Autotuning of PID Controllers

Cheng-Ching Yu

Autotuning of PID Controllers

A Relay Feedback Approach

2nd Edition

With 140 Figures

Springer

Cheng-Ching Yu, PhD
Department of Chemical Engineering
National Taiwan University
1 Roosevelt Road
Section 4
Taipei 106-17
Taiwan

British Library Cataloguing in Publication Data
Yu, Cheng-Ching, 1956-
Autotuning of PID controllers. - 2nd ed.
1. PID controllers
I. Title
629.8
ISBN-10: 1846280362

Library of Congress Control Number: 2005931920

ISBN-10: 1-84628-036-2 2nd edition e-ISBN 1-84628-037-0 Printed on acid-free paper
ISBN-13: 978-1-84628-036-8 2nd edition
ISBN: 3-540-76250-7 1st edition

First published 1999
Second edition 2006

Printed in Germany

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springeronline.com

To Patricia, Jessica, and Albert
獻給我的家人－鄭伶俐、余潔思、余肇偉

Preface

This edition is a major revision to the first edition. The revision is motivated by the new progress in relay feedback autotuning, as proposed by Bill Luyben, where the shape of the relay response can be utilized to identify likely model structure. Several new chapters have been added, notably the use of the shape-factor for autotuning and controller monitoring, incorporating autotuning in a multiple-model setup, dealing with an imperfect actuator. At the turn of the century, competitiveness in the global economy remains the same and the need for rapid and flexible manufacturing has become standard practice. This has given process control engineers an expanded role in process operation.

It has long been recognized that industrial control is one of the key technologies to make existing processes economically competitive. In theory, sophisticated control strategies–supervisory, adaptive, model predictive control–should be the norm of industrial practice in modern plants. Unfortunately, a recent survey, by Desborough and Miller has shown otherwise. This indicates that 97% of regulatory controllers are of the proportional–integral–derivative (PID) type and only 32% of the loops show “excellent” or “good” performance. Six years have passed since the first edition was published, and the practice of industrial process control is very much the same: PID controllers are widely used but poorly tuned.

This book is aimed at engineers and researchers who are looking for ways to improve controller performance. It provides a simple and yet effective method of tuning PID controllers automatically. Practical tools needed to handle various process conditions, *e.g.* load disturbance, nonlinearity and noise, are also given.

The mathematics of the subject is kept to a minimum level and emphasis is placed on experimental designs that give relevant process information for the intended tuning rules. Numerous worked examples and case studies are used to illustrate the autotuning procedure and closed-loop performance.

This book is an independent learning tool that has been designed to educate people in technologies associated with controller tuning. Most aspects of autotuning are covered, and you are encouraged to try them out on industrial control practice.

The book is divided into 12 chapters. In Chapter 1, perspectives on process control and the need for automatic tuning of PID controllers are given. The PID controller is introduced in Chapter 2. Corresponding P, I, and D actions are explained and typical tuning rules are tabulated. Chapter 3 shows how and why the relay feedback tests can be used as a means of autotuning, and an autotuning procedure is also given. A simple and an improved algorithm are explored and analytical expressions for relay feedback responses are also derived. The shape of relay feedback is discussed in Chapter 4. This gives useful information on possible model structure and ranges of model parameters. Once model structure is available, an appropriate tuning rule can be applied for improved control performance. In Chapter 5, a ramp type of relay is proposed to provide better accuracy in identifying process parameters. The improved experimental design is shown to work well for both single-input–single-output (SISO) and multivariable systems. Chapter 6 is devoted to a more common situation: multivariable systems. Experiments are devised and procedures are given for the automatic tuning of multiloop SISO controllers. Chapter 7 is devoted to a practical problem: autotuning under load disturbance. A procedure is presented to find controller parameters under load changes. The multiple-model approach is known to be effective in handling processes that are nonlinear, and Chapter 8 extends the relay feedback autotuning in a multiple-model framework. In Chapter 9, the controller monitoring problem is addressed. Again, the shape of relay feedback response gives a useful indication on the appropriateness of the tuning constant. Moreover, monitoring and retuning are completed in a single-relay feedback test. The issue of an imperfect actuator is dealt with in Chapter 10. For control valve with hyteresis, an autotuning procedure is proposed to overcome the frequently encountered problem in practice. In Chapter 11, the importance of control structure design is illustrated using a plantwide control example. Procedures for the design of the control structure and the tuning of the entire plant are given and the results clearly indicate that the combination of better process understanding and improved tuning makes the recycle plant much easier to operate. Chapter 12 summarizes the guidelines for autotuning procedures and describes when and what type of relay feedback test should be employed.

The book is based on work my students and I have been engaged in for almost 20 years to improve PID controller performance. I wrote the book because I believe strongly in the benefits of improved control, and a well-tuned PID controller is a fundamental step for improved process operation.

Acknowledgements

Thanks are due to K. J. Åström, T. Hägglund, W. L. Luyben, Q. G. Wang, I. B. Lee, and my colleague H. P. Huang, who have contributed to the development of relay feedback autotuning. Undergraduate and graduate students and postdoc fellows of NTU and NTUST have contributed to this book by their questions and interest in the subject. In particular, the continuous feedback from Walters Shen, K. L. Wu, D. M. Chang, Y. C. Cheng, Y. H. Chen, T. Thyagarajan, and R. C. Panda needs to be acknowledged. The superb editing work of Brenda Tsai and Vincent Chang made this book possible. Finally, without the understanding and support of my family, this book would not have been undertaken, or completed.

Contents

1

Introduction

1.1 Scope of Process Control

Over past 50 years, "process control" has developed into a vital part of the engineering curriculum. Textbooks ranges from 600 to 1200 pages [1–3] and cover various aspects of industrial process control. It is hopeless to discuss all subjects of process control in this book. However, a brief description of the scope of process control will be given and the specific role of this book will become clear.

For continuous manufacturing, on-demand production with on-aim quality is the goal of process operation. Many factors contribute to non-smooth process operation, and controller tuning is just one of them, as shown in Figure 1.1 [4]. Starting from the most fundamental level, process variations may come from the *infrastructure* of a control system in which the signal transmission, control panel arrangement, distributed control system (DCS) selection, and DCS configuration may be the source of the problem. If the infrastructure is not the source of variation, then one may go up to the *instrumentation* level, which includes the control valve sizing, sensor selection, and transmitter span determination. It is clear that a wrongly sized control valve or an incorrectly determined transmitter span cannot provide adequate resolution in the manipulated variable or the controlled variable. It then comes to the *controller tuning* level in which inadequate controller settings may lead to oscillation in process variables, and improved controller settings is the focus of this book. If a controller retuning still cannot fix the problem, then we go to the *controller structure* level, in which one can try different types of controller. The actions in this level include: remove or add the derivative action, take out or add the integral action, use the gain scheduling, and add the dead time compensation. For example, the use of a proportional (P) only controller is often recommended for maximum flow smoothing in level control, and avoid using the derivative (D) action when the measurement is corrupted with noise. If the process variation is still significant, then it may be a problem in the *control configuration*. Experienced designers always establish loop pairings by maximizing the steady state gain between the controlled and the manipulated variables and by shortening

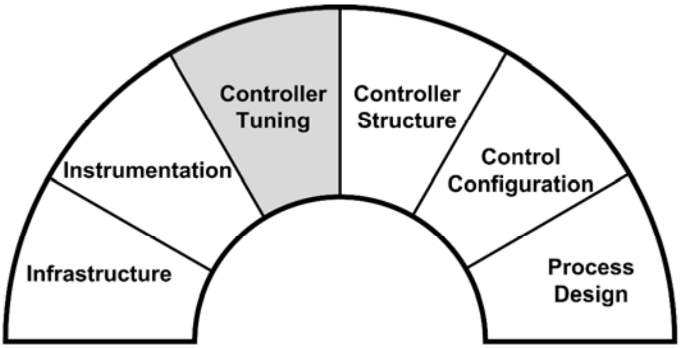


Figure 1.1. Spectrum of process operation

the response time (time constant) and dead time. Certainly, the inherent integral controllability should be maintained and the relative gain should be checked when dealing with multivariable systems. The other option is to explore the possibility of using a multivariable controller. However, one should be sure that we have enough engineering manpower for the maintenance of the much more complicated control system. Once all other possibilities are exhausted, we come to the rightmost part of the spectrum: *process design* can also be a possible cause of non-smooth operation. It has long been recognized that a process that has been design-based on some steady state economic objective will not necessarily provide good dynamic performance. This is especially true when new plants are typically designed using complex flowsheets with many streams for material recycles and for energy exchanges. The highly integrated plants generally lead to complex dynamics and difficulty in control and operation. Thus, in some cases, process redesign is required to ensure an operable process. The necessity of simultaneous design and control is advocated by Luyben and as can be seen in two recent books [4,5] and chapters of textbooks [2,3] devoted to this area. After studying the spectrum of process control, it should become clear that "controller tuning" only constitutes a fraction of the entire spectrum and it is even clearer that an improved controller tuning *cannot* solve all the problems associated with non-smooth process operation.

1.2 Proportional–Integral–Derivative Control Performance

Despite rapid evolution in control hardware, the proportional–integral–derivative (PID) controller remains the workhorse in process industries. The P action (mode) adjusts controller output according to the size of the error. The I action (mode) can eliminate the steady state offset and the future trend is anticipated via the D action (mode). These useful functions are sufficient for a large number of process applications and the transparency of the features leads to wide acceptance by the users. On the other hand, it can be shown that the internal model control (IMC) framework leads to PID controllers for virtually all models common in industrial practice [6].

Note that this includes systems with inverse responses and integrating (unstable) processes.

PID controllers have survived many changes in technology. It begins with pneumatic control, through direct digital control to the DCS. Nowadays, the PID controller is far different from that of 50 years ago. Typically, logic, function block, selector and sequence are combined with the PID controller. Many sophisticated regulatory control strategies, override control, start-up and shut-down strategies can be designed around the classical PID control. This provides the basic means for good regulatory, smooth transient, safe operation and fast start-up and shut-down. Moreover, even with model predictive control (MPC), PID controllers still serve as the fundamental building block at the regulatory level. The computing power of microprocessors provides additional features, such as automatic tuning, gain scheduling and model switching, to the PID controller. Eventually, all PID controllers will have the above-mentioned intelligent features.

In process industries, more than 97% of the regulatory controllers are of the PID type [7]. Most loops are actually under PI control (as a result of the large number of flow loops). More than 60 years after the publication of the Ziegler–Nichols tuning rule [8] and with the numerous papers published on the tuning methods since, one might think that the use of PID controllers has already met our expectations. Unfortunately, this is not the case. Surveys of Bialkowski [9], Ender [10], McMillan [11], Hersh and Johnson [12], and Desborough and Miller [7] show that:

1. Pulp and paper industry over 2000 loops [9]
   - Only 20% of loops worked well (*i.e.* less variability in the automatic mode over the manual mode).
   - 30% gave poor performance due to poor controller tuning.
   - 30% gave poor performance due to control valve problems (*e.g.* control valve stick-slip, dead band, backlash).
   - 20% gave poor performance due to process and/or control system design problems.
2. Process industries [10]
   - 30% of loops operated on manual mode.
   - 20% of controllers used factory tuning.
   - 30% gave poor performance due to sensor and control valve problems.
3. Chemical process industry [11]
   - Half of the control valves needed to be fixed (results of the Fisher diagnostic valve package).
   - Most poor tuning was due to control valve problems.
4. Manufacturing and process industries [12]
   - Engineers and managers cited PID controller tuning as a difficult problem.
5. Refining, chemicals, and pulp and paper industries over 26,000 controllers [7]

- Only 32% of loops were classified as “excellent” or “acceptable”.
- 32% of controllers were classified as “fair” or “poor”, which indicates unacceptably sluggish or oscillatory responses.
- 36% of controllers were on open- loop, which implies that the controllers were either in manual or virtually saturated.
- PID algorithms are used in vast majority of applications (97%). For the rare cases of complex dynamics or significant dead time, other algorithms are used. MPC acts less as a multivariable regulatory controller and more like a dynamic optimizer.

Surveys indicate that the process control performance is, indeed, “not as good as you think” [10], and the situation remains pretty much the same a decade later [7]. The reality leads us to reconsider the priorities in process control research. First, an improved process and control configuration redesign (*e.g.* selection and pairing of input and output variables) can improve control performance. As mentioned earlier, simultaneous design and control should be taken seriously to alleviate the problem of a small operating window and the requirement for sophisticated control configuration. Second, control valves contribute significantly to the poor control performance. It is difficult, if not impossible, to replace or to restore all the control valves to the expected performance. In other words, in many cases, this is a fact we have to face (*e.g.* dead band, stick-slip, *etc.* [13]). One thing we can do is to devise a diagnostic tool to identify potential problems in control valves. We have seen the beginning of research effort in this direction [14–17]. Third, and probably the easiest way to improve control performance, is to find appropriate tuning constants for PID controllers.

Sixty years after Ziegler and Nichols published their famous tuning rule, numerous tuning methods have been proposed in the literature. We do expect that engineers have gained proficiency in the design of simple PID controllers. The reality indicates that this is simply not the case. Moreover, the structure of current leaner corporations does not offer much opportunity to improve the situation. Another factor is the time required for the tuning of many *slow* loops (*e.g.* temperature loops in high-purity distillation columns). On many occasions, engineers simply do not have the luxury and patience to tune a loop over a long period of time (not being able to complete the task in a shift). It then becomes obvious that the PID controller with an automatic tuning feature is an attractive alternative for better control. That is, instead of continuous adaptation, the controller should be able to find the tuning parameters by itself: it is an autotuner.

Table 1.1 shows the current trend where major vendors provide one type or more autotuners in their products [18]. Identification methods include: open- or closed- loop step tests (step), relay feedback test (relay), and possibly pseudo-random binary signal (PRBS). The feature of gain-scheduling is also available in many of the products.

In devising such an automatic tuning feature, several factors should be considered:

Table 1.1. Autotuners from different vendors

Manufacturer	Identification method	Gain scheduling
ABB	Step/relay	Yes
Emerson Process Management	Relay	Yes
Foxboro	Step	No
Honeywell	Step	Yes
Siemens	Step	Yes
Yokogawa	Step	Yes

1. Control tuning can improve the performance, but it should be recognized that good tuning can only solve *part* of the problem.
2. The experimental design for system identification becomes rather important, since we are not able to keep all the control valves in perfect condition.
3. The system identification step should be time efficient. This is rather useful for many slow industrial processes.

1.3 Relay Feedback Identification

System identification plays an integral part in automatic tuning of the PID controller. Based on the information obtained, the methods for identification can be classified into the frequency-domain and time-domain approaches.

The time-domain approaches generate responses from step or pulse tests [2,3]. The characteristics of the process response are then utilized to back-calculate the parameters of an assumed process model [19]. The step tests can be performed in open-loop (manual mode) or closed-loop mode (while controller is working). The open-loop step test is fairly straightforward. However, it is vulnerable to load disturbances, especially for systems with large time constants. Moreover, the behavior of the control valve is not fully tested in the experiment. The closed-loop step tests, on the other hand, can shorten the time for experiment. But we have to choose a set of controller parameters in order to generate oscillatory (underdamped) responses [19]. The process model is then approximated from the damping behavior. The pattern recognition controller [20,21] is a typical example. Since step-like change is involved, it is not expected to work well for highly non-linear systems, (*e.g.* high purity distillation columns [22]).

Another category is the Ziegler–Nichols type of experimental design. Probably the more successful part of the Ziegler–Nichols method is *not* the tuning rule itself. Rather, it is the identification procedure: a way to find the important process information, ultimate gain K_u and ultimate frequency ω_u . This is often referred to as the trial-and-error procedure [2,3]. A typical approach can be summarized as follows:

1. Set the controller gain K_c at a low value, perhaps 0.2.
2. Put the controller in the automatic mode.
3. Make a small change in the set point or load variable and observe the response. If the gain is low, then the response will be sluggish.
4. Increase the gain by a factor of two and make another set point or load change.
5. Repeat step 4 until the loop becomes oscillatory and continuous cycling is observed. The gain at which this occurs is the ultimate gain K_u , and the period of oscillation is the ultimate period P_u ($P_u = 2\pi / \omega_u$).

This is a simple and reliable approach to obtain K_u and ω_u . The disadvantage is also obvious: it is time consuming. The present-day version is the relay feedback test proposed by Åström and Hägglund [23]. First, a continuous cycling of the controlled variable is generated from a relay feedback experiment and the important process information, K_u and ω_u , can be extracted directly from the experiment. The information obtained from the relay feedback experiment is exactly the same as that from the conventional continuous cycling method. It should be noticed that the relay feedback is an old and useful technique for feedback control, as can be seen from earlier results [24,25], and, here, a new meaning is assigned to the relay feedback. However, an important difference is that the sustained oscillation is generated in a *controlled* manner (*e.g.* the magnitude of oscillation can be controlled) in the relay feedback test. Moreover, in virtually all cases, this is a very efficient way, *i.e.* a one-shot solution, to generate a sustain oscillation. Applications of the Åström and Hägglund autotuner are found throughout process industries using single-station controllers or a DCS (Table 1.1). The success of this autotuner is due to the fact that the identification and tuning mechanism are so *simple* that operators understand how it works. It also works well in slow and highly nonlinear processes [22]. Over the past two decades, extensive research has been done on relay feedback tests. Refinements on the accuracy and improvements on the experimental design have been made. Discussions about potential problems, extensions to multivariable systems and incorporation of gain scheduling have also been reported. Luyben brings the autotuner to another level in which the "shape" of relay feedback can be utilized to identify the model structure. This motivates us for the revision. It is our view that the relay-feedback-based autotuners now can provide the necessary tools to improve control performance in a reliable way.

1.4 Conclusion

In this chapter we clearly define the scope of process control, and one should realize that the controller tuning only constitutes a fraction of the process operation problems. Surveys indicate that the PID controller is the major controller in process industries. After many years of experience, the control loops, often thought too simple, do not perform as well as one might expect. The failure comes from the lack of the required knowledge to maintain the control loops, to tune the controllers, to design an appropriate process for control and to design a suitable control configuration for a given process. Poor control performance may have many different causes. However, obtaining good tuning is always the most cost-effective way to improve control. You should recognize that controllers are working with imperfect valves, noisy sensors and frequent load disturbances. These factors have to be taken into account when you are designing the experiment to find controller parameters.

1.5 References

1. Ogunnaike BA, Ray WH. Process dynamics, modeling, and control. New York: Oxford University Press; 1994.
2. Luyben WL, Luyben ML. Essentials of process control. New York: McGraw-Hill; 1997.
3. Seborg DE, Edgar TF, Mellichamp DA. Process dynamics and control; 2nd ed. New York: John Wiley & Sons; 2004.
4. Luyben WL, Tyreus BD, Luyben ML. Plantwide control. New York: McGraw-Hill; 1999.
5. Seferlis P, Georgiadis MC. The integration of process design and control. 2nd ed. Amsterdam: Elsevier; 2004.
6. Morari M, Zafiriou E. Robust process control. Prentice Hall: Englewood Cliff; 1989.
7. Desborough L, Miller R. Increasing customer value of industrial control performance monitoring-Honeywell experience. In: 6th Int. Conf. Chemical Process Control, AIChE Symp. Series 326; Rawlings JB, Ogunnaike BA, Eaton JW. Eds. AIChE: New York; 2002.
8. Ziegler JG, Nichols NB. Optimum settings for automatic controllers. Trans. ASME 1942;12:759.
9. Bialkowski WL. Dream vs Reality: A view from both sides of the gap. Pulp Paper Can 1993;94:19.
10. Ender DB. Process control performance: Not as good as you think. Control Eng 1993;40:180.
11. McMillan GK. Tuning and control loop performance. Instrument Society of America: Research Triangle Park; 1994.
12. Hersh MA, Johnson MA. A study of advanced control systems in the workplace. Control Eng. Prac. 1997;5:771.
13. Cheng YC, Yu CC. Relay feedback identification for actuators with hysteresis. Ind. Eng. Chem. Res. 2000;39:4239.

14. Huang B, Shah SL. Control loop performance assessment: Theory and applications. Springer-Verlag; 1999.

15. Choudhury MAAS, Shah SL, Thornhill NF. A data-driven model for valve stiction. ADCHEM 2003; 2004 Jan. 11–14; Hong Kong; 2004.

16. Manabu K, Maruta H, Kugemoto H, Shimizu K. Practical model and detection algorithm for valve stiction. The 7th Int. Conference on Dynamics and Control of Process Systems (DYCOPS-7); 2004 July 5–7; Boston; 2004.

17. Rossi M, Scali C. A comparison of techniques for automatic detection of stiction: simulation and application to industrial data. J. Proc. Cont. 2005;15:505.

18. Hang CC, Lee TH, Ho TH. Adaptive control. Instrument Society of America: Research Triangle Park; 1993.

19. Yuwana M, Seborg DE. A new method for on-line controller tuning. AIChE J. 1982;28:434.

20. Bristol EH. Pattern recognition: An alternative to parameter adaptive PID controller. Automatica 1977;13:197.

21. Cao R, McAvoy TJ. Evaluation of pattern recognition adaptive PID controller. Automatica 1990;26:797.

22. Luyben WL. Derivation of transfer functions for highly nonlinear distillation columns. Ind. Eng. Chem. Res. 1987;26:2490.

23. Åström KJ, Hägglund T. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica 1984;20:645.

24. Tsypkin IZ. Relay control systems. Cambridge: Cambridge University Press; 1984.

25. Atherton DP. Nonlinear control engineering. New York: Van Nostrand Reinhold; 1982.

26. Luyben WL. Getting more information from relay feedback tests. Ind. Eng. Chem. Res. 2001;40:4391.

2

Features of Proportional–Integral–Derivative Control

2.1 Proportional–Integral–Derivative Controller

The proportional–integral–derivative controller consists of three simple actions, *i.e.* P, I, and D actions. Let us use a heat exchanger (a cooler to be exact) example to illustrate these three functions. Figure 2.1 shows the inlet stream is cooled to a specific temperature by exchanging heat with cooling water. So the controlled variable is the heat exchanger outlet temperature and the manipulated variable is the cooling water flow rate. The heat exchanger outlet temperature is measured using a thermocouple, and then it is converted into a signal, generally called the process variable (PV), which is compatible with the control system (typically in the range of 4–20 mA). The PV is compared with the set point (SP) and the controller output (CO) is generated based on the control algorithm. The controller output is further converted to an air pressure signal to drive the valve. In doing so, the real cooling water flow rate is set according to the stem position (determined by CO), size of the valve, pressure drop across the valve, and the valve characteristic. The feedback controller generates its move based on the error E between the SP and PVs, $E(t)=SP(t)-PV(t)$.

2.1.1 Proportional Control

The P controller changes its output CO in direct proportion to the error signal E.

$$CO = Bias + K_c(SP - PV) \tag{2.1}$$

The bias signal is the value of the controller output when there is no error. This is an intuitive and simple action which is quite similar to human behavior. Whenever we are far away from our goal, we make a larger adjustment, and when we come close to the target, a smaller step is taken. Here, K_c is called the *controller gain*, an adjustable parameter. Figure 2.2 shows the responses of a P controller with three values of K_c for a step decrease in the heat exchanger inlet temperature. It becomes

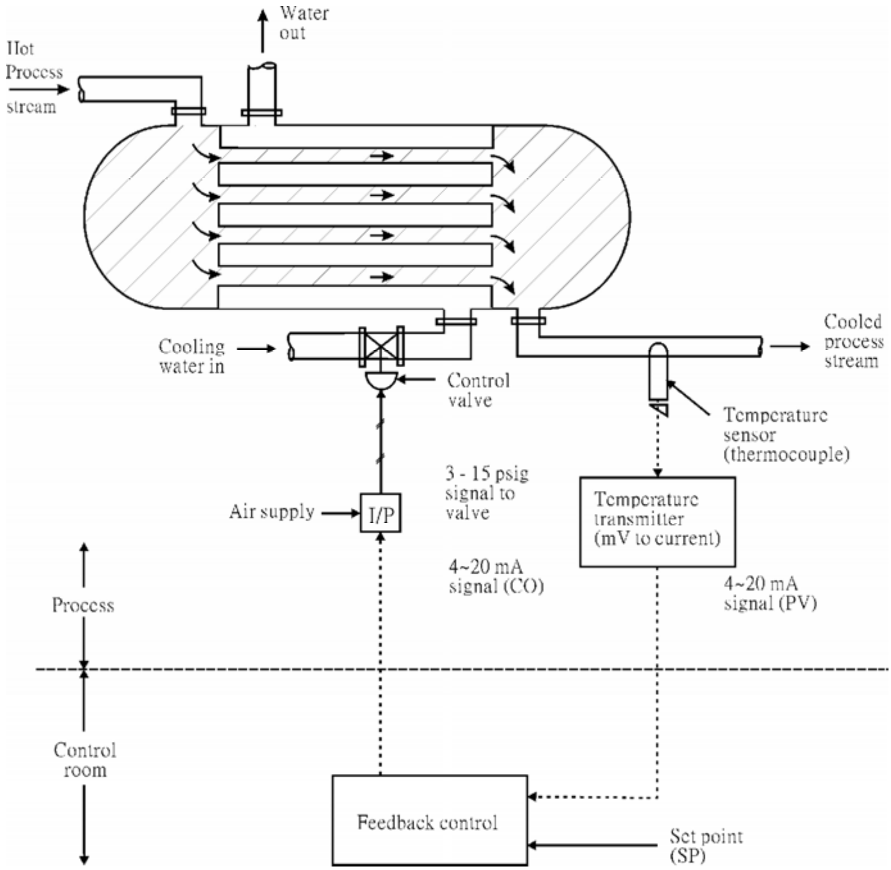


Figure 2.1. Process and control configuration of a heat exchanger

obvious that steady state errors (offset) exist for the P control. The responses indicate that an increase in the controller gain K_c can reduce the offset, but the response tends to be oscillatory. Certainly, when K_c is set to zero, the process is effectively open loop. To summarize the behavior of P control, we have: (1) it is a simple and intuitive, and (2) a steady state offset exists.

2.1.2 Proportional–Integral Control

In order to eliminate steady state offset, the I action is often included. I action moves the control valve in direct proportion to the time integral of the error. The resultant PI controller can be expressed as

$$CO = Bias + K_c\left((SP - PV) + \frac{1}{\tau_I}\int (SP - PV)dt \right) \tag{2.2}$$

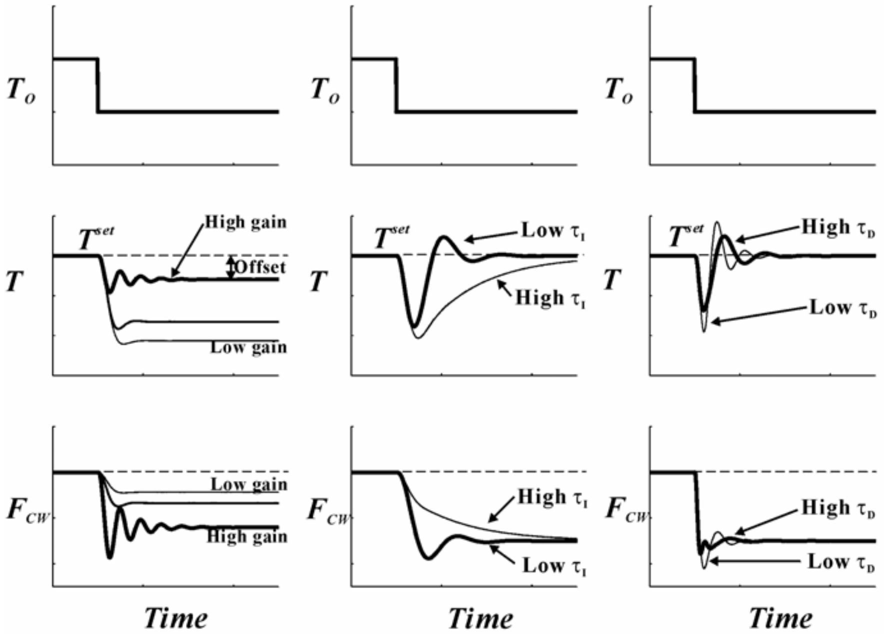


Figure 2.2. P, PI, and PID control performance using different controller settings for a step decrease in the heat exchanger inlet temperature T_o

Here, we have a second tuning parameter τ_I, which is called the *reset time* or the *integral time* with units of time (typically minutes). The PI controller equation indicates that the CO will keep changing until the difference between the SP and PV diminishes, *i.e.* E=0. This can be viewed as a relentless effort to meet the target by changing the input effort. In other words, the CO will not *rest* until the steady state error becomes zero. The integral action usually degrades the closed-loop performance. In a control notation, it introduces a 90° phase lag into the feedback loop. But the integral action is often needed for its ability to eliminate steady state offset. Figure 2.2 shows that, with I action, the heat exchanger outlet temperature does return to the set point. A smaller τ_I speeds up the temperature response while becoming a little oscillatory. It should be noticed that most of controllers (~70%) in industry are PI controllers. Instead of using the controller algorithm explicitly, most of the controller manuals express the PI controller in terms of a Laplace transformation (this is probably one of the few Laplace transformations you need to recognize when working in industry).

$$PI = \frac{CO}{E} = K_c\left(1+\frac{1}{\tau_I s}\right) = K_c\left(\frac{\tau_I s+1}{\tau_I s}\right) \tag{2.3}$$

2.1.3 Proportional–Integral–Derivative Control

The D action uses the *trend* of the process variable to make necessary adjustments. The process trend is estimated using the derivative of the error signal with respect to time. The *ideal* PID has the following form:

$$CO = Bias + K_c\left((SP - PV) + \frac{1}{\tau_I}\int (SP - PV)dt + \tau_D \frac{d(SP - PV)}{dt}\right) \tag{2.4}$$

Here, the third tuning parameter τ_D is the *derivative time* with units of time. It may be intuitive, appealing that the "process trend" can be incorporated into a control algorithm. We use these types of trend (or derivative) in numerical methods, *e.g.* Newton–Raphson method, and in stocks selling and buying. In theory, adding derivative action should always improve the dynamic response, and it should be the preference over the PI controller. The Laplace transformation of the *ideal* PID controller can be expressed as

$$PID_{ideal} = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right) \tag{2.5}$$

However, the ideal PID control algorithm has rarely been implemented in practice. Instead, a *filtered* D action is often used. The following is the *parallel* form of PID control with filtered D action:

$$PID_{parallel} = K_c\left(1 + \frac{1}{\tau_I s} + \frac{\tau_D s}{\alpha \tau_D s + 1}\right) \tag{2.6}$$

where α typically takes a value of 1/10. Figure 2.2 clearly shows that the PID controller outperforms the PI controller in the noise-free condition. But too large a τ_D will lead to significant oscillation in the controlled variable. However, when the process measurement is corrupted with noise, we have a completely different behavior, especially in the manipulated variable. Figure 2.3 indicates that the control valve is banging up and down, when we have fluctuating process measurements. This is certainly not desirable from the maintenance perspective. This also confirms why most controllers in industry are PI controllers, instead of PID controllers. This is typically true in chemical process industries when many flow loops are installed.

The P–I–D actions can be summarized as follows. P action is intuitive and effective, I action is relentless and offset free, and D action is the trend finder, but noise sensitive. After understanding the characteristic of each action, one should find the right combination of P–I–D actions for the controller to achieve good control performance.

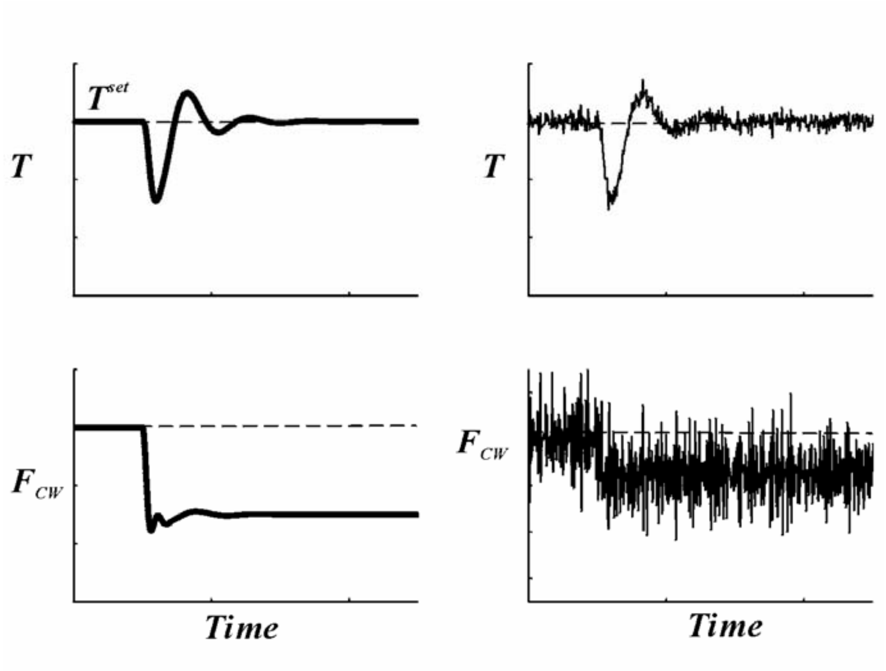


Figure 2.3. Load responses of PID controller with noise-free and noisy measurements

2.2 Proportional–Integral–Derivative Implementation

Two implementation issues for PID control are addressed. One is the anti-reset-windup associated with controllers with I action, and the other is the D action arrangement in a PID controller.

2.2.1 Reset Windup

The reset windup is an important and realistic problem in process control. It may occur whenever a controller contains the I action. When a sustained error occurs, the I term becomes quite large and the CO eventually goes beyond saturation limits (CO greater than 100% or less than 0%). Because all actuators have limitations, *e.g.* the flow through a control valve is limited by its size, and if the controller is asking for more than the actuator can deliver, there will be a difference between the CO and the actual control action (CO_A). When this happens, the controller is effectively disabled, because the valve remains unchanged, *e.g.* in full-open position. Not recognizing this circumstance, the controller continues to perform numerical integration, and the CO becomes even larger. It then requires (1) the error changing sign and (2) a long time to digest all the accumulated integrand, before the control valve moves away from the saturation limits. This is known as the *reset windup*. The consequence is a long transient and large overshoot in the controlled variable [1,2]. The reset windup may occur as a consequence of large disturbances or it may be caused by large SP changes, *e.g.* during the start-up of a batch process. Windup may also arise when the override control is used and so we have two controllers with only one control valve.

Conceptually, reset windup can be prevented by turning off the I action whenever the CO saturates. Many antiwindup methods have been proposed for different types of controller and for single-variable and multivariable systems [3,4]. One simple and effective approach for the integral windup is shown in Figure 2.4. The scheme involves a negative feedback loop around the I action with the CO in the loop. At normal operation (without saturation), the CO is equal to the actual control action CO_A, *i.e.* $CO= CO_A$, the feedback path disappears and the I action is in place. The actuator model is simply

$$CO_A = \begin{cases} 0 & CO<0 \\ CO & 0 \le CO \le 1 \\ 1 & CO > 1 \end{cases} \tag{2.7}$$

When the I action winds up, the actual control action remains unchanged, *e.g.* $CO_A=1$, and it can be treated as a reference value which is different from the controller output. Thus, the antiwindup scheme is best described by the following Laplace transformed relationship according to Figure 2.4:

$$CO(s) = \frac{1}{\tau s+1} CO_A(s) + \frac{\tau}{\tau s+1} E_I(s) \tag{2.8}$$

It becomes clear that the feedback loop tends to drive the CO to the actual control action following a first-order dynamics. The adjustable parameter τ is called the tracking time constant, and, typically, it is set to a small value. The antiwindup scheme now becomes a standard feature in commercial PID controllers.

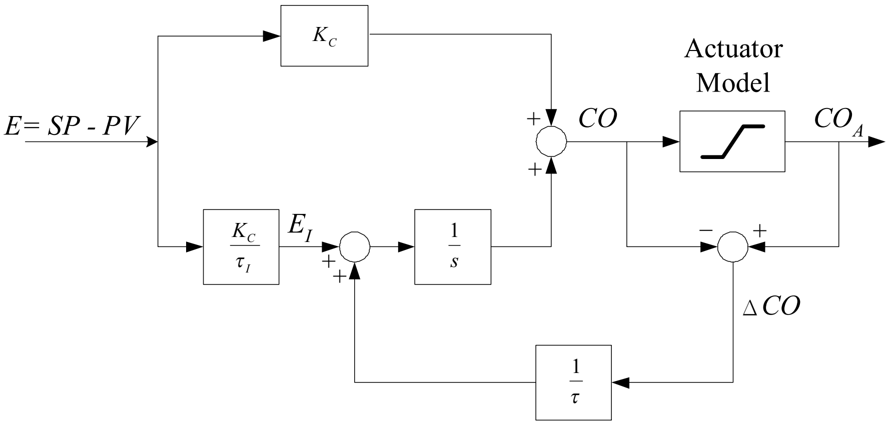


Figure 2.4. Antiwindup scheme with a tracking time constant τ

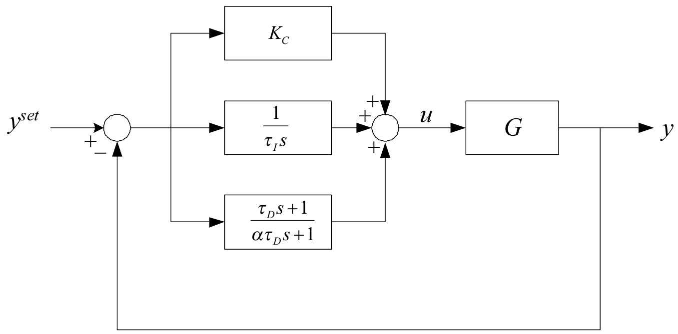
2.2.2 Arrangement of Derivative Action

For PI controllers, the proportional and the integral actions are additive, and the PI algorithm is universally used in all controllers. Unlike the PI controller, the PID controller appears to have many different forms. The two most common types are shown in Figure 2.5. The first type of PID controller has the three actions working additively. The continuous transfer function is given in Equation 2.6. It is called descriptively as the "parallel" form of PID controller. We label this as $PID_{parallel}$. The second type of PID controller can be expressed in terms of the following transfer function:

$$PID_{series} = K_c\left(\frac{\tau_I s+1}{\tau_I s}\right)\left(\frac{\tau_D s}{\alpha\tau_D s+1}\right) \tag{2.9}$$

This type of PID controller is known as the "series" form of PID controller, as can be seen from the equation and the block diagram arrangement in Figure 2.5. It was used in early analog controllers and has been implemented digitally in modern DCSs. Some of the popular tuning methods, *e.g.* Ziegler–Nichols [5], Tyreus and Luyben [6], and Luyben [7], are based on this algorithm. They also assume that the derivative filter parameter had a value of α=0.1. And yet another type of PID controller is the four-parameter PID controller, which is derived from the internal model control [8]. This is denoted as the "IMC" form of PID controller, PID_{IMC}. The following is the transfer function of the IMC PID controller:

(A) Parallel



(B) Series

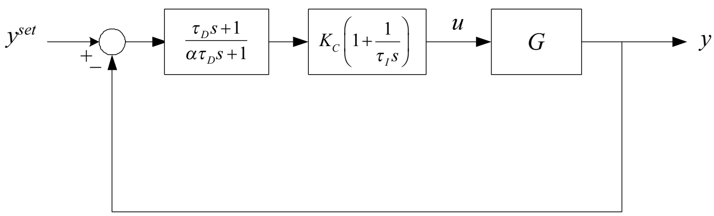


Figure 2.5. Parallel and series types of PID controller

$$PID_{IMC} = K_c\left(1+\frac{1}{\tau_I s}+\tau_D s\right)\frac{1}{\tau_F s+1} \tag{2.10}$$

Unlike the previous two types of PID controller, where α is a fixed value, the fourth tuning constant τ_F is also an adjustable parameter. These different PID forms clearly indicate that the settings of a PID controller depend on the algorithm used. The settings for the "series" and "parallel" can be very different, and one should always be aware of which algorithm the tuning rule is based on. However, the controller parameters for one algorithm can be transformed to the other as shown in Table 2.1. For example, the tuning constants of the "parallel" form can be transformed into the settings for the "series" form PID [2] and *vice versa.* Similarly, the relationship between the settings of PID_{IMC} and $PID_{parallel}$ can also be derived.

Another commonly used PID implementation is to take the derivative on the PV, instead of the error E = SP – PV, as shown in Figure 2.6. This can be understood, because a pure derivative of a step change corresponds to an impulse, and this implies a full swing of the control valve in an extremely short period of time, which is not desirable in practice. This is also known as the derivative kick [9]. This arrangement in Figure 2.6 is a standard feature in most commercial controllers.

Along this line, the PID controller can be extended further to a five-parameter controller by addressing the effects of derivative and proportional kicks [1,9].

$$CO = Bias + K_c\left((\beta\cdot SP-PV)+\frac{1}{\tau_I}\int(SP-PV)dt+\tau_D\frac{d(\gamma\cdot SP-PV)}{dt}\right) \tag{2.11}$$

Here, the two SP weightings, β and γ, are two additional adjustable parameters ranging from 0 to 1. This is often called the beta–gamma controller. The control algorithm in Equation 2.11 allows independent SP weightings in the proportional and derivative terms. To eliminate derivative kick, γ is set to zero, and similarly, β

Table 2.1. Interchangeable controller settings for different forms of PID controllers

$PID_{parallel} \rightarrow PID_{series}$ *	$PID_{series} \rightarrow PID_{parallel}$	$PID_{IMC} \rightarrow PID_{parallel}$
$K_{c,series} = \frac{K_C}{2}\left(1+\sqrt{1-\frac{4\tau_D}{\tau_I}}\right)$	$K_{c,parallel} = K_c\frac{\tau_I+\tau_D}{\tau_I}$	$K_{c,parallel} = K_c\frac{\tau_I-\tau_F}{\tau_I}$
$\tau_{I,series} = \frac{\tau_I}{2}\left(1+\sqrt{1-\frac{4\tau_D}{\tau_I}}\right)$	$\tau_{I,parallel} = \tau_I+\tau_D$	$\tau_{I,parallel} = \tau_I-\tau_F$
$\tau_{D,series} = \frac{2\tau_D}{1+\sqrt{1-\frac{4\tau_D}{\tau_I}}}$	$\tau_{D,parallel} = \frac{\tau_I\tau_D}{\tau_I+\tau_D}$	$\alpha = \frac{(\tau_I-\tau_F)\tau_F}{\tau_I\tau_D-(\tau_I-\tau_F)\tau_F}$
		$\tau_{D,parallel} = \frac{\tau_F}{\alpha}$

* Valid for $\tau_I/\tau_D \geq 4$