

PID loops in KMC controllers

Contents

Overview of loop operation	2
Inside the loop	3
Loop terms and constants	3
Loop algorithms	5
Integral windup	6
Loop action	6
Execution of loops in KMC BACnet controllers	7
Proportional and gain	7
Integral	7
Derivative	7
Loop output	7
Execution of loops in KMD controllers	8
Proportional	8
Integral	8
Derivative	8
Loop output	8
Tuning PID loops in BACnet controllers	9
Monitoring the loop	9
Choosing starting points	9
Tuning the loop	10
Tuning graphs	11

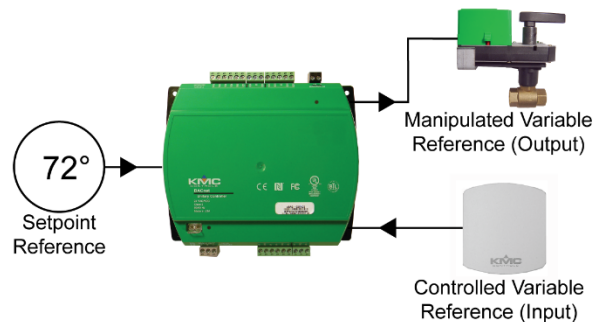
Overview of loop operation

A PID loop (Proportional, Integral, Derivative) is a control strategy used in many types of process control systems. In building automation systems, PID loops are used to maintain precise control of temperature, pressure, flow, or any other physical property within a system. A PID loop, also referred to as a control loop, consists of physical components—sensors, dampers, valves, etc.—and a loop controller. In BACnet devices, the loop controller takes the form of a loop object within a BACnet controller. In KMDigital devices, the loop controller is referred to as a Controller point.

The following diagram is a BACnet control loop that is typically found in a fan coil unit. In this loop, zone temperature is controlled with a valve that is opened or closed to change water flow through a coil. Air flowing through the coil is warmed or cooled to raise or lower the temperature in the room or zone.

- The zone setpoint is represented by the BACnet loop object's Setpoint Reference property and is fixed within the controller at 72°.
- The output of the loop controls the valve through the BACnet loop object's Manipulated Variable Reference property.
- The zone temperature sensor is represented to the loop with the BACnet loop object's Controlled Variable Reference property and provides feedback to the loop.

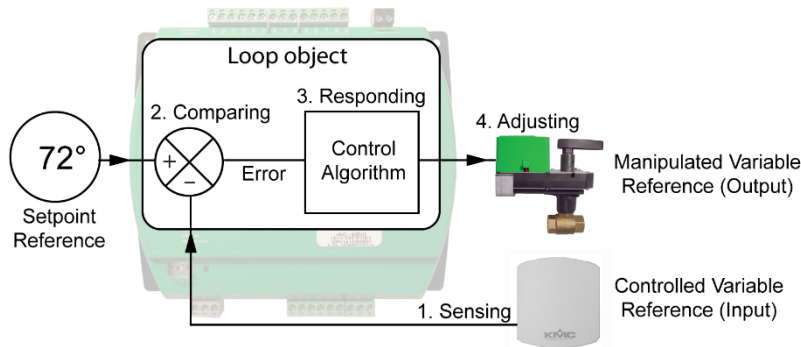
When the zone temperature does not match the setpoint, the loop commands the valve to open or close until the zone temperature matches the setpoint.



Inside the loop

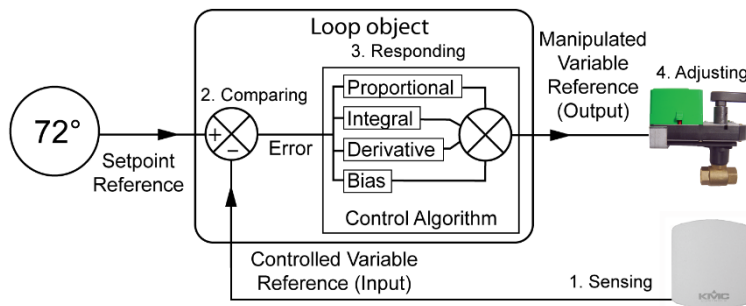
The loop maintains control of a flow, pressure, temperature, etc. with a continuous four-step process.

1. Sensing—A condition (zone temperature, air flow, duct pressure, etc.) is measured using the appropriate sensor for the process.
2. Comparing—The measurement of the current condition is compared to the setpoint. The setpoint may be a fixed value maintained in either the loop object or value object or from a user input device.
3. Responding—The control algorithm within the loop object reacts to any error that may exist by generating a corrective control signal.
4. Adjusting—The corrective control signal commands a device such as a valve, heater, damper, etc. to produce a change in the current condition.



Loop terms and constants

Within the control algorithm, a PID loop uses four terms—proportional, integral, derivative, and bias—to apply control to the output. In the example block diagram, the control algorithm continuously calculates an error as the difference between the values of the Setpoint Reference and the Controlled Variable. Next, the algorithm applies a correction based on the four loop terms. The terms are controlled in software with a BACnet constant property of the same name. Then, the Manipulated Variable Reference adjusts the position of the control valve based on the weighted sum of the control terms.



Proportional The proportional term of the control algorithm provides a linear gain to the loop. The loop error—the value of Setpoint Reference minus the value of the Controlled Variable Reference—is multiplied by the proportional value to yield an output value represented by Manipulated Variable Reference. For example, if the error is large and positive, the control output will be proportionately large and positive.

One way of looking at the proportional term is that it sets the controlled range or band. In the example diagram, a proportional constant of 4 sets the loop to control the valve over a band of 4 degrees. A Setpoint Reference of 74 degrees and a Controlled Variable of 72 degrees results in an error of 2 degrees. The error of 2 degrees sets the valve to $2/4$ or 50% of the proportional band. An error of 3 degrees results in an error of $3/4$ or 75% of the proportional band.

The proportional term and its effects on the loop output are shown in the following equation:

$$\text{Proportional term} = \frac{100 * e(t)}{Kp}$$

$$Kp = \text{proportional constant}$$

$$e(t) = \text{error that changes with time} = \text{setpoint} - \text{controlled variable}$$

Integral The integral term—also called “resets” or “repeats”—of the control algorithm provides loop correction based on the history of the loop error. As time progresses, the integral constant is multiplied by the error and the amount of time that has elapsed. The integral drives the steady-state error to zero. The steady-state error is the error that remains when the proportional term alone no longer provides a significant effect on the loop output.

The integral constant is set in units of either repeats per minute or repeats per hour. When the integral constant is set to 10 repeats per minute, the controller adds the proportional error term to the output every 6 seconds (60sec/min / 10/min).

The following equation shows the effect of the Integral term.

$$\text{Integral term} = \frac{100}{Kp} Ki \int e(t) dt$$

$$Ki = \text{integral constant}$$

$$dt = \text{elapsed delta time}$$

Derivative The derivative term—sometimes referred to as “rate”—provides a time-derived correction to the effects of the proportional and integral components. If left to proportional and integral correction alone, the output of the PID loop might greatly overshoot the intended setpoint. The

derivative component acts to slow, or dampen, the rate of correction value based on the slope change in error.

$$Dterm = \frac{100}{Kp} Kd \frac{de}{dt}$$

$Kd = \text{derivative constant}$

$dt = \text{elapsed delta time}$

$de = \text{delta error}$

Note that derivative is typically used only in systems without time lags and seldom is used in HVAC systems. If there is a time delay, the control loop will be unstable and will perform better without derivative correction.

Bias The bias component of the PID algorithm provides a steady offset to the correction value of the PID loop. After an error is driven to zero, the output value may still be required to be moved away from zero, which can be done with the bias term. Bias is important in proportional-only systems when the proportional term alone cannot maintain an error of zero.

Loop algorithms

The PID loop algorithm for KMC BACnet controllers adds the error, integral, and derivative terms. The sum of the terms is then multiplied by the gain term, and the bias term is then added to the product. This is shown in the following equation:

$$\text{Output} = G \left[e(t) + Ki \int e(t) dt + Kd \frac{de}{dt} \right] + B$$

$$G = \text{gain} = \frac{100}{Kp}$$

$e(t) = \text{error that changes with time}$

$Kp = \text{proportional constant}$

$Ki = \text{integral constant}$

$Kd = \text{derivative constant}$

$B = \text{Bias}$

$dt = \text{elapsed delta time}$

$de = \text{delta error}$

The KMD calculation for a PID loop is similar, but the gain term is applied only to the error term as shown in the following equation:

$$Output = Ge(t) + Ki \int e(t)dt + Kd \frac{de}{dt} + B$$

Integral windup

A common problem in PID loops is integral windup. Windup takes place in a loop when a large change in setpoint causes the integral term to accumulate a significant error during the change. This large error results in output overshoot. If left unchecked, the integral term will continue to accumulate and drive the loop well beyond the setpoint before it begins to decrease or “unwind”.

In KMC controllers, the integral term is limited to prevent windup. For a positive error, integral is limited to a value which, when added to the proportional term and bias, will equal the maximum output (100%).

$$Integral\ limit = 100 - Proportional\ term - Bias$$

For a negative error, the integral term can decrease to zero, but cannot be negative.

Loop action

Loop action controls the direction the manipulate variable moves when there is a difference between the controlled variable and the setpoint.

Direct-action loops are typically used for cooling systems. In a direct-action loop, when the value of the Controlled Variable Reference property rises above the value of the Setpoint Variable, the controller increases the value of the Manipulated Variable Reference property. In the fan coil unit example, the valve increases the flow of chilled water when the zone temperature rises above the value of the setpoint.

Reverse-action loops are typically used for heating systems. In a reverse-action loop, when the value of the Controlled Variable Reference property rises above the value of Setpoint Variable, the controller decreases the value of the Manipulated Variable Reference property. In the fan coil unit example, the valve decreases the flow of hot water when the zone temperature rises above the value of the setpoint.

Execution of loops in KMC BACnet controllers

The firmware in a KMC BACnet controller calculates the terms of a loop object once per second. When the calculations are complete, the output of the loop is updated.

Proportional and gain

The proportional constant is used to calculate the gain of the loop. Gain is used in the final calculation of the loop output.

$$\text{gain} = 100 * \text{error} / \text{proportional constant}$$

Loop error is calculated and used in several parts of the loop calculation. It is defined as follows:

$$\text{error} = \text{controlled variable reference} - \text{setpoint reference}$$

Integral

The integral term is calculated in either repeats per minute or repeats per hour.

Integral term, repeats per hour

$$\text{integral term} = \text{previous integral term} + \text{action} * \text{error} * \text{integral constant} * \text{time} / 3600$$

Integral term, repeats per minute

$$\text{integral term} = \text{previous integral term} + \text{action} * \text{error} * \text{integral constant} * \text{time} / 60$$

$$\text{previous integral term} = \text{integral value calculated from previous loop calculation}$$

$$\text{time} = \text{seconds between loop calculations}$$

Derivative

$$\text{derivative term} = \text{action} * \text{derivative constant} * (\text{error} - \text{previous error}) * 60 / \text{time}$$

$$\text{time} = \text{seconds between loop scans}$$

$$\text{derivative constant} = \text{rate constant minutes}$$

$$\text{previous error} = \text{error from previous loop calculation}$$

Loop output

When the calculation of each term is complete, output of the loop is then calculated.

$$\text{output} = \text{gain}(\text{error} + \text{integral} + \text{derivative}) + \text{bias}$$

The BACnet Present Value property of the loop is set equal to the output of the calculation. If an object is selected for Manipulated Variable Reference, the Present Value property of that object is also updated. Control Basic can also read the value of the output of the loop. Complete details for using Control Basic can be found in the Help and software manuals supplied with the software from KMC Controls, Inc.

Execution of loops in KMD controllers

The firmware in a KMD controller calculates the terms of a loop point once per second. When the calculations are complete, the output of the loop is updated.

Proportional

$$\text{proportional term} = \text{action} * 100 * \text{error} / \text{proportional constant}$$

$$\text{error} = \text{input} - \text{setpoint}$$

$$\text{action} = 1 \text{ for direct acting loops and } -1 \text{ for reverse acting loops}$$

Integral

The integral term is calculated in either repeats per minute or repeats per hour. In KMD controllers, the integral constant is labeled as "reset".

Integral term, repeats per hour

$$\text{integral term} = \text{previous integral term} + \text{action} * \text{error} * \text{reset} * \text{time} / 3600$$

Integral term, repeats per minute

$$\text{integral term} = \text{previous integral term} + \text{action} * \text{error} * \text{reset} * \text{time} / 60$$

$$\text{previous integral term} = \text{integral value calculated from previous loop calculation}$$

$$\text{time} = \text{seconds between loop calculations}$$

Derivative

The derivative constant is labeled "rate" in KMD controllers.

$$\text{derivative term} = \text{action} * \text{rate} * (\text{error} - \text{previous error}) * 60 / \text{time}$$

$$\text{time} = \text{seconds between loop scans}$$

$$\text{derivative constant} = \text{rate constant minutes}$$

$$\text{previous error} = \text{error from previous loop calculation}$$

Loop output

The output of the loop is then calculated by adding together the results of the calculation for each term.

$$\text{output} = \text{proportional} + \text{integral} + \text{derivative} + \text{bias}$$

The output of the loop is used by Control Basic to assign a value to another point within the KMD controller. Complete details for using Control Basic can be found in the Help and software manuals supplied with the software from KMC Controls, Inc.

Tuning PID loops in BACnet controllers

This topic covers tuning a BACnet PID loop under actual operating conditions by following a three-part process.

1. Setting up monitoring
2. Choosing starting points
3. Tuning the loop by first adjusting the Proportional constant and then adding and adjusting the Integral constant.

Monitoring the loop

To accurately tune a loop, the loop performance must be observed under operating conditions. Use the Trend Viewer in KMC Connect or TotalControl and add the following loop properties to the view.

- Loop output
- Setpoint Reference
- Manipulated Variable

Typically, the Logging Rate in Trend objects or the Update Interval in the Trend Viewer will be set to one minute or less during tuning.

Alternative viewing methods use either a Trend Log Multiple object or three Trend Log objects. For the purpose of tuning a loop, the Trend Log and Trend Log Multiple objects do not have to permanently store the results in the database.

Choosing starting points

Because this method of tuning a loop requires observing loop behavior while controlling operating equipment, you will need starting points for the Proportional and Integral constants.

Proportional constant Start with the controlled range (throttling range) of the Controlled Variable property. Several starting points are shown in the following chart.

Controlled Variable	Proportional Constant	Integral Constant	Integral Units (Repeats)	Notes
Room Temp	2	.08	Hour	4 cycles/hour
MAT/Discharge air	10	3	Hour	30 cycles/hour
CHW	10	1.0+	Hour	10 cycles/hour
HW	15	1.0+	Hour	15 cycles/hour

Integral constant A starting Integral constant can be calculated from the Proportional constant and the expected number of times (Setpoint Cycles) that the Controlled Variable is expected to reach setpoint in an hour or minute.

$$\text{Starting Integral} = \frac{P * \text{Setpoint Cycles}}{100}$$

Note: HVAC systems typically do not use the Derivative constant.

Tuning the loop

To tune the loop, the controller must be connected to the controlled equipment. The equipment must be running under normal operating conditions and changes must be allowed. In addition, the controller must be connected to a BACnet operator workstation such as TotalControl or KMC Connect.

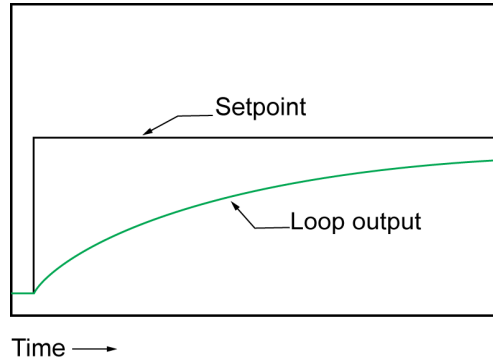
1. Install the controller and connect it to the controlled equipment.
2. Connect to the controller with TotalControl or KMC Connect.
3. Set up monitoring as previously described. At a minimum, monitor the loop output—the Present Value property—and Setpoint Reference properties in the loop.
4. Configure the Loop object.
 - Enter only the starting Proportional Constant.
 - Set the Bias, Integral, and Derivative constants to zero.
5. Make a change to the Setpoint Reference and monitor the effects on the loop output.
 - Proportional too small: The loop output will not reach setpoint. (Graph A)
 - Proportional too large: The loop output will oscillate. (Graph B)
 - Proportional correct: The loop output becomes stable after a few cycles of oscillations but may or may not reach setpoint. (Graph C)
6. If needed, adjust the Proportional Constant and repeat the previous step until the loop output becomes stable. (Graph C)
7. Enter the starting Integral Constant and Integral Units. Only Minutes or Hours can be entered for Integral Units.
8. Make a change to the Setpoint Reference and monitor the change to the loop output. See the graphs
 - Integral too small with stable Proportional (Graph C)
 - Integral too large, the loop output is ringing, or oscillating. (Graph B)
 - The loop output reaches setpoint and becomes stable after a few oscillations. (Graph D)

9. If needed, adjust the Integral constant and repeat the previous step until the loop output becomes stable. (Graph D)
10. When the loop is stable, record the results and save the trend information for reference.

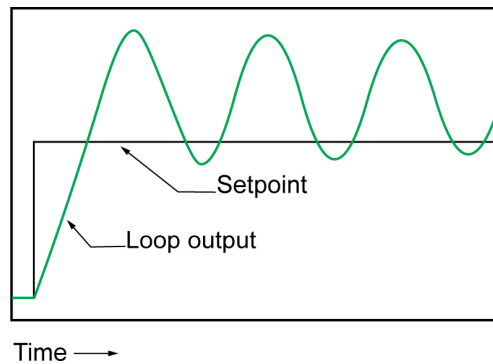
Tuning graphs

Use these graphs in conjunction with the previous procedure to evaluate loop performance.

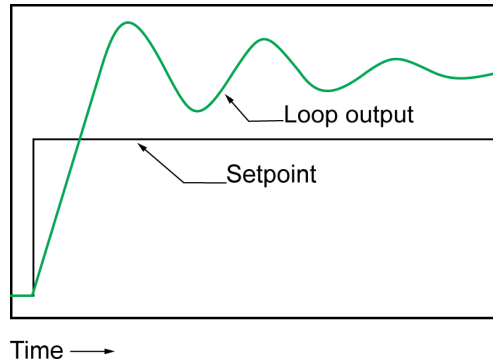
Graph A The loop is too slow and cannot reach setpoint because the Proportional constant is too large. Decrease the Proportional constant.



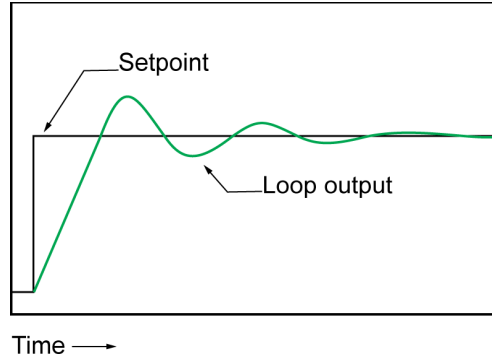
Graph B The loop is unstable as shown by the continuing oscillations (ringing). Increase the Proportional constant. If the loop was previously stable with only Proportional, increase the Integral constant.



Graph C The output oscillations decrease after a few cycles. The loop is stable but is not at setpoint.



Graph D The loop is stable. The output quickly reached setpoint with only a few, small cycles of ringing.



KMC Controls, Inc.

19476 Industrial Drive

New Paris, IN 46553

877.444.5622

Fax: 574.831.5252

www.kmccontrols.com