Contents

Preface xi

Acknowledgements xiii

Endorsement xv

About the authors xvii

1 A brief history of control and simulation 1
  1.1 Control 1
  1.2 Simulation 3
  1.3 References 10

2 Process control hardware fundamentals 13
  2.1 Control system components 13
  2.2 Primary elements 14
  2.3 Final control elements 30
  2.4 References 50

3 Fundamentals of single input—single output systems 51
  3.1 Open-loop control 51
  3.2 Disturbances 52
  3.3 Feedback control overview 53
  3.4 Feedback control: a closer look 56
  3.5 Process attributes: capacitance and dead time 61
  3.6 Process dynamic response 71
  3.7 Process modelling and simulation 73
  3.8 References 92

4 Basic control modes 93
  4.1 On—off control 93
  4.2 Proportional (P-only) control 95
  4.3 Integral (I-only) control 101
  4.4 Proportional plus integral (PI) control 104
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5 Derivative action</td>
<td>105</td>
</tr>
<tr>
<td>4.6 Proportional plus derivative (PD) controller</td>
<td>107</td>
</tr>
<tr>
<td>4.7 Proportional integral derivative (PID) control</td>
<td>110</td>
</tr>
<tr>
<td>4.8 Choosing the correct controller</td>
<td>111</td>
</tr>
<tr>
<td>4.9 Controller hardware</td>
<td>113</td>
</tr>
<tr>
<td>4.10 References</td>
<td>115</td>
</tr>
<tr>
<td>5 Tuning feedback controllers</td>
<td>117</td>
</tr>
<tr>
<td>5.1 Quality of control and optimisation</td>
<td>117</td>
</tr>
<tr>
<td>5.2 Tuning methods</td>
<td>122</td>
</tr>
<tr>
<td>5.3 References</td>
<td>130</td>
</tr>
<tr>
<td>6 Advanced topics in classical automatic control</td>
<td>131</td>
</tr>
<tr>
<td>6.1 Cascade control</td>
<td>131</td>
</tr>
<tr>
<td>6.2 Feedforward control</td>
<td>135</td>
</tr>
<tr>
<td>6.3 Ratio control</td>
<td>138</td>
</tr>
<tr>
<td>6.4 Override control (auto selectors)</td>
<td>140</td>
</tr>
<tr>
<td>6.5 References</td>
<td>146</td>
</tr>
<tr>
<td>7 Common control loops</td>
<td>147</td>
</tr>
<tr>
<td>7.1 Flow loops</td>
<td>147</td>
</tr>
<tr>
<td>7.2 Liquid pressure loops</td>
<td>149</td>
</tr>
<tr>
<td>7.3 Liquid level control</td>
<td>151</td>
</tr>
<tr>
<td>7.4 Gas pressure loops</td>
<td>162</td>
</tr>
<tr>
<td>7.5 Temperature control loops</td>
<td>163</td>
</tr>
<tr>
<td>7.6 Pump control</td>
<td>170</td>
</tr>
<tr>
<td>7.7 Compressor control</td>
<td>170</td>
</tr>
<tr>
<td>7.8 Boiler control</td>
<td>177</td>
</tr>
<tr>
<td>7.9 References</td>
<td>180</td>
</tr>
<tr>
<td>8 Distillation column control</td>
<td>183</td>
</tr>
<tr>
<td>8.1 Basic terms</td>
<td>183</td>
</tr>
<tr>
<td>8.2 Steady-state and dynamic degrees of freedom</td>
<td>184</td>
</tr>
<tr>
<td>8.3 Control system objectives and design considerations</td>
<td>186</td>
</tr>
<tr>
<td>8.4 Methodology for selection of a controller structure</td>
<td>188</td>
</tr>
<tr>
<td>8.5 Level, pressure, temperature and composition control</td>
<td>190</td>
</tr>
<tr>
<td>8.6 Optimizing control</td>
<td>198</td>
</tr>
<tr>
<td>8.7 Distillation control scheme design using steady-state models</td>
<td>202</td>
</tr>
<tr>
<td>8.8 Distillation control scheme design using dynamic models</td>
<td>213</td>
</tr>
<tr>
<td>8.9 References</td>
<td>214</td>
</tr>
<tr>
<td>9 Using steady-state methods in a multi-loop control scheme</td>
<td>215</td>
</tr>
<tr>
<td>9.1 Variable pairing</td>
<td>215</td>
</tr>
<tr>
<td>9.2 The relative gain array</td>
<td>216</td>
</tr>
<tr>
<td>9.3 Niederlinski index</td>
<td>221</td>
</tr>
<tr>
<td>9.4 Decoupling control loops</td>
<td>221</td>
</tr>
<tr>
<td>9.5 Tuning the controllers for multi-loop systems</td>
<td>223</td>
</tr>
</tbody>
</table>
## CONTENTS

9.6 Practical examples 223  
9.7 Summary 234  
9.8 References 234  

### 10 Plant-wide control 237  
10.1 Short-term versus long-term control focus 237  
10.2 Cascaded units 239  
10.3 Recycle streams 241  
10.4 General considerations for plant-wide control 246  
10.5 References 247  

### Appendices  

1 P&ID symbols 249  

2 Glossary of terms 253  

A2.1 Reference 259  

### Workshops  

1 Learning through doing 265  

2 Feedback control loop concepts 269  

3 Process capacity and dead time 275  

4 Feedback control 283  

5 Controller tuning for capacity and dead time processes 291  

6 Topics in advanced control 297  

7 Distillation control 307  

8 Plant operability and controllability 315  

Index 323
Preface

For decades, the subject of control theory has been taught using transfer functions, frequency-domain analysis, and Laplace transform mathematics. For linear systems (like those from the electromechanical areas from which these classical control techniques emerged) this approach is well suited. As an approach to the control of chemical processes, which are often characterized by nonlinearity and large doses of dead time, classical control techniques have some limitations.

In today’s simulation-rich environment, the right combination of hardware and software is available to implement a ‘hands-on’ approach to process control system design. Engineers and students alike are now able to experiment on virtual plants that capture the important non-idealities of the real world, and readily test even the most outlandish of control structures without resorting to non-intuitive mathematics or to placing real plants at risk.

Thus, the basis of this text is to provide a practical, hands-on introduction to the topic of process control by using only time-based representations of the process and the associated instrumentation and control. We believe this book is the first to treat the topic without relying at all upon Laplace transforms and the classical, frequency-domain techniques. For those students wishing to advance their knowledge of process control beyond this first, introductory exposure, we highly recommend understanding, even mastering, the classical techniques. However, as an introductory treatment of the topic, and for those chemical engineers not wishing to specialize in process control, but rather to extract something practical and applicable, we believe our approach hits the mark.

This text is organized into a framework that provides relevant theory, along with a series of hands-on workshops that employ computer simulations that test and allow for exploration of the theory. Chapter 1 provides a historical overview of the field. Chapter 2 introduces the very important and often overlooked topic of instrumentation. In Chapter 3 we ground the reader in some of the basics of single input – single output systems. Feedback control, the elements of control loops, system dynamics including capacitance and dead time, and system modelling are introduced here. Chapter 4 highlights the various PID control modes and provides a framework for understanding control-loop design and tuning. Chapter 5 focuses specifically on tuning. Armed with an understanding of feedback control, control loop structures, and tuning, Chapter 6
introduces some more advanced control configurations including feed-forward, cascade, and override control. Chapter 7 provides some practical rules of thumb for designing and tuning the more common control loops found in industry. In Chapter 8 we tackle a more complex control problem: the control of distillation columns. As with the rest of this text, a combination of theory and applied methodology is used to provide a practical treatment to this complex topic. Chapter 9 introduces the concept of multiple loop controllers. In Chapter 10 we take a look at some of the important issues relating to the plant-wide control problem. Finally, up-to-date information on computer simulation for the workshops can be found on the book website.

Although this text is designed as an introductory course on process control for senior university students in the chemical engineering curriculum, we believe this text will serve as a valuable desk reference for practising chemical engineers and as a text for technical colleges.

We believe the era of real-time, simulation-based instruction of chemical process control has arrived. We hope you’ll agree! We wish you every success as you begin to learn more about this exciting and ever changing field. Your comments on and suggestions for improving this textbook are most welcome.

William Y. Svrccek
Donald P. Mahoney
Brent R. Young
About the authors

William Svrcek is a Professor of Chemical and Petroleum Engineering at the University of Calgary, Alberta, Canada. He received his BSc (1962) and PhD (1967) degrees in Chemical Engineering from the University of Alberta, Edmonton. Prior to joining the University of Calgary he worked for Monsanto Company as a senior systems engineer and as an Associate Professor (1970–1975) in the Department of Biochemical and Chemical Engineering at the University of Western Ontario, London, Ontario. Dr Svrcek’s teaching and research interests centre on process simulation control and design. He has authored or co-authored over 150 technical articles/reports and has supervised over 30 graduate students. He has been involved for many years in teaching the continuing education course titled ‘Computer Aided Process Design – Oil and Gas Processing’ that has been presented worldwide. Most recently this course has been modified to include not only steady-state simulation, but also dynamic simulation and control strategy development and verification. Dr Svrcek was also a senior partner in Hyprotech, now part of Aspen Technology, from its incorporation in 1976. As a Principal, Director, and President (1981–1993) he was instrumental in establishing Hyprotech as a leading international process simulation software company. He is currently providing leadership and vision in process simulation software as the President of Virtual Materials Group Inc. He is a registered Professional Engineer, in both Alberta and Ontario, and a member of professional societies that include The Canadian Society for Chemical Engineering, American Institute for Chemical Engineers, Canadian Gas Processors Association and the Instrument Society of America.

Donald Mahoney is co-founder and Chief Operating Officer with BDMetrics, Inc., a company that develops and markets web-based analytics software. Mr Mahoney earned a Bachelor’s Degree in Mechanical Engineering from Penn State, a Master’s Degree in Control Theory from Purdue University, and an MBA from the University of Delaware. Mr Mahoney has held research and teaching positions at the US Navy’s Applied Research Lab and Purdue University, where he was awarded the staff’s ‘Outstanding Teaching Award’. He has also lectured extensively on process simulation and control topics, and has published a number of journal articles in the field. Prior to joining BDMetrics, Mr Mahoney was Vice President with AEA Technology Engineering Software/Hyprotech where he led the introduction and launch of more than a half dozen
design, modelling and optimization software products. He has held industrial positions at General Motors and DuPont as a control systems engineer and process modelling and control consultant. While at DuPont, Mr Mahoney was involved in the development and support of the chemical industry’s first object-oriented dynamic simulation package, TMODSTM.

**Brent Young** is Senior Lecturer of Chemical and Materials Engineering at the University of Auckland, New Zealand. He received his BE (1986) and PhD (1993) degrees in Chemical and Process Engineering from the University of Canterbury, New Zealand. Prior to his graduate studies, he worked as a Chemical Engineer for Ravensdown Fertilizer Coop’s Super Phosphate Plant in Christchurch and developed a process model for the simulation of a rock phosphate grinding circuit. In 1991, he joined the University of Technology in Sydney, Australia, as a lecturer, received tenure in 1994 and was promoted to Senior Lecturer in 1996, continuing his research in the areas of modelling and control of processes, particularly industrial processes. He was an Associate Professor of Chemical and Petroleum Engineering at the University of Calgary from late 1998 to the end of 2005. He joined the University of Auckland in January 2006. He is a registered Professional Engineer and a member of a number of professional societies. His research is centred on the two major areas of process simulation and control, and process design and development – particularly the processing of carbonaceous substances.
Basic control modes

The previous chapter discussed basic feedback control concepts, including the vital role of the controller. Again, the purpose of the controller in regulatory control is to maintain the controlled variable at a predetermined set point. This is achieved by a change in the manipulated variable using a pre-programmed controller algorithm. This chapter will describe the basic control modes or algorithms used in controllers in feedback control loops.

4.1 On–off control

The most rudimentary form of regulatory control is on–off control. This type of control is primarily intended for use with FCEs that are non-throttling in nature, i.e. some type of switch as opposed to a valve. An excellent example of on–off control is a home heating system. Whenever the temperature goes above the set point, the heating plant shuts off, and whenever the temperature drops below the set point, the heating plant turns on. This behaviour is shown by Equation 4.1:

$$MV = 0\% \text{ for } PV > SP \text{ and } MV = 100\% \text{ for } PV < SP$$  \hspace{1cm} (4.1)

The controller output MV is equal to 0 per cent or off whenever PV exceeds the set point SP. Whenever the process variable is below the set point, the controller output is equal to 100 per cent or on.

The most useful type of process where on-off control can be successfully applied is a large capacitance process where tight level control is not important, i.e. for the case of flow smoothing. A good example of this type of process is a surge tank. A large capacitance is important owing to the nature of the controller action and its effect on the operational life of the FCE. This leads us to one of the disadvantages of an on-off type of controller. Owing to the continual opening and closing of the controller, the FCE quickly becomes worn and must be replaced. This type of control action is illustrated in Figure 4.1, which shows the typical behaviour of an on–off controller.

In this example, at time $t = 0$, PV is less than SP, and MV is equal to 100 per cent. When PV crosses the set point, MV becomes 0 per cent. The temperature rises somewhat
To allegorise proportional control we will use the liquid level loop shown in Figure 4.3. Initially, the proportional controller is placed in manual and the level in the tank is manually adjusted to equal the set point. With $F_i$ equal to $F_0$, the level should stay at the set point. Also, set $F_0 = 50\% = F_i$, $CV = SP = 50\%$, and $K_c = 2$. Note that here we are using the percentage of span units. Now, if the controller is placed into automatic mode what will happen to the output? At the instant the controller is placed into automatic mode, the error will be zero since $CV$ is equal to $SP$, and the controller output will also be zero:

$$MV = 2(50 - 50) = 0 \text{ (from Equation 4.2)}$$

For a controller output of zero, what will the level do? The level will begin to drop. To stop this movement $F_i$ and $F_0$ must equal 50 per cent again. If a linear relationship is assumed between inflow and controller output, then for $F_i = 50\%$ we will have $MV = 50\%$.

$$\text{Since } MV = K_c e = 2(SP - CV) = 2(50\% - CV)$$
$$\text{For } MV = 50\%, e = 25\%$$
$$\text{Therefore } MV = 50\% \text{ when } CV = 25\%$$

Thus, the controller output becomes 50 per cent when the measurement $CV$ drops by 25 per cent, creating a 25 per cent error. For this case, in order to stop the level from dropping, the proportional controller had to drop $CV$ to create a large enough error so the controller could make $F_i = F_0$.

Suppose we set $K_c = 4$, giving $MV = 4e$. Now, the error would only need to be 12.5 per cent for $MV = 4(12.5\%) = 50\%$. Logically, it would appear that the larger the controller gain, the smaller the error. In theory, if $K_c$ is set to infinity then the error can be reduced to zero. The problem with this extrapolation is that the controller gain $K_c$, is multiplied by all the gains of the other elements to give the loop gain $K_L$. If $K_c$ becomes large enough, then the loop gain will be greater than unity, thus causing the
loop to become unstable. Because of this loop gain limit, there is a limit to how large the controller gain can be. However, there is another approach to reducing the error to zero. Suppose another term is added to the proportional controller equation:

\[ MV = K_c e + b \]  

(4.3)

This additional term is called the bias \( b \), and is simply defined as the output of the controller when the error is zero. Using the previous example again, let us set \( K_c = 2 \). Also, manually adjust \( CV = SP = 50\% \), \( F_i = F_0 = 50\% \), and set \( b = 50\% \). Now, when the controller is set to automatic, what will happen? Since \( CV \) is equal to \( SP \), \( e \) is equal to zero and, hence, \( K_c e = 2(0) = 0 \). There is no proportional contribution to the output and the output \( MV \) is equal to the bias, which is 50 per cent (Equation 4.3). Since \( F_0 \) is equal to 50 per cent and \( MV \) is also equal to 50 per cent, the level will stay the same. In general, if the bias equals the load \( MV \), then the error will always be zero.

Now, suppose \( F_0 \) becomes 75 per cent. In order to stop the level from dropping, \( MV \) must equal \( F_0 \), which in this case is 75 per cent. From Equation 4.3, \( MV = 2e + 50\% = 2(50\% - CV) + 50\% \), and \( CV \) must drop to 37.5 per cent to make the output \( MV \) equal to 75 per cent. When \( MV \) is equal to the outflow, the level will stop dropping. The level could also be prevented from dropping if the outflow \( F_0 \) was decreased. Suppose \( F_0 \) is equal to 25 per cent. In this case, the level will stop rising when \( CV \) is equal to 62.5 per cent, since that gives \( MV = 2(50\% - CV) + 50\% = 25\% \).

As mentioned previously, increasing \( K_c \) can decrease the error; but remember not to increase \( K_c \) such that it makes the loop unstable. There is a limit for each feedback control loop. If \( K_c \) has a value such that the loop gain \( K_L = 1 \), then the loop will oscillate with a period that is a function of the natural characteristics of the process. This is called the natural period \( \tau_n \). If \( K_c \) is adjusted such that the loop gain is equal to 0.5 and a change is made in \( F_0 \), then the response shown in Figure 4.4 could be expected.

\[ F_0 \]
\[ CV \]
\[ MV_0 \]
\[ t \]
\[ \tau_n \]
\[ e \]
\[ SP \]
\[ MV_1 \]

Figure 4.4  Typical proportional-only controller response
CV damps out with a quarter decay ratio (discussed in greater detail in Chapter 5) and a period approximately equal to the natural period. It then stabilizes with an offset that is a function of both the controller gain and the bias. The offset is the sustained error $e$, where CV does not return to the set point even when steady state is reached. This is a typical response for a loop under proportional-only control.

Now let us look again at Equation 4.3, and recall that the gain of any loop element is defined by

$$K = \frac{\Delta \text{output}}{\Delta \text{input}}$$  \hspace{1cm} (4.4)

The block diagram of a proportional controller can be represented as shown in Figure 4.5.

The controller gain is the ratio of the change in controller output to the change in error. Hence, the gain of the proportional controller $K_c$ is given by

$$K_c = \frac{\Delta MV}{\Delta e}$$  \hspace{1cm} (4.5)

Since there is a one-to-one relationship between CV and $e$, the controller gain can be written as

$$K_c = \frac{\Delta MV}{\Delta CV}$$  \hspace{1cm} (4.6)

The controller gain can also be defined as a change in controller output for a change in the process variable PV. This is true because the controlled variable CV is the transformed process variable from the transmitter to the controller. Therefore, CV is essentially PV, only in different units, i.e. percentage level instead of milliamperes.

Assuming that a linear relationship exists between CV and MV, as shown in Figure 4.6, Equation 4.6 may be written as

$$K_c = \frac{\Delta MV}{\Delta CV} = \frac{100\%}{\Delta CV}$$  \hspace{1cm} (4.7)
4.2 PROPORTIONAL (P-ONLY) CONTROL

Figure 4.6  Controller input–output relationship

The controller gain $K_c$ in Equation 4.7 is the amount that CV must change to make the controller output change by 100 per cent. The gain of the transmitter is similar and is given by

$$K_T = \frac{\Delta\text{out}}{\Delta\text{in}} = \frac{100\%}{\text{span}}$$  \hspace{1cm} (4.8)

In other words, the input of the transmitter changes by the amount of the span to make the transmitter output change by 100 per cent. The span is the difference between the upper and lower values of the range.

The case of the controller is analogous to that of the transmitter, but instead of calling $\Delta CV$ the span, it is called PB, the proportional band. The proportional band is defined as the change in CV that will cause the output of the controller to change by 100 per cent. Using this definition of PB, we can be define the controller gain as

$$K_c = \frac{\Delta\text{MV}}{\Delta\text{CV}} = \frac{100\%}{\text{PB}\%}$$  \hspace{1cm} (4.9)

If the proportional band setting on the controller is set to 40 per cent, the output of the transmitter, which is the input to the controller, changes over 40 per cent of its output span. The output of the controller would change by 100 per cent, or the controller gain $K_c$ would be

$$K_c = \frac{100\%}{40\%} = 2.5$$

Virtually all modern controllers use a gain adjustment; however, a few older controllers exist that still use a proportional band adjustment. Remember that $K_c = 100\%/\text{PB}\%$, or as the PB gets larger, the gain gets smaller and vice versa. The equation for a proportional controller in terms of PB can be written as

$$\text{MV} = \frac{100}{\text{PB}} e + b$$  \hspace{1cm} (4.10)
Note that:

\[ e = SP - CV \] (for reverse acting)
\[ e = CV - SP \] (for direct acting)

In order to make the error equal to zero, one of the following two possibilities must occur:

1. set \( PB = 0 \) (\( K_c = \infty \))
2. set \( b = MV \).

The first option, as discussed previously, is not plausible since as \( PB \to 0 \), \( K_c \to \infty \) and the loop becomes unstable. Furthermore, it is not possible to set \( PB = 0 \), because on many controllers the minimum setting is usually 2 to 5 per cent. However, if \( PB \) is very small, i.e. 2 per cent or \( K_c = 50 \), then the error would certainly be minimised, provided the loop remained stable. This case can be illustrated using Figure 4.7.

If in Figure 4.7 \( KV K_p K_T < 1/50 \), then the loop would be stable since the loop gain \( K_L < 1 \) (Equation 2.6). If the process had a lower gain \( K_p \), then a higher controller gain or smaller PB in the P-only controller could be used to minimize the error. One type of process where this is the case is a very large capacitance process, i.e. a large surge tank. Owing to the low process gain, a P-only controller is often used for level control.

The second option to make the error zero is to set the bias equal to the controller output MV. Some controllers have an adjustable bias, hence making this option viable, as in Equation 4.11:

\[ e = \frac{1}{K_c} (MV - b) \] (4.11)

However, this approach is only an option for processes that experience few load upsets, since a manual readjustment of the bias is required each time there is a load upset. There would be no error as long as the bias was equal to the load. Hence, if the
process had infrequent load upsets, the operators could readjust the bias to give zero error, and it would be possible to use a P-only controller.

In general, a proportional controller provides a fast response when compared with other controllers, but a sustained error occurs where the PV does not return to the set point even when steady state is reached. This sustained error is called offset and is undesirable in most cases. Therefore, it is necessary to eliminate offset by combining proportional control with one of the other basic control modes.

### 4.3 Integral (I-only) control

The action of integral control is to remove any error that may exist. As long as there is an error present, the output of this control mode continues to move the FCE in a direction to eliminate the error. The equation for integral control is

$$MV = \frac{1}{T_i} \int e \, dt + MV_0$$

(4.12)

$MV_0$ is defined as either the controller output before integration, the initial condition at time zero, or the condition when the controller is switched into automatic. The block diagram for an integral-only controller is given in Figure 4.8.

The action or response of the integral control algorithm for a given error is shown in Figure 4.9, assuming increase/decrease action.

If the measurement CV is increased in a step-wise fashion at time $t_1$ and then returned to the set point at $t_2$, the output would ramp up over the interval $t_1 < t < t_2$ since the controller is in effect integrating the step input. When the measurement is returned to the set point at $t = t_2$, the output would hold the value that the controller had integrated to, since the controller would think this was the correct value or the set point, i.e. $e = 0$.

The rate at which the controller output ramps is a function of two parameters: the integral time $T_i$, and the magnitude of the error. Obviously, the controller output MV would ramp in the opposite direction if CV had been moved below the set point.

The integral time $T_i$ is defined as the amount of time it takes the controller output to change by an amount equal to the error. In other words, it is the amount of time required to duplicate the error. Thus, $T_i$ is measured in minutes per repeat. Because of

![Figure 4.8 Block diagram of integral-only controller](image)
the form of Equation 4.12 some manufacturers measure the reciprocal of $T_i$ or repeats per minute in a controller:

$$\frac{1}{T_i} = \frac{1}{\text{min/repeat}} = \text{repeats/min}$$

(4.13)

As a result of this reciprocal relationship, if the controller is adjustable in min/repeat, then increasing the adjustment gives less integral action, whereas in repeats/min, increasing the number produces greater integral action. Therefore, it is important to be aware of how an individual controller adjusts $T_i$. The rate of change of MV also depends on the magnitude of $e$ as shown in Figure 4.10, in which $T_i$ is fixed.

Figure 4.11 illustrates the responses of P-only, I-only and PI controllers to a step input. Although an integral-only controller provides the advantage of eliminating offset, there is a significant difference in its response time when compared with a proportional-only controller. As mentioned earlier, the output of the proportional-only controller changes
as quickly as the measurement changes; in other words, the controller tracks the error. So, if the measurement changes as a step, then the controller output also changes as a step by an amount depending on the controller gain. For a step input to an integral controller the output does not change instantaneously, but rather by a rate that is affected by $T_i$ and $e$.

Hence, integral-only control, due to the additional lag introduced by this mode, has an overall response that is much slower than that for proportional-only control. The period of response for the PV under integral-only control can be up to 10 times that for proportional-only; so a trade-off is made when using an I-only controller. If no offset is required, then a slower period of response must be tolerated. If the requirement is a return to the set point with no offset, and a faster response time is necessary, then the controller must be composed of both proportional and integral action.

As a result of the above, controllers with both proportional and integral action are more common. However, a few examples of integral-only controllers do occur. In the Claus sulphur plant air-demand controller [1] the trim air valve position error may be used to drive the main air valve position using only integral action. The combination of small (trim) and large (main) valves permits fine control of the Claus sulphur plant furnace air demand. In energy or ‘BTU’ control of a coal-fired power station [2] the integral-only control compares the energy leaving the boiler (steam) with the energy entering the boiler (coal). If there is a sustained difference, then the integral-only controller modifies the energy content of the coal feed until there is an energy balance. The integral time is very small, and is intended to compensate for the energy content of the coal supplied. Typically, it takes some 12h for the BTU content of the coal to change a significant amount.

![Figure 4.11](image-url) Response of P-only, I-only and PI controllers
4.4 Proportional plus integral (PI) control

A *proportional plus integral controller* will give a response period that is longer than a P-only controller but much shorter than an I-only controller. Typically, the response period of the process variable PV under PI control is approximately 50 per cent longer than for the P-only (1.5\(\tau_n\), Figure 4.11). Since this response is much faster than I-only, and only somewhat longer than P-only control, the majority (>90 per cent) of controllers found in plants are PI controllers. The equation for a PI controller is

\[
 MV = K_c(e + \frac{1}{T_i} \int e \, dt) = K_c e + K_c \frac{1}{T_i} \int e \, dt
\]  

(4.14)

The PI controller gain has an effect not only on the error, but also on the integral action. When we compare the equation for a PI controller (Equation 4.14) with that for a P-only controller (Equation 4.11) we see that the bias term in the P-only controller has been replaced by the integral term in the PI controller. Thus, the bias term for PI control is given by

\[
b = K_c \frac{1}{T_i} \int e \, dt
\]  

(4.15)

Therefore, the integral action provides a bias that is automatically adjusted to eliminate any error. The PI controller is faster in response than the I-only controller because of the addition of the proportional action, as illustrated in Figure 4.12.

As shown in Figure 4.9, it takes \(T_i\) minutes for the output of the I-only controller to duplicate the error. With the addition of proportional action there is an immediate proportional step followed by integral action. The integral time in this case is defined as the amount of time it takes for the integral portion of the controller to replicate the proportional action. When the measurement is returned to the set point, the proportional action is lost, since \(e = 0\), and the controller output is determined solely by integral action.
As can be seen from Equation 4.16, $K_{PI}$, the gain of the PI controller, is the sum of the two component gains. These component gains are proportional action $K_P$ and integral action $K_I$.

$$K_{PI} = K_P + K_I = K_c + \frac{K_c}{T_i}$$ (4.16)

The $K_c$ and $T_i$ are used to adjust the PI controller gain to give the loop a desired response. Suppose $T_i = \infty$, which would result in $K_I = 0$, regardless of the value of $K_c$. In effect, the response would be that of a P-only controller with a period equal to $\tau_n$ and a sustained error. Although $T_i = \infty$ is not realizable, it can be set to a very large number in min/repeat to minimize the integral action.

Now, suppose $T_i$ were set to a very small value. In this case, the PI controller gain would approach that of an integral-only controller, since $K_I \gg K_P$. The control action in the loop would now be that of an I-only controller with a return to the set point, but with a long response period.

These are two extremes, and somewhere in between is a $T_i$ that will give a return to the set point with a reasonable response period of $1.5\tau_n$. The selection of $T_i$ will be discussed in more detail under controller tuning in Chapter 5.

In general, starting with only proportional action, as more integral action is added, the PV begins to return to the set point. We only want enough integral gain to return to the set point, since a $K_I$ greater than this will only serve to lengthen the response period. As more integral action is added by reducing $T_i$, we must compensate for the increased integral gain by reducing the proportional gain. Adjusting $T_i$ will have an effect on $K_1$, and thus affects $K_{PI}$; this, in turn, affects both the damping and the response period. Adjusting $K_c$ affects both $K_1$ and $K_P$ equally; thus $K_c$ only has an effect on $K_{PI}$, affecting the damping and not the response period. These interacting effects will be considered in more detail under controller tuning in Chapter 5.

Although the response period of a loop under PI control is only 50 per cent longer than that for a loop under P-only control, this may in fact be far too long if $\tau_n$ is as large as 3 or 4 h. In order to increase the speed of the response, it may be necessary to add an additional control mode.

### 4.5 Derivative action

The purpose of **derivative action** is to provide lead to overcome lags in the loop. In other words, it anticipates where the process is going by looking at the rate of change of error $\frac{de}{dt}$. For derivative action, the output equals the derivative time $T_d$ multiplied by the derivative of the input, which is the rate of change of error:

$$\text{output} = T_d \frac{de}{dt}$$ (4.17)
Figure 4.13 shows how the output from a derivative block would vary for different inputs given a fixed value of $T_d$.

As the rate of change of the input gets larger, the output gets larger. Since the slope of each of these input signals is constant, the output for each of these rate inputs will also be constant. However, what happens as the slope approaches infinity as in the case of a step change, (4) in Figure 4.13. Theoretically, the output should be a pulse that is of infinite amplitude and zero time long. This output is unrealizable, since a perfect step with zero rise time is physically impossible, but signals that have short rise and fall times do occur. These types of signal are referred to as noise. Thus, the output from the derivative block would be a series of positive and negative pulses, which would try to drive the FCE either full open or full close. This would result in accelerated wear on the FCE and no useful control.

Consider a temperature measurement with a small-amplitude and high-frequency noise. One might assume that since the noise is of such small amplitude in comparison with the average temperature signal that a controller would not even notice it. This is only the case if the controller does not have derivative action. If the controller contains derivative action, then the temperature signal would be completely masked by the noise into the derivative mode of the controller, and the controller output would be a series of large-amplitude pulses, entirely masking any output contributed by the other control modes. Fortunately, in a case such as this the noise is either easily filtered out or is eliminated by modifying the installation of the primary sensor.

However, there are cases where noise is inherent in the measurement of PV and the rise and fall time of the noise is of the same magnitude as that of the measurement itself. In such a case, noise filtering would only serve to degrade the accuracy of the measurement of PV. A good example of a situation like this is a flow control loop. Flow measurement by its very nature is noisy; therefore, derivative action cannot be successfully applied.

It is important to note that derivative control would never be the sole control mode used in a controller. The derivative action does not know what the set point actually is
and hence cannot control to a desired set point. Derivative action only knows that the error is changing.

### 4.6 Proportional plus derivative (PD) controller

The minimum controller configuration containing derivative action is the combination of proportional plus derivative action shown in Equation 4.18. This combination is not used very often and is primarily applied in batch pH control loops. However, it will help in the definition of derivative time $T_d$.

\[
MV = K_c \left( e + T_d \frac{de}{dt} \right) + b \tag{4.18}
\]

In Equation 4.18, the PD controller equation contains a bias term. A bias term will normally appear in any controller algorithm that does not contain integral action. This bias term does not appear when integral action is present, since integral action is in effect an automatic adjustment of bias. As with the PI controller, the proportional gain acts on the error as well as the derivative time $T_d$. Figure 4.14 shows the controller output MV for a typical input $e$ test signal for the proportional and derivative portions of a PD controller.

In Figure 4.14, $MVP$ is the proportional portion of the output and $MVD$ is the derivative portion. In the example, the measurement changes at a fixed rate of change; therefore, the derivative portion of the output is constant and depends on the rate of change, the derivative time $T_d$, and proportional gain $K_c$. This dependency is evident from Equation 4.18. The proportional output is a ramp whose slope is a function of the proportional controller gain $K_c$.

![Figure 4.14 Responses for P-only and D-only portions of a PD controller](image)
Now, let us superimpose MVP and MVD to get the actual output for a PD controller, as shown in Figure 4.15.

For a ramp input it takes a period of time for the proportional action to reach the same level as the derivative action. This period of time is called the derivative time $T_d$ and is measured in minutes. Increasing the derivative time $T_d$ increases MVD, or the contribution of the derivative action to the movement of the final control element.

In Equation 4.18, for the PD controller the derivative action acts on the error. Since $e = SP - CV$ for I/D action, $de/dt$ is a function of both the derivative of the set point $dSP/dt$, and the derivative of the controlled variable $dCV/dt$:

$$\frac{de}{dt} = \frac{dSP}{dt} - \frac{dCV}{dt}$$ (4.19)

If there is a load upset to the process, then the process variable PV will change at some rate $dCV/dt$ which will result in the error also changing at the same rate ($de/dt = -dCV/dt$), assuming there is no set-point change. Now, if a set-point change of even a few per cent is made and if the set point is changed quickly, then $dSP/dt$ can become very large. This would cause a large pulse to be generated at the output of the controller. To overcome this potential problem, the controller can be made so that the derivative mode simply ignores set-point changes, as shown in Equations 4.20–4.22.

$$\frac{de}{dt} = \frac{dSP}{dt} - \frac{dCV}{dt}$$ (4.20)

Ignoring set-point changes gives

$$\frac{de}{dt} = \frac{-dCV}{dt}$$ (4.21)
4.6 PROPORTIONAL PLUS DERIVATIVE (PD) CONTROLLER

Figure 4.16  Proportional derivative controller response to a load disturbance

Hence

\[ MV = K_c \left( e - T_d \frac{dCV}{dt} \right) + b \]  

(4.22)

In other words, there is no derivative action on a set-point change, only proportional action. On a load upset, both proportional and derivative actions are enabled. (Note also that, in some controller implementations, the proportional action is also decoupled from set-point changes, as the kick from a set-point change is also considered to be too aggressive.)

Figure 4.16 shows a comparison of the control-loop response to a load upset for both P-only and PD control. The response of the measurement PV under PD control is faster and results in a smaller offset than the loop under P-only control. This faster response is due to the addition of the derivative action.

In a PI controller, in order to minimize the integral action, \( T_i \) was made a large number. This makes the integral gain approach zero, and the controller then behaves essentially like a P-only controller. However, in the PD controller, even by setting \( T_d \) to a very small value, there is still the possibility of a sizeable derivative contribution if there is a noisy input, i.e. if \( \frac{dCV}{dt} \) is large.

In electronic controllers and DCSs the derivative action can be eliminated by setting \( T_d \) to zero. In a pneumatic controller the derivative action cannot be eliminated, but it can be reduced to a minimum value of approximately 0.01 min. If a PD controller is installed on a flow loop there will still be considerable derivative action due to the noisy flow measurement. Therefore, it is important when applying a pneumatic controller to a noisy loop, such as a flow loop, to make certain the controller does not contain a derivative block.

The main reason for interest in derivative action is to combine it with proportional and integral action to produce a three-mode controller, a PID.
4.7 Proportional integral derivative (PID) control

The primary purpose of a proportional integral derivative controller (see Equation 4.23) is to provide a response period $\tau_n$ that is much the same as with proportional control but which has no offset. The derivative action adds the additional response speed required to overcome the lag in the response from the integral action.

$$MV = K_c \left( e + \frac{1}{T_i} \int e \, dt - T_d \frac{CV}{dt} \right)$$  \hspace{1cm} (4.23)

Figure 4.17 presents a comparison of the responses for P-only, PI, and PID controllers to a step change in load.

The addition of the derivative mode in the PID controller provides a response similar to that of a P-only controller, but without the offset because of the integral action. Therefore, a PID controller provides a tight dynamic response, but since it contains a derivative block it cannot be used in any processes in which noise is anticipated.

4.7.1 Digital electronic controller forms

Controller algorithms are implemented in digital electronics using digital or ‘discrete-time’ forms of the analog or ‘continuous-time’ controller algorithms presented above. There are two basic digital electronic controller algorithms: the positional form and the velocity or differential form.

![Figure 4.17 P-only, PI, and PID controller responses to a load disturbance](image-url)
The positional form of the PID controller algorithm is

\[
MV(t) = K_c \left[ e(t) - e(t-1) + \frac{1}{T_i} \sum_{k=0}^{n} e(kh) - T_d \frac{CV(t) - CV(t-1)}{h} \right]
\]  

(4.24)

where MV(t), e(t) and CV(t) are the current controller output, error and controlled variables respectively, \( t \) is the enumerated sampling instant in time, MV(t – 1), e(t – 1) and CV(t – 1) are the values of the controller output, error and controlled variables respectively one sampling period ago, and \( h \) is the sampling period.

The velocity or differential form of the PID controller algorithm is

\[
MV(t) = MV(t-1) + K_c \left[ e(t) - e(t-1) + \frac{1}{T_i} e(t) - T_d \frac{CV(t) - 2CV(t-1) + CV(t-2)}{h} \right]
\]  

(4.25)

For the positional form it is important to note how to handle the summation term associated with the integral action properly. The integral term in Equation 4.24 could grow to become a very large value if the output device was saturated and the CV was not able to return to the set point. For situations such as this, it is important to reset the value of the summation to ensure that the output of the algorithm will be equal to the (upper or lower) limit of the controller output. Then, when the set point is changed to a region where the controller can control effectively, the controller will respond without having to decrease the summation term from a value that has grown way beyond the upper or lower limit of the output. This automatic resetting of the controller integral term is commonly called anti-reset windup.

The velocity or differential form does not suffer from reset windup and is, therefore, the preferred form of controller equation when integral action is required. However, the positional form is preferred when there is no integral term, because this is the fail-safe form (in that in the advent of a failure the controller output will fail fully open or closed depending on the design) – whereas the failure mode for the velocity or differential form is the last value of the output.

### 4.8 Choosing the correct controller

Now that the various basic control modes have been described, it is desirable to be able to choose a particular control mode for a specific process. Figure 4.18 graphically outlines a procedure for control mode selection.

Starting at the top of the flow diagram, the first decision block asks the question: ‘Can offset be tolerated?’ If the answer is yes, then a proportional-only controller can be used. If the answer is no, then proceed to the next block, which asks: ‘Is there noise present?’ If there is noise, then use a PI controller. If there is no noise, then proceed
Figure 4.18  Flow chart for controller selection

to the next block, which asks: ‘Is dead time excessive?’ If the ratio of the dead time to the process time constant is greater than 0.5, then the process can be assumed to be dead-time dominant and requires a PI controller. If the process has no excessive dead time, then the next block asks: ‘Is the capacitance extremely small?’ If the answer is yes, then a PI controller can be used. A process with a short dead time and small capacitance does not require derivative action to speed up the response since it is already fast enough, as is the case for a flow loop. In this instance we might even consider an I-only controller, since the loop is so fast that slowing down the response through the use of integral-only action will still provide a fast enough response for the majority
of applications in the fluid processing industries. Finally, if the process capacitance is large, then a PID controller can be used.

It was mentioned earlier that the PI controller is the most common controller found in the plant. Looking at this flow chart one can see why. There are three possible paths to the PI controller, whereas there are four decision blocks that must be passed through to reach a PID controller.

### 4.9 Controller hardware

Now that we have covered how the controller works it is necessary to discuss controller hardware. Figures 4.19 and 4.20 are examples of single-loop stand alone controllers. Figure 4.19 is an electronic analog controller from the 1970s.

Figure 4.20 is a more recent digital version of the electronic controller from the late 1980s. It contains additional functionality, such as alarm limits for process value, deviation output signal and set point, set point ramping, auto- or self-tuning, signal filter time constant adjustment, start up values, on/off modality, and gain schedule limits.

Figure 4.21 shows a screenshot from a modern DCS. Simply put, a DCS is an electronic digital control system where computers spread functionality over multiple processes in large-scale plants. The advantage of a DCS is that it allows operators to monitor and control entire plants from a central control room.

DCSs were introduced in the mid 1970s with the advent of the microcomputer. DCSs enabled more flexible and complex control, monitoring, alarming, and historic data trending than local, single-loop control, or the centralized control previously possible with mini-computers.

Modern DCSs feature the use of digital, multi-drop communications that can interconnect sensors, actuators and the control room. Control can be allocated to digital
devices that can communicate directly with each other, fully exploiting each other’s capabilities with remote diagnostics and supervisory control and data acquisition (SCADA). This control technology is known as Fieldbus [3,4].

The PC explosion of the 1980s and 1990s has also impacted modern DCSs with the advent of PC-based control systems [5,6] which feature object linking and embedding (OLE) software for process control (OPC) [7].

4.10 References

There is no absolute right way or, for that matter, absolute wrong way to tune a controller. Controller settings depend on what the engineer/operator deems to be good performance in terms of the desired response to process upsets. The type of process, the process gain, the time constant, and dead time all play a role in determining the controller settings.

The settings also depend on the anticipated types of disturbance that the process will encounter. A controller would be tuned differently for stability due to set-point changes (servo control) than for load disturbances (regulatory control). In process control systems, load disturbances are most frequently encountered; hence, most systems are optimised for regulatory control.

This chapter will discuss control quality and optimisation, including the performance criteria that need to be considered when tuning a controller. A number of methods that can be used to determine controller settings in order to achieve the desired control are also described.

5.1 Quality of control and optimisation

Controller tuning can be defined as an optimisation process that involves a performance criterion related to the form of controller response and to the error between the process variable and the set point. When tuning a controller, some of the questions that may be asked include:

- Can offset be tolerated?
- Is no overshoot desired?
- Is a certain decay ratio required?
- Is a fast rise time needed?

These questions address some of the performance criteria that are used in the tuning of a controller, including overshoot, decay ratio, and error performance.
5.1.1 Controller response

Depending on the process to be controlled, the first consideration is to decide what type of response is optimal, or at least acceptable. Typical process responses to a load change are illustrated in Figure 5.1.

The three possible general extremes of response that exist, as shown in Figure 5.1, are:

1. overdamped: slow response with no oscillation;
2. critically damped: fastest response without oscillation;
3. underdamped: fast return to set point but with considerable oscillation.

From these three general extremes, we can see that selection of good control is a trade off between the speed of response and deviation from the set point. A highly tuned controller may become unstable if large disturbances occur, whereas a sluggishly tuned controller provides poor performance but is very robust. What is typically required for most process control loops is a compromise between performance and robustness.

When examining the response, there are several common performance criteria that can be used for controller tuning, which are based on characteristics of the system’s closed-loop response. Some of the more common criteria include overshoot, offset, rise time and decay ratio. Of these simple performance criteria, control practitioners most often use decay ratio.

**Cyclic radian frequency**

The cyclic radian frequency $\omega$ is defined as

$$\omega = 2\pi f$$ (5.1)

and

$$f = \frac{1}{\text{period}}$$ (5.2)
5.1 QUALITY OF CONTROL AND OPTIMISATION

If Equation 5.2 is substituted into Equation 5.1, we obtain

$$\omega = \frac{2\pi}{\text{period}}$$  \tag{5.3}$$

The cyclic radian frequency can also be related to the undamped natural frequency \(\omega_n\) and the damping coefficient \(\xi\).

$$\omega = \omega_n \sqrt{1 - \xi^2}$$  \tag{5.4}$$

**Overshoot**

Overshoot is the amount the response exceeds the steady-state final value. Referring to Figure 5.2, the overshoot can be defined as

$$\frac{B}{A} = e^{-\pi \xi / \sqrt{1 - \xi^2}}$$  \tag{5.5}$$

**Decay ratio**

The decay ratio is the ratio of the amplitude of an oscillation to the amplitude of the proceeding oscillation, \(C/B\) in Figure 5.2. More specifically, we can define the quarter decay ratio (QDR), which lies between critical damping and underdamping:

$$\text{QDR} = \frac{C}{B} = \frac{1}{4}$$  \tag{5.6}$$
The decay ratio is often used to establish whether the controller as tuned is providing a satisfactory response. The QDR has been shown through experience to provide a good trade off between minimum deviation from the set point after an upset and the fastest return to the set point. The penalty of QDR is that some oscillation does occur. For a second-order system it can be shown that

\[
\frac{C}{B} = e^{-2\pi \xi / \sqrt{1-\xi^2}}
\]  

(5.7)

**Rise time**

The rise time is the initial time required by the transient response to reach the final steady-state value.

**Response time**

The response time is the time required for the response to settle within the specified arbitrary limits. These limits are typically set at ±3–5 per cent of the PV steady-state value.

### 5.1.2 Error performance criteria

The previously discussed simple performance criteria, i.e. decay ratio, overshoot, etc., use only a few points in the response and therefore are simple to use. On the other hand, error performance criteria are based on the entire response of the process but they are also more complicated.

**Integrated error**

The curve shown in Figure 5.3 represents the response of a loop due to a process upset. This graphical representation of the controlled variable’s return to the set point is known as **Figure 5.3** General response curve
5.1 QUALITY OF CONTROL AND OPTIMISATION

Figure 5.4 Sinusoidal error response

as a response curve. The integrated error (IE) is the area under the response curve, and the idea of using this as an error criterion is to attempt to minimise this area.

In mathematical terms, with \( e \) representing the error as a function of time, we can write

\[
IE = \int_0^\infty e \, dt
\]  
(5.8)

It may not always be possible to minimise IE without paying a penalty in some respect. For example, underdamping produces a minimum area under the response curve but has considerable oscillation.

The method of integrated error may not be 100 per cent reliable if there is no averaging elsewhere in the process. For example, if there is a sinusoidal oscillation about the set point, then the positive and negative areas tend to cancel each other out over time, which presents a misleading conclusion, as shown in Figure 5.4. However, barring this situation, integrated error is a perfectly adequate error criterion.

**Integrated absolute error**

Integrated absolute error (IAE) essentially takes the absolute value of the error. Negative areas are accounted for when IAE is used, thus dismissing the problem encountered with IE regarding sinusoidal responses.

\[
IAE = \int_0^\infty |e| \, dt
\]  
(5.9)

**Integrated squared error**

The integrated squared error (ISE) criterion uses the square of the error, thereby penalizing larger errors more than smaller errors. This gives a more conservative response, i.e. faster return to the set point.

\[
ISE = \int_0^\infty e^2 \, dt
\]  
(5.10)
Tuning Feedback Controllers

Integrated time absolute error

The integrated time absolute error (ITAE) criterion is the integral of the absolute value of the error multiplied by time. ITAE results in errors existing over time being penalised even though they may be small, which results in a more heavily damped response.

\[
\text{ITAE} = \int_0^\infty t |e| \, dt
\]  

Figure 5.5 shows the various responses of a loop that is tuned to the above criteria, including those for IAE and ISE.

5.2 Tuning methods

The following presents a very brief description of some of the various accepted methods used for controller tuning. In each case the suggested controller settings are optimised for a particular error performance criterion, often QDR. The first method described is based entirely on trial and error, whereas the rest are based upon some understanding of the physical nature of the process to be controlled. This understanding is generated from either open- or closed-loop process testing.

5.2.1 ‘Trial-and-error’ method

As the name suggests, tuning by trial and error is simply a guess-and-check type method. The following is a list of practical suggestions for tuning a controller by trial and error. These suggestions are also useful for fine tuning controllers tuned by other methods.

1. Proportional action is the main control. Integral and derivative actions are used to trim the proportional response.

2. The starting point for trial-and-error tuning is always with the controller gain, integral action, and derivative action all at a minimum.
3 Make adjustments in the controller gain by using a factor of two, i.e. 0.25, 0.5, 1.0, 2.0, 4.0, etc.

4 The optimal response is the QDR.

5 When in trouble, decrease the integral and derivative actions to a minimum and adjust the controller gain for stability.

**Rules of thumb**

The following rules of thumb should not be taken as gospel or as a methodology; rather, they are intended to indicate typical values encountered. As such, these rules can be useful when tuning a controller using the trial-and-error method. However, it is important to remember that controller parameters are strongly dependent upon the individual process and may not always abide by the rules outlined below.

**Flow** When dealing with a flow loop, P-only control can be used with a low controller gain. For accuracy, PI control is used with a low controller gain and high integral action. Derivative action is not used, because flow loops typically have very fast dynamics and flow measurement is inherently noisy.

\[ K_c = 0.4 - 0.65 \]
\[ T_i = 0.1 \text{min (or } 6 \text{s)} \]

**Level** Levels represent material inventory that can be used as surge capacity to dampen disturbances. Hence, loosely tuned P-only control is sometimes used. However, most operators do not like offset, so PI level controllers are typically used.

The following P-only settings ensure that the control valve will be 50 per cent open when the level is at 50 per cent, wide open when the level is at 75 per cent, and shut when the level is at 25 per cent:

\[ K_c = 2 \]
\[ \text{Bias term } b = 50\% \]
\[ \text{Set point SP} = 50\% \]

Typical PI controller settings are:

\[ K_c = 2 - 20 \]
\[ T_i = 1 - 5 \text{ min} \]
**Pressure**  Pressure control loops show large variation in tuning, depending on the dynamics of the pressure response. Typical ranges are as follows:

- **Vapour**  
  \[ K_c = 2–10 \]  
  \[ T_i = 2–10 \text{ min} \]
- **Liquid**  
  \[ K_c = 0.5–2 \]  
  \[ T_i = 0.1–0.25 \text{ min} \]

**Temperature**  Temperature dynamic responses are usually fairly slow, so PID control is used. Typical parameter values are:

\[
K_c = 2–10 \\
T_i = 2–10 \text{ min} \\
T_d = 0–5 \text{ min}
\]

### 5.2.2 Process reaction curve methods

In the process reaction curve methods a process reaction curve is generated in response to a disturbance. This process curve is then used to calculate the controller gain, integral time and derivative time. These methods are performed in open loop, so no control action occurs and the process response can be isolated.

To generate a process reaction curve, the process is allowed to reach steady state or as close to steady state as possible. Then, in open loop, so that there is no control action, a small disturbance is introduced and the reaction of the process variable is recorded. Figure 5.6 shows a typical process reaction curve generated using the above method for a generic self-regulating process. The term self-regulating refers to a process where the controlled variable eventually returns to a stable value or levels out without external intervention. The process parameters that may be obtained from this process reaction curve for the initial step disturbance \( P(\%) \) are as follows: the lag time \( L \) (min), the time constant \( T \) (min), the change in PV in response to step disturbance \( \Delta C_p(\%) \), i.e. \((\text{change in PV})/(\text{PV span}) \times 100\), the reaction rate \( N(\% \text{ min}^{-1}) \)

\[
N = \frac{\Delta C_p}{T}
\]

and the lag ratio \( R \) (dimensionless)

\[
R = \frac{L}{T} = \frac{NL}{\Delta C_p}
\]

![Process reaction curve](image.png)

**Figure 5.6**  Process reaction curve
Methods of process analysis with forcing functions other than a step input are possible, and include pulses, ramps, and sinusoids. However, step function analysis is the most common, as it is the easiest to implement.

**Ziegler—Nichols open-loop rules**

In 1942, Ziegler and Nichols [1] changed controller tuning from an art to a science by developing their open-loop step function analysis technique. They also developed a closed-loop technique, which is described in the next section on constant cycling methods.

The Ziegler–Nichols open loop recommended controller settings for the QDR are as follows:

\[
P\text{-only} \quad K_c = \frac{P}{NL} \quad (5.12)
\]

\[
\text{PI} \quad K_c = 0.9 \frac{P}{NL} \quad (5.13)
\]

\[
T_i = 3.33L \quad (5.14)
\]

\[
P\text{ID} \quad K_c = 1.2 \frac{P}{NL} \quad (5.15)
\]

\[
T_i = 2.0L \quad (5.16)
\]

\[
T_d = 0.5L \quad (5.17)
\]

These settings should be taken as recommendations only and tested thoroughly in closed loop, with fine tuning of the parameters to obtain the QDR.

**Cohen—Coon open-loop rules**

In 1953, Cohen and Coon [2] developed a set of controller tuning recommendations that correct for one deficiency in the Ziegler–Nichols open-loop rules. This deficiency is the sluggish closed-loop response given by the Ziegler–Nichols rules on the relatively rare occasion when process dead time is large relative to the dominant open-loop time constant.

The Cohen–Coon recommended controller settings are as follows:

\[
P\text{-only} \quad K_c = \frac{P}{NL} \left( 1 + \frac{R}{3} \right) \quad (5.18)
\]

\[
\text{PI} \quad K_c = \frac{P}{NL} \left( 0.9 + \frac{R}{12} \right) \quad (5.19)
\]
\[
T_i = L \frac{30 + 3R}{9 + 20R} \quad (5.20)
\]

\[
\text{PID } K_c = \frac{P}{NL} \left( 1.33 + \frac{R}{4} \right) \quad (5.21)
\]

\[
T_i = L \frac{32 + 6R}{13 + 8R} \quad (5.22)
\]

\[
T_d = L \frac{4}{11 + 2R} \quad (5.23)
\]

As with the Ziegler–Nichols open-loop method recommendations, the Cohen–Coon values should be implemented and tested in closed loop and adjusted accordingly to achieve the QDR.

**Internal model control tuning rules**

Many practitioners have found that the Ziegler–Nichols open loop and Cohen–Coon rules are too aggressive for most chemical industry applications, since they give a large controller gain and short integral time. Rivera *et al.* [3] developed the internal model control (IMC) tuning rules with robustness in mind. The tuning parameter from the IMC method (the closed-loop speed of response) relates directly to the closed-loop time constant and the robustness of the control loop. As a consequence, the closed-loop step load response exhibits no oscillation or overshoot. Lambda tuning, e.g. [4], is a term that is also used to refer to controller tuning methods that are based on a specified closed-loop time constant.

Since the general IMC method is unnecessarily complicated for processes that are well approximated by first-order dead time or integrator dead-time models, simplified IMC rules were developed by Fruehauf *et al.* [5] for PID controller tuning (see Table 5.1).

Of course, these recommendations need to be tested in the closed-loop situation and the final settings arrived at through the use of fine tuning.

**Table 5.1** Simplified IMC rules [4]

<table>
<thead>
<tr>
<th>( \frac{\tau}{L} )</th>
<th>( \frac{\tau}{L} &lt; 3 )</th>
<th>( L &lt; 0.5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_i )</td>
<td>( K_c )</td>
<td>( T_d )</td>
</tr>
<tr>
<td>( \leq 0.5L )</td>
<td>( 5L )</td>
<td>( \leq 0.5L )</td>
</tr>
<tr>
<td>( \leq 0.5L )</td>
<td>( \tau )</td>
<td>( \leq 0.5L )</td>
</tr>
<tr>
<td>( \leq 0.5L )</td>
<td>( 4 )</td>
<td></td>
</tr>
<tr>
<td>( \leq 0.5L )</td>
<td>( \frac{P}{NL} )</td>
<td></td>
</tr>
<tr>
<td>( \leq 0.5L )</td>
<td>( \frac{P}{N} )</td>
<td></td>
</tr>
<tr>
<td>( \leq 0.5L )</td>
<td>( \frac{P}{NL} )</td>
<td></td>
</tr>
</tbody>
</table>
5.2.3 Constant cycling methods

Ziegler–Nichols closed-loop method

The closed-loop technique of Ziegler and Nichols [6] is a technique that is commonly used to determine the two important system constants: ultimate period and ultimate gain. It was one of the first tuning techniques to be widely adopted.

When tuning using the Ziegler–Nichols closed-loop method, values for proportional, integral, and derivative controller parameters may be determined from the ultimate period and ultimate gain. These are determined by disturbing the closed-loop system and using the disturbance response to extract the values of these constants.

The following is a step-by-step approach to using the Ziegler–Nichols closed-loop method for controller tuning:

1. Attach a proportional-only controller with a low gain (no integral or derivative action).
2. Place the controller in automatic.
3. Increase the controller gain until a constant-amplitude limit cycle occurs.
4. Determine the ultimate period $P_u$ and ultimate gain $K_u$ from the constant-amplitude limit cycle (Figure 5.7):

$$
P_u = \text{period taken from limit cycle}
$$
$$
K_u = \text{controller gain that produces the limit cycle}
$$

5. Calculate the tuning parameters using the following equations:

$$
P\text{-only } K_c = \frac{K_u}{2} \quad (5.24)
$$

![Figure 5.7 Constant-amplitude limit cycle](image)
PI $K_c = \frac{K_u}{2.2}$ (5.25)

$T_i = \frac{P_u}{1.2}$ (5.26)

PID $K_c = \frac{K_u}{1.7}$ (5.27)

$T_i = \frac{P_u}{2}$ (5.28)

$T_d = \frac{P_u}{8}$ (5.29)

6 Fine tune by adjusting $K_c$, $T_i$, and $T_d$ as required to find the QDR.

**Auto-tune variation technique**

The auto-tune variation (ATV) technique of Åström and Hagglund [6] is another closed-loop technique used to determine the two important system constants, i.e. the ultimate period and the ultimate gain. However, the ATV technique determines these system constants without unduly upsetting the process. Tuning values for proportional, integral and derivative controller parameters can be determined from these two constants. Here, we recommend the use of Tyreus–Luyben [7] settings for tuning that is suitable for chemical process unit operations. All methods for determining the ultimate period and ultimate gain involve disturbing the system and using the disturbance response to extract the values of these constants.

In the case of the ATV technique, a small limit-cycle disturbance is set up between the manipulated variable (controller output) and the controlled variable (process variable). Figure 5.8 shows the instrument set-up, and Figure 5.9 shows the typical ATV response plot with critical parameters defined. It is important to note that the ATV technique is
applicable only to processes with significant dead time. The ultimate period will just equal the sampling period if the dead time is not significant. The general ATV tuning method for a PI controller is as follows:

1. Determine a reasonable value for valve change $h$ (typically 0.05, i.e., 5 percent). The value for $h$ should be small enough that the process is not unnecessarily upset, but large enough that the amplitude $a$ can be measured.

2. Move the valve $+h$ units.

3. Wait until the process variable starts to move, then move the valve $-2h$ units.

4. When the process variable (PV) crosses the set point, move the valve $+2h$ units.

5. Repeat until a limit cycle is established, as illustrated in Figure 5.9.

6. Record the value of $a$ by picking it off the response graph.

7. Perform the following calculations to determine the ultimate period $P_u$, ultimate gain $K_u$, and the controller gain $K_c$ and integral time $T_i$:

\[
P_u = \text{period taken from limit cycle}
\]
\[
K_u = \frac{4h}{3.14a}
\]
\[
K_c = \frac{K_u}{3.2}
\]
\[
T_i = 2.2P_u
\]

![Figure 5.9 ATV critical parameters](image-url)
Table 5.2  Tuning comparison

<table>
<thead>
<tr>
<th></th>
<th>Z–N CL</th>
<th>ATV</th>
<th>(Z–N/ATV) ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller gain $K_c$</td>
<td>$K_u$</td>
<td>$K_u$</td>
<td>1.45</td>
</tr>
<tr>
<td>Integral time $T_i$</td>
<td>$P_u$</td>
<td>$2.2 P_u$</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Comparison of ATV and Ziegler–Nichols closed-loop tuning techniques

Table 5.2 compares the tuning constants between the ATV and Ziegler–Nichols closed-loop (Z–N CL) tuning techniques. Notice that the Ziegler–Nichols tuning is more aggressive, with a larger controller gain and shorter integral time. This technique was originally developed for electromechanical systems control and is based on the more aggressive QDR criterion. ATV tuning was developed for fluid and thermal processes and emphasises minimizing overshoot. ATV, therefore, is often the preferred technique for process control.

5.3 References