About ControlSoft

We provide leading-edge technologies in process control solutions, performance monitoring and process diagnostic tools. For over 30 years, our superior technologies and solutions have been licensed and/or embedded in many other control systems (e.g., GE, Emerson and Rockwell Automation).

Services

Our in-depth expertise and understanding of process control technologies enables us to take on your toughest challenges and bring predictable success anytime, anywhere. Our services include:

- Consulting and technology development services
- Project development
- Economical and technical feasibility studies
- Project management
- Engineering and design
- Commissioning and start-up
- Training of operations and maintenance personnel
- Maintenance support

Training

We offerclassr oom and web-based training to help you learn the basic terminology and building blocks of process control as well as advanced process control techniques using our award-winning control solutions

View our current class schedule, detailed outline, and registration form at www.controlsoftinc.com/training.shtml.

Support

Building relationships with productivity-driven businesses, we've built a reputation for excellence in delivering not only technical support, but strategic direction in evolving technology environments.

Partnerships

ControlSoft is proud to partner with some of the best known names in leading industries. Our strong partnerships allow us to deliver the best process control solutions for our clients.

As a partner, we carefully guard all proprietary technologies and confidential information shared with us. We continue to seek these partnerships, forged on our values of integrity, professionalism, mutual respect and understanding.

5387 Avion Park Drive
Highland Heights, OH 44143 USA
440.443.3900 | info@controlsoftinc.com
www.controlsoftinc.com

CONTROLSOFT
ADVANCED PROCESS CONTROL MADE EASY

Publication DS405E-EN

PID Loop Tuning Pocket Guide

Version 5

Contents

CONTROLSOFT
ADVANCED PROCESS CONTROL MADE EASY

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 deal with 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% * (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 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 0.1*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) * P-Gain

Derivative Term (D)

The amount subtracted from 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

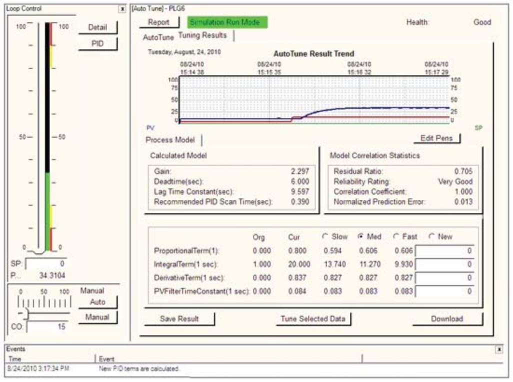
Integrating Loops (level & insulated temperature)

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.)

Efficient Tuning Tools



Calculated Model

Gain:	2.297
Deadtime(sec):	6.000
Lag Time Constant(sec):	9.597
Recommended PID Scan Time(sec):	0.390

Model Correlation Statistics

Residual Ratio:	0.705
Reliability Rating:	Very Good
Correlation Coefficient:	1.000
Normalized Prediction Error:	0.013

	Org	Cur	○ Slow	● Med	○ Fast	○ New
ProportionalTerm(1):	0.000	0.800	0.594	0.606	0.606	0
IntegralTerm(1 sec):	1.000	20.000	13.740	11.270	9.930	0
DerivativeTerm(1 sec):	0.000	0.837	0.827	0.827	0.827	0
PVFilterTimeConstant(1 sec):	0.000	0.084	0.083	0.083	0.083	0

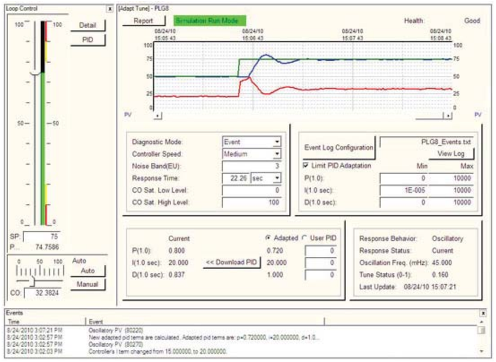
Events

Time	Event
8/24/2010 3:17:34 PM	New PID terms are calculated.

Auto-Tune

Use Auto-Tune to generate PID parameter values for your PID loops. Auto-Tune will collect and analyze process data and then calculate robust tuning parameters for your PID loops.

Save over 75% of your engineering time over manual tuning!



Diagnostic Mode:	Event
Controller Speed:	Medium
Noise Band(EU):	3
Response Time:	22.26 sec
CO Sat. Low Level:	0
CO Sat. High Level:	100

PLG8_Events.txt

☑ Limit PID Adaptation

	Min	Max
P(1.0):	0	10000
I(1.0 sec):	1E-005	10000
D(1.0 sec):	0	10000

	Current	● Adapted	○ User PID
P(1.0):	0.800	0.720	0
I(1.0 sec):	20.000	20.000	0
D(1.0 sec):	0.837	1.000	0

Response Behavior:	Oscillatory
Response Status:	Current
Oscillation Freq. (mHz):	45.000
Tune Status (0-1):	0.160
Last Update:	08/24/10 15:07:21

Events

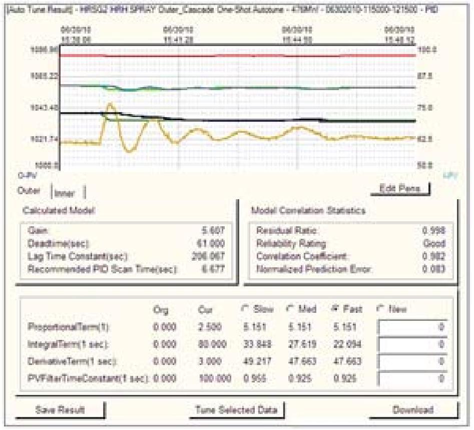
Time	Event
8/24/2010 3:07:21 PM	Oscillatory PV (80220)
8/24/2010 3:02:57 PM	New adapted pid terms are calculated. Adapted pid terms are: p=0.720000, i=20.000000, d=1.0...
8/24/2010 3:02:57 PM	Oscillatory PV (80270)
8/24/2010 3:02:03 PM	Controller's I term changed from 15.000000, to 20.000000.

Adapt-Tune

Use sophisticated pattern recognition techniques to analyze and interpret process data in real time.

These fine tuning adjustments can be applied automatically or after operator acknowledgement.

Non-intrusive monitoring and recommendations frees you up to do other things.



Calculated Model

Gain:	5.607
Deadtime(sec):	61.000
Lag Time Constant(sec):	206.067
Recommended PID Scan Time(sec):	6.677

Model Correlation Statistics

Residual Ratio:	0.998
Reliability Rating:	Good
Correlation Coefficient:	0.982
Normalized Prediction Error:	0.083

	Org	Cur	○ Slow	○ Med	● Fast	○ New
ProportionalTerm(1):	0.000	2.500	5.151	5.151	5.151	0
IntegralTerm(1 sec):	0.000	80.000	33.848	27.619	22.094	0
DerivativeTerm(1 sec):	0.000	3.000	49.217	47.663	47.663	0
PVFilterTimeConstant(1 sec):	0.000	100.000	0.955	0.925	0.925	0

One-Shot Cascade Auto-Tune

Performs tuning of both inner and outer cascaded loops in one test, eliminating the two-step manual tuning of cascaded control loops.

Efficiently tune cascade loops simultaneously in one test. Improve cascade loop results using auto-generated parameters.

INTUNE Tuning Tools

INTUNE™

Loop Tuning, Performance Monitoring, & Diagnostic Software

A well-tuned PID controller, in a properly designed control loop, can satisfy the control objectives for most process control applications. However, in many plants today:

- 80% of control loops are not optimally tuned
- 25% of control loops still have factory-default settings in place
- 65% of control loops in automatic mode actually have less variation in manual mode
- Control loop performance declines at a rate of about 50% every 6 months
- Without proper maintenance, 75% of loop performance is lost in one year

By investing in INTUNE tuning software to tune your process control PID loops (temperature, level, pressure and flow), your control loops will operate efficiently, rather than just being functional. INTUNE software provides:

- Tools to reduce the time required to tune PID loops by up to 70%
- Repeatable, robust PID tuning parameters
- PID tuning reports for each loop you tune (history of PID optimization)
- No-bump adaptive-tuning technology to monitor tuning needs and recommend adjustments without disturbing your process

Easy Connectivity

- INTUNE connects to most I/O devices and their associated controllers using OPC technology. The OPC client allows easy access to any control system through the controller OPC server.
- For commonly used controllers, INTUNE automatically maps the parameters of the selected controller to the INTUNE controller object.
- INTUNE maintains real-time connections with your process PID loops, and updates all process parameters at every sampling interval specified.
- INTUNE is also its own OPC Server, so other clients have access to the unique performance calculations and measures in INTUNE.
- Ability to define multiple FeedForward (FF) Controllers for each loop. Quickly respond to process disturbances that can affect bottom line.

Superior Support

You get a full year of technical support included with your software purchase so you can get expert help when you need it. Our process control specialists can analyze your data and recommend changes in INTUNE can help you optimize plant operating conditions and improve plant efficiency.

Contact us for additional information on purchasing INTUNE software or to schedule a free on-site consultation, at 440.443.3900 or info@controlsoftinc.com.

Tuning Closed Loops

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 minutes or seconds), Reset Rate (1/second or 1/minute), or I-Gain (Reset rate * Proportional gain).
- **Derivative term (D-Term)** - tuning parameter can be Time Constant (in seconds or minutes) or Derivative Gain (Derivative Time Constant * Proportional Gain).

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

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 3 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 Open Loops

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 * Proportional gain).
- **Derivative term (D-Term)** - tuning parameter can be Time Constant (in seconds or minutes) or Derivative Gain (derivative time constant * 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 7) is a summary of many common controllers available on the market.
For units of other controllers, please contact ControlSoft.

Rockwell Automation (cont.)

Micro800 Controllers

IPID
P = (1 / MG) [* 1]
I = (DT + LagTC) [Sec * 1]
D = (DT / 3) [Sec * 1]

MicroLogix Controllers

PID Block with RG bit on
P = (1 / MG) [* 100]
I = (DT + LagTC) [Min * 100]
D = (DT / 3) [Min * 100]

PID Block with RG bit off
P = (1 / MG) [* 10]
I = (DT + LagTC) [Min * 10]
D = (DT / 3) [Min * 100]

PLC-5 Controllers

Integer Block ISA
P = (1 / MG) [* 100]
I = (DT + LagTC) [Min * 100]
D = (DT / 3) [Min * 100]

Integer Block Independent Gains
P = (1 / MG) [* 100]
I = (1 / MG) / (DT + LagTC)
    [Reps / Sec * 1000]
D = (1 / MG) * (DT / 3) [Sec * 100]

PD Block ISA
P = (1 / MG) [* 1]
I = (DT + LagTC) [Min * 1]
D = (DT / 3) [Min * 1]

PD Block Independent Gains
P = (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Sec * 1]
D = (1 / MG) * (DT / 3) [Sec * 1]

SLC Controllers

PID Block with RG bit on
P = (1 / MG) [* 100]
I = (DT + LagTC) [Min * 100]
D = (DT / 3) [Min * 100]

PID Block with RG bit off
P = (1 / MG) [* 10]
I = (DT + LagTC) [Min * 10]
D = (DT / 3) [Min * 100]

Siemens

TI

PID Block
P = (1 / MG) [* 1]
I = (DT + LagTC) [Min * 1]
D = (DT / 3) [Min * 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) [* 1]
I = (DT + LagTC) [Sec * 1]
D = (DT / 3) [Sec * 1]

Schneider Electric

Eurotherm Controllers

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

Invensys Foxboro I/A Series Blocks

PID, PIDA, PIDE, PIDX, PIDXE
P = (MG * 100) [* 1]
I = (DT + LagTC) [Min * 1]
D = (DT / 3) [Min * 1]

Square D Controllers

PID Block
P = (1 / MG) [* 100]
I = (1 / MG) / (DT + LagTC)
    [Reps / Min * 100]
D = (1 / MG) * (DT / 3) [Sec * 100]

Yokogawa

CS1000/CS3000 I-PD, PD-MR, PI-BLEND,
PID, P-ID, PID-BSW, PID-TP, PI-HLD, PI-STC
P = (MG * 100) [* 1]
I = (DT + LagTC) [Sec * 1]
D = (DT / 3) [Sec * 1]

GE (cont.)

RX3i/RX7i Controllers (cont.)

PID_IND
P = (1 / MG) [* 100]
I = (1 / MG) / (DT + LagTC)
[Reps / Sec * 1000]
D = (1 / MG) * (DT / 3) [Sec * 100]

ADV_PID Block - Parallel
P = (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Min * 1]
D = (1 / MG) * (DT / 3) [Sec * 1]

ADV_PID Block - Series
P = (1 / MG) [* 1]
I = 1 / (DT + LagTC) [1 / Min * 1]
D = (DT / 3) [Sec * 1]

Mark VI

PID, PID Basic, PID_MA_ENH
P = (1 / MG) [* 1]
I = 1 / (DT + LagTC) [1 / Min * 1]
D = (DT / 3) [Min * 1]

Honeywell

TDC3000 Controllers

Interactive A, Interactive B, Interactive C, Interactive D, Non-Interactive A, Non-Interactive B, Non-Interactive C, Non-Interactive D
P = (1 / MG) [* 1]
I = (DT + LagTC) [Min * 1]
D = (DT / 3) [Min * 1]

Experion Controllers

PID, PIDFF, PIDER
P = (1 / MG) [* 1]
I = (DT + LagTC) [Min * 1]
D = (DT / 3) [Min * 1]

UDC Controllers

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

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

Metso

Series, Ideal PID Block
P = (1 / MG) [* 1]
I = 1 / (DT + LagTC) [1 / Min * 1]
D = (DT / 3) [Min * 1]

Parallel PID Block
P = (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Min * 1]
D = (DT / 3) [Min * 1]

Modicon

PID, KPID Blocks
P = (1 / MG) [* 1]
I = (DT + LagTC) [Sec * 1]
D = (DT / 3) [Sec * 1]

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

Rockwell Automation

ControlLogix Controllers

PID Block - Independent
P = (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Sec * 1]
D = (1 / MG) * (DT / 3) [Sec * 1]

PID Block - Dependent
P = (1 / MG) [* 1]
I = (DT + LagTC) [Min * 1]
D = (DT / 3) [Min * 1]

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

PIDE Block - Dependent
P = (1 / MG) [* 1]
I = (DT + LagTC) [Min * 1]
D = (DT / 3) [Min * 1]

3 Test the loop.

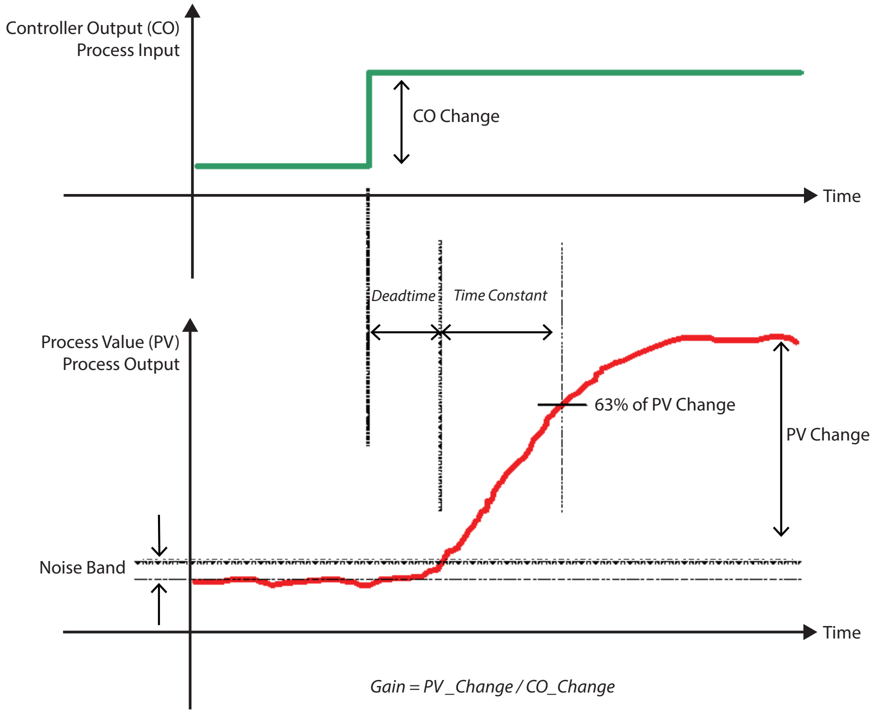
a. Put loop in manual control mode, keep control output constant, and wait for process to stabilize.
b. Make a small step change on control output (say 5 or 10%), and watch the response.
c. Estimate process model according to Figure 1:
Model gain = PV change / CO change.
Deadtime = time lapse between 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 = 1 / Model Gain
I = (Deadtime + Time Constant)
D* = (Deadtime / 3) or (Time Constant / 6)
	P = 1/Model Gain
	I = P/(DT+LagTC)=(1/Model Gain)/(DT+LagTC)
	D=P*(DT/3)=(1/Model Gain)*(DT/3)

* For slow loop, select whichever is greater. For fast loop, select smaller value.	

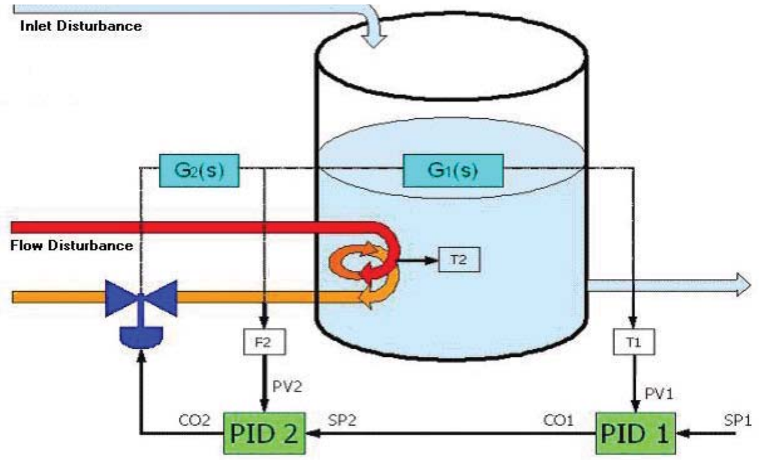
These initial PID values should give you a reasonable closed-loop response. Fine-tune the controller by using the closed-loop tuning method (see step 3 on page 3) to get your desired response.



Tuning Cascade Loops

A frequently asked question is, "I have a cascade loop, how do I tune it?" The simple answer is, "Tune the inner loop first and then tune the outer loop."

The figure below shows a cascaded control application, a tank heated by a steam valve. The ultimate goal is to control outlet temperature T1 by the steam valve.



In this example of cascaded PID control of tank outlet temperature, PID 1 can be referred to as Master, Primary, or Outer Loop; PID 2 can be referred to as Slave, 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.)*

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.**

PID Controller Reference

Following is a summary of many of the more common controllers available on the market. For units of other controllers, please contact ControlSoft (see page 1 for contact information).

Controller Reference Key:

- DT = Deadtime
- MG = Model Gain
- Ext. Reset = External Reset
- Man. Reset = Manual Reset
- LagTC = Lag Time Constant
- Reps = Repeats

ABB

Symphony Harmony INFI 90 Controllers

18 and 19 PID Block
P = (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Min * 1]
D = (1 / MG) * (DT / 3) [Min * 1]

156,0 PID Block - Classical
P = (1 / MG) [* 1]
I = 1 / (DT + LagTC) [1 / Min * 1]
D = (DT / 3) [Min * 1]

156,1 PID Block - Non-Interacting
P (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Min * 1]
D = (1 / MG) * (DT / 3) [Min * 1]

156,2 PID Block - Classical w/ Ext. Reset
P = (1 / MG) [* 1]
I = 1 / (DT + LagTC) [1 / Min * 1]
D = (DT / 3) [Min * 1]

156,3 PID Block - Classical w/ Man. Reset
P = (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Min * 1]
D = (1 / MG) * (DT / 3) [Min * 1]

ControlSoft

MANTRA

PID Block - Series
P = (1 / MG) [* 1]
I = (DT + LagTC) [Sec * 1]
D = (DT / 3) [Sec * 1]

PID Block - Parallel
P = (1 / MG) [* 1]
I = (1 / MG) / (DT + LagTC) [Reps / Sec * 1]
D = (1 / MG) * (DT / 3) [Sec * 1]

PID Block - Non-Interacting
P = (1 / MG) [* 1]
I = (DT + LagTC) [Sec * 1]
D = (DT / 3) [Sec * 1]

Emerson

Ovation Controllers

PID, PIDFF Blocks
P = (1 / MG) [* 1]
I = (MG) * (DT + LagTC) [Sec / Reps * 1]
D = (1 / MG) * (DT / 3) [Sec * 1]

Westinghouse WDPF

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

GE

Series 90-30 & 90-70 Controllers

PID Block - ISA
P = (1 / MG) [* 100]
I = 1 / (DT + LagTC) [1 / Sec * 1000]
D = (DT / 3) [Sec * 100)]

PID Block - Independent
P = (1 / MG) [* 100]
I = (1 / MG) / (DT + LagTC)
[Reps / Sec * 1000]
D = (1 / MG) * (DT / 3) [Sec * 100]

RX3i/RX7i Controllers

PID_ISA
P = (1 / MG) [* 100]
I = 1 / (DT + LagTC) [1 / Sec * 1000]
D = (DT / 3) [Sec * 100)]