13 Control Systems

J O Flower Bsc(Eng), PhD, DSc(Eng), CEng, FIEE, FIMarE, MSNAME
University of Warwick

E A Parr MSc, CEng, MIEE, MInstMC
CoSteel Sheerness

Contents

13.1 Introduction 13/3

13.2 Laplace transforms and the transfer function 13/3
13.2.1 Laplace transformation 13/3
13.2.2 The transfer function 13/3
13.2.3 Certain theorems 13/5

13.3 Block diagrams 13/6

13.4 Feedback 13/6

13.5 Generally desirable and acceptable behaviour 13/7

13.6 Stability 13/8

13.7 Classification of system and static accuracy 13/9
13.7.1 Classification 13/9
13.7.2 Static accuracy 13/9
13.7.3 Steady-state errors due to disturbances 13/10

13.8 Transient behaviour 13/10
13.8.1 First-order system 13/10
13.8.2 Second-order system 13/11
13.8.3 Velocity feedback 13/12
13.8.4 Incorporation of integral control 13/12

13.9 Root-locus method 13/13
13.9.1 Rules for construction of the root locus 13/14

13.10 Frequency-response methods 13/15
13.10.1 Nyquist plot 13/16
13.10.2 Bode diagram 13/17
13.10.3 Nichols chart 13/20

13.11 State-space description 13/20

13.12 Sampled-data systems 13/24

13.13 Some necessary mathematical preliminaries 13/25
13.13.1 The *z* transformation 13/25

13.14 Sampler and zero-order hold 13/25

13.15 Block diagrams 13/26

13.16 Closed-loop systems 13/27

13.17 Stability 13/28

13.18 Example 13/28

13.19 Dead-beat response 13/30

13.20 Simulation 13/30
13.20.1 System models 13/30
13.20.2 Integration schemes 13/32
13.20.3 Organisation of problem input 13/32
13.20.4 Illustrative example 13/32

13.21 Multivariable control 13/33

13.22 Dealing with non-linear elements 13/35
13.22.1 Introduction 13/35
13.22.2 The describing function 13/35
13.22.3 State space and the phase plane 13/38

13.23 Disturbances 13/42
13.23.1 Introduction 13/42
13.23.2 Cascade control 13/43
13.23.3 Feedforward 13/44

13.24 Ratio control 13/45
13.24.1 Introduction 13/45
13.24.2 Slave follow master 13/46
13.24.3 Lead lag control 13/47

13.25 Transit delays 13/47
13.25.1 Introduction 13/47
13.25.2 The Smith predictor 13/48

13.26 Stability 13/48
13.26.1 Introduction 13/48
13.26.2 Definitions and performance criteria 13/48
13.26.3 Methods of stability analysis 13/51

13.27 Industrial controllers 13/52
13.27.1 Introduction 13/52
13.27.2 A commercial controller 13/52
13.27.3 Bumpless transfer 13/54
13.27.4 Integral windup and desaturation 13/54
13.27.5 Selectable derivative action 13/55
13.27.6 Variations on the PID algorithm 13/55
13.27.7 Incremental controllers 13/56
13.27.8 Scheduling controllers 13/56
13.27.9 Variable gain controllers 13/56
13.27.10 Inverse plant model 13/56

13.28 Digital control algorithms 13/57
13.28.1 Introduction 13/57
13.28.2 Shannon's sampling theorem 13/58
13.28.3 Control algorithms 13/59

13.29 Auto-tuners 13/59

13.30 Practical tuning methods 13/60
13.30.1 Introduction 13/60
13.30.2 Ultimate cycle methods 13/61
13.30.3 Bang/bang oscillation test 13/61
13.30.4 Reaction curve test 13/61
13.30.5 A model building tuning method 13/62
13.30.6 General comments 13/63

13.1 Introduction

Examples of the conscious application of feedback control ideas have appeared in technology since very early times: certainly the float-regulator schemes of ancient Greece were notable examples of such ideas. Much later came the automatic direction-setting of windmills, the Watt governor, its derivatives, and so forth. The first third of the 1900s witnessed applications in areas such as automatic ship steering and process control in the chemical industry. Some of these later applications attracted considerable analytical effort aimed at attempting to account for the seemingly capricous dynamic behaviour that was sometimes found in practice.

However, it was not until during, and immediately after, World War II that the fundamentals of the above somewhat disjointed control studies were subsumed into a coherent body of knowledge which became recognised as a new engineering discipline. The great thrust in achieving this had its main antecedents in work done in the engineering electronics industry in the 1930s. Great theoretical strides were made and the concept of feedback was, for the first time, recognised as being all pervasive. The practical and theoretical developments emanating from this activity, constitute the classical approach to control which are explored in some detail in this chapter.

Since the late 1940s, tremendous efforts have been made to expand the boundaries of control engineering theory. For example, ideas from classical mechanics and the calculus of variations have been adapted and extended from a control-theoretic viewpoint. This work is based largely on the state-space description of systems (this description is briefly described in Section 13.11). However, it must be admitted that the practical uses and advantages of many of these developments have yet to be demonstrated. Most control system design work is still based on the classical work mentioned previously. Moreover, nowadays these applications rely, very heavily, on the use of computer techniques; indeed, computers are commonly used as elements in control loops.

Techniques from the 'classical' period of control engineering development is easily understood, wide-ranging in application and, perhaps most importantly, capable of coping with deficiencies in detailed knowledge about the system to be controlled.

These techniques are easily adapted for use in the computer-aided design of control systems, and have proved themselves capable of extension into the difficult area of multi-variable system control; however, this latter topic is beyond the scope of this chapter. So with the above comments in mind, a conventional basic approach to control theory is presented, with a short discussion of the state-space approach and a more extensive forage into sampled-data systems. These latter systems have become important owing to the incorporation of digital computers, particularly micro-computers, into the control loop. Fortunately, an elementary theory for sampled data can be established which nicely parallels the development of basic continuous control theory.

The topics covered in this introduction, and extensions of them, have stood practitioners in good stead for several decades now, and can be confidently expected to go on delivering good service for some decades to come.

13.2 Laplace transforms and the transfer function

In most engineering analysis it is usual to produce mathematical models (of varying precision) to predict the behaviour of physical systems. Often such models are manifested by a differential equation description. This appears to fit in with the causal behaviour of idealised components, e.g. Newton's law relating the second derivative of displacement to the applied force. It is possible to model such behaviour in other ways (for example, using integral equations), although these are much less familiar to most engineers. All real systems are non-linear; however, it is fortuitous that most systems behave approximately like linear ones, with the implication that superposition holds true to some extent. We further restrict the coverage here in that we shall be concerned particularly with systems whose component values are not functions of time—at least over the time-scale of interest to us.

In mathematical terms this latter point implies that the resulting differential equations are not only linear, but also have constant coefficients, e.g. many systems behave approximately according to the equation

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+2\zeta\omega_n\frac{\mathrm{d}x}{\mathrm{d}t}+\omega_n^2x=\Leftarrow\omega_n^2f(t) \tag{13.1}$$

where x is the dependent variable (displacement, voltage, etc.), $f(t)$ is a forcing function (force, voltage source, etc.), and ω_n^2 and ζςare constants the values of which depend on the size and interconnections of the individual physical components making up the system (spring-stiffness constant, inductance values, etc.).

Equations having the form of Equation (13.1) are called 'linear constant coefficient ordinary differential equations' (LCCDE) and may, of course, be of any order. There are several techniques available for solving such equations but the one of particular interest here is the method based on the Laplace transformation. This is treated in detail elsewhere, but it is useful to outline the specific properties of particular interest here.

13.2.1 Laplace transformation

Given a function $f(t)$, then its Laplace transformation $F(s)$ is defined as

$$L[f(t)]=\Leftarrow F(s)=\int_0^{\infty\Leftarrow}\Leftarrow f(t)\exp(-st)\mathrm{d}t$$

where, in general, s is a complex variable and of such a magnitude that the above integral converges to a definite functional value.

A list of Laplace transformation pairs is given in *Table 13.1.*

The essential usefulness of the Laplace transformation technique in control engineering studies is that it transforms LCCDE and integral equations into algebraic ones and, hence, makes for easier and standard manipulation.

13.2.2 The transfer function

This is a central notion in control work and is, by definition, the Laplace transformation of the output of a system divided by the Laplace transformation of the input, with the tacit assumption that all initial conditions are at zero.

Thus, in *Figure 13.1*, where $y(t)$ is the output of the system and $u(t)$ is the input, then the transfer function $G(s)$ is

$$L[y(t)]/L[u(t)]=Y(s)/U(s)=\Leftarrow G(s)\Leftarrow$$

Supposing that $y(t)$ and $u(t)$ are related by the general LCCDE

$$a_n\frac{\mathrm{d}^ny}{\mathrm{d}t^n}+a_{n-1}\frac{d^{n-1}y}{\mathrm{d}t^{n-1}}+\cdots\Leftarrow a_0y$$

Table 13.1 Laplace transforms and *z* transforms

$f(t)$	$F(s)$	$F(z)$
0	0	0
$f(t-nT)$	$\exp(-nsT)F(s)$	$z^{-n}F(z)$
$\delta(t)$	1	1
$\delta(t-nT)$	$\exp(-nsT)$	z^{-n}
$\sum_{n=0}^{\infty\Leftarrow}\delta(t-\Leftarrow nT)$	$[1-\exp(-st)]^{-1}$	$z(z-1)^{-1}$
$h(t)$	s^{-1}	$z(z-1)^{-1}$
$u_T(t)$	$[1-\exp(-sT)]s^{-1}$	—
A	As^{-1}	$Az(z-1)^{-1}$
t	s^{-2}	$Tz(z-1)^{-2}$
$f(t)t$	$-\mathrm{d}F(s)/\mathrm{d}s$	—
$(t-nT)h(t-nT)$	$\exp(-nsT)s^{-2}$	$Tz^{-(n-1)}(z-1)^{-2}$
t^2	$2s^{-3}$	$T^2z(z+1)(z-1)^{-3}$
t_n	$n!s^{-(n+1)}$	—
$\exp(\alpha t)$	$(s-\alpha)^{-1}$	$z(z-\exp(\alpha T))^{-1}$
$f(t)\exp(\alpha t)$	$F(s-\alpha)$	$F[z\exp(-\alpha T)]$
$\delta(t)+\alpha\varsigma\exp(\alpha t)$	$s(s-\alpha)^{-1}$	—
$t\ \exp(\alpha t)$	$(s-\alpha)^{-2}$	$TZ\ \exp(\alpha T)[z-\exp(\alpha T)]^{-2}$
$t^n\ \exp(\alpha t)$	$n!(s-\alpha)^{-(n+1)}$	—
$\sin\ \omega t$	$\frac{\omega\varsigma}{s^2+\Leftarrow\omega^2}$	$\frac{z\sin\omega T}{z^2-\Leftarrow 2z\cos\omega T+\Leftarrow 1\Leftarrow}$
$\cos\ \omega t$	$\frac{s}{s^2+\Leftarrow\omega^2}$	$\frac{z(z-\Leftarrow\cos\omega T)}{z^2-\Leftarrow 2z\cos\omega T+\Leftarrow 1\Leftarrow}$
$\frac{t}{2\omega\varsigma}\sin\omega t$	$\frac{s}{(s^2+\Leftarrow\omega^2)}$	—
$\frac{1}{2\omega}(\sin\omega t-\Leftarrow\omega t\cos\omega t)$	$\frac{\omega^2}{(s^2+\Leftarrow\omega^2)^2}$	—
$\frac{1}{\cos\delta\varsigma}\sin(\omega t+\Leftarrow\delta)$	$\frac{A}{s^2+\Leftarrow\omega^2}\left(s+\frac{\omega\Leftarrow}{A}\right)$ where $\tan\delta\varsigma=\Leftarrow A$	—
$\frac{1}{\cos\delta\varsigma}\cos(\omega t+\Leftarrow\delta)$	$\frac{1}{s^2+\Leftarrow\omega^2}(s-\Leftarrow A\omega)$	—
$\exp(\alpha t)\ \sin\ \omega t$	$\frac{\omega\varsigma}{(s-\Leftarrow\alpha)^2+\Leftarrow\omega^2}=\Leftarrow\frac{\omega\varsigma}{(s-\Leftarrow\alpha\varsigma+\Leftarrow j\omega)(s-\Leftarrow\alpha\varsigma-\Leftarrow j\omega)}$	$\frac{z\exp(\alpha T)\sin\omega T}{z^2-\Leftarrow 2z\exp(\alpha T)\cos\omega T+\Leftarrow\exp(2\alpha T)}$
$\exp(\alpha t)\ \cos\ \omega t$	$\frac{s-\Leftarrow\alpha\varsigma}{(s-\Leftarrow\alpha)^2+\Leftarrow\omega^2}$	$\frac{z[z-\Leftarrow\exp(\alpha T)\cos\omega T]\Leftarrow}{z^2-\Leftarrow 2z\exp(\alpha T)\cos\omega T+\Leftarrow\exp(2\alpha T)}$
$\frac{t}{2\omega}\exp(\alpha t)\sin\omega t$	$\frac{s-\Leftarrow\alpha\varsigma}{[(s-\Leftarrow\alpha)^2+\Leftarrow\omega^2]^2}$	—
$\frac{1}{2\omega}\exp(\alpha t)(\sin\omega t-\Leftarrow\cos\omega t)$	$\frac{\omega^2}{[(s-\Leftarrow\alpha)^2+\Leftarrow\omega^2]^2}$	—
$\frac{1}{\cos\delta\varsigma}\exp(\alpha t)\sin(\omega t+\Leftarrow\delta)$	$\frac{A}{(s-\Leftarrow\alpha)^2+\Leftarrow\omega^2}\left(s-\Leftarrow\alpha\varsigma+\frac{\omega\Leftarrow}{A}\right)$ where $\tan\delta\varsigma=\Leftarrow A$	—
$\frac{1}{\cos\delta\varsigma}\exp(\alpha t)\cos(\omega t+\Leftarrow\delta)$	$\frac{1}{(s-\Leftarrow\alpha)^2+\Leftarrow\omega^2}(s-\Leftarrow\alpha\varsigma-\Leftarrow A\omega)$	—
$\sinh\ \omega t$	$\omega(s^2-\omega^2)^{-1}$	—
$\cosh\ \omega t$	$s(s^2-\omega^2)^{-1}$	—

cont'd

Table 13.1 (*continued*)

$f'(t)$	$sF(s) - f(0-)$	—
$f''(t)$	$s^2F(s) - sf(0-) - f'(0-)$	—
$f^n(t)$	$s^nF(s) - s^{n-1}f(0-) - s^{n-2}f'(0-) \ldots - f^{n-1}(0-)$	—
$f^{-1}(t)$	$\frac{F(s)}{s} + \Leftarrow \frac{f^{-1}(0-)}{s}$	—
$f(t)$ $t \to 0$	$sF(s)$ $s \to \infty \Leftarrow$	$F(z)$ $z \to \infty \Leftarrow$
$f(t)$ $t \to \infty \Leftarrow$	$sF(s)$ $s \to 0$	$(z-1)z^{-1}F(z)$ $z \to 1$

$\delta(t)$, The unit impulse function.
$h(t)$, The unit step function.
$u_T(t)$, The unit step function followed by a unit negative step at $t = T$, where T is the sampling period.

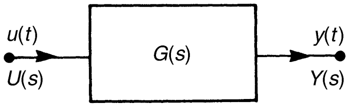


Figure 13.1 Input–output representation

$$= b_m \frac{d^m u}{dt^m} + b_{m-1}\frac{d^{m-1}u}{dt^{m-1}} + \cdots + b_0 u \qquad (13.2)$$

then, on Laplace transforming and ignoring initial conditions, we have (see later for properties of Laplace transformation)

$$(a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0)Y(s) \Leftarrow$$
$$= (b_m s^m + b_{m-1}s^{m-1} + \cdots + b_0)U(s) \Leftarrow$$

whence

$$\frac{Y(s)\Leftarrow}{U(s)\Leftarrow} \equiv G(s) = \sum_{i=0}^{m} b_i s^i \Big/ \sum_{i=0}^{n} a_i s^i$$

There are a number of features to note about $G(s)$.

(1) Invariably $n > m$ for physical systems.
(2) It is a ratio of two polynomials which may be written

$$G(s) = \Leftarrow \frac{b_m(s - z_1)\ldots(s - z_m)\Leftarrow}{a_n(s - p_1)\ldots(s - p_n)\Leftarrow}$$

$z_1, \ldots, z_m$ are called the *zeros* and $p_1, \ldots, p_n$ are called the *poles* of the transfer function.
(3) It is not an explicit function of input or output, but depends entirely upon the nature of the system.
(4) The block diagram representation shown in *Figure 13.1* may be extended so that the interaction of composite systems can be studied (provided that they do not load each other); see below.
(5) If $u(t)$ is a delta function $\delta(t)$, then $U(s) = 1$, whence $Y(s) = G(s)$ and $y(t) = g(t)$, where $g(t)$ is the *impulse response* (or weighting function) of the system.
(6) Although a particular system produces a particular transfer function, a particular transfer function does not imply a particular system, i.e. the transfer function specifies merely the input–output relationship between two variables and, in general, this relationship may be realised in an infinite number of ways.
(7) Although we might expect that all transfer functions will be ratios of finite polynomials, an important and common element which is an exception to this is the pure-delay element. An example of this is a loss-free transmission line in which any disturbance to the input of the line will appear at the output of the line without distortion, a finite time (say τ) later. Thus, if $u(t)$ is the input, then the output $y(t) = u(t - \tau)$ and the transfer function $Y(s)/U(s) = \exp(-s\tau)$. Hence, the occurrence of this term within a transfer function expression implies the presence of a pure delay; such terms are common in chemical plant and other fluid-flow processes.

Having performed any manipulations in the Laplace transformation domain, it is necessary for us to transform back to the time domain if the time behaviour is required. Since we are dealing normally with the ratio of polynomials, then by partial fraction techniques we can arrange $Y(s)$ to be written in the following sequences:

$$Y(s) = \Leftarrow \frac{K(s - z_1)(s - z_2)\ldots(s - z_m)\Leftarrow}{(s - p_1)(s - p_2)\ldots(s - p_n)\Leftarrow}$$

$$Y(s) = K\left[\frac{A_1}{s - p_1} + \Leftarrow \frac{A_2}{s - p_2} + \cdots + \Leftarrow \frac{A_n}{s - p_n}\right]$$

and by so arranging $Y(s)$ in this form the conversion to $y(t)$ can be made by looking up these elemental forms in *Table 13.1*.

Example Suppose that

$$Y(s) = \frac{5(s^2 + 4s + 3)\Leftarrow}{s^3 + 6s^2 + 8s} = \Leftarrow \frac{5(s^2 + 4s + 3)\Leftarrow}{s(s + 2)(s + 4)\Leftarrow}$$
$$= 5\left[\frac{3}{8s} + \frac{1}{4(s + 2)\Leftarrow} + \Leftarrow \frac{3}{8(s + 4)}\Leftarrow\right]$$

Then

$$y(t) = \Leftarrow \frac{5}{4}\left[\frac{3}{2}\{1 + \exp(-4t)\} + \exp(-2t)\Leftarrow\right]$$

13.2.3 Certain theorems

A number of useful transform theorems are quoted below, without proof.

(1) *Differentiation*
If $F(s)$ is the Laplace transformation of $f(t)$, then

$$L[d^n f(t)/dt^n] = s^n F(s) - s^{n-1}f(0) = s^{n-2}f'(0) - \cdots - f^{n-1}(0)\Leftarrow$$

For example, if $f(t) = \exp(-bt)$, then

$$L\left[\frac{d^3}{\mathrm{d}t^3}\exp(-bt)\right]\Bigg(= \Leftarrow \frac{s^3}{s+b} - s^2 + bs - b^2$$

(2) *Integration*
If $L[f(t)] = F(s)$, then

$$L\left[\int_0^1 f(t)\mathrm{d}t\right]\Bigg(= \frac{F(s)\Leftarrow}{s} + f(0)\Leftarrow$$

Repeated integration follows in a similar fashion.

(3) *Final-value theorem*
If $f(t)$ and $f'(t)$ are Laplace transformable and if $L[f(t)] = F(s)$, then if the limit of $f(t)$ exists as t goes towards infinity, then

$$\lim_{s\to 0} sF(s) = \lim_{t\to\infty\Leftarrow} f(t)\Leftarrow$$

For example,

$$F(s) = \Leftarrow \frac{b-a}{s(s+a)(s+b)\Leftarrow}$$

then

$$\lim_{s\to 0}\frac{s(b-a)\Leftarrow}{s(s+a)(s+b)\Leftarrow} = \frac{b-a}{ab} = \Leftarrow \lim_{t\to\infty\Leftarrow} f(t)\Leftarrow$$

(4) *Initial-value theorem*
If $f(t)$ and $f'(t)$ are Laplace transformable and if $L[f(t)] = F(s)$,
then

$$\lim_{s\to\infty\Leftarrow} sF(s) = \lim_{t\to 0} f(t)\Leftarrow$$

(5) *Convolution*
If $L[f_1(t)] = F_1(s)$ and $L[f_2(t)] = F_2(s)$, then

$$F_1(s)\cdot F_2(s) = L\left[\int_0^{\infty\Leftarrow} f_1(t-\tau)\cdot f_2(\tau)\mathrm{d}\tau\right]\Bigg($$

13.3 Block diagrams

It is conventional to represent individual transfer functions by boxes with an input and output (see note (4) in Section 13.2.2). Provided that the components represented by the transfer function do not load those represented by the transfer function in a connecting box, then simple manipulation of the transfer functions can be carried out. For example, suppose that there are two transfer functions in cascade (see *Figure 13.2*): then we may write $X(s)/U(s) = G_1(s)$ and $Y(s)/X(s) = G_2(s)$. Eliminating $X(s)$ by multiplication, we have

$$Y(s)/U(s) = G_1(s)G_2(s)\Leftarrow$$

which may be represented by a single block. This can obviously be generalised to any number of blocks in cascade.

Another important example of block representation is the prototype feedback arrangement shown in *Figure 13.3*. We see that $Y(s) = G(s)E(s)$ and $E(s) = U(s) - H(s)Y(s)$. Eliminating $E(s)$ from these two equations results in

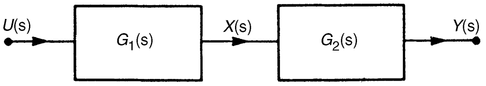


Figure 13.2 Systems in cascade

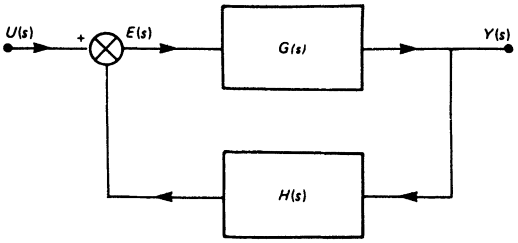


Figure 13.3 Block diagram of a prototype feedback system

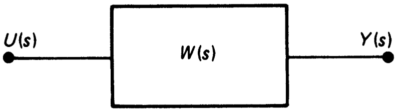


Figure 13.4 Reduction of the diagram shown in *Figure 13.3* to a single block

$$\frac{Y(s)\Leftarrow}{U(s)\Leftarrow} = \Leftarrow \frac{G(s)\Leftarrow}{1 + H(s)G(s)\Leftarrow} = W(s)\Leftarrow$$

In block diagram form we have *Figure 13.4*. If we eliminate $Y(s)$ from the above equations, we obtain

$$\frac{E(s)\Leftarrow}{U(s)\Leftarrow} = \Leftarrow \frac{1}{1 + (H(s)G(s)\Leftarrow}$$

13.4 Feedback

The last example is the basic feedback conceptual arrangement, and it is pertinent to investigate it further, as much effort in dealing with control systems is devoted to designing such feedback loops. The term 'feedback' is used to describe situations in which a portion of the output (and/or processed parts of it) are fed back to the input of the system. The appropriate application may be used, for example, to improve bandwidth, improve stability, improve accuracy, reduce effects of unwanted disturbances, compensate for uncertainty and reduce the sensitivity of the system to component value variation.

As a concrete example consider the system shown in *Figure 13.5*, which displays the arrangements for an angular position control system in which a desired position θ_r is indicated by tapping a voltage on a potentiometer. The actual position of the load being driven by the motor (usually via a gearbox) is monitored by θ_o, indicated, again electrically, by a potentiometer tapping. If we assume identical potentiometers energised from the same voltage supply, then the misalignment between the desired output and the actual output is indicated by the difference between the respective potentiometer voltages. This difference (proportional to error) is fed to an amplifier whose output, in turn,

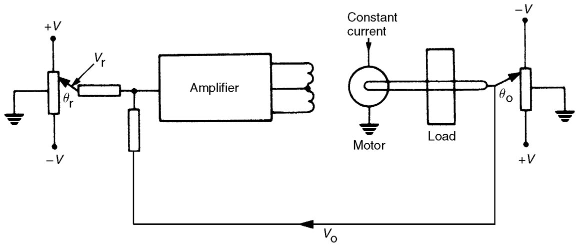


Figure 13.5 Schematic diagram of a simple position and control system

drives the motor. Thus, the arrangement seeks to drive the system until the output θ_o and input θ_r are coincident (i.e. the error is zero).

In the more general block diagram form, the above schematic will be transformed to that shown in *Figure 13.6*, where $\theta_r(s)$, $\theta_o(s)$ are the Laplace transforms of the input, output position: $K_1(s)$ and $K_2(s)$ are the potentiometer transfer functions (normally taken as straight gains); $Vr(s)$ is the Laplace transform of the reference voltage; $Vo(s)$ is the Laplace transform of the output voltage; $G_m(s)$ is the motor transfer function; $G_l(s)$ is the load transfer function; and $A(s)$ is the amplifier transfer function.

Let us refer now to *Figure 13.3* in which $U(s)$ is identified as the transformed input (reference or demand) signal, $Y(s)$ is the output signal and $E(s)$ is the error (or actuating) signal. $G(s)$ represents the *forward transfer function* and is the product of all the transfer functions in the forward loop, i.e. $G(s)=\Leftarrow A(s)G_m(s)G_1(s)$ in the above example.

$H(s)$ represents the *feedback transfer function* and is the product of all transfer functions in the feedback part of the loop.

We saw in Section 13.3 that we may write

$$\frac{Y(s)\Leftarrow}{U(s)\Leftarrow}=\Leftarrow\frac{G(s)\Leftarrow}{1+\Leftarrow H(s)G(s)\Leftarrow}$$

and

$$\frac{E(s)\Leftarrow}{U(s)\Leftarrow}=\Leftarrow\frac{1}{1+\Leftarrow H(s)G(s)\Leftarrow}$$

i.e. we have related output to input and the error to the input.

The product $H(s)G(s)$ is called the *open-loop transfer function* and $G(s)/[1 + H(s)G(s)]$ the *closed-loop transfer function*. The open-loop transfer function is most useful in studying the behaviour of the system, since it relates the error to the demand. Obviously it would seem desirable for this error to be zero at all times, but since we are normally considering systems containing energy storage components, total elimination of error at all times is impossible.

13.5 Generally desirable and acceptable behaviour

Although specific requirements will normally be drawn up for a particular control system, there are important general requirements applicable to the majority of systems. Usually an engineering system will be assembled from readily available components to perform some function, and the choice of these components will be restricted. An example of this would be a diesel engine–alternator set for delivering electrical power, in which normally the most convenient diesel engine–alternator combination will be chosen from those already manufactured.

Even if such a system were assembled from customer-designed components, it would be fortuitous if it performed in a satisfactory self-regulatory way without further consideration of its control dynamics. Hence, it is the control engineer's task to take such a system and devise economical ways of making the overall system behave in a satisfactory manner under the expected operational conditions.

For example, a system may oscillate, i.e. it is unstable; or, although stable, it might tend to settle after a change in input demand to a value unacceptably far from this new demand, i.e. it lacks static accuracy. Again, it might settle to a satisfactory new steady state, but only after an unsatisfactory transient response. Alternatively, normal operational load disturbances on the system may cause unacceptably wide variation of the output variable, e.g. voltage and frequency of the engine–alternator system.

All these factors will normally be quantified in an actual design specification, and fortunately a range of techniques is available for improving the behaviour. But the application of a particular technique to improve the performance of one aspect of behaviour often has a deleterious effect on another, e.g. improved stability with improved static accuracy tends to be incompatible. Thus, a compromise is sought which gives the 'best' acceptable all-round performance. We now discuss

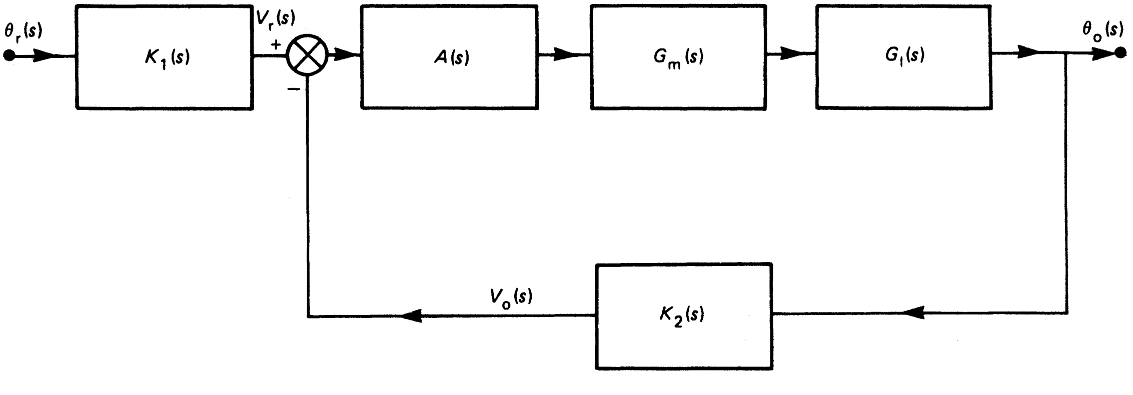


Figure 13.6 Block diagram of the system shown in *Figure 13.5*

some of these concepts and introduce certain techniques useful in examining and designing systems.

13.6 Stability

This is a fairly easy concept to appreciate for the types of system under consideration here. Equation (13.2) with the right-hand side made equal to zero governs the free (or unforced, or characteristic) behaviour of the system, and because of the nature of the governing LCCDE it is well known that the solution will be a linear combination of exponential terms, *viz*.

$$y(t) = \sum_{i=1}^{n} A_i \exp(\alpha_i t) \Leftarrow$$

where the α_i values are the roots of the so-called 'characteristic equation'.

It will be noted that should any α_i have a positive real part (in general, the roots will be complex), then any disturbance will grow in time. Thus, for stability, no roots must lie in the right-hand half of the complex plane or s plane. In a transfer function context this obviously translates to 'the roots of the denominator must not lie in the right-hand half of the complex plane'.

For example, if $W(s) = G(s)/[1 + H(s)G(s)]$, then the roots referred to are those of the equation

$$1 + H(s)G(s) = 0$$

In general, the determination of these roots is a non-trivial task and, as at this stage we are interested only in whether the system is stable or not, we can use certain results from the theory of polynomials to achieve this without the necessity for evaluating the roots.

A preliminary examination of the location of the roots may be made using the *Descartes rule of signs*, which states: if $f(x)$ is a polynomial, the number of positive roots of the equation $f(x) = 0$ cannot exceed the number of changes of sign of the numerical coefficients of $f(x)$, and the number of negative roots cannot exceed the number of changes of sign of the numerical coefficients of $f(-x)$. 'A change of sign' occurs when a term with a positive coefficient is immediately followed by one with a negative coefficient, and vice versa.

Example Suppose that $f(x) = x^3 + 3x - 2 = 0$; then there can be at most one positive root. Since $f(-x) = -x^3 - 3x - 2$, the equation has no negative roots. Further, the equation is cubic and must have at least one real root (complex roots occur in conjugate pairs); therefore the equation has one positive-real root.

Although Descartes' result is easily applied, it is often indefinite in establishing whether or not there is stability, and a more discriminating test is that due to Routh, which we give without proof.

Suppose that we have the polynomial

$$a_0 s^n + a_1 s^{n-1} \ldots a_{n-1} s + a_n = 0$$

where all coefficients are positive, which is a necessary (but not sufficient) condition for the system to be stable, and we construct the following so-called 'Routh array':

$$\begin{array}{lllllll}
s^n & : & a_0 & a_2 & a_4 & a_6 & \ldots \\
s^{n-1} & : & a_1 & a_3 & a_5 & a_7 & \ldots \\
s^{n-2} & : & b_1 & b_2 & b_3 & \ldots & \\
s^{n-3} & : & c_1 & c_2 & c_3 & \ldots & \\
s^{n-4} & : & d_1 & d_2 & \ldots & &
\end{array}$$

where

$$b_1 = \Leftarrow \frac{a_1a_2 - a_0a_3}{a_1}, \varsigma b_2 = \Leftarrow \frac{a_1a_4 - a_0a_5}{a_1}, \varsigma b_3 = \Leftarrow \frac{a_1a_6 - a_0a_7}{a_1}, \ldots$$

$$c_1 = \Leftarrow \frac{b_1a_3 - a_1b_2}{b_1}, \varsigma c_2 = \Leftarrow \frac{b_1a_5 - a_1b_3}{b_1}, \ldots \%$$

$$d_1 = \Leftarrow \frac{c_1b_2 - b_1c_2}{c_1}, \ldots$$

This array will have $n + 1$ rows.

If the array is complete and *none* of the elements in the first column vanishes, then a sufficient condition for the system to be stable (i.e. the characteristic equation has all its roots with negative-real parts) is for all these elements to be positive. Further, if these elements are not all positive, then the number of changes of sign in this first column indicates the number of roots with positive-real parts.

Example Determine whether the polynomial $s^4 + 2s^3 + 6s^2 + 7s + 4 = 0$ has any roots with positive-real parts. Construct the Routh array:

$$\begin{array}{llll}
s^4 : & 1 & 6 & 4 \\
s^3 : & 2 & 7 & \\
s^2 : & \dfrac{(2)(6) - (1)(7)}{2} = 2.5 & \dfrac{(2)(4) - (1)(0)}{2} = 4 & \\
s :! & \dfrac{(2.5)(7) - (2)(4)}{2.5} = 3.8 & & \\
s^0 : & 4 & &
\end{array}$$

There are five rows with the first-column elements all positive, and so a system with this polynomial as its characteristic would be stable.

There are cases that arise which need a more delicate treatment.

(1) Zeros occur in the first column, while other elements in the row containing a zero in the first column are non-zero.

In this case the zero is replaced by a small positive number, ε, which is allowed to approach zero once the array is complete.

For example, consider the polynomial equation

$$s^5 + 2s^4 + 2s^3 + 4s^2 + 11s + 8 = 0:$$

$$\begin{array}{llll}
s^5 : & 1 & 2 & 11 \\
s^4 : & 2 & 4 & 8 \\
s^3 : & \varepsilon\varsigma & 5 & 0 \\
s^2 : & \alpha_1 & 8 & \\
s^1 : & \alpha_2 & 0 & \\
s^0 : & 8 & &
\end{array}$$

where

$$\alpha_1 = \Leftarrow \frac{4\varepsilon\varsigma - 10}{\varepsilon\varsigma} \simeq -\Leftarrow \frac{10}{\varepsilon} \text{ and } \alpha_2 = \frac{5\alpha_1 - 8\varepsilon\varsigma}{\alpha_1} \simeq \Leftarrow 5$$

Thus, α_1 is a large negative number and we see that there are effectively two changes of sign and, hence, the equation has two roots which lie in the right-hand half of this plane.

(2) Zeros occur in the first column and other elements of the row containing the zero are also zero.

This situation occurs when the polynomial has roots that are symmetrically located about the origin of the s plane, i.e. it contains terms such as $(s + j\omega)(s - j\omega)$ or $(s + v)(s - v)$.

This difficulty is overcome by making use of the auxiliary equation which occurs in the row immediately before the zero entry in the array. Instead of the all-zero row the equation formed from the preceding row is differentiated and the resulting coefficients are used in place of the all-zero row.

For example, consider the polynomial $s^3+3s^2+2s+6=0$.

$$\begin{array}{lll} s^3: & 1 & 2 \\ s^2: & 3 & 6 \quad \text{(auxiliary equation } 3s^2+6=0) \\ s^1: & 0 & 0 \end{array}$$

Differentiate the auxiliary equation giving $6s=0$, and compile a new array using the coefficients from this last equation, *viz.*

$$\begin{array}{lll} s^3: & 1 & 2 \\ s^2: & 3 & 6 \\ s^1: & 6 & 0 \\ s^0: & 1 & \end{array}$$

Since there are no changes of sign, the system will not have roots in the right-hand half of the s plane.

Although the Routh method allows a straightforward algorithmic approach to determining the stability, it gives very little clue as to what might be done if stability conditions are unsatisfactory. This consideration is taken up later.

13.7 Classification of system and static accuracy

13.7.1 Classification

The discussion in this section is restricted to unity-feedback systems (i.e. $H(s)=1$) without seriously affecting generalities. We know that the open-loop system has a transfer function $KG(s)$, where K is a constant and we may write

$$KG(s) = \frac{K(s-z_1)(s-z_2)\ldots(s-z_m)}{s^1(s+p_1)(s+p_2)\ldots(s-p_3)} = \frac{K\sum_{k=0}^{m} b_k s^k}{s^1\sum_{k=0}^{n-1} a_k s^k}$$

and for physical systems $n \geq m+1$.

The *order* of the system is defined as the degree of the polynomial in s appearing in the denominator, i.e. n.

The *rank* of the system is defined as the difference in the degree of the denominator polynomial and the degree of the numerator polynomial, i.e. $n-m \geq 1$.

The *class* (or *type*) is the degree of the s term appearing in the denominator (i.e. l), and is equal to the number of integrators in the system.

Example

(1) $G(s) = \dfrac{s+1}{s^4+6s^3+9s^2+3s}$

implies order 4, rank 3 and type 1.

(2) $G(s) = \dfrac{s^2+4s+1}{(s+1)(s^2+2s+4)}$

implies order 3, rank 1 and type 0.

13.7.2 Static accuracy

When a demand has been made on the system, then it is generally desirable that after the transient conditions have decayed the output should be equal to the input. Whether or not this is so will depend both on the characteristics of the system and on the input demand. Any difference between the input and output will be indicated by the error term $e(t)$ and we know that for the system under consideration

$$E(s) = \frac{U(s)}{1+KG(s)}$$

Let $e_{\text{ss}} = \lim_{t\to\infty} e(t)$ (if it exists), and so e_{ss} will be the steady-state error. Now from the final-value theorem we have

$$e_{\text{ss}} = \lim_{t\to\infty} e(t) = \lim_{s\to 0}[sE(s)]$$

Thus,

$$e_{\text{ss}} = \lim_{s\to 0}\left[\frac{sU(s)}{1+KG(s)}\right]$$

13.7.2.1 *Position-error coefficient K_{p}*

Suppose that the input is a unit step, i.e. $R(s)=1/s$; then

$$e_{\text{ss}} = \lim_{s\to 0}\left[\frac{1}{1+KG(s)}\right] = \frac{1}{1+\lim_{s\to 0}[KG(s)]} = \frac{1}{1+K_{\text{p}}}$$

where $K_{\text{p}} = \lim_{s\to 0}[KG(s)]$ and this is called the *position-error coefficient*.

Example For a type-0 system

$$KG(s) = \left[K\sum_{k=0}^{m} b_k s^k\right] \Big/ \left[\sum_{k=0}^{n} a_k s^k\right]$$

Therefore $K_{\text{p}} = K(b_0/a_0)$ and $e_{\text{ss}} = 1/(1+K_{\text{p}})$.

It will be noted that, after the application of a step, there will always be a finite steady-state error between the input and the output, but this will decrease as the gain K of the system is increased.

Example For a type-1 system

$$KG(s) = \left[K\sum_{k=0}^{m} b_k s^k\right] \Big/ \left[s\sum_{k=0}^{n-1} a_k s^k\right]$$

and

$$K_{\text{p}} = \lim_{s\to 0}\left[K\sum_{k=0}^{m} b_k s^k\right] \Big/ \left[s\sum_{k=0}^{n-1} a_k s^k\right] \to \infty$$

Thus,

$$e_{\text{ss}} = \frac{1}{1+\infty} \to 0$$

i.e. there is no steady-state error in this case and we see that this is due to the presence of the integrator term $1/s$. This is an important practical result, since it implies that steady-state errors can be eliminated by use of integral terms.

13.7.2.2 *Velocity-error coefficient, K_{v}*

Let us suppose that the input demand is a unit ramp, i.e. $u(t)=t$, so $U(s)=1/s^2$. Then

$$e_{\text{ss}} = \lim_{s\to 0}[sE(s)] = \lim_{s\to 0}\left[\frac{1}{s+sKG(s)}\right] = \frac{1}{\lim_{s\to 0}[sKG(s)]}$$
$$= \frac{1}{K_{\text{v}}}$$

where $K_v = \lim_{s\to 0}[sKG(s)]$ is called the *velocity-error coefficient*.

Examples For a type-0 system $K_v = 0$, whence $e_{ss} \to \infty$.

For a type-1 system $K_v = K(b_0/a_0)$ and so this system can follow but with a finite error.

For a type-2 system

$$K_v = \lim_{s\to 0}\left[\frac{K}{s}\frac{b_0}{a_0}\right]\Big(\to \infty \Leftarrow$$

whence $e_{ss} \to 0$ and so the system can follow in the steady state without error.

13.7.2.3 Acceleration-error coefficient K_a

In this case we assume that $u(t) = t^2/2$, so $U(s) = 1/s^3$ and so

$$e_{ss} = \lim_{s\to 0}[sE(s)] = \lim_{s\to 0}\left[\frac{1}{s^2 + s^2KG(s)\Leftarrow}\right]\Big($$

$$= \frac{1}{\lim_{s\to 0}[s^2KG(s)]\Leftarrow} = \frac{1}{\Leftarrow K_a}$$

where $K_a = \lim_{s\to 0}[s^2\ KG(s)]$ is called the *acceleration-error coefficient* and similar analyses to the above may be performed.

These error-coefficient terms are often used in design specifications of equipment and indicate the minimum order of the system that one must aim to design.

13.7.3 Steady-state errors due to disturbances

The prototype unity-feedback closed-loop system is shown in *Figure 13.7* modified by the intrusion of a disturbance $D(s)$ being allowed to affect the loop. For example, the loop might represent a speed-control system and $D(s)$ might represent the effect of changing the load. Now, since linear systems are under discussion, in order to evaluate the effects of this disturbance on $Y(s)$ (denoted by $Y_D(s)$), we may tacitly assume $U(s) = 0$ (i.e. invoke the superposition principle)

$$Y_D(s) = D(s) - KG(s)Y_D(s)\Leftarrow$$

$$Y_D(s) = D(s)/[1 + KG(s)]\Leftarrow$$

Now $E_D(s) = -Y_D(s) = -D(s)/[1 + KG(s)]$, and so the steady-state error, e_{ssD} due to the application of the disturbance, may be evaluated by use of the final-value theorem as

$$e_{ssD} = -\lim_{s\to 0}\left[\Big(\frac{sD(s)\Leftarrow}{1 + KG(s)}\right]\Big($$

Obviously the disturbance may enter the loop at other places but its effect may be established by similar analysis.

13.8 Transient behaviour

Having developed a means of assessing stability and steady-state behaviour, we turn our attention to the transient behaviour of the system.

13.8.1 First-order system

It is instructive to examine first the behaviour of a first-order system (a first-order lag with a time constant T) to a unit-step input (*Figure 13.8*).

Now

$$\frac{Y(s)\Leftarrow}{U(s)\Leftarrow} = G(s) = \frac{1}{\Leftarrow 1 + sT}$$

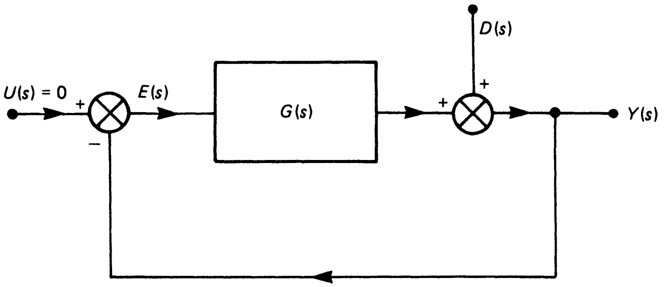


Figure 13.7 Schematic diagram of a disturbance entering the loop

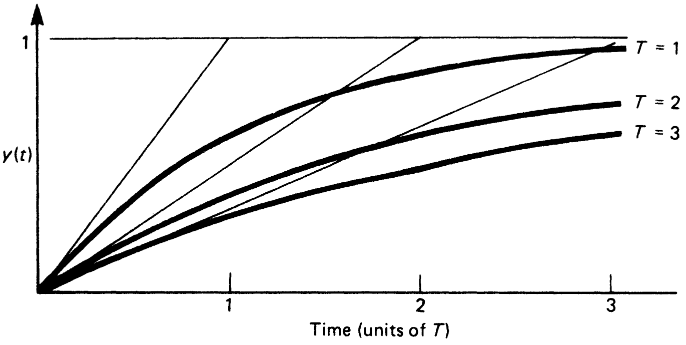


Figure 13.8 First-order lag response to a unit step (time constant = 1, 2, 3, units)

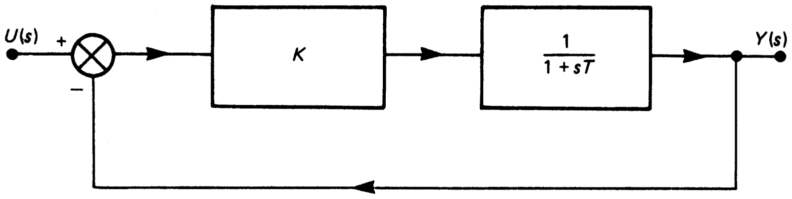


Figure 13.9 First-order lag incorporated in a feedback loop

where $U(s)=1/s$

$$Y(s)=\frac{1}{s(1+sT)}=\frac{1}{Ts[s+(1/T)]}=\frac{1}{s}-\frac{1}{[s+(1/T)]}$$

or $y(t)=1-\exp(-t/T)$;
note also that $\mathrm{d}y/\mathrm{d}t=(1/T)\exp(-t/T)$.

Figure 13.8 shows this time response for different values of T where it will be noted that the corresponding trajectories have slopes of $1/T$ at time $t=0$ and reach approximately 63% of their final values after T.

Suppose now that such a system is included in a unity-feedback arrangement together with an amplifier of gain K (*Figure 13.9*); therefore

$$\frac{Y(s)}{U(s)}=\frac{K/(1+sT)}{1+K/(1+sT)}=\frac{K}{(1+K)\left(1+s\dfrac{T}{1+K}\right)}$$

For a unit-step input the time response will be

$$y(t)=\frac{K}{1+K}[1-\exp\{-(1+K)(t/T)\}]$$

This expression has the same form as that obtained for the open loop but the effective time constant is modified by the gain and so is the steady-state condition (*Figure 13.10*). Such an arrangement provides the ability to control the effective time constant by altering the gain of an amplifier, the original physical system being left unchanged.

13.8.2 Second-order system

The behaviour characteristics of second-order systems are probably the most important of all, since many systems of seemingly greater complexity may often be approximated by a second-order system because certain poles of their transfer function dominate the observed behaviour. This has led to system specifications often being expressed in terms of second-order system behavioural characteristics.

In Section 13.2 the importance of the second-order behaviour of a generator was mentioned, and this subject is now taken further by considering the system shown in *Figure 13.11*.

The closed-loop transfer function for this system is given by

$$W(s)=\frac{KG(s)}{1+KG(s)}=\frac{K}{s^2+as+K}$$

and this may be rewritten in general second-order terms in the form

$$W(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

where $K=\omega_n^2$ and $\zeta=a/(2\sqrt{K})$. The unit-step response is given by

$$y(t)=1-\exp(-\zeta\omega_n t)[\cos(\gamma\omega_n t)-(\zeta/\gamma)\sin(\gamma\omega_n t)]$$

where $\gamma=\sqrt{(1-\zeta^2)}$. This assumes, of course, that $\zeta<1$, so giving an oscillating response decaying with time.

The *rise time* t_r will be defined as the time to reach the first overshoot (note that other definitions are used and it is important to establish which particular definition is being used in a particular specification):

$$t_\mathrm{r}=\pi/(\gamma\omega_n)=\pi/\sqrt{[K-(a/2)^2]}$$

i.e. the rise time decreases as the gain K is increased.

The *percentage overshoot* is defined as:

$$\text{Percentage overshoot}=\frac{100(\text{Max. value of }y(t)-\text{Steady-state value})}{\text{Steady-state value}}$$

$$=100\exp(-\zeta\pi/\gamma)=100\exp[-\alpha\pi/\sqrt{(4K-a^2)}]$$

i.e. the percentage overshoot increases as the gain K increases.

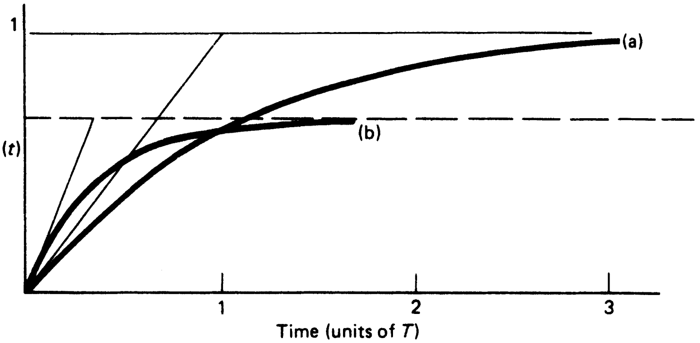


Figure 13.10 Response of first-order lag: (a) open-loop condition ($T=1$); (b) closed-loop condition ($T=1$, $K=2$)

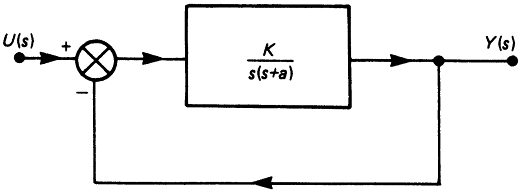


Figure 13.11 Second-order system

The *frequency of oscillation* ω_r is immediately seen to be

$$\omega_r = \omega_n \gamma \varsigma = \sqrt{[K - (a/2)^2]} \Leftarrow$$

i.e. the frequency of oscillation increases as the gain K increases.

The *predominant time constant* is the time constant associated with the envelope of the response (*Figure 13.12*) which is given by $\exp(-\zeta\omega_n t)$ and thus the predominant time constant is $1/\zeta\omega_n$:

$$\frac{1}{\zeta\omega_n} = \Leftarrow \frac{1}{(a/2\sqrt{K})\sqrt{K}} = \Leftarrow \frac{2}{a}$$

Note that this time constant is unaffected by the gain K and is associated with the 'plant parameter a', which will normally be unalterable, and so other means must be found to alter the predominant time constant should this prove necessary.

The *settling time* t_s is variously defined as the time taken for the system to reach 2–5% (depending on specification) of its final steady state and is approximately equal to four times the predominant time constant.

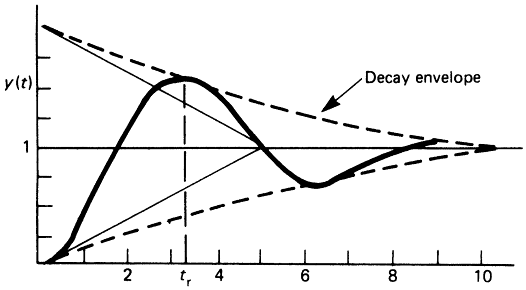


Figure 13.12 Step response of the system shown in *Figure 13.11*. The rise time t_r is the time taken to reach maximum overshoot. The predominant time constant is indicated by the tangents to the envelope curve

It should be obvious from the above that characteristics desired in plant dynamical behaviour may be conflicting (e.g. fast rise time with small overshoot) and it is up to the skill of the designer to achieve the best compromise. Overspecification can be expensive.

A number of the above items can be directly affected by the gain K and it may be that a suitable gain setting can be found to satisfy the design with no further attention. Unfortunately, the design is unlikely to be as simple as this, in view of the fact that the predominant time constant cannot be influenced by K. A particularly important method for influencing this term is the incorporation of so-called *velocity feedback*.

13.8.3 Velocity feedback

Given the prototype system shown in *Figure 13.11*, suppose that this is augmented by measuring the output $y(t)$, differentiating to form $\dot{y}(t)$, and feeding back in parallel with the normal feedback a signal proportional to $\dot{y}(t)$: say $T\dot{y}(t)$. The schematic of this arrangement is shown in *Figure 13.13*. Then, by simple manipulation, the modified transfer function becomes

$$W'(\Leftarrow s) = \Leftarrow \frac{K}{s^2 + (a + KT)s + K}$$

whence the modified predominant time constant is given by $2/(a + TK)$. The designer effectively has another string to his bow in that manipulation of K and T is normally very much in his command.

A similar effect may be obtained by the incorporation of a *derivative* term to act on the error signal (*Figure 13.14*) and in this case the transfer function becomes

$$W'(\Leftarrow s) = \Leftarrow \frac{K(1 + Ts)\Leftarrow}{s^2 + (a + KT)s + K}$$

It may be demonstrated that this derivative term when correctly adjusted can both stabilise the system and increase the speed of response. The control shown in *Figure 13.14* is referred to as *proportional-plus-derivative* control and is very important.

13.8.4 Incorporation of integral control

Mention has previously been made of the effect of using integrators within the loop to reduce steady-state errors; a particular study with reference to input/output effects was given. In this section consideration is given to the effects of disturbances injected into the loop, and we consider again

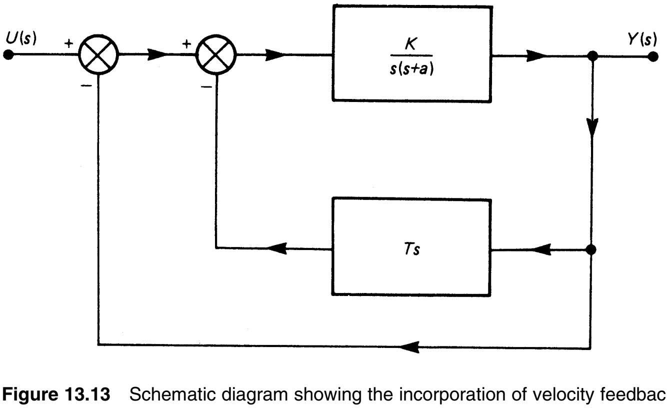


Figure 13.13 Schematic diagram showing the incorporation of velocity feedback

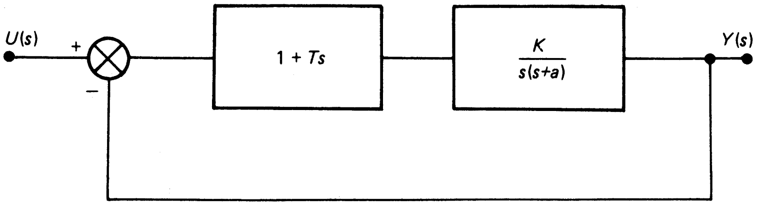


Figure 13.14 Schematic diagram of the proportional-plus-derivative control system

the simple second-order system shown in *Figure 13.11* but with a disturbance occurring between the amplifier and the plant dynamics. Appealing to superposition we can, without loss of generality, put $U(s) = 0$ and the transfer function between the output and the disturbance is then given by

$$\frac{Y(s)}{D(s)} = \frac{1}{s(s+a)+K}$$

Assuming that $d(t)$ is a unit step, $D(s) = 1/s$, and using the final-value theorem, $\lim_{t\to\infty} y(t)$ is obtained from

$$\lim_{t\to\infty} y(t) = \lim_{s\to\infty}\left[\frac{s}{[s(s+a)+K]s}\right] = \frac{1}{K}$$

and so the effect of this disturbance will always be present. By incorporating an integral control as shown in *Figure 13.15*, the output will, in the steady state, be unaffected by the disturbance, *viz*.

$$Y(s) = \frac{Ts}{Ts^2(s+a)+K(1+Ts)}D(s)$$

and so

$$y_{ss} \to 0$$

This controller is called a *proportional-plus-integral* controller.

An unfortunate side-effect of incorporating integral control is that it tends to destabilise the system, but this can be minimised by careful choice of T. In a particular case it might be that *proportional-plus-integral-plus-derivative (PID) control* may be called for, the amount of each particular control type being carefully proportioned.

In the foregoing discussions we have seen, albeit by using specific simple examples, how the behaviour of a plant might be modified by use of certain techniques. It is hoped that this will leave the reader with some sort of feeling for what might be done before embarking on more general tools, which tend to appear rather rarefied and isolated unless a basic physical feeling for system behaviour is present.

13.9 Root-locus method

The root locus is merely a graphical display of the *variation of the poles of the closed-loop system* when some parameter, often the gain, is varied. The method is useful since the loci may be obtained, at least approximately, by straightforward application of simple rules, and possible modification to reshape the locus can be assessed.

Considering once again the unity-feedback system with the open-loop transfer function $KG(s) = Kb(s)/a(s)$, where $b(s)$ and $a(s)$ represent mth- and nth-order polynomials, respectively, and $n > qn$, then the closed-loop transfer function may be written as

$$W(s) = \frac{KG(s)}{1+KG(s)} = \frac{Kb(s)}{a(s)+Kb(s)}$$

Note that the system is nth order and the zeros of the closed loop and the open loop are identical for unity feedback. The characteristic behaviour is determined by the roots of $1 + KG(s) = 0$ or $a(s) + Kb(s) = 0$. Thus, $G(s) = -(1/K)$ or $b(s)/a(s) = -(1/K)$.

Let s_r be a root of this equation; then

$$\text{mod}\left[\frac{b(s_r)}{a(s_r)}\right] = \frac{1}{K}$$

and

$$\text{phase}\left[\frac{b(s_r)}{a(s_r)}\right] = 180^\circ + n360^\circ$$

where n may take any integer value, including $n = 0$. Let $z_1, \ldots, z_m$ be the roots of the polynomial $b(s) = 0$, and $p_1, \ldots, p_n$ be the roots of the polynomial $a(s) = 0$. Then

$$b(s) = \prod_{i=1}^{m}(s - z_i)$$

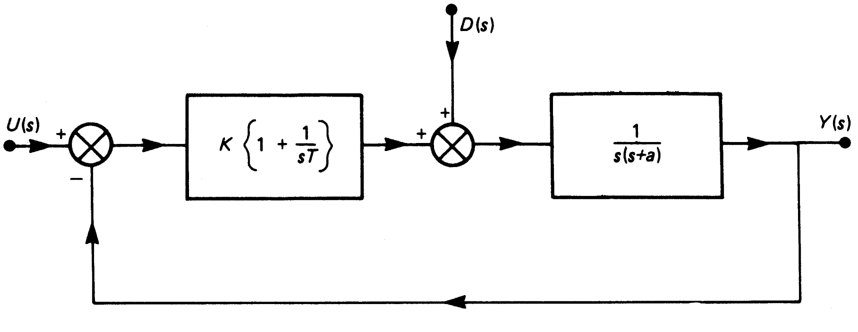


Figure 13.15 Schematic diagram of the proportional-plus-integral control system

and

$$a(s) = \prod_{i=1}^{n} (s - p_i) \Leftarrow$$

Therefore

$$\frac{\prod_{i=1}^{m} |s_r - z_i| \Leftarrow}{\prod_{i=1}^{n} |s_r - p_i| \Leftarrow} = \Leftarrow \frac{1}{K}, \varsigma \textit{ the magnitude condition}$$

and

$$\sum_{i=1}^{m} \text{phase}(s_r - z_i) - \sum_{i=1}^{n} \text{phase}(s_r - p_i) = 180^\circ + n360^\circ,$$

the angle or phase condition

Now, given a complex number p_j, the determination of the complex number $(s - p_j)$, where s is some point in the complex plane, is illustrated in *Figure 13.16*, where the mod$(s - p_j)$ and phase$(s - p_j)$ are also illustrated. The determination of the magnitudes and phase angles for all the factors in the transfer function, for any s, can therefore be done graphically.

The complete set of all values of s, constituting the root locus may be constructed using the angle condition alone; once found, the gain K giving particular values of s_r may be easily determined from the magnitude condition.

Example Suppose that $G(s) = K/[(s + a)(s + b)]$, then it is fairly quickly established that the only sets of points satisfying the angle condition

$$-\text{phase}(s_r + a) - \text{phase}(s_r + b) = 180 + n360^{\circ\Leftarrow}$$

are on the line joining $-a$ to $-b$ and the perpendicular bisector of this line (*Figure 13.17*).

13.9.1 Rules for construction of the root locus

(1) The angle condition must be obeyed.
(2) The magnitude condition enables calibration of the locus to be carried out.
(3) The root locus on the real axis must be in sections to the left of an odd number of poles and zeros. This follows immediately from the angle condition.

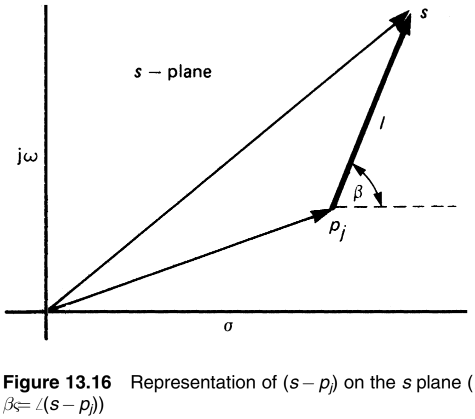


Figure 13.16 Representation of $(s - p_j)$ on the s plane ($l = |s - p_j|$; $\beta \Leftarrow \angle(s - p_j)$)

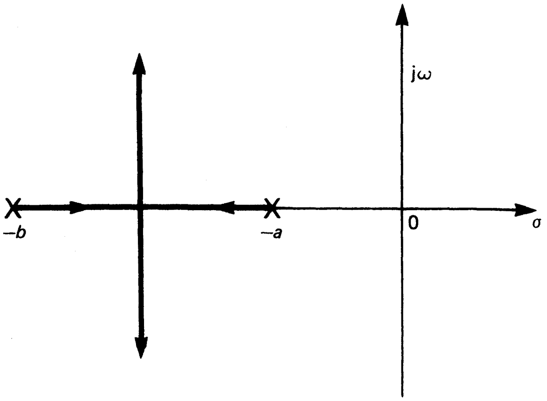


Figure 13.17 Root-locus diagram for $KG(s) = K/[(s + a)(s + b)]$

(4) The root locus must be symmetrical with respect to the horizontal real axis. This follows because complex roots must appear as complex conjugate pairs.
(5) Root loci always emanate from the poles of the open-loop transfer function where $K = 0$. Consider $a(s) + Kb(s) = 0$; then $a(s) = 0$ when $K = 0$ and the roots of this polynomial are the poles of the open-loop transfer function. Note that this implies that there will be n branches of the root locus.
(6) m of the branches will terminate at the zeros for $K \to \infty$. Consider $a(s) + Kb(s) = 0$, or $(1/K)a(s) + b(s) = 0$, whence as $K \to \infty$, $b(s) \to 0$ and, since this polynomial has m roots, these are where m of the branches terminate. The remaining $n - m$ branches terminate at infinity (in general, complex infinity).
(7) These $n - m$ branches go to infinity along asymptotes inclined at angles ϕ_i to the real axis, where

$$\phi_i = \Leftarrow \frac{(2_i + 1)\Leftarrow}{n - m} 180^\circ, \varsigma \quad i = 0, 1, \ldots, (n - m - 1)\Leftarrow$$

Consider a root s_r approaching infinity, $(s_r - a) \to s_r$ for all finite values of a. Thus, if ϕ_i is the phase s_r, then each pole and each zero term of the transfer function term will contribute approximately ϕ_i and $-\phi_i$, respectively. Thus,

$$\phi_i(n - m) = 180^\circ + i360^\circ$$

$$\phi_i = \Leftarrow \frac{(2i + 1)}{n - m} 180^\circ, \varsigma i = 0, 1, \ldots, (n - m - 1)\Leftarrow$$

(8) The centre of these asymptotes is called the '*asymptote centre*' and is (with good accuracy) given by

$$\sigma_A = \Leftarrow \left(\sum_{i=1}^{n} p_i - \sum_{j=1}^{m} z_i \right) \Big/ (n - m)\Leftarrow$$

This can be shown by the following argument. For very large values of s we can consider that all the poles and zeros are situated at the point σ_A on the real axis. Then the characteristic equation (for large values of s) may be written as

$$1 + \frac{K}{(s + \sigma_A)^{n-m}} = 0$$

or approximately, by using the binomial theorem,

$$1 + \frac{K}{s^{n-m} + (n - m)s^{n-m-1}\sigma_A} = 0$$

Also, the characteristic equation may be written as

$$1 + \Leftarrow \frac{K\prod_{i=1}^{m}(s+z_i)\Leftarrow}{\prod_{i=1}^{m}(s+p_i)} = 0$$

Expanding this for the first two terms results in

$$1 + \frac{K}{s^{n-m} + (a_{n-1} - b_{m-1})s^{n-m-1}} = 0$$

where

$$b_{m-1} = \Leftarrow \sum_{i=1}^{m} z_i \quad \text{and} \quad a_{n-1} = \Leftarrow \sum_{i=1}^{n} p_j$$

whence

$$(a_{n-1} - b_{m-1}) = (n-m)\sigma_A$$

$$\sigma_A = \Leftarrow \frac{a_{n-1} - b_{m-1}}{n-m}$$

as required.

(9) When a locus breaks away from the real axis, it does so at the point where K is a local maximum. Consider the characteristic equation $1 + K[b(s)/a(s)] = 0$; then we can write $K = p(s)$, where $p(s) = -[a(s)/b(s)]$. Now, where two poles approach each other along the real axis they will both be real and become equal when K has the maximum value that will enable them both to be real and, of course, coincident. Thus, an evaluation of K around the breakaway point will rapidly reveal the breakaway point itself.

Example Draw the root locus for

$$KG(s) = \Leftarrow \frac{K(s+1)\Leftarrow}{s(s+2)(s+3)\Leftarrow}$$

Procedure (*Figure 13.18*):

(1) Plot the poles of the open-loop system (i.e. at $s = 0$, $s = -2$, $s = -3$).
(2) Plot the zeros of the system (i.e. at $z = -1$).
(3) Determine the sections on the real axis at which closed-loop poles can exist. Obviously these are between 0 and -1 (this root travels along the real axis between these values as K goes from $0 \to \infty$), and between -2 and -3 (two roots are moving towards each other as K increases and, of course, will break away).
(4) Angle of asymptotes

$$\phi_1 = \frac{1}{2} + 180^\circ = 90^{\circ\Leftarrow}$$

$$\phi_2 = \frac{3}{2} \times 180^\circ = 270^{\circ\Leftarrow}$$

(5) Centroid σ_A is located at

$$\sigma_A = \Leftarrow \frac{-2-3+1}{2} = -2$$

(6) Breakaway point, σ_B

σ_B	-2.45	-2.465	-2.48
K	0.418	0.4185	0.418

(7) Modulus. For a typical root situated at, for example, point A, the gain is given by $K = l_2 l_3 l_4 / l_1$.

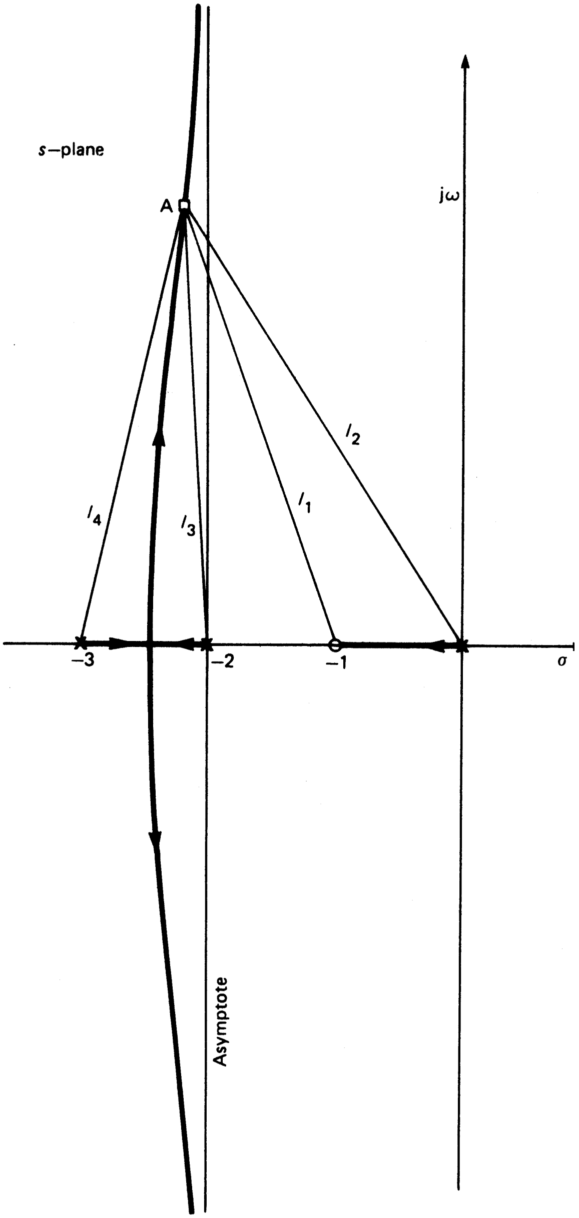


Figure 13.18 Root-locus construction for $KG(s) = [K(s+1)]/[s(s+2)(s+3)]$

After a little practice the root locus can be drawn very rapidly and compensators can be designed by pole-zero placement in strategic positions. A careful study of the examples given in the table will reveal the trends obtainable for various pole-zero placements.

13.10 Frequency-response methods

Frequency-response characterisation of systems has led to some of the most fruitful analysis and design methods in the whole of control system studies. Consider the situation of a linear, autonomous, stable system, having a transfer function $G(s)$, and being subjected to a unit-magnitude sinusoidal input signal of the form exp ($j\omega t$), starting at $t = 0$. The Laplace transformation of the resulting output of the system is

$$C(s) = G(s)/(s - j\omega)\Leftarrow$$

and the time domain solution will be

$$c(t) = G(j\omega)\exp(j\omega t) + \Leftarrow\left(\begin{array}{l}\text{Terms whose exponential terms}\\ \text{correspond to the roots}\\ \text{of the denominator of } G(s)\Leftarrow\end{array}\right)$$

Since a stable system has been assumed, then the effects of the terms in the parentheses will decay away with time and so, after a sufficient lapse of time, the steady-state solution will be given by

$$c_{\mathrm{ss}}(t) = G(j\omega)\exp(j\omega t)\Leftarrow$$

The term $G(j\omega)$, obtained by merely substituting $j\omega$ς for s in the transfer function form, is termed the *frequency-response function*, and may be written

$$G(j\omega) = |G(j\omega)|\angle G(j\omega)\Leftarrow$$

where $|G(j\omega)| =$ mod $G(j\omega)$ and $\angle G(j\omega) =$ phase $G(j\omega)$. This implies that the output of the system is also sinusoidal in magnitude $|G(j\omega)|$ with a phase-shift of $\angle G(j\omega)$ with reference to the input signal.

Example Consider the equation of motion

$$m\ddot{y} + bz + ky = f(t)\Leftarrow$$

$$\frac{Y(s)\Leftarrow}{F(s)\Leftarrow} = G(s) = \Leftarrow\frac{1}{ms^2 + bs + k}$$

If $f(t) = F_0$ exp($j\omega t$), then

$$y_{\mathrm{ss}}(t) = \Leftarrow\frac{F_0\exp(j\omega t)\Leftarrow}{(k - \omega^2 m) + j\omega b}$$

whence

$$y_{\mathrm{ss}}(t) = \frac{F_0\exp[j(\omega t - \phi)]\Leftarrow}{\sqrt{(k - \omega^2 m)^2 + (b\omega)^2}}$$

where ϕ⇐ arctan $b\omega/(k - m\omega^2)$.

Within the area of frequency-response characterisation of systems three graphical techniques have been found to be particularly useful for examining systems and are easily seen to be related to each other. These techniques are based upon:

(1) The *Nyquist plot*, which is the locus of the frequency-response function plotted in the complex plane using ως as a parameter. It enables stability, in the closed-loop condition, to be assessed and also gives an indication of how the locus might be altered to improve the behaviour of the system.
(2) The *Bode diagram*, which comprises two plots, one showing the amplitude of the output frequency response (plotted in decibels) against the frequency ως (plotted logarithmically) and the other of phase angle θς of the output frequency response plotted against the same abscissa.
(3) The *Nichols chart*, a direct plot of amplitude of the frequency response (again in decibels) against the phase angle, with frequency ως as a parameter, but further enables the closed-loop frequency response to be read directly from the chart.

In each of these cases it is the *open-loop* steady-state frequency response, i.e. $G(j\omega)$, which is plotted on the diagrams.

13.10.1 Nyquist plot

The closed-loop transfer function is given by

$$\frac{C(s)\Leftarrow}{R(s)\Leftarrow} = \Leftarrow\frac{G(s)\Leftarrow}{1 + H(s)G(s)\Leftarrow}$$

and the stability is determined by the location of the roots of $1 + H(s)G(s) = 0$, i.e. for stability no roots must have positive-real parts and so must not lie on the positive-real half of the complex plane. Assume that the open-loop transfer function $H(s)G(s)$ is stable and consider the contour C, the so-called 'Nyquist contour' shown in *Figure 13.19*, which consists of the imaginary axis plus a semicircle of large enough radius in the right half of the s plane such that any zeros of $1 + H(s)G(s)$ will be contained within this contour. This contour C_n is mapped via $1 + H(s)G(s)$ into another curve γς into the complex plane s'. It follows immediately from complex variable theory that the closed loop will be stable if the curve γς does not encircle the origin in the s'⇐plane and unstable if it encircles the origin or passes through the origin. This result is the basis of the celebrated Nyquist stability criterion. It is rather more usual to map not $1 + H(s)G(s)$ but $H(s)G(s)$; in effect this is merely a change of origin from (0, 0) to (−1, 0), i.e. we consider curve γ_n'⇐

The statement of the stability criterion is that the closed-loop system will be stable if the mapping of the contour C_n by the open-loop frequency-response function $H(j\omega)G(j\omega)$ does not enclose the so-called critical point (−1, 0). Actually further simplification is normally possible, for:

(1) $|H(s)G(s)| \to 0$ as $|s| \to \infty$, so that the very large semicircular boundary maps to the origin in the s'⇐plane.
(2) $H(-j\omega)G(-j\omega)$ is the complex conjugate of $H(j\omega)G(j\omega)$ and so the mapping of $H(-j\omega)G(-j\omega)$ is merely the mirror image of $H(j\omega)G(j\omega)$ in the real axis.
(3) Note: $H(j\omega)G(j\omega)$ is merely the frequency-response function of the open loop and may even be directly measurable from experiments. Normally we are mostly interested in how this behaves in the vicinity of the (−1, 0) point and, therefore, only a limited frequency range is required for assessment of stability.

The mathematical mapping ideas stated above are perhaps better appreciated practically by the so-called *left-hand rule* for an open-loop stable system, which reads as follows: if the open-loop sinusoidal response is traced out going from low frequencies towards higher frequencies, the closed loop will be stable if the critical point (−1, 0) lies on the left of all points on $H(j\omega)G(j\omega)$. If this plot passes through the critical point, or if the critical point lies on the right-hand side of $H(j\omega)G(j\omega)$, the closed loop will be unstable.

If the open loop has poles that actually lie on the imaginary axis, e.g. integrator $1/s$, then the contour is indented as shown in *Figure 13.20* and the above rule still applies to this modification.

13.10.1.1 Relative stability criteria

Obviously the closer the $H(j\omega)G(j\omega)$ locus approaches the critical point, the more critical is the consideration of stability, i.e. we have an indication of relative stability,

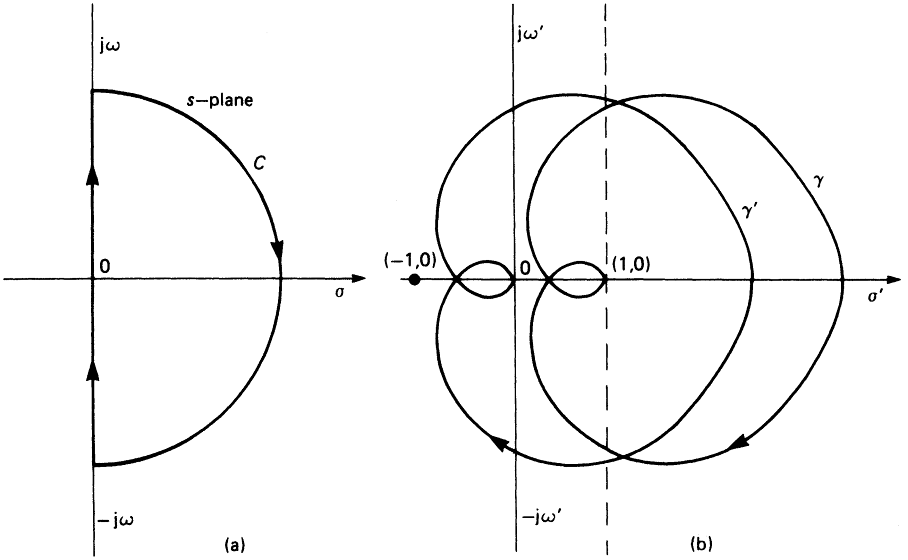


Figure 13.19 Illustration of Nyquist mapping: (a) mapping contour on the *s* plane; (b) resulting mapping of $1 + H(s)G(s) = 0$ and the shift of the origin

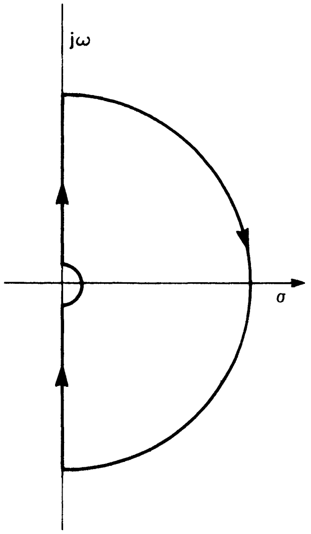


Figure 13.20 Modification of the mapping contour to account for poles appearing at the origin

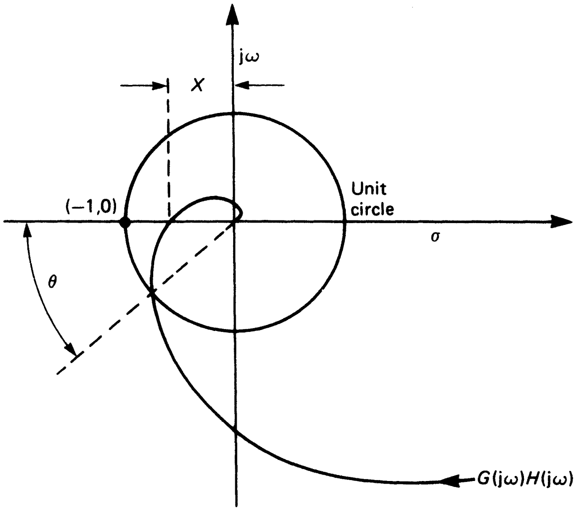


Figure 13.21 Illustration of the gain and phase margins. Gain margin $= 1/X$; phase margin $= 0$

given a measure by the gain and phase margins of the system.

If the modulus of $H(j\omega)G(j\omega) = X$ with a phase shift of 180°, then the *gain margin* is defined as

Gain margin $= 1/X$

The gain margin is usually specified in decibels, where we have

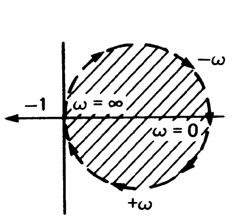
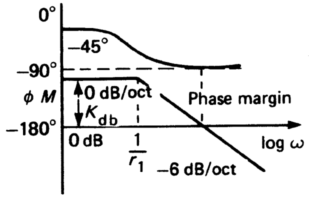
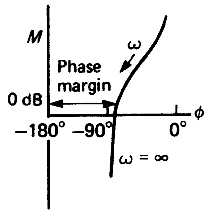
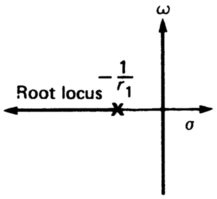
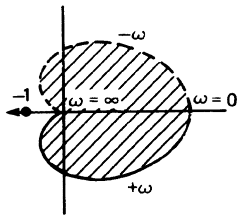
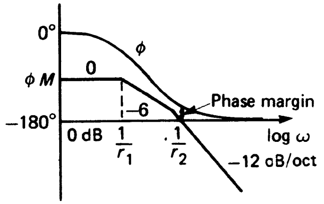
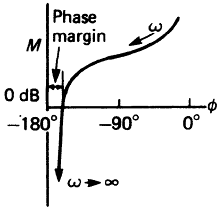
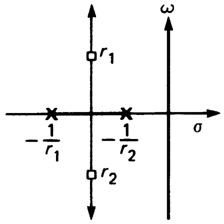
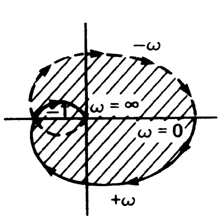
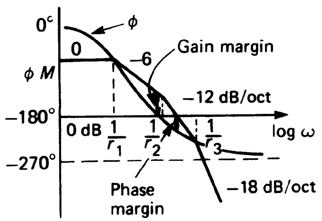
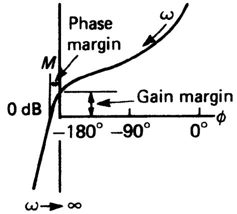
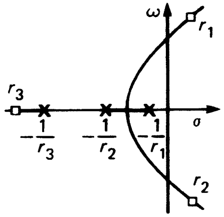
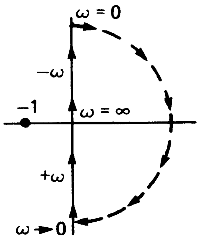
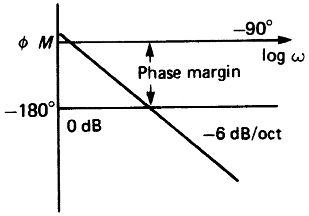
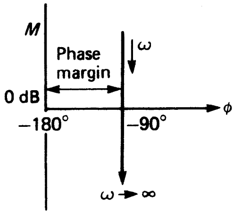
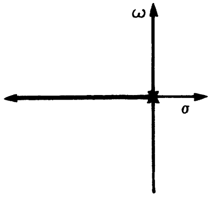
Gain margin (dB) $= 20\log(1/X) = -20\log X$

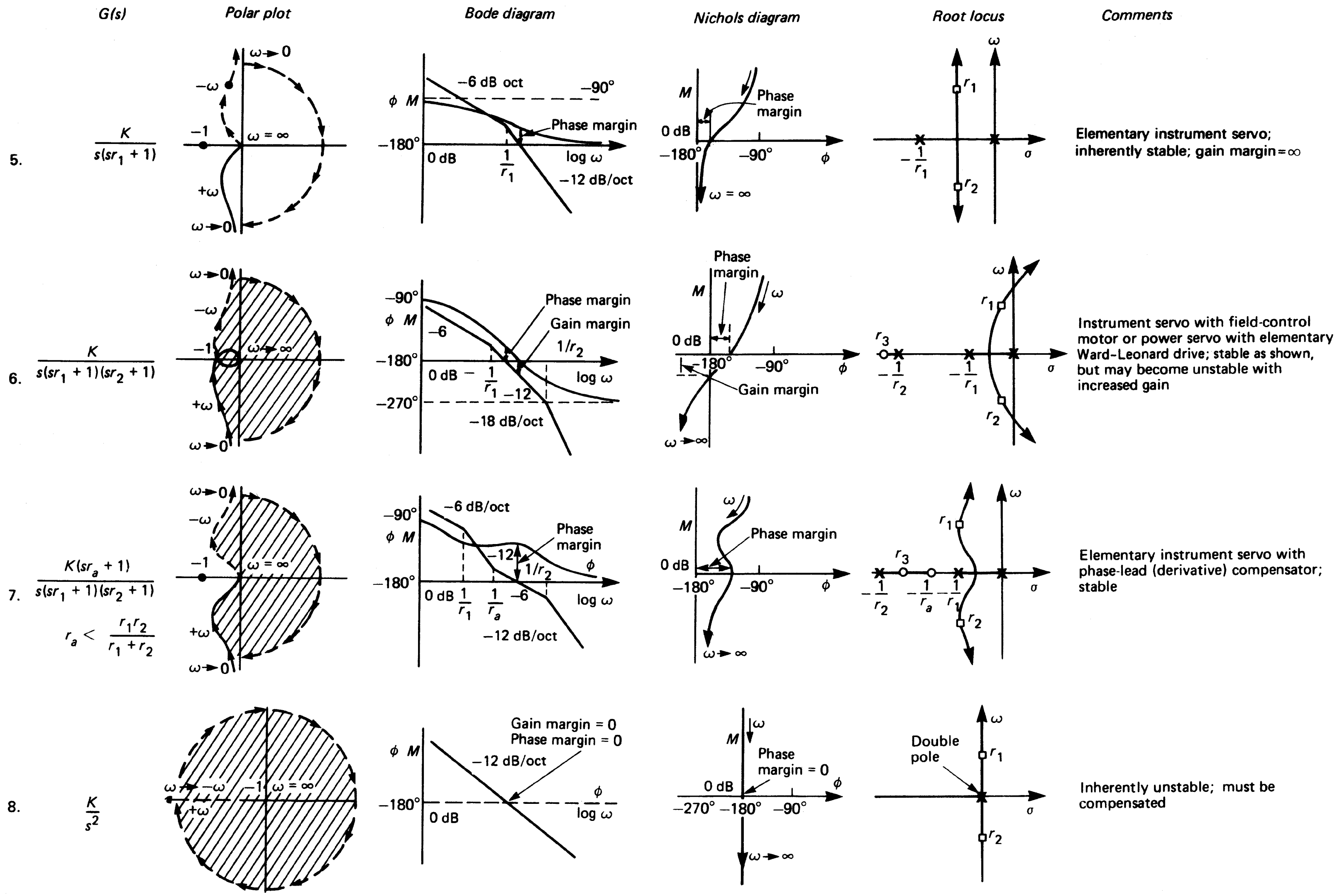
The *phase margin* is the angle which the line joining the origin to the point on the open-loop response locus corresponding to unit modulus of gain makes with the negative-real axis. These margins are probably best appreciated diagrammatically (*Figure 13.21*). They are useful, since a rough working rule for reasonable system damping and stability is to shape the locus so that a gain margin of at least 6 dB is achieved and a phase margin of about 40°.

Examples of the Nyquist plot are shown in *Figure 13.22*. Although from such plots the modifications necessary to achieve more satisfactory performance can be easily appreciated, precise compensation arrangements are not easily determined, since complex multiplication is involved and an appeal to the Bode diagram can be more valuable.

13.10.2 Bode diagram

As mentioned above, the Bode diagram is a logarithmic presentation of the frequency response and has the advantage

	G(*s*)	*Polar plot*	*Bode diagram*	*Nichols diagram*	*Root locus*	*Comments*
1.	$\frac{K}{sr_1 + 1}$		 0° −45° −90° φ M 0 dB/oct K_{db} Phase margin −180° 0 dB $\frac{1}{r_1}$ log ω −6 dB/oct	 M ω Phase margin 0 dB φ −180° −90° 0° ω = ∞		Stable; gain margin = ∞
2.	$\frac{K}{(sr_1+1)(sr_2+1)}$		 0° φ 0 φ M −6 Phase margin −180° 0 dB $\frac{1}{r_1}$ $\frac{1}{r_2}$ log ω −12 oB/oct	 Phase margin ω M 0 dB φ −180° −90° 0° ω → ∞	 ω r_1 σ $-\frac{1}{r_1}$ $-\frac{1}{r_2}$ r_2	Elementary regulator; stable; gain margin = ∞
3.	$\frac{K}{(sr_1+1)(sr_2+1)(sr_3+1)}$		 0° φ 0 −6 Gain margin φ M −12 dB/oct −180° 0 dB $\frac{1}{r_1}$ $\frac{1}{r_2}$ $\frac{1}{r_3}$ log ω −270° Phase margin −18 dB/oct	 Phase margin ω M Gain margin 0 dB φ −180° −90° 0° ω → ∞	 ω r_1 r_3 σ $-\frac{1}{r_3}$ $-\frac{1}{r_2}$ $-\frac{1}{r_1}$ r_2	Regulator with additional energy-storage component; unstable, but can be made stable by reducing gain
4.	$\frac{K}{s}$		 −90° φ M log ω Phase margin −180° 0 dB −6 dB/oct	 M Phase margin ω 0 dB φ −180° −90° ω → ∞		Ideal integrator; stable



	G(s)	Comments
5.	$\frac{K}{s(sr_1+1)}$	Elementary instrument servo; inherently stable; gain margin=∞
6.	$\frac{K}{s(sr_1+1)(sr_2+1)}$	Instrument servo with field-control motor or power servo with elementary Ward-Leonard drive; stable as shown, but may become unstable with increased gain
7.	$\frac{K(sr_a+1)}{s(sr_1+1)(sr_2+1)}$ $r_a < \frac{r_1 r_2}{r_1+r_2}$	Elementary instrument servo with phase-lead (derivative) compensator; stable
8.	$\frac{K}{s^2}$	Inherently unstable; must be compensated

Figure 13.22 Transfer function plots for typical transfer functions

over the Nyquist diagram that individual factor terms may be added rather than multiplied, the diagram can usually be quickly sketched using asymptotic approximations and several decades of frequency may be easily considered.

Now suppose that

$$H(s)G(s) = H(s)G_1(s)G_2(s)G_3(s)\dots$$

i.e. the composite transfer function may be thought of as being composed of a number of simpler transfer functions multiplied together, so

$$|H(j\omega)G(j\omega)| = |H(j\omega)||G_1(j\omega)||G_2(j\omega)||G_3(j\omega)|\dots$$
$$20\log|H(j\omega)G(j\omega)| = \Leftarrow 20\log|H(j\omega)| + 20\log|G_1(j\omega)| \Leftarrow$$
$$+\ 20\ \log|G_2(j\omega)| + 20\ \log|G_3(j\omega)| + \cdots \Leftarrow$$

This is merely each individual factor (in decibels) being *added* algebraically to a grand total. Further,

$$\angle H(j\omega)G(j\omega) = \angle H(j\omega) + \angle G_1(j\omega) + \angle G_2(j\omega) \Leftarrow$$
$$+\ \angle G_3(j\omega) + \cdots \Leftarrow$$

i.e. the individual phase shift at a particular frequency may be *added* algebraically to give the total phase shift.

It is possible to construct Bode diagrams from elemental terms including gain (K), differentiators and integrators (s and $1/s$), lead and lag terms ($(as+1)$ and $(1+as)^{-1}$), quadratic lead and lag terms ((bs^2+cs+1) and $(bs^2+cs+1)^{-1}$), and we consider the individual effects of their presence in a transfer function on the shape of the Bode diagram.

(*a*) *Gain term*, K The gain in decibels is simply 20 log K and is *frequency independent*; it merely raises (or lowers) the combined curve 20 log K dB.

(*b*) *Integrating term*, $1/s$ Now $|G(j\omega)| = 1/\omega$ς and $\angle G(j\omega) = -90°$⇐(a constant) and so the gain in decibels is given by 20 $\log(1/\omega) = -20$ log ω. On the Bode diagram this corresponds to a straight line with slope -20 dB/decade (or -6 dB/octave) of frequency and passes through 0 dB at $\omega = 1$ (see plot 4 in *Figure 13.22*).

(*c*) *Differentiating term*, s Now $|G(j\omega)| = \omega$ςand $\angle G(j\omega) = 90°$⇐ (a constant) and so the gain in decibels is given by 20 log ω. On the Bode diagram this corresponds to a straight line with slope 20 dB/decade of frequency and passes through 0 dB at $\omega = 1$.

(*d*) *First-order lag term*, $(1+s\tau)^{-1}$ The gain in decibels is given by

$$20\log\left(\left(\frac{1}{1+\omega^2\tau^2}\right)^{1/2} = -10\log(1+\omega^2\tau^2)\right.$$

and the phase angle is given by $\angle G(j\omega) = -\tan^{-1}\ \omega\tau$. When $\omega^2\tau^2$ is small compared with unity, the gain will be approximately 0 dB, and when $\omega^2\tau^2$ is large compared with unity, the gain will be -20 log $\omega\tau$. With logarithmic plotting this specifies a straight line having a slope of -20 dB/decade of frequency (6 dB/octave) intersecting the 0 dB line at $\omega = 1/\tau$. The actual gain at $\omega = 1/\tau$ςis -3 dB and so the plot has the form shown in plot 1 of *Figure 13.22*. The frequency at which $\omega = 1/\tau$ςis called the *corner or break frequency*. The two straight lines, i.e. those with 0 dB and -20 dB/decade, are called the 'asymptotic approximations' to the Bode plot. These approximations are often good enough for not too demanding design purposes.

The phase plot will lag a few degrees at low frequencies and fall to $-90°$⇐at high frequency, passing through $-45°$⇐at the break frequency.

(*e*) *First-order lead term*, $1+\omega\tau$ς The lead term properties may be argued in a similar way to the above, but the gain, instead of falling, rises at high frequencies at 20 dB/decade and the phase, instead of lagging, leads by nearly 90°⇐at high frequencies.

(*f*) *Quadratic-lag term*, $1/(1+2\tau\zeta s+\tau^2 s^2)$ The gain for the quadratic lag is given by

$$-10\log\left[\ 1-\left(\frac{\omega}{\omega_n}\right)^2\right)^2 + \left(\Leftarrow 2\zeta\frac{\omega}{\omega_n}\right)^2\Bigg]\Bigg($$

and the phase angle by

$$\angle G(j\omega) = \text{arctan}\left[-\frac{2\zeta(\omega/\omega_n)\Leftarrow}{1-(\omega/\omega_n)^2}\right]\Bigg($$

where $\tau = 1/\omega_n$. At low frequencies the gain is approximately 0 dB and at high frequencies falls at -40 dB/decade. At the break frequency $\omega = 1/\tau$ the actual gain is $20\log(1/2\zeta)$. For low damping (say $\zeta < 0.5$) an asymptotic plot can be in considerable error around the break frequency and more careful evaluation may be required around this frequency. The phase goes from minus a few degrees at low frequencies towards $-180°$⇐at high frequencies, being $-90°$⇐at $\omega = 1/\tau$.

(*g*) *Quadratic lead term*, $1+2\tau\zeta s+\tau^2 s^2$ This is argued in a similar way to the lag term with the gain curves inscribed and the phase going from plus a few degrees to 180°⇐in this case.

Example Plot the Bode diagram of the open-loop frequency-response function

$$G(j\omega) = \Leftarrow\frac{10(1+j\omega)\Leftarrow}{j\omega(j\omega+2)(j\omega+3)\Leftarrow}$$

and determine the gain and phase margins (see *Figure 13.23*). Note: *Figure 13.22* shows a large number of examples and also illustrates the gain and phase margins.

13.10.3 Nichols chart

This is a graph with the open-loop gain in decibels as co-ordinate and the phase as abscissa. The open-loop frequency response is for a particular system and is plotted with frequency ως as parameter. Now the closed-loop frequency response is given by

$$W(j\omega) = \Leftarrow\frac{G(j\omega)\Leftarrow}{1+G(j\omega)\Leftarrow}$$

and corresponding lines of constant magnitude and constant phase of $W(j\omega)$ are plotted on the Nichols chart as shown in *Figure 13.24*.

When the open-loop frequency response of a system has been plotted on such a chart, the closed-loop frequency response may be immediately deduced from the contours of $W(j\omega)$.

13.11 State-space description

Usually in engineering, when analysing time-varying physical systems, the resulting mathematical models are in differential equation form. Indeed, the introduction of the Laplace transformation, and similar techniques, leading to the whole edifice of transfer-function-type analysis and design methods are, essentially, techniques for solving, or manipulating to

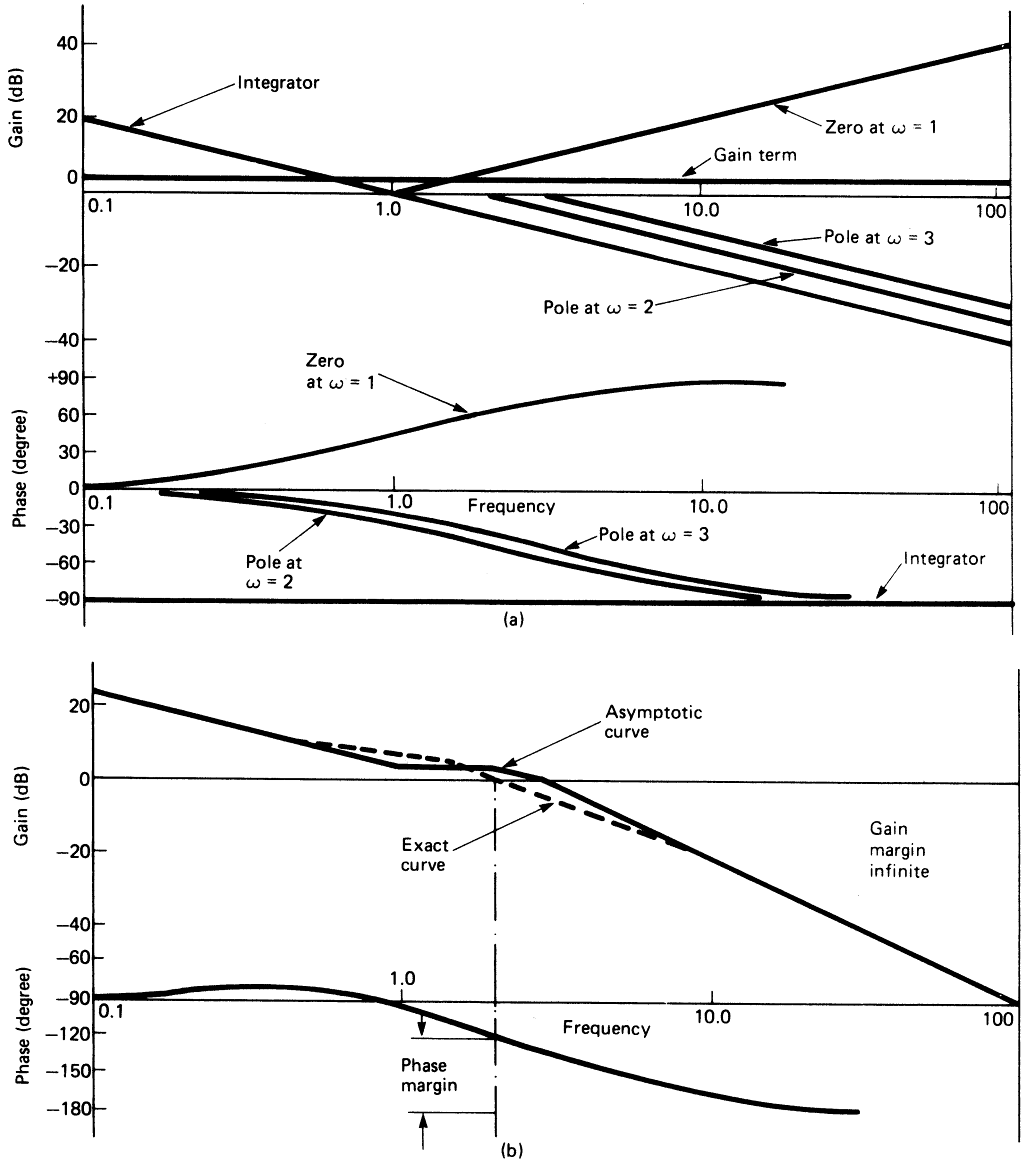


Figure 13.23 (a) Gain and phase curves for individual factors (see *Figure 13.18*); (b) Composite gain and phase curves. Note that the phase margin is about 60°, and the gain margin is infinite because the phase tends asymptotically to −180°⇐

advantage, differential equation models. In the state-space description of systems, which is the concern of this section, the models are left in the differential equation form, but rearranged into the form of a set of first-order simultaneous differential equations. There is nothing unique to systems analysis in doing this, since this is precisely the required form that differential equations are placed in if they are to be integrated by means of many common numerical techniques, e.g. the Runge–Kutta methods. Most of the interest in the state-space form of studying control systems stems from the 1950s, and intensive research work in this area has continued since then; however, much of it is of a highly theoretical nature. It is arguable that these methods have yet to fulfill the hopes and aspirations of the research workers who developed them. The early expectation was that they would quickly supersede classical techniques. This has been very far from true, but they do have a part to play, particularly if there are good mathematical models of the plant available and the real plant is well instrumentated.

Consider a system governed by the *n*th order linear constant-coefficient differential equation

$$\frac{d^n y}{dt^n} + \cdots \Leftarrow a_1 \frac{dy}{dt} + a_0 y = \Leftarrow ku(t) \Leftarrow$$

where *y* is the dependent variable and *u*(*t*) is a time-variable forcing function.

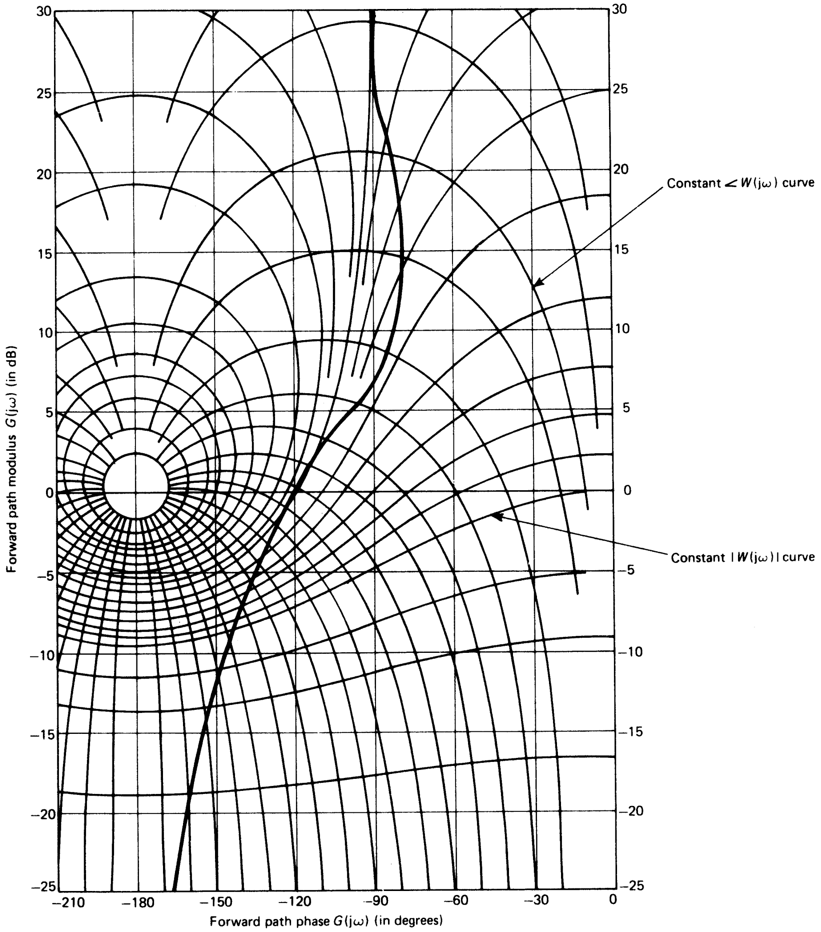


Figure 13.24 Nichols chart and plot of the system shown in *Figure 13.23*. Orthogonal families of curves represent constant $W(j_\omega)$ and constant $\angle W(j_\omega)$

Let $y = x_1$, then

$$\frac{dy}{dt} = \Leftarrow \frac{dx_1}{dt} = \Leftarrow x_2$$

say, and

$$\frac{d^2y}{dt^2} = \frac{dx_2}{dt} = x_3$$

$$\frac{d^{n-1}y}{dt^{n-1}} = \frac{dx_{n-1}}{dt} = \Leftarrow x_n$$

From the governing differential equation we can write

$$\frac{d^n y}{dt^n} = \Leftarrow \frac{dx_n}{dt} = -a_0x_1 - a_1x_2 \cdots - a_{n-1}x_n + ku(t) \Leftarrow$$

i.e. the *n*th order differential equation has been transformed into *n* first-order equations. These can be arranged into matrix form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \Leftarrow \begin{bmatrix} 0 & 1 & 0 & \dots & \varsigma \\ 0 & 0 & 1 & 0 \dots & \varsigma \\ & & 0 & 1 & \\ -a_0 & -a_1 & \dots & & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \Leftarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ k \end{bmatrix} u(t) \Leftarrow \qquad (13.3)\Leftarrow$$

which may be written in matrix notation as

$$\dot{\mathbf{x}} = \Leftarrow \mathbf{A}\mathbf{x} + \mathbf{b}u(t) \Leftarrow$$

where $\mathbf{x} = [x_1, \ldots, x_n]^T$ and is called the 'state vector', $\mathbf{b} = [0, 0, \ldots, k]^T$ and **A** is the $n \times n$ matrix pre-multiplying **x** on the right-hand side of Equation (13.3).

It can be shown that the eigenvalues of **A** are equal to the characteristic roots of the governing differential equation which are also equal to the poles of the transfer function $Y(s)/U(s)$. Thus the time behaviour of the matrix model is essentially governed by the position of the eigenvalues of the **A** matrix (in the complex plane) in precisely the same manner as the poles govern the transfer function behaviour. Hence, if these eigenvalues do not lie in acceptable positions in this plane, the design process is to somehow modify the **A** matrix so that the corresponding eigenvalues do have acceptable positions (cf. the placement of closed-loop poles in the *s* plane).

Example Consider a system governed by the general second-order linear differential equation

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n\frac{dy}{dt} + \omega_n^2 y = \Leftarrow \omega_n^2 u$$

Let $y = x_1$, then

$$\frac{dy}{dt} = \Leftarrow \frac{dx_1}{dt} = x_2$$

and so

$$\frac{dx_2}{dt} = -\omega_n^2 x_1 - 2\zeta\omega_n x_2 + \omega_n^2 u$$

or

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega_n^2 & -2\zeta\omega_n\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \omega_n^2\end{bmatrix}u \qquad (13.4)\Leftarrow$$

The eigenvalues of the **A** matrix are given by the solution to the equation $\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0$, i.e.

$$\lambda_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

Now let $u = r - k_1x_1 - k_2x_2$ where r is an arbitrary, or reference, value or input, and k_1 and k_2 are constants. Note this is a feedback arrangement, since u has become a linear function of the state variables which, in a dynamic system, might be position and velocity. Substituting for u in equation (13.3), gives

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega_n^2(1+k_1) & -\omega_n(2\zeta + \omega_n k_2)\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \omega_n^2\end{bmatrix}r$$

The eigenvalues of the **A** matrix are given by the roots of $\lambda^2 + (2\zeta\omega_n + \omega_n^2k_2)\lambda + \omega_n^2(1+k_1) = 0$ and, by choosing suitable values for k_1 and k_2 (the feedback factors), the eigenvalues can be made to lie in acceptable positions in the complex plane. Note that, in this case, k_1 alters the effective undamped natural frequency, and k_2 alters the effective damping of the second-order system.

If the governing differential equation has derivatives on the right-hand side, then the derivation of the first-order set involves a complication. Overcoming this is easily illustrated by an example. Suppose that

$$\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_2y = b_0u + b_1\frac{du}{dt}$$

Let $y = x_1$, and

$$\frac{dy}{dt} = \Leftarrow \frac{dx_1}{dt} = x_2 + b_1u$$

then

$$\frac{d^2y}{dt^2} = \frac{dx_2}{dt} + b_1\frac{du}{dt} = -a_1(x_2 + b_1u) - a_0x + b_0u + b_1\frac{du}{dt}$$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -a_0 & -a_1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}b_1\\ b_0 - a_1b_1\end{bmatrix}u$$

Note that care may be necessary in interpreting the x derivatives in a physical sense.

The state-space description is also a convenient way of dealing with multi-input/multi-output systems. A simple example is shown in *Figure 13.25*, where $U_1(s)$ and $U_2(s)$ are the inputs and $Y_1(s)$ and $Y_2(s)$ are the corresponding outputs, and so

$$Y_1(s) = \frac{k_1}{s + a_1}U_1(s) + \frac{k_3}{s + a_3}U_2(s)\Leftarrow$$

and

$$Y_2(s) = \frac{k_2}{s + a_1}U_1(s) + \frac{k_4}{s + a_4}U_2(s)$$

The first of these two equations may be written as

$$[s^2 + s(a_1 + a_2) + a_1a_3]Y_1(s) = k_1U_1(s) + k_3U_2(s)\Leftarrow$$

or

$$\frac{d^2y_1}{dt^2} + (a_1 + a_2)\frac{dy_1}{dt} + a_1a_3y_1 = k_1u_1 + k_zu_2$$

let $y_1 = x_1$, then

$$\frac{dy_1}{dt} = \frac{dx_1}{dt} = x_2$$

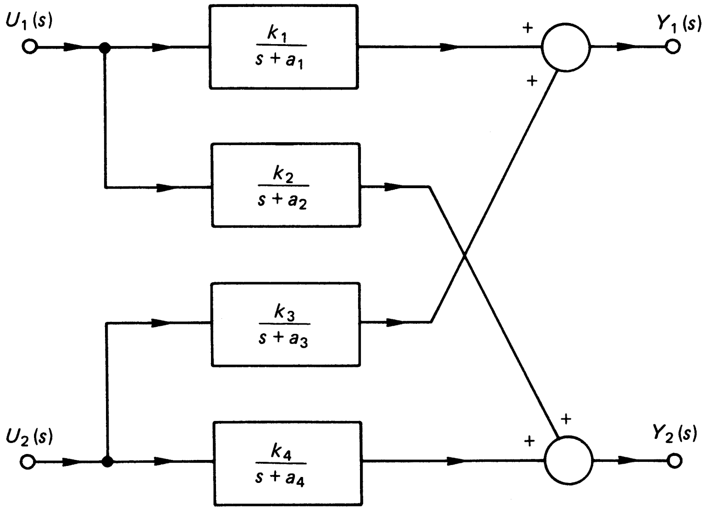


Figure 13.25 Block diagram of a two-input/two-output multi-variable system

and

$$\frac{d^2y_1}{dt^2} = \frac{dx_2}{dt} = -(a_1 + a_3)x_2 - a_1a_3x_1 + k_1u_1 + k_3u_2$$

Similarly, for the second of the two equations, writing

$$y_2 = x_3 \text{ and } \frac{dy_2}{dt} = \frac{dx_3}{dt} = x_4$$

leads to

$$\frac{d^2y_2}{dt^2} = \frac{dx_4}{dt} = -(a_2 + a_4)x_4 - a_2a_4x_3 + k_2u_1 + k_4u_2$$

Whence the entire set may be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -a_1a_3 & -(a_1 - a_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -a_2a_4 & -(a_2 + a_4) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k_1 & k_3 \\ 0 & 0 \\ k_2 & k_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The problem is now how to specify u_1 and u_2 (e.g. a linear combination of state variables similar to the simple second-order system above), so as to make the plant behave in an acceptable manner. It must be pointed out that the theory of linear matrix-differential equations is an extremely well developed mathematical topic and has been extensively plundered in the development of state-space methods. Thus a vast literature exists, and this is not confined to linear systems. Such work has, among other things, discovered a number of fundamental properties of systems (for example, controllability and observability); these are well beyond the scope of the present treatment. The treatment given here is a very short introduction to the fundamental ideas of the state-space description.

13.12 Sampled-data systems

Sampled-data systems are ones in which signals within the control-loop are sampled at one or more places. Some sort of sampling action may be inherent in the very mode of operation of some of the very components comprising the plant, e.g. thyristor systems, pulsed-radar systems and reciprocating internal combustion engines. Moreover, sampling is inevitable if a digital computer is used to implement the control laws, and/or used in condition monitoring operations. Nowadays, digital computers are routinely used in control-system operation for reasons of cheapness and versatility, e.g. they may be used not only to implement the control laws, which can be changed by software alterations alone, but also for sequencing control and interlocking in, say, the start up and safe operation of complex plant. Whatever the cause, sampling complicates the mathematical analysis and design of systems.

Normally most of the components, comprising the system to be controlled, will act in a continuous (analogue) manner, and hence their associated signals will be continuous. With the introduction of a digital computer it is necessary to digitise the signal, by an analogue-to-digital converter before the signal enters the computer. The computer processes this digital sequence, and then outputs another digital sequence which, in turn, passes to a digital-to-analogue converter. This process is shown schematically in *Figure 13.26.*

In this diagram the sampling process is represented by the periodic switch (period T), which at each sampling instant is closed for what is regarded as an infinitesimal time. The digital-to-analogue process is represented by the hold block. Thus the complete system is a hybrid one, made up of an interconnection of continuous and discrete devices. The most obvious way of representing the system mathematically is by a mixed difference-differential equation set. However, this makes a detailed analysis of the complete system difficult.

Fortunately, provided the investigator or system designer is prepared to accept knowledge of the system's behaviour at the instants of sampling only, a comparatively simple approach having great similarity to that employed for wholly continuous systems is available. At least for early stages of the analysis or design proposal, the added complications involved in this process are fairly minor. Further, the seemingly severe restriction of knowing the system's behaviour at the instants of sampling only is normally quite acceptable; for example, the time constants associated with the plant will generally be much longer than the periodic sampling time, so the plant effectively does not change its state significantly in the periodic time. The sampling time period is a parameter which often can be chosen by the designer, who will want sampling to be fast enough to avoid aliasing problems; however, the shorter the sampling period the less time the computer has available for other loops. Suffice it to say that the selection of the sampling period is normally an important matter.

If we take a continuous signal $y(t)$, say, and by the periodic sampling process convert it into a sequence of values $y(n)$, where n represents the nth sampling period, then the sequence $y(n)$ becomes the mathematical entity we manipulate, and the values of $y(t)$ between these samples will not be known. However, if at an early stage it is essential to know the inter-sample behaviour of the system with some accuracy, then advance techniques are available for this purpose.[1] In addition, it is now fairly routine to simulate control system behaviour before implementation, and a

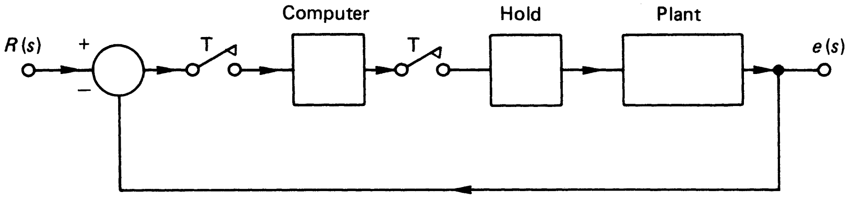


Figure 13.26 General arrangement of a sampled-data system

good simulation package should be capable of illustrating the inter-sample behaviour.

We need techniques for mathematically manipulating sequences, and these are discussed in the following section.

13.13 Some necessary mathematical preliminaries

13.13.1 The *z* transformation

This transformation plays the equivalent role in sampled-data system studies as the Laplace transformation does in the case of continuous systems; indeed, these two transformations are mathematically related to each other. It is demonstrated below that the behaviour of sampled-data systems at the sampling instant is governed mathematically by difference equations, e.g. a linear system might be governed by the equation

$$y(n) + a_1 y(n-1) + a_2 y(n-2) = b_1 x(n) + b_2 x(n-1)\Leftarrow$$

where, in the case of $y(n)$, the value of a variable at instant n is in fact dependent on a linear combination of its previous two values and the current and previous values of an independent (forcing variable) $x(n)$. In a similar way to using the Laplace transformation to convert linear differential equations to transfer-function form, the z transformation is used to convert linear difference equations into the so-called 'pulse transfer-function form'. The definition of the z transformation of a sequence $y(n)$, $n = 0, 1, 2, \ldots$, is

$$Z[y(n)] = Y(z) = \sum_{n=0}^{\propto} y(n)z^{-n}$$

The z transformations of commonly occurring sequences are listed in *Table 13.1*, and a simple example will illustrate how such transformations may be found.

Suppose $y(n) = nT(n = 0, 1, 2, \ldots)$ such a sequence would be obtained by sampling the continuous ramp function $y(t) = t$, at intervals of time T. Then, by definition,

$$\begin{aligned} Z[y(n)] &= Y(z) = 0 + Tz^{-1} + 2Tz^{-2} + \cdots \Leftarrow \\ &= T(z^{-1} + 2z^{-2} + \cdots) \Leftarrow \\ &= \frac{Tz}{(z-1)^2} \end{aligned}$$

It can also be shown that

$$Z[y(n-1)] = z^{-1}Z[y(n)] = z^{-1}Y(z)\Leftarrow$$

and

$$Z[y(n-2)] = z^{-2}Z[y(n)] = z^{-2}Y(z)\Leftarrow$$

Then, applying this to the difference equation above, we have

$$Y(z) = -(a_1 z^{-1} + a_2 z^{-2})Y(z) + (b_0 + b_1 z^{-1})X(z)\Leftarrow$$

or

$$Y(z) = \Leftarrow\frac{(b_0 + b_1 z^{-1})X(z)\Leftarrow}{(1 + a_1 z^{-1} + a_2 z^{-2})\Leftarrow}$$

So that, if $x(n)$ or $X(z)$ is given, $Y(z)$ can be rearranged into partial fraction form, and $y(n)$ determined from the table. For example, suppose that

$$Y(z) = \Leftarrow\frac{z(z-0.25)\Leftarrow}{(z-1)(z-0.5)\Leftarrow}$$

then

$$\frac{Y(z)\Leftarrow}{z} = \Leftarrow\frac{1.5}{z-1} - \Leftarrow\frac{0.5}{z-0.5}$$

or

$$Y(z) = \Leftarrow\frac{1.5z}{z-1} - \Leftarrow\frac{0.5z}{z-0.5}$$

Whence, from the tables we see that

$$y(n) = 1.5 - 0.5\exp(-0.60n)\Leftarrow$$

The process of dividing $Y(z)$ by z before taking partial fractions is important, as most tabulated values of the transformation have z as a factor in the numerator, and the partial function expansion process needs the order of the denominator to exceed that of the numerator.

An alternative method of approaching the z transform is to assume that the sequence to be transformed is a direct consequence of sampling a continuous signal using an impulse modulator. Thus a given signal $y(t)$ is sampled with periodic time T, to give the assumed signal $y^*(t)$, where

$$y^*(t) = y(o)\delta(t) + y(T)\delta(t-T) + y(2T)\delta(t-2T) + \cdots \Leftarrow$$

where $\delta(t)$ is the delta function.

Taking the Laplace transformation of $y^*(t)$ gives the series

$$\mathscr{L}[y^*(t)] = y(o) + y(T)e^{-sT} + y(2T)e^{-2sT} + \cdots \Leftarrow$$

On making the substitution $e^{sT} = z$, then the resulting series is identical to that obtained by taking the z transformation of the sequence $y(n)$. For convenience, we often write $Y(z) = Z[y^*(t)]$.

$z = e^{sT}$ may be regarded as constituting a transformation of points in an s plane to those in a z plane, and this has exceedingly important consequences. If, for example, we map lines representing constant damping ζ, and constant natural frequency ω_n, for a system represented in an s plane onto a z plane, we obtain *Figure 13.27*.

There are important results to be noted from this diagram.

(1) The stability boundary in the s plane (i.e. the imaginary axis) transforms into the unit circle $|z| = 1$ in the z plane.
(2) Points in the z plane indicate responses relative to the periodic sampling time T.
(3) The negative real axis of the z plane always represents half the sampling frequency ω_s, where $\omega_s = 2\pi/T$.
(4) Vertical lines (i.e. those with constant real parts) in the left-half plane of the s plane map into circles *within* the unit circle in the z plane.
(5) Horizontal lines (i.e. lines of constant frequency) in the s plane map into radial lines in the z planes.
(6) The mapping is not one-to-one; and frequencies greater than $\omega_s/2$ will coincide on the z plane with corresponding points below this frequency. Effectively this is a consequence of the Nyquist sampling theorem which states, essentially, that faithful reconstruction of a sampled signal cannot be achieved if the original continuous signal contained frequencies greater than one-half the sampling frequency.

A vitally important point to note is that *all the roots* of the denominator of a pulse transfer function of a system must fall *within* the unit circle, on the z plane, if the system is to be stable; this follows from (1) above.

13.14 Sampler and zero-order hold

The sampler produces a series of discrete values at the sampling instant. Although in theory these samples exist for

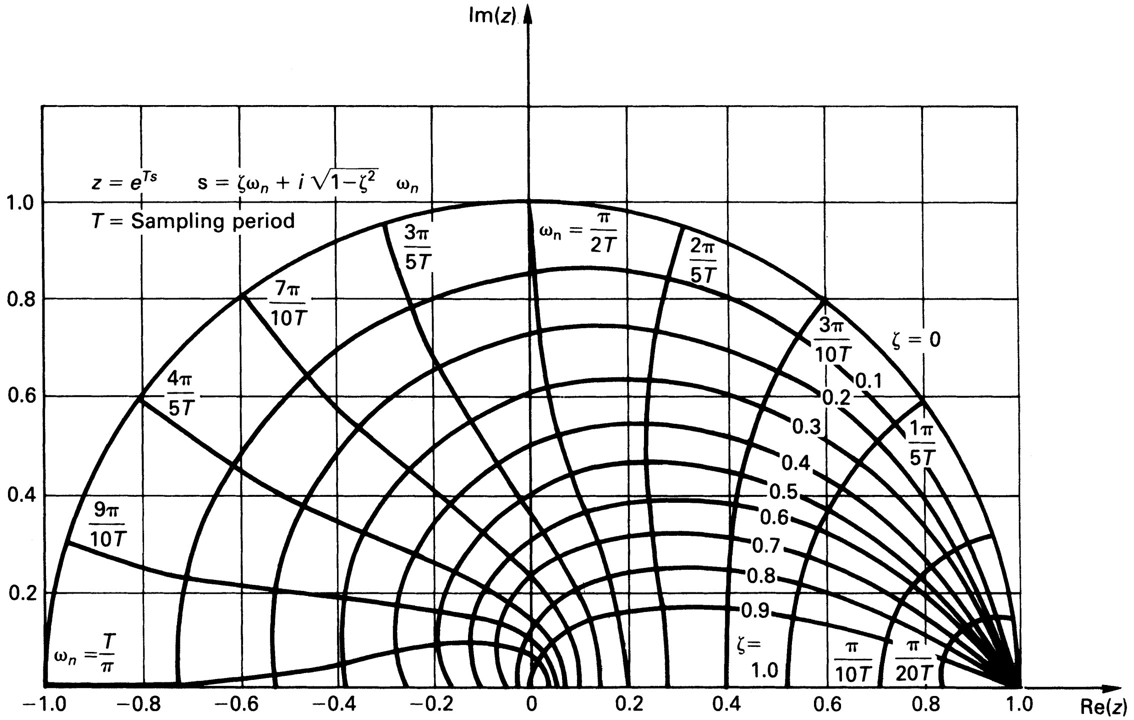


Figure 13.27 Natural frequency and damping loci in the *z* plane. The lower half is the mirror image of the half shown. (Reproduced from Franklin *et al.*,[2] courtesy of Addison-Wesley)

zero time, in practice they can be taken into the digital computer and processed. The output from the digital computer will be a sequence of samples with, again in theory, each sample existing for zero time. However, it is necessary to have a continuous signal constructed from this output, and this is normally done using a *zero-order hold*. This device has the property that, as each sample (which may be regarded as a delta function) is presented to its input, it presents the strength of the delta function at its output until the next sample arrives, and then changes its output to correspond to this latest value, and so on.

This is illustrated diagrammatically in *Figure 13.28*. Thus a unit delta function $\delta(t)$ arriving produces a positive unit-value step at the output at time t. At time $t = \Leftarrow T$, we may regard a negative unity-value step being superimposed on the output. Since the transfer function of a system may be regarded as the Laplace transformation of the response of that system to a delta function, the zero-order hold has the transfer function

$$\frac{1}{s}[1-\Leftarrow e^{-sT}]\Leftarrow$$

13.15 Block diagrams

In a similar way to their use in continuous-control-system studies, block diagrams are used in sampled-data-system studies. It is convenient to represent individual pulse transfer functions in individual boxes. The boxes are joined together by lines representing their *z* transformed input/output sequences to form the complete block diagrams. The manipulation of

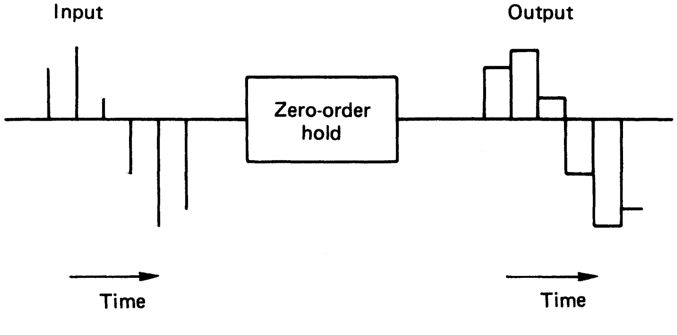


Figure 13.28 Diagrammatic representation of input/output for zero-order hold

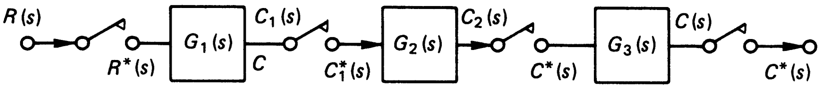


Figure 13.29 Cascade transfer functions with sampling between connections

the block diagrams may be conducted in a similar fashion to that adopted for continuous systems. Again, it must be stressed that such manipulation breaks down if the boxes load one another.

Consider the arrangement shown in *Figure 13.29*. Here we have a number of continuous systems, represented by their transfer functions, in cascade. However, a sampler has been placed in each signal line, and so for each box we may write

$$C_1(s) = G_1(s)R^*(s) \to C_1^*(s) = G_1^*(s)R^*(s)\Leftarrow$$

$$C_2(s) = G_2(s)C_1^*(s) \to C_2^*(s) = G_2^*(s)C_1^*(s)\Leftarrow$$

$$C(s) \ = G_3(s)C_2^*(s) \to C^*(s) = G_1(s)C_2^*(s)\Leftarrow$$

Thus

$$C^*(s) = G_1^*(s)G_2^*(s)G_3^*(s)R^*(s)\Leftarrow$$

i.e.

$$\frac{C(z)\Leftarrow}{R(z)\Leftarrow} = G_1(z)G_2(z)G_3(z)\Leftarrow$$

This, of course, generalises for n similar pulse transfer functions in series to give

$$\frac{C(z)\Leftarrow}{R(z)\Leftarrow} = \Leftarrow\prod_{i=1}^{n} G_i(z)\Leftarrow$$

It is necessary to realise that this result does not apply if there is no sampler between two or more boxes. As an illustration, *Figure 13.30*(*a*) shows the arrangement for which the above result applies. We have

$$G_1(s) = \Leftarrow\frac{1}{s}$$

whence (see *Table 13.1*)

$$G_1(z) = \Leftarrow\frac{z}{z-1}$$

and

$$G_2(s) = \Leftarrow\frac{1}{s+1}$$

whence (see *Table 13.1*)

$$G_2(z) = \Leftarrow\frac{z}{z-e^{-T}}$$

Therefore,

$$\frac{C(z)\Leftarrow}{R(z)} = G_1(z)G_2(z) = \Leftarrow\frac{z^2}{(z-1)(z-e^{-T})\Leftarrow}$$

Figure 13.30(*b*) shows the arrangement *without* a sampler between $G_1(s)$ and $G_2(s)$, and so

$$\frac{C(z)\Leftarrow}{R(z)\Leftarrow} = Z\left[\frac{1}{s(s+1)}\right]\Leftarrow\frac{z(1-e^{-T})}{(z-1)(z-e^{-T})}$$

Note that $Z[G_1(s)G_2(s)]$ is often written $G_1G_2(z)$, and thus, in general, $G_1(z)G_2(z) \neq G_1G_2(z)$.

13.16 Closed-loop systems

Figure 13.31 shows the sampler in the error channel of an otherwise continuous system. We may write

$$C(s) = G(s)E^*(s)\Leftarrow$$

and

$$E(s) = R(s) - H(s)C(s)\Leftarrow$$

or

$$E(s) = R(s) - H(s)G(s)E^*(s)\Leftarrow$$

and

$$E^*(s) = R^*(s) - HG^*(s)E^*(s)\Leftarrow$$

and so

$$E^*(s) = \Leftarrow\frac{R^*(s)\Leftarrow}{1 + HG^*(s)\Leftarrow}$$

Thus

$$\frac{C^*(s)}{R^*(s)\Leftarrow} = \Leftarrow\frac{G^*(s)\Leftarrow}{1 + HG^*(s)\Leftarrow}$$

or

$$\frac{C(z)\Leftarrow}{R(z)\Leftarrow} = \Leftarrow\frac{G(z)\Leftarrow}{1 + HG(z)\Leftarrow}$$

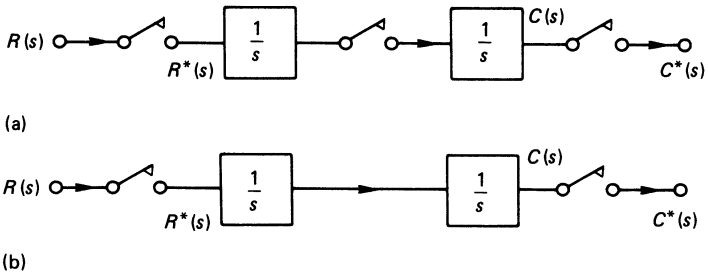


Figure 13.30 Two transfer functions with (a) sampler interconnection and (b) with continuous signal connecting transfer functions

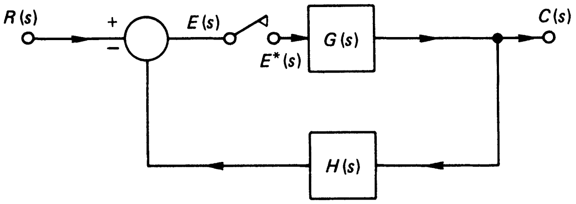


Figure 13.31 Prototype sampled system with a sampler in the error channel

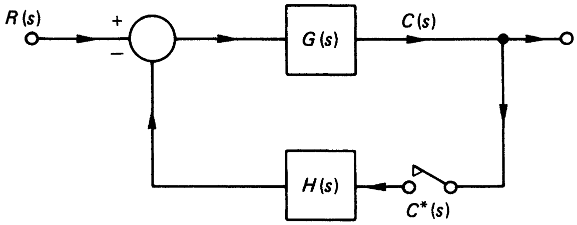


Figure 13.32 Prototype sampled system with a sampler in the feedback channel

If the sampler is in the feedback loop, as shown in *Figure 13.32*, a similar analysis would show that

$$C(z) = \Leftarrow\frac{GR(z)\Leftarrow}{1 + HG(z)\Leftarrow}$$

Note that, in this case, it is not possible to take the ratio $C(z)/R(z)$. We may conclude that the position of the sampler(s) within the loop has a vitally important effect on the behaviour of the system.

Example Consider the arrangement shown in *Figure 13.33*. To calculate the pulse transfer function it is necessary to determine

$$L\left[\frac{1}{s}\,\frac{1}{s}\,(1\Big(- \mathrm{e}^{-sT})\right]\Big(\Big($$

Consider (from *Table 13.1*)

$$Z\left[\frac{1}{s^2}\right] =\Big(\Leftarrow\frac{Tz}{(z-1)^2}$$

and, therefore,

$$Z\left[\mathrm{e}^{-sT}\,\frac{1}{s^2}\right]\Big(= z^{-1}Z\left[\frac{1}{s^2}\right] =\Big(\Leftarrow\frac{T}{(z-1)^2}$$

Thus

$$Z\left[\frac{1}{s^2}(1 - \mathrm{e}^{-Ts})\right]\Big(=\Leftarrow\frac{T}{(z-1)\Leftarrow} = G(z)\Leftarrow$$

and

$$\frac{C(z)\Leftarrow}{R(z)\Leftarrow} = \frac{G(z)\Leftarrow}{1 + G(z)\Leftarrow} =\Leftarrow\frac{T}{z + (T-1)\Leftarrow}$$

13.17 Stability

It should be appreciated from the above that, in general, $C(z)/R(z)$ results in a ratio of polynomials in z in a similar way as, for continuous systems, $C(s)/R(s)$ results in a ratio of polynomials in s. Thus, just as the equation $1 + G(s)H(s) = 0$ is called the 'characteristic equation' for the continuous system, $1 + GH(z) = 0$ is the characteristic equation for the sampled-data system. Both of these characteristic equations are polynomials in their respective variables, and the positions of the roots of these equations determine the characteristic behaviour of the corresponding closed-loop systems. Mathematically, the process of determining the roots is identical in the two cases. The difference between the two characteristic equations arises because of the need to interpret the effects of the location of the roots, when they are plotted in their respective s and z planes, on the two plants. For continuous systems, if any of these poles are located in the right-half s plane, then the system is unstable. Similarly, since the whole of the left-hand s plane maps into the unit-circle of the z plane under the transformation $z = \mathrm{e}^{sT}$, then in the simple-data case, for stability *all* of the roots of $1 + GH(z) = 0$ must lie within the unit circle.

Much of the design process of control systems is to arrange for the roots of the characteristic equation to locate at desired positions in either the s or z plane. It will be recalled, from continuous theory, that the locus of these roots, as a particular parameter is varied, may be determined by using the root-locus technique. Thus, since the characteristic equation of the sample-data system has a similar form (i.e. a polynomial), the root-locus technique may be applied to $1 + GH(z) = 0$ in exactly the same way. Only once the root-locus has been determined is there a difference in interpreting the effects of pole positions between the two cases.

13.18 Example

Consider the system shown in *Figure 13.34*, and suppose that the requirement is to draw the root-locus diagrams for, say, sampling periods of 1 and 0.5 secs.

The first requirement is to determine the pulse-transfer function for the open loop, i.e. $G(z)$:

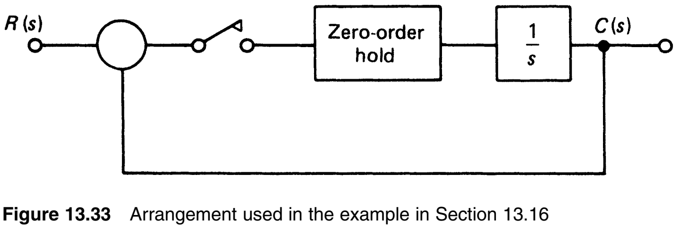


Figure 13.33 Arrangement used in the example in Section 13.16

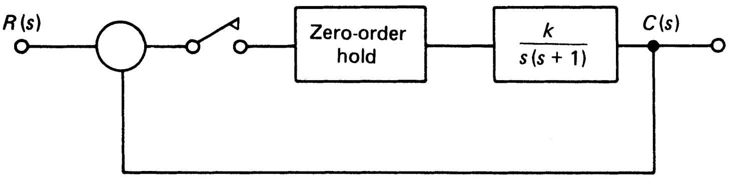


Figure 13.34 Arrangement used in the example in Section 13.18

$$G(z) = Z\left[\frac{K(1-e^{-Ts})}{s^2(s+1)}\right]$$

$$= K(1-z^{-1})Z\left[\frac{1}{s^2(s+1)}\right]$$

Consider

$$Z\left[\frac{1}{s^2(s+1)}\right] = Z\left[\frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1}\right]$$

where, from *Table 13.2*, we have

$$Z\left[\frac{1}{s^2(s+1)}\right] = \frac{T_z}{(z-1)^2} - \frac{z}{z-1} + \frac{z}{(z-e^{-T})}$$

$$= z\left[\frac{z(T+e^{-T}-1)+1-e^{-T}(1+T)}{(z-1)^2(z-e^{-T})}\right]$$

and so

$$G(z) = \frac{K[z(T+e^{-T}-1)+1-e^{-T}(1+T)]}{(z-1)(z-e^{-T})}$$

Thus, when $T = 1$ secs,

$$G_1(z) = \frac{0.368K(z+0.718)}{(z-1)(z-0.368)}$$

and when $T = 0.5$ secs

$$G_2(s) = \frac{0.107K(z+0.841)}{(z-1)(z-0.606)}$$

Both of these equations have two real poles and one zero pole, and the root loci are as shown in *Figures 13.35* and *13.36*. It can be seen that the difference in the two sampling times T causes fairly dramatic changes; when $T = 1$ secs the system becomes unstable at $K = 1.9$, and when $T = 0.5$ secs the system becomes unstable at $K = 3.9$. The process of drawing the root locus for either a continuous plant or a sampled-data plant is identical. It is the interpretation of the positions of the roots that is different, although in both cases the design is to place the roots in acceptable locations in the two planes. It is possible to use Bode diagrams in sampled-data design work and this is explained in many of the references given in the Bibliography at the end of this chapter.

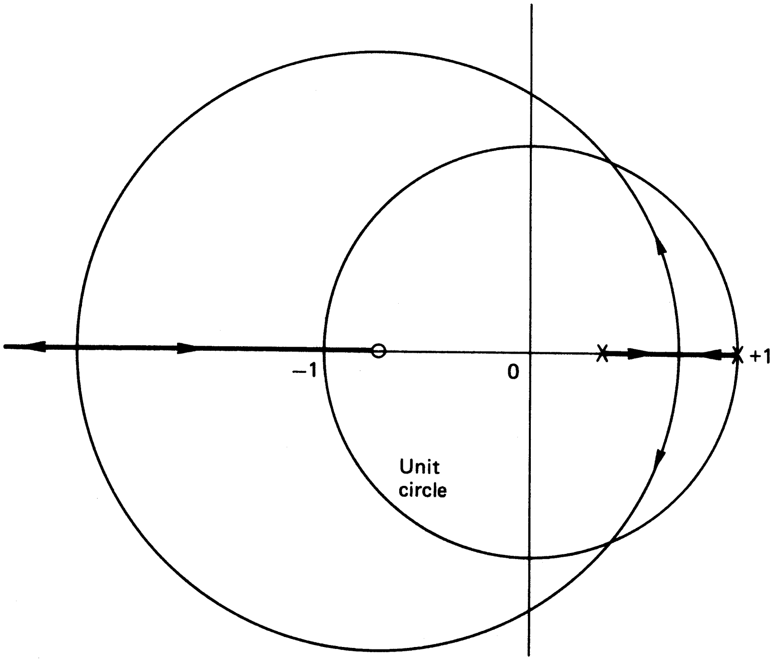


Figure 13.35 Root locus plot: $G(z) = [0.368K(z + 0.718)]/[(z - 1)(z - 0.368)]$

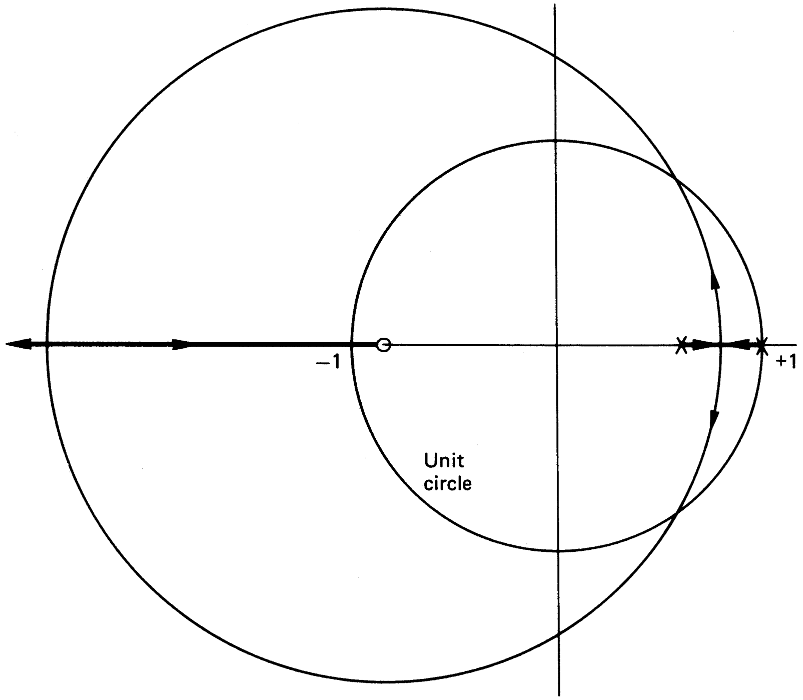


Figure 13.36 Root-locus plot: $G(z) = [0.107K(z+0.841)]/[(z-1)(z-0.606)]$

13.19 Dead-beat response

Consider the system shown above where $T = 1$ secs and $K = 1$; suppose that compensation of the form

$$D(z) = \Leftarrow\frac{1.582(z-0.368)\Leftarrow}{(z+0.418)\Leftarrow}$$

is inserted immediately after the sampler. Then it is easy to show that

$$\frac{C(z)\Leftarrow}{R(z)\Leftarrow} = \Leftarrow\frac{0.582(z+0.71)\Leftarrow}{z^2}$$

If

$$R(z) = \Leftarrow\frac{z}{z-1}$$

i.e. $r(t)$ is a unit step function, then

$$\begin{aligned} C(z) &= \Leftarrow\frac{0.582(z+0.718)\Leftarrow}{z(z-1)\Leftarrow} \\ &= \Leftarrow\frac{1}{z}\left[\frac{0.582z+0.418}{z-1}\right]\Bigg(\\ &= \Leftarrow\frac{1}{z}\left[0.582 + \Leftarrow\frac{1}{z} + \Leftarrow\frac{1}{z^2} + \cdots\Leftarrow\right]\Bigg(\end{aligned}$$

i.e. $c(0) = 0$, $c(1) = 0.582$ and $c(n) = 1$, for $n = 2, 3, \ldots$.

The implication is that $c(t)$ has reached its target position after two sample periods. If an nth order system reaches its target position in, at most, nth sampling instants, then this is called a 'dead-beat response'; a controller that achieves this, such as $D(z)$ above, is called a 'dead-beat controller' for this system. This is an interesting response, for it is not possible to achieve this with a continuous control system.

At least two dangers are inherent in dead-beat controllers:

(1) the demanded controller outputs during the process may be excessive; and
(2) there may be an oscillation set up which is not detected without further analysis.

In fact, the system is only 'dead beat' at the sampling instants. Indeed, in the above example, there is an oscillation between sampling instants of about 10% of the step value. However, theoretically it is possible for a sampled-data system to complete a transient of the above nature in finite time.

13.20 Simulation

13.20.1 System models

Regardless of the simulation language to be used, a necessary prerequisite is a description of the system of interest by a mathematical model. Some physical systems can be described in terms of models that are of the state transition type. If such a model exists, then given a value of the system variable of interest, e.g. voltage, charge position, displacement, etc., at time t, then the value of the variable (state) at some future time $t + \Delta t$ can be predicted. The prediction of the variable of interest (state variable) $x(t)$ at time $t + \Delta t$, given a state transition model S, can be expressed by the state equation:

$$x(t+\Delta t) = S[x(t), t, \Delta t] \qquad (13.5)\Leftarrow$$

Equation (13.5) shows that the future state is a function of the current state $x(t)$ at the current time t and the time increment Δt. Thus, once the model is known, from either empirical or theoretical considerations, Equation (13.5), given an initial condition (value), allows for the recursive computation of $x(t)$ for any number of future steps in time. For an initial value of the state variable $\bar{x} = x(t_1)$ at time t_1, then

$$x(t_1 + \Delta t) = S[\bar{x}, t_1, \Delta t] \Leftarrow$$

then letting $t_2 = t_1 + \Delta t$, Equation (13.5) for the next time step Δt, becomes

$$x(t_2 + \Delta t) = S[x(t_2), t_2, \Delta t] \Leftarrow$$

Obviously, this operation is continued until the calculation of the state variable has been performed for the total time period of interest.

Systems of interest will clearly not be characterised only by a single state variable but by several state variables. *Figure 13.37* is a schematic representation of a multivariable system that has r inputs, n states and m outputs.

In general, the simulation will involve calculation of all of the state variables, even though the response of only a selected number of output variables is of interest. For many systems, the output variables may well exhibit a simple one-to-one correspondence to the state variables. As shown by the representation in *Figure 13.37*, the values of the state variables depend on the inputs to the system. For a single interval, between the k and $k+1$ time instants, the state equations for the n state variable system for a change in the jth input $(j \leq r) u_j(t)$ is written as

$$\begin{aligned} x_1(t_k + \Delta t) &= S_1[x_1(t_k), u_j(t_k), t_k, \Delta t] \Leftarrow \\ x_2(t_k + \Delta t) &= S_2[x_2(t_k), u_j(t_k), t_k, \Delta t] \Leftarrow \\ \vdots \quad & \quad \vdots \\ x_n(t_k + \Delta t) &= S_n[x_n(t_k), u_j(t_k), t_k, \Delta t] \Leftarrow \end{aligned} \tag{13.6}\Leftarrow$$

The above system of equations, a collection of difference equations, would be used to predict the state variables x_1, $x_2, \ldots, x_n$ at time intervals of Δt from the initial time t_0 until the total time duration of interest $T = t_0 + K\,\Delta t$. For engineering systems, the dependent variable will generally be a continuous variable. In this case the system description will be in terms of a differential equation of the form

$$\mathrm{d}x/\mathrm{d}t = g(x, t) \tag{13.7}\Leftarrow$$

Recalling basic calculus for a small time increment, the left-hand side of Equation (13.7) can be expressed as

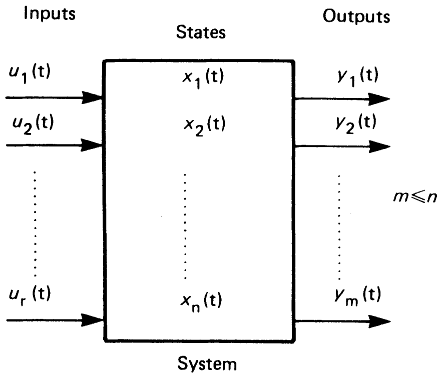


Figure 13.37 Schematic representation of a multi-variable system

$$\lim_{\Delta t \to 0} \frac{x(t + \Delta t) - x(t) \Leftarrow}{\Delta t}$$

so, for a small time increment Δt, Equation (13.7) can be written as

$$x(t + \Delta t) = x(t) + [g(x, t)]\Delta t$$

or (13.8)⇐

$$x(t + \Delta t) = G[x(t), t, \Delta t)] \Leftarrow$$

Equation (13.8) is a form of Equation (13.5), so for a small time increment, a first-order ordinary differential equation can be approximated by a state transition representation.

It thus follows from the preceding discussion that, in digital continuous system simulation, the principal numerical task is the approximate integration of Equation (13.7). For a small time increment DT, the integration step size, the computation involves the evaluation of the difference equation

$$x(t + DT) = x(t) + [g(xt)]DT) \tag{13.9a}\Leftarrow$$

which can be written explicitly as

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} \Leftarrow g[x(t_k), t_k]DT) \tag{13.9b}\Leftarrow$$

where $DT = t_{k+1} - t_k$. The calculation starts with a known value of the initial state $x(0)$ at time t_0 and proceeds successively to evaluate $x(t_1)$, $x(t_2)$, etc. The computation involves successive computation of $x(t_{k+1})$ by alternating calculation of the derivative $g[x(t_k), t_k]$ followed by integration to compute $x(t_{k+1})$ at time $t_{k+1} = t_k + DT$.

Obviously, most physical systems will be described by second or higher order ordinary differential equations so the higher order equation must be re-expressed in terms of a group of first-order ordinary differential equations by introducing state variables. For an nth order equation,

$$\frac{\mathrm{d}^n z}{\mathrm{d}t^n} = f\left[z, \frac{\mathrm{d}z}{\mathrm{d}t}, \varsigma \frac{\mathrm{d}^2 z}{\mathrm{d}t^2} \cdots \Leftarrow \frac{d^{n-1} z}{\mathrm{d}t^{n-1}}; t\right] \Bigg(\tag{13.10}\Leftarrow$$

the approach involves the introduction of new variables as state variables to yield the following first-order differential equations

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}t} &= x_2 \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} &= x_3 \\ \frac{\mathrm{d}x_3}{\mathrm{d}t} &= x_4 \\ &\vdots \\ \frac{\mathrm{d}x_{n-1}}{\mathrm{d}t} &= x_n \\ \frac{\mathrm{d}x_n}{\mathrm{d}t} &= f(x_1, x_2, x_3, \ldots, x_n; t) \Leftarrow \end{aligned} \tag{13.11}\Leftarrow$$

It should be noted that this equation can be expressed in shorthand notation as a vector-matrix differential equation. In an analogous manner, Equation (13.6) can be expressed as a vector different equation. There is no unique approach to the selection of state variables for system representation, but for many systems the choice of state variables will be obvious. In electric circuit problems, capacitor voltages and inductor currents would be logical choices, as would position, velocity and acceleration for mechanical systems.

13.20.2 Integration schemes

The simple integration step, embodied by the first-order Euler form in Equation (13.9a) only provides a satisfactory approximation of the solution of the differential equation, within specified error limits, for a very small integration step size DT. Since the small integration interval leads to substantial computing effort and to round-off error accumulation, all digital simulation languages use improved integration schemes. Despite the wide variety of different integration schemes that are available in the many different simulation languages, the calculational approach can be categorised into two groups. The types of algorithm are:

(1) *Multi-step formulae* In such algorithms, the value of $x(t+DT)$ is not calculated by the simple linear extrapolation of Equation (13.9a). Rather than use only $x(t)$ and one derivative value, the algorithms use a polynomial approximation based on past values of $x(t)$ and $g[x(t),t]$, that is at times $t-DT$, $t-2DT$, etc.

(2) *Runge–Kutta formulae* In Runge–Kutta type algorithms, the derivative value used for the calculation of $x(t+DT)$ is not the point value at time t. Instead, two or more approximate derivative values in the interval t, $t+DT$ are calculated and then a weighted average of these derivative values is used instead of a single value of the derivative to compute $x(t+DT)$.

13.20.3 Organisation of problem input

Most simulation language input is structured into three separate sections, although in some programs the statement can be used with limited sectioning of the program. A typical structure and the type of statements, functions or parts of the simulation program that appear are as follows.

(1) *Initialisation*
Problem documentation (e.g. name, date, etc.).
Initial conditions for state variables.
Parameter values (problem variables that may not be constant, problem time, integration order, integration step size, etc.).
Problem constants.

(2) *Dynamic*
Derivative statements.
Integration statements (including any control parameters not given in the initialisation section).

(3) *Terminal*
Conditional statements (e.g. total time, variable(s), value(s), etc.).
Multiple run parameters.
Output (print/plot/display) option(s).
Output format (e.g. designation of independent variable; increment for independent variable; dependent variable(s) to be output; maximum and minimum values of variable(s); or automatic scaling; total number of points for the independent variable or total length of time).

It should be understood that the specific form of the statements within each section is not exactly the same for all digital simulation languages. However, from the continuous system modelling package (CSMP) simulation programs presented in the next section, with the aid of the appropriate language manual, there should be no difficulty in formulating a simulation program using any continuous system simulation language (CSSL)-type digital simulation program.

13.20.4 Illustrative example

Simulation programs are presented, using the CSMP language, that would be suitable for investigating system dynamic behaviour. The system model, although relatively simple in nature, is typical of those used for system representation.

13.20.4.1 Example

Frequently, it will be found that system dynamic behaviour can be described by a differential equation of the form

$$\begin{aligned} y^n + a_1 y^{n-1} + a_2 y^{n-2} + a_{n-1} y^1 + a_n y \\ = b_0 r^m + b_1 r^{m-1} + b_{m-1} r^1 + b_m r \end{aligned} \qquad (13.12)\Leftarrow$$

where

$$y^n = \Leftarrow \frac{\mathrm{d}^n y}{\mathrm{d}t^n} \text{ and } r^m = \Leftarrow \frac{\mathrm{d}^m r}{\mathrm{d}t^m}$$

Use of CSMP for studying the dynamic behaviour of a system described by a high-order differential equation is illustrated here using a simulation program for the differential equation

$$y^3 + 2.5y^2 + 3.4y^1 + 0.8y = 7.3r \qquad (13.13)\Leftarrow$$

with the initial conditions

$$y^2(0) = 0; y^1(0) = -4.2; \; y^0 = 2.5$$

Development of the simulation program follows logically by rewriting Equation (13.13) as

$$\frac{\mathrm{d}^3 y}{\mathrm{d}t^3} = -2.5\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} - 3.4\frac{\mathrm{d}y}{\mathrm{d}t} - 0.8y + 7.3r \qquad (13.14)\Leftarrow$$

$$\left.\frac{\mathrm{d}^2 y}{\mathrm{d}t^2}\right|_{t=0} = \Leftarrow 0; \quad \left.\frac{\mathrm{d}y}{\mathrm{d}t}\right|_{t=0} = -4.2; \quad y|_{t=0} = 2.5$$

A block diagram showing the successive integrations to be solved for the dependent variable y is given in *Figure 13.38*. As can be seen from the labelling on the diagram, the output of the integration blocks is successive derivative values and

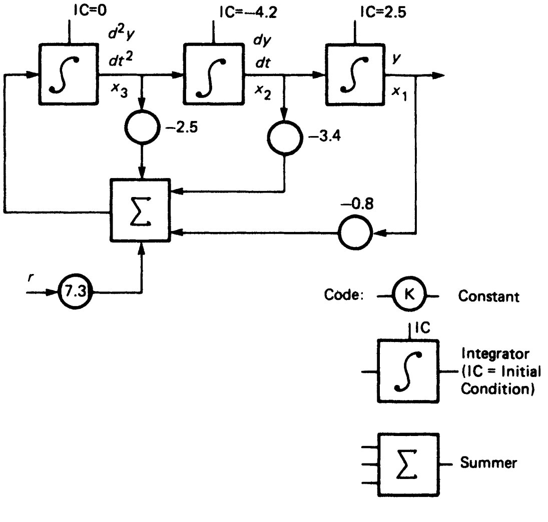
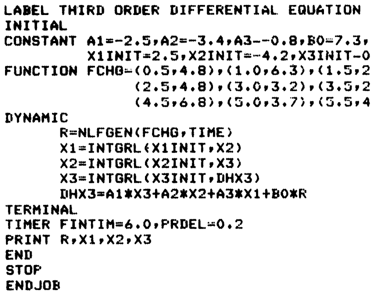


Figure 13.38 CSMP block diagram for a third-order differential equation



```
LABEL THIRD ORDER DIFFERENTIAL EQUATION
INITIAL
CONSTANT A1=-2.5,A2=-3.4,A3--0.8,B0=7.3,   ...
         X1INIT=2.5,X2INIT=-4.2,X3INIT-0.0
FUNCTION FCHG=(0.5,4.8),(1.0,6.3),(1.5,2.8),(2.0,3.9),   ...
              (2.5,4.8),(3.0,3.2),(3.5,2.1),(4.0,5.6),   ...
              (4.5,6.8),(5.0,3.7),(5.5,4.6),(6.0,3.4)
DYNAMIC
     R=NLFGEN(FCHG,TIME)
     X1=INTGRL(X1INIT,X2)
     X2=INTGRL(X2INIT,X3)
     X3=INTGRL(X3INIT,DHX3)
     DHX3=A1*X3+A2*X2+A3*X1+B0*R
TERMINAL
TIMER FINTIM=6.0,PRDEL=0.2
PRINT R,X1,X2,X3
END
STOP
ENDJOB
```

Figure 13.39 Simulation program for studying the dynamic behaviour of a system described by a third-order differential equation

the dependent variable. In fact, the output of each integration block is a state variable. This becomes obvious by introducing new variables, x_1, x_2, x_3 defined as:

$$x_1 = y$$

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = x_3$$

which allows Equation (13.14) to be expressed as

$$\frac{dx_1}{dt} = x_2 \qquad (13.15)\Leftarrow$$

$$\frac{dx_2}{dt} = x_3$$

$$\frac{dx_3}{dt} = -2.5x_3 - 3.4x_2 - 0.8x_1 + 7.3r$$

with the initial conditions

$$x_3(0) = 0; \quad x_2(0) = -4.2; \quad x_1(0) = 2.5$$

A program for solving equation (13.15) is given in *Figure 13.39*. Examination of the program shows that the value of the forcing function r is not constant but varies with time. The variation is provided using the quadratic interpolation function, NLFGEN. Total simulation time is set for 6 min with the interval for tabular output specified as 0.2 min. The time unit is determined by the problem parameters. It is to be noted that the program does not include any specification for the method of integration. The CSMP language does not require that a method of integration be given, but a particular method may be specified. If a method is not given, then by default the variable step size fourth-order Runge–Kutta method is used for calculation. The initial step size, by default, is taken as 1/16 of the PRDEL (or OUTDEL) value. Minimum step size can be limited by giving a value for DELMIN as part of the TIMER statement. If a DELMIN value is not given then, by default, the minimum step size is FINTIM × 10^{-7}.

13.21 Multivariable control

Classical process control analysis is concerned with single loops having a single setpoint, single actuator and a single controlled variable. Unfortunately, in practice, plant variables often interact, leading to interaction between control loops. A typical interaction is shown on *Figure 13.40*, where a single combustion air fan feeds several burners in a multizone furnace. An increase in air flow, via V_1 say to raise the temperature in zone 1, will lead to a reduction in the duct air pressure P_d, and a fall in air flow to the other zones. This will lead to a small fall in temperature in the other zones which will cause their temperature controllers to call for increased air flow which affects the duct air pressure again. The temperature control loops interact via the air valves and the duct air pressure.

Where interaction between variables is encountered an attempt should always be made to remove the source of the interaction, as this leads to a simpler, more robust, system. In *Figure 13.40*, for example, the interaction could be reduced significantly by adding a pressure control loop which maintains duct pressure by using a VF to set the speed of the combustion airfan. Often, however, the interaction is inherent and cannot by removed.

Figure 13.41 is a general representation of two interacting control loops. The blocks C_1 and C_2 represent the controllers comparing setpoint R with process variable V to give a controller output U. The blocks K_{ab} represent the transfer function relating variable a to controller output b. Blocks K_{11} and K_{12} are the normal forward control path, with blocks K_{21} and K_{12} representing the interaction between the loops.

The process gain of process 1 can be defined as $\Delta V_1/\Delta U_2$ where Δ denotes small change. This process gain can be measured with loop 2 in open loop (i.e. U_2 fixed) or loop 2 in closed loop control (i.e. V_2 fixed) we can thus observe two gains

$$K_{2OL} =\Leftarrow \frac{\Delta V_1}{\Delta U_1} \text{ for loop 2 open loop}$$

and

$$K_{2CL} =\Leftarrow \frac{\Delta V_1}{\Delta U_1} \text{ for loop 2 closed loop}$$

The gains will, of course vary with frequency and have magnitude and phase shift components. We can now define a relative gain $\lambda\varsigma$for loop 1

$$\lambda\varsigma=\Leftarrow\frac{K_{2OL}}{K_{2CL}}$$

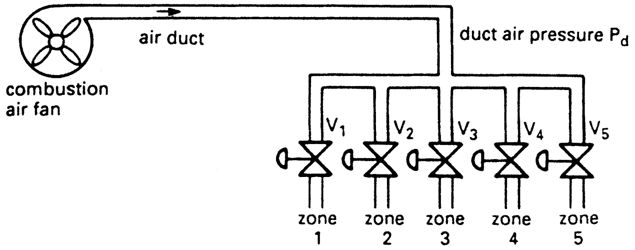


Figure 13.40 A typical example of interaction between variables in multi-variable control. The air flows interact via changes in the duct air pressure

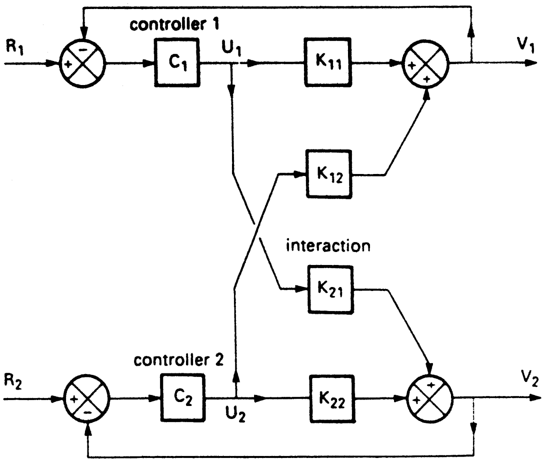


Figure 13.41 General representation of interacting loops

If λ is unity, changing from manual to auto in loop 2 does not affect loop 1, and there is no interaction between the loops.

If $\lambda < 1$, the interaction will apparently increase process 1 gain when loop 2 is switched to automatic. If $\lambda > 1$, process 1 gain will apparently be decreased when loop 2 is in automatic.

This apparent change in gain can be seen with loop 2 in manual, U_2 is fixed, so $K_{2\mathrm{OL}}$ is simply K_{11}. To find $K_{2\mathrm{CL}}$ we must consider what happens when loop 2 effectively shunts K_{11}. We have

$$V_1 = \Leftarrow K_{11} U_1 + \Leftarrow K_{12} U_2 \qquad (13.16)\Leftarrow$$

and

$$V_2 = \Leftarrow K_{22} U_2 + \Leftarrow K_{21} U_1 \qquad (13.17)\Leftarrow$$

Re-arranging Equation 13.17 gives

$$U_2 = \Leftarrow \frac{V_2 - \Leftarrow K_{21} U_1}{K_{22}}$$

which can be substituted in Equation 13.16 giving

$$V_1 = \Leftarrow K_{11} U_1 + \Leftarrow \frac{K_{12}}{K_{22}} (V_2 - \Leftarrow K_{21} U_1) \Leftarrow$$

The process 1 gain with loop 2 in auto is

$$K_{2\mathrm{CL}} = \Leftarrow \frac{\mathrm{d}V_1}{\mathrm{d}U_1}$$

$$= \Leftarrow \frac{K_{11}K_{22} - \Leftarrow K_{12}K_{21}}{K_{22}}$$

The relative gain, λ , is

$$\lambda = \Leftarrow \frac{K_{2\mathrm{OL}}}{K_{2\mathrm{CL}}}$$

$$= \frac{1}{1 - \Leftarrow K_{12}K_{21}/K_{11}K_{22}}$$

It should be remembered that the gains K_{ab} are dynamic functions, so λ will vary with frequency.

The term $(K_{12}K_{21}/K_{11}K_{22}$) is the ratio between the interaction and forward gains. This should be in the range 0 to 1. If the term is greater than unity, the interactions have more effect than the supposed process, and the process variables are being manipulated by the wrong actuators!

It is possible to determine the range of λ from the relationship $(K_{12}K_{21}/K_{11}K_{22})$. If this is positive, λ will be greater than unity, and loop 1 process gain will decrease when loop 2 is switched to auto. This will occur if there is an even number of K_{ab} blocks with negative sign (0, 2 or 4). If the relationship is negative, λ will be less than unity and loop 1 process gain will increase when loop 2 is closed. This occurs if there is an odd number of blocks with negative sign (1 or 3).

The combustion air flow system of *Figure 13.40* is redrawn on *Figure 13.42(a)*. Increasing U_1 obviously decreases V_2, and increasing U_2 similarly decreases V_1. The interaction block diagram thus has the signs of *Figure 13.42(b)*. There are two negative blocks, so λçis greater than unity.

If λçis greater than unity, the interaction can be considered benign as the reduced process gain will tend to increase the loop stability (albeit at the expense of response time). The loops can be tuned individually in the knowledge that they will remain stable with all loops in automatic control.

If λ is in the range $0 < \lambda < 1$, care must be taken as it is possible for loops to be individually stable but collectively unstable requiring a reduction in controller gains to maintain stability. The closer λ gets to zero, the greater the interaction and the more de-gaining will be required.

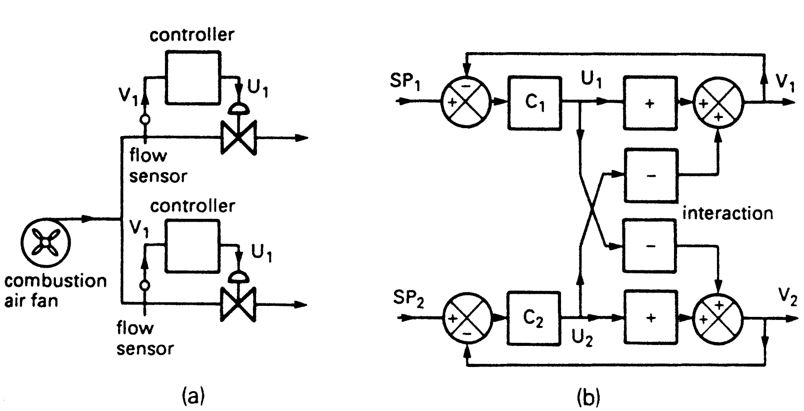


Figure 13.42 The combustion air system redrawn to show interactions: (a) block diagram; (b) interaction diagram, with two negative blocks the interaction decreases the apparent process gain and the interaction is benign

The calculation of dynamic interaction is difficult, even for the two variable case.With more interacting variables, the analysis becomes exceedingly complex and computer solutions are best used. Ideally, though, interactions once identified, should be removed wherever possible.

13.22 Dealing with non-linear elements

13.22.1 Introduction

All systems are non linear to some degree. Valves have non linear transfer functions, actuators often have a limited velocity of travel and saturation is possible in every component. A controller output is limited to the range 4–20 mA, say, and a transducer has only a restricted measurement range.

One of the beneficial effects of closed loop control is the reduced effect of non linearities. The majority of non linearities are therefore simply lived with, and their effect on system performance is negligible. Occasionally, however, a non linear element can dominate a system and in these cases its effect must be studied.

Some non linear elements can be linearised with a suitable compensation circuit. Differential pressure flow meters have an output which is proportional to the square of flow. Following a non linear differential pressure flow transducer with a non linear square root extractor gives a linear flow measurement system.

Cascade control can also be used around a non linear element to linearise its performance as seen by the outer loop. Butterfly valves are notoriously non linear. They have an S shaped flow/position characteristic, suffer from backlash in the linkages and are often severely velocity limited. Enclosing a butterfly valve within a cascade flow loop, for example, will make the severely non linear flow control valve appear as a simple linear first order lag to the rest of the system.

There are two basic methods of analysing the behaviour of systems with non linear elements. It is also possible, of course, to write computer simulation programs and often this is the only practical way of analysing complex non linearities.

13.22.2 The describing function

If a non linear element is driven by a sine wave, its output will probably not be sinusoidal, but it will be periodic with the same frequency as the input, but of differing shape and possibly shifted in phase as shown on *Figure 13.43*. Often the shape and phase shift are related to the amplitude of the driving signal.

Fourier analysis is a technique that allows the frequency spectrum of any periodic waveform to be calculated. A simple pulse can be considered to be composed of an infinite number of sine waves.

The non linear output signals of *Figure 13.43* could therefore be represented as a frequency spectrum, obtained from Fourier analysis. This is, however, unnecessarily complicated. Process control is generally concerned with only dominant effects, and as such it is only necessary to consider the fundamental of the spectrum. We can therefore represent a non linear function by its gain and phase shift at the fundamental frequency. This is known as the *describing function*, and will probably be frequency and amplitude dependant.

Figure 13.44 shows a very crude bang/bang servo system used to control level in a header tank. The level is sensed by a capacitive probe which energises a relay when a nominal depth of probe is submerged. The relay energises a solenoid which applies pneumatic pressure to open a flow valve. This system is represented by *Figure 13.45*.

The level sensor can be considered to be a level transducer giving a 0–10 V signal over a 0.3 m range. The signal is filtered with a 2 sec time constant to overcome noise from splashing, ripples etc. The level transducer output is compared with the voltage from a setpoint control and the error signal energises or de-energises the relay. We shall assume no hysteresis for simplicity although this obviously would be desirable in a real system.

The relay drives a solenoid assumed to have a small delay in operation which applies 15 psi to an instrument air pipe to open the valve. The pneumatic signal takes a finite time to travel down the pipe, so the solenoid valve and piping are considered as a 0.5 sec transit delay. The valve actuator turns on a flow of 150 m^3/min for an applied pressure of 15 psi. We shall assume it is linear for other applied pressures. The actuator/valve along with the inertia of the water in the pipe appear as a first order lag of 4 sec time constant. The tank itself appears as an integrator from flow to level.

This system is dominated by the non linear nature of the level probe and the solenoid. The rest of the system can be considered linear if we combine the level comparator, relay and solenoid into a single element which switches 0 to 15 psi according to the sign of the error signal (15 psi for negative error, i.e. low level).

This non linear element will therefore have the response of *Figure 13.46*. when driven with a sinusoidal error signal. The output will have a peak to peak amplitude of 15 psi regardless of the error magnitude.

From Fourier analysis, the fundamental component of the output signal is a sine wave with amplitude $4 \times 7.5/\pi$ psi as shown. The phase shift is zero at all frequencies. The non linear element of the comparator/relay/solenoid can thus be considered as an amplifier whose gain varies with the amplitude of the input signal.

For a 1 V amplitude error signal the gain is

$$(4 \times 7.5)/(\pi \times 1) = 9.55$$

For a 2 V amplitude error signal the gain is

$$(4 \times 7.5)/(\pi \times 2) = 4.78$$

In general, for an E volt error signal the gain is

$$(4 \times 7.5)/(\pi \times E) = 9.55/E \qquad (13.18)$$

Figure 13.47 is a Nichols chart for the linear parts of the system. This has 180° phase shift for $\omega = 0.3$ rads/sec, so if it was controlled by a proportional controller, it would oscillate at 0.3 rads/sec if the controller gain was sufficiently high. The linear system gain at this frequency is −7dB, so a proportional controller gain of 7dB would just sustain continuous oscillation.

Let us now return to our non linear level switch. This has a gain which varies inversely with error amplitude. If we are, for some reason, experiencing a large sinusoidal error signal the gain will be low. If we have a small sinusoidal error signal the gain will be high.

Intuitively we know the system of *Figure 13.44* will oscillate. The non linear element will add just sufficient gain to make the Nichols chart of *Figure 13.47* pass through the 0 dB/−180° origin. Self sustaining oscillations will result at 0.3 rads/sec. If these increase in amplitude for some reason, the gain will decrease causing them to decay again. If they cease, the gain will increase until oscillations recommence.

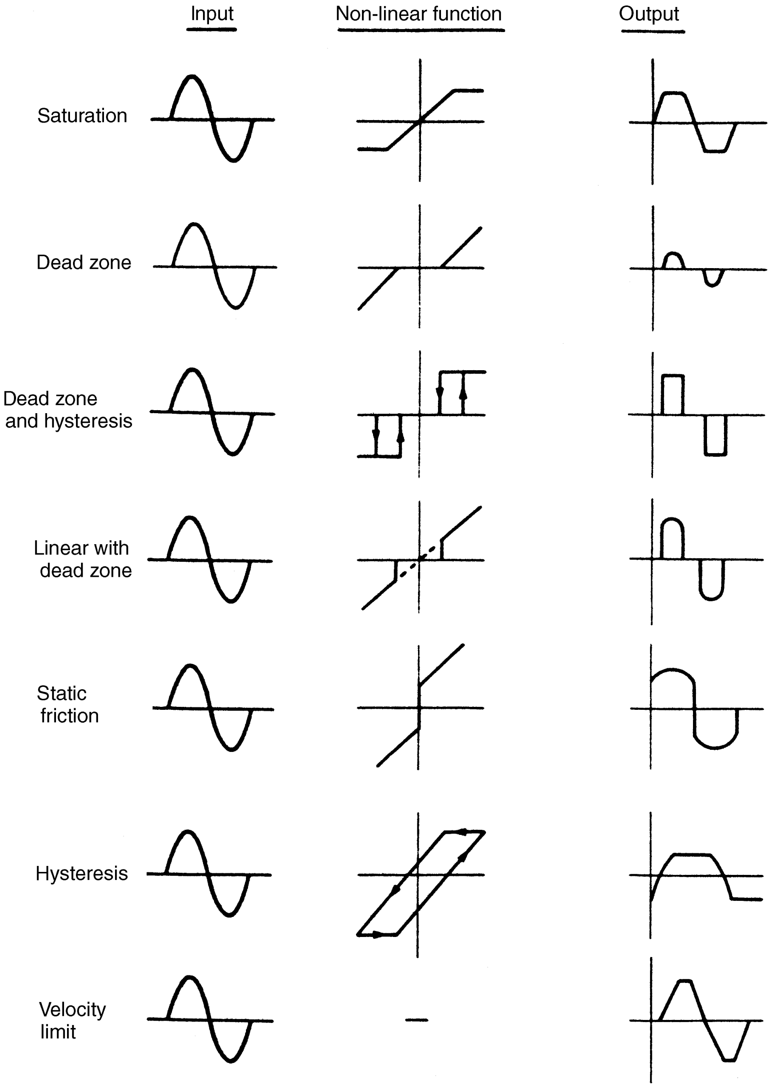


Figure 13.43 Common non-linearities

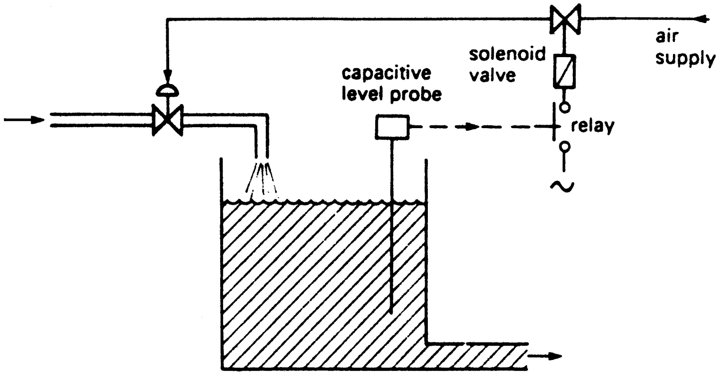


Figure 13.44 Bang-bang level control system

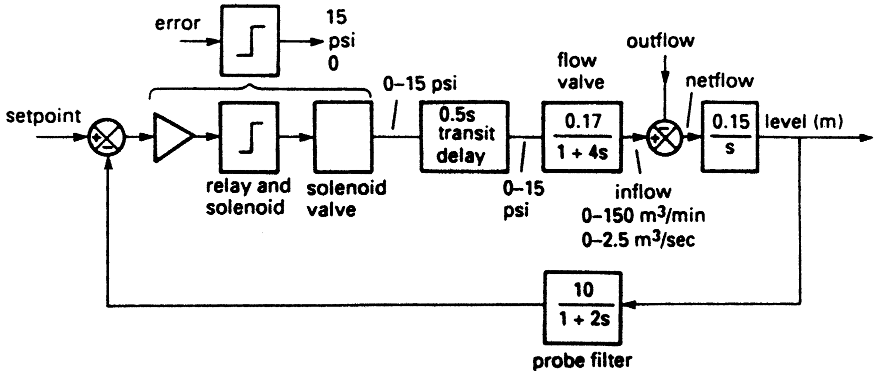


Figure 13.45 Block diagram of bang-bang level control system

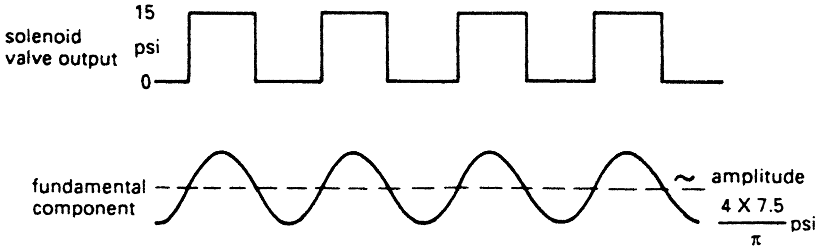


Figure 13.46 Action of solenoid valve in level control system

The system stabilises with continuous constant amplitude oscillation.

To achieve this the non linear element must contribute 7 dB gain, or a linear gain of 2.24. From Equation 13.18 above, the gain is 9.55/*E* where *E* is the error amplitude. The required gain is thus given by an error amplitude of 9.55/2.24 = 4.26 V. This corresponds to an oscillation in level of 0.426 m.

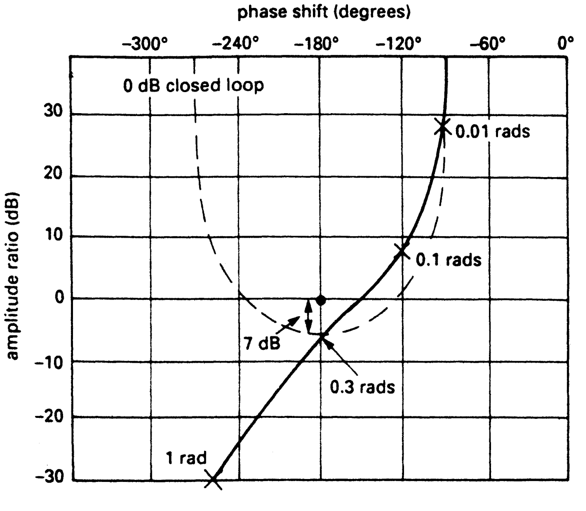


Figure 13.47 Nichols chart for linear portion of level control system

The system will thus oscillate about the set level with an amplitude of 0.4 m (the assumptions and approximations give more significant figures a relevance they do not merit) and an angular frequency of 0.4 rads/sec (period fractionally over 20 s).

There is a hidden assumption in the above analysis that the outgoing flow is exactly half the available ingoing flow to give equal mark/space ratio at the valve. Other flow rates will give responses similar to *Figure 13.48*, exhibiting a form of pulse width modulation. The relatively simple analysis however has told us that our level control system will sustain constant oscillation with an amplitude of around half a metre and a period of about 20 sec at nominal flow.

Similar techniques can be applied to other non linearities; a limiter, shown on *Figure 13.49*(*a*) and (*b*), for example, will have unity gain for input amplitudes less than the limiting level. For increasing amplitude the apparent gain will decrease. The describing function when limiting occurs has a gain dependent on the ratio between the input signal amplitude and the limiting value as plotted on *Figure 13.49*(*c*). There is no phase shift between input and output.

Hysteresis, shown on *Figure 13.50*, introduces a phase shift, and a flat top to the output waveform. This is not the same waveform as the limiter; the top is simply levelled off at 2*a* below the peak where *a* is half the dead zone width. If the input amplitude is large compared to the dead zone, the gain is unity and the phase shift can be approximated by

$\phi\varsigma = \sin^{-1}(a/V_{\mathrm{i}})\Leftarrow$

As the input amplitude decreases, the gain increases and becomes zero when the input peak to peak amplitude is

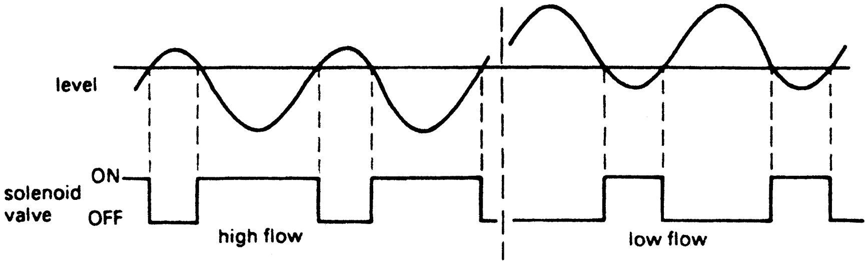


Figure 13.48 Response of level control system to changes in flow

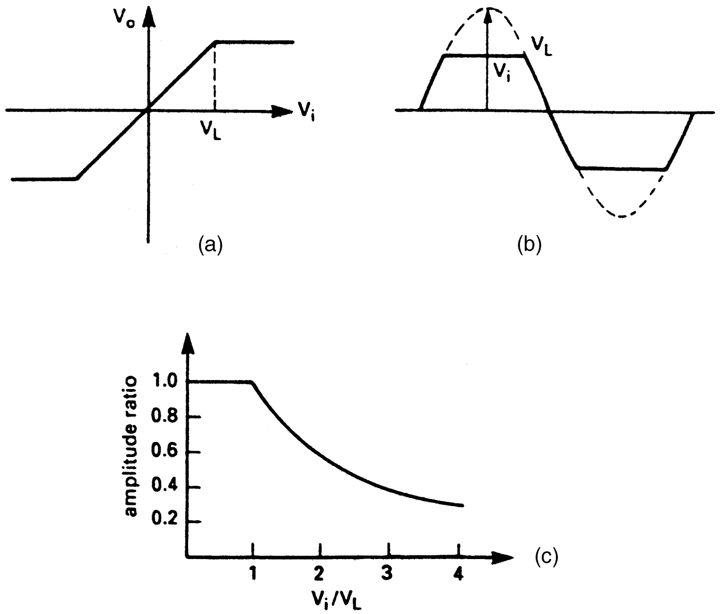


Figure 13.49 The limiter circuit: (a) relationship between input and output; (b) effect of limiting on a sine wave input signal; (c) 'Gain' of a limiter related to the input signal amplitude

less than the dead zone width. The exact relationship is complex, but is shown on *Figure 13.50*(*c*) and (*d*).

Non linear elements generally have gains and phase shift which increase or decrease with input amplitude (usually a representation of the error signal). *Figure 13.51* illustrates the two gain cases. For a loop gain of unity, constant oscillations will result. For loop gains greater than unity, oscillations will increase in amplitude, for loop gain less than unity oscillations will decay.

In *Figure 13.51*(*a*), the gain falls off with increasing amplitude. The system thus tends to approach point X as large oscillations will decay and small oscillations increase. The system will oscillate at whatever gain gives unity loop gain. This is called *limit cycling*. Most non linearities (bang-bang servo, saturation etc.) are of this form.

Where loop gain increases with amplitude as *Figure 13.51*(*b*), decreasing gain gives increasing damping as the amplitude decreases, so oscillations will quickly die away. This response is sometimes deliberately introduced into level controls. If, however, the system is provoked beyond Y by a disturbance, the oscillation will rapidly increase in amplitude and control will be lost.

13.22.3 State space and the phase plane

Figure 13.52(*a*) shows a simple position control system. The position is sensed by a potentiometer, and compared with a setpoint from potentiometer RV_1. The resulting error signal is compared with an error 'window' by comparators C_1, C_2. Preset RV_2 sets the deadband, i.e. the width of the window. The comparators energise relays RL_F and RL_R which drive the load to the forward and reverse respectively.

Initially, we shall analyse the system with RV_2 set to zero, i.e. no deadband. This has the block diagram of *Figure 13.52*(*b*), with a first order lag of time constant T arising from the inertia of the system, and the integral action converting motor velocity to load position.

The system is thus represented by

$$x =\Leftarrow \frac{\pm K}{s(1 + \Leftarrow sT)\Leftarrow} \qquad (13.19)\Leftarrow$$

where K represents the acceleration resulting from the motor torque and inertia with the sign of K indicating the sign of the error. This has the solution

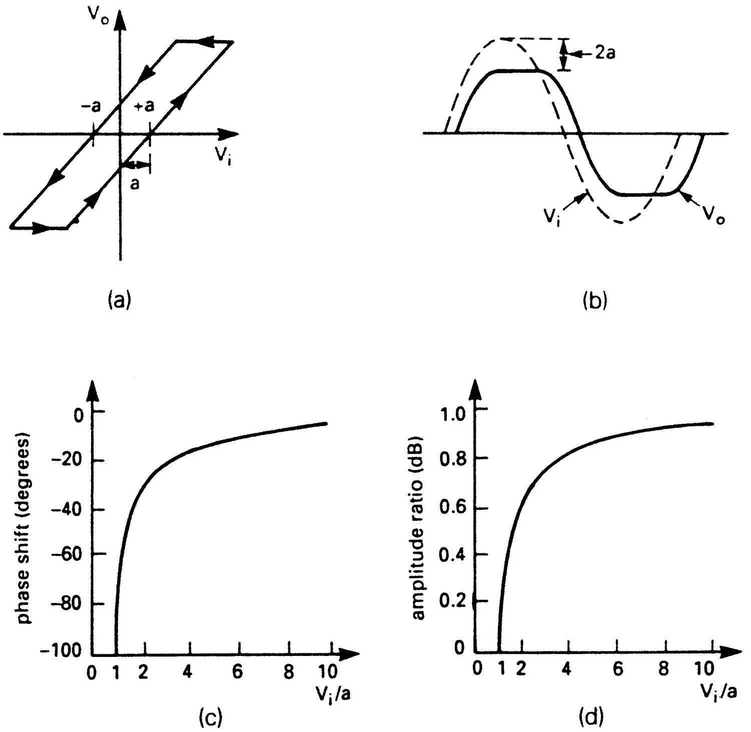


Figure 13.50 The effect of hysteresis: (a) relationship between input and output signals; (b) the effect of hysteresis on a sine wave input signal; (c) the relationship between phase shift and signal amplitude; (d) the relationship between 'gain' and signal amplitude

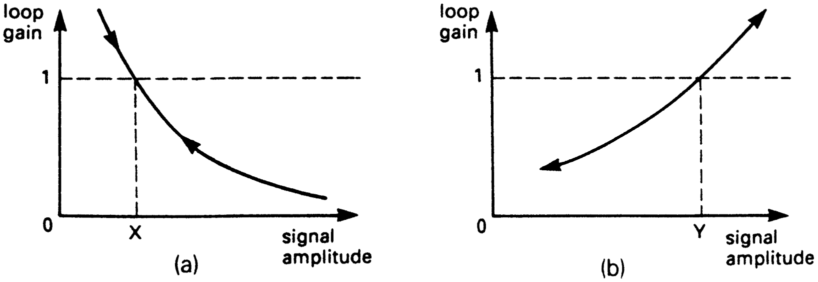


Figure 13.51 Possible relationships between gain and signal amplitude: (a) gain decreases with increasing amplitude; (b) gain increases with increasing amplitude

$$x = x_0 - \mathrm{T}K + \mathrm{T}V_0 + Kt + \mathrm{T}(K - V_0)e^{-t/\mathrm{T}} \qquad (13.20)\Leftarrow$$

where x_0 and V_0 respectively represent the initial position and velocity.

Differentiating gives the velocity, V

$$V = K - (K - V_0)e^{-t/\mathrm{T}} \qquad (13.21)\Leftarrow$$

Equations 13.20 and 13.21 fully describe the behaviour of the system. These can be plotted graphically as *Figure 13.53* with velocity plotted against position for positive K for various times from $t = 0$. Each curve represents a different starting condition; curve D, for example, starts at $x_0 = -5$ and $v_0 = -2$

In each case, the curve ends towards $v = 2$ units/sec as t gets large. The family of curves have an identical shape, and the different starting conditions simply represent a horizontal shift of the curve.

A similar family of curves can be drawn for negative values of K. These are sketched on *Figure 13.54*. In this case, the velocity tends towards $V = -2$ units/sec.

Given these curves, we can plot the response of the system. Let us assume that the system is stationary at $x = -5$, and the setpoint is switched to +5. The subsequent behaviour is shown on *Figure 13.55*. The system starts by initially following the curve passing through $x = -5$, $V = 0$ for positive K, crossing $x = 0$ with a velocity 1.5 units/sec, reaching the setpoint at point X with a velocity of 1.76 units/sec. It cannot stop instantly however, so it overshoots.

At the instant the overshoot occurs K switches sign. The system now has a velocity of +1.76, with K negative, so it follows the corresponding curve of *Figure 13.54* from point X to point Y. It can be seen that an overshoot to $x = 7$ occurs. At point Y, another overshoot occurs and K switches

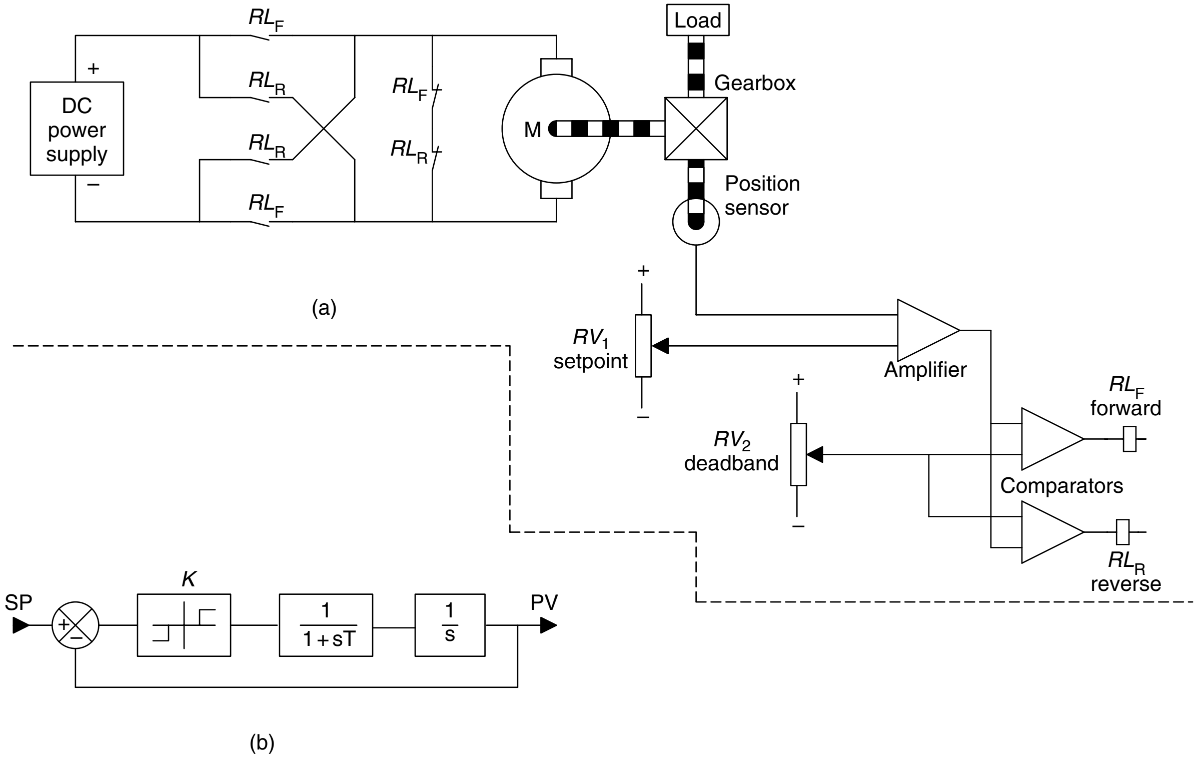


Figure 13.52 A non-linear position control system: (a) system diagram; (b) block representation

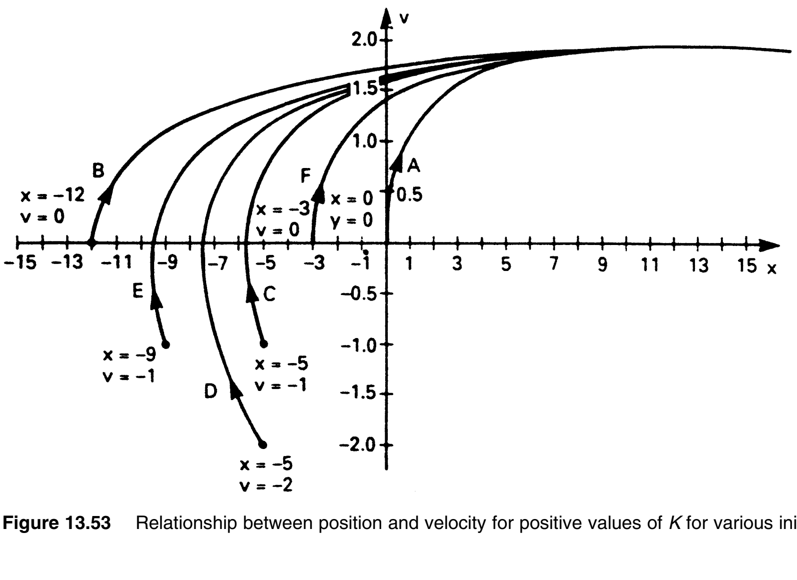


Figure 13.53 Relationship between position and velocity for positive values of *K* for various initial conditions

back positive. The system now follows the curve to Z with an undershoot of $x=$~~4.~~1. At Z another overshoot occurs and the system spirals inwards as shown. The predicted step response is shown on *Figure 13.56*.

In *Figure 13.57*(*a*), the deadband control (RV_2 on the earlier *Figure 13.52*) has been adjusted to energise RL_F for error voltages more negative than -1 unit and energise RL_R for error voltages above $+1$ unit. There is thus a deadband 2 units wide around the setpoint.

Figure 13.57(*b*) shows the effect of this deadband. We will assume initial values of $x_0=$~~0,~~ $v_0=$~~0~~when we switch the setpoint to $x=$~~5.~~ The system accelerates to point *U* ($x=$~~4,~~ $v=$~~1.~~40) at which point RL_1 de-energises. The system loses speed ($K=$~~0)~~ until point *V*, where the position

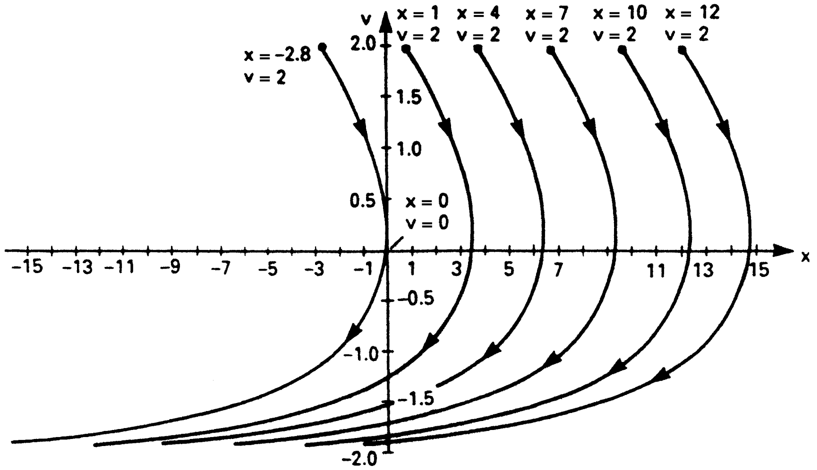


Figure 13.54 Relationship between position and velocity for negative values of *K* for various initial conditions

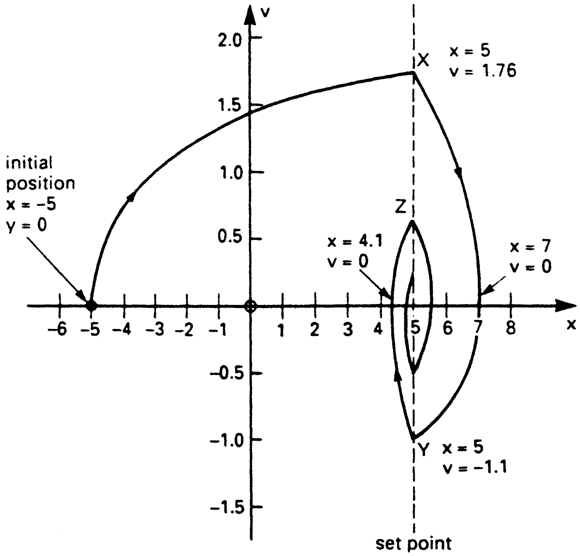


Figure 13.55 System behaviour following change of setpoint from $x = -5$ to $x = +5$

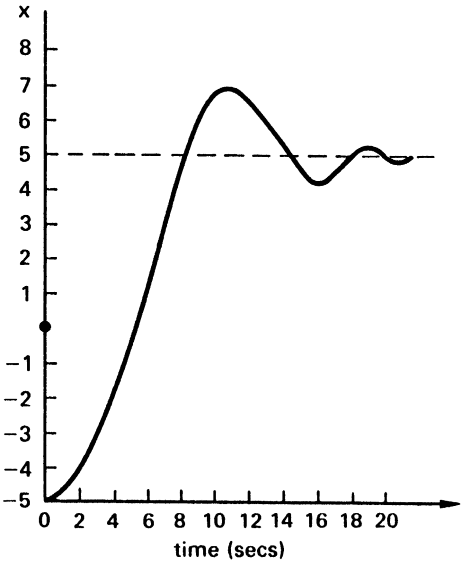


Figure 13.56 Predicted step response following change of setpoint

passes out of the deadband and RL_R energises. The system reverses, and re-enters the deadband at point *W*, where RL_F de-energises. An undershoot then occurs (*X* to *Y*) where the deadband is entered for the last time, coming to rest at point *Z* ($x = 4.75$, $v = 0$).

The position x and velocity dx/dt completely describe the system and are known as *state variables*. A linear system can be represented as a set of first order differential equations relating the various state variables. For a second order system there are two state variables, for higher order systems there will be more.

For the system described by Equation 13.19, we can denote the state variables by x (position) and v (velocity). For a driving function *K*, we can represent the system by *Figure 13.58* which is called a *state space model*. This describes the position control system by the two first order differential equations.

$$T\frac{dv}{dt} = K - x$$

and

$$v = \Leftarrow \frac{dx}{dt}$$

Figure 13.56 and *Figure 13.57* plot velocity against position, and as such are plots relating state variables. For two state variables (from a second order system) the plot is known as a *phase plane*. For higher order systems, a multi-dimensional plot, called *state space*, is required. Plots such as *Figure 13.53* and *Figure 13.54* which show a family of possible curves are called *phase plane portraits*.

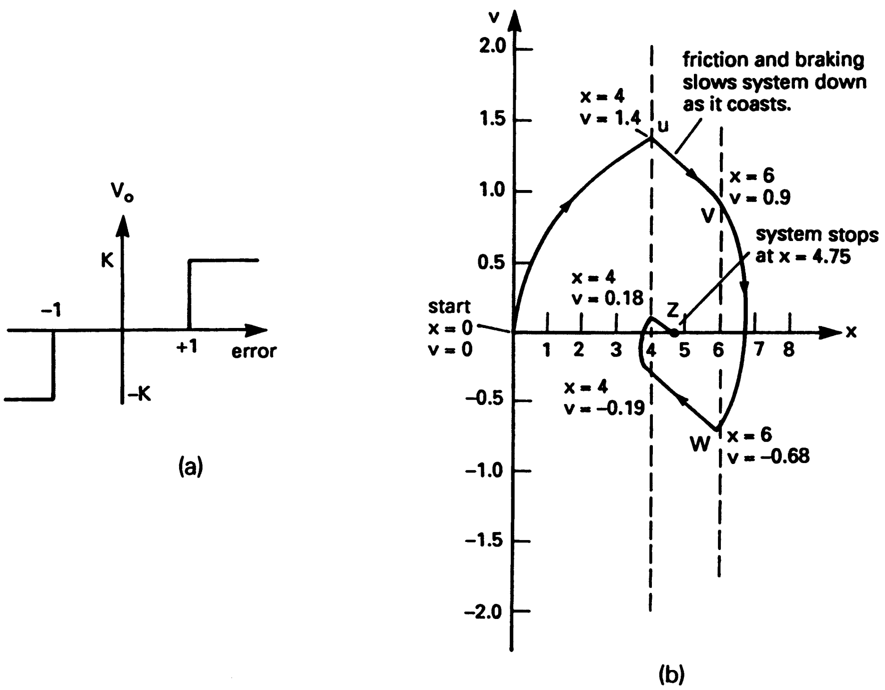


Figure 13.57 System with deadband and friction: (a) deadband response; (b) position/velocity curve for setpoint change from $x = 0$ to $x = 5$. Note system does not attain the setpoint

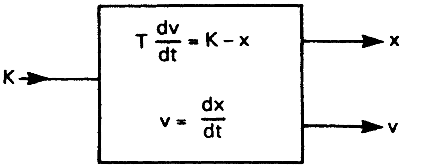


Figure 13.58 State variables for position control system

Similar phase planes can be drawn for other non linearities such as saturation, hysteresis etc. Various patterns emerge, which are summarised on *Figure 13.59*. The system behaviour can be deduced from the shape of the phase trajectory.

In a linear closed loop system stability is generally increased by adding derivative action. In a position control system this is equivalent to adding velocity (dx/dt) feedback. The behaviour of a non linear system can also be improved by velocity feedback. In *Figure 13.60*(*a*) velocity feedback has been added to our simple Bang/Bang position servo.

The switching point now occurs where

$$S_P - x - Lv = 0$$

or

$$v = \frac{1}{L}(S_P - x)\Leftarrow$$

This is a straight line of slope $-1/L$, passing through $x = S_P$, $v = 0$ on the phase plane. Note that L has the units of time. The line is called the switching line, and advances the changeover as shown on *Figure 13.60*(*b*), thereby reducing the overshoot. Too much velocity feedback as *Figure 13.60*(*c*) simulates an overdamped system as the trajectory runs to the setpoint down the switching line.

13.23 Disturbances

13.23.1 Introduction

A closed loop control system has to deal with the malign effects of outside disturbances. A level control system, for example, has to handle varying throughput, or a gas fired furnace may have to cope with changes in gas supply pressure. Although disturbances can enter a plant at any point, it is usual to consider disturbances at two points; supply disturbances at the input to the plant and load/demand disturbances at the point of measurement as shown on *Figure 13.61*(*a*).

The closed loop block diagram can be modified to include disturbances as shown on *Figure 13.61*(*b*). A similar block diagram could be drawn for load disturbances or disturbances entering at any point by subdividing the plant block. By normal analysis we have

$$V = \frac{CPS_p}{1 + HCP} + \frac{PD}{1 + HCP} \qquad (13.22)\Leftarrow$$

Equation 13.22 has two components; the first relates the plant output to the setpoint and is the normal closed loop transfer function $GH/(1 + GH)$. The product of controller and plant transfer function $C.P.$ is the forward gain G. The second term relates the performance of the plant to disturbance signals. In general, closed loop control reduces the effect of disturbances. If the plant was run open loop, the effect of the disturbances on the output would be simply

$$V = PD$$

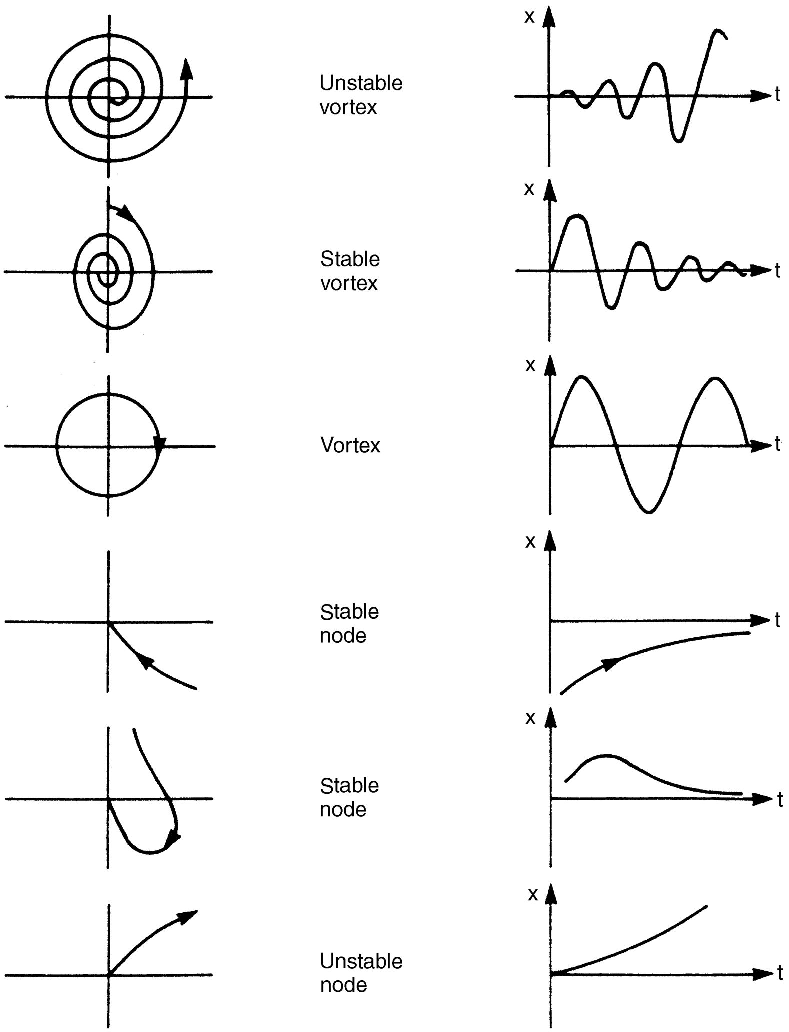


Figure 13.59 Various possible trajectories and their response

From Equation 13.22 with closed loop control, the effect of the disturbance is

$$V =\Leftarrow \frac{PD}{1 +\Leftarrow HCP}$$

i.e. it is reduced providing the magnitude of $(1 + HCP)$ is greater than unity. If the magnitude of $(1 + HCP)$ becomes less than unity over some range of frequencies, closed loop control will magnify the effect of disturbances in that frequency range. It is important, therefore, to have some knowledge of the frequency spectra of expected disturbances.

13.23.2 Cascade control

Closed loop control gives increased performance over open loop control, so it would seem logical to expect benefits from adding an inner control loop around plant items that are degrading overall performance. *Figure 13.62* shows a typical example, here the output of the outer loop controller becomes the setpoint for the inner controller. Any problems in the inner loop (disturbances, non linearities, phase lag etc.) will be handled by the inner controller, thereby improving the overall performance of the outer loop. This arrangement is known as *cascade control.*

To apply cascade control, there must obviously be some intermediate variable that can be measured (P_{Vi} on *Figure 13.62*) and some actuation point that can be used to control it.

Cascade control brings several benefits. The secondary controller will deal with disturbances before they can affect the outer loop. Phase shift within the inner loop is reduced, leading to increased stability and speed of response in the outer loop. Devices with inherent integral action (such as a motorised valve) introduce an inherent $-90°$⇐integrator phase lag. This can be removed by adding a valve positioner in cascade. Cascade control will also reduce the effect of non linearities (e.g. non linear gain, backlash etc.) in the inner loop.

There are a few precautions that need to be taken, however. The analysis so far ignores the fact that components saturate and stability problems can arise when the inner

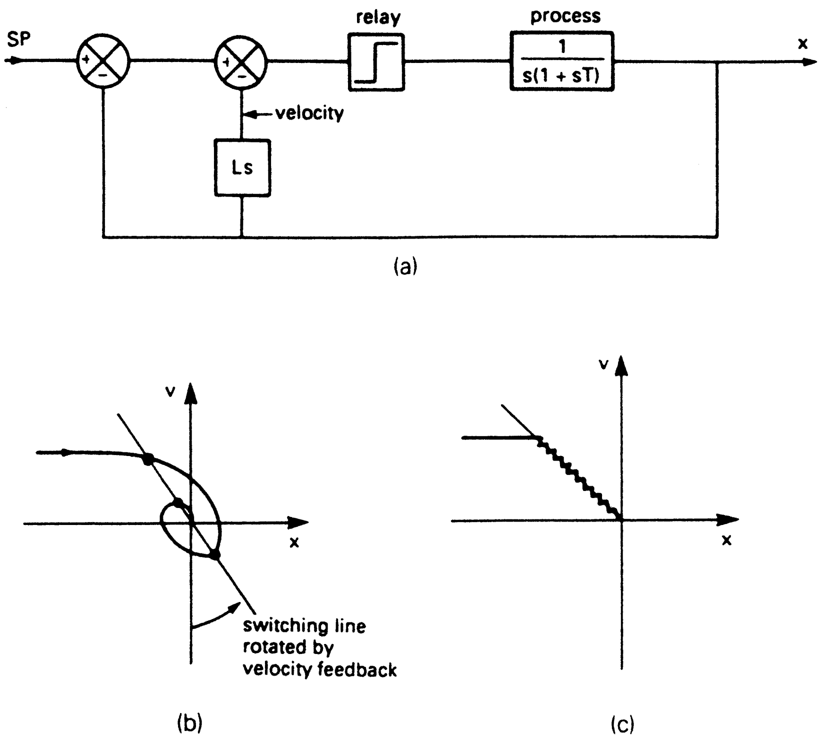


Figure 13.60 Addition of velocity feedback to a non-linear system: (a) block diagram of velocity feedback; (b) system behaviour on velocity/position curve; (c) overdamped system follows the switching line

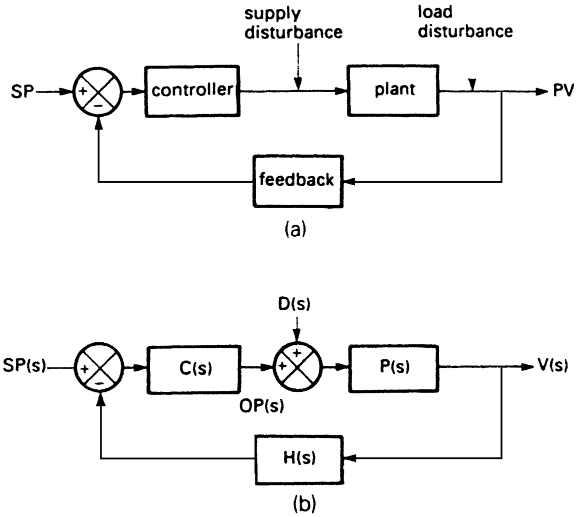


Figure 13.61 The effect of disturbances: (a) points of entry for disturbances; (b) block diagram of disturbances

loop saturates. This can be overcome by limiting the demands that the outer loop controller can place on the inner loop (i.e. ensuring the outer loop controller saturates first) or by providing a signal from the inner to the outer controller which inhibits the outer integral term when the inner loop is saturated.

The application of cascade control requires an intermediate variable and control action point, and should include, if possible, the plant item with the shortest time constant. In general, high gain proportional only control will often suffice for the inner loop, any offset is of little concern as it will be removed by the outer controller. For stability, the inner loop must always be faster than the outer loop.

Tuning a system with cascade control requires a methodical approach. The inner loop must be tuned first with the outer loop steady in manual control. Once the inner loop is tuned satisfactorily, the outer loop can be tuned as normal. A cascade system, once tuned, should be observed to ensure that the inner loop does not saturate, which can lead to instability or excessive overshoot on the outer loop. If saturation is observed, limits must be placed on the output of the outer loop controller, or a signal provided to prevent integral windup as described later in Section 13.27.6

13.23.3 Feedforward

Cascade control can reduce the effect of disturbances occurring early in the forward loop, but generally cannot deal with load/demand disturbances which occur close to, or affect directly the process variable as there is no intermediate variable or accessible control point.

Disturbances directly affecting the process variable must produce an error before the controller can react. Inevitably, therefore, the output signal will suffer, with the speed of recovery being determined by the loop response. Plants which are difficult to control tend to have low gains and long integral times for stability, and hence have a slow response. Such plants are prone to error from disturbances.

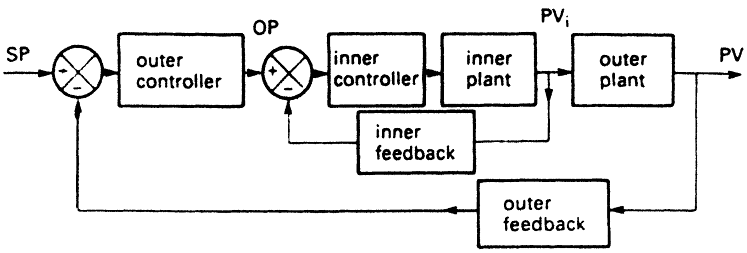


Figure 13.62 A system with cascade control

In general a closed loop system can be considered to behave as a second order system, with a natural frequency ω_n, and a damping factor. At frequencies above ω_n, the closed loop gain falls off rapidly (at 12 dB/octave). Disturbances occurring at a frequency much above $2\omega_n$ will be uncorrected. If the closed loop damping factor is less the unity (representing an underdamped system), the effect of disturbances with frequency components around ω_n can be magnified.

Figure 13.63(*a*) shows a system being affected by a disturbance. Cascade control cannot be applied because there is no intermediate variable between the point of entry and the process variable. If the disturbance can be measured, and its effect known, (even approximately), a correcting signal can be added to the controller output signal to compensate for the disturbance as shown on *Figure 13.63*(*b*). This is known as *feedforward* control.

This correcting signal, arriving by blocks H, F, and P_1 should ideally exactly cancel the original disturbance, both in the steady state and dynamically under changing conditions. The transfer functions of the transducer H and plant P_1 are fixed, with F a compensator block designed to match H and P_1.

In general, the compensator block transfer function will be

$$F = -\frac{1}{HP_1}$$

If the plant acts as a simple lag with time constant T (i.e. $1/(1+s\mathrm{T})$), the compensator will be a simple lead $(1+s\mathrm{T})$. In many cases a general purpose compensator $(1+s\mathrm{T}_a)/(1+s\mathrm{T}_b)$ is used.

The feedforward compensation does not have to match exactly the plant characteristics; even a rough model will give a significant improvement (although a perfect model will give perfect control). In most cases a simple compensator will suffice.

Cascade control can usually deal with supply disturbances and feedforward with load or demand disturbances. These neatly complement each other so it is very common to find a system where feedforward modifies the setpoint for the inner cascade loop.

13.24 Ratio control

13.24.1 Introduction

It is a common requirement for two flows to be kept in precise ratio to each other; gas/oil and air in combustion control, or reagants being fed to a chemical reactor are typical examples.

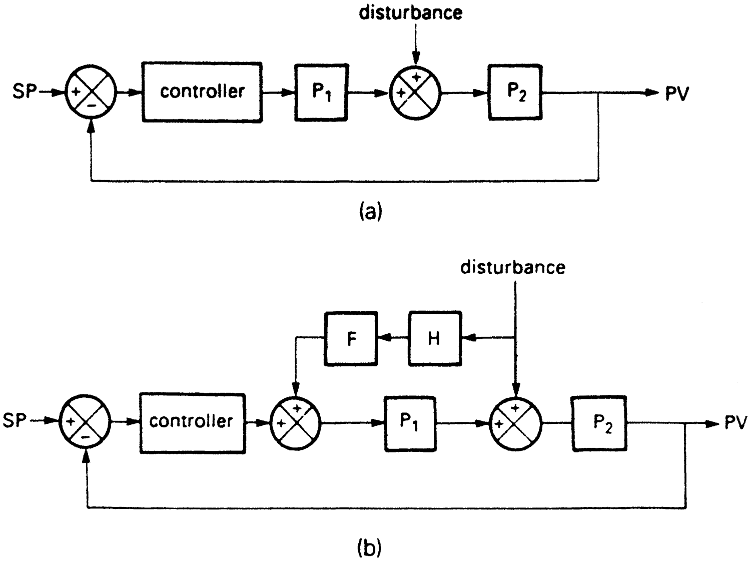


Figure 13.63 Effect of a disturbance reduced by feedforward: (a) a system to which cascade control cannot be applied being subject to a disturbance; (b) correcting signal derived by measuring the disturbance

13.24.2 Slave follow master

In simple ratio control, one flow is declared to be the master. This flow is set to meet higher level requirements such as plant throughput or furnace temperature. The second flow is a slave and is manipulated to maintain the set flow ratio.

The controlled variable here is ratio, not flow, so an intuitive solution might look similar to *Figure 13.64* where the actual ratio *A*/*B* is calculated by a divider module and used as the process variable for a controller which manipulated the slave control valve.

This scheme has a hidden problem. The slave loop includes the divider module and hence the term *A*. The loop gain varies directly with the flow *A*, leading to a sluggish response at low flows and possible instability at high flow. If the inverse ratio *B*/*A* is used as the controller variable the saturation becomes worse as the term 1/*A* now appears in the slave loop giving a loop gain which varies inversely with A, becoming very high at low flows. Any system based on *Figure 13.64* would be impossible to tune for anything other than constant flow rates.

Ratio control systems are often based on *Figure 13.65*. The master flow is multiplied by the ratio to produce the setpoint for the slave flow controller. The slave flow thus follows the master flow. Note that in the event of failure in the master loop (a jammed valve for example) the slave controller will still maintain the correct ratio.

The slave flow will tend to lag behind the master flow. On a gas/air burner, the air flow could be master and the gas loop the slave. Such a system would run lean on increasing heat and run rich on decreasing heat. To some extent this can be overcome by making the master loop slower acting than the slave loop, possibly by tuning.

In a ratio system, a choice has to be made for master and slave loops. The first consideration is usually safety. In a gas/air burner, for example, air master/gas slave (called *gas follow air*) is usually chosen as most failures in the air loop cause the gas to shut down. If there are no safety considerations, the slowest loop should be the master and the fastest loop the slave to overcome the lag described above. Since 'fuel' (in both combustion and chemical terms) is usually the smallest flow in a ratio system and consequently has smaller valves/actuators, the safety and speed requirement are often the same.

The ratio block is a simple multiplier. If the ratio is simply set by an operator this can be a simple potentiometer acting

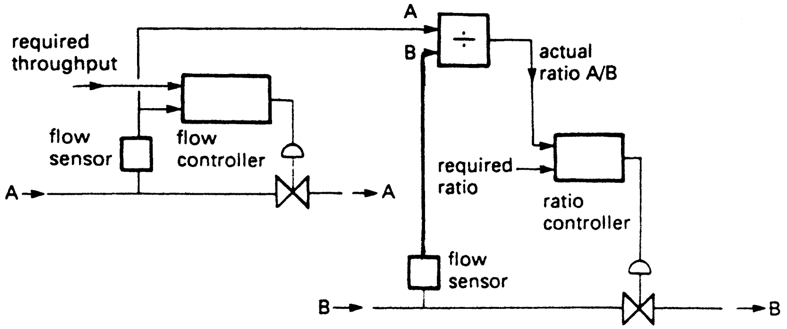


Figure 13.64 An intuitive, but incorrect, method of ratio control. The loop gain varies with throughput

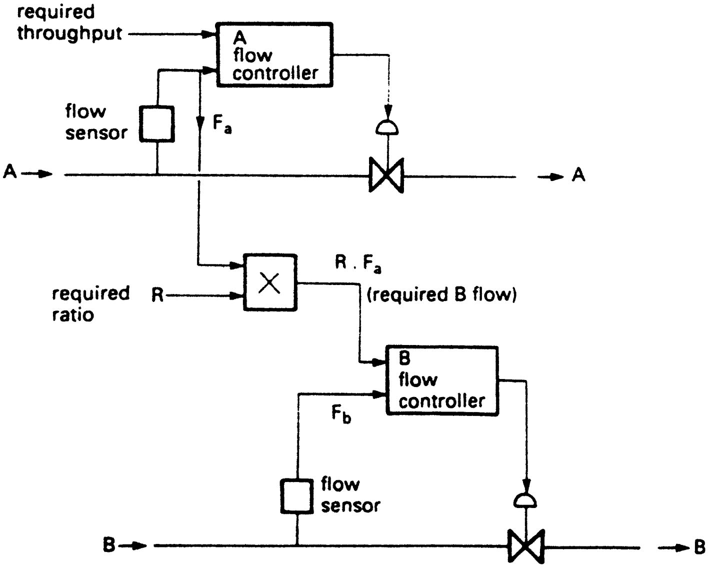


Figure 13.65 Master/slave ratio system with stable loop gains

as a voltage divider (for ratios less the unity) or an amplifier with variable gain (for ratios greater than unity). In digital control systems, of course, it is a simple multiply instruction. If the ratio is to be changed remotely (a trim control from an automatic sampler on a chemical blending system for example) a single quadrant analog multiplier is required.

Ratio blocks are generally easier to deal with in digital systems working in real engineering units. True ratios (an air/gas ratio of 10/1 for example) can then be used. In analog systems the range of the flow meters needs to be considered. Suppose we have a master flow with FSD of 12000 l/min, a slave flow of FSD 2000 l/min and a required ratio (master/slave) of 10/1. The required setting of R on *Figure 13.65* would be 0.6. In a well designed plant with correctly sized pipes, control valves and flow meters, analog ratios are usually close to unity. If not, the plant design should be examined.

Problems can arise with ratio systems if the slave loop saturates before the master. A typical scenario on a gas follow air burner control could go; the temperature loop calls for a large increase in heat (because of some outside influence). The air valve (master) opens fully, and the gas valve follows correctly but cannot match the requested flow. The resulting flame is lean and cold (flame temperature falls off rapidly with too lean a ratio) and the temperature does not rise. The system is now locked with the temperature loop demanding more heat and the air/gas loops saturated, delivering full flow but no temperature rise. The moral is; the master loop must saturate before the slave. If this is not achieved by pipe sizing the output of the master controller should be limited.

13.24.3 Lead lag control

Slave follow Master is simple, but one side effect is that the mixture runs lean for increasing throughput and rich for decreasing throughput because the master flow must always change first before the slave can follow. There is also a possible safety implication because a failure of the slave valve or controller could lead to a gross error in the actual ratio such as the fuel valve wide open and the air valve closed.

Better performance can be obtained with a system called *Lead-lag control* shown for an air/fuel burner on *Figure 13.66*. This uses cross linking and selectors to provide an air setpoint which is the highest of the external power demand signal or ratio'd fuel flow. The fuel setpoint is the lowest of the external power demand or ratio'd air flow.

This cross linking provides better ratio during changes, both air and fuel will change together. There is also higher security; a jammed open fuel valve will cause the air valve to open to maintain the correct ratio and prevent an explosive atmosphere of unburned fuel forming.

13.25 Transit delays

13.25.1 Introduction

Transit delays are a function of speed, time and distance. A typical example from the steel industry is the tempering process of *Figure 13.67* where red hot rolled steel travelling at 15 m/sec is quenched by passing beneath high pressure water sprays. The recovery temperature, some 50 m downstream, is the controlled variable which is measured by a pyrometer and used to adjust the water flow control valve. There is an obvious transit delay of 50/15 = 3.3 sec in the loop. A transit delay is a simple time shift which is independent of frequency.

Transit delays give an increasing phase shift with rising frequency which is de-stabilising. If conventional controllers are used significant detuning (low gain, large T_i) is necessary to maintain stability. The effect is shown on the

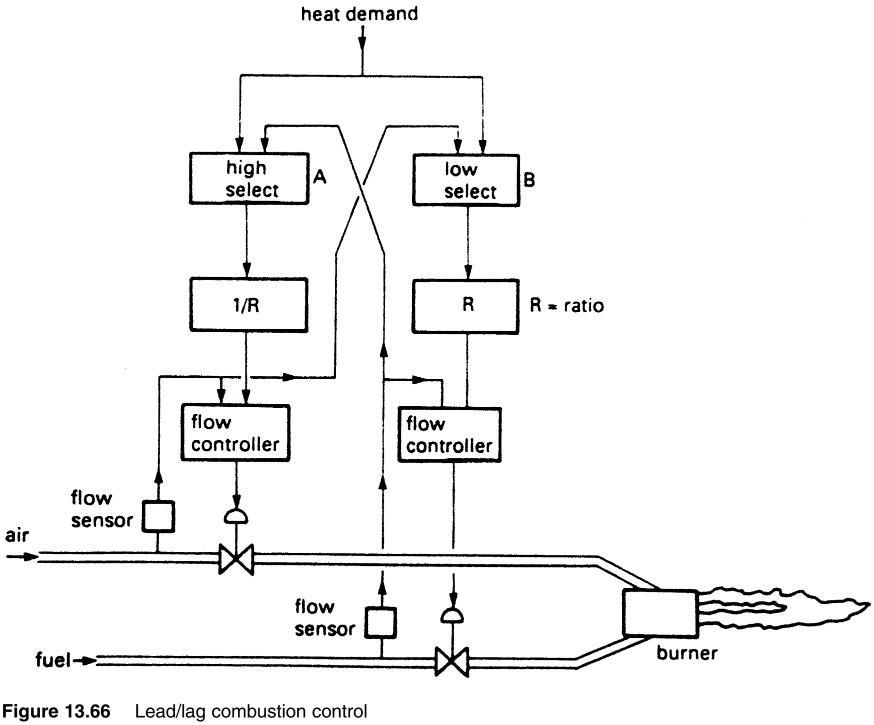


Figure 13.66 Lead/lag combustion control

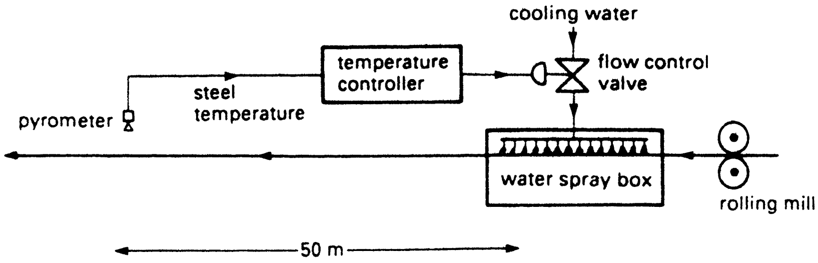


Figure 13.67 A tempering system dominated by a transit delay

Nichols charts of *Figure 13.68* for a simple system of two first order lags controlled by a PI controller. The de-stabilising effect of the increasing phase shift can clearly be seen. Derivative action, normally a stabilising influence, can also adversely affect a loop in which a transit delay is the dominant feature.

13.25.2 The Smith predictor

The effects of a transit delay can be reduced by the arrangement of *Figure 13.69* called a *Smith predictor*. The plant is considered to be an ideal plant followed by a transit delay. (This may not be true, but the position of the transit delay, before or after the plant, makes no difference to the plant behaviour.) The plant and its associated delay are modelled as accurately as possible in the controller.

The controller output, O_P, is applied to the plant and to the internal controller model. Signal A should thus be the same as the notional (and unmeasurable) plant signal X, and the signal B should be the same as the measurable controlled variable signal Y.

The PID controller however, is primarily controlling the model, not the plant, via summing junction 1. There are no delays in this loop, so the controller can be tuned for tight operation. With the model being the only loop, however, the plant is being operated in open loop control, and compensation will not be applied for model inaccuracies or outside disturbances.

Signal Y and B are therefore compared by a subtractor to give an error signal which encompasses errors from both disturbances and the model. These are added to the signal A from the plant model to give the feedback signal to the PID control block.

Discrepancies between the plant model and the real plant will be compensated for in the outer loop, so exact modelling is not necessary. The poorer the model, however, the less tight the control that can be applied in the PID block as the errors have to be compensated via the plant transit delay.

Smith predictors are usually implemented digitally, analog transit delays being difficult to construct. A digital delay line is simply a shift register in which values are shifted one place at each sample.

The Smith predictor is not a panacea for transit delays; it still takes the delay time from a setpoint change to a change in the process variable, and it still takes the delay time for a disturbance to be noted and corrected. The response to change, however, is considerably improved.

Systems with transit delays can benefit greatly from feedforward described previously in Section 13.23.3. Feedforward used in conjunction with a Smith predictor can be a very effective way of handling control systems with significant transit delays.

13.26 Stability

13.26.1 Introduction

At first sight it would appear that perfect control can be obtained by utilising a large proportional gain, short integral time and long derivative time. The system will then respond quickly to disturbances, alterations in load and set point changes.

Unfortunately life is not that simple, and in any real life system there are limits to the settings of gain T_i and T_d beyond which uncontrolled oscillations will occur. Like many engineering systems, the setting of the controller is a compromise between conflicting requirements.

13.26.2 Definitions and performance criteria

It is often convenient, (and not too inaccurate), to consider that a closed loop system behaves as a second order system, with a natural frequency ω_n and a damping factor β.

$$\frac{d^2x}{dt^2} + \Leftarrow 2\beta\omega_n \frac{dx}{dt} + \Leftarrow \omega_n^2 x = \Leftarrow f(t) \Leftarrow$$

It is then possible to identify five possible performance conditions, shown for a set point change and a disturbance in *Figure 13.70*(*a*) and (*b*).

An unstable system exhibits oscillations of increasing amplitude. A marginally stable system will exhibit constant amplitude oscillations. An underdamped system will be somewhat oscillatory, but the amplitude of the oscillations decreases with time and the system is stable. (It is important to appreciate that oscillatory does not necessarily imply instability). The rate of decay is determined by the damping factor. An often used performance criteria is the '*quarter amplitude damping*' of *Figure 13.70*(*c*) which is an underdamped response with each cycle peak one quarter of the amplitude of the previous. For many applications this is an adequate, and easily achievable response.

An overdamped system exhibits no overshoot and a sluggish response. A critical system marks the boundary between underdamping and overdamping and defines the fastest response achievable without overshoot.

For a simple system the responses of *Figure 13.70*(*a*) and (*b*) can be related to the gain setting of a P only controller, overdamped corresponding to low gain with increasing gain causing the response to become underdamped and eventually unstable.

It is impossible for any system to respond instantly to disturbances and changes in set point. Before the adequacy of a control system can be assessed, a set of performance criteria is usually laid down by production staff. Those defined in *Figure 13.71* are commonly used.

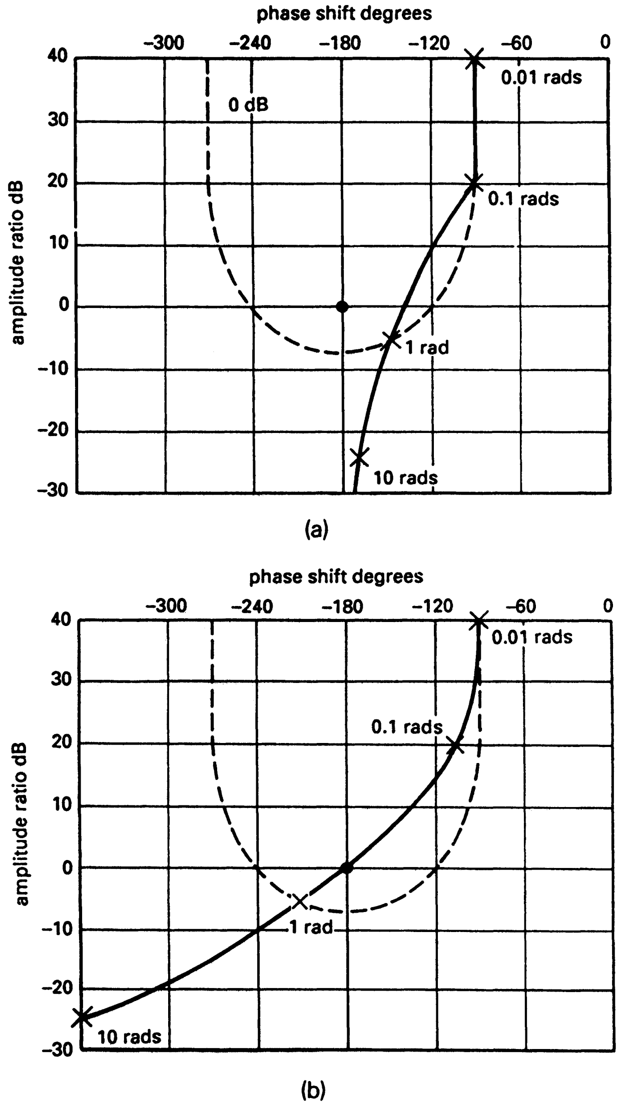


Figure 13.68 The effect of a transit delay on stability: (a) sketch of a Nichols chart for a system comprising a PI controller ($K=5$, $T_i=5$ s) and two first order lags of time constants 5 secs and 2 secs. The system is unconditionally stable; (b) the same system with a one second transit delay. The transit delay introduces a phase shift which increases with rising frequency and makes the system unstable

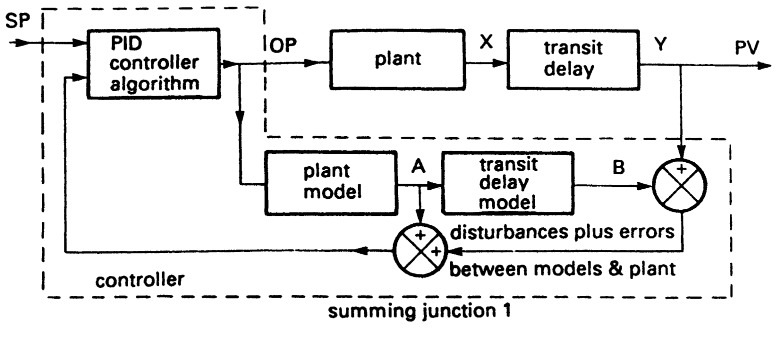


Figure 13.69 The Smith predictor used to reduce the effect of transit delays

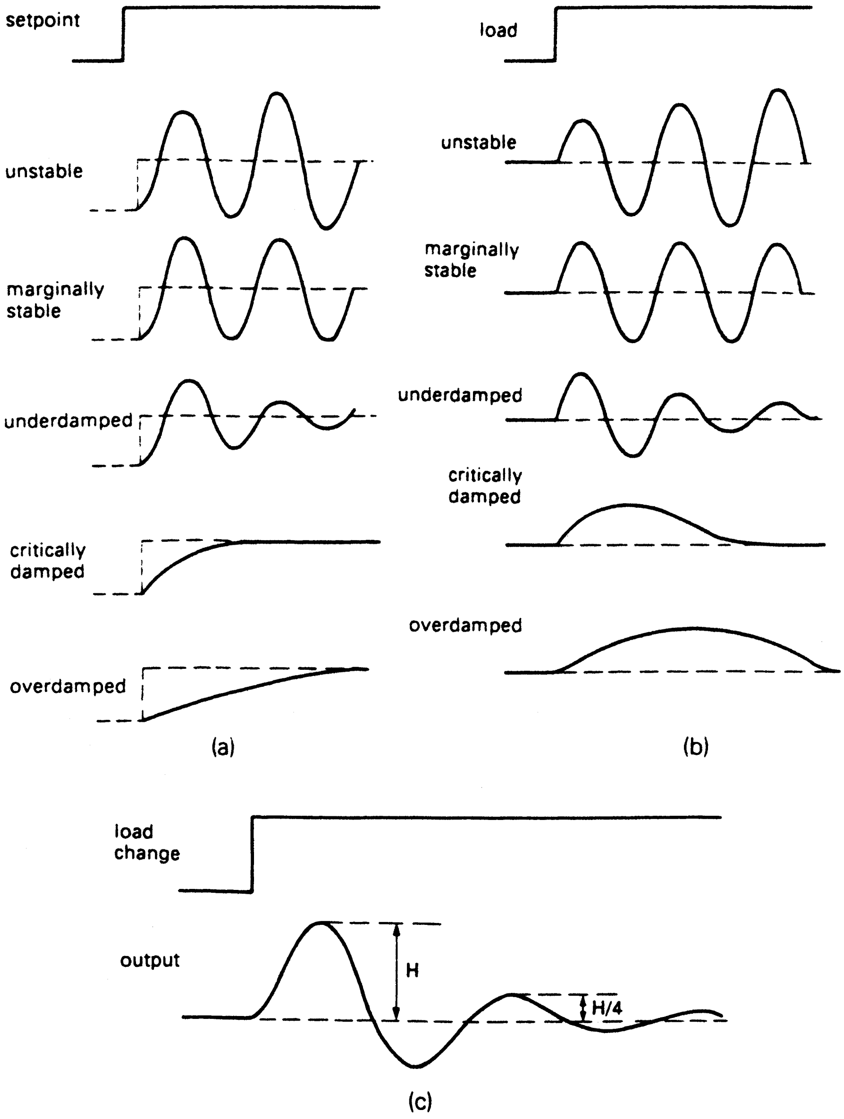


Figure 13.70 Various forms of system response: (a) step change in setpoint; (b) step change in load; (c) quarter amplitude damping

The '*rise time*' is the time taken for the output to go from 10% to 90% of its final value, and is a measure of the speed of response of the system. The time to achieve 50% of the final value is called the '*delay time*'. This is a function of, but not the same as, any transit delays in the system. The first overshoot is usually defined as a percentage of the corresponding set point change, and is indicative of the damping factor achieved by the controller.

As the time taken for the system to settle completely after a change in set point is theoretically infinite, a '*settling band*', '*tolerance limit*' or '*maximum error*' is usually defined. The settling time is the time taken for the system to enter, and remain within, the tolerance limit. Surprisingly an underdamped system may have a better settling time than a critically damped system if the first overshoot is just within the settling band. *Table 13.2* shows optimum damping factors for various settling bands. The settling time is defined in units of $1/\omega_n$.

Table 13.2

Settling band	*Optimum 'b'*	*Settling time*
20%	0.45	1.80
15%	0.55	2.00
10%	0.60	2.30
5%	0.70	2.80
2%	0.80	3.50

The shaded area is the integral of the error and this can also be used as an index of performance. Note that for a system with a standing offset (as occurs with a *P* only controller) the area under the curve will increase with time and not converge to a final value. Stable systems with integral action control have error areas that converge to a finite value. The area

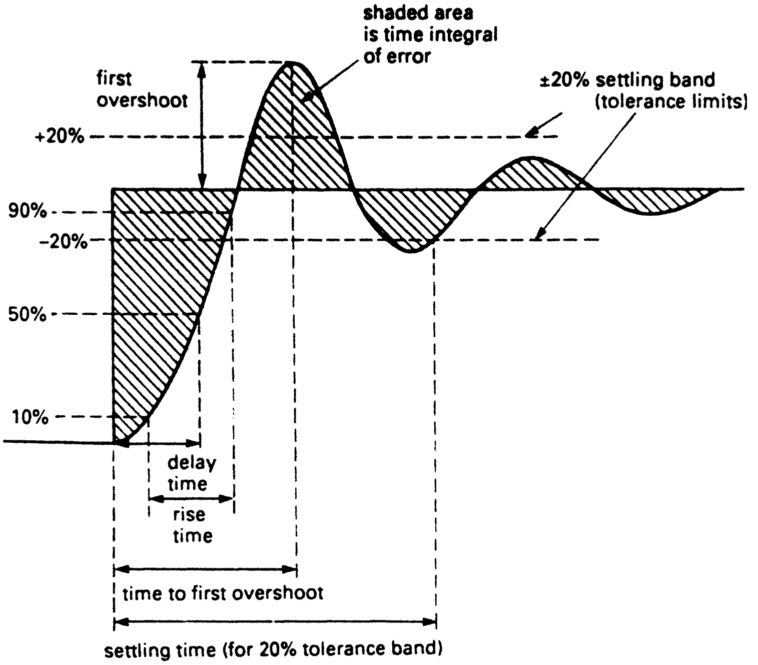


Figure 13.71 Common definitions of system response

between the curve and the set point is called the *integrated absolute error* (IAE) and is an accepted performance criterion.

An alternative criterion is the integral of the square of the instantaneous error. This weights large errors more than small errors, and is called *integrated squared error* (ISE). It is used for systems where large errors are detrimental, but small errors can be tolerated.

The performance criteria above were developed for a set point change. Similar criteria can be developed for disturbances and load changes.

13.26.3 Methods of stability analysis

The critical points for stability are open loop unity gain and a phase shift of −180°. It is therefore reasonable to give two figures of 'merit':

(a) The *Gain Margin* is the amount by which the open loop gain can be increased at the frequency at which the phase shift is −180°. It is simply the inverse of the gain at this critical frequency, for example if the gain at the critical frequency is 0.5, the gain margin is two.

(b) The *Phase Margin* is the additional phase shift that can be tolerated when the open loop gain is unity. With −140°⇐ phase shift at unity gain, there is a phase margin of 40°⇐.

For a reasonable, slightly underdamped, closed loop response the gain margin should be of the order of 6–12 dB and the phase margin of the order of 40–65°⇐.

Any closed loop control system can be represented by *Figure 13.72* where *G* is the combined block transfer function of the controller and plant and *H* the transfer function of the transducer and feed back components. The output will be given by:

$$P_{\mathrm{V}} = \Leftarrow \frac{G}{1+GH} S_{\mathrm{p}}$$

The system will be unstable if the denominator goes to zero or reverses in sign, i.e. $GH < \Leftarrow -1$. This is not as simple a relationship as might be first thought, as we are dealing with the dynamics of the process. The response of the system (gain and phase shift) will vary with frequency; generally the gain will fall and the phase shift will rise with increasing frequency. A phase shift of 180°⇐corresponds to multiplying a sine wave by −1, so if at some frequency the phase shift is 180°⇐and the gain at that frequency is greater than unity the system will be unstable.

There are several methods of representing the gain/phase shift relationship, and inferring stability from the plot. *Figure 13.73* is called a Bode diagram and plots the gain (in dB) and phase shift on separate graphs. Log-Lin graph paper

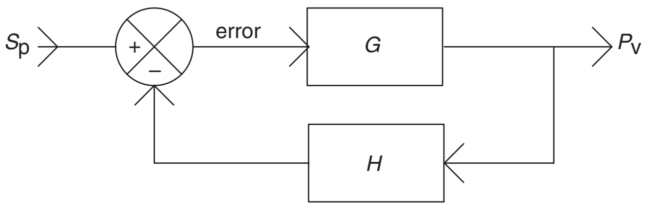


Figure 13.72 General block diagram of a closed loop control system

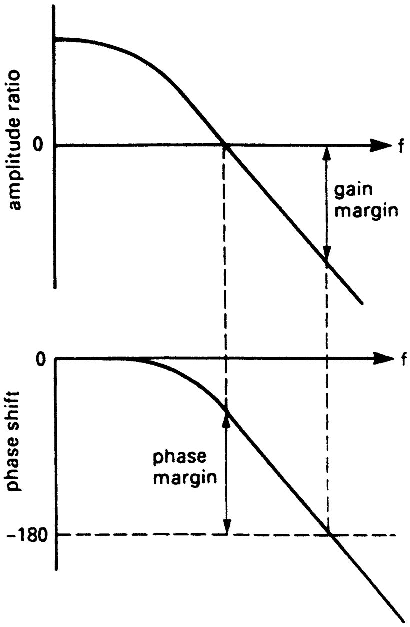


Figure 13.73 Gain and phase margins on the Bode diagram

(e.g. Chartwell 5542) is required. For stability, the gain curve must cross the 0 dB axis before the phase shift curve crosses the 180° line. From these two values, the gain margin and the phase margin can be read as shown.

Figure 13.74 is a Nichols chart and plots phase shift against gain (in dB). For stability, the 0 dB/ −180° intersection must be to the right of the curve for increasing frequency. Nichols charts are plotted on pre-printed graph paper (Chartwell 7514 for example) which allows the closed loop response to be read directly. If for example the curve is inside the closed loop 0 dB line damped oscillations will result. The gain and phase margins can again be read from the graph.

The final method is the Nyquist diagram of *Figure 13.75*. This plots gain again phase shift as a polar diagram (gain represented by distance from the origin). Chartwell graph paper 4001 is suitable. For stability the −180° point must be to the left of the graph for increasing frequency. Gain and phase margin can again be read from the graph.

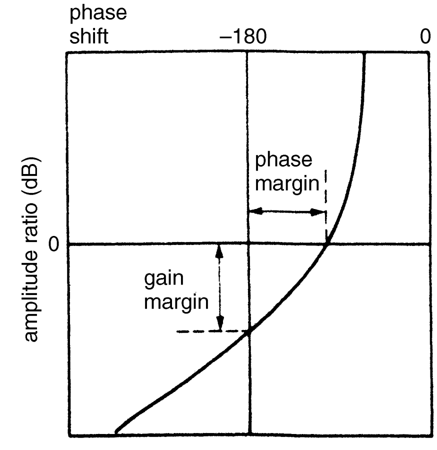


Figure 13.74 Gain and phase margins on a Nichols chart

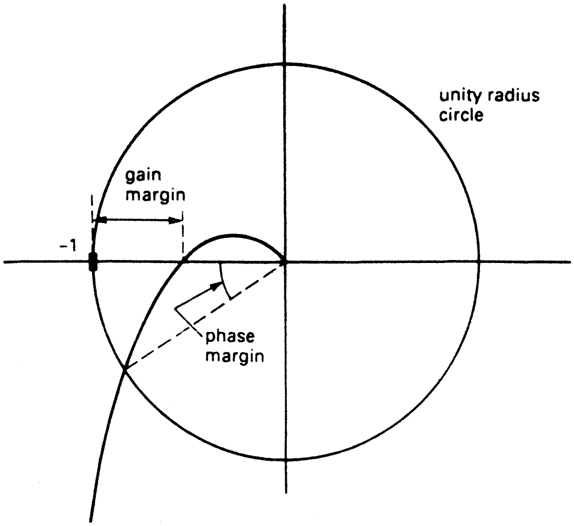


Figure 13.75 Gain and phase margins on a Nyquist diagram

13.27 Industrial controllers

13.27.1 Introduction

The commercial three term controller is the workhorse of process control and has evolved to an instrument of great versatility. This section describes some of the features of practical modern microprocessor based controllers.

13.27.2 A commercial controller

The description in this section is based on the 6360 controller manufactured by Eurotherm Process Automation Ltd of Worthing, Sussex.

The controller front panel is the 'interface' with the operator who may have little or no knowledge of process control. The front panel controls should therefore be simple to comprehend. *Figure 13.76* shows a typical layout.

The operator can select one of three operating modes—manual, automatic or remote—via the three push buttons labelled M, A, R. Indicators in each push-button show the current operating mode.

In manual mode, the operator has full control over the driven plant actuator. The actuator drive signal can be ramped up or down by holding in the M button and pressing the ▲ or ▼ buttons. The actuator position is shown digitally on the digital display, whilst the M button is depressed and continuously in analog form on the horizontal bargraph.

In automatic mode the unit behaves as a three term controller with a set point loaded by the operator. The unit is scaled into engineering units (i.e. real units such as °C, psi, litres/min) as part of the set up procedure so that the operator is working with real plant variables. The digital display shows the set point value when the SP button is depressed and the value can be changed with the ▲ and ▼ buttons.

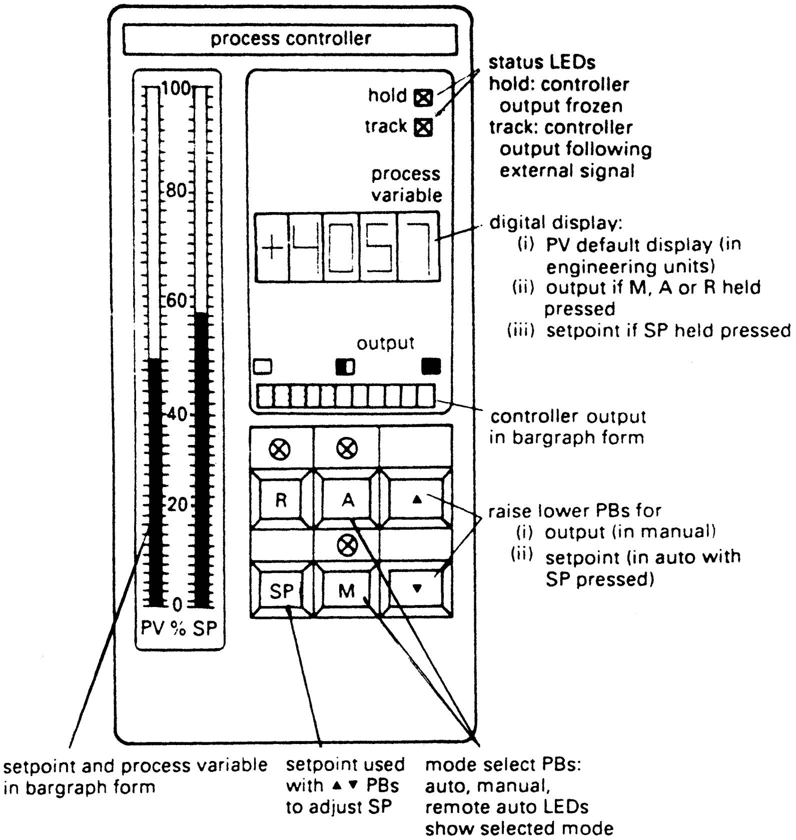

process controller
100
80
60
40
20
0
PV % SP
hold ☒
track ☒
process variable
+4057
output
R
A
▲
SP
M
▼
status LEDs
hold: controller output frozen
track: controller output following external signal
digital display:
(i) PV default display (in engineering units)
(ii) output if M, A or R held pressed
(iii) setpoint if SP held pressed
controller output in bargraph form
raise lower PBs for
(i) output (in manual)
(ii) setpoint (in auto with SP pressed)
setpoint and process variable in bargraph form
setpoint used with ▲ ▼ PBs to adjust SP
mode select PBs: auto, manual, remote auto LEDs show selected mode

Figure 13.76 Front panel operator controls on a typical controller

The set point is also displayed in bargraph form on the right-hand side of the dual vertical bargraph.

Remote mode is similar to automatic mode except the set point is derived from an external signal. This mode is used for ratio or cascade loops (see Sections 13.23 and 13.24) and batch systems where the setpoint has to follow a predetermined pattern. As before the setpoint is displayed in bargraph form and the operator can view, but not change, the digital value by depressing the SP button.

The process variable itself is displayed digitally when no push button is depressed, and continually on the left-hand bargraph. In automatic or remote modes the height of the two left-hand bargraphs should be equal, a very useful quick visual check that all is under control.

Alarm limits, (defined during the controller set up), can be applied to the process variable or the error signal. If either move outside acceptable limits, the process variable bargraph flashes, and a digital output from the controller is given for use by an external annunciator audible alarm of data logger.

Figure 13.77 shows a simple block diagram representation of a controller.

Input analog signals enter at the left-hand side. Common industrial signal standards are 0–10 V, 1–5 V, 0–20 mA and 4–20 mA. These can be accommodated by two switchable ranges 0–10 V and 1–5 V plus suitable burden resistors for the current signals (a 250 ohm resistor, for example, converts 4–20 mA to 1–5 V).

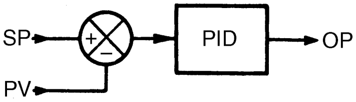


Figure 13.77 Block diagram of a typical controller

4–20 mA and 0–20 mA signals used on two wire loops require a DC power supply somewhere in the loop. A floating 30 V power supply is provided for this purpose.

Open circuit detection is provided on the main P_V input. This is essentially a pull up to a high voltage via a high value resistor. A comparator signals an open circuit input when the voltage rises. Short circuit detection can also be applied on the 1–5 V input (the input voltage falling below 1 V). Open circuit or short circuit P_V is usually required to bring up an alarm and trip the controller to manual, with the output signal driven high, held at last value, or driven low according to the nature of the plant being controlled. The open circuit trip mode is determined by switches as part of the set up procedure.

The P_V and remote S_P inputs are scaled to engineering units and linearised. Common linearisation routines are thermocouples, platinum resistance thermometers and square root (for flow transducers). A simple adjustable first order filter can also be applied to remove process or signal noise. The set point for the PID algorithm is selected from the internal set point or the remote set point by the from panel auto and remote push button contacts A, R.

The error signal is obtained by a subtractor (P_V and S_P both being to the same scale as a result of the scaling and engineering unit blocks). At this stage two alarm functions are applied. An absolute input alarm provides adjustable high and low alarm limits on the scaled and linearised P_V signals, and a deviation alarm (with adjustable limits) applied to the error signal. These alarm signals are brought out of the controller as digital outputs.

The basic PID algorithm is implemented digitally and includes a few variations to deal with some special circumstances. These modifications utilise the additional signals to the PID block (P_V, hold, track, output balance) and are described later.

The PID algorithm output is the actuator drive signal scaled 0–100%. The PID algorithm assumes that an increasing drive signal causes an increase in P_V. Some actuators, however, are reverse acting, with an increasing drive signal reducing P_V. A typical example is cooling water valves which are designed to fail open delivering full flow on loss of signal. Before the PID algorithm can be used with reverse acting actuators (or reverse acting transducers) its output signal must be reversed. A set up switch selects normal or inverted PID output. Note that reverse action does not alter the polarity of the controller output, merely the sign of the gain.

The output signal is selected from the manual raise/lower signal or the PID signal by the front panel manual/auto/remote pushbuttons M, A, R. At this stage limits are applied to the selected output drive. This limiting can be used to constrain actuators to a safe working range. The output limit allows the controller output to be limited just before the actuator's ends of travel, keeping the P_V under control at all times.

Two controller outputs are provided, 0–10 V and 4–20 mA for use with voltage and current driven actuators. The linearised P_V signal is also retransmitted as a 0–10 V signal for use with the separate external indicators and recorders.

13.27.3 Bumpless transfer

The output from the PID algorithm is a function of time and the values of the set point and the process variable. When the controller is operating in manual mode it is highly unlikely that the output of the PID block will be the same as the demanded manual output. In particular the integral term will probably cause the output from the PID block to eventually saturate at 0% or 100% output.

If no precautions are taken, therefore, switching from auto to manual, then back to auto again some time later will result in a large step change in controller output at the transition from manual to automatic operation.

To avoid this 'bump' in the plant operation, the controller output is fed back to the PID block, and used to maintain a PID output equal to the actual manual output. This balance is generally achieved by adjusting the contribution from the integral term.

Mode switching can now take place between automatic and manual modes without a step change in controller output. This is known as *manual/auto balancing*, *preload* or (more aptly) *bumpless transfer*.

A similar effect can occur on set point changes. With a straightforward PID algorithm, a setpoint change of ΔS_P will produce an immediate change in controller output of $K \cdot \Delta S_P$ where K is the controller gain. In some applications this step change in output is unacceptable. In *Figure 13.78* a term $K \cdot S_P$ is subtracted from the PID block output. The controller now responds to errors caused by changes in P_V in the normal way, but only reacts to changes in S_P via the integral and derivative terms. Changes in S_P thus result in a slow change in controller output. This is known as *setpoint change balance*, and is a switch selectable set up option.

This balance signal fed back from the output to the PID block is also used when the controller output is forced to follow an external signal. This is called *track mode*.

As before, the PID algorithm needs to be balanced to avoid a bump when transferring between track mode and automatic mode. The feedback output signal achieves this balance as described previously.

13.27.4 Integral windup and desaturation

Large changes in S_P or large disturbances to P_V can lead to saturation of the controller output or a plant actuator. Under these conditions the integral term in the PID algorithm can cause problems.

Figure 13.79 shows the probable response of a system with unrestricted integral action. At time A a step change in set point occurs. The output O_P rises first in a step ($K \times$ set point change) then rises at a rate determined by the integral time. At time B the controller saturates at 100% output, but the integral term keeps on rising.

At the time C P_V reaches, and passes, the required value, and as the error changes sign the integral term starts to decrease, but it takes until time D before the controller desaturates. Between times B and D the plant is uncontrolled, leading to an unnecessary overshoot and possibly even instability.

This effect is called *'integral windup'* and is easily avoided by disabling the integral term once the controller saturates either positive or negative. This is naturally a feature of all commercial controllers, but process control engineers should always by suspicious of 'home brew' control algorithms constructed (or written in software) by persons without control experience.

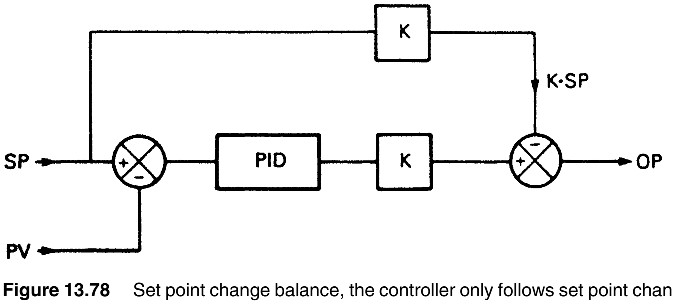


Figure 13.78 Set point change balance, the controller only follows set point changes on the integral term giving a ramped response to a change of set point

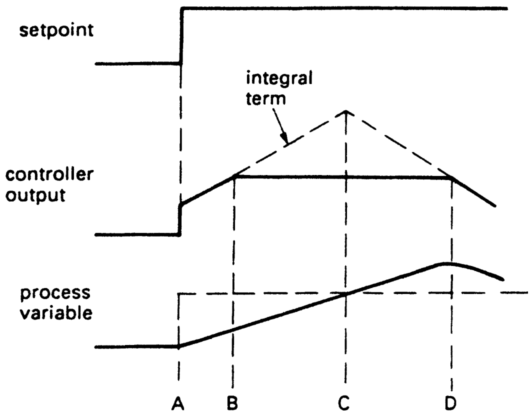


Figure 13.79 The effect of integral windup

In any commercial controller, the integral term would be disabled at point B on *Figure 13.80* to prevent integral windup. The obvious question now is at what point it is re-enabled again. Point C is obviously far too late (although much better than point D in the unprotected controller).

A common solution is to desaturate the integral term at the point where the rate of increase of the integral action equals the rate of decrease of the proportional and derivative terms. This occurs when the slope of the PID output is zero, i.e. when

$$e = -T_i\left(\frac{de}{dt} + T_d\frac{d^2e}{dt^2}\right)\Bigg(\qquad (13.23)\Leftarrow$$

with e being the error and T_i and T_d the controller constants.

Equation 13.23 brings the controller out of saturation at the earliest possible moment, but this can, in some cases, be too soon leading to an unnecessarily damped response. Some controllers allow adjustment of the desaturation point by adding an error limit circuit to delay the balance point to Equation 13.23 forcing the controller to remain in saturation for a longer time. The speed of desaturation and the degree of overshoot can thus be adjusted by the commissioning engineer.

13.27.5 Selectable derivative action

The term T_d (de/dt) in the three term controller algorithm can be rearranged as

$$T_d\left(\frac{dS_P}{dt} -\Leftarrow\frac{dP_V}{dt}\right)\Bigg($$

where S_P is the set point and P_V the process variable. The derivative term thus responds to changes in both the set point and the plant feedback signal.

This is not always desirable; in particular a step change in set point leads to an infinite spike controller output and a vicious 'kick' to the actuator. Commercial controller therefore include a selectable option for the derivative term to be based on true error (S_P–P_V) or purely on the value of P_V alone.

There is generally no noticeable difference in plant performance between these options; stability or the ability to deal with disturbances or load changes are unaffected, and derivative on P_V is normally the preferred choice. The only occasion when true derivative on error is advantageous is where the P_V is required to track a continually changing S_P.

13.27.6 Variations on the PID algorithm

The theoretical PID algorithm is described by the equation

$$O_P = K\left(e +\Leftarrow\frac{1}{T_i}\int\Bigg(e\,dt + T_d\frac{de}{dt}\right)\Bigg($$

where e is the error, K is the gain, T_i is the integral time and T_d the derivative time. Unfortunately different manufacturers use different terminology and even different algorithms.

Many manufacturers define the gain as the *proportional band*, denoted as *PB* or P_B. This is the inverse of the gain expressed as a percentage, i.e

$$P_B =\Leftarrow\frac{100}{K}\%$$

A gain of two is thus the same as a proportional band of 50%, and decreasing the proportional band increases the gain.

The integral time is commonly expressed as '*Repeats per Minute*' or rpm. The relationship is given by:

$$\text{Repeats per min} = 1/T_i \ (\text{for } T_i \text{ in min})$$
$$= 60/T_i \ (\text{for } T_i \text{ in sec})$$

The derivative time is often called the *rate* or *pre-act* term but these are all identical to T_d.

More surprisingly there are variations on the basic algorithm. Some manufacturers use a so called '*non interacting*' or '*parallel*' equation which can be expressed as:

$$O_P = K_e +\Leftarrow\frac{1}{T_i}\int\Bigg(e\,dt + T_d\frac{de}{dt}$$

or

$$O_P = K_e + K_i\int\Bigg(e\,dt + K_d\frac{de}{dt}$$

In these the three terms are totally independent. In the second version K_i is called the integral gain and K_d the derivative gain. Note that increasing K_i has the same effect as decreasing T_i. It is tempting to think that the non interacting equations are simpler to use, but in practice the theoretical model is more intuitive. In particular, as the gain K is reduced in the non interacting equation, any integral action has more effect and contributes more phase shift. Increasing or decreasing the gain with a non interacting controller can thus cause instability.

There is yet a third form of PID algorithm known as the '*series*' equation. This can be expressed as:

$$O_P = K\left(\Bigg(e +\Leftarrow\frac{1}{T_i}\int\Bigg(e\,dt\right)\left(\Bigg(1 + T_d\frac{de}{dt}\right)\Bigg($$

This algorithm is based on pneumatic and early electronic controllers, and some manufacturers have maintained it to give backward compatibility. This has the odd characteristic that the T_i and T_d controls interact with each other, with the maximum derivative action occurring when T_d and T_i are set equal. In addition the ratio between T_i and T_d interacts with the overall gain.

There are further variations on the way the derivative contribution is handled. We have already discussed the effect of derivative on process variable and derivative on error. Because the pure derivative term gives increasing gain with increasing frequency it amplifies any high frequency noise resulting in continual twitchy movements of the plant actuators. Many manufacturers therefore deliberately roll off the high frequency gain, either by filtering the signal applied to the derivative function or directly limiting the derivative action.

13.27.7 Incremental controllers

Diaphragm operated actuators can be arranged to fail open or shut by reversing the relative positions of the drive pressure and return spring. In some applications a valve will be required to hold its last position in the event of failure. One way to achieve this is with a motorised actuator, where a motor drives the valve via a screw thread.

Such an actuator inherently holds its last position but the position is now the integral of the controller output. An integrator introduces 90°⇐phase lag and gain which falls off with increasing frequency. A motorised valve is therefore a destabilising influence when used with conventional controllers.

Incremental controllers are designed for use with motorised valves and similar integrating devices. They have the control algorithm

$$O_P = K\left(\frac{1}{T_i}e + \frac{de}{dt} + \Leftarrow \frac{1}{T_d}\frac{d^2e}{dt^2}\right)\Bigg($$

which is the time derivative of the normal control algorithm.

Incremental controllers are sometimes called *boundless controllers* or *velocity controllers* because the controller output specifies the actuator rate of change (i.e. velocity) rather than actual position.

Incremental controllers cannot suffer from integral windup per se, but it is often undesirable to keep driving a motorised valve once the end of travel is reached. End of travel limits are often incorporated in motorised valves to prevent jamming. The controller also has no real ‘idea’ of the valve true position, and hence cannot give valve position indication. If end of travel signals are available, a valve model can be incorporated into the controller to integrate the controller output to give a notional valve position. This model would be corrected whenever an end of travel limit is reached. Alternatively a position measuring device can be fitted to the valve for remote indication.

Pulse width modulated controllers are a variation on the incremental theme. Split phase motor drive valves require logic raise/lower signals, and normal proportional control can be simulated by using time proportional raise/lower outputs.

13.27.8 Scheduling controllers

Many loops have properties which change under the influence of some measurable outside variable. The gain of a flow control valve, (i.e. the change in flow for change in valve position) varies considerably over the stroke of a valve. The levitation effect of steam bubbles in a boiler drum causes the drum level control to have different characteristics under start-up, low load and high load conditions.

A scheduling controller has a built-in look up table of control parameters (gain, filtering, integral time etc.) and the appropriate values selected for the measured plant conditions.

13.27.9 Variable gain controllers

Process variable noise occurs in many loops; level and flow being possibly the worst offenders. This noise causes unnecessary actuator movement, leading to premature wear and inducing real changes in the plant state. Noise can, of course, be removed by first or second order filters, but these reduce the speed of the loop and the additional phase shift from the filters can often act to de-stabilise a loop.

A controller with gain K will pass a noise signal $K{\cdot}n(t)$ to the actuator where $n(t)$ is the noise signal. One obvious way to reduce the effect of the noise is to reduce the controller gain, but this degrades the loop performance. Usually the noise signal has a small amplitude compared with the signal range, if it has not the process will be practically uncontrollable. What is intuitively required is a low gain when the error is low, but a high gain when the error is high.

Figure 13.80(*a*) shows how such a scheme operates. The noise amplitude lies in the range AB, so this is made a low gain region. Outside this band the gain is much higher. The gain in the region AB should be low, but not zero, to keep the process variable at the set point. With a pure deadband (i.e. zero gain in region AB) the process variable would cycle between one side of the centre band and the other.

Figure 13.80(*b*) shows a possible implementation. A comparator switches between a low gain and high gain controller according to the magnitude of the error. Note that integral balancing is required between the two controllers to stop integral windup in the unselected controller.

Figure 13.80 has two gain regions. It is possible to construct a controller whose gain varies continuously with error. Such a controller has a response

$$O_P = Kf(e)\Bigg(\Leftarrow e + \Leftarrow\frac{1}{T_i}\int\Bigg(e\,dt + T_d\frac{de}{dt}\Bigg)\Bigg($$

where $f(e)$ is a function of error.

A common function is

$$f(e) = abs\left(\frac{m + (1-m)e}{100}\right)\Bigg(\qquad (13.24)\Leftarrow$$

where e is expressed as a percentage (0–100%) and m is a user set linearity adjustment ($0 < m < 1$). The *abs* operation (which always returns a positive sign) is necessary to prevent the controller action changing sign on negative error.

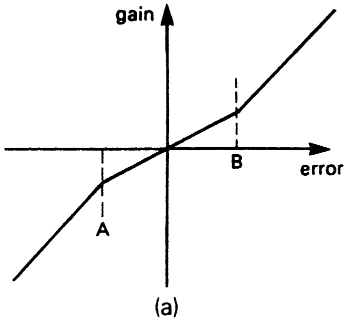
With $m = 1, f(e) = 1$ and Equation 13.24 behaves as a normal three term controller. With $m = 0$. the proportional part of Equation 13.24 follows a square law. Like *Figure 13.80*(*a*), this has low gain or small error (zero gain at zero error) but progressively increasing gain as the error increases.

Position control systems often need a fast response but cannot tolerate an overshoot. These often use Equation 13.24 with m at a low value approximating to the quadratic curve. This gives a high take off speed, but a low speed of approach.

13.27.10 Inverse plant model

The ideal control strategy, in theory, is one which mimics the plan behaviour. Given a totally accurate model of the plant, it should be possible to calculate what controller output is required to follow set point change, or compensate for a disturbance. The problem here is, of course, having an accurate plant model, but even a rough approximation should suffice as the controller output will converge to the correct value eventually.

One possible solution is shown in *Figure 13.81*. The process is represented by a block with transfer function $K{\cdot}f(s)$ where K is the d.c. (low frequency) gain. Following a change in set point, the signal A should mimic exactly the process



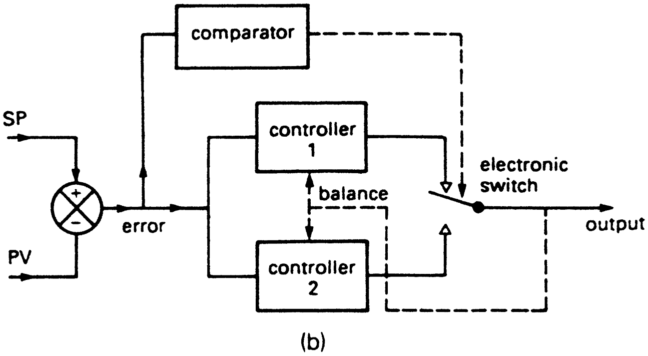


Figure 13.80 Variable gain controller: (a) system response; (b) block diagram

variable *B*, leading to a constant output from the controller exactly correct to bring the plant to the set point without overshoot. With a perfect model, the change at *A* should match the change at *B* as the set point is approached.

The inverse plant model is usually implemented with a sampled digital system. The problem with this simple, and apparently ideal, controller is that it will probably demand actuation signals which will drive the controller output, the actuator or parts of the plant, into saturation. It also requires an accurate plant model. A more gentle version of this technique aims to get a fraction, say 0.1 of the way from the current value to the desired value of the process variables at each sample time. This approximates to an exponential response.

13.28 Digital control algorithms

13.28.1 Introduction

So far we have assumed that controllers deal with purely analog signals. Increasingly controllers are digital, with the analog signal from the transducer being sampled by an ADC, the control algorithm being performed by software and the analog output being obtained from a DAC. ADCs and DACs are described in Chapter 14, Section 14.9. The system does not therefore continually control but takes 'snapshots' of the system state. Such an approach is called a *sampled system.*

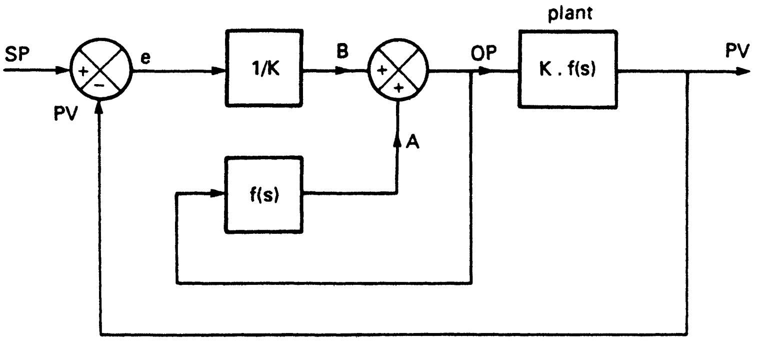


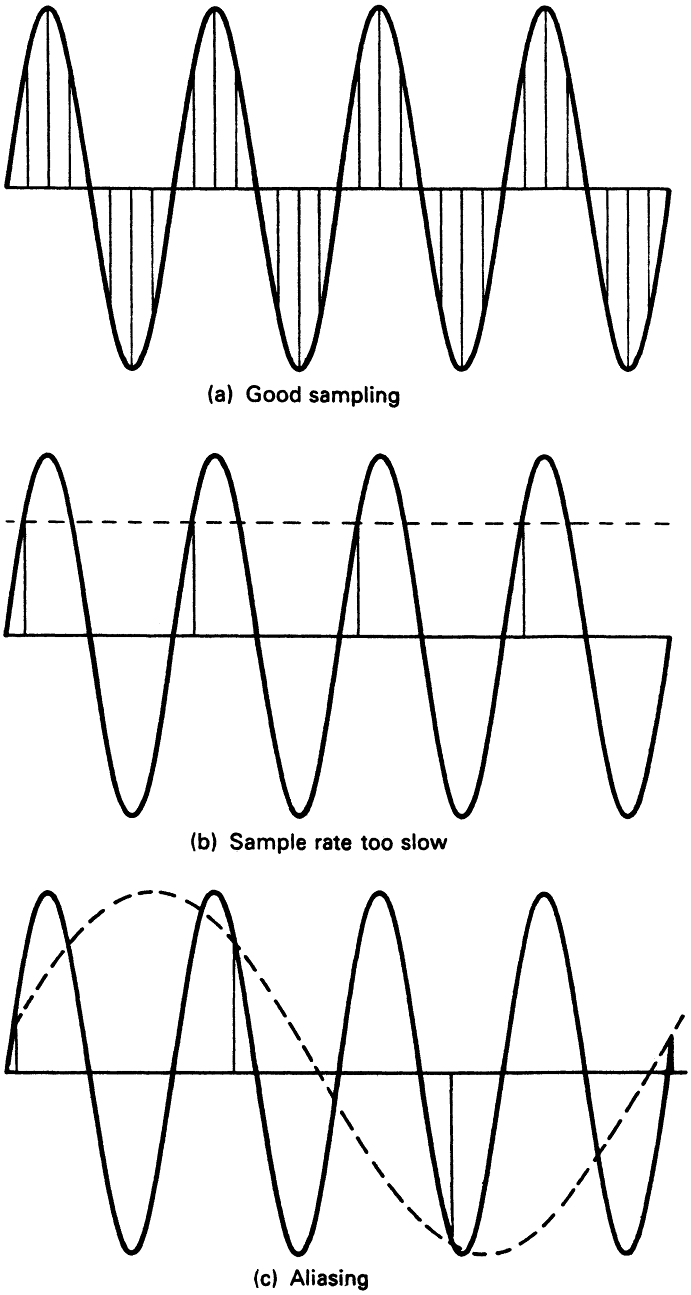
Figure 13.81 The inverse plant model

13.28.2 Shannon's sampling theorem

A sampled system only knows about the values of its samples. It cannot infer any other information about the signals it is dealing with. An obvious question, therefore, is what sample rate we should choose if our samples are to accurately represent the original analog signals.

In *Figure 13.82*(*a*) a sine wave is being sampled at a relatively fast rate. Intuitively one would assume this sampling rate is adequate. In *Figure 13.82*(*b*) the sample rate and the frequency are the same. This is obviously too slow as the samples imply a constant unchanging output.

In *Figure 13.82*(*c*) the sample rate is lower than the frequency and the sample values are implying a sine wave of much lower frequency than the signal. This is called '*aliasing*'. A visual effect of aliasing can be seen on cinema screens where moving wheels often appear to go backwards. This effect occurs because the camera samples the world at about 50 times per second.


(a) Good sampling

(b) Sample rate too slow

(c) Aliasing

Figure 13.82 The effect of the sampling rate: (a) good sampling rate; (b) sampling frequency same as signal frequency, too slow; (c) sampling rate much too slow, aliasing is occurring

Any continuous signal will have a bandwidth of interest. The sampling frequency should be at least twice the bandwidth of interest. This is known as *Shannon's sampling theorem*. Any real life system will not, however, have a well defined bandwidth and sharp cut-off point. Noise and similar effects will cause any real signal to have a significant component at higher frequencies. Aliasing may occur with these high frequency components and cause apparent variations in the frequency band of interest. Before sampling, therefore, any signal should be passed through a low pass *anti-aliasing filter* to ensure only the bandwidth of interest is sampled.

Most industrial control signals have a bandwidth of a few Hz, so sampling within Shannon's limit is usually not a problem. Normally the critical bandwidth is not known precisely so a sample rate of about 5 to 10 times the envisaged bandwidth is used.

13.28.3 Control algorithms

To achieve three term control with sampled signals we must find the derivative and the integral of the error. As shown on *Figure 13.83* we are dealing with a set of sampled signals, y_n, y_{n-1}, y_{n-2} etc. where y_n is the most recent. If the sample time is Δt, the slope is then given by:

$$\text{slope} =\Leftarrow \frac{Y_n - \Leftarrow Y_{n-1}}{\Delta t}$$

Integration is equivalent to finding the area under a curve as shown for an analog and digital signal on *Figure 13.83(b)*. The trapezoid integration of *Figure 13.83(c)* is commonly used where the area is given by:

$$\text{area} =\Leftarrow \frac{\Delta t(Y_n + \Leftarrow Y_{n-1})\Leftarrow}{2}$$

Combining these gives a digital sampled PID algorithm

$$O_P =\Leftarrow K\left(e_n + \Leftarrow \frac{1}{T_i}\sum \frac{\Delta t(e_n + \Leftarrow e_{n-1}}{2} + \frac{T_d}{\Delta t}(e_n - \Leftarrow e_{n-1})\Leftarrow \right)\Leftarrow\Big($$

where e_n is the error for sample n and Δt the sample time as before.

13.29 Auto-tuners

Tuning a controller is more of an art than an exact science and can be unbelievably time consuming. Time constants of tens of minutes are common in temperature loops, and lags of hours occur in some mixing and blending processes. Performing, say, the ultimate cycle test of Section 13.30.2 on such loops can take several days.

Self tuning controllers aim to take the tedium out of setting up a control loop. They are particularly advantageous if the process is slow (i.e. long time constants) or the loop characteristics are subject to change (e.g. a flow control loop where pressure/temperature changes in the fluid alter the behaviour of the flow control valve.)

Self tuning controllers give results which are generally as good, if not slightly better, than the manual methods of Section 13.30 (possibly because self tuning controllers have more patience than humans!). In the author's experience, however, the results from a self tuner should be viewed as recommendations or initial settings in the same way as the results from the manual methods described in the following sections. One early decision to be made when self tuners are used is whether they should be allowed to alter control parameters without human intervention. Many engineers (of whom the author is one) view self tuners as commissioning aids to be removed before a plant goes into production.

There are essentially two groups of self tuners. *Modelling self tuners* try to build a mathematical model of the plant (usually second order plus transit delay) then determine controller parameters to suit the model. These are sometimes called *explicit self tuners*.

Model identification is usually based on the principles of *Figure 13.84*. The controller applies a control action O_P to the plant and to an internal model. The plant returns a

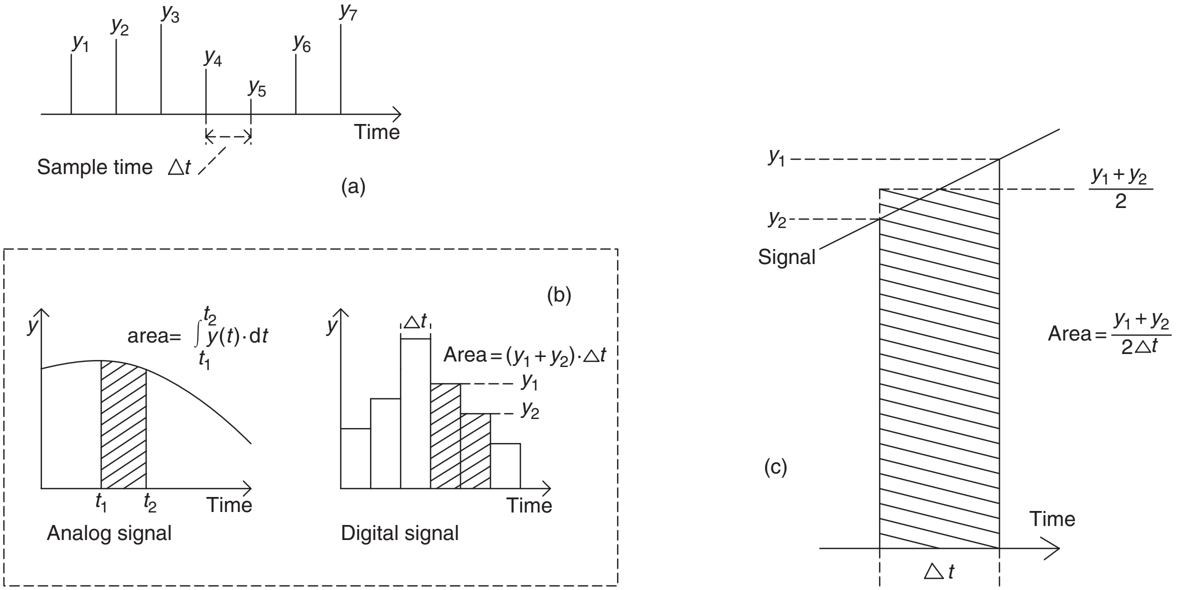


Figure 13.83 Control algorithm with sampled signals: (a) the sampled signals; (b) integration for an analog and digital signal; (c) trapezoid integration

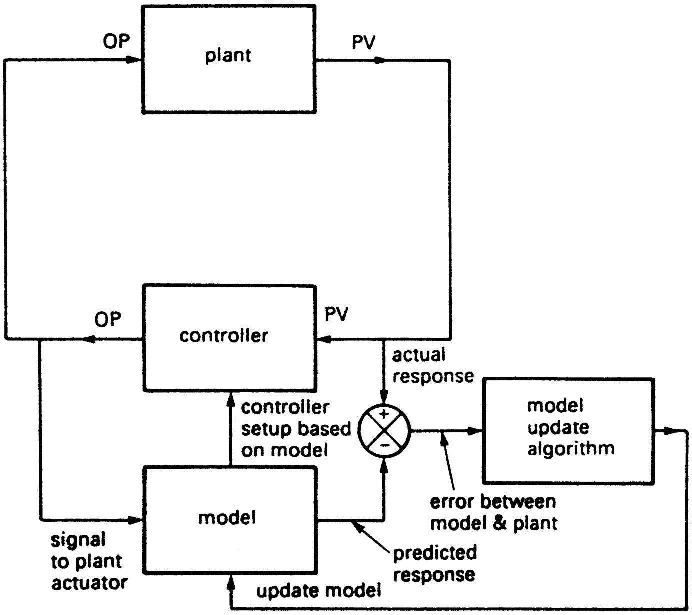


Figure 13.84 A modelling self tuning controller

process variable P_V and the model a prediction P_{Vm}. These are compared, and the model updated (often via the statistical least squares technique). On the basis of the new model, new control parameters are calculated, and the sequence repeated.

A model building self tuner requires actuation changes to update its model, so it follows that self tuners do not perform well in totally static conditions. In a totally stable unchanging loop, the model, and hence the control parameters, can easily drift off to ridiculous values. To prevent this, most self tuners are designed to 'kick' the plant from time to time, with the size and repetition rate of the kick being set by the control engineer. Less obviously, model building self tuners can be confused by outside disturbances which can cause changes in P_V that are not the result of the controller output.

The second group of self tuners (sometimes called *implicit tuners*) use automated versions of the manual tests described in Section 13.30, and as such do not attempt to model the plant. A typical technique will vary the controller gain until a damped oscillatory response is observed. The control parameters can then be inferred from the controller gain, the oscillation period and the oscillation decay rate.

The useful bang/bang test of Section 13.30.3 can be performed automatically by a controller which forces limit cycling in the steady state via a comparator. A limit block after the comparator restricts the effect on the plant.

Implicit self tuners, like their modelling brothers, do not perform well on a stable unchanging loop, and can be equally confused by outside disturbances.

13.30 Practical tuning methods

13.30.1 Introduction

Values must be set for the gain and integral/derivative times before a controller can be used. In theory, if a plant model is available, these values can be determined from Nichols charts or Nyquist diagrams. Usually, however, the plant characteristics are not known (except in the most general terms) and the controller has to be tuned by experimental methods.

It should be noted that all these methods require pushing the plant to the limit of stability. The safety implications of these tests must be clearly understood. Tuning can also be very time consuming. With large chemical plants tuning of one loop can take days.

Most of the tests aim to give a quarter cycle decay and assume the plant consists of a transit delay in series with a second order block (or two first order lags) plus possible integral action.

In conducting the tests, it is useful to have a two pen recorder connected to the P_V (process variable) and O_P (controller output) as shown on *Figure 13.85*. The range of the pens (e.g. 0–10 V or 1–5 V) should be the same.

In the tests below, the gain is expressed as proportional band (P_B) per cent. Time is used for integral and derivative

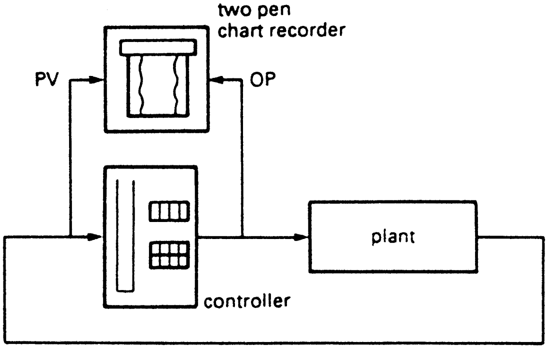


Figure 13.85 Suggested equipment setup for controller tuning

action. Conversion to gain or repeats per minute is straightforward.

13.30.2 Ultimate cycle methods

The basis of these methods is determining the controller gain which just supports continuous oscillation, i.e. point *A* and gain *K* on the Nichols chart and Nyquist diagram of *Figure 13.86*. The method is based on work by J. G. Ziegler and N. B. Nichols and is often called the Ziegler Nichols method.

The integral and derivative actions are disabled to give proportional only control, and the control output manually adjusted to bring P_V near the required value. Auto control is selected with a low gain.

Step disturbances are now introduced and the effect observed. One way of doing this is to go back into manual, shift O_P by, say 5%, then reselect automatic control. At each trial the gain is increased. The increasing gain will give a progressively underdamped response and eventually continuous oscillation will result. Care must be taken in these tests to allow all transients to die away before each new value of gain is tried.

If the value of gain is too high, the oscillations will increase. The value of gain which gives constant oscillations neither increasing or decreasing is called the ultimate gain, or P_u (expressed as proportional band). The period of the oscillations T_u should also be noted from the chart recorder (or with a watch).

The required controller settings are:

Proportional only control

P_B	$2P_\mathrm{u}\,\%$

PI Control

P_B	$2.2P_\mathrm{u}\,\%$
T_i	$0.8.T_\mathrm{u}$

PID Control

P_B	$1.67\,.\,P_\mathrm{u}\,\%$
T_i	$T_\mathrm{u}/2$
T_d	$T_\mathrm{u}/8$

$T_\mathrm{i} = 4T_\mathrm{d}$ is a useful rule of thumb.

Other recommended settings for a PID controller are:

P_B	$2P_\mathrm{u}\,\%$
T_i	T_u
T_d	$T_\mathrm{u}/5$

and

P_B	$2P_\mathrm{u}\,\%$
T_i	$0.34T_\mathrm{u}$
T_d	$0.08T_\mathrm{u}$

All of these values should be considered as starting points for further tests.

13.30.3 Bang/bang oscillation test

This is the fastest, but most vicious, test. It can, though, be misleading if the plant is non linear. Integral and derivative actions are disabled and the controller gain set as high as possible (ideally infinite) to turn the controller into a bang-bang controller. The controller output is set manually to bring the process value near the set point then the controller switch into automatic mode.

Violent oscillations will occur as shown on *Figure 13.87*. The period of the oscillations T_o is noted along with the peak to peak height of the process variable oscillations as a percentage $H_\mathrm{o}\%$ of full scale.

The required controller settings are:

Proportional control

P_B	$2.H_\mathrm{o}\%$

PI Control

P_B	$3.H_\mathrm{o}\%$
T_i	$2.T_\mathrm{o}$

PID Control

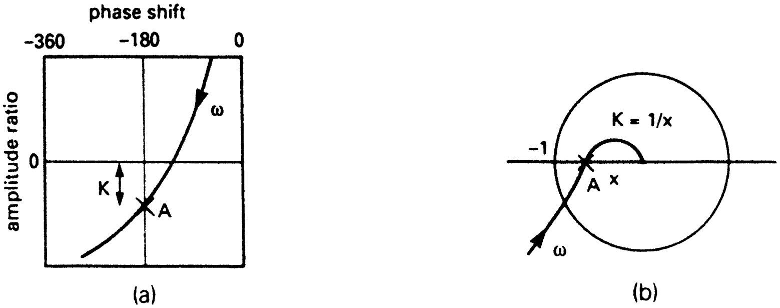
P_B	$2.H_\mathrm{o}\%$
T_i	T_o
T_d	$T_\mathrm{o}/4$

13.30.4 Reaction curve test

This is an open loop test originally proposed by American engineers Cohen and Coon. It assumes the plant consists of a measurable transit delay and a dominant time constant. It cannot be applied to plants with integral action (e.g. level control systems).

A chart recorder must be connected to the plant as shown earlier on *Figure 13.85* to perform the test. The controller output is first adjusted manually to bring the plant near to the desired operating point. After the transients have died away a small manual step ΔO_P is applied which results in a small change ΔP_V as shown on *Figure 13.88*.

The process gain K_p is then simply $\Delta P_\mathrm{V}/\Delta O_\mathrm{P}$.



(a)

(b)

Figure 13.86 Basis of the ultimate cycle test. Point A determines the frequency at which continuous oscillations will occur when gain K is applied: (a) Nichols chart; (b) Nyquist diagram

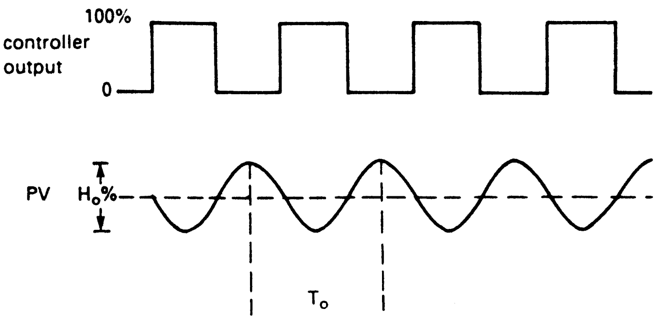


Figure 13.87 The bang-bang oscillation test

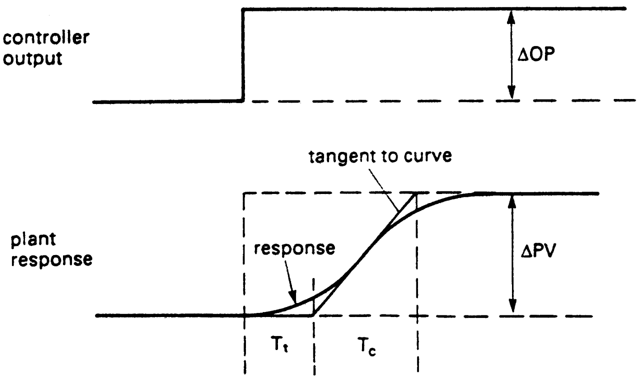


Figure 13.88 The reaction curve test

A tangent is drawn to the process variable curve at the steepest point from which an apparent transit delay T_t and time constant T_c can be read. The settings for the controller are then given by:

Proportional
P_B $100.K_p.T_t/T_c$ %

PI
P_B $110.K_p.T_t/T_c$ %
T_i $3.3\ T_t$

PID
P_B $80.K_p.T_t/T_c$ %
T_i $2.5\ T_t$
T_d $0.4\ T_t$

Because the test uses an open loop trial with a small change in the controller output it is the gentlest and least hazardous tuning method.

13.30.5 A model building tuning method

The closed loop tuning methods described so far require the plant to be pushed to, (and probably beyond), the edge of instability in order to set the controller. An interesting gentle tuning method was described by Yuwana and Seborg in the journal *AIChE* Vol 28 no 3 in 1982.

The method assumes the plant has gain K_M and behaves as a dominant first order lag T_M in series with a transit delay D_M. This assumption can give gross anomalies with plants with integral action such as level or position controls, but is commonly used for many manual and automatic/adaptive controller tuning methods. With the above warning noted, the suggested controller settings can be found from:

$$K = <\mathrm{A}(\mathrm{D_M}/T_\mathrm{M})^{-\mathrm{B}}/K_\mathrm{M}$$

$$T_\mathrm{I} = <\mathrm{C}T_\mathrm{M}(\mathrm{D_M}/T_\mathrm{M})^{\mathrm{D}}$$

$$T_\mathrm{D} = <T_\mathrm{M}\mathrm{E}(\mathrm{D_M}/T_\mathrm{M})^{\mathrm{F}}$$

where A, B, C, D, E, F are constants defined:

Mode	*A*	*B*	*C*	*D*	*E*	*F*
P	0.490	1.084				
PI	0.859	0.997	1.484	0.680		
PID	1.357	0.947	1.176	0.738	0.381	0.99

These apparently random equations and constants come from experimental work described by Miller *et al* in *Control Engineering* Vol 14 no 12.

The method of finding the plant gain, time constant and transit delay is based on a single quick test with the plant operating under closed loop control. The test is performed on the plant operating under proportional only control, with a gain sufficient to produce a damped oscillation as *Figure 13.89* when a step change in set point from R_0 to R_1 is applied.

The subsequent process maximum C_{P1}, minimum C_{M1} and next maximum C_{P2} are noted along with the time D_{T2} between C_{P1} and C_{P2}. The controller proportional band used for the test, P_B, is also recorded, from which the controller gain $K_{PB} = <100/P_B$ is found.

Given the values from the test the method estimates the value of the plant steady state gain K_M, lag time constant T_M and the transit delay time D_M. The background mathematics is given at length in the original paper.

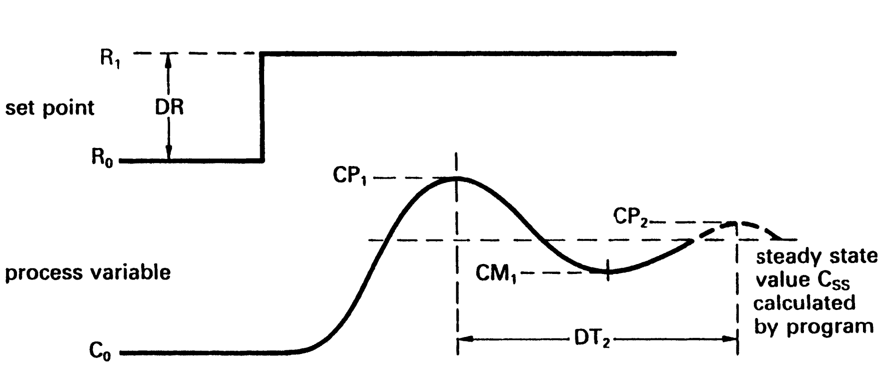


Figure 13.89 Test performed for model building tuning test. Note that R_0 and C_0 need not be the same and DR can be positive or negative

The equations above are not very practical for manual use on site, so the original paper was developed into a program for the Hewlett Packard HP-67 calculator by Jutan and Rodriguez and published in the magazine *Chemical Engineering September* 1984. The nomenclature used in the above equations is based on this article.

13.30.6 General comments

The above test procedures do not give guaranteed results and should be viewed as a method of putting the engineer in the right area. They should be viewed as the starting point for further trials. The important thing in these trials is only change one thing at once.

With values set as above the effect of changing the gain should be tried first. It is always useful to have the proportional gain as high as possible to give the largest initial control action to changes and disturbances. However, a large gain can give undesirable changes in the controller output if the process variable is noisy. The gain should be adjusted to give the desired overshoot and damping.

Integral action should be adjusted next to give best removal of offset error. During these trials it is best to disable any derivative action. Decreasing the integral time reduces the time taken to remove the offset error. It may be necessary to reduce the gain again as integral time is decreased. A useful rule of thumb is that the ratio of T_i/Gain is an 'index' of stability for a given system, i.e. a T_i of 12 sec and a gain of 2 will give a similar damping to a T_i of 24 sec and a gain of 4.

The derivative action should be adjusted last. Many systems do not benefit from derivative action, particularly those with a noisy process variable signal which causes large controller output swings. Where derivative action is required, $T_\mathrm{d} = \rlap{4}T_\mathrm{i}/4$ is a good starting point. Many controllers allow the user to select derivative action on error or derivative action on process variable. The former is best for tracking systems, but gives large controller output swings for step changes in the set point. Derivative action on process variable is usually the best choice.

One final observation, based on experience rather than theory, is that a P_B of 200% (gain of 0.5), T_i of 20 sec and no derivative action is a good starting point for a majority of plants. Adjust the gain to give the required overshoot then adjust $\mathrm{T_i}$ to be as small as possible. Finally set T_d, if needed, to $T_\mathrm{i}/4$.

References

1 SAUCEDO, R. and SCHIRING, E. E., *Introduction to Continuous and Digital Control Systems*, Macmillan, New York (1968)

2 FRANKLIN, G. F., POWELL, J. D. and EMAMI-NAEINI, A., *Feedback Control of Dynamic Systems*, Addison-Wesley, New York (1986)

Bibliography

The authors have found the following books useful for basic control engineering studies. This list is by no means exhaustive.

ANAND, D. K., *Introduction to Control Systems*, Pergamon Press, Oxford (1974)

CHEN, C. F. and HAAS, I. J., *Elements of Control Systems Analysis*, Prentice-Hall, Englewood Cliff, NJ (1968)

DISTEFANO,J.J.,STUBBERUD,A.R.andWILLIAMS,I.J., *Theory and Problems of Feedback and Control Systems*, Schaum's Outline Series, McGraw-Hill, New York (1990)

DORF, R. C., *Modern Control Systems*, Addison-Wesley, New York (1980)

DOUCE, J. L., *The Mathematics of Servomechanisms*, English Universities Press, London (1963)

ELGARD, O. I., *Control Systems Theory*, McGraw-Hill, New York (1967)

GOLTEN, J. and VERWER, A., *Control System Design and Simulation*, McGraw Hill (1991)

HEALEY, M., *Principles of Automatic Control*, Hodder and Stoughton, New York (1975)

JACOBS, O. L. R., *Introduction to Control Theory*, Oxford University Press, Oxford (1974)

LANGILL, A. W., *Automatic Control Systems Engineering*, Vols I and II, Prentice-Hall, Englewood Cliffs, NJ (1965)

MARSHALL, S. A., *Introduction to Control Theory*, Macmillan, New York (1978)

POWER, H. M. and SIMPSON, R. J., *Introduction to Dynamics and Control*, McGraw-Hill, New York (1978)

RAVEN, F. H., *Automatic Control Engineering*, McGraw-Hill, New York (1961)

SHINSKEY, F. G., *Process Control Systems*, McGraw Hill (1988)