CONTROLSOFT
ADVANCED PROCESS CONTROL MADE EASY

PID Loop Tuning
Pocket Guide

TABLE OF CONTENTS

VERSION 8

PUBLICATION **DS405Hb-EN**

Process Definitions

Common terms used in process control.

CONTROLLER OUTPUT (CO) | CONTROLLER VARIABLE (CV)

Output from controller to control mechanism (e.g., valve position, pump stroke or speed). Can be 0-100% or time proportional output (on/off).

PROCESS VARIABLE (PV)

Value of interest for Control (e.g., temperature, flow, level, pressure).

SETPOINT (SP)

Desired value of the Process Variable.

ERROR

Difference between the PV and the SP.

DEADTIME (DT)

Measure of how long a process takes to start responding to an output change (i.e., the elapsed time between the application of the step input and the time that the output first begins to change from its initial value).

PROCESS GAIN (G) | MODEL GAIN (MG)

Measure of how much a process changes as a result of an output change (PV change per unit of CO change).

LAG TIME CONSTANT (LAGTC)

Measure of how long a process takes to reach steady state after deadtime. Process reaches 63% of steady state after one time constant, 95% after three time constants. The steady state for non-integrating processes is the final steady level of process variable. The steady state for integrating processes is the final steady rate of change of the level of process variable.

PID Controller Terms

A PID controller continuously calculates an error value as the difference between a measured process variable and a desired setpoint. By tuning three parameters, a PID controller can address specific process requirements.

PROPORTIONAL TERM (P)

The amount added to the output based on the current error.

Tuning parameters for Proportional Term are:

- **Proportional Gain** is a Multiplier.
  If the error is 10% and the Gain is 0.8, then output will move 8%
- **Proportional Band** is a Divider as a percentage.
  If the error is 10% and the Band is 125, then output is (10% x (100/125))=8%
- **Conversion between P-Gain and P-Band:** P-Band=100/P-Gain

INTEGRAL TERM (I)

The amount added to the output based on the sum of the error.

Tuning parameters for Integral Term are:

- **Time Constant** is the time for one full repeat of P-Term.
  If the P-Term is 8% and the Time Constant is 10 seconds, then the output will ramp up 8% every 10 seconds
- **Inverse Time Constant** is the amount the output will move in one second.
  If the P-Term is 8% and the Reset Rate is .1 repeat/sec, then the output will move 0.1x 8% every second and take 10 seconds for the full repeat of the P-Term of 8%
- **Reset Rate (I-Gain)** is the same as Inverse Time Constant multiplied by P-Gain.
- **Conversion between Time and Reset Rate:** Inverse Time Constant = 1/ Time Constant, Reset Rate = (1/Time Constant) x P-Gain.

DERIVATIVE TERM (D)

The amount added to the output based on the rate of change of the error.

Tuning parameters for Derivative Term are:

- **Time Constant** is the amount of time the controller will look forward.
- **Derivative Gain** is the amount of time the controller looks forward multiplied by the P-Gain.

Types of PID Loops

FAST LOOPS (flow, pressure)

P Little (too much will cause cycling)
I More
D Not needed

SLOW LOOPS (temperature)

P More
I Some (too much will cause cycling)
D Some

INTEGRATING LOOPS (level & insulated temperature)

P More
I Little (will cause cycling)
D Must (If D is not used, the loop will cycle)

NOISY LOOPS (measurement constantly changing)

P Low (will cause cycling)
I Most (accumulated error)
D Off (will cause cycling)

Process Types

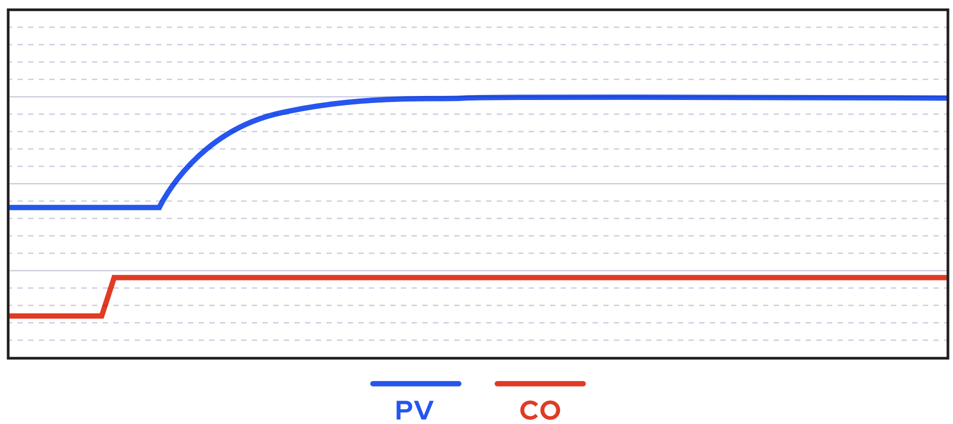
The two most common process types in manufacturing processes are **non-integrating** and **integrating processes**.

NON-INTEGRATING PROCESS

In a non-integrating process, when the Controller Output changes from one steady-state value to another, the Process Variable undergoes a transient phase but ends up at a new steady-state value (see Figure 1). Example processes include: pressure, flow, and some temperature control.

Be careful when making this classification, however, as a temperature control process, for example, can behave as an integrating loop in some applications. Thus, it is a good idea to check the response characteristics of the process before making a definite decision on its type.

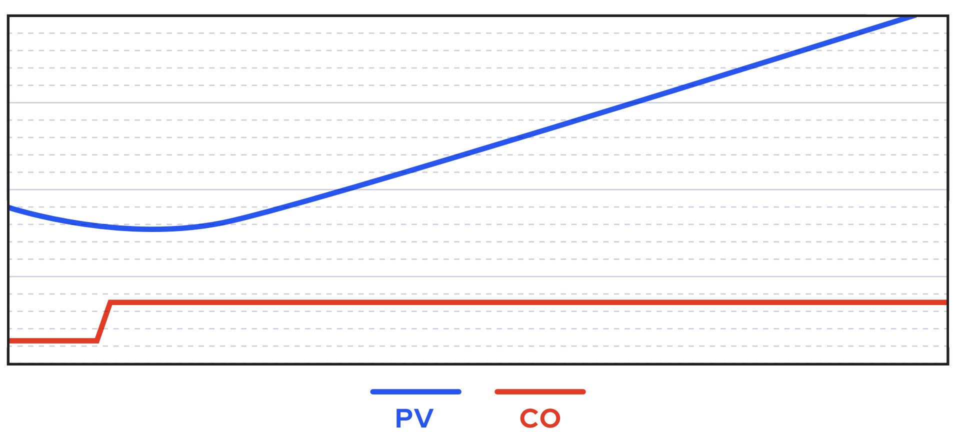
Non-Integrating Process (Figure 1)



INTEGRATING PROCESS

In an integrating process, a steady-state change in the Controller Output causes the Process Variable to increase (or decrease) steadily until it reaches its upper (or lower) limit (see Figure 2). Example processes include: level, well-insulated temperature.

Integrating Process (Figure 2)



Tuning Bump Test Methods

The best way to learn about the behavior of a process is to perform a bump test. The bump test establishes the cause-and-effect relationship between a control loop's Process Variable (PV) and Controller Output (CO). Without a clear understanding of the PV-CO relationship, it's difficult to generate tuning parameters.

Bump tests involve a change in CO in order to determine how the PV responds and reveal how far, how fast, and with how much delay the PV responds to a given change in CO. With an established PV-CO relationship, it's possible to calculate model parameters and to convert those values into tuning parameters.

This guide covers two types of bump tests:

→ **OPEN LOOP BUMP TEST (I.E., MANUAL MODE)**
Applies to any type of feedback controller, including both PID and non-PID controllers.

→ **CLOSED LOOP BUMP TEST (I.E., AUTOMATIC MODE)**
Applies to PID controllers.

OPEN LOOP BUMP TEST
(RECOMMENDED METHOD)

1 Know the Process

Identify the loop you intend to tune and determine the speed of the loop.

FAST LOOP	Response time < 1s to about 10s (e.g., a flow loop). Use of PI controller is sufficient.
MEDIUM LOOP	Response time of several seconds up to about 30s (e.g., flow, temperature, and pressure loops). Use either PI or PID controller.
SLOW LOOP	Response time > 30s (e.g., many temperature loops and level loops). Use of PID controller is recommended.

2 Know the Controller

Identify the units of your PID controller:

PROPORTIONAL TERM (P-TERM)
Tuning parameter is either a Proportional Gain (P-Gain) or Proportional Band (P-Band).

INTEGRAL TERM (I-TERM)
Tuning parameter can be a Time Constant (in minutes or seconds), Reset Rate (1/second or 1/minute), or I-Gain (Reset rate x Proportional gain).

DERIVATIVE TERM (D-TERM)
Tuning parameter can be Time Constant (in seconds or minutes) or Derivative Gain (derivative time constant x proportional gain).

In this document, Proportional Gain, Integral Time Constant (in seconds), and Derivative Time Constant (in seconds) are assumed. Convert back to your controller units if necessary.

The PID Controller Reference (starting on Page 10) is a summary of many common controllers available on the market. ***For units of other controllers, please contact ControlSoft.***

CONTINUED

OPEN LOOP BUMP TEST

(RECOMMENDED METHOD)

3 Test the Loop

NON-INTEGRATING PROCESSES

a. Put loop in manual control mode, keep control output constant, and wait for process to stabilize.

b. Make a small step change on control output (say 5 or 10%), and watch the response.

c. Estimate process model according to Figure 3:

Model Gain = PV change / CO change

Deadtime = time lapse between change of CO and observable changes on PV

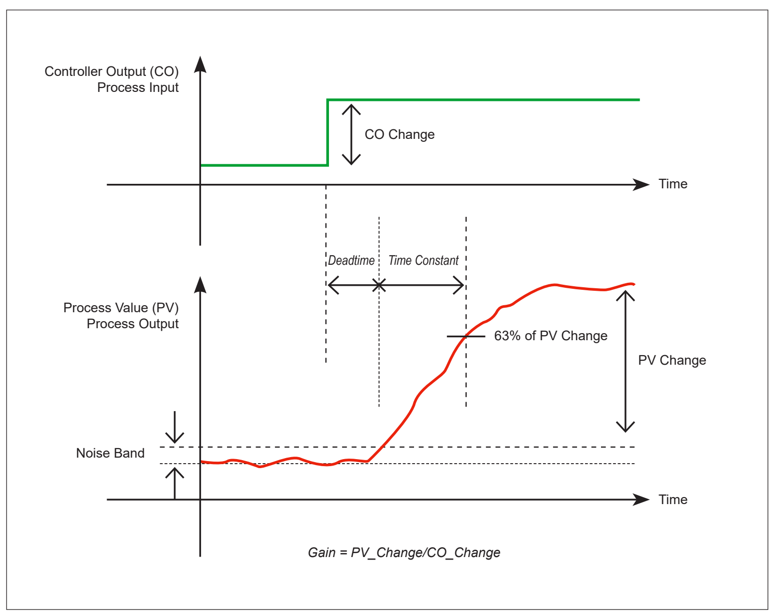
Time Constant = the time it takes for PV to reach about 63% of total changes

d. Select initial PID values using applicable equation:

Non-Interacting (ISA) or Series	Parallel
P_{NI} = 1 / Model Gain I_{NI} = (Deadtime + Time Constant) D_{NI}[1] = (Deadtime / 3) or (Time Constant / 6)	$P_P = P_{NI}$ $I_P = P_{NI} / I_{NI}$ $D_P = P_{NI} \times D_{NI}$

[1] For slow loop, select whichever is greater. For fast loop, select smaller value.

Non-Integrating Process Model (Figure 3)



OPEN LOOP BUMP TEST

(RECOMMENDED METHOD)

3 Test the Loop

INTEGRATING PROCESSES

a. Put loop in manual control mode, keep control output constant, wait for a steady slope in PV.

b. Make a small step change on control output (say 5 or 10%) and watch the response. Slope of PV changes. Place a close watch on PV, integrating nature of the process causes PV to continuously change. Stop the test once enough data is collected.

c. Estimate process model according to Figure 4:

Model Gain (MG) = (slope2 -slope1) / CO change.

Observed Deadtime (L) = time between change of CO till the time when slope 1 and slope2 intersect each other

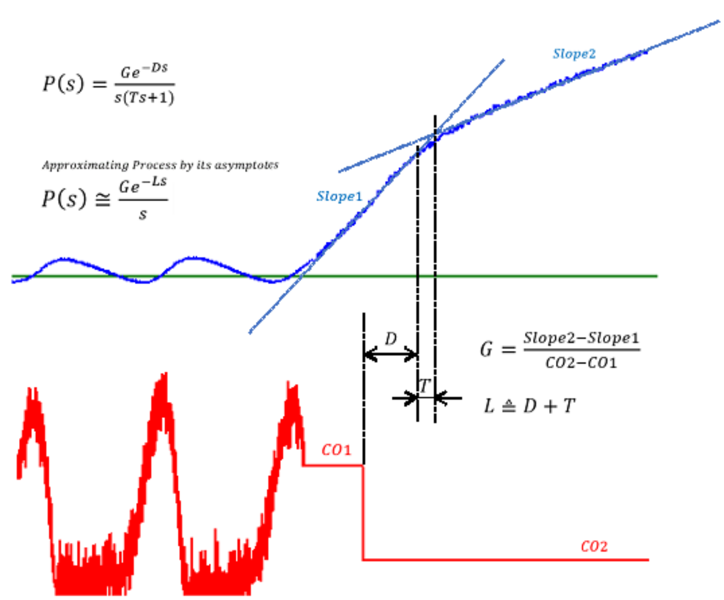
d. Select initial PID values using applicable equation:

Non-Interacting (ISA) or Series[1]	Parallel
P_N[2] = 0.45 / (Model Gain x Observed Deadtime) I_{NI} = 8 x (Observed Deadtime) D_{NI} = 0.5 x (Observed Deadtime)	$P_P = P_{NI}$ $I_P = P_{NI} / I_{NI}$ $D_P = P_{NI} \times D_{NI}$

[1] Acknowledgment: tuning formula for integrating process of this pocket guide is modified version of one given in Advanced PID Control (2006), by Karl J. Åström and Tore Hägglund, ISA.org.

[2] For calculation of P term, time units for observed deadtime (L) should be consitent with the time units used to calculate model gain (MG).

Integrating Process Model (Figure 4)



CLOSED LOOP BUMP TEST

In this document, Proportional Gain, Integral Reset Rate, and Derivative Gain are assumed. Convert back to your controller units if necessary.

1 Know the Process

Identify the loop you intend to tune and determine the speed of the loop.

FAST LOOP Response time < 1s to about 10s (e.g., a flow loop). Use of PI controller is sufficient.

MEDIUM LOOP Response time of several seconds up to about 30s (e.g., flow, temperature, and pressure). Use either PI or PID controller.

SLOW LOOP Response time of > 30s (e.g., many temperature loops and level loops). Use of PID controller is recommended.

2 Know the Controller

Identify the units of your PID controller:

PROPORTIONAL TERM (P-TERM)

Tuning parameter is either a Proportional Gain (P-Gain) or Proportional Band (P-Band).

INTEGRAL TERM (I-TERM)

Tuning parameter can be a Time Constant (in seconds or minutes), Reset Rate (1/second or 1/minute), or I-Gain (Reset rate x Proportional gain).

DERIVATIVE TERM (D-TERM)

Tuning parameter can be Time Constant (in seconds or minutes) or Derivative Gain (Derivative Time Constant x Proportional Gain).

3 Watch the Response

Make a small change of setpoint (SP), say 5%, or wait for a disturbance in the process. Then watch for process variable (PV) and control output (CO) responses.

a. If no visible instantaneous change of CO upon the change of SP or no apparent overshoot (over damped), increase proportional gain by 50%.

b. If PV is unstable or has sustained oscillation, with overshoot greater than 25%, reduce Proportional Gain by 50% and reduce Integral Reset Rate by 50%.

c. If PV oscillation persists with tolerable overshoot, reduce Proportional Gain by 20% and reduce Integral Reset Rate by 50%.

d. If three or more consecutive peaks occur upon the change of SP, reduce Integral Reset Rate by 30% and increase Derivative Gain by 50%.

e. If PV stays fairly flat and below (or above) the SP for a long time, after change of SP or beginning of disturbance (long tail scenario), increase Integral Reset Rate by 100%.

f. Repeat Step 3 until the closed-loop response is satisfactory to you.

Tuning Cascade Loops

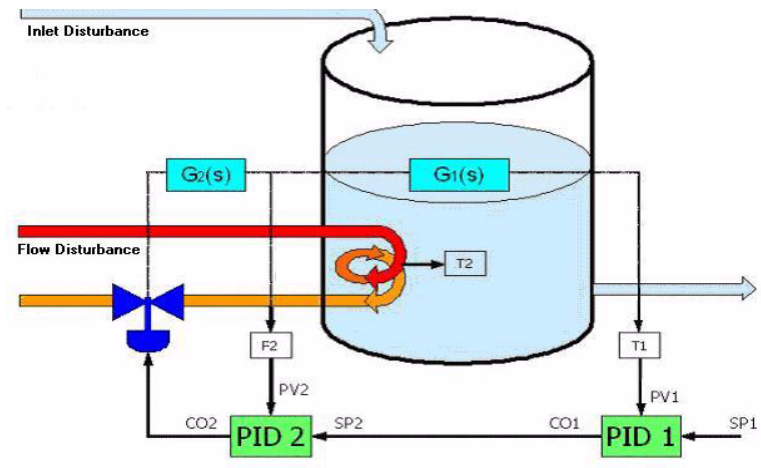
We're often asked,
What's the best way to tune a cascade loop?

The simple answer is, *Tune the inner loop first and the outer loop second.*

CASCADED PID CONTROL OF TANK OUTLET TEMPERATURE

In this example, PID 1 is the Primary or Outer Loop; PID 2 is the Secondary or Inner Loop. The temperature controller, PID 1, determines the desired amount of flow to control the temperature. Instead of controlling temperature directly, the valve is controlling the flow of the steam to the process from PID 2.

In applying loop tuning to a cascade control configuration, you need to tune one loop at a time due to interaction of inner loop dynamics on the outer loop. (Note that the inner loop must be tuned first before tuning the outer loop unless using a one-shot cascade tuning tool.)



1. Put the outer loop in Manual.
2. Do a regular loop tuning procedure on the inner loop.
3. Put the inner loop in Auto.
4. Wait for the outer loop to stabilize.
5. Do a regular loop tuning procedure on the outer loop.

Scan Time Considerations

PID Scan Time is the recommended time interval between successive calculations of the PID controller. Values for scan time can be in Millisecond, Second, Minute, Hour, or Day.

If the calculation is performed too seldom, you won't be able to control the process; if performed too frequently, the loop might be hard to tune.

Generally the PID scan time should be faster than the loop deadtime ((deadtime plus the lagtime) / 30). And the shorter the loop deadtime, the more frequent your scan time should be.

At all times keep in mind the speed limits, if any, of your input and output devices, both sensors and actuators. There's no point in scanning faster than they can update. Since the PID calculation is only performed at the scan rate, be aware that the output will remain static until the next scan. This is why it's important to correctly set the scan time period, so the output will react to the input in a timely manner.

PID Controller Reference

A summary of many of the more common controllers available on the market.

For units of other controllers, please contact ControlSoft.

(see Page 17)

CONTROLLER REFERENCE KEY

Non-Integrating Processes	Integrating Processes
MG – Model Gain*	**MG –** Model Gain*
DT – Deadtime	**L –** Observed Deadtime + Lagtime
LagTC – Lag Time Constant	**Reps –** Repeats per Unit of Time
Reps – Repeats per Unit of Time	**Ext. Reset –** External Reset
Ext. Reset – External Reset	**Man. Reset –** Manual Reset
Man. Reset – Manual Reset	

Note that Model Gain (MG) for tuning must be calculated in the units that the PID equation uses. Generally, the unit of MG is the inverse of the unit of P tuning parameter of PID.

ABB
AC 800

Non-Integrating Processes

C_CS, C_CU

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

Integrating Processes

C_CS, C_CU

P = (0.45 / (MG x L)) [x 1]

I = (8xL) [Sec x 1]

D = (0.5 x L) [Sec x 1]

ABB
Symphony Harmony INFI 90 Controllers

Non-Integrating Processes

18 and 19 PID Block

P = (1 / MG) [x 1]

I = (1 / MG) / (DT + LagTC) [Reps / Min x 1]

D = (1 / MG) x (DT / 3) [Min x 1]

156,0 PID Block - Classical

P = (1 / MG) [x 1]

I = 1 / (DT + LagTC) [/ Min x 1]

D = (DT / 3) [Min x 1]

156,1 PID Block - Non-Interacting

P (1 / MG) [x 1]

I = (1 / MG) / (DT + LagTC) [Reps / Min x 1]

D = (1 / MG) x (DT / 3) [Min x 1]

156,2 PID Block - Classical w/ Ext. Reset

P = (1 / MG) [x 1]

I = 1 / (DT + LagTC) [1 / Min x 1]

D = (DT / 3) [Min x 1]

156,3 PID Block - Classical w/ Man. Reset

P = (1 / MG) [x 1]

I = (1 / MG) / (DT + LagTC) [Reps / Min x 1]

D = (1 / MG) x (DT / 3) [Min x 1]

Integrating Processes

18 and 19 PID Block

P = (0.45 / (MG x L)) [x 1]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 1]

D = (0.45 / (MG x L)) x (0.5 x L) [Min x 1]

156,0 PID Block – Classical

P = (0.45 / (MG x L)) [x 1]

I = 1 / (8 x L) [1 / Min x 1]

D = (0.5 x L) [Min x 1]

156,1 PID Block - Non-Interacting

P= (0.45 / (MG x L)) [x 1]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 1]

D = (0.45 / (MG x L)) x (0.5 x L) [Min x 1]

156,2 PID Block - Classical w/ Ext. Reset

P = (0.45 / (MG x L)) [x 1]

I = 1 / (8 x L) [1 / Min x 1]

D = (0.5 x L) [Min x 1]

156,3 PID Block - Classical w/ Man. Reset

P = (0.45 / (MG x L)) [x 1]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 1]

D = (0.45 / (MG x L)) x (0.5 x L) [Min x 1]

ControlSoft
MANTRA

Non-Integrating Processes

PID Block - Series

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

PID Block - Parallel

P = (1 / MG) [x 1]

I = (1 / MG) / (DT + LagTC) [Reps / Sec x 1]

D = (1 / MG) x (DT / 3) [Sec x 1]

PID Block - Non-Interacting

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

Integrating Processes

PID Block - Series

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

PID Block - Parallel

P = (0.45 / (MG x L)) [x 1]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Sec x 1]

D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 1]

PID Block - Non-Interacting

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

Emerson
DeltaV

Non-Integrating Processes

PID Form = Standard

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

PID Form = Series

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

Integrating Processes

PID Form = Standard

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

PID Form = Series

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

Emerson
Ovation Controllers

Non-Integrating Processes

PID, PIDFF Blocks

P = (1 / MG) [x 1]

I = (MG) x (DT + LagTC) [Sec / Reps x 1]

D = (1 / MG) x (DT / 3) [Sec x 1]

Integrating Processes

PID, PIDFF Blocks

P = (0.45 / (MG x L)) [x 1]

I = (0.45 / (MG x L)) x (8 x L) [Sec / Reps x 1]

D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 1]

Emerson
Westinghouse WDPF

Non-Integrating Processes

PID Block

P = (1 / MG) [x 100]

I = (DT + LagTC) x MG [Sec/Reps x 1]

D = (1 / MG) x (DT / 3) [Sec x 1]

Integrating Processes

PID Block

P = (0.45 / (MG x L)) [x 100]

I = ((8 x L) x (MG x L) / 0.45) [Sec / Reps x 1]

D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 1]

GE
Mark VI

Non-Integrating Processes

PID, PID Basic, PID_MA_ENH

P = (1 / MG) [x 1]

I = 1 / (DT + LagTC) [1 / Min x 1]

D = (DT / 3) [Min x 1]

Integrating Processes

PID, PID Basic, PID_MA_ENH

P = (0.45 / (MG x L)) [x 1]

I = 1 / (8 x L) [1 / Min x 1

D = (0.5 x L) [Min x 1]

GE
RX3i/RX7i Controllers

Non-Integrating Processes

PID_ISA

P = (1 / MG) [x 100]

I = 1 / (DT + LagTC) [1 / Sec x 1000]

D = (DT / 3) [Sec x 100)]

PID_IND

P = (1 / MG) [x 100]

I = (1 / MG) / (DT + LagTC) [Reps / Sec x 1000]

D = (1 / MG) x (DT / 3) [Sec x 100]

ADV_PID Block - Parallel

P = (1 / MG) [x 1]

I = (1 / MG) / (DT + LagTC) [Reps / Min x 1]

D = (1 / MG) x (DT / 3) [Sec x 1]

ADV_PID Block - Series

P = (1 / MG) [x 1]

I = 1 / (DT + LagTC) [1 / Min x 1]

D = (DT / 3) [Sec x 1]

Integrating Processes

PID_ISA

P = (0.45 / (MG x L)) [x 100]

I = 1 / (8 x L) [1 / Sec x 1000]

D = (0.5 x L)[Sec x 100)]

PID_IND

P = (0.45 / (MG x L)) [x 100]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Sec x 1000]

D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 100]

ADV_PID Block - Parallel

P = (0.45/(MG x L)) [x 1]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 1]

D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 1]

ADV_PID Block - Series

P = (0.45 / (MG x L)) [x 1]

I = 1 / (8 x L) [1 / Min x 1]

D = (0.5 x L) [Sec x 1]

GE
Series 90-30 &
90-70 Controllers

Non-Integrating Processes

PID Block - ISA

P = (1 / MG) [x 100]

I = 1 / (DT + LagTC) [1 / Sec x 1000]

D = (DT / 3) [Sec x 100)]

PID Block - Independent

P = (1 / MG) [x 100]

I = (1 / MG) / (DT + LagTC) [Reps / Sec x 1000]

D = (1 / MG) x (DT / 3) [Sec x 100]

Integrating Processes

PID Block - ISA

P = (0.45/(MG x L)) [x 100]

I = 1 / (8 x L) [1 / Sec x 1000]

D = (0.5 x L) [Sec x 100)]

PID Block - Independent

P = (0.45 / (MG x L)) [x 100]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Sec x 1000]

D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 100]

Honeywell
Experion Controllers

Non-Integrating Processes

PID, PIDFF, PIDER

P = (1 / MG) [x 1]

I = (DT + LagTC) [Min x 1]

D = (DT / 3) [Min x 1]

Integrating Processes

PID, PIDFF, PIDER

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Min x 1]

D = (0.5 x L) [Min x 1]

Honeywell
TDC3000 Controllers

Non-Integrating Processes

Interactive A, Interactive B, Interactive C, Interactive D, Non-Interactive A, Non-Interactive B, Non-Interactive C, Non-Interactive D

P = (1 / MG) [x 1]

I = (DT + LagTC) [Min x 1]

D = (DT / 3) [Min x 1]

Integrating Processes

Interactive A, Interactive B, Interactive C, Interactive D, Non-Interactive A,Non-Interactive B, Non-Interactive C, Non-Interactive D

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Min x 1]

D = (0.5 x L) [Min x 1]

Honeywell
UDC Controllers

Non-Integrating Processes
Interactive A, Interactive B,
Non-Interactive A, Non-Interactive B
P = (1 / MG) [x 1]
I = (DT + LagTC) [Sec x 1]
D = (DT / 3) [Sec x 1]

PID
P = (MG x 100) [x 1]
I = (DT + LagTC) [Sec x 1]
D = (DT / 3) [Sec x 1]

Integrating Processes
Interactive A, Interactive B,
Non-Interactive A, Non-Interactive B
P = (0.45 / (MG x L)) [x 1]
I = (8 x L) [Sec x 1]
D = (0.5 x L) [Sec x 1]

PID
P = (((MG x L) / 0.45) x 100) [x 1]
I = (8 x L) [Sec x 1]
D = (0.5 x L) [Sec x 1]

Modicon

Non-Integrating Processes
PID, KPID Blocks
P = (1 / MG) [x 1]
I = (DT + LagTC) [Sec x 1]
D = (DT / 3) [Sec x 1]

PID2 Block
P = (MG x 100) [x 1]
I = 1 / (DT + LagTC) [1 / Min x 100]
D = (DT / 3) [Min x 100]

Integrating Processes
PID, KPID Blocks
P = (0.45 / (MG x L)) [x 1]
I = (8 x L) [Sec x 1]
D = (0.5 x L) [Sec x 1]

PID2 Block
P = (((MG x L) / 0.45) x 100) [x 1]
I = 1 / (8xL) [1 / Min x 100]
D = (0.5 x L) [Min x 100]

Rockwell Automation
CompactLogix
& ControlLogix
Controllers
& PlantPAx DCS

Non-Integrating Processes
PPID Independent
P = (1 / MG) [x 1]
I = (1 / MG) / (DT + LagTC) [Reps / Min x 1]
D = (1 / MG) x (DT / 3) [Min x 1]

PPID Dependent
P = (1 / MG) [x 1]
I = (DT + LagTC) [Min x 1]
D = (DT / 3) [Min x1]

PIDE, P_PIDE Block Independent
P = (1 / MG) [x 1]
I = (1 / MG) / (DT + LagTC) [Reps / Min x 1]
D = (1 / MG) x (DT / 3) [Min x 1]

PIDE, P_PIDE Block Dependent
P = (1 / MG) [x 1]
I = (DT + LagTC) [Min x 1]
D = (DT / 3) [Min x 1]

Integrating Processes
PPID Independent
P = (0.45 / (MG x L)) [x 1]
I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 1]
D = (0.45 / (MG x L) x (0.5 x L) [Min x 1]

PPID Dependent
P = (0.45 / (MG x L)) [x 1]
I = (8 x L) [Min x 1]
D = (0.5 x L) [Min x 1]

PIDE, P_PIDE Block - Independent
P = (0.45 / (MG x L)) [x 1]
I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 1]
D = (0.45 / (MG x L)) x (0.5 x L) [Min x 1]

PIDE, P_PIDE Block – Dependent
P = (0.45 / (MG x L)) [x 1]
I = (8 x L) [Min x 1]
D = (0.5 x L) [Min x 1]

Rockwell Automation
Micro800 Controllers

Non-Integrating Processes

IPID

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

Integrating Processes

IPID

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

Rockwell Automation
MicroLogix Controllers

Non-Integrating Processes

PID Block with RG bit on

P = (1 / MG) [x 100]

I = (DT + LagTC) [Min x 100]

D = (DT / 3) [Min x 100]

PID Block with RG bit off

P = (1 / MG) [x 10]

I = (DT + LagTC) [Min x 10]

D = (DT / 3) [Min x 100]

Integrating Processes

PID Block with RG bit on

P = (0.45 / (MG x L)) [x 100]

I = (8 x L) [Min x 100]

D = (0.5 x L) [Min x 100]

PID Block with RG bit off

P = (0.45 / (MG x L)) [x 10]

I = (8 x L) [Min x 10]

D = (0.5 x L)[Min x 100]

Rockwell Automation
PLC-5 Controllers

Non-Integrating Processes

Integer Block ISA

P = (1 / MG) [x 100]

I = (DT + LagTC) [Min x 100]

D = (DT / 3) [Min x 100]

Integer Block Independent Gains

P = (1 / MG) [x 100]

I = (1 / MG) / (DT + LagTC) [Reps / Sec x 1000]

D = (1 / MG) x (DT / 3) [Sec x 100]

PD Block ISA

P = (1 / MG) [x 1]

I = (DT + LagTC) [Min x 1]

D = (DT / 3) [Min x 1]

PD Block Independent Gains

P = (1 / MG) [x 1]

I = (1 / MG) / (DT + LagTC) [Reps / Sec x 1]

D = (1 / MG) x (DT / 3) [Sec x 1]

Integrating Processes

Integer Block ISA

P = (0.45 / (MG x L)) [x 100]

I = (8 x L) [Min x 100]

D = (0.5 x L) [Min x 100]

Integer Block Independent Gains

P = (0.45 / (MG x L)) [x 100]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Sec x 1000]

D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 100]

PD Block ISA

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Min x 1]

D = (0.5 x L) [Min x 1]

PD Block Independent Gains

P = (0.45 / (MG x L)) [x 1]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Sec x 1]

D = (0.45 / (MG xL)) x (0.5 x L) [Sec x 1]

Rockwell Automation
SLC Controllers

Non-Integrating Processes

PID Block with RG bit on

P = (1 / MG) [x 100]

I = (DT + LagTC) [Min x 100]

D = (DT / 3) [Min x 100]

PID Block with RG bit off

P = (1 / MG) [x 10]

I = (DT + LagTC) [Min x 10]

D = (DT / 3) [Min x 100]

Integrating Processes

PID Block with RG bit on

P = (0.45 / (MG x L)) [x 100]

I = (8 x L) [Min x 100]

D = (0.5 x L) [Min x 100]

PID Block with RG bit off

P = (0.45 / (MG x L)) [x 10]

I = (8 x L) [Min x 10]

D = (0.5 x L) [Min x 100]

Schneider Electric

	Non-Integrating Processes	Integrating Processes
	PID, PID1, COMP_PID	**PID, PID1, COMP_PID**
	P = (1 / MG) [x 1]	P = (0.45 / (MG x L)) [x 1]
	I = (DT + LagTC) [Msec x 1]	I = (8 x L) [Msec x 1]
	D = (DT / 3) [Msec x 1]	D = (0.5 x L) [Msec x 1]
	PIDFF mix_par=0 (Mixed)	**PIDFF mix_par=0 (Mixed)**
	P = (1 / MG) [x 1]	P = (0.45 / (MG x L)) [x 1]
	I = (DT + LagTC) [Msec x 1]	I = (8 x L) [Msec x 1]
	D = (DT / 3) [Msec x 1]	D = (0.5 x L) [Msec x 1]
	PIDP1, PID_P	**PIDP1, PID_P**
	P = (1 / MG) [x 1]	P = (0.45 / (MG x L)) [x 1]
	I = (1 / MG) / (DT + LagTC) [Reps / Msec x 1]	I = (0.45 / (MG x L)) / (8 x L) [Reps / Msec x 1]
	D = (1 / MG) x (DT / 3) [Msec x 1]	D = (0.45 / (MG x L)) x (0.5 x L) [Msec x 1]
	PIDFF mix_par=1 (Parrallel)	**PIDFF mix_par=1 (Parrallel)**
	P = (1 / MG) [x 1]	P = (0.45 / (MG x L)) [x 1]
	I = MG x (DT + LagTC) [Msec / Reps x 1]	I = ((8 x L) x (MG x L) / 0.45) [Msec / Reps x 1]
	D = (1 / MG) x (DT / 3) [Msec x 1]	D = (0.45 / (MG x L) x (0.5 x L) [Msec x 1]
	PI, PI1, PI_B	**PI, PI1, PI_B**
	P = (1 / MG) [x 1]	P = (0.45 / (MG x L)) [x 1]
	I = (DT + LagTC) [Msec x 1]	I = (8 x L) [Msec x 1]
Schneider Electric Invensys Foxboro I/A ISA Blocks	**PIDA (MODOPT = 6,8)** P = (MG x 100) [x 1] I = (DT + LagTC) [Min x 1] D = (DT / 3) [Min x 1]	**PIDA (MODOPT = 6, 8)** P = (((MG x L) / 0.45) x 100) [x 1] I = (8 x L) [Min x 1] D = (0.5 x L) [Min x 1]
Schneider Electric Invensys Foxboro I/A Series Blocks	**PID, PIDA, PIDE, PIDX, PIDXE, PIDA (MODOPT = 1,2,3,4,5,7)** P = (MG x 100) [x 1] I = (DT + LagTC) [Min x 1] D = (DT / 3) [Min x 1]	**PID, PIDA, PIDE, PIDX, PIDXE, PIDA (MODOPT = 1,2,3,4,5,7)** P = (((MG x L) / 0.45) x 100) [x 1] I = (8 x L) [Min x 1] D = (0.5 x L) [Min x 1]
Schneider Electric Square D Controllers	**PID Block** P = (1 / MG) [x 100] I = (1 / MG) / (DT + LagTC) [Reps / Min x 100] D = (1 / MG) x (DT / 3) [Sec x 100]	**PID Block** P = (0.45 / (MG x L)) [x 100] I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 100] D = (0.45 / (MG x L)) x (0.5 x L) [Sec x 100]

Siemens

TI

PID Block

P = (1 / MG) [x 1]

I = (DT + LagTC) [Min x 1]

D = (DT / 3) [Min x 1]

TI

PID Block

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Min x 1]

D = (0.5 x L) [Min x 1]

S7 Controllers

FB41 CONT_C, FB41 CONT_S, FB61 CTRL_PID, FB61 CTRL_S, PID_CP, PID_ES, FB 1874 PIDConL

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

S7 Controllers

FB41 CONT_C, FB41 CONT_S, FB61 CTRL_PID, FB61 CTRL_S, PID_CP, PID_ES, FB 1874 PIDConL

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

Toshiba

Non-Integrating Processes

Basic PID, Advanced PID

P = (1 / MG) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

Integrating Processes

Basic PID, Advanced PID

P = (0.45 / (MG x L)) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

Valmet (Metso)

Non-Integrating Processes

Series, Ideal PID Block

P = (1 / MG) [x 1]

I = 1 / (DT + LagTC) [1 / Min x 1]

D = (DT / 3) [Min x 1]

Parallel PID Block

P = (1 / MG) [x 1]

I = (1 / MG) / (DT + LagTC) [Reps / Min x 1]

D = (DT / 3) [Min x 1]

Integrating Processes

Series, Ideal PID Block

P = (0.45 / (MG x L)) [x 1]

I = 1 / (8 x L) [1 / Min x 1]

D = (0.5 x L) [Min x 1]

Parallel PID Block

P = (0.45 / (MG x L)) [x 1]

I = (0.45 / (MG x L)) / (8 x L) [Reps / Min x 1]

D = (0.5 x L) [Min x 1]

Watlow

Eurotherm Controllers

PID Block

P = (MG x 100) [x 100]

I = (DT + LagTC) [Sec x 100]

D = (DT / 3) [Sec x 100]

PID Block

P = (((MG x L) / 0.45) x 100) [x 100]

I = (8 x L) [Sec x 100]

D = (0.5 x L) [Sec x 100]

Yokogawa

CS1000/CS3000 I-PD, PD-MR, PI-BLEND, PID, P-ID, PID-BSW, PID-TP, PI-HLD, PI-STC

P = (MG x 100) [x 1]

I = (DT + LagTC) [Sec x 1]

D = (DT / 3) [Sec x 1]

CS1000/CS3000 I-PD, PD-MR, PI-BLEND, PID, P-ID, PID-BSW, PID-TP, PI-HLD, PI-STC

P = (((MG x L) / 0.45) x 100) [x 1]

I = (8 x L) [Sec x 1]

D = (0.5 x L) [Sec x 1]

Optimize Parameters with INTUNE

ControlSoft's INTUNE PID Loop Tuning software empowers teams to optimize tuning, reject disturbances, stabilize their processes and enhance precision when it comes to a loop's performance.

INTUNE®
Loop Tuning, Performance Monitoring & Diagnostic Software

ADVISORY ADAPTIVE TUNING

Backed by 40+ years of process control expertise, our advisory adaptive tuning brings proven field knowledge directly into your operation. This proprietary learning algorithm works within your natural process cycles to automatically spot tuning issues, reject disturbances, and recommend optimized PID parameters, delivering more stable, efficient, and profitable performance without disrupting production.

- MINIMIZE THE TIME TO TUNE AND MAINTAIN CONTROL LOOPS
- ELIMINATE THE NEED FOR BUMP TESTS AND WORK WITHIN THE NATURAL CYCLE OF YOUR SYSTEM
- INCREASE PRECISION WITH ADVISORY ADAPTIVE TUNING

CONNECTIVITY

- PROVIDES REAL-TIME UPDATES AT SPECIFIED INTERVALS
- MAPS PARAMETERS FOR COMMON CONTROLLERS
- SHARES PERFORMANCE CALCULATIONS AND MEASURES
- QUICKLY RESPONDS TO PROCESS DISTURBANCES

OPC FOUNDATION

AVEVA PI System

FREE FIRST YEAR TECHNICAL PRODUCT SUPPORT
WITH SOFTWARE PURCHASE

CONTROLSOFT
ADVANCED PROCESS CONTROL MADE EASY

www.controlsoftinc.com
(440) 443-3900 | info@controlsoftinc.com

SOFTWARE | CUSTOM SOLUTIONS | TRAINING

PUBLICATION **DS405Hb-EN**