = Error
b = Bias

Electronic Proportional Controller

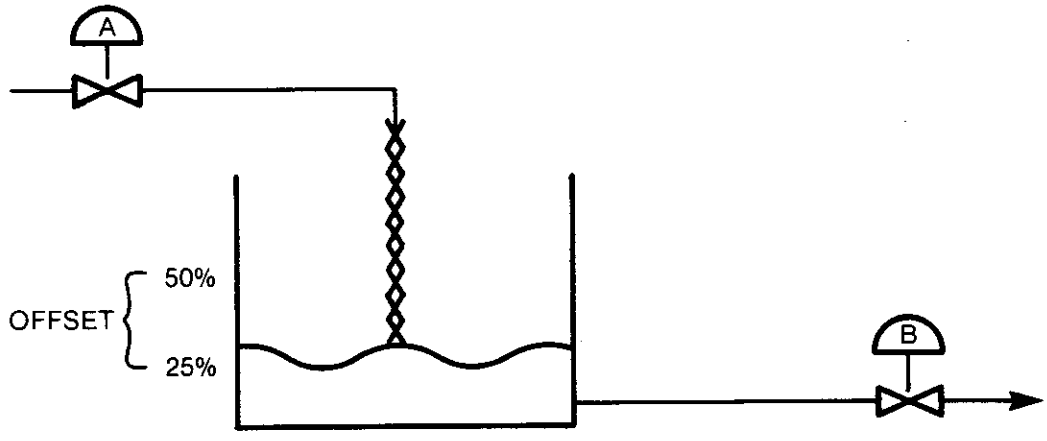


Integral Control Action



The above figure is a simplified illustration of the typical process loop diagram shown on Page 19-2. Assume: Both valves are open 50%
Tank level = 50%
Proportional control of valve "A"
Proportional band = 100 (gain = 1)
Loop is stable (liquid in = liquid out)

Now assume that the process demands more liquid and an operator opens valve "B" 75%. The level will begin to drop and the proportional action of controller will cause valve "A" to open 75%. Eventually the loop will stabilize, but at a lower level as shown below.



This difference in level is termed "offset".

Integral control action will add to the proportional action an additional amount based on the duration of the measurement deviation from set point. Integral control action (also referred to as "Reset" action) will eliminate offset.

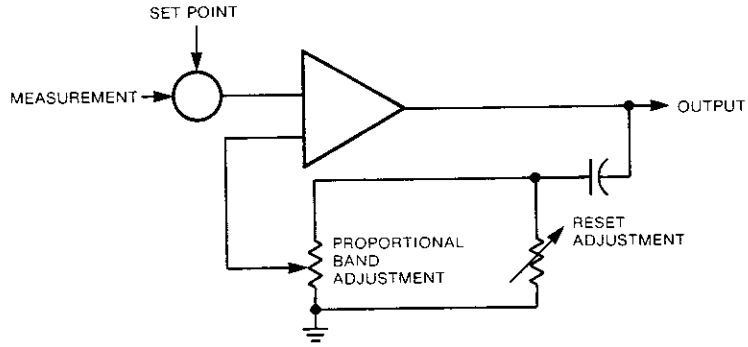
PID

Integral Control Action Formula

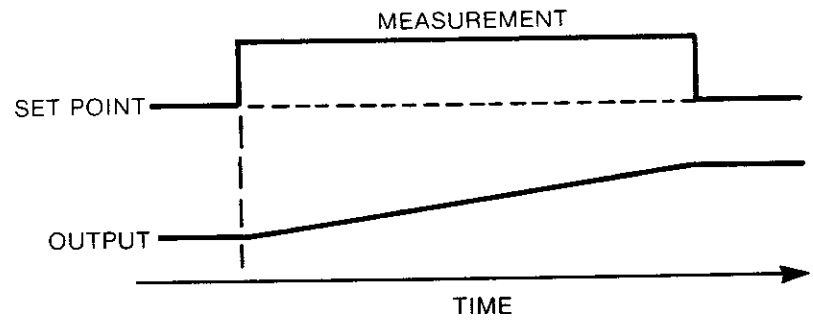
$$m = \frac{1}{R} \int e \, dt$$

where m = Output
R = Integral or reset term (controller time constant)
$\int$ = Integral
e = Error (measurement — set)
dt = Change in time

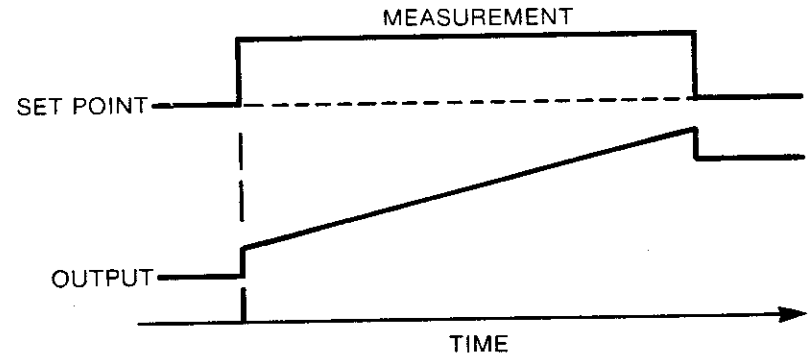
Electronic Proportional Plus Reset Controller



Integral Action Open Loop Response



Proportional Plus Reset Open Loop Response



Derivative Control Action

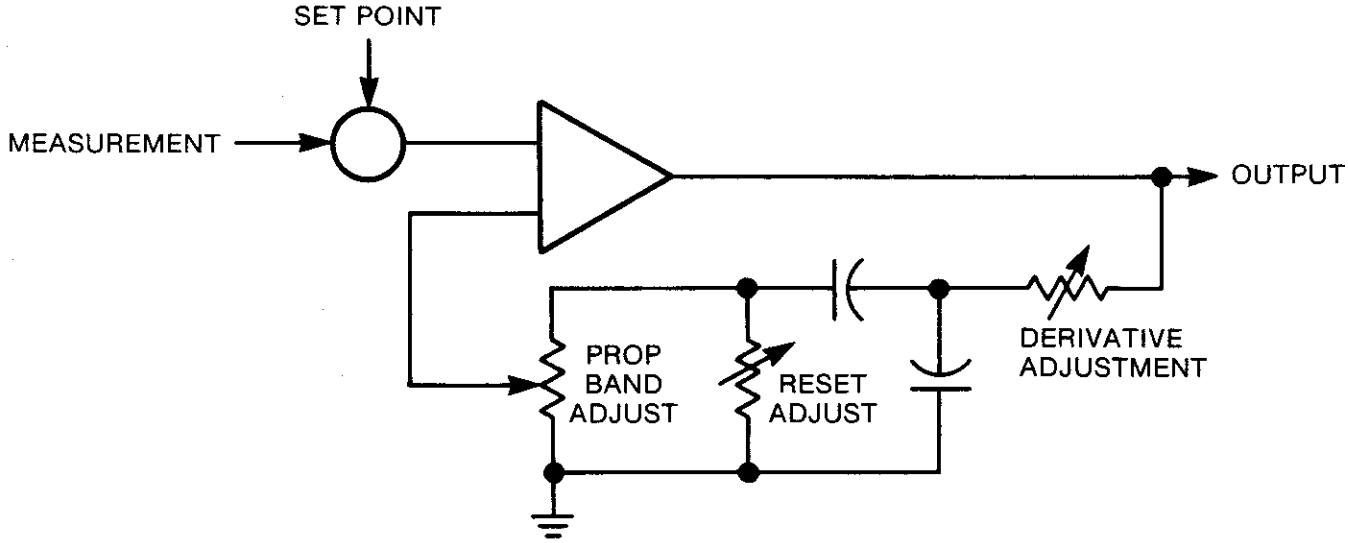
Derivative control action will add to the proportional action an additional amount proportional to the rate at which the measurement is changing.

Proportional-Derivative Control Action Formula

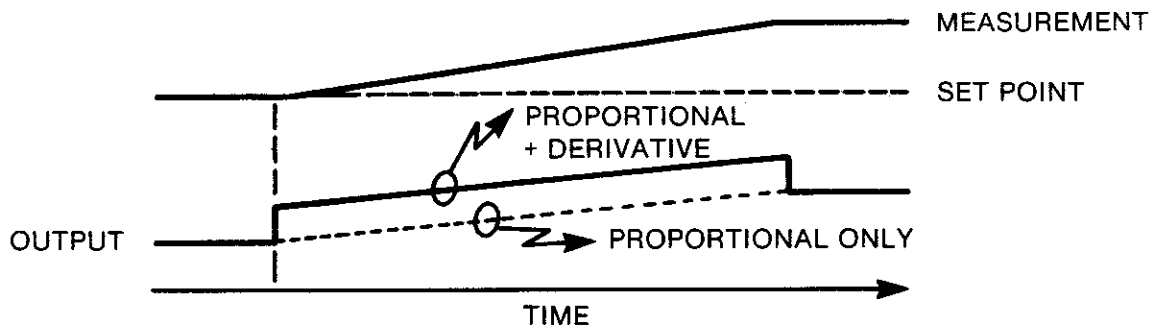
$$m = \frac{100}{P}\left(e + D\frac{de}{dt}\right) + b$$

where m = Output
P = Proportional band
e = Error (measurement — set)
D = Derivative term
de = Change in error
dt = Change in time
b = Bias

Electronic Proportional-Integral-Derivative Controller



Proportional-Derivative Open Loop Response



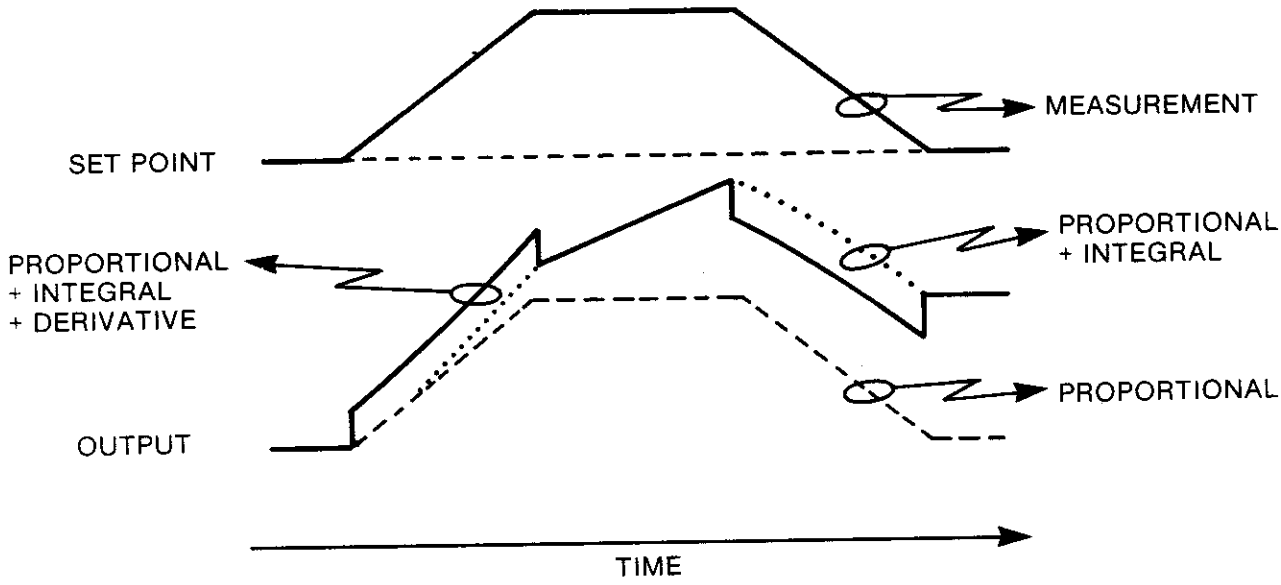
PID

Proportional-Integral-Derivative Formula

$$m = \frac{100}{P}\left(e + \int \frac{1}{R} e\,dt + D\frac{de}{dt}\right) + b$$

where
- m = Output
- P = Proportional band
- e = Error (measurement — set)
- R = Integral term
- $\int$ = Integral
- dt = Change in time
- D = Derivative term
- de = Change in error
- b = Bias

Proportional-Integral-Derivative Controller Open Loop Response



PID

Configuration

"T584-101 PID Module Program" tape (SON) contains approximately 448 words of additional software necessary to implement PID functionality. This tape must be loaded as part of the configuration procedure.

Adding PID Functionality to a New Controller

1. Load the configurator tape into the P190 and simultaneously press the 2 red keys, "INIT" and "INIT lock"
2. Enter the 584 unit I.D. # into the assembly register
3. Press "584 config"
4. Starting with the leftmost soft key, configure the system for skips, size, port parameters, ASCII, and specials (see Chapter 12 for details)
5. Press "next menu"
6. Press "modules"
7. Press "load modules"
8. Remove the configurator tape if you have not already done so, and load a PID software module SON tape, T584-101
9. Press "proceed"
10. Type "PID"
11. Press "load program"

PID Help Frames

To examine 7 screens of information about the PID block:

1. Type "PID"
2. Press "display help frame", screen 1 is displayed
3. Press "next screen", screen 2 is displayed
4. Continue pressing "next screen" until screen 7 is displayed
5. Press "directory display" to return to step 1

Pages 19-10 and 19-11 are print-outs of these help screens.

Adding PID Functionality to an Existing Program

Be certain to have a copy of your program on tape before starting this procedure!

1. Load the configurator tape into the P190, and simultaneously press the 2 red keys, "INIT" and "INIT lock"
2. Enter the 584 unit I.D. into the assembly register
3. Press "attach"
4. Press "exit"
5. Press "controller operations"
6. Press "stop 584"
7. Press "proceed"
8. Hold the "shift key" down, and press "reset"
9. Press "release 584"
10. Repeat step 2
11. Press "584 config"
12. Do steps 5 thru 11 from the "adding PID functionality to a new controller" above
13. Insert the tape loader tape and reload
14. Press "previous menu" twice
15. Press "write config"
16. Reload your traffic COP (see Chapter 12 for details)
17. Reload your user logic using the "relocate logic" function on the tape loader tape (see Chapter 20 for details)
18. Load a programmer tape to check the results of this procedure

PID SOFTWARE MODULE TAPE HELP SCREENS

584 PID Program Description Screen: 01

The 584 has been enhanced to provide a new positional, proportional-integral-derivative (PID) DX function. The 584 software provides features which include:

1. Bumpless/balanceless transfer
2. Direct/reverse action
3. Separate high/low alarm
4. Mode lock-out alarm
5. Programmable solution interval
6. Anti-reset windup protection
7. Scaled process variable and setpoint
8. Selectable maximum PID loops per scan
9. Tuning of gain, integral and derivative term

584 PID Program Description Screen: 02

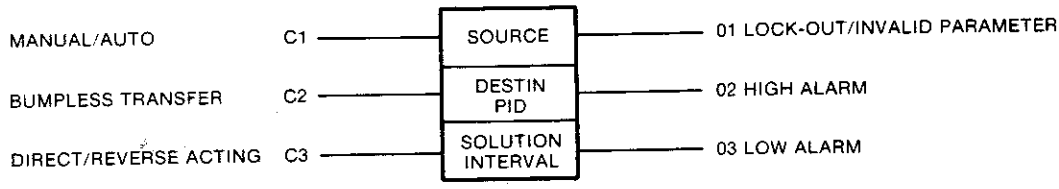
Hardware changes will not be required as long as Rev. G or later executive resides in the 584 IOP board. The initial release will be in the form of special P190 tape containing the 584 minicode PID function software that is loaded into 584 user logic memory. The user may specify a configuration with more or less functionality at the gain or loss of user logic and register space.

The maximum solution time will not exceed 1.5 milliseconds. Also, the PID minicode will not use more than 500 words of user logic memory.

Note that even though the PID node is not activated, it still takes time to analyze the high/low alarm and scaled PV. The effect will be from 0.3 ms to .5 ms per node. This virtually limits the number of PID nodes in a system.

584 PID Program Description Screen: 03

PID FUNCTION BLOCK



584 PID Program Description Screen: 04

PID Controls and Outputs

C1 = Off = Manual
On = Auto
C2 = Off = No operation
On = Bumpless transfer
C3 = Off = Direct
On = Reverse
01 = Off = Loop solving ok
On = Loop not solved, invalid parameter
02 = Off = No high alarm
On = High alarm
03 = Off = No low alarm
On = Low alarm

584 PID Program Description **Screen: 05**

Definition of Source Registers

Source = 16 consecutive 4XXXX registers to contain loop solution parameters
4XXX1 = scaled process variable
2 = scaled process setpoint
3 = output register (0-4095)
4 = high alarm limit
5 = low alarm limit
6 = PB proportional band (5-500 percent)
7 = K2 reset rate (0.00-99.99 repeats/minute)
8 = K3 derivative time (0.00-99.99 minutes)
9 = bias (0-4095)

584 PID Program Description **Screen: 06**

Definition of Source Registers

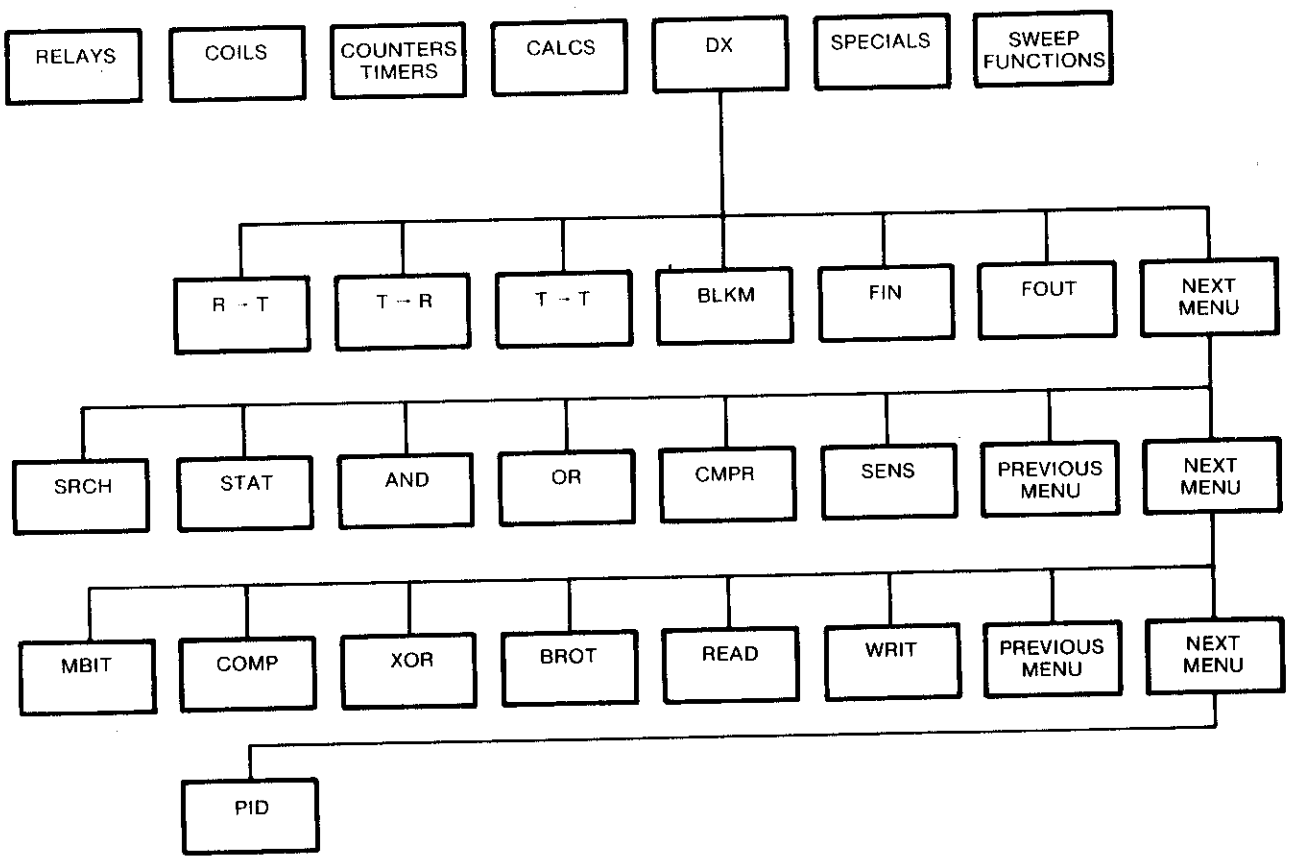
4XX10 = high integral windup limit (0-4095)
11 = low integral windup limit (0-4095)
12 = high engineering unit scale value (max=9999)
13 = low engineering unit scale value (min=0000)
14 = raw analog input (0-4095)
15 = offset to loop counter register
16 = max # loops solved per scan

584 PID Program Description **Screen: 07**

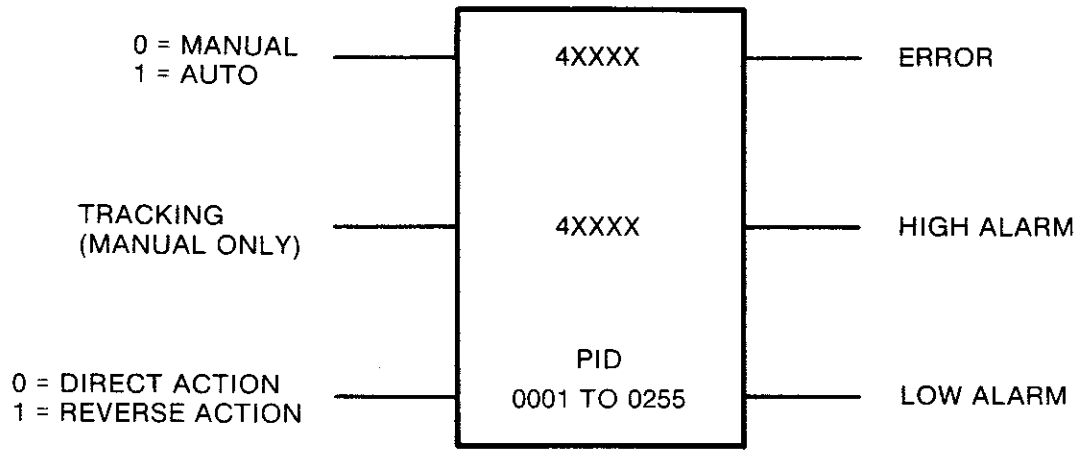
Definition of Destination Registers and Solution Interval

Destin = 5 consecutive 4XXXX registers to store calculation results for loop.
4YYY1 = initialization flag
4YYY2 = loop timer
4YYY3 = error summation hi-byte
4YYY4 = error summation lo-byte
4YYY5 = previous PV

Solution Time = interval for solution in 100 msec intervals, sampling time.
min = 0001 (0.1 sec)
max = 255 (25.5 sec)


RELAYS
COILS
COUNTERS TIMERS
CALCS
DX
SPECIALS
SWEEP FUNCTIONS
R → T
T → R
T → T
BLKM
FIN
FOUT
NEXT MENU
SRCH
STAT
AND
OR
CMPR
SENS
PREVIOUS MENU
NEXT MENU
MBIT
COMP
XOR
BROT
READ
WRIT
PREVIOUS MENU
NEXT MENU
PID

PID FUNCTION BLOCK



Function Block

The top node must be a holding register (4XXXX). This register is the first in a table of 16 holding registers, which are allocated as follows:

1. 4XXXX = scaled process variable
2. 4XXXX+1 = setpoint
3. 4XXXX+2 = output
4. 4XXXX+3 = high alarm limit
5. 4XXXX+4 = low alarm limit
6. 4XXXX+5 = proportional band
7. 4XXXX+6 = reset time constant
8. 4XXXX+7 = rate time constant
9. 4XXXX+8 = bias
10. 4XXXX+9 = high integral wind-up limit
11. 4XXXX+10 = low integral wind-up limit
12. 4XXXX+11 = high engineering range
13. 4XXXX+12 = low engineering range
14. 4XXXX+13 = analog measurement
15. 4XXXX+14 = pointer to loop counter register
16. 4XXXX+15 = maximum number of loops which can be solved per scan

The middle node must be a holding register (4YYYY). This register is the first in a table of 5 holding registers, which are allocated as follows:

4YYYY = loop status register
4YYYY+1 = loop timer register
4YYYY+2 = high order integral summation
4YYYY+3 = low order integral summation
4YYYY+4 = process variable - previous solution

The bottom node contains the symbol PID and a number from 1 to 255 which specifies how often the PID calculation should be done in tenths of seconds.

PID FUNCTION BLOCK

Inputs

Top: When powered, the PID block is in auto.
When not powered, the PID block is in manual. Scaling of the process variable and alarming are still done.

Middle: When powered in the manual mode (top input not powered), the transfer back to auto will be "bumpless".

Bottom: When powered, an increasing error will cause a decreasing output.
When not powered, an increasing error will cause an increasing output.

Outputs

Top: Error. This output will activate for the following reasons:

proportional band < 5 or > 500
integral setting > 9999
derivative setting > 9999
setpoint outside of specified engineering range
high engineering range $<$ or = low engineering range
high anti-reset wind-up limit = 0
block has not been solved for 2 X solution interval

Middle: High alarm

Bottom: Low alarm

PID

Top Node Register Usage

1. 4XXXX = Scaled process variable (engineering units)
   This register is loaded by the PID block every time it is solved. The scaled process variable is derived from a linear scaling performed on register 4XXXX+13 using the high and low engineering range entered in registers 4XXXX+11 and 4XXXX+12.
2. 4XXXX+1 = Setpoint (engineering units)
   The user must load this register with the desired setpoint, in engineering units. The setpoint must be greater than or equal to the low engineering range entered in 4XXXX+12 and less than or equal to the high engineering range entered in 4XXXX+11.
3. 4XXXX+2 = Output
   This register is loaded by the PID block every time it is solved. It is clamped to a range of 0 to 4095, which makes it compatible with analog output modules, B260, and B262. This register can be put directly in the traffic cop to cause the output to be sent to the analog output module without additional logic.
4. 4XXXX+3 = High alarm limit (engineering units)
   The user must load this register with the desired high alarm point in engineering units. This value should be above the setpoint and within the engineering range specified by registers 4XXXX+11 and 4XXXX+12.
5. 4XXXX+4 = Low alarm limit (engineering units)
   The user must load this register with the desired low alarm point in engineering units. This value should be below the setpoint and within the engineering range specified by registers 4XXXX+11 and 4XXXX+12.
6. 4XXXX+5 = Proportional band
   The user must load this register with the desired proportional band constant between 5 and 500. The smaller the number, the larger the proportional contribution to the output.
7. 4XXXX+6 = Reset time constant
   The user may load this register to add integral action to the calculation. The value entered ranges from 00.00 to 99.00 "repeats per minute". The larger the number, the larger the integral contribution to the output.
8. 4XXXX+7 = Rate time constant
   The user may load this register to add derivative action to the calculation. The value entered ranges from 00.00 to 99.99 inverse minutes. The larger the number, the larger the derivative contribution to the output.
9. 4XXXX+8 = Bias
   The user may load this register with a value from 0 - 4095, which will be added to the output.
10. 4XXXX+9 = High integral wind-up limit
    The user must load this register with the maximum value from 0 - 4095, which the output may be allowed to reach due to the integral contribution, in the presence of a continuous error.
11. 4XXXX+10 = Low integral wind-up limit
    The user must load this register with the minimum value from 0 - 4095, which the output may be allowed to reach due to the integral contribution, in the presence of a continuous error.

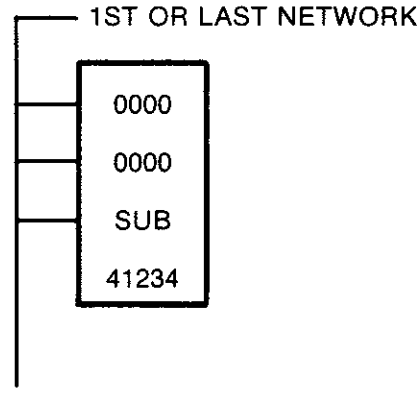
Top Node Register Usage

12. 4XXXX+11 = High engineering range
    The user must load this register with the highest value for which the measurement device is spanned ranging from 0 to +9999.

13. 4XXXX+12 = Low engineering range
    The user must load this register with the lowest value for which the measurement device is spanned ranging from 0 to +9998.

14. 4XXXX+13 = Raw analog measurement
    The user must load this register with the process variable (measurement). The PID block is designed to take a digitized (0 - 4095) measurement via a B243 A/D converter. If the input does not come from a B243 (e.g. a thermocouple), it must be scaled to 0 - 4095 before being loaded into this register.
    Non-linear inputs (e.g. D/P cells) must be linearized before they are loaded into this register.

15. 4XXXX+14 = Pointer to loop counter register
    NOTE: It may be helpful to read 16. 4XXXX+15 first.
    If the user limits the number of loops which are solved per scan by an entry other than zero in 4XXXX+15, then the holding register number which will count the number of loops solved per scan must be entered here. The actual number entered drops the prefix "4" from the holding register reference number (e.g. if 41234 is the register selected to hold this count, the number entered is 1234).
    The same number must be loaded into 4XXXX+14 of each PID block programmed.
    The register selected (e.g. 41234) must be zeroed at the beginning or end of each scan. This can be done via a subtract block:

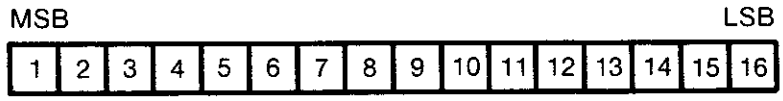


16. 4XXXX+15 = Maximum number of loops which can be solved per scan
    If an entry other than zero is made here, a register to hold the "number of loops solved" count must be specified in 4XXXX+14.
    The same number must be loaded into 4XXXX+15 of each PID block programmed.
    NOTE: If 4XXXX+15 = 0, 4XXXX+14 should = 0.

Middle Node Register Usage

In addition to the 16 registers used in the top node of the PID block, the 584 requires 5 additional registers to execute this function block. Their use is transparent to the user. The user does not load them and need not understand their function in order to use the PID block. The registers are used as follows:

1. 4YYYY = Loop status register
   This register is used to store loop status bits. It is cleared to zero on the first solution of the block following a start of the 584, or a transfer of the loop from manual to auto. At all other times it will have a non-zero number.
   The bits are assigned as follows:



   BIT 1: Status of top output (error)
   BIT 2: Status of middle output (high alarm)
   BIT 3: Status of bottom output (low alarm)
   BIT 4: This bit is set when:
      1. Loop is in auto and time since last solution is > or = the loop solution interval. It is set during the scan the loop is being solved.
      2. 4XXXX+15 is < or = loop counter register referenced by 4XXXX+14.
   BIT 5: Not used
   BIT 6: Loop is in auto, but is not being solved. Indicates the user has set a limit on the number of loops which can be solved per scan.
   BIT 7: The loop counter register 4XXXX pointed to by 4XXXX+14 is a valid register.
   BIT 8: Not used
   BIT 9: Not used
   BIT 10: Not used
   BIT 11: Not used
   BIT 12: Used to indicate negative numbers in various places in the PID algorithm.
   BIT 13: Status of bottom input (direct/reverse action)
   BIT 14: Status of middle input (tracking)
   BIT 15: Status of top input (auto/manual)
   BIT 16: Set after initial start-up or installation of loop. Clearing this bit will, on the next scan:
      1. Reset itself
      2. Move current value of real time clock into register 4YYYY+1
      3. Zero 4YYYY+2 and 4YYYY+3
      4. Load contents of 4XXXX+13 into 4YYYY+4

2. 4YYYY+1 = Loop timer register
   This register stores the "real time clock" reading from the 584 system clock each time the loop is solved. Elapsed time between the previous and current scans is calculated, and if the difference is > or = the solution interval, the loop should be solved on the current scan.

3. 4YYYY+2 = High order integral summation
   This register stores the high order 16 bits of the 32 bit sum created by the integral term.

4. 4YYYY+3 = Low order integral summation
   This register stores the low order 16 bits of the 32 bit sum created by the integral term.
5. 4YYYY+4 = Process variable - previous solution
   This register stores the raw analog input (4XXXX+13) each time the loop is solved. It is used in the derivative control mode to determine how fast the measurement is moving away from set point.

PID

Bottom Node Details

The loop solution interval is entered into this node in tenths of seconds ranging from 1 - 255 (1 = 0.1 seconds). The loop solution interval specifies how often the loop will be examined by the 584 (alarms checked, process variable scaled, and the PID calculation solved to update the output).

How often a loop must be solved is a function of what is being controlled (flow, temperature, pressure, PH, etc.), and how fast the final control element (valve, electric heater, etc.) can respond to a change in the PID blocks output. For example, a servo motor can respond in milliseconds, while a pneumatic valve is much slower, and an electric valve is vastly slower with response time measured in seconds.

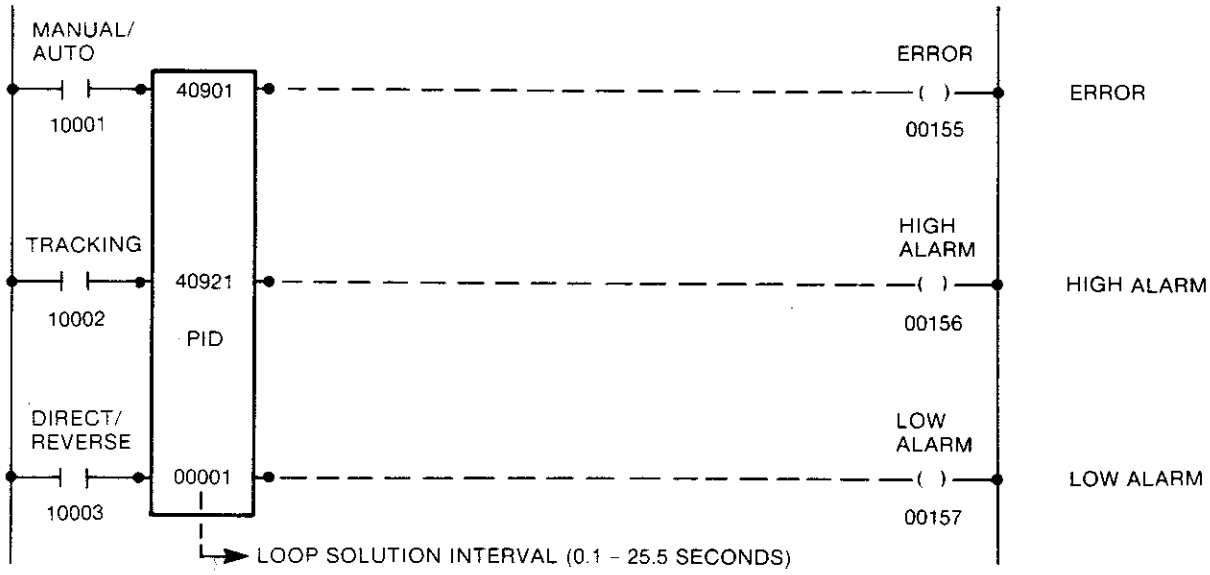
In addition, the process under control may require very little time to respond to a change in the process variable (e.g. PH), or may take much longer (e.g. steam heating a 50,000 gallon tank).

If the loop solution interval is too short, the loop may constantly overcompensate. If the interval is too long the system may not respond to upsets fast enough. Either of these conditions may result in unstable, unsafe or, otherwise unsatisfactory control.

In general, no process control application will require solution intervals of less than every 0.5 seconds (solution interval = 5), and 85% of all process applications will require intervals of 1 second (solution interval = 10). Most temperature loops can be controlled with solution intervals of 1.5 to 5 seconds.

PID (PROPORTIONAL-INTEGRAL-DERIVATIVE) BLOCK

Network #44



Register	Description	Range	Comments
40901	Scaled process variable	Engineering units	
40902	Set point	Engineering units	*
40903	Output	0-4095	
40904	High alarm limit	Engineering units	*
40905	Low alarm limit	Engineering units	*
40906	Proportional band	5-500	*
40907	Reset term	0-99.99 repeats/min	**
40908	Derivative term	0-99.99 inverse min	**
40909	Bias	0-4095	**
40910	High integral wind-up limit	0-4095	*
40911	Low integral wind-up limit	0-4095	*
40912	High engineering units	Engineering units	*
40913	Low engineering units	Engineering units	*
40914	Analog input	0-4095	*
40915	Pointer to loop counter register	XXXX of valid 4XXXX	**
40916	# PID loops solved per scan	0-XXX	**
40921	Reserved for 584		
40922	Reserved for 584		
40923	Reserved for 584		
40924	Reserved for 584		
40925	Reserved for 584		

* = Mandatory entry, loop will not function without this parameter
** = Optional entry

Tuning

The gain and time functions of each PID loop must match the rest of the elements in the loop (transmitter, valve, application, etc.). This is accomplished by "tuning" the loop. A 584 PID block is tuned by loading the appropriate values into registers 4XXXX+5 (proportional band), 4XXXX+6 (integral term), 4XXXX+7 (derivative term), and 4XXXX+8 (bias).

The following general procedures apply:

Proportional:

1. Select manual mode
2. Proportional band = 500
3. Select auto mode
4. Change the set point
5. Note the resultant measurement cycle
6. Reduce the proportional band and repeat steps 4 & 5
7. Repeat steps 4, 5, & 6 until loop cycles

Integral:

1. Select manual mode
2. Proportional band = 500
3. Integral term = 99.99
4. Select auto mode
5. Adjust proportional band per proportional tuning procedure
6. Decrease integral term until loop cycles
7. Increase integral term until cycling stops

Derivative:

1. Select manual mode
2. Proportional band = 500
3. Integral term = 99.99
4. Derivative term = 00.00
5. Adjust proportional band per proportional tuning procedure, however, loop should cycle slightly
6. Increase derivative term until cycling stops
7. Narrow proportional band until cycling starts again
8. Repeat steps 6 & 7 until further increases in the derivative term fail to stop the cycling
9. Widen the proportional band until cycling stops
10. Set integral term = derivative term

General Tuning Guidelines

Application	Proportional Band	Reset	Derivative
Flow	High	Fast	Never
Level	Low	Capacity Dependent	Rarely
Temperature	Low	Capacity Dependent	Usually
Pressure	Low	Usually Fast	Sometimes
Analytical	High	Usually Slow	Sometimes