Derek P. Atherton

Control Engineering

An introduction with the use of Matlab

Derek Atherton

Control Engineering

An introduction with the use of Matlab

Control Engineering: An introduction with the use of Matlab
2nd edition

ISBN 978-87-403-0473-2

Contents

Preface

It is almost four years since the first edition of this book so it seemed appropriate to reread it carefully again and make any suitable changes. Also during the intervening period I have added two further bookboon books one on 'An Introduction to Nonlinearity in Control Systems' and another very recently on 'Control Engineering Problems with Solutions'. This later book contains worked examples and some problems with answers only, which cover the material in this book and 'An Introduction to Nonlinearity in Control Systems'. It is hoped that the relevant chapters of 'Control Engineering Problems with Solutions' will help the reader gain a better understanding and deeper knowledge of the topics covered in this textbook.

Minor changes have been made to this second edition mainly with respect to a few changes in wording, but sadly despite repeated reading a few minor technical errors were found and corrected, for which I apologise. These were Figure 3.6 which had some incorrect markings and was not very clear due to the numbers chosen giving lines almost on top of each other. This has been corrected by choosing a different frequency for illustrating the frequency response calculation procedure. Further, some negative signs were omitted from equation (2.14), the units of H on page 50 were given incorrectly as were the subscripts on the a's and a matrix in the material in section 10.5.1, page 131, on transforming to the controllable canonical form. Finally the cover page has been changed to contain a picture which is more relevant to the book.

Derek P. Atherton

Brighton, June 2013.

About the author

Professor Derek. P. Atherton
BEng, PhD, DSc, CEng, FIEE, FIEEE, HonFInstMC, FRSA

Derek Atherton studied at the universities of Sheffield and Manchester, obtaining a PhD in 1962 and DSc in 1975 from the latter. He spent the period from 1962 to 1980 teaching in Canada where he served on several National Research Council committees including the Electrical Engineering Grants Committee.

He took up the post of Professor of Control Engineering at the University of Sussex in 1980 and is currently retired but has an office at the university, gives some lectures, and has the title of Emeritus Professor and Associate Tutor. He has been active with many professional engineering bodies, serving as President of the Institute of Measurement and Control in 1990, President of the IEEE Control Systems Society in 1995, being the only non North American to have held the position, and as a member of the IFAC Council from 1990–96. He served as an Editor of the IEE Proceedings on Control Theory and Applications (CTA) for several years until 2007 and was formerly an editor for the IEE Control Engineering Book Series. He has served EPSRC on research panels and as an assessor for research grants for many years and also served as a member of the Electrical Engineering Panel for the Research Assessment Exercise in 1992.

His major research interests are in non-linear control theory, computer aided control system design, simulation and target tracking. He has written two books, is a co-author of two others and has published more than 350 papers in Journals and Conference Proceedings. Professor Atherton has given invited lectures in many countries and supervised over 30 Doctoral students.

Derek P. Atherton.
February 2009.

1 Introduction

1.1 What is Control Engineering?

As its name implies control engineering involves the design of an engineering product or system where a requirement is to accurately control some quantity, say the temperature in a room or the position or speed of an electric motor. To do this one needs to know the value of the quantity being controlled, so that being able to measure is fundamental to control. In principle one can control a quantity in a so called open loop manner where 'knowledge' has been built up on what input will produce the required output, say the voltage required to be input to an electric motor for it to run at a certain speed. This works well if the 'knowledge' is accurate but if the motor is driving a pump which has a load highly dependent on the temperature of the fluid being pumped then the 'knowledge' will not be accurate unless information is obtained for different fluid temperatures. But this may not be the only practical aspect that affects the load on the motor and therefore the speed at which it will run for a given input, so if accurate speed control is required an alternative approach is necessary.

This alternative approach is the use of feedback whereby the quantity to be controlled, say C, is measured, compared with the desired value, R, and the error between the two,

E = R – C used to adjust C. This gives the classical feedback loop structure of Figure 1.1.

In the case of the control of motor speed, where the required speed, R, known as the reference is either fixed or moved between fixed values, the control is often known as a regulatory control, as the action of the loop allows accurate speed control of the motor for the aforementioned situation in spite of the changes in temperature of the pump fluid which affects the motor load. In other instances the output C may be required to follow a changing R, which for example, might be the required position movement of a robot arm. The system is then often known as a servomechanism and many early textbooks in the control engineering field used the word servomechanism in their title rather than control.

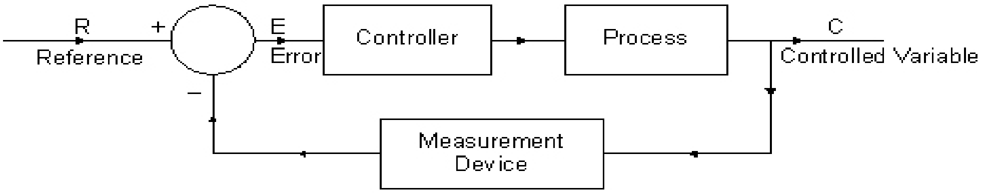

Figure 1.1 Basic Feedback Control Structure

The use of feedback to regulate a system has a long history [1.1, 1.2], one of the earliest concepts, used in Ancient Greece, was the float regulator to control water level, which is still used today in water tanks. The first automatic regulator for an industrial process is believed to have been the flyball governor developed in 1769 by James Watt. It was not, however, until the wartime period beginning in 1939, that control engineering really started to develop with the demand for servomechanisms for munitions fire control and guidance. With the major improvements in technology since that time the applications of control have grown rapidly and can be found in all walks of life. Control engineering has, in fact, been referred to as the 'unseen technology' as so often people are unaware of its existence until something goes wrong. Few people are, for instance, aware of its contribution to the development of storage media in digital computers where accurate head positioning is required. This started with the magnetic drum in the 50's and is required today in disk drives where position accuracy is of the order of 1μm and movement between tracks must be done in a few ms.

Feedback is, of course, not just a feature of industrial control but is found in biological, economic and many other forms of system, so that theories relating to feedback control can be applied to many walks of life.

1.2 Contents of the Book

The book is concerned with theoretical methods for continuous linear feedback control system design, and is primarily restricted to single-input single-output systems. Continuous linear time invariant systems have linear differential equation mathematical models and are always an approximation to a real device or system. All real systems will change with time due to age and environmental changes and may only operate reasonably linearly over a restricted range of operation. There is, however, a rich theory for the analysis of linear systems which can provide excellent approximations for the analysis and design of real world situations when used within the correct context. Further, simulation is now an excellent means to support linear theoretical studies as model errors, such as the affects of neglected nonlinearity, can easily be assessed.

There are a total of 11 chapters and some appendices, the major one being Appendix A on Laplace transforms. The next chapter provides a brief description of the forms of mathematical model representations used in control engineering analysis and design. It does not deal with mathematical modelling of engineering devices, which is a huge subject and is best dealt with in the discipline covering the subject, since the devices or components could be electrical, mechanical, hydraulic etc. Suffice to say that one hopes to obtain an approximate linear mathematical model for these components so that their effect in a system can be investigated using linear control theory. The mathematical models discussed are the linear differential equation, the transfer function and a state space representation, together with the notations used for them in MATLAB.

Chapter 3 discusses transfer functions, their zeros and poles, and their responses to different inputs. The following chapter discusses in detail the various methods for plotting steady state frequency responses with Bode, Nyquist and Nichols plots being illustrated in MATLAB. Hopefully sufficient detail, which is brief when compared with many textbooks, is given so that the reader clearly understands the information these plots provide and more importantly understands the form of frequency response expected from a specific transfer function.

The material of chapters 2–4 could be covered in other courses as it is basic systems theory, there having been no mention of control, which starts in chapter 5. The basic feedback loop structure shown in Figure 1.1 is commented on further, followed by a discussion of typical performance specifications which might have to be met in both the time and frequency domains. Steady state errors are considered both for input and disturbance signals and the importance and properties of an integrator are discussed from a physical as well as mathematical viewpoint. The chapter concludes with a discussion on stability and a presentation of several results including the Mikhailov criterion, which is rarely mentioned in English language texts. Chapter 6 first introduces the properties of a time delay before continuing with further material relating to the analysis and properties of the closed loop. Briefly mentioned are the root locus and its plotting using MATLAB and various concepts of relative stability. These include gain and phase margins, sensitivity functions, and M and N circles.

Chapter 7 is a relatively long chapter dealing with classical controller design methods. The basic concept of classical control design is that one decides on a suitable control strategy and then the design problem becomes one of obtaining appropriate parameters for the controller elements in order to meet specified control performance objectives. Typically a controller with a specified structure is placed in either the forward or feedback paths, or even both, of the closed loop. The first point discussed is therefore the difference between a feedforward and a feedback controller on the closed loop transfer function. The design of lead and lag controllers is then discussed followed by a long section on PID control, a topic on which far too much has probably been written in the literature in recent years due in no part to its extensive use in practice. The early work of Ziegler and Nichols is the starting point which largely focuses on the control of a plant with a time constant plus time delay. By dealing with this plant in so called normalised form, where its behaviour is expressible in terms of the time delay to time constant ratio, new results are presented comparing various suggested parameter settings, usually known as tuning, for PID controllers. It is pointed out that if a mathematical model is obtained for the plant then the principles and possibilities for obtaining parameters for a PID controller are no different to those which may be used for any other type of controller. However a major contribution of Ziegler and Nichols in their loop cycling method was to show how the PID controller parameters might be chosen without a mathematical model, but simply from knowledge of the so called plant transfer function critical point, namely the magnitude and frequency of the transfer function for 180° phase shift. Its modern equivalent is known as relay autotuning and this topic is covered in some detail at the end of the chapter.

The controller design concepts presented in the previous chapter based on open loop frequency response compensation were regularly used in the early days of control engineering by designers who were adept at sketching Bode diagrams, so that the use of modern software has simply brought more efficiency to the design process. Some significant theoretical work on optimising controller parameters to meet specific performance criteria was also done in the early days but here the limitation was the difficulty of using the theory to obtain results of significance. With modern computation tools numerical approaches can be used to solve these problems either by writing MATLAB programs based on linear system theory or writing optimisation programs around digital simulations in programs such as SIMULINK. These are appropriate industrial design methods which appear to receive little attention in textbooks, possibly because they are not suitable for traditional examinations. Chapter 8 covers parameter optimisation based on integral performance criteria because it allows some simple results to be obtained and concepts understood. Further it leads to a design approach based on closed loop transfer function synthesis, known as standard forms, presented at the end of the chapter. Chapter 9 discusses further aspects of classical controller design and highlights the difficulty of trying to design series compensators for, so called uncertain plants, plants whose parameters may vary or not be accurately known. This leads to consideration of some elegant recent results on uncertain plants but which unfortunately appear too conservative for practical use in many instances.

The final two chapters are concerned with the use of state space methods in control system analysis and design. Chapter 10 provides basic coverage of state space concepts covering state equations and their solution, state transformations, state representations of transfer functions, and controllability and observability. Some state space design methods are covered in Chapter 11, including state variable feedback, LQR design and state variable feedback design to achieve the closed loop standard forms of chapter 8.

1.3 References

Bennett, S. A history of Control Engineering, 1800–1930. IEE control engineering series. Peter Peregrinus, 1979.

Bennett, S. A history of Control Engineering, 1930–1955. IEE control engineering series. Peter Peregrinus, 1993.

2 Mathematical Model Representations of Linear Dynamical Systems

2.1 Introduction

Control systems exist in many fields of engineering so that components of a control system may be electrical, mechanical, hydraulic etc. devices. If a system has to be designed to perform in a specific way then one needs to develop descriptions of how the outputs of the individual components, which make up the system, will react to changes in their inputs. This is known as mathematical modelling and can be done either from the basic laws of physics or from processing the input and output signals, in which case it is known as identification. Examples of physical modelling include deriving differential equations for electrical circuits involving resistance, inductance and capacitance and for combinations of masses, springs and dampers in mechanical systems. It is not the intent here to derive models for various devices which may be used in control systems but to assume that a suitable approximation will be a linear differential equation. In practice an improved model might include nonlinear effects, for example Hooke's Law for a spring in a mechanical system is only linear over a certain range; or account for time variations of components. Mathematical models of any device will always be approximate, even if nonlinear effects and time variations are also included by using more general nonlinear or time varying differential equations. Thus, it is always important in using mathematical models to have an appreciation of the conditions under which they are valid and to what accuracy.

Starting therefore with the assumption that our model is a linear differential equation then in general it will have the form:-

$$A(D)y(t) = B(D)u(t) \tag{2.1}$$

where D denotes the differential operator d/dt. $A(D)$ and $B(D)$ are polynomials in D with $D^i = d^i / dt^i$, the i^{th} derivative, $u(t)$ is the model input and $y(t)$ its output. So that one can write

$$A(D) = D^n + a_{n-1}D^{n-1} + a_{n-2}D^{n-2} ....a_1 D + a_0 \tag{2.2}$$

$$B(D) = D^m + b_{m-1}D^{m-1} + b_{m-2}D^{m-2} ....b_1 D + b_0 \tag{2.3}$$

where the a and b coefficients will be real numbers. The orders of the polynomials A and B are assumed to be n and m, respectively, with $n \geq m$.

Thus, for example, the differential equation

$$\frac{d^2 y}{dt^2} + 4\frac{dy}{dt} + 3y = 2\frac{du}{dt} + u \tag{2.4}$$

with the dependence of *y* and *u* on *t* assumed can be written

$$(D^2 + 4D + 3)y = (2D + 1)u \tag{2.5}$$

In order to solve an n^{th} order differential equation, that is determine the output *y* for a given input *u*, one must know the initial conditions of *y* and its first *n-1* derivatives. For example if a projectile is falling under gravity, that is constant acceleration, so that D^2y= constant, where *y* is the height, then in order to find the time taken to fall to a lower height, one must know not only the initial height, normally assumed to be at time zero, but the initial velocity, *dy/dt*, that is two initial conditions as the equation is second order ($n = 2$). Control engineers typically study solutions to differential equations using either Laplace transforms or a state space representation.

2.2 The Laplace Transform and Transfer Functions

A short introduction to the Laplace transformation is given in Appendix A for the reader who is not familiar with its use. It is an integral transformation and its major, but not sole use, is for differential equations where the independent time variable *t* is transformed to the complex variable *s* by the expression

$$F(s) = \int_0^\infty f(t)e^{-st}dt \tag{2.6}$$

Since the exponential term has no units the units of *s* are seconds^{-1}, that is using mks notation *s* has units of s^{-1}. If $\mathscr{L}$ denotes the Laplace transform then one may write $\mathscr{L}[f(t)] = F(s)$ and $\mathscr{L}^{-1}[F(s)] = f(t)$. The relationship is unique in that for every *f*(*t*), [*F*(*s*)], there is a unique *F*(*s*), [*f*(*t*)]. It is shown in Appendix A that when the *n*-1 initial conditions, $D^{n-1}y(0)$ are zero the Laplace transform of $D^n y(t)$ is $s^n Y(s)$. Thus the Laplace transform of the differential equation (2.1) with zero initial conditions can be written

$$A(s)Y(s) = B(s)U(s) \tag{2.7}$$

or simply

$$A(s)Y = B(s)U \tag{2.8}$$

with the assumed notation that signals as functions of time are denoted by lower case letters and as functions of *s* by the corresponding capital letter.

If equation (2.8) is written

$$\frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)} = G(s) \tag{2.9}$$

then this is known as the transfer function, *G*(*s*), between the input and output of the ‘system’, that is whatever is modelled by equation (2.1). *B*(*s*), of order *m*, is referred to as the numerator polynomial and *A*(*s*), of order *n*, as the denominator polynomial and are from equations (2.2) and (2.3)

$$A(s) = s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2}.........a_1 s + a_0 \tag{2.10}$$

$$B(s) = s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2}........b_1 s + b_0 \tag{2.11}$$

Since the *a* and *b* coefficients of the polynomials are real numbers the roots of the polynomials are either real or complex pairs. The transfer function is zero for those values of *s* which are the roots of *B*(*s*), so these values of *s* are called the zeros of the transfer function. Similarly, the transfer function will be infinite at the roots of the denominator polynomial *A*(*s*), and these values are called the poles of the transfer function. The general transfer function (2.9) thus has *m* zeros and *n* poles and is said to have a relative degree of *n-m*, which can be shown from physical realisation considerations cannot be negative. Further for $n > m$ it is referred to as a strictly proper transfer function and for $n \geq m$ as a proper transfer function.

When the input $u(t)$ to the differential equation of (2.1) is constant the output $y(t)$ becomes constant when all the derivatives of the output are zero. Thus the steady state gain, or since the input is often thought of as a signal the term d.c. gain (although it is more often a voltage than a current!) is used, and is given by

$$G(0) = b_0 / a_0 \tag{2.12}$$

If the n roots of $A(s)$ are $\alpha_{i,}$ $i = 1....n$ and of $B(s)$ are β_j, $j = 1....m$, then the transfer function may be written in the zero-pole form

$$G(s) = \frac{K\prod_{j=1}^{m}(s-\beta_j)}{\prod_{i=1}^{n}(s-\alpha_i)} \tag{2.13}$$

where in this case

$$G(0) = \frac{K\prod_{j=1}^{m}-\beta_j}{\prod_{i=1}^{n}-\alpha_i} \tag{2.14}$$

When the transfer function is known in the zero-pole form then the location of its zeros and poles can be shown on an s plane zero-pole plot, where the zeros are marked with a circle and the poles by a cross. The information on this plot then completely defines the transfer function apart from the gain K. In most instances engineers prefer to keep any complex roots in quadratic form, thus for example writing

$$G(s) = \frac{4(s+1)}{(s+2)(s^2+s+1)} \tag{2.15}$$

rather than writing $(s + 0.5 + j0.866)(s + 0.5 - j0.866)$ for the quadratic term in the denominator. This transfer function has $K = 4$, a zero at -1, three poles at -2, -0.5 ± 0.866 respectively, and the zero-pole plot is shown in Figure 2.1

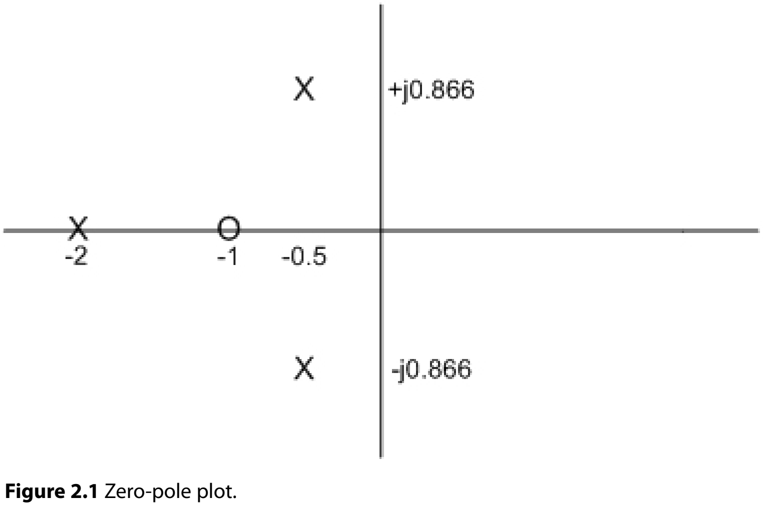

Figure 2.1 Zero-pole plot.

2.3 State space representations

Consider first the differential equation given in equation (2.4) but without the derivative of *u* term, that is

$$\frac{d^2 y}{dt^2} + 4\frac{dy}{dt} + 3y = u \tag{2.16}$$

To solve this equation, as mentioned earlier, one must know the initial values of *y* and *dy/dt*, or put another way the initial state of the system. Let us choose therefore to represent *y* and *dy/dt* by x_1 and x_2 the components of a state vector ***x*** of order two. Thus we have $\dot{x}_1 = x_2$, by choice, and from substitution in the differential equation $\dot{x}_2 = -4x_2 - 3x_1 + u$. The two equations can be written in the matrix form

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -3 & -4 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \tag{2.17}$$

and the output *y* is simply, in this case, the state x_1 and can be written

$$y = (1 \;\; 0)x \tag{2.18}$$

For this choice of state vector the representation is often known as the phase variable representation. The solution for no input, that is $u = 0$, from an initial state can be plotted in an x_1-x_2 plane, known as a phase plane with time a parameter on the solution trajectory. Equation (2.17) is a state equation and (2.18) an output equation and together they provide a state space representation of the differential equation or the system described by the differential equation.

Since this system has one input, u, and one output, y, it is often referred to as a single-input single-output (SISO) system. The choice of the state variable $\boldsymbol{x}$ is not unique and more will be said on this later, but the point is easily illustrated by considering the simple *R*-*C* circuit in Figure 2.2. If one derives the differential equation for the output voltage in terms of the input voltage, it will be a second order one similar to equation (2.16) and one could choose as in that equation the output, the capacitor voltage, and its derivative as the components of the state variable, or simply the states, to have a representation similar to equation (2.17). From a physical point of view, however, any initial non zero state will be due to charge stored in one or both of the two capacitors and therefore it might be more appropriate to choose the voltages of these two capacitors as the states.

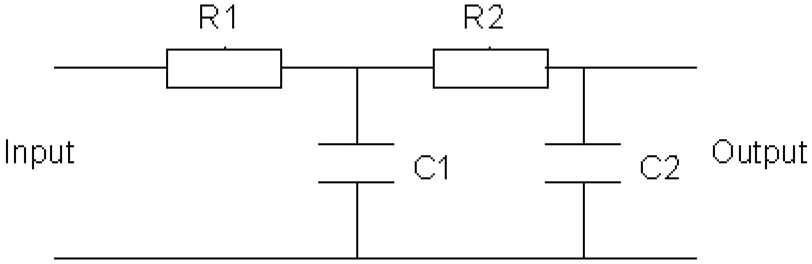

Figure 2.2 Simple *R*-*C* circuit.

In the state space representation of (2.17) and (2.18) x_1 is the same as y so that for the state equation (2.18) the transfer function between $U(s)$ and $X_1(s)$ is obviously

$$\frac{X_1(s)}{U(s)} = \frac{1}{s^2 + 4s + 3} \tag{2.19}$$

That is x_1 replacing y in the transfer function corresponding to the differential equation (2.16). Now the transfer function corresponding to equation (2.5) is

$$\frac{Y(s)}{U(s)} = \frac{2s + 1}{s^2 + 4s + 3} \tag{2.20}$$

which can be written as

$$\frac{Y(s)}{U(s)} = 2sX_1(s) + X_1(s) \tag{2.21}$$

Since in our state representation $\dot{x}_1 = x_2$, which in transform terms is $sX_1(s) + X_2(s)$, this means in this case with the same state equation the output equation is now $y = 2x_2 + x_1$. Thus a state space representation for equation (2.5) is

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -3 & -4 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u\,,\ \ y = \begin{pmatrix} 1 & 2 \end{pmatrix} x \tag{2.22}$$

It is easy to show that for the more general case of the differential equation (2.1) a possible state space representation, which is known as the controllable canonical form, illustrated for $m < n$-1, is

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & . & . & . & . & 0 \\ 0 & 0 & 1 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 1 & 0 & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & 0 & 1 & 0 \\ . & . & . & . & . & . & . & 1 \\ -a_0 & -a_1 & -a_2 & . & . & . & . & -a_{n-1} \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ 1 \end{pmatrix} u \tag{2.23}$$

$$y = \begin{pmatrix} b_0 & b_1 & \ldots & b_{m-1} & 1 & \ldots.. & 0 \end{pmatrix} x \tag{2.24}$$

In matrix form the state and output equations can be written

$$\dot{x} = Ax + Bu \qquad y = Cx \tag{2.25}$$

where the state vector, $\boldsymbol{x}$, is of order n, the A matrix is nxn, B is a column vector of order, n, and C is a row vector of order, n. Because B and C are vectors for the SISO system they are often denoted by $\boldsymbol{b}$ and $\boldsymbol{c}^{\mathrm{T}}$, respectively. Also in the controllable canonical form representation given above the A matrix and B vector take on specific forms, the former having the pole polynomial coefficients in the last row and the latter being all zeros apart from the unit value in the last row. If m and n are of the same order, for example if they are both 2 and the corresponding transfer function is $\frac{s^2+5s+6}{s^2+4s+3}$, then this can be written as $1+\frac{s+3}{s^2+4s+3}$, which means there is a unit gain direct transmission between input and output, then the state representation takes the more general form

$$\dot{x} = Ax + Bu \qquad y = Cx + Du \tag{2.26}$$

where D is a scalar, being unity of course in the above example. A state space representation can be used for a mathematical model of a system with multiple inputs and outputs, denoted by MIMO, and in this case B, C and D will be matrices of appropriate dimensions which accounts for the use of capital letters.

Thus, in conclusion, a mathematical model of a linear SISO dynamical system may be a differential equation, a transfer function or a state space representation. A state space representation has a unique transfer function but the reverse is not the case.

2.4 Mathematical Models in MATLAB

MATLAB, although not the only language with good facilities for control system design, is easy to use and very popular. As well as tools for analysis it also contains a simulation language, SIMULINK, which is also very useful. If it has a weakness it is probably with regard to physical modelling but for the contents of this book, where our starting point is a mathematical model, this is not a problem. Models of system components can be entered into MATLAB either as transfer functions or state space representations. A model is an object defined by a symbol, say *G*, and its transfer function can be entered in the form ***G=tf(num,den)*** where ***num*** and ***den*** contain a string of coefficients describing the numerator and denominator polynomials respectively. MATLAB statements in the text, such as the above for *G*, will be entered in bold italics but not in program extracts such as that below. The coefficients are entered beginning with the highest power of *s*.

Thus the transfer function $G(s)=\frac{2s+1}{s^2+4s+3}$, can be entered by typing:-

```
>>num=[2 1];
>> den=[1 4 3];
>> G=tf(num,den)

Transfer function:

            2s + 1
        --------------
        s^2 + 4 s + 3
```

The >> is the MATLAB prompt and the semicolon at the end of a line suppresses a MATLAB response. This has been omitted from the expression for *G* so MATLAB responds with the transfer function *G* as shown. Alternatively, the entry could have been done in one expression by typing:-

```
>>G=tf([2 1],[1 4 3])
```

The roots of a polynomial can be found by typing ***roots*** before the coefficient string in square brackets. Thus typing:-

```
>> roots(den)

ans =

-3

-1
```

Alternatively the transfer function can be entered in zero, pole, gain form where the command is in the form ***G=zpk(zeros,poles,gain)***

Thus for the same example

```
>> G=zpk([-0.5],[-1;-3],2)

Zero/pole/gain:

         2 (s+0.5)
        -----------
        (s+1) (s+3)
```

where the values of zeros or poles in a string are separated by a semicolon. Also to enter a string with a single number, here used for the value of *K* but not for the single zero, the square brackets may be omitted.

A state space model or object formed from known A,B,C,D matrices, often denoted by (A,B,C,D),can be entered into MATLAB with the command ***G=ss(A,B,C,D)***.

Thus for the same example by entering the following commands one defines the state space model

```
>> A=[0,1;-3,-4];
>> B=[0;1];
>> C=[1,2];
>> D=0;
>> G=ss(A,B,C,D);
```

And asking afterwards for the transfer function of the model by typing

```
>> tf(G)
```

One obtains

```
Transfer function:

      2 s + 1
    -------------
    s^2 + 4 s + 3
```

Obviously the above have been very simple examples but hopefully they have covered the basics of putting the mathematical model of a linear dynamical system into MATLAB. The only way to learn is by doing examples and since MATLAB has an excellent **help** facility the reader should not find this difficult. For a more extensive coverage of MATLAB routines and examples of their use in control engineering the reader is referred to the book given in reference 2.1.

2.5 Interconnecting Models in MATLAB

Control systems are made up of several components, so as well as describing a component by a mathematical model, one needs to deal with the mathematical models for interconnected components. Typically a component is represented as a block with input and output signals and labelled, usually with a transfer function, say $G_1(s)$, as shown in Figure 2.3. Strictly speaking if the block is labelled with a transfer function the input and output signals should also be in the s domain, as the block in Figure 2.3 implies

$$Y(s) = G_1(s)U(s) \tag{2.27}$$

but it is usually accepted that the time domain notations, $y(t)$ and $u(t)$ for the signals, may also be used.

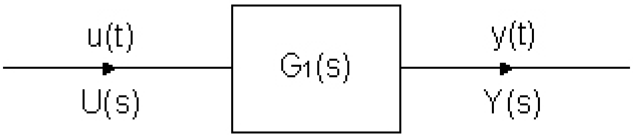

Figure 2.3 Block representation of a transfer function

When a second block, with transfer function $G_2(s)$, is connected to the output of the first block, to give a series connection, then it is assumed that in making the connection of Figure 2.4 that the second block does not affect the output of the first one. In this case the resultant transfer function of the series combination between input u and output y is $G_1(s)G_2(s)$, which is obtained directly by substitution from the individual block relationships $X(s)=G_1(s)U(s)$ and $Y(s)=G_2(s)X(s)$ where x is the output of the first block.

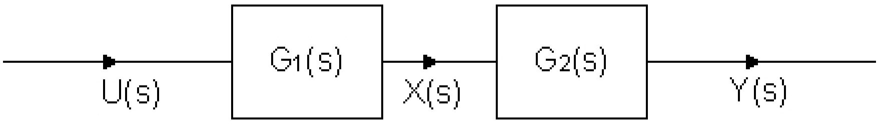

Figure 2.4 Series (or cascade) connection of blocks

If two system objects G_1 and G_2 are provided to MATLAB then the system object corresponding to the series combination can be obtained by typing $\boldsymbol{G=G_1{*}G_2}$.

If two transfer function models, $G_1(s)$ and $G_2(s)$ are connected in parallel, as shown in Figure 2.5, then the resultant transfer function between the input u and output y is obtained from the relationships $X_1(s) = G_1(s)U(s)$, $X_2(s) = G_2(s)U(s)$ and $Y(s) = X_1(s)+X_2(s)$ and is $G_1(s)+G_2(s)$. It can be obtained in MATLAB by typing $\boldsymbol{G=G_1+G_2}$

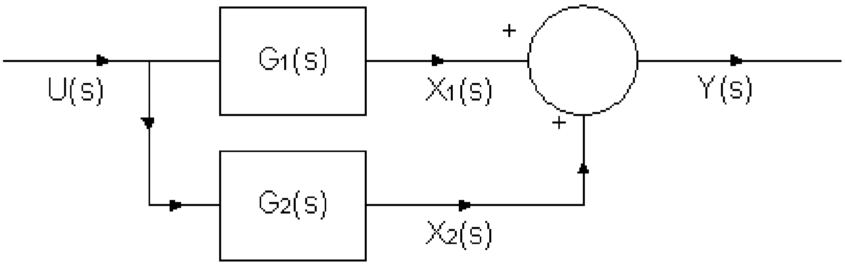

Figure 2.5 Parallel connection of blocks

Another connection of blocks which will be used is the feedback connection shown in Figure 2.6. For the negative feedback connection of Figure 2.6 the relationship is $Y(s) = G(s)[U(s) - H(s)Y(s)]$, where the expression in the square brackets is the input to $G(s)$. This can be rearranged to give a transfer function between the input u and output y of

$$\frac{Y(s)}{U(s)} = \frac{G(s)}{1+G(s)H(s)}. \tag{2.28}$$

If this transfer function is denoted by $T(s)$ then the MATLAB command to obtain $T(s)$ is ***T=feedback(G,H).*** If the positive feedback configuration is required then the statement ***T=feedback(G,H,sign)*** can be used where the *sign* = 1. This can also be used for the negative feedback with *sign* = -1

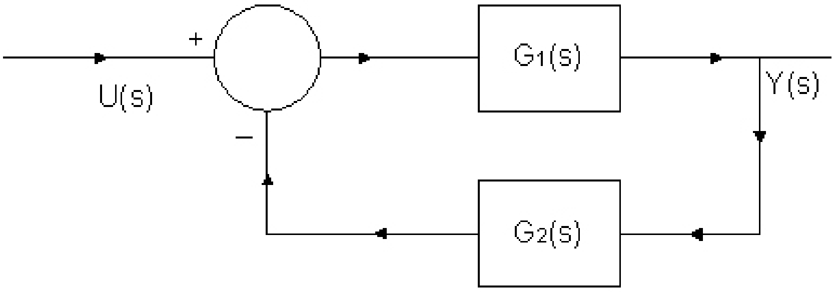

Figure 2.6 Feedback connection of blocks.

2.6 Reference

2.1 Xue D., Chen Y. and Atherton D.P. Linear Feedback Control: Analysis and Design in MATLAB, Siam, USA, 2007.

3 Transfer Functions and Their Responses

3.1 Introduction

As mentioned previously a major reason for wishing to obtain a mathematical model of a device is to be able to evaluate the output in response to a given input. Using the transfer function and Laplace transforms provides a particularly elegant way of doing this. This is because for a block with input $U(s)$ and transfer function $G(s)$ the output $Y(s) = G(s)U(s)$. When the input, $u(t)$, is a unit impulse which is conventionally denoted by $\delta(t)$, $U(s) = 1$ so that the output $Y(s) = G(s)$. Thus in the time domain, $y(t) = g(t)$, the inverse Laplace transform of $G(s)$, which is called the impulse response or weighting function of the block. The evaluation of $y(t)$ for any input $u(t)$ can be done in the time domain using the convolution integral (see Appendix A, theorem (ix))

$$y(t) = \int_0^t g(\tau)u(t-\tau)dt \tag{3.1}$$

but it is normally much easier to use the transform relationship $Y(s) = G(s)U(s)$. To do this one needs to find the Laplace transform of the input $u(t)$, form the product $G(s)U(s)$ and then find its inverse Laplace transform. $G(s)U(s)$ will be a ratio of polynomials in s and to find the inverse Laplace transform, the roots of the denominator polynomial must be found to allow the expression to be put into partial fractions with each term involving one denominator root (pole). Assuming, for example, the input is a unit step so that $U(s) = 1/s$ then putting $G(s)U(s)$ into partial fractions will result in an expression for $Y(s)$ of the form

$$Y(s) = \frac{C_0}{s} + \sum_{i=1}^{n} \frac{C_i}{s-\alpha_i} \tag{3.2}$$

where in the transfer function $G(s) = B(s)/A(s)$, the n poles of $G(s)$ [zeros of $A(s)$] are α_i, $i = 1\ldots n$ and the coefficients C_0 and C_i, $i = 1\ldots n$, will depend on the numerator polynomial $B(s)$, and are known as the residues at the poles. Taking the inverse Laplace transform yields

$$y(t) = C_0 + \sum_{i=1}^{n} C_i e^{\alpha_i t} \tag{3.3}$$

The first term is a constant C_0, sometimes written $C_0u_0(\mathrm{t})$ because the Laplace transform is defined for $t \geq 0$, where $u_0(\mathrm{t})$ denotes the unit step at time zero. Each of the other terms is an exponential, which provided the real part of α_i is negative will decay to zero as t becomes large. In this case the transfer function is said to be stable as a bounded input has produced a bounded output. Thus a transfer function is stable if all its poles lie in the left hand side (lhs) of the s plane zero-pole plot illustrated in Figure 2.1. The larger the negative value of α_i the more rapidly the contribution from the i^{th} term decays to zero. Since any poles which are complex occur in complex pairs, say of the form $\alpha_1,\alpha_2 = \sigma \pm j\omega$, then the corresponding two residues C_1 and C_2 will be complex pairs and the two terms will combine to give a term of the form $Ce^{\sigma t} \sin(\omega t + \varphi)$. This is a damped oscillatory exponential term where σ, which will be negative for a stable transfer function, determines the damping and ω the frequency [strictly angular frequency] of the oscillation. For a specific calculation most engineers, as mentioned earlier, will leave a complex pair of roots as a quadratic factor in the partial factorization process, as illustrated in the Laplace transform inversion example given in Appendix A. For any other input to a stable $G(s)$, as with the step input, the poles of the Laplace transform of the input will occur in a term of the partial fraction expansion (3.2), [as for the C_0/s term above], and will therefore produce a bounded output for a bounded input.

3.2 Step Responses of Some Specific Transfer Functions

In control engineering the major deterministic input signals that one may wish to obtain responses to are a step, an impulse, a ramp and a constant frequency input. The purpose of this section is to discuss step responses of specific transfer functions, hopefully imparting an understanding of what can be expected from a knowledge of the zeros and poles of the transfer function without going into detailed mathematics.

3.1.1 A Single Pole Transfer Function

A transfer function with a single pole is $G(s) = \frac{K_1}{s+a}$, which may also be written in the so-called time constant form $G(s) = \frac{K}{1+sT}$, where $K = K_1/a$ and $T = 1/a$ The steady state gain $G(0) = K$, which is the final value of the response to a unit step input, and T is called the time constant as it determines the speed of the response. K will have units relating the input quantity to the output quantity, for example °C/V, if the input is a voltage and the output temperature. T will have the same units of time as s^{-1}, normally seconds. The output, $Y(s)$, for a unit step input is given by

$$Y(s) = \frac{K}{s(1+sT)} = \frac{K}{s} - \frac{KT}{(1+sT)} \qquad (3.4).$$

Taking the inverse Laplace transform gives the result

$$y(t) = K(1 - e^{-t/T}) \qquad (3.5)$$

The larger the value of T (i.e. the smaller the value of a), the slower the exponential response. It can easily be shown that $y(T) = 0.632K$, $\frac{dy(0)}{dt} = T$ and $y(5T) = 0.993K$ or in words, the output reaches 63.2% of the final value after a time T, the initial slope of the response is T and the response has essentially reached the final value after a time $5T$. The step response in MATLAB can be obtained by the command **step(num,den)**. The figure below shows the step response for the transfer function with $K = 1$ on a normalised time scale.

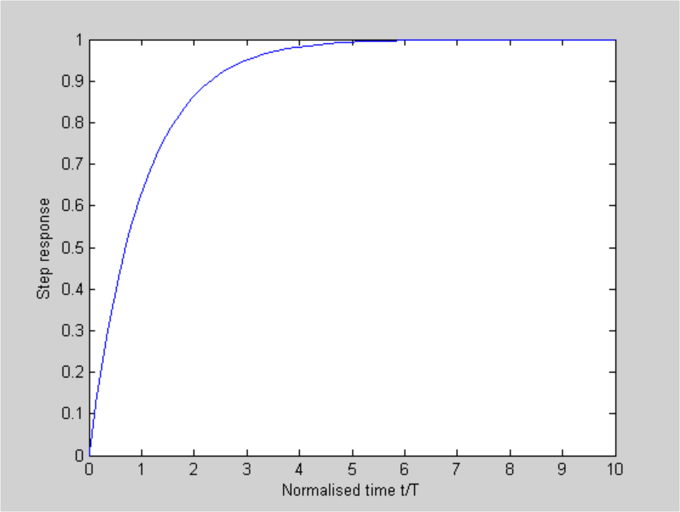

Figure 3.1 Normalised step response for a single time constant transfer function.

3.1.2 Two Complex Poles

Here the transfer function *G*(s) is often assumed to be of the form

$$G(s) = \frac{\omega_o^2}{s^2 + 2\zeta s\omega_o + \omega_o^2}. \tag{3..6}$$

It has a unit steady state gain, i.e $G(0) = 1$, and poles at $s = -\zeta\omega_o \pm j\omega_o\sqrt{1-\zeta^2}$, which are complex when $\zeta < 1$. For a unit step input the output *Y*(*s*), can be shown after some algebra, which has been done so that the inverse Laplace transforms of the second and third terms are damped cosinusoidal and sinusoidal expressions, to be given by

$$Y(s) = \frac{\omega_o^2}{s(s^2 + 2\zeta s\omega_o + \omega_o^2)} = \frac{1}{s} - \frac{s + \zeta\omega_o}{(s+\zeta\omega_o)^2 + \omega_o^2(1-\zeta^2)} - \frac{\zeta\omega_o}{(s+\zeta\omega_o)^2 + \omega_o^2(1-\zeta^2)} \tag{3.7}$$

Taking the inverse Laplace transform it yields, again after some algebra,

$$y(t) = 1 - \frac{e^{-\zeta\omega_o t}}{\sqrt{1-\zeta^2}}\sin(\sqrt{1-\zeta^2}\,\omega_o t + \varphi) \tag{3.8}$$

where $\varphi = \cos^{-1}\zeta$. ζ is known as the damping ratio. It can also be seen that the angle to the negative real axis from the origin to the pole with positive imaginary part is $\tan^{-1}(1-\zeta^2)^{1/2}/\zeta = \cos^{-1}\zeta = \varphi$. Measurement of the angle φ and this relationship is often used to refer to the damping of complex poles even when not dealing with a second order system. The response on the normalised time scale $\omega_0 t$ can be found from Matlab by taking ω_0 equal to one. The damping of the response then depends on ζ and the oscillatory behaviour on the normalised damped frequency, that is $\omega/\omega_o = \sqrt{1-\zeta^2}$. Figure 3.2 shows a normalised plot for several values of ζ.

The response can be shown to have the following properties:-

1) For $\zeta = 0$ the response is undamped and continues to oscillate with frequency ω_o (ω_o =1 on the normalised plot)
2) The overshoots and undershoots occur at half periods of the damped frequency, ω, that is times of $n\pi/\omega$, for integers n greater and equal to 1.
3) The first overshoot is $\Delta = e^{-\zeta\pi/\sqrt{1-\zeta^2}}$, then the undershoot is Δ^2, the next overshoot is Δ^3 and so on.
4) The overshoot is often given as a percentage, i.e.100 Δ, and is shown in Figure 3.3 as a function of ζ.
5) For $\zeta > 1$ the transfer function has two real poles and the response has no overshoot.
6) For ζ =1 both poles are at $-\ \omega_o$ and the response is the fastest with no overshoot.

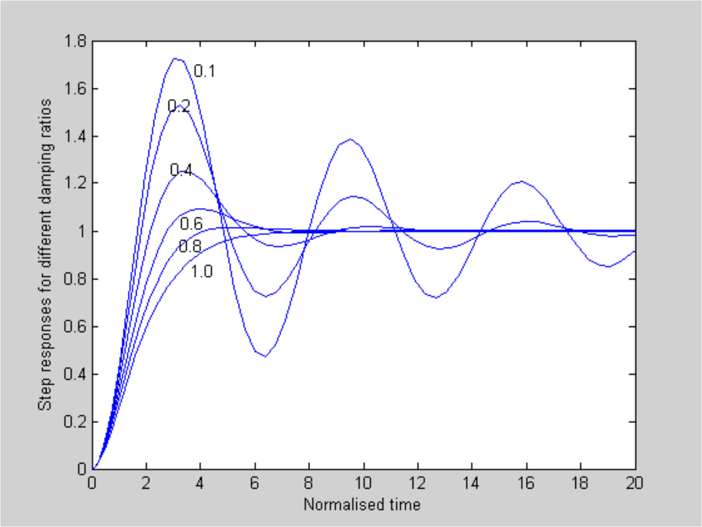

Figure 3.2 Normalised step response of second order system for different ς

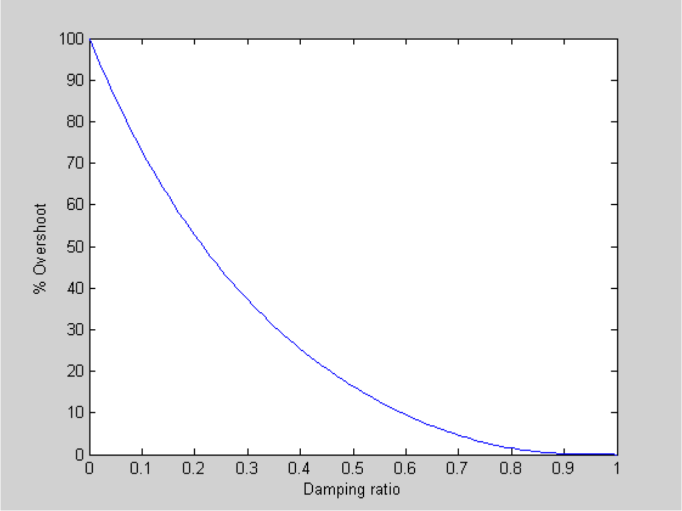

Figure 3.3 Graph of % overshoot as a function of the damping ratio.

3.1.3 The Effect of a Zero

Consider the general transfer function $G(s) = B(s)/A(s)$ again, and also $G_0(s) = 1/A(s)$, that is $G(s)$ with $B(s) = 1$. As outlined above the effect of a non-unity $B(s)$ will be to give different values of the C coefficients in the partial fraction expansion of equation (3.2). Thus one can find the new partial fraction expansion when $B(s)$ is not a constant and invert to find the time response. There is another way, however, which also helps in understanding the response and that is to recognise that s can be regarded as a derivative operator. Thus, for example, suppose the response of G_0(s) to a unit step input is y_0(t) then the response of G(s) to a unit step input can be written as

$$y(t) = \frac{d^m y_0(t)}{dt^m} + \frac{b_{m-1} d^{(m-1)} y_0(t)}{dt^{(m-1)}} + ...\frac{b_1 dy_0(t)}{dt} + b_0 y_0(t) \tag{3.9}$$

where the b's are the coefficients of $B(s)$ in equation (2.11).

To illustrate this consider

$$G(s) = \frac{1+sT}{(s+1)(s+2)} \quad \text{so that } Y_o(s) = \frac{1}{s(s+1)(s+2)} = \frac{1}{2s} - \frac{1}{2(s+1)} + \frac{1}{2(s+2)}$$

then the solution for $y_0(t)$ is $y_0(t) = 0.5(1 - e^{-t} + e^{-2t})$, which cannot have an overshoot as the exponentials decrease with increase in time. Using the above result $y(t)$ is given by

$$y(t) = 0.5(1 - e^{-t} + e^{-2t}) + 0.5T\frac{d(1 - e^{-t} + e^{-2t})}{dt}$$

It is easy to show mathematically that the response will have an overshoot for $T > 1$. The responses for $T = 0.5$, $T = 1$ and $T = 2$ are shown in Figure 3.4, obtained using the following MATLAB statements.

```
>> G0=tf([1],[1 3 2]);
>> step(G0)
>> hold
Current plot held
>> G1=tf([0.5 1],[1 3 2]);
>> G2=tf([1 1],[1 3 2]);
>> G3=tf([2 1],[1 3 2]);
>> step(G1)
>> step(G2)
>> step(G3)
```

where the ***hold*** statement keeps the plot allowing the responses to be compared.

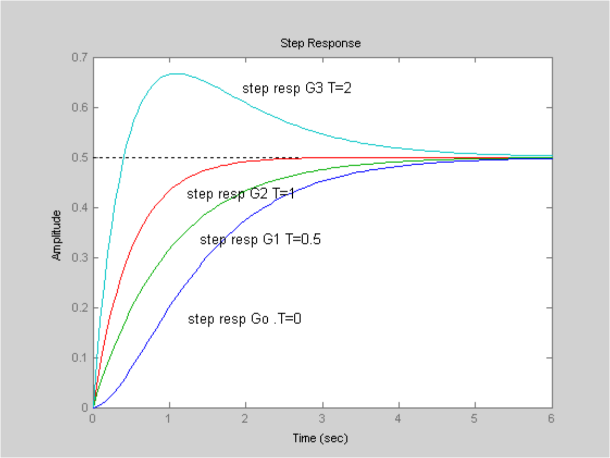

Figure 3.4 Step responses for various values of T

The unit impulse, δ(t), is the derivative of the unit step and has a Laplace transform of unity. Thus the response to a unit impulse is the derivative of the response to a unit step.

3.1.4 A 3 Pole Transfer Function

In order to appreciate the response from multiple poles consider the step responses of two transfer functions each with three poles, a real pole and a complex pair. The example transfer functions are written in factored form, which of course corresponds to transfer functions in parallel, and are:-

$$G_1(s) = \frac{0.5}{s^2 + 0.2s + 1} + \frac{0.1}{s + 0.2}$$

and

$$G_2(s) = \frac{0.5}{s^2 + 0.2s + 1} + \frac{2.5}{s + 5}.$$

Both transfer functions when written with a common denominator have two zeros and each term in G_1 and G_2 contributes a final value of 0.5, with the response from the complex poles the same. The step responses are shown in Figure 3.5. The time constant of the single pole in G_1 is 5 seconds but only 0.2 seconds in $G_{2.}$ Thus for the step response of G_1 the time constant slows the response down and the overshoot is not as large as it would be for the complex poles alone, although the response still oscillates. The smaller time constant of G_2 is evident in the rapid initial change in the step response.

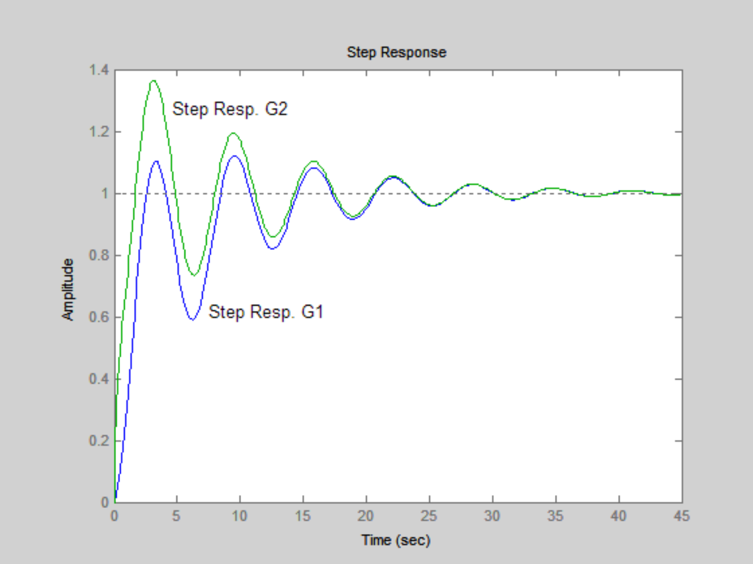

Figure 3.5 Step responses of 3 pole transfer functions.

3.3 Response to a Sinusoid

The Laplace transform of sin ωt is $\omega/(s^2 + \omega^2)$, so that when the partial fraction expansion is used to get $Y(s)$ it will now be of the form

$$Y(s) = \frac{C_{01} + sC_{02}}{s^2 + \omega^2} + \sum_{i=1}^{n} \frac{C_i}{s - \alpha_i} \tag{3.10}$$

For a stable transfer function, $G(s)$, all the exponential terms in the summation will eventually go to zero and the inverse Laplace transform of the first term will be expressible as M sin($\omega t + \varphi$), a sinusoidal signal of magnitude (or amplitude) M and phase lag φ relative to the input sinusoid, where M and φ will be functions of ω. This is known as the steady state frequency response of $G(s)$, often simply shortened to frequency response. To determine its value it is not necessary to go through the partial fraction and Laplace transform process indicated above as it can be shown that it can be obtained from the complex number $G(j\omega)$, where M is the modulus and φ the argument of $G(j\omega)$. This is a very basic property of a linear system that for a sine wave input the output, in the steady state, will also be a sine wave of the same frequency with the magnitude and phase shift dependent on the frequency.

The value of a transfer function $G(s)$ for a specific value of $s = s_1$ is $G(s_1)$ and from consideration of the zero-pole representation of equation (2.13) it can be seen that it is given by

$$M(s_1) = \frac{K\prod_{j=1}^{m} PB_j}{\prod_{i=1}^{n} PA_i} \tag{3.11}$$

where P is the point s_1 in the s plane and $\mathrm{PB_j}$ and $\mathrm{PA_i}$ are the distances from P to the *m* zeros, β_j and *n* poles, α_i. Also the argument φ is given by

$$\varphi(s_1) = \sum_{j=1}^{m} \theta_j - \sum_{i=1}^{n} \psi_i \tag{3.12}$$

where θ_j and ψ_i are the zero and pole angles respectively, that is the angles measured from the direction of the positive real axis to the lines drawn from zero *j* to the point P and from pole *i* to the point, P, respectively. Evaluating the frequency response as ω goes from 0 to ∞ means evaluating the above as s_1 goes from 0 to ∞ on the imaginary axis of the *s*-plane. The value of understanding this is that it enables one to appreciate how *M* and φ of a frequency response will vary as ω is increased.

As a simple example consider again the transfer function of equation (2.15) that is

$$G(s) = \frac{4(s+1)}{(s+2)(s^2+s+1)} \tag{3.13}$$

Its zero-pole plot, shown in Figure 2.1, is repeated below as Figure 3.6 but with the edition of lines joining the one zero and three poles to the point P = 3*j* on the imaginary axis. The lengths of the lines and angles are marked from which it can be seen that the frequency response of *G* at $\omega = 3$, has

$$M = \frac{4 * PB_1}{PA_1 * PA_2 * PA_3} = 4\frac{\sqrt{10}}{\sqrt{13} * 2.192 * 3.898} = 0.411 \tag{3.14}$$

and

$$\varphi = \theta_1 - \psi_1 - \psi_2 - \psi_3 = \tan^{-1} 3 - \tan^{-1} 1.5 - \tan^{-1} 4.268 - \tan^{-1} 7.732$$

giving

$$\varphi = 71.57^o - 56.31^o - 76.81^o - 82.63^o = -141.2^o \tag{3.15}$$

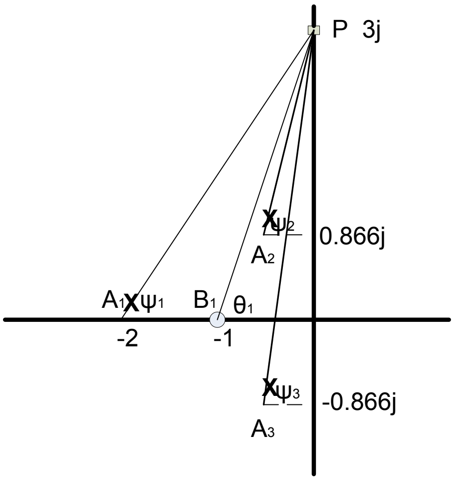

Figure 3.6 Graphical evaluation of a frequency response from the zero-pole plot.

Magnitude and phase of the output for a sinusoidal input have a very physical meaning but mathematically they are a polar representation of the output, which can therefore be written in the rectangular form for a complex number, that is

$$G(j\omega) = M(\omega)e^{j\phi(\omega)} = X(\omega) + jY(\omega) \tag{3.16}$$

The relationships between the polar and rectangular representations are

$$M(\omega) = [X^2\,(\omega) + Y^2(\omega)]^{\ 1/2} \tag{3.17}$$

$$\varphi = a\ \tan\ 2(Y\ (\omega), X(\omega)) \tag{3.18}$$

a tan 2 is the arctangent function used in MATLAB which correctly gives the phase φ between 0 and 360°. Most books write $\varphi = \tan^{-1}(Y(\omega)\ /\ X(\omega))$ which is simply incorrect without further qualification as the mathematical function $\tan^{-1}$ only exists between -90° and 90°.

4 Frequency Responses and Their Plotting

4.1 Introduction

The frequency response of a transfer function *G(jω)* was introduced in the last chapter. As *G(jω)* is a complex number with a magnitude and argument (phase) if one wishes to show its behaviour over a frequency range then one has 3 parameters to deal with the frequency, ω, the magnitude, M, and the phase φ. Engineers use three common ways to plot the information, which are known as Bode diagrams, Nyquist diagrams and Nichols diagrams in honour of the people who introduced them. All portray the same information and can be readily drawn in MATLAB for a system transfer function object $G(s)$.

One diagram may prove more convenient for a particular application, although engineers often have a preference. In the early days when computing facilities were not available Bode diagrams, for example, had some popularity because of the ease with which they could, in many instances, be rapidly approximated. All the plots will be discussed below, quoting many results without going into mathematical detail, in the hope that the reader will obtain enough knowledge to know whether MATLAB plots obtained are of the general shape expected.

4.2 Bode Diagram

A Bode diagram consists of two separate plots the magnitude, M, as a function of frequency and the phase φ as a function of frequency. For both plots the frequency is plotted on a logarithmic (log) scale along the x axis. A log scale has the property that the midpoint between two frequencies ω_1 and ω_2 is the frequency $\omega = \sqrt{\omega_1\omega_2}$. A decade of frequency is from a value to ten times that value and an octave from a value to twice that value. The magnitude is plotted either on a log scale or in decibels (dB), where $dB = 20\log_{10} M$. The phase is plotted on a linear scale. Bode showed that for a transfer function with no right hand side (rhs) s-plane zeros the phase is related to the slope of the magnitude characteristic by the relationship

$$\varphi(\omega_1) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{dA}{du} \log\coth\frac{|u|}{2}\,du \tag{4.1}$$

where $\varphi(\omega_1)$ is the phase at frequency ω_1, $u = \log_e(\omega/\omega_1)$ and $A(\omega) = \log_e |G(j\omega)|$.

It can be further shown from this expression that a relatively good approximation is that the phase at any frequency is 15° times the slope of the magnitude curve in dB/octave. This was a useful concept to avoid drawing both diagrams when no computer facilities were available.

For two transfer functions G_1 and G_2 in series the resultant transfer function, G, is their product, this means for their frequency response

$$G(j\omega) = G_1(j\omega)G_2(j\omega) \tag{4.2}$$

which in terms of their magnitudes and phases can be written

$$M = M_1 M_2 \text{ and } \varphi = \varphi_1 + \varphi_2 \tag{4.3}$$

Thus since a log scale is used on the magnitude of a Bode diagram this means Bode magnitude plots for two transfer functions in series can be added, as also their phases on the phase diagram. Hence a transfer function in zero-pole form can be plotted on the magnitude and phase Bode diagrams simple by adding the individual contributions from each zero and pole. It is thus only necessary to know the Bode plots of single roots and quadratic factors to put together Bode plots for a complicated transfer function if it is known in zero-pole form.

4.2.1 A single time constant

The single pole transfer function is normally considered in time constant form with unit steady state gain, that is

$$G(s) = \frac{1}{1+sT} \tag{4.4}$$

It is easy to show that this transfer function can be approximated by two straight lines, one constant at 0 dB, as $G(0) = 1$, until the frequency, $1/T$, known as the break point, and then from that point by a line with slope -6dB/octave. The actual curve and the approximation are shown in Figure 4.1 together with the phase curve. The differences between the exact magnitude curve and the approximation are symmetrical, that is a maximum at the breakpoint of 3dB, 1dB one octave each side of the breakpoint, 0.3 dB two octaves away etc. The phase changes between 0° and -90° again with symmetry about the breakpoint phase of -45°. Note a steady slope of -6 dB/octave has a corresponding phase of -90°.

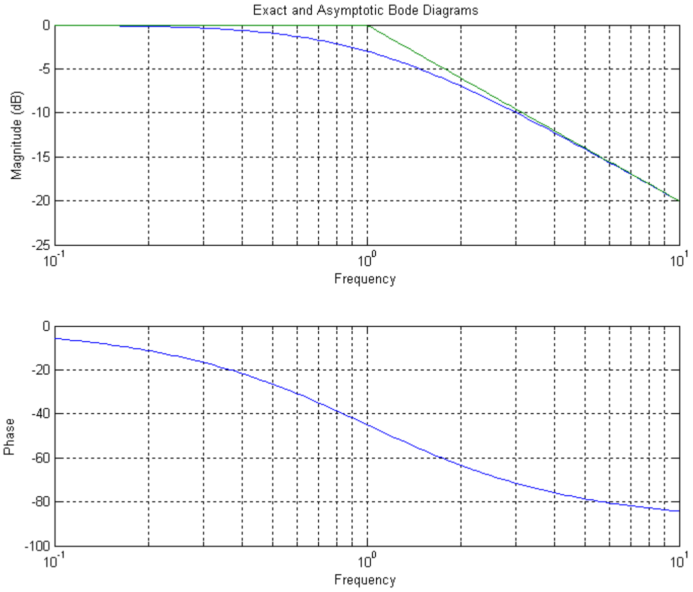

Figure 4.1 Bode exact and approximate magnitude curves, and phase curve, for a single time constant.

The Bode magnitude plot of a single zero time constant, that is

$$G(s) = 1 + sT \tag{4.5}$$

is simply a reflection in the 0 dB axis of the pole plot. That is the approximate magnitude curve is flat at 0 dB until the break point frequency, $1/T$, and then increases at 6 dB/octave. Theoretically as the frequency tends to infinity so does its gain so that it is not physically realisable. The phase curve goes from 0° to +90°.

4.2.2 An Integrator

The transfer function of an integrator, which is a pole at the origin in the zero-pole plot, is 1/s. It is sometimes taken with a gain *K*, i.e.*K*/s. Here *K* will be replaced by $1/T$ to give the transfer function

$$G(s) = \frac{1}{sT} \tag{4.6}$$

On a Bode diagram the magnitude is a constant slope of -6 dB/octave passing through 0 dB at the frequency $1/T$. Note that on a log scale for frequency, zero frequency where the integrator has infinite gain (the transfer function can only be produced electronically by an active device) is never reached. The phase is -90° at all frequencies. A differentiator has a transfer function of sT which gives a gain characteristic with a slope of 6 dB/octave passing through 0dB at a frequency of $1/T$. Theoretically it produces infinite gain at infinite frequency so again it is not physically realisable. It has a phase of +90° at all frequencies.

4.2.3 A Quadratic Form

The quadratic factor form is again taken for two complex poles with ζ < 1 as in equation (3.7), that is

$$G(s) = \frac{\omega_o^2}{s^2 + 2\zeta s\omega_o + \omega_o^2} \tag{4.7}$$

Again $G(0) = 1$ so the response starts at 0 dB and can be approximated by a straight line at 0 dB until ω_o and by a line from ω_o at -12 dB/octave. However, this is a very coarse approximation as the behaviour around ω_o is highly dependent on ζ. It can be shown that the magnitude reaches a maximum value of $M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}}$, which is approximately 1/2ζ for small ζ, at a frequency of $\omega = \omega_o\sqrt{1-2\zeta^2}$. This frequency is thus always less than ω_o and only exists for ζ < 0.707. The response with ζ = 0.707 always has magnitude, M < 1. The phase curve goes from 0° to -180° as expected from the original and final slopes of the magnitude curve, it has a phase shift of -90° at the frequency ω_o independent of ζ and changes more rapidly near ω_o for smaller ζ, as expected due to the more rapid change in the slope of the corresponding magnitude curve. Figure 4.2 shows Bode plots for various values of ζ against normalised frequency ω/ω_o. For the quadratic zero

$$G(s) = \frac{s^2 + 2\zeta s\omega_o + \omega_o^2}{\omega_o^2} \tag{4.8}$$

the Bode plots are just reflections in the 0 dB and zero phase axes of the graphs for the quadratic pole.

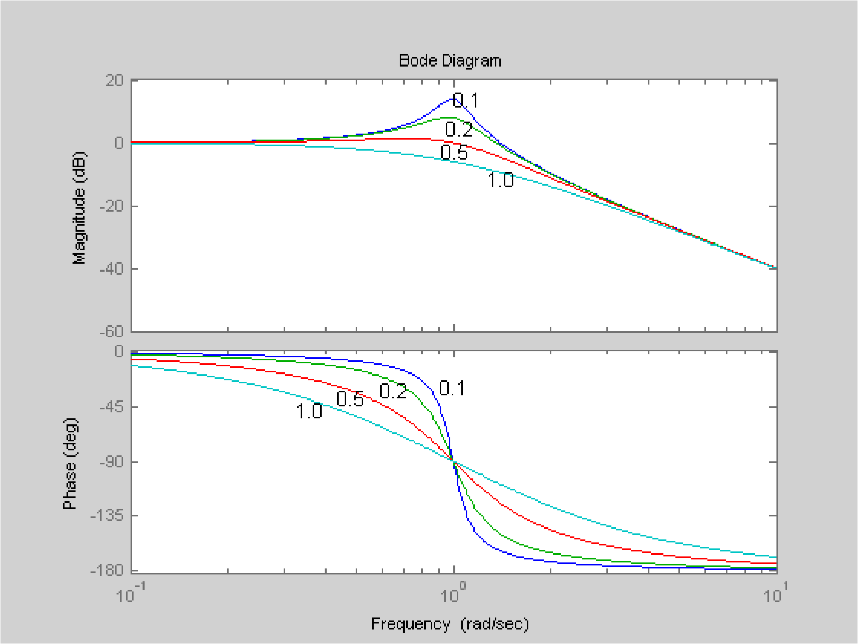

Figure 4.2 Normalised Bode plots for the quadratic pole form for different ς

4.2.4 An Example Bode Plot

Consider again the one zero, three pole transfer function

$$G(s) = \frac{4(s+1)}{(s+2)(s^2+s+1)} \tag{4.9}$$

Dividing numerator and denominator by 2, it can be written in the form

$$G(s) = \frac{2(1+s)}{(1+0.5s)(s^2+s+1)} \tag{4.10}$$

For plotting the Bode diagram it can be thought of as 4 transfer functions:-

1) a constant gain of 2
2) a single zero with a breakpoint of 1
3) a single pole with a breakpoint of 2
4) a quadratic pole with natural frequency 1 and damping ratio, ζ = 0.5.

The instruction in MATLAB to obtain the Bode plot of a transfer function object *G* is simply ***bode(G)***. The resultant Bode magnitude plot, marked (R), is shown in Figure 4.3 together with the individual plots of its four constituents, marked (1) to (4) as given above. The grid is added to the plot by typing ***grid***.

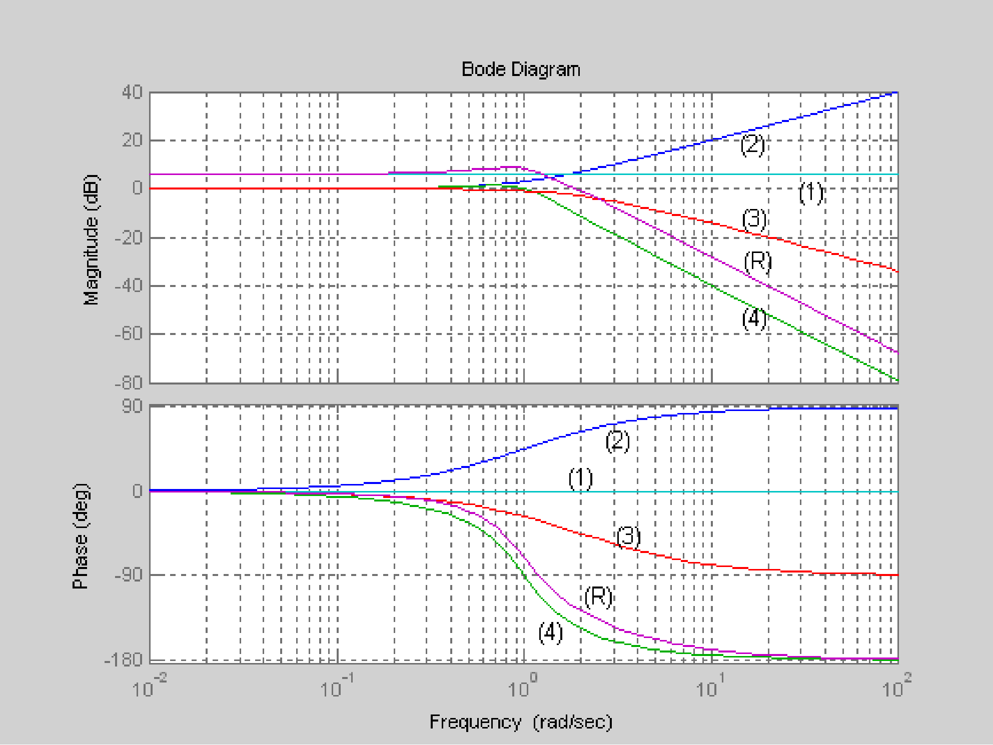

Figure 4.3 Bode plot of *G*(*s*) of equation (4.10) and its constituents.

4.3 Nyquist Plot

Since

$$G(j\omega) = M(\omega)e^{j\varphi(\omega)} = X(\omega) + jY(\omega) \tag{4.11}$$

every choice of ω gives a point in a complex plane either plotted in polar coordinates for the M, φ form or in rectangular coordinates in X, Y form. Joining the points together as ω is varied produces a locus with ω as a parameter which is known as a polar or Nyquist plot. To obtain analytical results one needs to be able to work in both polar and rectangular coordinates, since one may be more appropriate than the other for a particular evaluation. From consideration of the individual elements of a transfer function in the Bode approach of the previous section one should be able to estimate the shape of a Nyquist plot.

Important points in this respect are the high and low frequency limits, that is the value of $G(j\omega)$ as $\omega\to\infty$ and 0. For large s the limit of the general transfer function $G(s)$ of equation (2.9) will tend to $1/s^{(n-m)}$, that is 1 over s to the relative degree. Thus for a strictly proper transfer function the gain will tend to zero and the phase to $-90(n - m)°$ as $\omega\to\infty$. For a proper transfer function with $n = m$ the gain will tend to a finite value and the phase to zero. At low frequencies the transfer function, $G(0)$ will tend either to a constant or s to the power of the number of differentiation terms minus integration terms in the transfer function. Typically only integration terms exist in transfer functions for control systems so the behaviour at low frequencies depends on the number of integrators and G(s) tends to $1/s^i$ where i is the number of integrators. Thus at low frequencies for $i > 0$ the magnitude tends to infinity and the phase to $-90i°$. This phase result does not mean that the locus starts on an axis as sketches in many books incorrectly show. As a simple example of this point consider the transfer function

$$G(s) = \frac{1}{s(s+1)^2} \tag{4.12}$$

then putting $s = j\omega$, and writing $G(j\omega)$ in the form $X(\omega) + jY(\omega)$ gives $X(\omega) = \frac{-2}{(1+\omega^2)^2}$ and $Y(\omega) = \frac{-(1-\omega^2)}{(1+\omega^2)^2}$. Clearly as $\omega \to 0$, $X(\omega) \to -2$, not the imaginary axis, although the phase does tend to -90°. This will always be the case that the locus for a transfer function with one or more integrators will tend to an asymptote which in principle can be calculated. The Nyquist plot of this transfer function is obtained with the instruction ***nyquist(G)***. It is shown in Figure 4.4, which is obtained by the following single instruction defining the transfer function of G in the Nyquist statement:-

```
>> nyquist(tf(1,[1 2 1 0]))
```

Information about where a Nyquist plot cuts the axes can be obtained from the facts that the real axis is cut when $Y(\omega) = 0$ or arg $G(j\omega) = 0°$ or $180°$, and the imaginary axis when $X(\omega) = 0$ or arg $G(j\omega) = -90°$ or $+90°$. Which are the easiest calculations can depend on the transfer function. For the above example it is easily seen from $Y(\omega)$ that the real axis is cut when $\omega = 1$ and the imaginary axis is only reached as ω tends to infinity. However for $G(s) = 1/(1 + s)^6$ then where it cuts the axes is best obtained using arg $G(j\omega)$, which is simply equal to $- 6 \tan^{-1}\omega$.

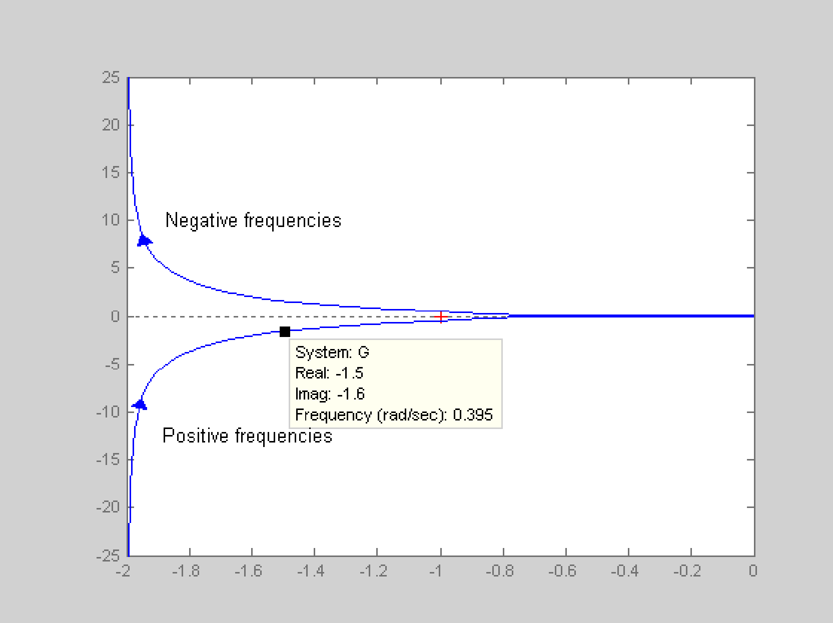

Figure 4.4 Nyquist plot of $1/s(s + 1)^2$.

Three further comments must be made here about the plot of Figure 4.4:-

1) For reasons to be explained later the graph is drawn for both positive and negative frequencies. The labelling of these has been added to the plot afterwards.
2) It can be shown for all transfer functions that $X(\omega)$ is an even function and $Y(\omega)$ an odd function of ω. Thus the negative frequency part of the plot is a reflection of the positive frequency plot in the real axis.
3) MATLAB does not label the frequencies automatically on the plot but they can be selected by use of the cursor as has been done to obtain one frequency point on this plot.
4) The frequency response plot instructions ***bode(G)*** and ***nyquist(G)*** in MATLAB automatically select the frequency range. This can be done by the user by selecting a vector ω, typically on a log scale using the instruction ***ω = logspace(a,b,n)***, which generates n points on a log scale between 10^a and 10^b. If n is omitted the default is 50 points. The plot instructions are then ***bode(G, ω)*** and ***nyquist(G, ω)***.

The last instruction in (4) has been used with the ω vector generated by logspace for a = -0.5 and b = 1 to show a more detailed plot near the origin in Figure 4.5. From the two plots it can be clearly seen that at low frequencies, where the gain tends to infinity because of the single integration, the locus starts from the asymptote at $X(\omega)$ = -2 with a phase of -90°, crosses the real axis, that is has a phase of -180°, at X = -0.5 and tends to the origin (zero gain) at high frequencies with a phase of -270° (relative degree of 3 times -90°). The real axis crossing occurs at a frequency of unity.

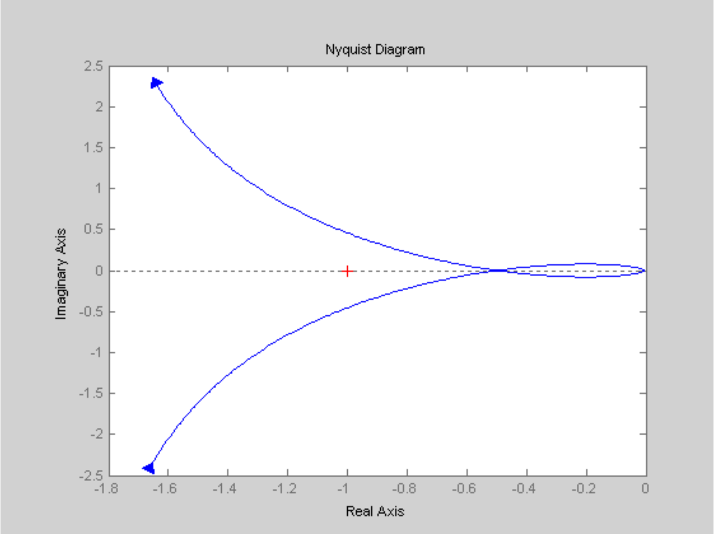

Figure 4.5 Nyquist plot with new ω vector.

A final comment on Nyquist plots is that sometimes Inverse Nyquist plots are drawn, these are simply the Nyquist plot of the inverse of the transfer function, i.e a Nyquist plot of $G(j\omega)^{-1}$.

4.4 Nichols Plot

The Nichols plot is similar to the Nyquist plot in that it is a locus as a function of ω, the difference being the chosen axes. On a Nichols plot these are the magnitude in dB on the ordinate and the phase in degrees on the abscissa. The origin is chosen, for reasons which will be explained, later as 0 dB and -180°. The Nichols plot for the same transfer function as the Nyquist plot of Figure 4.4 is obtained by the instruction ***nichols(G)*** and is shown in Figure 4.6. The grid is obtained by typing ***ngrid***. As expected the plot shows the magnitude decreasing monotonically with increase in frequency, the arrow for which was added to the plot, and the phase changing from -90° to -270°.

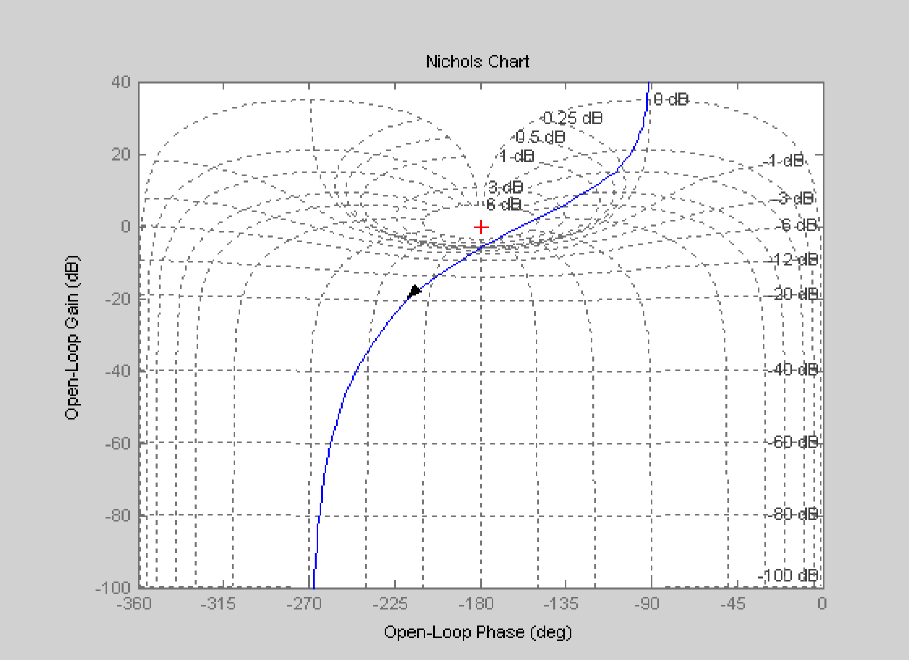

Figure 4.6 Nichols plot of $1/s(s + 1)^2$.

5 The Basic Feedback Loop

5.1 Introduction

The basic concept, of feedback control, as mentioned in the first chapter is to measure the quantity to be controlled, usually called the controlled variable and denoted by *C*, and to compare it with the desired or reference value, usually denoted by *R*, and to use any error to adjust *C* to the desired value. Thus a basic feedback loop has the structure shown in the diagram of Figure 5.1 where the various physical elements are represented by their mathematical models in transfer function form. The process being controlled, denoted $G(s)$, is usually referred to as the process or plant transfer function.

Measurement of its output, *C*, is obtained by a sensor of transfer function *H*, which is also known as the feedback transfer function. In many cases the dynamics of *H* may be neglected, so that *H* is just a constant with units converting the output to appropriate units for use in the control system. For example, if the output controlled variable is a temperature and the control system error channel uses voltage then *H* will have units of V/°C. The importance of *H* cannot be underestimated since if the sensor is supposed to give 5V at 50°C but actually gives 5V at 48°C the perfect control system with its reference set to 5V for 50°C will control the temperature at 48°C. In other cases *H* will contain dynamics of the sensor and/or loop compensation dynamics. Some sensors introduce a significant noise level into the loop and this can be represented by the signal, *N*, shown. The error signal is normally processed through a controller of transfer function $G_c(s)$, as shown, before providing the plant input signal *U*. The transfer function of the forward path of the loop is $G_c(s)G(s)$.

The loop is often subject to disturbance inputs, for instance in a position control system for a large antenna dish, varying wind speeds impacting the dish will produce a torque disturbance. A disturbance signal *D* is therefore shown in Figure 5.1. Since the loop is linear the effect of all of the three input signals, *R*, *D* and *N* at a particular point can be found independently and then summed.

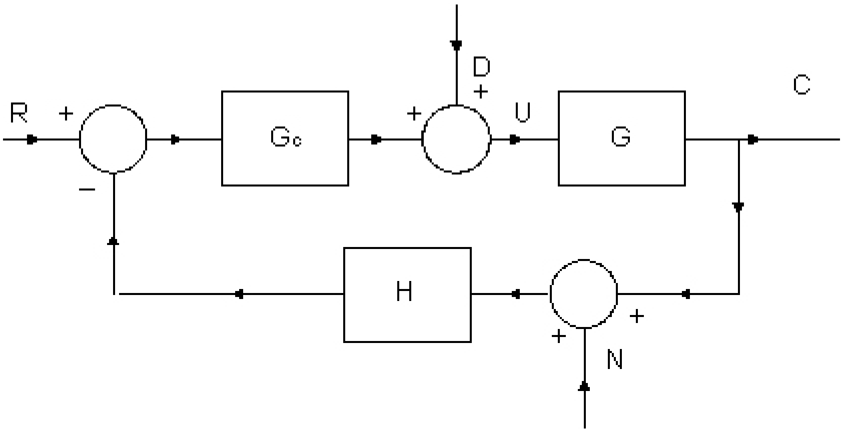

Figure 5.1 Basic Block Diagram of Feedback Control System.

5.2 The Closed Loop

It can be shown for the closed loop of Figure 5.1 that

$$C(s) = \frac{G_c(s)G(s)R(s)}{1+G_c(s)G(s)H(s)} + \frac{G(s)D(s)}{1+G_c(s)G(s)H(s)} + \frac{G_c(s)G(s)H(s)N(s)}{1+G_c(s)G(s)H(s)} \tag{5.1}$$

The numerator terms are the loop transfer functions from the specific input to the output C(s) and the denominator term is 1 plus the product of the transfer functions in the loop, which is known as the open loop transfer function, $G_{ol}(s)$ That is

$$G_{\phi}\ (s) = G_c(s)G(s)H(s) \tag{5.2}$$

The negative feedback is always assumed so in actual fact if the loop were opened and a signal, V(s), injected, it would return as $-G_{ol}(s)V(s)$. From here, unless otherwise stated, our concern will be with the response to the input *R*, so that *D* and *N* will be assumed to be zero.

The transfer function from *R* to *C*, often denoted by *T*(*s*), is given by

$$T(s) = \frac{C(s)}{R(s)} = \frac{G_c(s)G(s)}{1+G_c(s)G(s)H(s)} \tag{5.3}$$

Its poles are the roots of

$$F(s) = 1 + G_c(s)G(s)H(s) = 1 + G_b\ (s) = 0 \tag{5.4}$$

which is known as the characteristic equation of the closed loop system, and the closed loop will be stable if all its roots are in the left hand side (lhs) of the *s*-plane. Denoting each of the individual element transfer functions in terms of their numerator and denominator polynomials, that is

$$G_c(s) = \frac{N_c(s)}{D_c(s)},\ G(s) = \frac{N(s)}{D(s)} \text{ and } H(s) = \frac{N_h(s)}{D_h(s)} \tag{5.5}$$

then the closed loop transfer function

$$T(s) = \frac{N_c(s)N(s)D_h(s)}{N_c(s)N(s)N_h(s) + D_c(s)D(s)D_h(s)}. \tag{5.6}$$

The important point to note is that the zeros of $T(s)$ are the zeros of $G_c(s)$ and $G(s)$, but the poles of $H(s)$

5.3 System Specifications

The designer of a closed loop control system will be given specifications which the resulting system has to meet. Design is invariably an iterative process and begins with the selection and modelling of the various system components before the performance of the closed loop system can be evaluated. It may be after some analysis, say for a position control system, it is found that the required speed of response can only be achieved with a larger motor, so the designer returns to the component selection and modelling process. Here it is assumed that the plant transfer function is fixed and G_c and possibly H have to be chosen to try and meet the specifications. The actual design specifications, which, for example, may involve a limit on the use of energy, may have to be 'translated' into appropriate quantifiable properties of the closed loop, which is all that can be satisfied with analytical control techniques. To make the design easier it is often assumed that there are a limited number of transfer functions that might be used in G_c and H and the design objective then becomes one of selecting suitable parameters for these fixed form controllers. The feedback loop of Figure 5.1 can be redrawn, as in Figure 5.2, with H in the forward path of the loop and the reference, R, going through $1/H$.

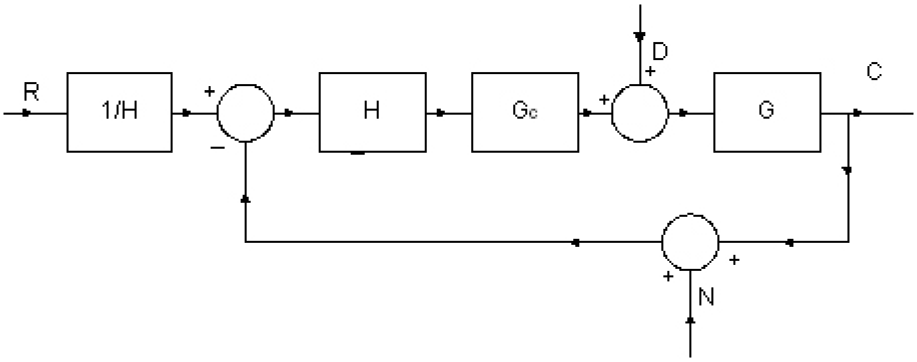

Figure 5.2 Equivalent block diagram to Figure 5.1.

The input to H in the forward path of the loop is then the error in output units, say °C for a temperature control, and its output the error in the sensor output units, say voltage V. For loop analysis the gain of H can then be included in G_c, so for this reason many results are derived with H=1.

Some typical closed loop specifications are therefore discussed below:-

1) **Stability.** Obviously the prime requirement for a feedback loop is that it be stable. Methods for investigating stability are discussed in the next section.
2) **Steady state error**. In many instances the inputs, particularly R and D, may be assumed constant and it is often required that in the steady state they do not produce an error. Mathematically the steady state error can be found by applying the final value theorem to the s domain expression $E(s)$ for the error, which for the feedback loop of Figure 5.1 with H = 1, for the input $R(s)$ a step of amplitude R is

$$E(s) = \lim_{s \to 0} s \frac{R}{s} \frac{1}{1 + G_c(s)G(s)} \tag{5.7}$$

which will only be zero if $\lim_{s \to 0} G_c(s)G(s) \to \infty$, and this will only be true if $G_c(s)G(s)$ contains an integrator. Obviously if this is not the case then equation (5.7) allows the error magnitude to be calculated. However, if one just needs to determine if the steady state error is zero then this can be done simply from a consideration of the d.c. gain of the loop elements. This is infinite for an integrator, as in the steady state it can have a finite output with zero input. For example, for the above case a finite output C has to be obtained in the steady state for zero E, which is only possible with an integrator in the forward loop. If E were required to be zero for a ramp input the forward loop would require two integrators, the first producing a constant output for no input in the steady state and the second integrating the constant to produce a ramp. In the case of a disturbance D the error it produces is the value it has at C, (i.e C should not be affected by it). For D not to affect C then the forward path signal at the point where D enters the loop, U, should be equal and opposite to it. Thus if D is a constant $G_c(s)$ must contain an integrator for the output not to be affected in the steady state.

3) **Step response**. Characteristics of the closed loop response to a step input are often specified. These are based on the typical response with zero steady state error to a unit step shown in Figure 5.3 and are:-

   a) **Rise time,t_r**. This is the time taken to reach the steady state value of unity for the first time. If the response has no overshoot the time is often given for the response to go from 0.1 to 0.9, that is from 10 to 90%
   b) **Peak time,t_p** This is the time taken to reach the first overshoot of the response.

c) **Overshoot,%O.** This is the magnitude of the first overshoot in the response, normally expressed as a percentage. If the peak value is 1.2 then the overshoot is 20%.

d) **Settling time,t_s.** This is a measure of the time for the response to have approximately reached the steady state of unity. It is normally defined as the time to reach within a 2% band of the steady state (between 0.98 and 1.02) and remain there. Sometimes a 5% band is used as in the figure illustration.

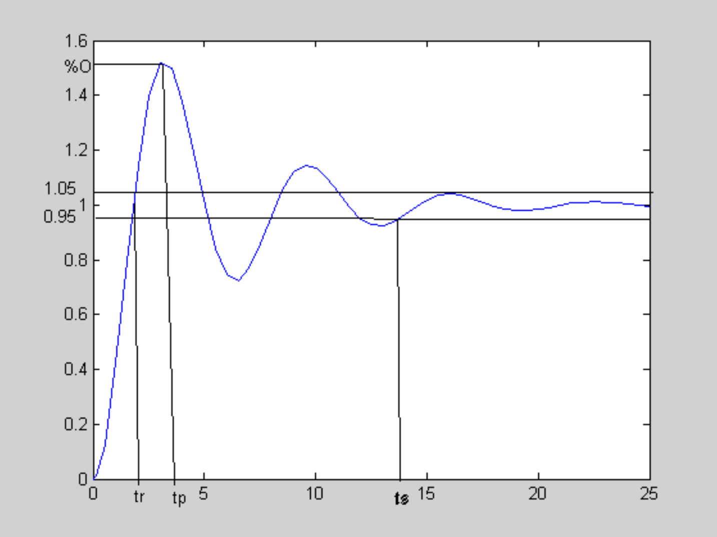

Figure 5.3 Typical step response for specifications.

For the response illustrated in the figure t_r is approximately 2 seconds, t_p approximately 4 seconds with a %O of 52% and t_s of 13.5 seconds for a 5% band.

4) **Frequency response.** Sometimes specifications are given with respect to the closed loop frequency response requirements of the system. The ideal requirement is for *C* to follow *R* exactly but this cannot be achieved as the input frequency is increased but is normally the case at low frequencies. Thus the closed loop frequency response $T(j\omega)$ typically starts at unit gain (0dB) and zero phase shift. The magnitude response may have one or more peaks, the usual case as shown in Figure 5.4, and then decrease.

   a) **Bandwidth, Bw.** This is defined from zero frequency to the first time (often the only time) the magnitude goes through -3dB (value of 0.707)

   b) **Frequency peak, M_p.** This is the maximum of the frequency response, provided it exceeds 0dB (unit gain).

For the response shown in the figure M_p is approximately 16dB at 0.9 rads/s and Bw is around 1.3 rads/s. The frequency response specifications are related to the step response ones dependent on the specific transfer function. For a simple transfer function like the second order one of equation (3.6) these relationships can easily be found as in section 3.2.2 it was shown how the overshoot was related to ς and in section 4.2.3 how M_p was related to ς. Thus if one is given time domain and frequency domain specifications one must look at their consistency. A rise time of 0.01 seconds, for example, would require a bandwidth significantly greater than 10 rads/s.

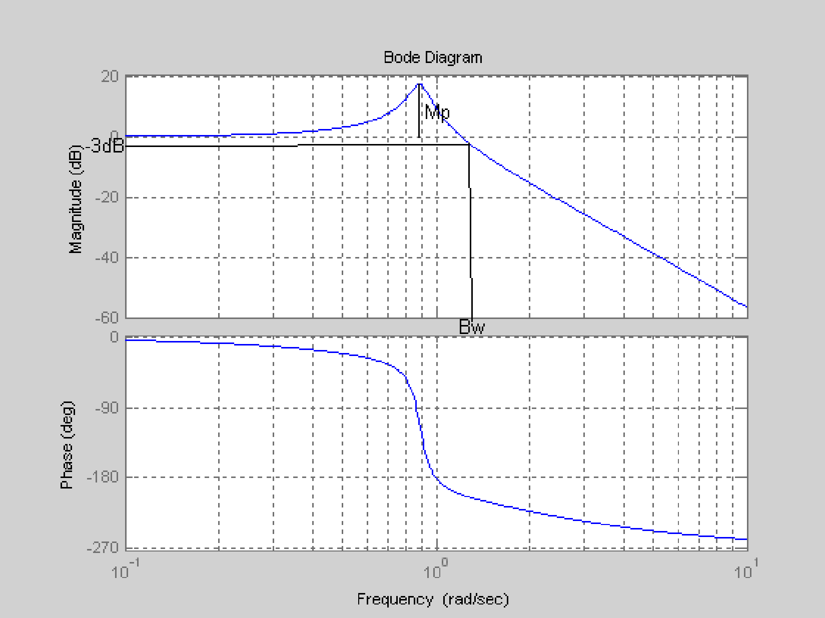

Figure 5.4 Typical frequency response for specifications.

5.4 Stability

The requirement for stability of the closed loop is that all the poles of the closed loop transfer function $T(s)$ of equation (5.3) lie in the lhs s-plane. The poles are the zeros of the characteristic equation (5.4), which will be a polynomial in s. If this polynomial is denoted by

$$F(s) = f_n s^n + f_{(n-1)} s^{(n-1)} + \ldots . f_1 s + f_0 \text{ with } f_0 > 0 \tag{5.8}$$

then its roots can easily be found using Matlab by the command **roots (poly)**, where ***poly*** is entered like ***num*** or ***den*** as a string of coefficients with the highest power of s first. For example

```
>> roots([1 6 11 6])
ans =

-3.0000
-2.0000
-1.0000
```

5.4.1 Routh Hurwitz Criterion.

Finding the roots of a polynomial of large order was very difficult before the advent of modern computational techniques and in 1876 a major contribution was made by Routh who obtained conditions which had to be satisfied for all roots of a polynomial to lie in the lhs s-plane. A polynomial which satisfies this condition is known as a stable polynomial. The criterion was later modified by Hurwitz to give the Routh-Hurwitz results presented in Appendix B.

Two simple results which prove useful are

a) A necessary but not sufficient condition, apart from the second order polynomial where it is both necessary and sufficient, is that all the coefficients of s must be positive that is $f_j > 0$ for all j
b) For the third order polynomial a necessary and sufficient condition is all the coefficients must be positive and $f_1 f_2 > f_0 f_3$

5.4.2 Mikhailov Criterion

The Mikhailov criterion is a simple graphical approach only normally mentioned in Russian textbooks. If the polynomial $F(j\omega)$ is plotted for ω increasing from zero on a complex plane, then all its roots will lie in the lhs s-plane if from starting on the positive real axis at f_0 it moves in a counter clockwise direction passing successively through the positive imaginary axis, negative real axis etc in turn until it cuts no further axes but 'heads' for infinity as illustrated in Figure 5.5. The number of axes cut will be n-1

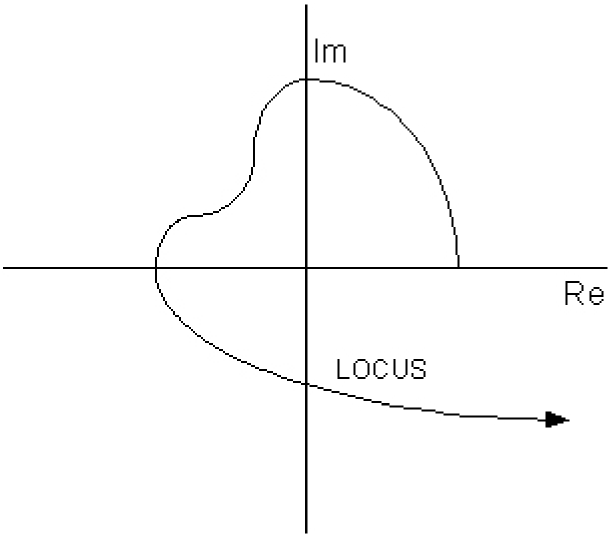

Figure 5.5 Illustration of Mikhailov Criterion for Stable Fourth Order $F(j\omega)$.

5.4.3 Nyquist Criterion

In the early days when control engineering was developing as a discipline it was very desirable to try and develop concepts to predict aspects of the closed loop system behaviour based on properties of the open loop transfer function. There were three major reasons for this:-

a) When a compensator (controller) is within the loop it is much easier to see how changes in its parameters will affect the open loop properties, for example the frequency response, than the closed loop properties.
b) Plant models were often obtained by frequency response testing so that $G(j\omega)$ was then available as a plot from experimental data.
c) Even when all the loop transfer functions were known calculating a closed loop step response was a laborious procedure.

For these reasons the Nyquist stability criterion, which is based on the open loop frequency response, was thus not only useful but also very practical. The derivation of the criterion, which uses the mathematics of functions of a complex variable, is relatively easy to explain in principle. It is based on Cauchy's mapping theorem which states that if a complex function, $F(s)$, is mapped around a closed contour in a clockwise direction in the s-plane (that is its value calculated at points on the contour and plotted in its own complex plane) the origin will be encircled N_{o} times in the clockwise direction where N_{o} is the difference between the number of zeros and poles of $F(s)$ enclosed by the chosen s-plane contour. When the contour is taken as the imaginary axis, this means taking ω from –∞ to ∞, and then the infinite semicircle in the right hand side (rhs) s-plane (around this ω remains infinite), known as the Nyquist D contour, shown in Figure 5.6, then the origin will be encircled by $F(j\omega)$ N_{o} times in a clockwise direction, where N_{o} is given by:-

$$\mathrm{N}_{\mathrm{o}} = [\text{zeros of } F(s) - \text{poles of } F(s)] \text{ in rhs } s\text{-plane} \tag{5.9}$$

The zeros of F(s) are required to assess stability so the equation may be written

$$\text{zeros of } F(s) \text{ in rhs} = \mathrm{N}_{\mathrm{o}} + \text{poles of } F(s) \text{ in rhs.} \tag{5.10}$$

From equation (5.4) it can be seen that the poles of $F(s)$ are the same as the poles of $G_{\mathrm{ol}}(s)$ and that the only difference between a mapping of $F(j\omega)$ and $G_{\mathrm{ol}}(j\omega)$ is that the latter is shifted from the former by -1 along the real axis. Thus equation (5.10) can be written

$$\text{zeros of } F(s) \text{ in rhs} = \mathrm{N} + \text{poles of } G_{\mathrm{ol}}(s) \text{ in rhs.} \tag{5.11}$$

where N now denotes the clockwise encirclements of the (-1,0) point by the plot of $G_{ol}(j\omega)$. Thus if the number of poles of G_{ol} in the rhs s-plane is known, which will of course be zero for a stable G_{ol}, the number of zeros of $F(s)$ in the rhs s-plane can be found from equation (5.11) to determine the stability of the feedback loop.

This equation gives the Nyquist stability criterion which may be formally stated as the closed loop system will be stable if the number of clockwise encirclements by the frequency response locus $G_{ol}(j\omega)$ of the (-1,0) point plus the number of rhs s-plane poles of $G_{ol}(s)$ is zero. Showing the Nyquist plot for both negative and positive frequencies allows the encirclements to be found.

So that the D contour does not pass through them when singularities (poles or zeros) exist on the imaginary axis, it has to be modified so that they lie outside by indentations of infinitesimal radius, ε, as shown dotted in figure 5.6. In many instances the plant to be controlled will be stable so that $G_{ol}(s)$ will have no rhs s-plane poles so for stability the Nyquist plot of $G_{ol}(j\omega)$ must have N = 0. Control engineers are, however, required to control plants which are unstable, a modern fighter aircraft being a good example.

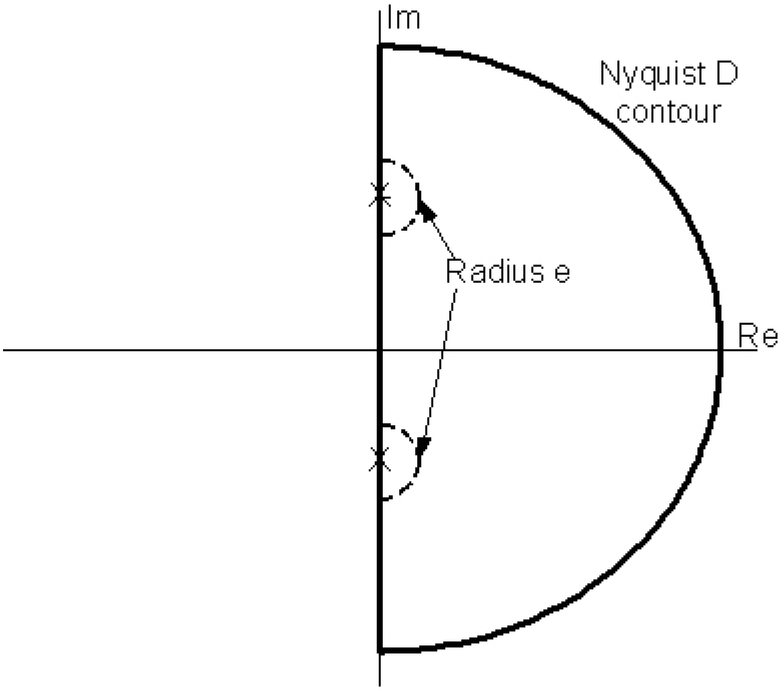

Figure 5.6 Nyquist D contour.

6 More on Analysis of the Closed Loop System

6.1 Introduction

In the previous chapter the basic feedback loop was discussed and typical specifications that might be required for its performance introduced. Before going on to discuss analytical methods that can be used for designing the controller to try and meet given specifications it is necessary to present some further analytical concepts used for feedback loop analysis. Also up to this point it has been assumed that the transfer function representation for block descriptions is a ratio of polynomials in s. There is, however, one linear element which often exists in a control system for which this is not the case, namely a time delay. This is therefore covered first in the next section.

6.2 Time Delay

A time delay as its name suggests is an element which produces an output which is a time delayed version of its input. It is also known as dead time or transport delay. The latter name reflects the fact that a common occurrence is due to say, a temperature measurement being made on a moving fluid down stream from where it has been heated. It is normally assumed that its initial output is zero. Thus for example, if the time delay is τ seconds the input, $v(t)$, and output will be as shown in Figure 6.1 for the linear input $v(t) = t$.

Mathematically the input, $v(t)$ can be defined as either $v(t) = 0$ for $t < 0$ and t for $t > 0$, or $tu_0(t)$ where $u_0(t)$ is the unit step at $t = 0$. Using the unit step notation the output can be written as $(t - \tau)u_0(t - \tau)$, that is a unit ramp beginning at time τ. The Laplace transform of the input is $1/s^2$ and of the output $e^{-s\tau}/s^2$ (see theorem (vi) Appendix A). Thus the transfer function for the time delay block, the ratio of the output to the input in the s-domain, is $e^{-s\tau}$.

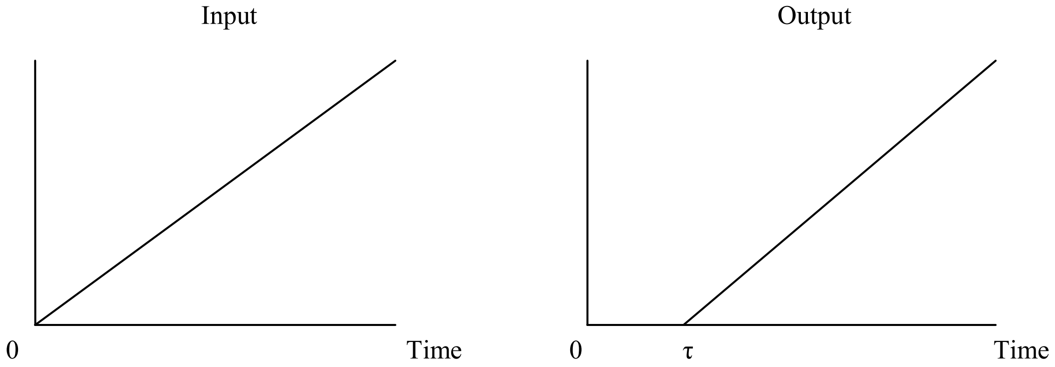

Figure 6.1 Illustration of a time delay

The time delay transfer function is easily handled when using frequency domain methods as with $s = j\omega$, it is $e^{-j\omega\tau}$, which has a unit magnitude at all frequencies and a phase lag of $\omega\tau$. Thus, for example, its Nyquist plot is a unit circle which has frequency points of value $\pi/2\tau$, π/τ, $3\pi/2\tau$, $2\pi/\tau$, $5\pi/2\tau$ etc, at -90°, -180°, -270°, -360°, -450°, etc. Figure 6.2 shows Nyquist plots of the transfer functions $G(s) = 1/(s^2 + s + 1)$ and $G(s)e^{-2}$, that is the former with an additional time delay of 1 second.

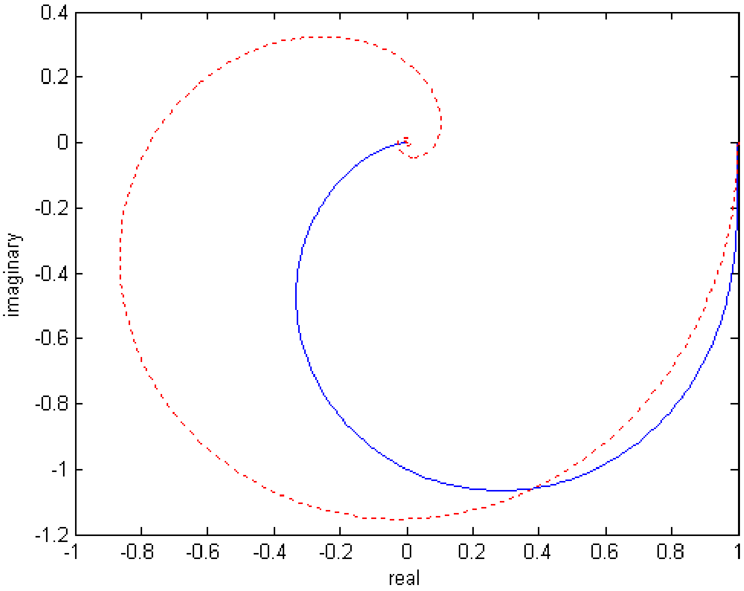

Figure 6.2 Nyquist plot of $G(s) = 1/(s^2 + s + 1)$ with (dotted) and without time delay of 1 second.

Although a time delay can be approximated by the standard series for an exponential a better approximation is to use a ratio of polynomials, a result due to Pade. This allows choice of the order of the numerator and denominator polynomials. The Pade Table of approximations is given in Appendix C.

6.3 The Root Locus

Design of a simple control loop may sometimes just involve the choice of a suitable gain, K, in which case the characteristic equation will be

$$1+ KG_{ol}(s) = 0 \tag{6.1}$$

and the poles of the closed loop transfer function, the roots of equation (6.1), will vary with K. Evans in 1948 found a diagrammatic method for showing how these roots would vary as K changed, known as a root locus, by recognising that, since s is complex, equation (6.1) could be written as the two equations

$$\text{Arg}(G_{ol}(s)) = -180°\sim \tag{6.2}$$

and

$$K|G_{ol}(s)|= 1 \tag{6.3}$$

Based on equation (6.2) he was able to prove several results indicating where the roots would be and then used equation (6.3) to mark the corresponding value of gain on the locus. MATLAB plots a root locus with the command ***rlocus(G)***.

Some simple rules which enable a quick check of a root locus, assuming G_{ol} is in the form of $G(s)$ given in equations (2.9) to (2.11), and K is positive are:-

1. The number of root locus paths will be n, assuming $n \geq m$.
2. The loci start at the poles of the open loop transfer function, G_{ol}, with $K = 0$
3. The loci finish at the zeros of the open loop transfer function, G_{ol}, as $K \to \infty$
4. A number of loci equal to the relative degree, (n-m), or the so-called number of zeros at infinity, of the open loop transfer function will tend to infinity as K tends to infinity
5. Loci exist on the real axis to the left of an odd number of singularities (poles plus zeros).

As a simple example the command

```
>> rlocus(tf([1],[1 3 2 0]))
```

produces the root locus shown in Figure 6.3. This transfer function for, G_{ol}, has no zeros and 3 poles at 0,-1 and -2, and it can easily be seen that the plot satisfies the above five rules for the loci. The characteristic equation has three roots all of which, initially for low gain, are real. As gain increases the root moving from the origin moves towards that from the one at -1, they meet at the so called breakaway point where they are equal and then form a complex pair for further increase in gain. The third root moves all the time to the left on the negative real axis from -2. Information for a specific point on the locus can be obtained by pointing at it and using a left side cursor click, with the result shown by the label on the Figure. The value of the Gain, K, and the coordinates of the point as Pole, are given. The other information given on the damping is based on the assumption, as explained in section 3.2.2, that this complex pole was one of the two complex poles of a second order system with the transfer function of equation (3.6). This is obviously only indicative since, for example, the closed loop response to a step input will, as indicated earlier depend upon any zeros; and in this case when there are no zeros on the relative weighting of the responses from the real pole beyond -2, actually at -2.50 for K = 1.89, and the complex pair, as illustrated in section 3.2.4. The overshoot of the closed loop step response for this third order transfer function with the gain value of 1.89 is 37.0 not 39.4%

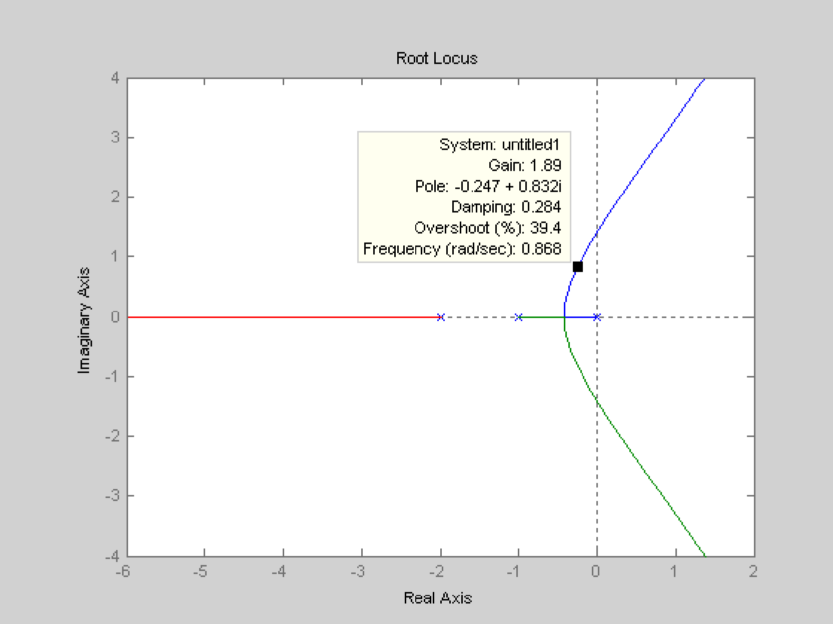

Figure 6.3 Illustrative root locus plot.

The gain range for the root locus plot is selected automatically but as for ω in a frequency response may be selected if preferred by the command ***K=linspace(a,b,n)***, which generates n points linearly spaced from a to b, inclusive, and then use of the plotting command ***rlocus(G,K)***. When a root passes into the rhs s-plane the closed loop becomes unstable. For the above case intersection with the imaginary axis occurs for K = 6 at a frequency of $2^{1/2}$ = 1.414, the value of the imaginary axis coordinate. This condition is often referred to as being neutrally stable, since mathematically a constant amplitude oscillation exists.

Although the root locus method is normally used for a varying gain in the characteristic equation it can be used for any variable parameter. Consider for example

$$G_{ol}(s) = \frac{4}{s(s+1)(s+a)} \tag{6.4}$$

The closed loop characteristic equation is

$$s^3 + (1+a)s^2 + as + 4 = 0 \tag{6.5}$$

This can be written, by dividing by the terms independent of *a*, as

$$1 + \frac{as(s+1)}{s^3 + s^2 + 4} = 0 \tag{6.6}$$

Thus comparing with equation (6.1) *a* replaces *K* and the equivalent G_{ol} is $\frac{s(s+1)}{s^3 + s^2 + 4}$. Plotting the root locus of this transfer function will then show how the closed loop poles vary as a function of *a*. Note that to find the poles a cubic has to be solved, which is as mentioned previously easily done in MATLAB with ***roots***.

6.4 Relative Stability

Relative stability, which may also be called robustness, is a measure of how near a system is to being unstable. Robustness, however, is used with respect to many properties and its use is best qualified by using it in the form robustness of property X with respect to property Y if the context is not clear. There are several measures which are used to indicate relative stability and some are discussed below.

6.4.1 Pole positions.

Obviously since instability results from a pole entering the rhs *s*-plane, the nearer a pole in the lhs s-plane is to the imaginary axis the nearer the system being studied will be to instability. Thus in the previous root locus example the nearer the gain to the value of 6 the nearer the two complex poles are to the imaginary axis. The information in the panel of the root locus plot therefore gives an indication of the relative stability of the system.

6.4.2 Gain and Phase Margin.

If the open loop system transfer function is stable then from the Nyquist criterion given in section 5.4.3 the closed loop system will be stable if its Nyquist plot does not encircle the Nyquist point (-1, 0). Passing of the locus through this point corresponds to neutral stability like the crossing of the imaginary axis in a root locus plot. For the $G(s)$ considered in the root locus plot this would occur at a frequency of 1.414 rad/s. with an additional gain of $K = 6$. In gain-phase terms the Nyquist point has a gain of unity and a phase of -180°, hence the choice of this point as the origin for a Nichols plot. Obviously therefore measures of how near the open loop frequency response locus is to this critical point, on the Nyquist or Nichols plots, are indicators of relative stability. Figure 6.4 shows a typical open loop frequency response Nyquist plot, it is in fact

$$G_{ol}(s) = \frac{3}{s(s+1)((s+2)} \tag{6.7}$$

the plant transfer function as used in the root locus plot with an additional gain of 3. The (-1, 0) point is labelled N, the origin O, and the point where the locus cuts the negative real axis as P.

1) **Gain Margin**. The gain margin is the amount by which the gain needs to be increased for the closed loop to become unstable. It is usually given in dB's and is $20\log_{10}$(ON/OP). For the example plot the negative real axis is cut at -0.5, so for the locus to pass through N the gain has to be increased by a factor of 2, which is 6dB. As the phase shift is -180° the frequency at this point is usually known as the phase crossover frequency, which will be denoted by ω_{pc} and is 1.414 rad/s. in the example.

2) **Phase Margin** The phase margin is the amount by which the loop phase needs to be changed for the loop to become unstable. The point G on the frequency response has a gain of unity, that is OG = 1, so for this point to pass through N the phase needs to be changed by the amount of the angle GON marked in the figure. Mathematically the phase margin is 180° + arg $(G(j\omega_{gc})$, where ω_{gc} is the frequency at G and is known as the gain crossover frequency since $|(G(j\omega_{gc})|=1$. In the example ω_{gc} = 0.969 rads/s. and the phase margin is 20.0°

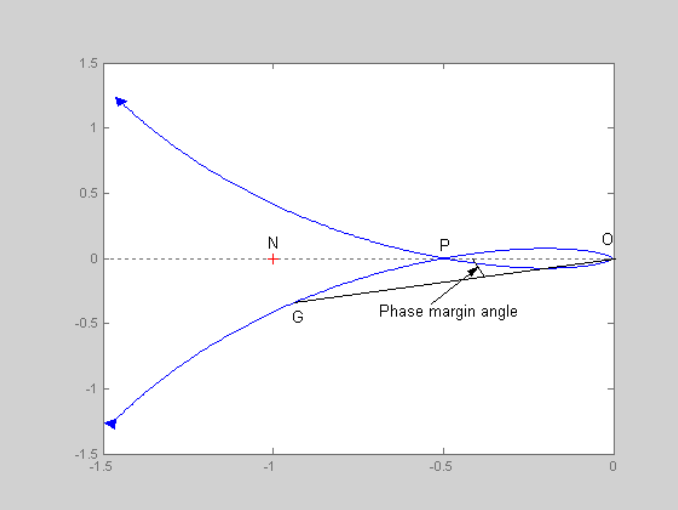

Figure 6.4 Nyquist plot illustrating gain and phase margins.

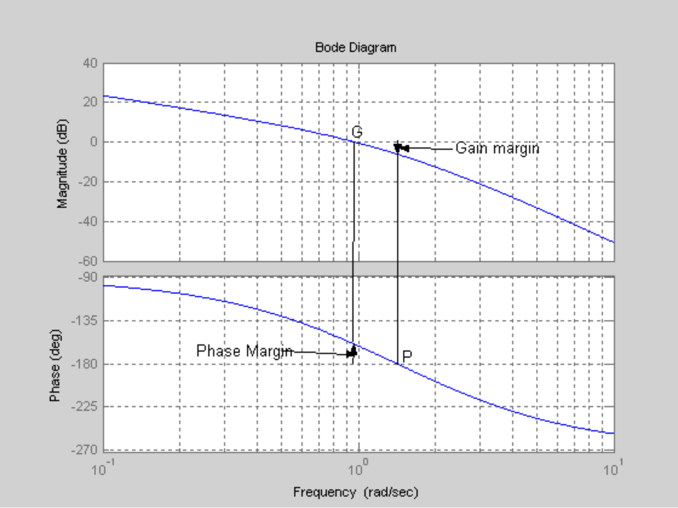

Figure 6.5 Bode plot illustrating gain and phase margins.

Figure 6.5 gives the Bode plots of the same open loop transfer function of equation (6.7) and shows how the closed loop gain and phase margins are found from it. Figure 6.4 also illustrates another point that for a stable open loop transfer function the closed loop will be stable if the open loop frequency response traced with increasing frequency passes the critical point to its left. Sometimes Nyquist loci are much more complicated than the simple smooth one shown, for example with multiple crossings of the negative real axis, and in such cases further clarification may be necessary when using gain and phase margin terms. Note also that with a non smooth locus it would be possible to have a large gain margin and a small phase margin or visa versa.

6.4.3 Sensitivity functions

The closed loop transfer function, with $H = 1$, is

$$T(s) = \frac{G_{ol}(s)}{1 + G_{ol}(s)} \tag{6.8}$$

If one regards $G_{\mathrm{ol}}(s)$ as a variable and wishes to describe the sensitivity, S, of T to changes in G_{ol} then this may be written as

$$S = \frac{\Delta T / T}{\Delta G_{ol} / G_{ol}} = \frac{G_{ol}}{T}\frac{dT}{dG_{ol}} \tag{6.9}$$

which on evaluating the differentiation gives

$$S = \frac{1}{1 + G_{ol}(s)} \tag{6.10}$$

The complimentary sensitivity function is defined as

$$1 - S = T \tag{6.11}$$

which is the closed loop transfer function. Since the vector 1 + $G_{\mathrm{ol}}(j\omega)$ is a measure of the distance of G_{ol} at any frequency from the Nyquist point {i.e. in figure 6.4 the length NG is 1+$G_{\mathrm{ol}}(j\omega_{\mathrm{gc}})$}, when this is small $S(j\omega)$ and $T(j\omega)$ will have large peak magnitudes, so their maximum magnitudes may be used as relative stability indicators. That of $T(j\omega)$ is discussed further in the next section

6.5 M and N Circles

The idea of M and N circles again relates to the days when computer software was not available and designers were interested in finding out about the closed loop frequency response behaviour from the open loop frequency response. The computations from open loop to closed loop properties are now easily done but the concepts are still of some value, particularly that of M circles. If the feedback transfer function H = 1 then, $G_{\mathrm{ol}}(j\omega) = G_{\mathrm{c}}(j\omega)G(j\omega)$, and the closed loop frequency response function is

$$T(j\omega) = \frac{G_{ol}(j\omega)}{1 + G_{ol}(j\omega)} \tag{6.12}$$

If the magnitude of this function is required to remain constant, at a value M, as ω varies then it can be shown that $G_{\mathrm{ol}}(j\omega)$ should move on a circular path on a Nyquist plot. Thus by superimposing this grid, known as M circles, on a Nyquist plot of the open loop frequency response one can see how the magnitude of the closed loop frequency response will vary with frequency. The magnitude values of M are normally labelled in dBs. It can further be shown that if the closed loop phase is to remain constant then this also produces a grid of circles, known as N circles. The M circles can be overlaid on a Nyquist plot in MATLAB by using the right hand mouse button and selecting ***grid*** from the resulting options. This is shown in Figure 6.6 for the same transfer function as used in Figures 6.4.and 6.5.

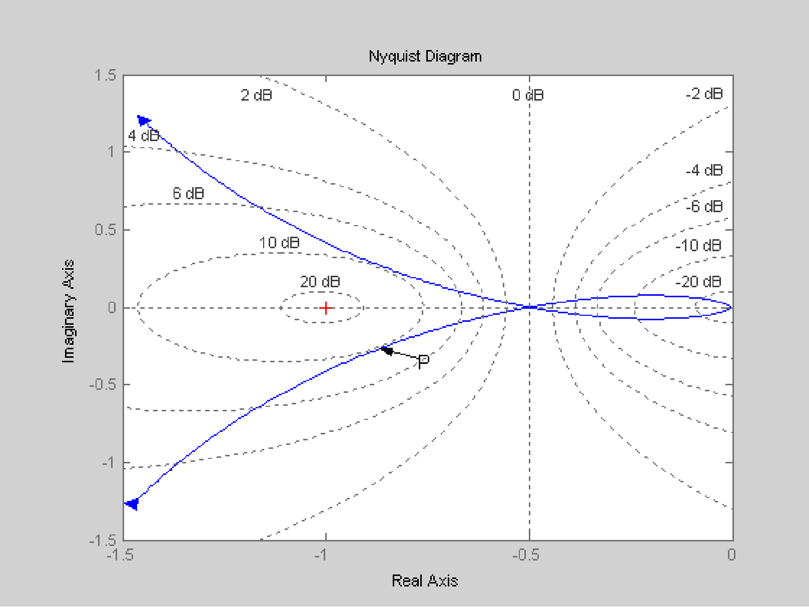

Figure 6.6 Nyquist plot with M circles.

The largest M value circle which the plot reaches is seen to be approximately 10dB at the point P. Thus the closed loop frequency response should have a maximum magnitude M_p of 10dB at the frequency of the point P, which is approximately 1 rad/s., as is seen to be the case in the MATLAB plot of Figure 6.7. The observant reader may also have noticed that the 0dB (unit gain) M circle is in fact a straight line (circle with centre at infinity and infinite radius) through the point (-0.5,0). This means for a typical open loop Nyquist plot it must always stay to the right of this line if the closed loop frequency response must not have a gain greater than unity. Obviously specifying the M_p of a frequency response is another measure of relative stability.

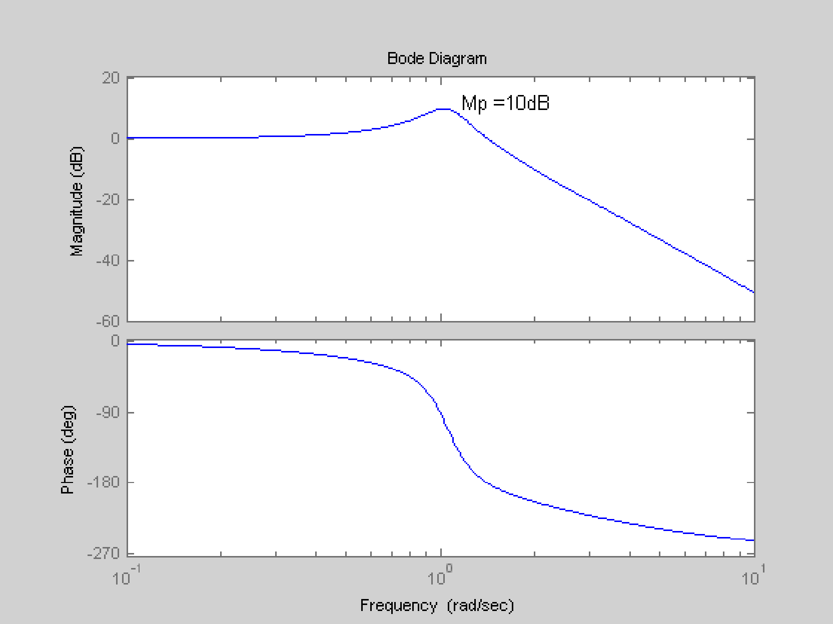

Figure 6.7 Closed loop frequency response for $G_{ol}(j\omega)$ of equation (6.7) with $H = 1$.

7 Classical Controller Design

7.1 Introduction

Classical controller design involves the choice of a suitable transfer function in the controller G_c, or possibly H, of Figure 5.1 so that the closed loop performance meets the required specifications. This can often be achieved with quite simple transfer functions with three common ones being the phase lead controller, the phase lag controller and the PID (Proportional, Integral and Derivative) controller. Since the system specification often includes that there should be no steady state error to a step input, the phase lead and lag controllers, which do not include an integral term, are normally used with plant transfer functions with an integral term. Many plant transfer functions in process control, for example temperature control, do not include an integral term so that PID controllers, or sometimes just PI controllers, are often used to control them. PID controllers are also used on plants with an integration term to eliminate steady state errors caused by a constant disturbance, D, in Figure 5.1; a topic which will be discussed in chapter 9.

Most textbooks discuss the design of phase lead and lag controllers using both frequency domain and root locus methods but here only frequency domain methods will be covered. The main reason for this is that frequency domain methods involve 'loop shaping' which is used in recent approaches to multivariable control. Both methods invariably involve iteration as in the frequency domain approach one is shaping the open loop frequency response and in the root locus approach one is selecting the closed loop poles. As explained earlier the relationship between these properties and the resulting closed loop step response, which is often a system specification, is based on qualitative concepts. For example, it can be seen from equation (5.6) that if the compensator is moved from the forward path to the feedback path, the closed loop transfer function, T(s) changes, but both the open loop frequency response and location of the closed loop poles are unchanged.

7.2 Phase Lead Design

A phase lead controller as stated above is normally used when the plant transfer function *G*(*s*) has an integration. Assuming this to be the case then, with $G_c = K$ and $H = 1$, it will be found that $\lim_{s\to 0} G_c(s)G(s) = K_v / s$, where K_v is a constant and the error to a ramp input will be smaller the larger the value of K_v. This consideration often affects the choice of the controller gain so that the phase lead content of the controller is normally determined assuming $G_c(0) = 1$ so as not to affect K_v. The transfer function of the phase lead controller is therefore taken as

$$G(s) = \frac{1+sT}{1+s\alpha T} \tag{7.1}$$

which produces a lead when α < 1.

A common frequency domain approach for selecting the parameters *α* and *T* for a phase lead controller is to do the design so that the compensated open loop frequency response locus achieves a preselected phase margin, φ°. This is based on the assumption that for a smooth open loop frequency response increasing the phase margin will reduce the overshoot in the step response. There are several ways of trying to achieve this and dependent on the choice of j° for a given *G*(*s*) they may or may not be successful. Possible methods are:-

1) The 'classical' method, which will be described below, and is covered in most textbooks.
2) Choosing the controller zero to cancel the dominant pole of the plant, assuming of course one exists.
3) Designing for a chosen phase or gain crossover frequency, i.e. where the open loop gain is unity or the phase -180°.
4) Fix a, usually based on bandwidth or noise considerations, and find a suitable value of *T*.

Before outlining the procedure of (i) a few facts regarding the phase lead network are needed. The derivations are straightforward and can be found in many textbooks on control. The network gives a maximum phase lead, ϕ_m at the frequency $\omega_m = \frac{1}{T\sqrt{\alpha}}$ of $\sin^{-1}\frac{1-\alpha}{1+\alpha}$ and the corresponding gain is $-20\log_{10}\sqrt{\alpha} = -10\log_{10}\alpha$. Note that on the Bode diagram with the logarithmic scale for frequency, the frequency ω_m lies half way between the two break points, $1/T$ and $1/\alpha T$, and the corresponding gain in dB is half the gain of the starting and finishing gains Values of the phase lead and the corresponding gain for choices of α are given in Table 7.1 and Fig 7.1 shows a Bode diagram for the phase lead network for T = 1 and α= 1/8.

	α	1/2	1/3	1/4	1/5	1/6	1/8	1/10
Max. Lead	φ_m	19.5	30.0	38.9	41.8	45.6	51.1	54.9
Gain at φ_m	$G_m dB$	3.01	4.77	6.02	6.99	7.78	9.03	10.00

Table 7.1 Phase lead network parameters.

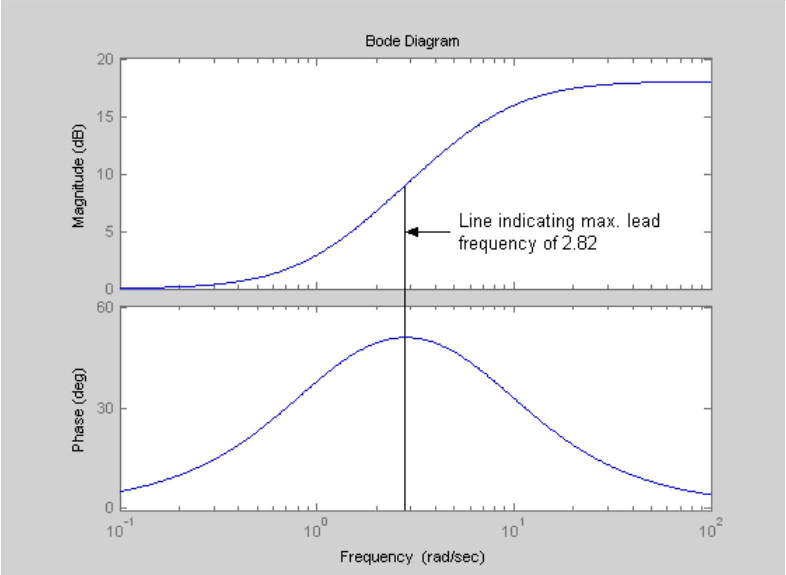

Figure 7.1 Bode diagram of phase lead with T = 1 and α = 1/8.

The procedure for (i), if the desired phase margin is φ, is then as follows:-

1. Evaluate the uncompensated system phase margin ϕ.
2. Allowing for a small amount of safety, e, estimate the required phase lead, $\phi_m = \varphi - \phi + \varepsilon$ (ε typically 5°-18°).
3. Evaluate α for this value of ϕ_m from the above equation (or use Table 7.1).

4. Evaluate $10\log_{10}$ α (or take from Table 7.1 where G_m dB = $-10\log_{10}$ a *dB*) and determine the frequency where the uncompensated Bode frequency magnitude curve is equal to $10\log_{10}$α *dB*. This frequency is the estimated new 0 *dB* crossover frequency and ω_m simultaneously (if the guess for e is correct), because the compensation network provides a gain of $-10\log_{10}$ α at ω_m.
5. Draw the compensated frequency response, check the resulting phase margin, and repeat the steps from 2 if necessary (i.e. change *ε*).
6. If the design does not meet the specifications repeat for a different choice of phase margin, i.e. increase if the overshoot is too high.

The problem with the approach is in estimating e although it is much easier interacting with Bode plots in MATLAB than it was with sketched Bode plots and pencil and paper. Critical to the success of the method is the rate of change of phase of the plant transfer function beyond the frequency of the uncompensated system phase margin. If this is too high the method will fail. This also affects the estimate that should be taken for e. A guide, is 5°-10° for a plant of relative degree 2, and 12°-18° for one of relative degree 3.

As an example consider a plant with transfer function

$$G(s) = \frac{6}{s(1+0.5s)(1+0.1s)} \tag{7.2}$$

If after doing the Bode plots the command ***margin(G)*** is given in Matlab then the values of the gain and phase margins for the transfer function will be given on the Bode diagram as shown in Figure 7.2. The phase margin is seen to be 15.6°. For a second order system a phase margin of 40° corresponds to an overshoot of slightly over 25%, so let us assume the phase lead compensator is required to produce a phase margin of around 40°.

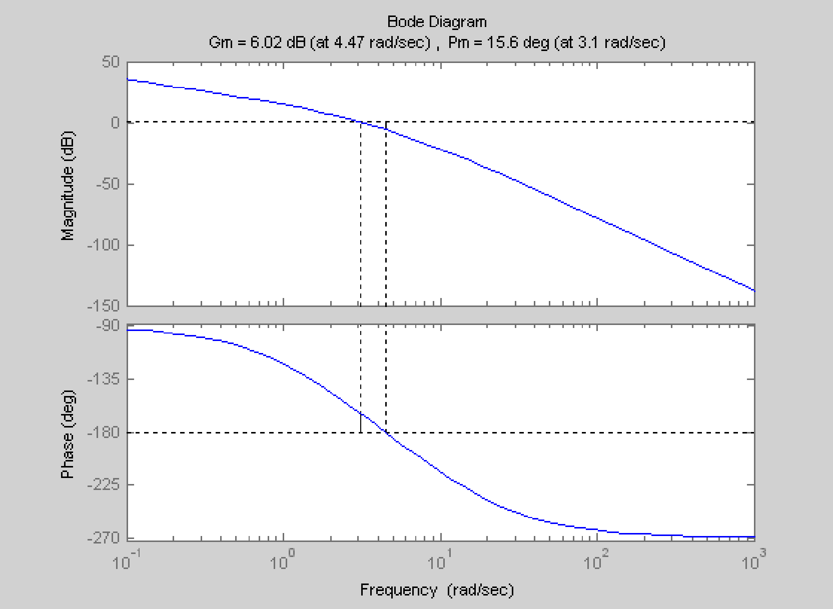

Figure 7.2 Bode diagram for the transfer function G(s)

The extra phase required to increase the phase margin to 40° is (40-15.6+ε)° is 40° if ε is taken equal to 15.6° which is reasonable for a transfer function with relative degree 3. From Table 7.1 this suggests an α of either 1/4 or 1/5. Taking the latter value means ε = 17.4° and from Table 7.1 the dB for α = 1/5 is 6.99. From Figure 7.2 the gain plot is approximately 7dB down at a frequency of 4.73, which is slightly higher than the phase crossover frequency. Thus 4.73 = 1/T√0.2, giving T = 0.472. The compensator is therefore taken as $G_c(s) = \frac{1+0.472s}{1+0.094s}$. Figure 7.3 shows the compensated open loop frequency response together with that of the plant alone. It can be seen that the guess for e was very good with the resulting gain crossover frequency being 4.72 and the phase margin 39.6°. Use of a phase lead compensator is seen to increase the gain and phase crossover frequencies and the open loop bandwidth. It therefore can be expected that the closed loop step response of the compensated system will be faster than that for the plant with a unit gain compensator. These step responses are shown in Figure 7.4 where the overshoot of the phase lead compensated system is significantly less as expected because of its increased phase margin. The overshoot of the compensated system is, however, slightly more than 25%. It is of interest to compare this closed loop response with that obtained with the phase advance compensator placed in the feedback path and this is shown in Figure 7.5. The compensator in the feedback path is seen to result in a response which has a longer rise time, a comparable settling time, and no overshoot. Another interesting aspect of placing the compensator in the feedback path is seen by looking at the controller output signals, normally called the control signals, for step inputs. These are shown in Figure 7.6.

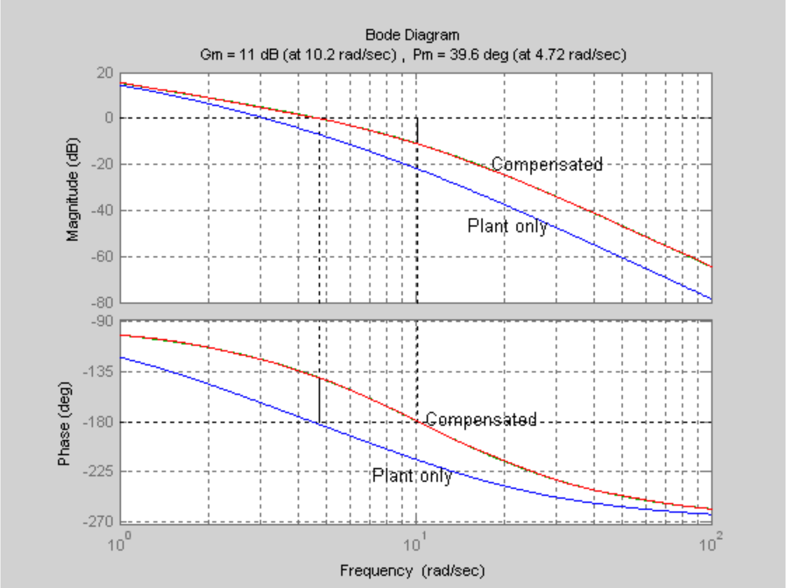

Figure 7.3 Open loop frequency responses of the compensated and uncompensated systems.

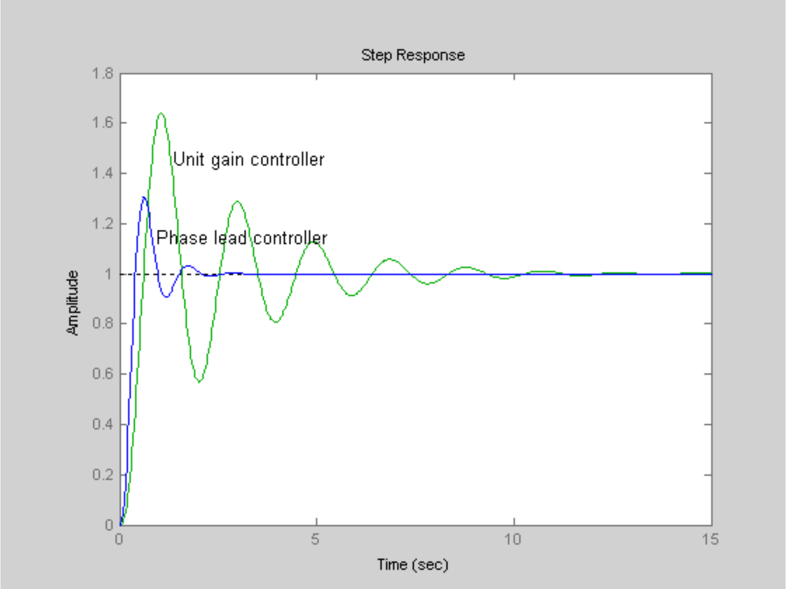

Figure 7.4 Closed loop step responses of the compensated and uncompensated systems.

As the relative degree of the phase advance compensator transfer function is zero, its initial output in response to a step input is 1/α, in this case 5. Thus the closed loop response for the compensator in the forward loop produces a 'derivative kick' of 5. This accounts for the faster rise time but also means that the control signal may cause saturation for large step inputs. In contrast, with the compensator in the feedback path the control signal starts from zero and reaches, in this case, a maximum of around unity.

Before concluding phase lead controller design a few comments on the other suggested approaches (ii) to (iv) are appropriate. First with respect to method (ii), the dominant plant pole in the example $G(s)$ of equation (7.2), i.e the largest time constant, is 0.5, so using this method the phase advance controller transfer function would be $G_c(s)$ = $\frac{1+0.5s}{1+sT'}$, where T' would be evaluated to give the required phase margin. The result for this particular example, choosing a phase margin of 40°, gives T' = 0.09, which makes the compensator almost the same as in (i) due to the nearness of 0.5 to 0.472. With method (iii), which is usually done for the gain crossover frequency, the problem is selecting how much higher this frequency should be than for the plant

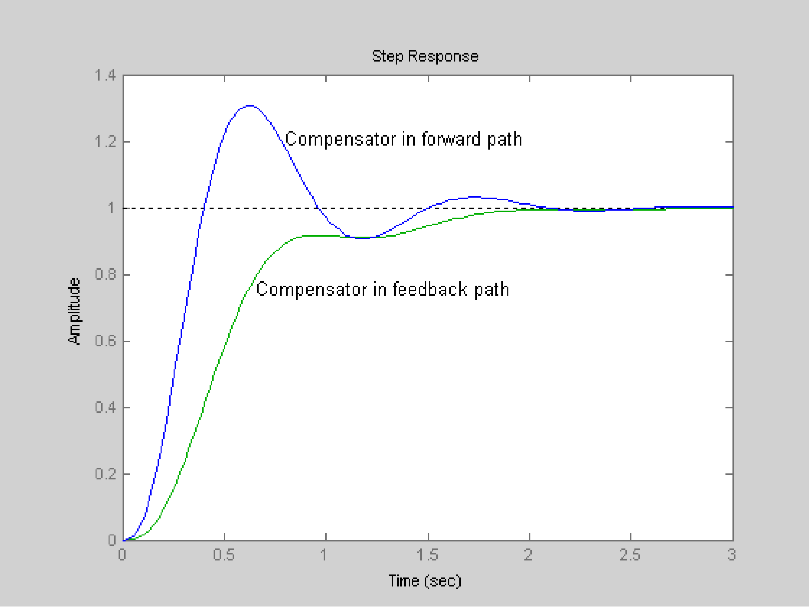

Figure 7.5 Comparison of step responses for forward path and feedback path locations of the compensator.

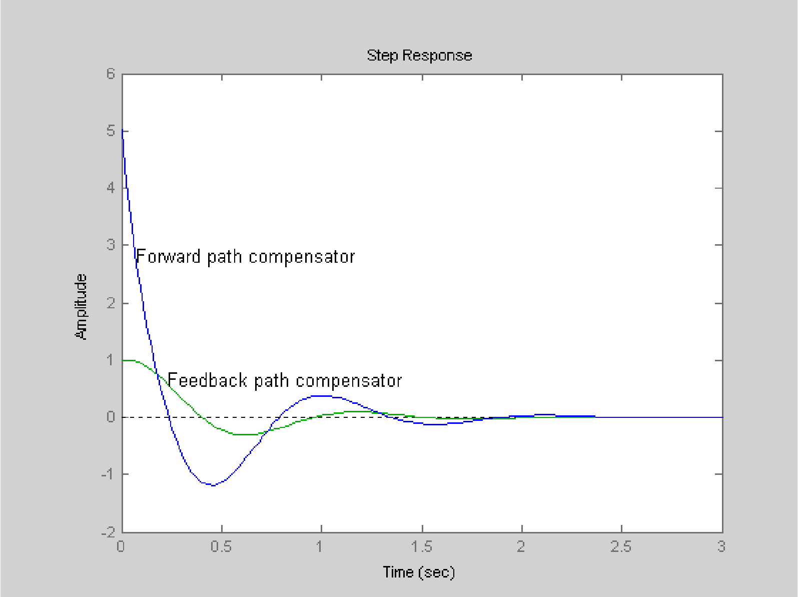

Figure 7.6 Corresponding control signals for the different compensator positions.

alone. If it is selected too high a design will not be possible. Method (iv) is straight forward, a value of α, usually in the range 1/8 to 1/16, is selected and then T found to give the desired phase margin. If in the above example α is selected as 1/8, then T can be found by iteration in MATLAB to be about 0.19, with a corresponding gain crossover frequency of 3.41 rads/s. Interestingly this open loop frequency response has a higher gain margin but a smaller phase margin than the design using method (i) as shown in Figure 7.7. Lead compensation may not be possible in some cases as it depends very much how rapidly the phase of the plant transfer function changes beyond the existing uncompensated gain crossover frequency.

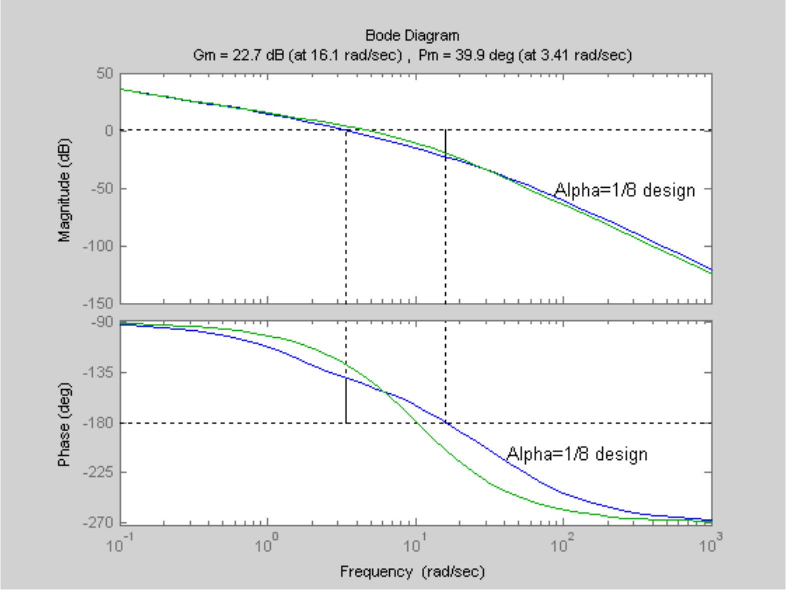

Figure 7.7 Comparison of Bode diagrams for method (i) with a design using an alpha of 1/8.

7.3 Phase Lag Design

A phase lag compensator is achieved with the transfer function of equation (7.1) with α >1. In doing a phase lag design one uses the fact that the compensator gain changes from 0dB at low frequencies to $20\log_{10}(1/\alpha)$dB = $-20\log_{10}(\alpha)$dB at high frequencies. The Bode diagram for the phase lag transfer function $G(s) = \frac{1+sT}{1+\alpha sT}$ is shown in Figure 7.8 for α = 10 and T=1. The phase lag, δ, a decade above the second break point is $\delta = \tan^{-1} 10 - \tan^{-1} 10\alpha$, which depends upon α, with values of 2.85°, 4.27°, 4.99° and 5.13° for α = 2, 4, 8 and 10 respectively. The corresponding gain differs by less than 0.05dB from the asymptotic value of $-20\log_{10}\alpha$. The idea is to have this point as the gain crossover frequency of the compensated locus. Thus if the required phase margin of the compensated system is φ, then one needs to find the frequency ω, where $\arg G(j\omega) = -(180- \varphi) - \delta$ and $|G(j\ \omega)| = 20\log_{10}\alpha$.

As an example the same plant as previously, given by equation (7.2), is taken for a phase lag compensator design with again the requirement for the compensated system to have a phase margin of 40°. Assuming $\delta = 4°$ the frequency where $G(j\omega)$ has a phase of -136° is required. From the Bode diagram this is approximately at 1.51rads/s. where the corresponding gain is 9.89dB, a gain of 3.12. Thus, 10/T = 1.51, $\alpha = 3.12$ and the required compensator transfer function is $G_c(s) = \frac{1+6.62s}{1+20.7s}$.

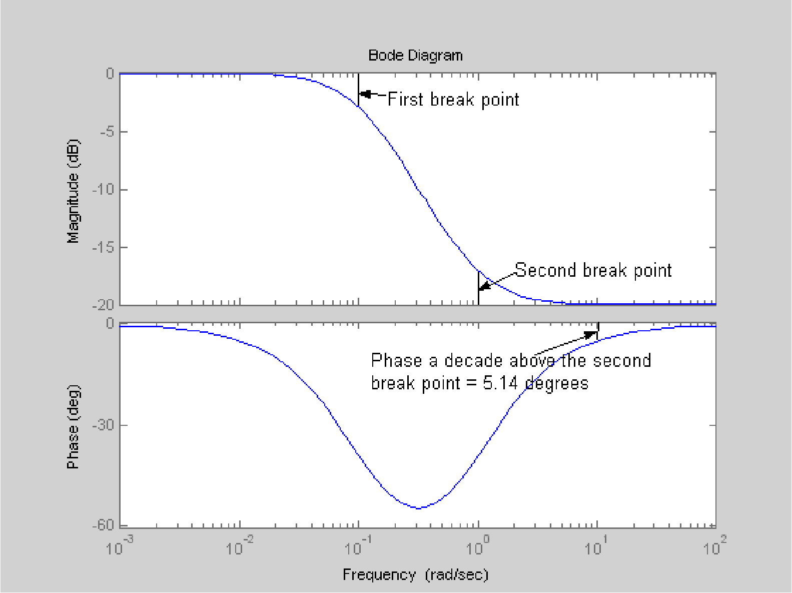

Figure 7.8 Bode diagram of lag compensator.

The Bode diagrams for the plant alone and the lag compensated system are shown in Figure 7.9. The bandwidth of the compensated system with the lag network is lower and the closed loop step response is slower. This response and that for the phase lead compensated system with the compensators in the forward path are shown in Figure 7.10. Note the % overshoot for both plots is roughly the same. The lag compensator does produce a small initial jump in the control signal to 1/3.12 which is roughly its peak value. Unlike lead compensation the use of lag compensation in the feedback path produces very poor results due to the delay it causes to the feedback signal.

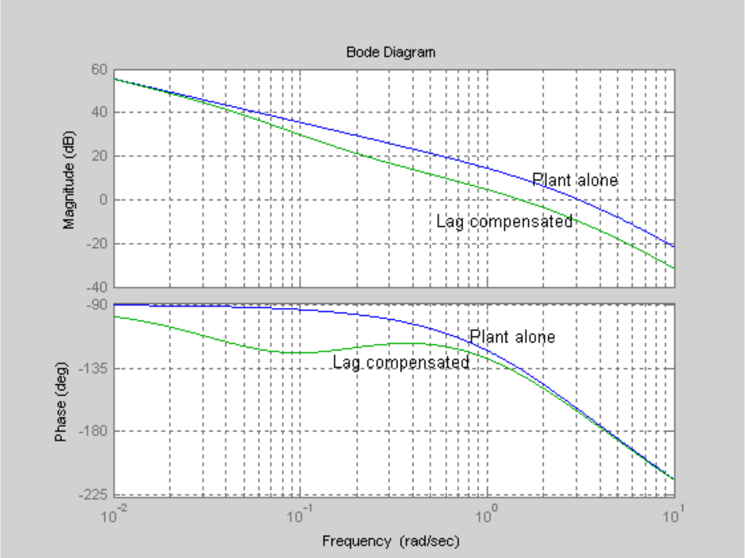

Figure 7.9 Bode diagrams for the plant alone and the lag compensated system

For this example a very much slower response, with an overshoot of over 250%, results. Unlike lead compensation lag compensation is usually possible, however since it slows down the response and reduces the bandwidth it may not be desirable.

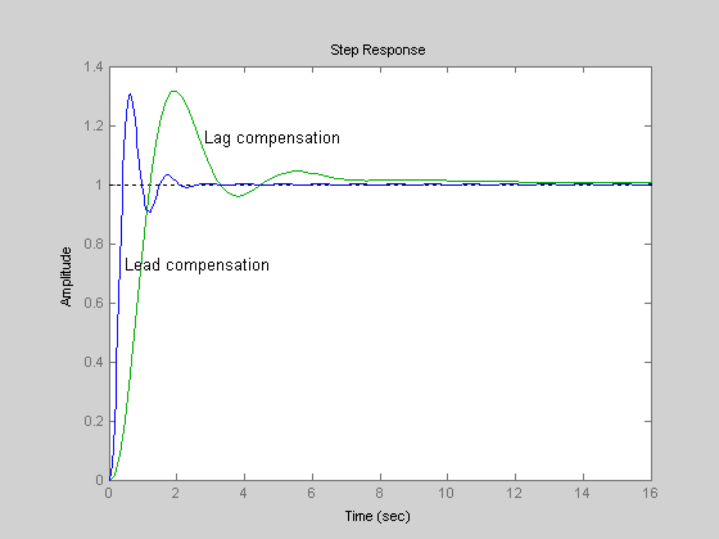

Figure 7.10 Closed loop step responses for lead and lag compensated systems.

7.4 PID Control

Most plants in the process industries do not contain an integration term in their transfer function. It has been seen that it is necessary to have an integration term in the forward path to achieve zero error in the steady state response to a reference step input, so an integration term is normally required in the controller for these plants. The ideal phase lead controller with α = 0, is a PD, that is proportional plus derivative, controller. Thus the use of a PID controller containing proportional, integral and derivative terms is a logical form of fixed term controller for plants without an integration term in their transfer function. PID controllers have thus been used extensively in the process control industries for many years. PID control was first implemented with pneumatic controllers and subsequently went through the use of vacuum tubes, transistors, integrated circuits to today's situation where it is typically software in a microprocessor.

There are various ways in which the controller may be implemented with most academic papers considering its representation by the ideal transfer function

$$G_c(s) = K_c(1 + sT_d + [1/sT_i]) \tag{7.3}$$

with the loop error as input.

An alternative form with real zeros only which is also frequently used is

$$G_c(s) = K_c^{'}(1 + sT_1)(1 + [1/sT_2]) \tag{7.4}$$

The above can be converted to the former using, $K_cT_2 = K_c'T_i$, $K_c = K_c'(T_1+T_2)/T_2$, and $K_cT_d = K_c'T_1$. Sometimes the derivative term is fed from the output rather than the error, which will be denoted PI-D. This avoids the 'derivative kick' discussed earlier with respect to the phase lead compensator. Also in practice the derivative term has an additional time constant being of the form $sT_{\mathrm{d}}/(1 + s\alpha T_{\mathrm{d}})$, with α typically around 0.1.

7.4.1 The Ziegler–Nichols Approach

The earliest work usually referenced on PID control is that of Ziegler and Nichols (Z-N) [7.1], which related to identification and control, the idea being to present techniques which could be used to set the parameters of the PID controller by process commissioning engineers. Thus the procedure is often known as controller tuning and Z-N suggested the following two methods

Method 1.

An open loop step response identification of the plant was suggested with the resulting response modelled ('fit') by a first order plus dead time (FOPDT) transfer function.

Based on the FOPDT model

$$G(s) = \frac{K_p e^{-s\tau}}{1 + sT} \tag{7.5}$$

they suggested the controller parameters be set according to Table 7.1

Type	K_c	T_i	T_d
P	$T/\tau K_p$		
PI	$0.9T/\tau K_p$	3.33τ	
PID	$1.2T/\tau K_p$	2τ	0.5τ

Table 7.1 Z-N Method 1 Parameters

Method 2

They suggested that with the controller 'in situ' in the loop it should be put into the P mode and the gain turned up until an oscillation took place. The gain, of the P term, known as the critical (or ultimate) gain, K'_c, and the frequency of the oscillation, $\omega_c=2\pi/T_c$, known as the critical (ultimate) frequency were then recorded.

Based on these values the controller parameters should then be set according to Table 7.2.

Type	K_c	T_i	T_d
P	$0.5K_c'$		
PI	$0.45K_c'$	$0.8T_c$	
PID	$0.6K_c'$	$0.5T_c$	$0.125T_c$

Table 7.2 Z-N Method 2 Parameters

The first method is a classical identification and then controller design approach. Step responses were regularly used in the early days of control for identification and trying to fit to another model, if felt appropriate, say from physical considerations could be done. The FOPDT model, however, is often a good estimate for many processes. Others that have been used are second order plus time delay (SOPDT) and a single time constant to a large power, say 6 or higher. There is no reason why after identifying an FOPDT model another design method should not be used to set the controller parameters. The Z-N design method was based on achieving around a 25% overshoot to a set point step response.

The second method is based on the fact that if a simple linear feedback loop becomes unstable it will do so at a frequency where the phase shift is -180°. Thus, in principle, the approach gives the frequency, ω_c, and the gain, $1/K_c$', of the point at -180° on the plant frequency response. From a theoretical viewpoint the method is fine but it suffers from many practical problems. These include:-

1. Even if the loop were linear the fact that many processes have very long time constants makes it extremely difficult and time consuming to try and find the gain, K_c'.
2. It can be dangerous if there is no satisfactory limiting effect in the loop as an adjustment to an over estimated value of K_c' can result in a large oscillation.
3. Many practical loops are nonlinear. Saturation is helpful in limiting the amplitude of the oscillation but dead zone effects make finding K_c' even more difficult.

7.4.2 Time Scaling and the FOPDT Plant

Time scaling and amplitude scaling were very familiar to users of an analogue computer. Time scaling was seen to be useful in plotting step responses in chapter 3 since it basically reduces the parameter dependence by one. Here its relevance to selecting (or tuning) controller parameters for a PID controller controlling an FOPDT plant is discussed. If for the transfer function of equation (7.5) a normalised s, s_n, is taken equal to sT and with $\rho = \tau/T$ the transfer function becomes

$$G(s_n) = \frac{K_p e^{-s_n\rho}}{1+s_n} \tag{7.6}$$

which can be referred to as a normalised FOPDT transfer function.

The normalised transfer function has a unit time constant and only two parameters K_p and ρ. The actual system has a step response which is T times slower and markings on its frequency response T times smaller. If this normalised plant is controlled by an ideal PID controller in the error channel with the transfer function of equation (7.3) then this becomes

$$K_c(1+s_nT_d^{'}+[1/s_nT_i^{'}]) \tag{7.7}$$

where $T'_d = T_d/T$ and $T'_i = T_i/T$ and the normalised open loop transfer function can be written

$$G_{nol}(s)=\frac{Ke^{-s_n\rho}}{1+s_n}(1+s_nT_d^{'}+[1/s_nT_i^{'}]) \tag{7.8}$$

where $K = K_pK_c$. This means that if the controller parameters are designed based on some property of the open or closed loop transfer function, the results will be of the form:-

$K = f_1(\rho)$, $T'_i = f_2(\rho)$, and $T'_d = f_3(\rho)$.

Thus for any FOPDT plant the controller parameters must be of the form

$K_c = f_1(\rho)\ /K_p$, $T_i = Tf_2(\rho)$, and $T_d = Tf_3(\rho)$.

for the same performance property to be maintained for all plants which can be time scaled to the same normalised plant. This will be referred to as consistent tuning. Method 1 of Z-N is easily seen to be consistent with a simple choice for the functions f_1, f_2 and f_3 of being inversely proportional, proportional, and proportional to ρ, respectively. It is also easy to show that Method 2 is also consistent. Table 7.3 lists the functions of ρ for several consistent tuning formula which have been suggested. Apart from Z-N method 1 the others are Cohen and Coon (C-C) [7.2], Zhuang and Atherton (Z-A) [7.3] and Wang, Juang and Chan (W-J-C) [7.4]. Figures 7.11 to 7.13 show these relationships graphically for both Z-N methods and the others. The C-C results are omitted from f_1 as its value at small values of ρ becomes large. The a and b parameters in the Z-A method depend on the integral performance tuning criterion used.

The book by O'Dwyer [7.5] gives a large number of so-called tuning rules for PID controllers but they are unfortunately not given in the normalised form which has been demonstrated here for the FOPDT plant. Two other plant transfer functions which can also be normalised in terms of the parameter ρ are

$$G_1(s)=\frac{K_pe^{-s\tau}}{s(1+sT)} \tag{7.9}$$

and

$$G_2(s)=\frac{K_pe^{-s\tau}}{(1+sT)^n}, \tag{7.10}$$

For time scaling consistency the required controller parameters must again be in the form $K_c = f_1(\rho)/K_p, T_i = Tf_2(\rho)$ and $T_d = Tf_3(\rho)$.

	f$_1$(ρ)	f$_2$(ρ)	f$_3$(ρ)
Z-N (Method1)	$1.2/\rho$	2ρ	0.5ρ
C-C	$\frac{16+3\rho}{12\rho^2}$	$\frac{\rho(32+6\rho)}{13+8\rho}$	$\frac{4\rho}{11+2\rho}$
Z-A	$a_1\rho^{b_1}$	$\frac{1}{a_2+b_2\rho}$	$a_3\rho^{b_3}$
W-J-C	$\frac{(0.53+0.73\rho)(1+0.5\rho)}{\rho(1+\rho)}$	$1+0.5\rho$	$\frac{0.5\rho}{1+0.5\rho}$

Table 7.3 Functions of Rho for Different Tuning Formulas.

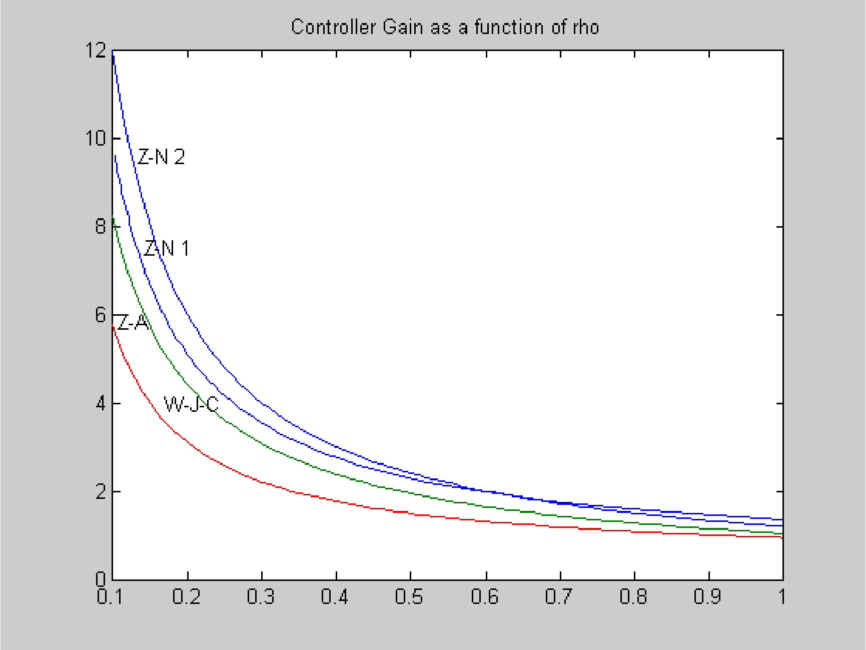

Figure 7.11 Gain $K (= K_c K_p)$ as a function of rho

7.4.3 Relay Autotuning and Critical Point Design

The principle of the Z-N second method is very useful as it can be used in closed loop but the difficulty of adjusting the P, as mentioned earlier was a practical difficulty. With the advent of microprocessor controllers, however, Astrom and Hagglund [7.6] suggested a much more suitable method for practical implementation for estimating the critical point.

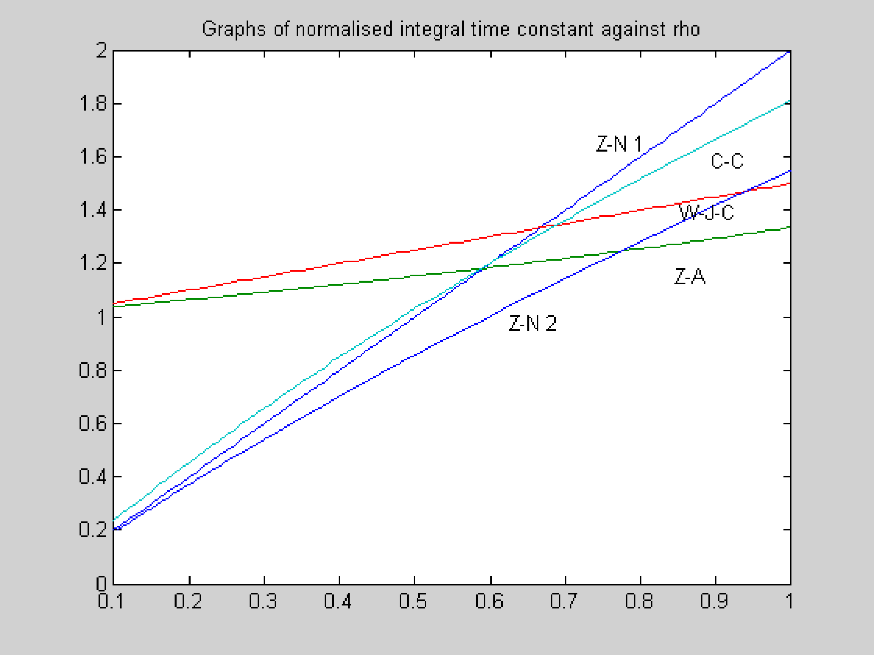

Figure 7.12 Graph of normalised integral time against rho

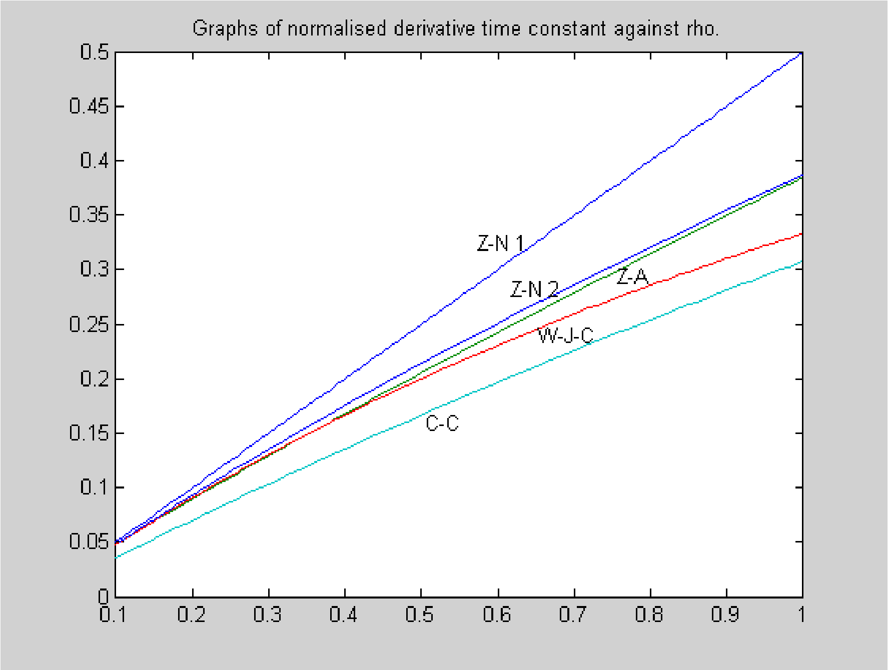

Figure 7.13 Graph of normalised derivative time against rho.

This involved replacing the P term by an ideal relay function to obtain a limit cycle. It can then easily be shown using a describing function (DF) analysis that the frequency of the limit cycle, ω_o, is approximately the critical frequency, ω_c, and the critical gain, K_c, is given approximately by $K_c = 4h/a\pi$, where $2h$ is the peak to peak amplitude of the relay output and, a, is the fundamental frequency amplitude of the limit cycle. It can be shown that the estimate for ω_c will be better than for K_c and that the results will be better the nearer the limit cycle at the relay input is to a sinusoid. The error introduced by replacing a by half the peak to peak amplitude of the limit cycle is usually quite small and is normally done in practice because of the ease of measurement.

Thus with a more practical approach for estimating the critical point it is appropriate to comment further on use of the critical point in PID tuning. First what is the principle of the Z-N method 2 in using the critical point for tuning? Since all that is known about the plant is its critical point then all one can do in selecting the controller parameters is to place this frequency at a known point on the compensated open loop frequency response locus. Since in the Z-N method T_i is taken equal to $4T_d$, which corresponds to the two zeros of the PID controller transfer function being real and equal, it is easy to show that for the PI controller this point is 0.46 arg -192° and for the PID controller it is 0.66 arg -155°.

This concept is a useful design approach and if felt appropriate a different point can be chosen, within the allowable range. Also since one only has freedom to adjust two controller parameters the T_i/T_d ratio may be selected to be other than 4. It is easy to show with $T_i/T_d = 4$, that for the FOPDT plant the controller parameters are obtained from the following equations for moving the critical point to g arg – (180 – φ)°.

$$\tan^{-1}\omega_o + \rho\omega_o = \pi \tag{7.11}$$

$$T_d^{'} = \frac{1+\tan(\varphi/2)}{2\omega_o(1-\tan(\phi/2)} \tag{7.12}$$

$$K = \frac{4g\omega_o T_d^{'}(1+\omega_o^2)^{1/2}}{1+4\omega_o^2 T_d^{'2}} \tag{7.13}$$

Critical point design is a useful concept since closed loop performance is very dependent on the open loop frequency response in the region of the Nyquist critical point (-1, 0). Certainly if one has little knowledge of the plant dynamics it can be very useful as illustrated here for a plant with transfer function $G(s)$ and a possible reduced order model $G_{\mathrm{r}}(s)$, for which the step responses are shown in Figure 7.14. The difference between the step responses is very small and it would be difficult to detect with noisy measurements. However, the frequency responses shown in Figure 7.15 are quite different around the Nyquist critical point, indeed $G(s)$ has a finite gain margin whilst that of $G_r(s)$ is infinite.

Thus designing a controller based on what appears to be a reasonable reduced order model from step response information would be extremely poor compared with use of an approximate critical point. The trouble with a step response is that the higher frequency information is contained in the early part of the response. For the interested reader the two transfer functions are

$$G(s) = \frac{320(s+1)}{(s+0.5)(s+2)(s+4)(s+8)} \tag{7.14}$$

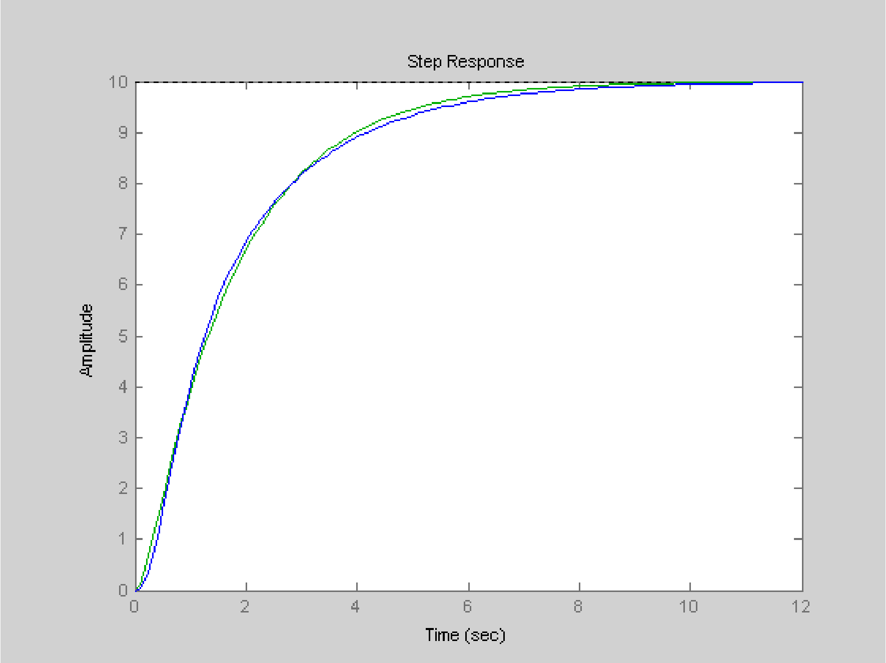

Figure 7.14 Step responses of $G(s)$ and $G_r(s)$.

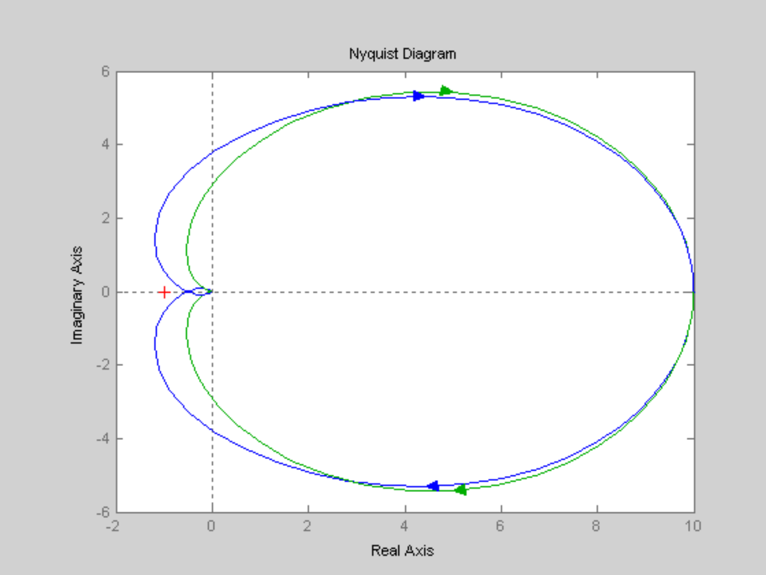

Figure 7.15 Frequency responses of $G(s)$ and $G_r(s)$

and

$$G_r(s) = \frac{35.91}{s^2 + 6.537s + 3.591} \tag{7.15}$$

Before concluding this section it is probably worth mentioning that although relay autotuning has been primarily associated with PID controllers it can also be used for other simple controllers, such as phase lead and phase lag [7.7]. Also by including known networks with the relay, additional information can be obtained, for instance including a tuned filter at the estimated limit cycle frequency will make the limit cycle almost sinusoidal and yield more accurate critical point information.

7.4.4 Further Design Aspects

When a transfer function model is available for a plant for which a PID controller is to be used then a frequency response approach to achieve certain properties, say a phase margin as used for the lead and lag network designs, can be used. Often to make it easier this is done with a fixed ratio for T_i/T_d, typically 4 since the approximate Bode amplitude diagram for this is a 'V' shape. Pole placement designs are also often suggested but these have a major difficulty that for the ideal PID controller in the error channel one has two zeros in the closed loop transfer function. Their effect on the closed loop response is not easy to predict and their location is affected by the choice of the poles. In general the best method of design for selecting parameters of fixed form controllers is to use optimisation methods, which will be discussed in the next chapter.

Practical PID controllers always have a facility to prevent 'integral windup' that is a mechanism, and many algorithms are used, for stopping the integrator integrating when plant input or actuator saturation occurs. Also it is quite common for PID controllers to be sold in pairs as they are often used in cascade in process control, as illustrated in the block diagram of Figure 7.16. The set point for the inner loop controller comes from the outer loop controller and two measurements are available as feedback from the process. The main advantage is to obtain a faster reaction to the inner loop disturbance D_1. But often an improved input-output response can also be achieved. Also the inner loop controller is often set in just the P or PI mode.

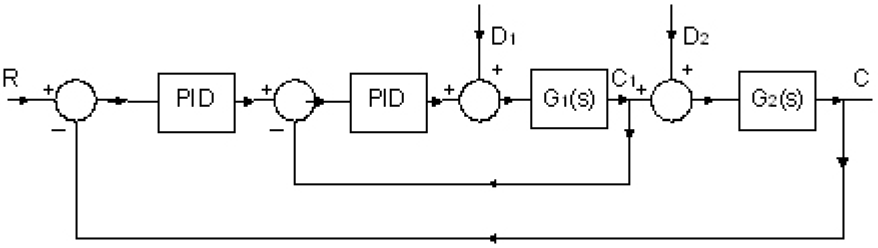

Figure 7.16 Block Diagram of Cascade Control.

7.5 References

7.1 Ziegler J.G., Nichols N.B. Optimum settings for automatic controllers. Transactions of ASME, 1942, 64:759-768.

7.2 Cohen G.H., Coon G.A. Theoretical considerations of retarded control. Transactions of ASME, 1953, 827–834.

7.3 Zhuang M., Atherton D.P. Automatic tuning of optimum PID controllers. Proceedings of IEE, Part D, 1993, 140:216-224.

7.4. Wang F.S., Juang W.S., Chan C.T. Optimal tuning of PID controllers for single and cascade control loops. Chemical Engineering Communications, 1995, 132:15-34.

7.5 O'Dwyer, A. Handbook of PI and PID controller tuning rules, 2006, Imperial College Press, UK.

7.6 Astrom, K.J. and Hagglund, T. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica 1984 Vol 20, pp. 645–51.

7.7 Atherton D.P. and Boz A.F., Autotuning of Phase Lead Controllers. 4th IFAC Workshop on Algorithms and Architectures for Real-Time Control (AARTC '97), Vilamoura, Portugal, 9–11 April 1997, pp. 111–116.

8 Parameter Optimisation for Fixed Controllers

8.1 Introduction

The basic concept here is to optimise the controller parameters to meet a performance criterion. Before the prevalence of digital computers criteria were put forward for which analytical results could possibly be found, or computations could be done using analogue simulation. A logical choice was to choose a criterion based on minimisation of the error over time, as the objective of good control is to maintain a minimum error between the desired and actual output,. Thus integral performance criteria of the form

$$J = \int_{-\infty}^{\infty} f(e(t),t)dt \tag{8.1}$$

where $f(e(t),t)$ is a function of time and the time varying error, were suggested. Typical criteria used are summarised in Table 8.1.

Function *f*	**Name**
$\|e(t)\|$	Integral absolute error – IAE
$t\|e(t)\|$	Integral time absolute error – ITAE
$t^2\|e(t)\|$	Integral time squared absolute error – IT^{2}AE
$e^2(t)$	Integral squared error – ISE
$[te(t)]^2$	Integral squared time error – ISTE
$[t^n e(t)]^2$	Integral squared time to n error – IST^{n}E

Table 8.1 List of Some Error Functions.

It is possible in principle to obtain analytical solutions for the last three since the integral squared error, denoted by J_0, can be found in the s-domain from

$$J_0 = \int_0^{\infty} e^2(t)dt = \frac{1}{2j\pi}\int_{-j\infty}^{j\infty} E(s)E(-s)ds \tag{8.2}$$

which is known as Parseval's integral. It can be evaluated when $E(s) = c(s)/d(s)$ is a ratio of polynomials in s, as given in Table D.1 in Appendix D for low order polynomials $d(s)$, and for higher order polynomials, $d(s)$, can be evaluated using recursion relationships as given by Astrom [8.1]. For the higher time weightings

$$J_n = \int_0^{\infty} [t^n e(t)]^2 dt \tag{8.3}$$

can again, in principle, be evaluated in the *s*-domain by utilising the time multiplication formula of the Laplace transform $\mathscr{L}\left[tf(t)\right]=-\frac{dF(s)}{ds}$ (Theorem (v) Appendix A.). The difficulty for hand calculations is that the order of *d*(*s*) doubles with each differentiation, however, it is easy to write a computer program to compute the results for some low values of *n*. An excellent treatise on the analytical approach which also considers a weighting on the control effort in the performance criterion and possible satisfaction of constraints is reference [8.2].

8.2 Some Simple Examples

Here some simple analytical examples are given to illustrate the approach and bring out some basic ideas. For realistic practical problems, however, results will normally have to be obtained computationally.

Example 1

Consider a feedback loop with $G = 1/s$, i.e. an integrator, and $G_c = K$ a gain. For a unit step input $R = 1/s$ and $E = 1/(s + K)$. Clearly $e = \exp(-Kt)$ and since the expression for *e* is so simple its integral squared value can be found from either the time domain or *s*- domain integral to give the ISE = 1/2*K*. This is as expected, since the maximum phase lag of the loop is 90° it remains stable no matter how high the gain. However, the initial value of the control signal at the input to the plant, given by $u = K\exp(-Kt)$, increases as *K* increases. One way to find a finite gain value is to put a constraint on some function of *u*. A simple solution is to minimise the time domain integral

$$I=\int_0^\infty [e^2(t)+\lambda^2u^2(t)]dt \tag{8.4}$$

which is easily shown, by substituting $u = Ke$, to have the value

$$I=(1+\lambda^2K^2)/2K \tag{8.5}$$

By differentiation, it is found that $K = 1/\lambda$ yields the minimum value for *I* of λ. This example although trivial brings out the point that care has to be taken in obtaining solutions to optimisation problems. It is important to understand the problem so as to know whether a solution will only exist if some constraints are imposed and also when a minimum has been found that it is realistic. Systematic approaches may be necessary, for example if a controller has two variable parameters it may be desirable to fix one initially and just look at the effects of varying the other.

Example 2

Consider $G = 1/s(s + 1)^2$ and $G_c = K$. In this case, it is easily shown that if K is increased the system will go unstable for $K > 2$ so there should be a value of $K < 2$ which minimises the ISE. For a unit step input one obtains

$$E = \frac{s^2 + 2s + 1}{s^3 + 2s^2 + s + K} \tag{8.6}$$

Using Table D.1 to evaluate I_3 gives

$$I_3 = \frac{3K + 2}{2K(2 - K)} \tag{8.7}$$

Differentiating to find the minimum yields $K = 2/3$ and the corresponding minimum value of the performance index is $I_{3\min} = 2.25$. Checking the step response for this value of K shows it to have an overshoot of 36%. Incidentally, a second order transfer function corresponding to the dominant complex pair of poles has an overshoot of 40%.

Example 3

As another example consider the control of a double integrator plant, $G(s) = 1/s^2$, by a phase lead controller with transfer function $(1+ sT)/(1 + asT)$, in both the forward, G_c, and feedback, H, paths. With the controller in G_c, the value of, E, for R a unit step is

$$E = \frac{\alpha s^2 T + s}{\alpha s^3 T + s^2 + sT + 1} \tag{8.8}$$

and using Table D1 for I_3, gives for the ISE,

$$I_3 = \frac{\alpha T^2 + 1}{2T(1 - \alpha)} \tag{8.9}$$

Differentiating with respect to T, shows that the optimum value of $T = a^{-1/2}$, and the corresponding minimum value of I_3 is $a^{1/2}/(1 - a)$. This can be seen to be infinite when $a = 1$, as the system is neutrally stable, and tends to zero as a tends to zero, which results in the derivative kick of the control signal tending to infinity. On the other hand if the controller is placed in H, the closed loop transfer function is

$$\frac{C}{R} = \frac{\alpha sT + 1}{\alpha s^3 T + s^2 + sT + 1} \tag{8.10}$$

The error signal is

$$E = R - C \tag{8.11}$$

so that

$$\frac{E}{R} = 1 - \frac{C}{R} = \frac{\alpha s^3 T + s^2 + sT(1 - \alpha)}{\alpha s^3 T + s^2 + sT + 1} \tag{8.12}$$

giving for a unit step input $R = 1/s$

$$E = \frac{\alpha s^2 T + s + T(1 - \alpha)}{\alpha s^3 T + s^2 + sT + 1} \tag{8.13}$$

Again using the Table D1 one obtains

$$I_3 = \frac{1 + T^2 - 3\alpha T^2 + 3\alpha^2 T^2}{2T(1 - \alpha)} \tag{8.14}$$

Differentiating to find the optimal value of T yields

$$T = \frac{1}{(1 - 3\alpha + 3\alpha^2)^{1/2}} \tag{8.15}$$

and the corresponding value of the ISE is

$$ISE_{\min} = \frac{(1-3\alpha+3\alpha^2)^{1/2}}{1-\alpha} \tag{8.16}$$

Differentiation of this expression shows that the absolute minimum value obtainable is 0.866 when $a = 1/3$ and the value of $T = 1.732$.

Three simple examples have been taken to illustrate the analytical approach to minimising the *ISE*, which corresponds to $n = 0$ in the general criterion of equation (8.3). This has been done to illustrate the procedure whilst keeping the algebra relatively simple. If, for example, one wished to investigate the last example for the *ISTE* criterion, that is $n = 1$, then one has to differentiate $E(s)$ with respect to s. This increases the order of the denominator from 3 to 6, the algebra for the integral becomes 'horrendous' and differentiation of the result is then required for the optimum values. Computationally, however, the minimum can be found very quickly, one selects values for T and a, evaluates the *ISTE* and has an optimisation algorithm built around to adjust T and a to converge to the optimum values, which will of course exist if the compensator is in *H*.

One reason for having covered the classical optimization approach in some detail is that it leads naturally to the consideration of standard forms. These provide an interesting closed loop direct synthesis approach for obtaining controller parameters.

8.3 Standard Forms

Based on the approach of the previous section it is possible to obtain normalised closed loop transfer functions which satisfy error performance criteria. Their value is that they indicate 'good' pole locations for the closed loop transfer function. To illustrate the approach consider a feedback system with $G = 1/s(s + a)$, $G_c = K$ and $H = 1$. For a unit step input $E = (s + a)/(s^2 + as + K)$. The ISE can be found from Table D.1 and since the denominator of E is second order it is denoted, I_2, and is given by

$$I_2 = (K + a^2)/2aK. \tag{8.17}$$

This can be shown to be a minimum for $a = \sqrt{K}$ and the corresponding closed loop transfer function is

$$T(s) = \frac{K}{s^2 + s\sqrt{K} + K}. \tag{8.18}$$

Comparing this with the standard form for the second order equation of

$$T(s) = \frac{\omega_o^2}{s^2 + 2\zeta s\omega_o + \omega_o^2} \tag{8.19}$$

shows that $K = \omega_O^2$, so the natural frequency increases with K but the damping ratio $\varsigma = 0.5$.

This value of ς gives a step response overshoot of around 16%. The value is less than the unit value required for no overshoot in the step response and the value of 0.707 required for no peak in the frequency response.

Time scaling the standard second order equation (8.19), that is replacing s/ω_o by s_n, gives the transfer function

$$T(s_n) = \frac{1}{s_n^2 + 2\zeta s_n + 1} \tag{8.20}$$

This equation is known as the time normalised form and as explained in chapter 3 has exactly the same time response as eqn.(8.19) but eqn. (8.19) is a factor ω_o faster. Eqn. (8.20) with $\zeta = 0.5$ is referred to as the standard form of the second order all pole closed loop transfer function which minimises the ISE, J_0, for $H = 1$. Note also that the forward loop transfer function, GG_c, must contain an integrator to ensure zero steady state error to a step input. Standard forms for any order of the denominator polynomial and for various integral performance criteria can be found and written, with the subscript n dropped from s, as

$$T(s) = \frac{1}{s^j + d_{j-1}s^{j-1} + \ldots.. + d_1 s + 1} \tag{8.21}$$

They have been derived in reference [8.3] for the more general performance index IST^nE for different values of n and are denoted as:-

$$T_{0j}(s) = \frac{1}{s^j + d_{j-1}s^{j-1} + \ldots.. + d_1 s + 1} \tag{8.22}$$

The required coefficient values as well as the resulting value of the performance index are given in Table 8.2. Note the value of the index increases for larger n because of the time weighting factor as the settling time is greater than unity.

It is interesting to look at the coefficients of these transfer functions. First, purely of academic interest, is the fact that for the ISE, that is $n = 0$, all the coefficients are integer values. More important, however, is the fact that as n is increased the coefficients increase in value and the step responses have less overshoot with only a small change in settling time. This can be seen for the second order system as the damping ratio V, equal to $d_1/2$ in Table 8.2, increases as n increases. These points are further demonstrated by Fig. 8.1 which shows the step responses for $j = 4$ for $n = 0$ to 3.

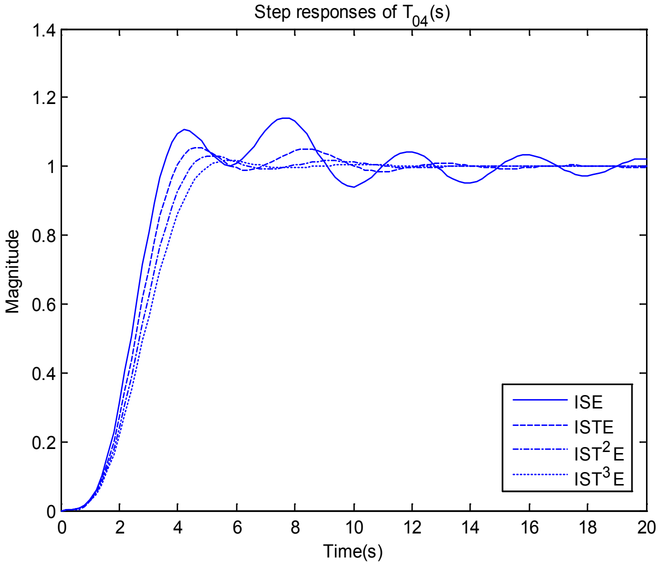

Figure 8.1 Step responses for $j = 4$

$T_{0j}(s)$	*n*	*d_1*	*d_2*	*d_3*	*d_4*	*d_5*	*J_n*
$T_{02}(s)$	0	1.000					1
	1	1.335					0.8686
	2	1.537					3.2823
	3	1.665					28.1005
$T_{03}(s)$	0	2.000	1.000				1.5000
	1	2.042	1.472				2.1142
	2	2.155	1.825				10.400
	3	2.281	2.082				105.1355
$T_{04}(s)$	0	2.000	3.000	1.000			2.0000
	1	2.372	3.072	1.539			4.2355
	2	2.620	3.295	1.990			27.350
	3	2.809	3.577	2.349			329.4304
$T_{05}(s)$	0	3.000	3.000	4.000	1.000		2.5000
	1	3.052	3.897	4.094	1.576		7.4816
	2	3.195	4.572	4.402	2.092		62.0700
	3	3.360	5.129	4.827	2.527		898.3668
$T_{06}(s)$	0	3.000	6.000	4.000	5.000	1.000	3.0000
	1	3.385	6.145	5.489	5.110	1.597	12.1017
	2	3.649	6.570	6.705	5.481	2.157	126.4600
	3	3.862	7.108	7.756	6.026	2.649	2189.100

Table 8.2 All Pole Standard Forms for T_{0j} (From reference 8.4)

Since adding a compensator often results in a closed loop transfer function having a zero, a suggested use for these standard forms in controller design has been to add a prefilter with a pole to cancel the zero. To avoid the use of a prefilter studies have been done to investigate standard forms with a zero, that is for

$$T_{1j}(s) = \frac{c_1 s + 1}{s^j + d_{j-1}s^{j-1} + ... + d_1 s + 1} \tag{8.23}$$

In early work on this topic results were given to minimise the ITAE criterion but with the requirement that the system should also have zero steady state error to a ramp input, which requires the constraint that $c_1 = d_1$. Recently [8.4] results have been derived without this constraint but the required *d* coefficients vary with the choice of c_1, as illustrated in Fig 8.2 for T_{14}(s). It is also possible to plot how the poles of the transfer function vary with the zero parameter c_1, and some of these can be found in reference [8.4]. These plots show that the optimal pole positions vary appreciably with the value of the zero.

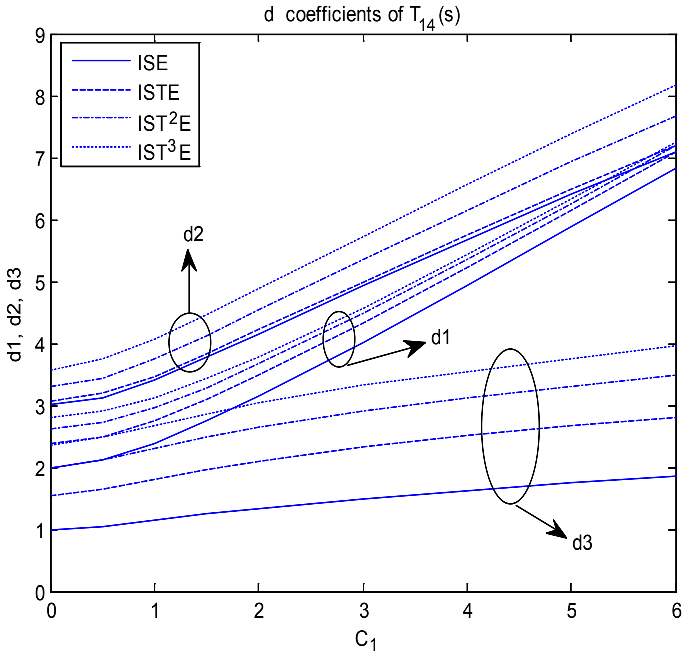

Figure 8.2 *d* coefficients for T_{14} for 3 values of *n*.

The concept of trying to design a controller so that the closed loop transfer function has a specific form is useful, as it addresses the closed loop performance directly, and will be illustrated by an example in the next section, as well as being given further consideration in chapter 11. There are many closed loop standard forms that might be chosen apart from those based on the criteria of Table 8.1. For example, the Butterworth filter form could be selected.

8.4 Control of an Unstable Plant

A simple linearised model for a magnetic suspension is often taken as

$$G(s) = \frac{K_p}{s^2 - \lambda} \tag{8.24}$$

It is required to control this plant transfer function with a PID type controller. If the classical PID controller of equation (7.3) is used in the error channel then the closed loop transfer function, *T*(*s*), is given by

$$T(s) = \frac{K_c K_p (1 + s + s^2 T_i T_d)}{s^3 T_i + s^2 T_i T_d K_c K_p + s(K_c K_p - T_i \lambda) + K_c K_p} \tag{8.25}$$

It can be seen that the 3 poles of the transfer function can be allocated by the choice of the three controller parameters, but the two zeros cannot then be located independently as their location is dependent on the parameters chosen to locate the poles. If, however a PI-PD controller is used, that is a controller whose output is obtained from the error feeding the PI and the plant output the PD, then the open loop transfer function is

$$G_{ol}(s) = K_c(\frac{1+sT_i}{sT_i})(\frac{K_p}{s^2 - \lambda + sK_pT_d + K_fK_p}) \tag{8.26}$$

where the controller parameters are K_c and T_i for the PI terms and K_f and T_d for the PD terms.

This gives the closed loop transfer function

$$T(s) = \frac{K_cK_p(1+sT_i)}{s^3T_i + s^2T_iT_dK_p + sT_i(K_cK_p + K_pK_f - \lambda) + K_cK_p} \tag{8.27}$$

In this case the 3 poles and the zero can be adjusted independently by the controller parameters. To design the controller using the standard form approach equation (8.27) can be written in normalised form as

$$T(s_n) = \frac{1 + s_n\alpha T_i}{s_n^3 + s_n^2(T_dK_p/\alpha) + s_n[(K_cK_p + K_fK_p - \lambda)/\alpha^2] + 1} \tag{8.28}$$

where α is the timescale factor $(K_cK_p/T_i)^{1/3}$ by which the system is faster than the normalised one.

In principle the time scale, α, can be selected by the choice of K_c, and the coefficients for the chosen standard form; d_2 by the choice of T_d, d_1 by the choice of K_f, and c_1 by the choice of T_i. In practice K_c will normally be constrained to an upper value, possibly to limit the initial control effort, and T_i will involve a trade off between the values chosen for α and c_1.

Consider, for example, the case of the plant parameters $K_p = 2$ and $\lambda = 4$ and constraining K_c to a maximum of 1. Then two possible designs could be:-

1) **Time scaling by 2.** This means selecting $\alpha = 2$ which gives $T_i = 0.25$ and $c_1 = 0.5$. For this value of c_1 the values of d_2 and d_1, respectively, to minimise the ISTE are 1.595 and 2.120. This gives $T_d = 1.595$ and $K_f = 5.240$
2) **No time scaling.** This means selecting $\alpha = 1$ which gives $T_i = 2$ and $c_1 = 2$. For this value of c_1 the values of d_2 and d_1, respectively, to minimise the ISTE are 2.215 and 2.991. This gives $T_d = 1.107$ and $K_f = 2.495$.

Because both designs are based on minimisation of the ISTE the closed loop step responses are quite similar in shape with around 10% overshoot, and the first twice as fast as the second. The faster time scaled response in this case has been achieved not by increasing the controller gain K_c but by variations of T_i and c_1.

8.5 Further Comments

The topic of optimising the parameters of a fixed form controller has been discussed in this chapter. Some simple analytical examples based on the ISE integral performance criterion were first discussed and then a simple algebraic approach based on standard forms was given, which will be considered further in chapter 11. It is very common to use fixed form controllers in industrial design and with today's computation facilities optimisation of the parameters to meet the design specifications is an excellent practical approach. References [8.5, 8.6] describe software that has been developed to do designs using optimisation techniques and to show how the various performance constraints might be traded off against each other. When a large number of criteria need to be considered these programs can become very complicated. Optimisation approaches, such as the use of integral performance criteria when there are only a few variable parameters, can often be achieved using simulation, as the performance criteria listed in Table 8.1 can easily be found from a simulation run. One may then either interact with the simulation manually to obtain the optimal parameters or do so with an optimisation program controlling the simulation runs.

8.6 References

8.1 Astrom, K.J. Introduction to Stochastic Control Theory. Academic Press, 1970. pp. 133–9.

8.2 Newton, G.C., Gould, L.A. and Kaiser, J.F. Analytical Design of Linear Feedback Control. Wiley, New York, 1957.

8.3 Atherton, D.P. and Boz, A.F. Using Standard Forms for Controller Design. Proceedings UKACC International Conference on Control 1998 (Control'98), University of Wales, Swansea, September 1998, pp 1066–1071.

8.4 Boz, A.F. Computational Approaches to and Comparisons of Design Methods for Linear Controllers. D.Phil. thesis, University of Sussex, 1999.

8.5 Fleming P.J.: Managing Competing objectives in control systems engineering design: Proc UKACC International Control Conference, Control 2006, Glasgow, 2006.

8.6 Fleming P.J., Purshouse R.C., Lygoe R.J.: Many-objective optimization: An engineering design perspective, Lecture Notes in Computer Science 3410, pp. 14–32, 2005.

9 Further Controller Design Considerations

9.1 Introduction

Additional aspects related to compensator design are covered in this chapter. The first topic discussed in the next section is lag-lead compensator design an extension of the lead and lag compensators discussed in chapter 7. In the next two sections some aspects of speed and position control are discussed with particular emphasis on the rejection of steady state disturbances. It is shown that this requires an integration in the controller which complicates the design for obtaining a good step response. Simple rigid body type plant transfer functions are used in these sections, whereas in many cases it is required to control the speed or position of a shaft which is driven through a flexible link. Typically this results in a transfer function containing not only complex poles but also complex zeros. To illustrate the difficulties of controlling such systems with a series compensator the next section considers the control of a plant transfer function with complex poles.

The final section discusses the problem of the effect of parameter variations on control system performance. Although this is a topic of major interest in design it is a very difficult theoretical one and few results of practical significance have been obtained.

However modern calculation and simulation methods are now so fast that the increase in time required for doing studies with different sets of parameters is usually economically justifiable.

9.2 Lag-Lead Compensation

As mentioned in section 7.2 it may not be possible to achieve a satisfactory phase lead design and the bandwidth achievable by a phase lag design may be less than desired. It may be possible to improve the loop performance by a lag-lead design. This is illustrated by taking a system with the same transfer function dynamics but with a higher gain in the numerator, which might be required to reduce the steady state error to a ramp input, K_v, as mentioned in section 7.2. Consider therefore

$$G(s) = \frac{12}{s(1+0.5s)(1+0.1s)} \tag{9.1}$$

The closed loop system with this G(s) and H(s) = G_c(s) = 1 is neutrally stable so that the phase margin is zero compared with a value of 15.6° for the previously considered transfer function

$$G_1(s) = \frac{6}{s(1+0.5s)(1+0.1s)} \tag{9.2}$$

To add a phase lead network to G(s) to achieve the same phase margin of 40° will require a lead of around 60° which is very high and the design may not be achievable. An alternative is to use a lag-lead design where the gain is reduced by a lag network before the gain crossover frequency is reached. If after adding the lag network the frequency response around the gain crossover frequency is similar to that of G_1(s) then the phase lead network of section 7.2 will be suitable. Thus, choosing a lag network with transfer function

$$G_{c1}(s) = \frac{1+10s}{1+20s} \tag{9.3}$$

and plotting the Bode diagram of the series combination $G_{c1}G$, it is seen to be almost identical to G_1, in the required region, as shown in Figure 9.1.

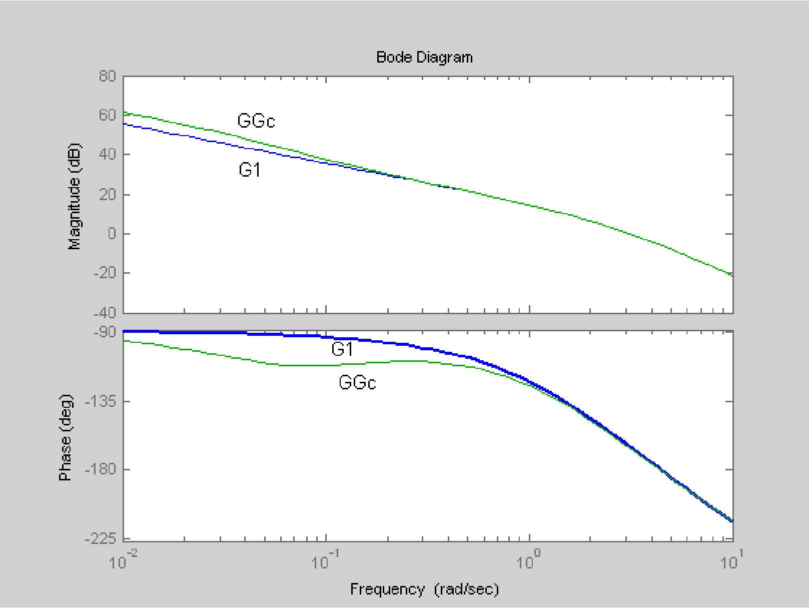

Figure 9.1 Bode diagrams for the example.

Adding the phase lead compensator of section 7.2 the lag-lead compensator is

$$G_c(s) = (\frac{1+10s}{1+20s})(\frac{1+0.472s}{1+0.094s}) \tag{9.4}$$

The resulting system has a phase margin of 39° at a frequency of 4.72 rads/s. The closed loop step response is shown in Figure 9.2 together with that using a lag network design, and as expected an appreciable increase in the speed of response has been achieved.

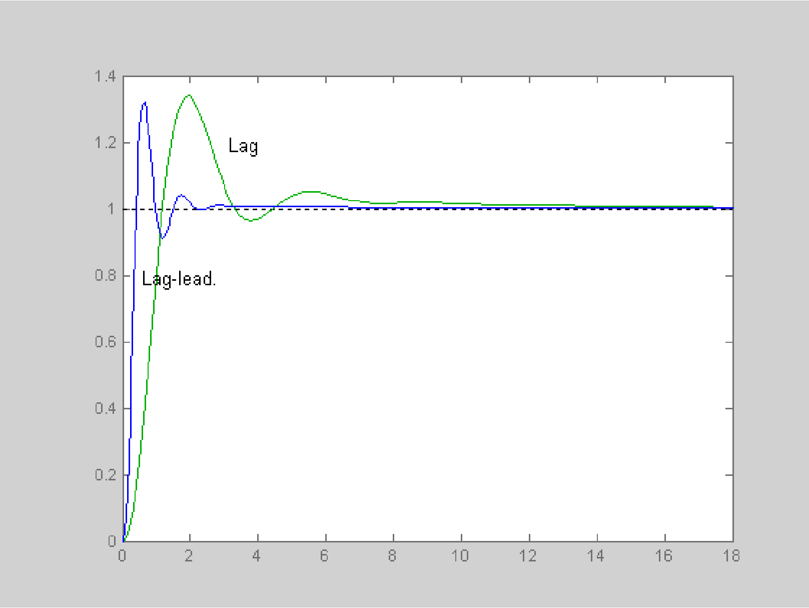

Figure 9.2 Comparison of step responses.

9.3 Speed Control

Control of speed is a common problem encountered by many control engineers, perhaps the most common well known situation being the cruise control fitted to many automobiles. Here it will be assumed that the speed is rotary and the transfer function from the torque to the load, where the speed has to be controlled, is

$$G(s) = K/(1+sT). \tag{9.5}$$

In practice there may be more than one time constant but quite often there is one dominant one as assumed here. A problem which often arises, however, is when the coupling from the drive torque to the load is not rigid and a more complicated transfer function results with both complex poles and zeros. This presents a much more difficult control problem which will be commented on further in section 9.5. The control loop is typically as shown in Figure 5.1, where *N* is assumed zero and *H* will convert speed, say radians per second to voltage with a time constant which is probably small enough to be neglected. In order that the speed should remain constant at the required value with a fixed reference input, *R*, assuming *H* is calibrated correctly, the controller G_c must contain an integration. This can also be seen from Figure 5.1 to be a requirement for a constant disturbance *D* to have no effect on the load speed in the steady state as the output of the controller must have a signal equal and opposite to *D*. Thus, G_c, is typically a PI controller and the open loop transfer function is

$$GG_cH = \frac{K(K_i + sK_P)H}{s(1+sT)} = \frac{k(1+sT_1)}{s(1+sT)} \tag{9.6}$$

where K_P and K_i are respectively the controller proportional and integral gains and $k = KK_iH$ and $T_1 = K_P / K_i$. Moving *H* inside the loop as explained in section 5.3, the closed loop transfer function can be written as

$$T(s) = \frac{k(1+sT_1)}{s(1+sT)+k(1+sT_1)} \tag{9.7}$$

One approach to the design is to choose $T_1 = T$, so that *T*(s) becomes a single time constant transfer function, although such a zero-pole cancellation will never be correct in practice due to uncertainty in the system parameters. This does provide a simple analytical approach and *k* can in principle be increased, by increasing K_i and consequently K_P for a given T_1, as much as required to speed up the system response. In practice *k* will be restricted as increasing *k* increases the maximum magnitude of the controller output signal. Another approach is to do a design based on the open loop transfer function of equation (9.6) with $T_1 \neq T$, using say frequency response or root locus techniques. One point to note is that if the closed loop transfer function is written as

$$T(s) = \frac{\omega_o^2(1+sT_1)}{(s+\omega_o)^2} \tag{9.8}$$

that is with critical damping for the second order denominator, where $\omega_o^2 = k / T$ and $2\omega_o = (1 + kT_1)/T$, then an overshoot may still exist in the closed loop step response due to the zero. It is easily shown that the step response of equation (9.7) is

$$1 - e^{-\omega_o t} - \omega_o t e^{-\omega_o t} + T_1 \omega_o^2 t e^{-\omega_o t} \tag{9.9}$$

and that it has a maximum when $t = T_1/(\omega_o T_1 - 1)$. Thus an overshoot will exist when this is positive, that is for $\omega_o T_1 > 1$.

9.4 Position Control

Many of the early applications of control engineering were involved with position control due to the requirement for accurate position control of guns and other devices during the 1940s. Indeed several of the early textbooks written in control engineering used the word servomechanisms in the title to account for the fact that much of their coverage was related to position or speed control. Today there remain many requirements for accurate position control from large drives and robotics to heads for reading or writing to rotary storage media. Again if flexure in the drive dynamics can be neglected the simple rigid body transfer function for the plant in Figure 5.1 of a position control may often be taken as

$$G(s) = K / s(1 + sT) \tag{9.10}$$

There will be no steady state error to a step input as $G(s)$ contains an integration term, so that a satisfactory closed loop step response may be achieved with G_c(s) a constant gain, phase lead or lag network. Also velocity feedback may be used which means that the transfer function H(s) is of the form 1+sT_1 and the closed loop transfer function will become

$$T(s) = \frac{K}{s(1 + sT) + (K(1 + sT_1)} \tag{9.11}$$

Apart from the dynamic response requirements more stringent steady state requirements are often required of a position control system such as being able to follow a ramp input with no position error or reject the effect of any constant disturbance D on the output. Both these require the controller to have an integration term. If $H = 1$ and G_c is a PI controller $G_c = (K_p + sK_i)$ then the closed loop transfer function is

$$T(s) = \frac{K(sK_P + K_i)}{s^3 T + s^2 + sKK_P + KK_i} \tag{9.12}$$

On the other hand if velocity feedback is used $H = 1 + sT_1$ and with $G_c(s) = K_c/s$ the closed loop transfer function is

$$T(s) = \frac{K_C K}{s^3 T + s^2 + sK_C KT_1 + K_C K} \tag{9.13}$$

This avoids the zero and therefore is somewhat easier to design for a required dynamic response. One simple way is to make use of standard forms. The transfer function in normalised form is

$$T(s_n) = \frac{1}{s_n^3 + (\alpha^2 / KK_C)s_n + \alpha T_1 s_n + 1} \tag{9.14}$$

where the time scale factor $\alpha = (K_C K / T)^{1/3}$. Since there are only two variable controller parameters K_C and T_1 one can choose the values so that the denominator coefficients fit a standard form but in doing so the time scale factor, α, is fixed. Figure 9.3 shows the closed loop step responses obtained using this design procedure to achieve the performance indices J_n for n = 0 to 3.

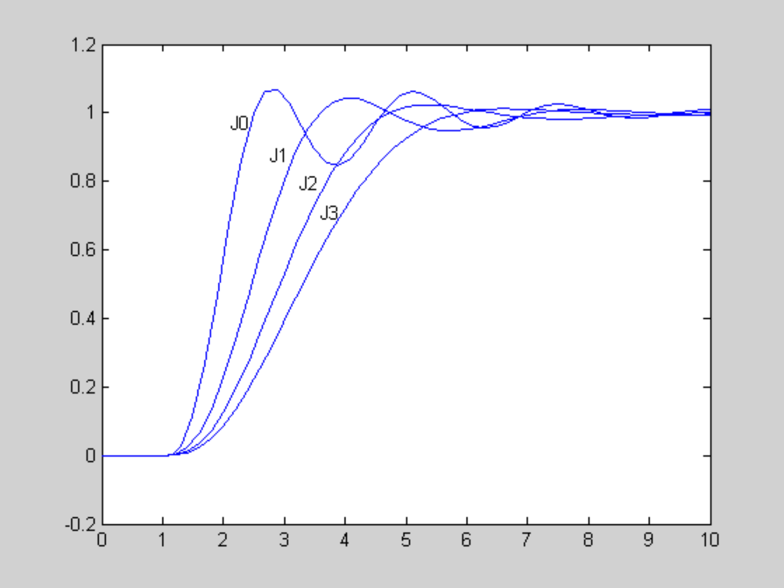

Figure 9.3 Response to a unit step at unit time for the standard form designs.

9.5 A Transfer Function with Complex Poles

As mentioned earlier plant transfer functions may involve complex poles which may be lightly damped. Designing of satisfactory series compensators for such systems is not easy so this is a problem which will be examined again in chapter 11 when state feedback compensation is discussed. To see the difficulties consider the transfer function

$$G(s) = \frac{0.1}{s(s^2 + 0.2s + 1)} \tag{9.15}$$

where the complex poles have a natural frequency of unity and a damping ratio of 0.1. The phase of the Bode plot changes rapidly near the resonant frequency of unity as seen in Figure 9.4 and with a unit gain controller the closed loop has a gain margin of 6dB and a phase margin of 89°. The step response, however, has oscillations on it due to the complex poles (see curve K_c = 1 in Figure 9.5).

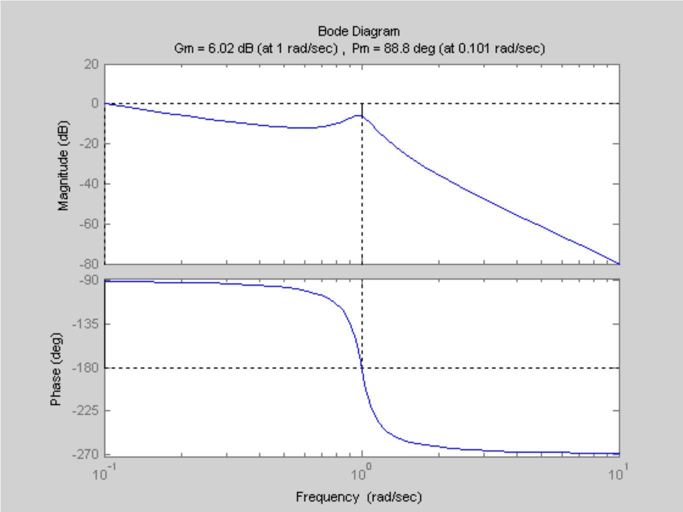

Figure 9.4 Bode diagram of transfer function of equation (9.15).

The poor step response is due to the relatively low gain margin, not low phase margin, and a simple phase lead design to speed up the response is not possible. A possible approach is to use a compensator with two zeros and two poles with the former being chosen to cancel the complex poles of the plant. Choosing

$$G(s) = \frac{K_c(s^2 + 0.2s + 1)}{s^2 + 2s + 1} \tag{9.16}$$

K_c can be chosen equal to 0.5 which gives a gain margin of 12dB, a phase margin of 44° and a step response with around 25% overshoot. The problem is if the parameters of the plant are not as assumed. Figure 9.5 shows step responses for a controller with gain 1 and 0.5, the slower responses, and with the transfer function of equation (9.16) with $K_c = 0.5$ for the three cases of the plant complex poles having the nominal damping ratio of 0.1 and also 0.15 and 0.05. The step response is thus hardly affected by an incorrect assumption for the damping ratio.

On the other hand the open loop Bode diagrams are shown in Figure 9.6 for the compensator plus the plant with resonant frequencies for the poles of 0.8 and 1.2, not the nominal value of unity. It can be seen that the gain margin for the resonant frequency of 0.8 is very small and therefore not surprisingly the step response is highly oscillatory as shown in Figure 9.7, together with that for a natural frequency of 1.2. It is thus seen from this example that the closed loop step response is very sensitive to an over estimation of the resonant frequency of the plant poles.

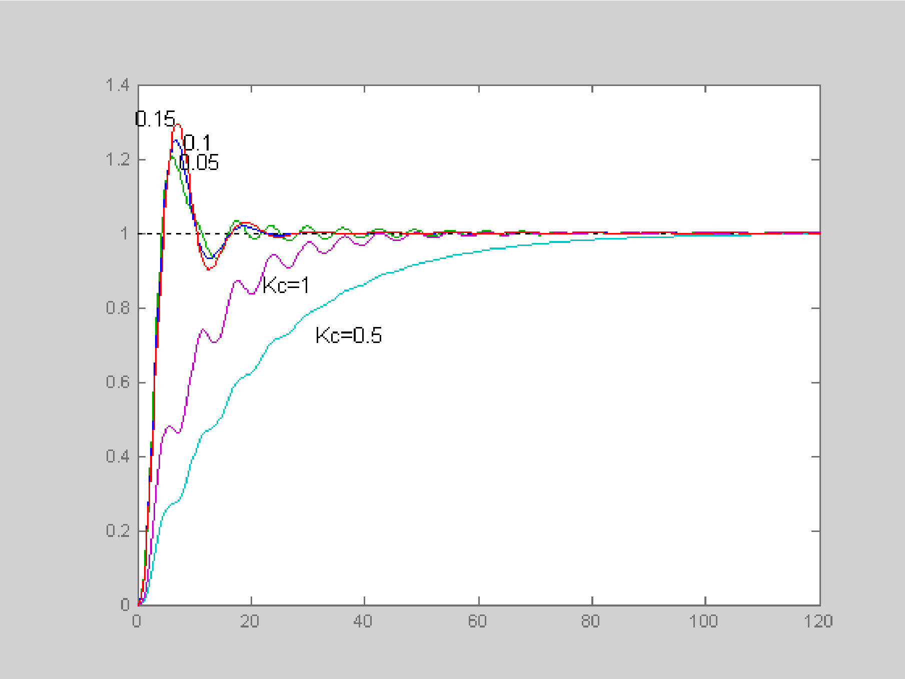

Figure 9.5 Step responses for different controllers and plant pole damping ratio.

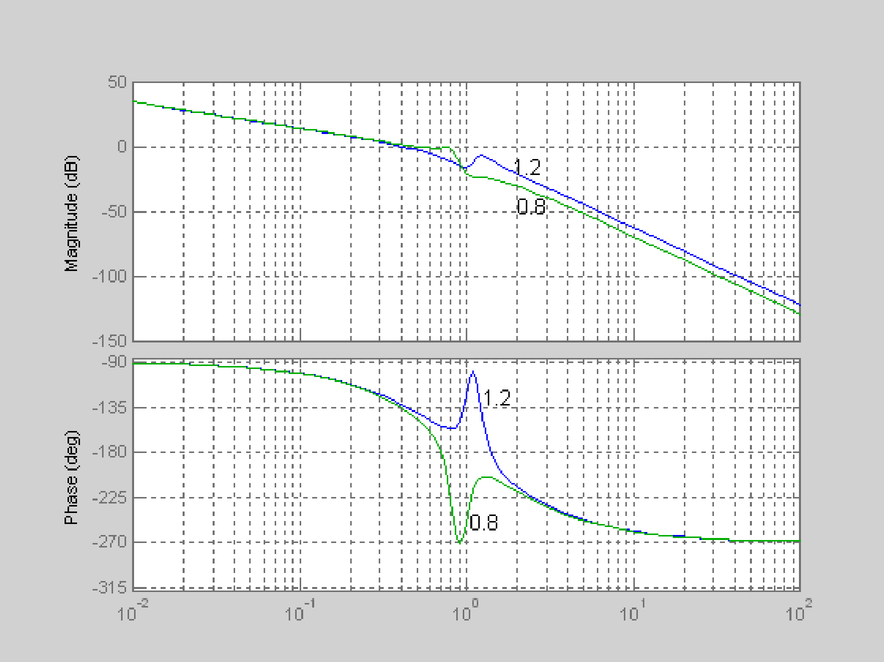

Figure 9.6 Bode diagrams for compensating controller plus plant for different natural frequencies of the plant poles.

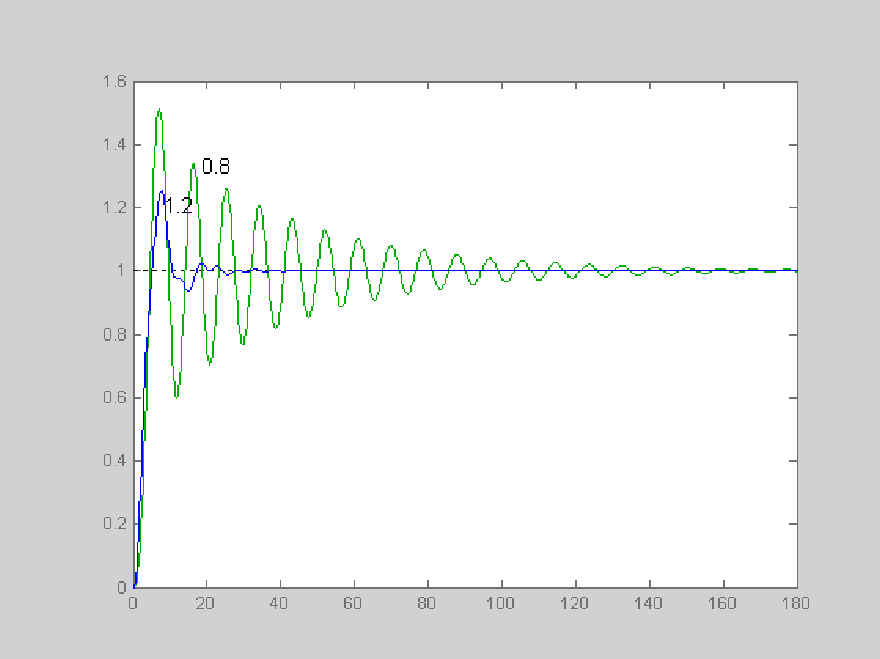

Figure 9.7 Step responses for closed loop with controller and plant for different natural frequencies of the plant poles.

9.6 The Effect of Parameter Variations

Most methods of controller design, as has been seen, require the use of a mathematical model for the process. In practice this model, which may be called the nominal model, is always an approximation of the real situation. Further the model may change dependent on environmental changes or with age. The effect of inaccuracies in a model on the system performance has therefore always been a concern of the design engineer. The comments in this section will assume that the form of the model is not in doubt but uncertainty exists in the estimates for some of its parameters. From a practical viewpoint today's simulation facilities are so fast that for the majority of situations after a design is completed multiple simulations can be done to assess the effect of changes in model parameters. However some theoretical results are available which will be commented on here. Much of the recent interest in this topic was started as a result of the work on the stability of interval polynomials by Kharitonov considered in the next section.

9.6.1 Stability of Interval Polynomials

Much recent work on systems with uncertain parameters has been based on Kharitonov's result [9.1] on the stability of interval polynomials. Kharitonov showed that for the interval polynomial

$$P(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4 + \cdots + a_n s^n \tag{9.17}$$

where $a_i \in [\underline{a_i}, \overline{a_i}]$, $i = 1,2,.....,n$, the stability of the set could be found by applying the Routh-Hurwitz criterion to only the following four polynomials

$$\begin{aligned}
p_1(s) &= \underline{a_0} + \underline{a_1}s + \overline{a_2}s^2 + \overline{a_3}s^3 + \underline{a_4}s^4 + \cdots\cdots \\
p_2(s) &= \underline{a_0} + \overline{a_1}s + \overline{a_2}s^2 + \underline{a_3}s^3 + \underline{a_4}s^4 + \cdots\cdot \\
p_3(s) &= \overline{a_0} + \underline{a_1}s + \underline{a_2}s^2 + \overline{a_3}s^3 + \overline{a_4}s^4 + \cdots\cdots \\
p_4(s) &= \overline{a_0} + \overline{a_1}s + \underline{a_2}s^2 + \underline{a_3}s^3 + \overline{a_4}s^4 + \cdots\cdots
\end{aligned} \tag{9.18}$$

Although this may seem a surprising result it is easily proved from the Mikhailov criterion of section 5.4.2. It can be easily shown that the value set of an interval polynomial at a fixed frequency is a rectangle (Kharitonov rectangle) as shown in Figure 9.8, that is the value of every polynomial of the family at that frequency lies within or on the rectangle, whose sides are parallel to the real and imaginary axes. Since the sides of the rectangular value set are parallel to the real and imaginary axes, it can easily be shown that the exclusion of the origin from the rectangular value set at all frequencies, which will be required for all the polynomials to satisfy the Mikhailov criterion, can be checked by using the corner points which correspond to the four Kharitonov polynomials. The Kharitonov theorem is only applicable to interval uncertain parameters but unfortunately the characteristic equations of even simple control systems do not normally have an interval uncertainty structure. For example, to take a simple case, consider a plant transfer function model of the form

$$G(s) = K / s(T_1 s + 1)(T_2 s + 1) \tag{9.19}$$

where uncertainty may exist in K, T_1 and T_2. If the plant is in the feedback loop of Figure 5.1 with $G_c = H = 1$ the characteristic equation for assessing stability is

$$\delta(s) = T_1 T_2 s^3 + (T_1 + T_2)s^2 + s + K = 0 \tag{9.20}$$

which is not an interval polynomial. The only simple way to use the Kharitonov result is

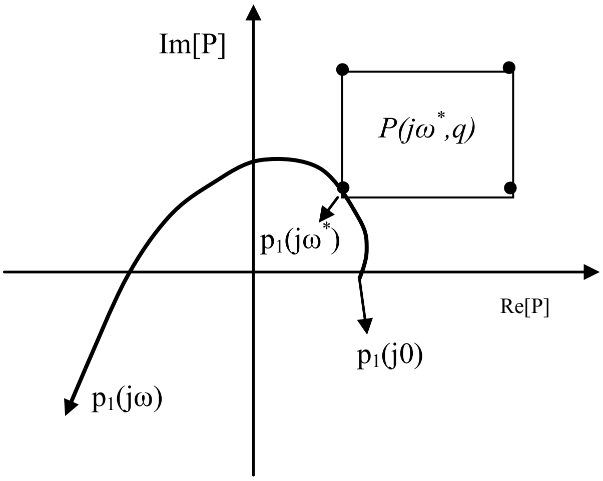

Fig.9.8 : Kharitonov box and the Mikhailov locus for $p_1(s)$

to overbound and underbound the parameters of s^3 and s^2, which produces a very conservative result. Assuming $T_i \in [\underline{T_i}, \overline{T_i}]$ the gain required to satisfy the four equations is $K < (\underline{T_1} + \underline{T_2}) / \overline{T}_1 \overline{T_2}$. For this specific example the direct application of the Routh-Hurwitz criterion gives the result $K < (T_1 + T_2)/T_1T_2 = (1/T_1) + (1/T_2)$, which obviously has a minimum value when T_1 and T_2 have their maximum values. For the specific case of the bounds $T_1 \in [1,2]$ and $T_2 \in [2,4]$ the exact result of the Routh-Hurwitz criterion is $K < 3/4$ whereas the Kharitonov result is $K < 3/8$, which is conservative by a factor of 2. The fact that one can obtain an exact solution from the Routh-Hurwitz criterion is because this is one of a few unique situations. It does, however, serve to show the conservativeness of results that can be expected from the Kharitonov criterion when applied to practical control situations, because of the parameter dependence typical of the terms in the closed loop characteristic equation.

9.6.2 Envelopes on Bode Plots

It is possible to obtain bounds on Bode plots for transfer functions with variable parameters [9.2]. To see this consider a general transfer function factorised into zero-pole form as given below

$$G(s) = \frac{\prod_{i=1}^{a}(s^2 + 2\varsigma_{ni}\omega_{ni}s + \omega_{ni}^2)/\omega_{ni}^2}{\prod_{j=1}^{b}(s^2 + 2\varsigma_{dj}\omega_{dj}s + \omega_{dj}^2)/\omega_{dj}^2} \cdot \frac{K\prod_{k=1}^{m}(1 + sL_k)}{s^N \prod_{l=1}^{n}(1 + sT_l)} e^{-\tau s} \tag{9.21}$$

where s^N in the denominator represents a pole of multiplicity N at the origin and $2a + m \le N + n + 2b$. It is assumed that the parameters K, L_k, ς_{ni}, ω_{ni}, ς_{dj}, ω_{dj}, T_l and τ are not known exactly but vary within the following intervals

$$K \in [\underline{K}, \overline{K}], \quad L_k \in [\underline{L_k}, \overline{L_k}], \quad \varsigma_{ni} \in [\underline{\varsigma_{ni}}, \overline{\varsigma_{ni}}], \quad \omega_{ni} \in [\underline{\omega_{ni}}, \overline{\omega_{ni}}], \quad \varsigma_{dj} \in [\underline{\varsigma_{dj}}, \overline{\varsigma_{dj}}], \tag{9.22}$$
$$\omega_{dj} \in [\underline{\omega_{dj}}, \overline{\omega_{dj}}], \; T_l \in [\underline{T_l}, \overline{T_l}] \text{ and } \tau \in [\underline{\tau}, \bar{\tau}]$$

The maximum gain and maximum phase lag at a particular frequency will be obtained from the product of the maximum gains and the sum of the maximum phase lags at that frequency of the individual elements, with a similar result for the minimum values. Therefore considering the individual elements in turn beginning with the time delay, since its gain is always unity, the maximum (minimum) phase lag is obtained with the maximum (minimum) value of τ. Also from sketches of Bode gain and phase diagrams for a single time constant transfer function it is obvious that the curves for $\underline{T}$ give the maximum gain and minimum phase shift and those for $\overline{T}$ give the minimum gain and maximum phase shift, respectively. The curves for all other values of T lie within the respective gain and phase boundaries of these plots. Finally the Bode gain diagram for the second order complex pole transfer function

$$G(s) = \frac{\omega_0^2}{s^2 + 2\varsigma\omega_0 s + \omega_0^2} \tag{9.23}$$

only has a peak in the response if ς is less than 0.707. Thus, if the Bode gain and phase diagrams are considered for this transfer function with $\varsigma \in [\underline{\varsigma}, \overline{\varsigma}]$ and $\omega_0 \in [\underline{\omega_0}, \overline{\omega_0}]$, and are drawn for the four cases of $\overline{\omega_0}$ with $\underline{\varsigma}$ and $\overline{\varsigma}$, and $\underline{\omega_0}$ with $\underline{\varsigma}$ and $\overline{\varsigma}$ it can easily be seen that:

1. The minimum magnitude if (a) $\bar{\varsigma} > 0.707$ is given for all ω by

$$\left|G(j\omega)\right| = \left|\frac{\underline{\omega_0^2}}{-\omega^2 + 2\bar{\varsigma}\underline{\omega_0}j\omega + \underline{\omega_0^2}}\right| \tag{9.24}$$

and if (b) $\bar{\varsigma} < 0.707$ then from $\omega = 0$ to $\omega_x = \sqrt{(2\overline{\omega_0^2}\,\underline{\omega_0^2}(1 - 2\overline{\varsigma^2})(\,\underline{\omega_0^2} + \overline{\omega_0^2})}$ it is given by

$$\left|G(j\omega)\right| = \left|\frac{\overline{\omega_0^2}}{-\omega^2 + 2\bar{\varsigma}\overline{\omega_0}j\omega + \overline{\omega_0^2}}\right| \tag{9.25}$$

and from ω_x to ∞ by

$$\left|G(j\omega)\right| = \left|\frac{\underline{\omega_0^2}}{-\omega^2 + 2\bar{\varsigma}\underline{\omega_0}j\omega + \underline{\omega_0^2}}\right| \tag{9.26}$$

2. The maximum magnitude if (a) $\underline{\varsigma} > 0.707$ is given for all ω by

$$\left|G(j\omega)\right| = \left|\frac{\overline{\omega_0^2}}{-\omega^2 + 2\underline{\varsigma}\overline{\omega_0}j\omega + \overline{\omega_0^2}}\right| \tag{9.27}$$

and if (b) $\underline{\varsigma} < 0.707$ then from $\omega = 0$ to $\omega_{p\,\min} = \underline{\omega_0}\sqrt{1 - 2\underline{\varsigma}^2}$ it is given by

$$\left|G(j\omega)\right| = \left|\frac{\underline{\omega_0^2}}{-\omega^2 + 2\underline{\varsigma}\underline{\omega_0}j\omega + \underline{\omega_0^2}}\right| \tag{9.28}$$

The maximum value of the gain at $\omega_{p\,\min}$ is

$$\frac{1}{2\underline{\varsigma}\sqrt{1 - 2\underline{\varsigma}^2}} \tag{9.29}$$

and the maximum possible gain remains constant at this value independent of ω until $\omega_{p\,\max} = \overline{\omega_0}\sqrt{1 - 2\underline{\varsigma}^2}$ and then for $\omega \in [\omega_{p\,\max}, \infty)$ it is given by

$$\left|G(j\omega)\right| = \left|\frac{\overline{\omega_0^2}}{-\omega^2 + 2\underline{\varsigma}\overline{\omega_0}j\omega + \overline{\omega_0^2}}\right| \tag{9.30}$$

3. The maximum phase for $\omega \in [0, \underline{\omega_0})$ is given by

$$\arg[G(j\omega)] = \arg\left[\frac{\underline{\omega_0^2}}{-\omega^2 + 2\bar{\varsigma}\underline{\omega_0}j\omega + \underline{\omega_0^2}}\right] \tag{9.31}$$

and for $\omega \in [\underline{\omega_0}, \infty)$ by

$$\arg[G(j\omega)] = \arg[\frac{\underline{\omega_0^2}}{-\omega^2 + 2\underline{\varsigma}\underline{\omega_0} j\omega + \underline{\omega_0^2}}] \tag{9.32}$$

4. The minimum phase for $\omega \in [0, \overline{\omega_0})$ is given by

$$\arg[G(j\omega)] = \arg[\frac{\overline{\omega_0^2}}{-\omega^2 + 2\underline{\varsigma}\overline{\omega_0} j\omega + \overline{\omega_0^2}}] \sim \tag{9.33}$$

and for $\omega \in [\overline{\omega_0}, \infty)$ by

$$\arg[G(j\omega)] = \arg[\frac{\overline{\omega_0^2}}{-\omega^2 + 2\overline{\varsigma}\overline{\omega_0} j\omega + \overline{\omega_0^2}}] \tag{9.34}$$

Unfortunately it is not possible to derive similar results for Nyquist plots. Some results have been obtained but they do not provide accurate bounds on the plots [9.3]. Also although accurate bounding of the Bode plots is obtained by the above approach results obtained using them are still conservative because the link between the gain plot and the phase plot of a specific transfer function is lost. To show this, consider the closed loop stability problem for the open loop transfer function of equation (9.19) with the same bounds on the time constants. The bounds of the Bode plots are shown in Figure 9.9 and to ensure stability one has to use the lower bound of the phase plot and the upper bound of the gain plot. This give a value for stability of $K < 0.46$, an improvement on the Kharitonov result but still very conservative.

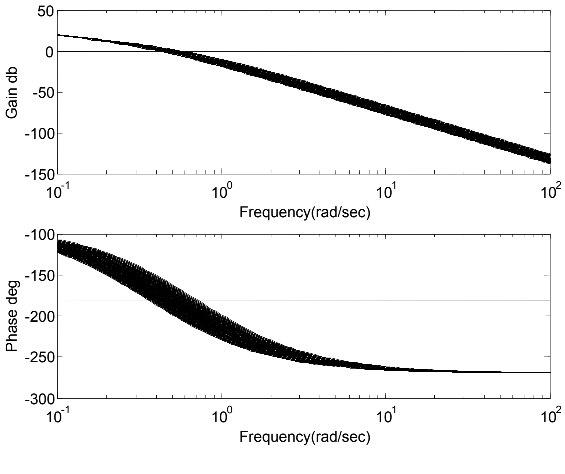

Figure 9.9 Stability from the Bounds of the Bode plots.

9.7 References

9.1 Kharitonov, V.L.: 'Asymptotic stability of an equilibrium position of a family of systems of linear differential equations,' *Differential Equations*, 1979, **14**, pp. 1483–1485.

9.2 Tan, N. and Atherton, D.P.: New Approach to Assessing the Effects of Parametric Variations in Feedback Loops. *IEE Proceedings Control Theory and Applications*, Vol 150, No. 2 March 2003, pp. 101–111.

9.3 Hollot, C.V. and Tempo, R.: 'On the Nyquist envelope of an interval plant family,' *IEEE Trans. Automat. Contr.,* 1994, **39**, (2), pp. 391–396.

10 State Space Methods

10.1 Introduction

State space modelling was briefly introduced in chapter 2. Here more coverage is provided of state space methods before some of their uses in control system design are covered in the next chapter. A state space model, or representation, as given in equation (2.26), is denoted by the two equations

$$\dot{x} = Ax + Bu \tag{10.1}$$

$$y = Cx + Du \tag{10.2}$$

where equations (10.1) and (10.2) are respectively the state equation and output equation.

The representation can be used for both single-input single-output systems (SISO) and multiple-input multiple-output systems (MIMO). For the MIMO representation A, B, C and D will all be matrices. If the state dimension is n and there are r inputs and m outputs then A, B, C and D will be matrices of order, $n \times n$, $n \times r$, $m \times n$ and $m \times r$, respectively. For SISO systems B will be an $n \times 1$ column vector, often denoted by $\mathbf{b}$, C a $1 \times n$ row vector, often denoted by $\mathbf{c}^{\mathrm{T}}$, and D a scalar often denoted by d. Here the capital letter notation will be used, even though only SISO systems are considered, and B, C, and D will have the aforementioned dimensions. As mentioned in chapter 2 the choice of states is not unique and this will be considered further in section 10.3. First, however, obtaining a solution of the state equation is discussed in the next section.

10.2 Solution of the State Equation

Obtaining the time domain solution to the state equation is analogous to the classical approach used to solve the simple first order equation

$$\dot{x} = ax + u \tag{10.3}$$

The procedure in this case is to take $u = 0$, initially, and to assume a solution for $x(t)$ of $e^{\mathrm{a}t}x(0)$ where $x(0)$ is the initial value of $x(t)$. Differentiating this expression gives

$\dot{x}(t) = ae^{at}x(0) = ax(t)$ so that the assumed solution is valid. Now if the input u is considered this is assumed to yield a solution of the form $x(t) = e^{\mathrm{a}t}f(t)$, which on differentiating gives

$\dot{x}(t) = ae^{at}f(t) + e^{at}\dot{f}(t)$. Thus the differential equation is satisfied if

$ae^{at}f(t)+e^{at}\dot{f}(t)=ae^{at}f(t)+u(t)$, giving $\dot{f}(t)=[e^{at}]^{-1}u(t)$, which has the solution $f(t)=\int_0^t[e^{a\tau}]^{-1}u(\tau)d\tau$, giving $x(t)=e^{at}\int_0^t[e^{a\tau}]^{-1}u(\tau)d\tau$, where τ is a dummy variable. This solution can be written $x(t)=\int_0^t e^{a(t-\tau)}u(\tau)d\tau$ so that the complete solution for $x(t)$ consists of the sum of the two solutions, known as the complimentary function (or initial condition response) and particular integral (or forced response), respectively and is

$$x(t)=e^{at}x(0)+\int_0^t e^{a(t-\tau)}u(\tau)d\tau \tag{10.4}$$

For equation (10.1) x is an n vector and A an $n \times n$ matrix not a scalar a and to obtain the complimentary function one assumes $x(t)=e^{At}x(0)$. e^{At} is now a function of a matrix, which is defined by an infinite power series in exactly the same way as the scalar expression, so that

$$e^{At}=I+At+At^2/2!+A^3t^3/3!+.... \tag{10.5}$$

where I is the $n \times n$ identity matrix. Term by term differentiation of equation (10.5) shows that the derivative of e^{At} is Ae^{At} and that $x(t)=e^{At}x(0)$ satisfies the differential equation with $u = 0$. e^{At} is often denoted by $\varphi(t)$ and is known as the state transition matrix. Using the same approach as for the scalar case to get the forced response the total solution is found to be

$$x(t)=\varphi(t)x(0)+\varphi(t)\int_0^t\varphi^{-1}(\tau)Bu(\tau)d\tau \tag{10.6}$$

It is easily shown that the state transition matrix $\varphi(\tau)=e^{A\tau}$ has the property that $\varphi(t-\tau)=\varphi(t)\varphi^{-1}(\tau)$ so that equation (10.6) can be written alternatively as

$$x(t)=\varphi(t)x(0)+\int_0^t\varphi(t-\tau)Bu(\tau)d\tau \tag{10.7}$$

This time domain solution of equation (10.1) is useful but most engineers prefer to make use of the Laplace transform approach. Taking the Laplace transform of equation (10.1) gives

$$sX(s)-x(0)=AX(s)+BU(s) \tag{10.8}$$

which on rearranging as $X(s)$ is an n vector and A a $n \times n$ matrix gives

$$X(s)=(sI-A)^{-1}x(0)+(sI-A)^{-1}BU(s) \tag{10.9}$$

Taking the inverse Laplace transform of this and comparing with equation (10.7) indicates that

$$\mathcal{L}\,[\varphi(t)]=\Phi(s)=(sI-A)^{-1} \tag{10.10}$$

Also taking the Laplace transform of the output equation (10.2) and substituting for $X(s)$ gives

$$Y(s) = C(sI - A)^{-1}x(0) + [C(sI - A)^{-1}B + D]U(s) \tag{10.11}$$

so that the transfer function, $G(s)$, between the input u and output y is

$$Y(s)/U(s) = G(s) = C(sI - A)^{-1}B + D = C\Phi(s)B + D \tag{10.12}$$

This will, of course, be the same independent of the choice of the states.

10.3 A State Transformation

Obviously there must be an algebraic relationship between different possible choices of state variables. Let this relationship be

$$x = Tz \tag{10.13}$$

where x is the original choice in equations (10.1) and (10.2) and z is the new choice. Substituting this relationship in equation (10.2) gives $T\dot{z} = ATz + Bu$ which can be written

$$\dot{z} = T^{-1}ATz + T^{-1}Bu \tag{10.14}$$

Also substituting in the output equation (10.2) gives

$$y = CTz + Du \tag{10.15}$$

Thus under the state transformation of equation (10.13) a different state space representation (T^{-1} *AT,* $T^{-1}B$*, CT, D*) is obtained. If the new *A* matrix is denoted by A_z = T^{-1} *AT* then it is easy to show that A and A_z have the following properties

1) The same eigenvalues
2) The same determinant
3) The same trace (Sum of elements on the main diagonal)

There are some specific forms of the *A* matrix which are often commonly used in control engineering and not unsurprisingly these relate to how one might consider obtaining a state space representation for a transfer function, the topic of the next section.

10.4 State Representations of Transfer Functions

This topic was introduced in section 2.3 where the controllable canonical form for a differential equation was considered. Here this and some other forms will be considered by making use of block diagrams where every state will be an integrator output. To develop some representations consider the transfer function

$$\frac{Y(s)}{U(s)} = G(s) = \frac{s^2 + 3s + 4}{s^3 + 7s^2 + 14s + 8} \tag{10.16}$$

10.4.1 Controllable Canonical Form.

As seen from equation (2.20) the first n-1 state variables are integrals of the next state, that is $x_{(j-1)} = \int x_j dx$, or as shown in the equation by $\dot{x}_{(j-1)} = x_j$, for j = 2 to n. Thus the block diagram to represent this is *n* integrators in series. The input to the first integrator is $\dot{x}_n$ and its value is given by $\dot{x}_n = -a_0x_1 - a_1x_2 - a_2x_3 ..... + u$, the last row of the matrix representation of equation (2.20). The numerator terms are provided by feeding forward from the states to give the required output. Thus, for our simple example, this can be shown in the block diagram of Figure 10.1, done in SIMULINK, where since the transfer function is third order *n* = 3, there are three integrators, blocks with transfer functions 1/*s*, in series. Feedback from the states, where the integrator outputs from left to right are the states x_3, x_2, and x_1, respectively, is by the coefficients -8, -14 and -7. (negative and in the reverse order of the transfer function denominator). The numerator coefficients provide feedforward from the states, with the s^2 term from x_3.

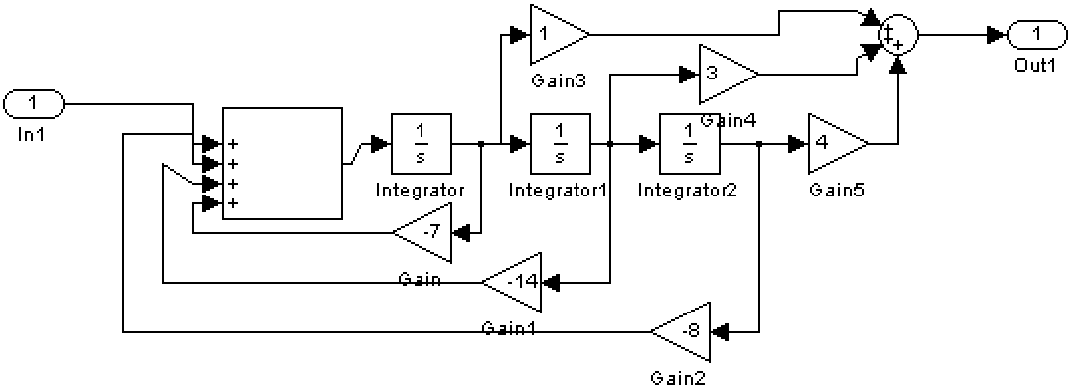

Figure 10.1 Controllable Canonical Form Diagram for the Example.

The matrices for the state representation are

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -14 & -7 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } C = \begin{pmatrix} 4 & 3 & 1 \end{pmatrix}.$$

MATLAB has a companion form, which for any state space system **G=ss(A,B,C,D)**, will be returned on typing **sys=canon(G,'companion')**. The companion representation sys will have an *A* matrix which is the transpose of the above *A* matrix and a $B = (1\ 0\ 0)^T$.

10.4.2 Observable Canonical Form

The observable canonical form is related to the controllable form by the following relationships, $A_o = A_c^T$, $B_o = C_c^T$ and $C_o = B_c^T$. The subscripts c and o relate to the controllable and observable form matrices respectively and ^{T} denotes the transpose. It is left to the interested reader to develop a block diagram similar to Figure 10.1 for this form.

10.4.3 Diagonal (or Modal) Form.

If the impulse response of *G*(*s*) is required then its evaluation by inverse Laplace transforms requires a partial fraction expansion of *G*(*s*).

This is $G(s) = \dfrac{2/3}{s+1} - \dfrac{1}{s+2} + \dfrac{4/3}{s+4}$,

which is simply a parallel connection of three first order transfer functions. The first order transfer function *K*/(s + *a*) can be modelled with one integrator as shown in Figure 10.2. If the output of the integrator is denoted by the state variable x_1 then its state and output equations are $\dot{x}_1 = -ax_1 + bu$, $y = cx_1$.

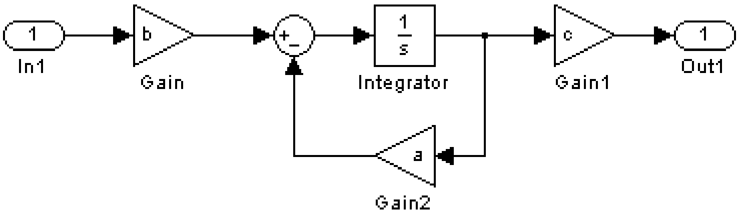

Figure 10.2 Diagram of Model for One State Variable.

Note that there is no unique value for *b* and *c* as all that is required is that their product should equal *K*. Thus a state representation for *G*(*s*) has

$$A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -4 \end{pmatrix}, \ B = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \text{ and } C = \begin{pmatrix} 2/3 & -1 & 4/3 \end{pmatrix}$$

where we have chosen to take all the B values as unity. This form of the A matrix is known as a diagonal form and can always be found if the denominator of the transfer function has real roots. To keep the matrix real for complex roots, say $\sigma \pm j\omega$, the corresponding rows around the diagonal are replaced by the matrix $\begin{pmatrix} \sigma & \omega \\ -\omega & \sigma \end{pmatrix}$. For example if $G(s) = \frac{1}{s(s^2+s+1)}$, the controllable canonical form, with the matrices subscripted with c, is

$$A_c = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix},\ B_c = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } C_c = (1 \quad 0 \quad 0).$$

A diagonal form is

$$A_\Lambda = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -0.5 & 0.866 \\ 0 & -0.866 & -0.5 \end{pmatrix},\ B_\Lambda = \begin{pmatrix} 1 \\ -1.732 \\ -1 \end{pmatrix} \text{ and } C_\Lambda = \begin{pmatrix} 1 & 0.5774 & 0 \end{pmatrix}$$

This is the one given by MATLAB if the instruction **csys=canon(G,'modal')** is used, where csys is the new state space representation in the chosen canonical form 'modal'. A special case is when $G(s)$ has a repeated root, for example if $G(s) = \frac{1}{s(s+1)^2}$, which has a state space representation of $A_J = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$, $B_J = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ and $C_J = (1 \ \ -1 \ \ -1)$. This can be seen from the partial fraction expansion of $G(s)$, which is $G(s) = \frac{1}{s} - \frac{1}{s+1} - \frac{1}{(s+1)^2}$. The numerator coefficients are in C, the three roots 0, -1, -1 remain on the diagonal but the off-diagonal unit term in A and the zero in B are due to the fact that the last term of the partial fraction expansion has as input the output from the second state, not the input u. This form of A matrix is known as a Jordan form and due to the numerical methods used cannot be found with MATLAB.

10.4.4 Guillemin Form

Another simple way of obtaining a state space representation of a transfer function is to make repeated use of the state representation of Figure 10.3 for the one-zero one-pole transfer function $(ds + c)/(s + a)$ which has a state space representation $(a,1,e\text{-}ad,d)$ for the single state x.

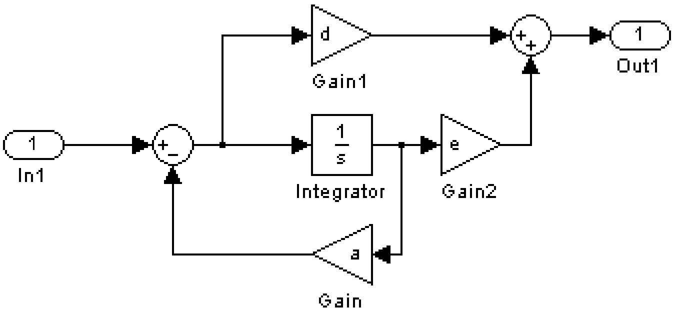

Figure 10.3 Diagram for One State Variable for Guillemin Form.

Thus for a transfer function with real poles and zeros in factored form given by $G(s) = (s + 1)\ (s + 2)/s\ (s + 3)\ (s + 5)$ one can split it into one of several possible series (cascade) combinations such as $(\frac{1}{s})(\frac{s+1}{s+3})(\frac{s+2}{s+5})$ and then use the representation of Figure 10.3 for each transfer function to constitute the overall model as in Figure 10.4, which has, assuming the outputs of the integrators from left to right are x_3, x_2 and x_1, respectively, the equations

$\dot{x}_3 = u \quad \dot{x}_2 = -3x_2 + x_3 \quad \dot{x}_1 = -5x_1 + \dot{x}_2 + x_2$ and $y = 2x_1 + \dot{x}_1$. Substituting appropriately for the derivatives gives the state representation

$$A = \begin{pmatrix} -5 & -2 & 1 \\ 0 & -3 & 1 \\ 0 & 0 & 0 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } C = (-3 \quad -2 \quad 1)$$

where the A matrix is upper triangular.

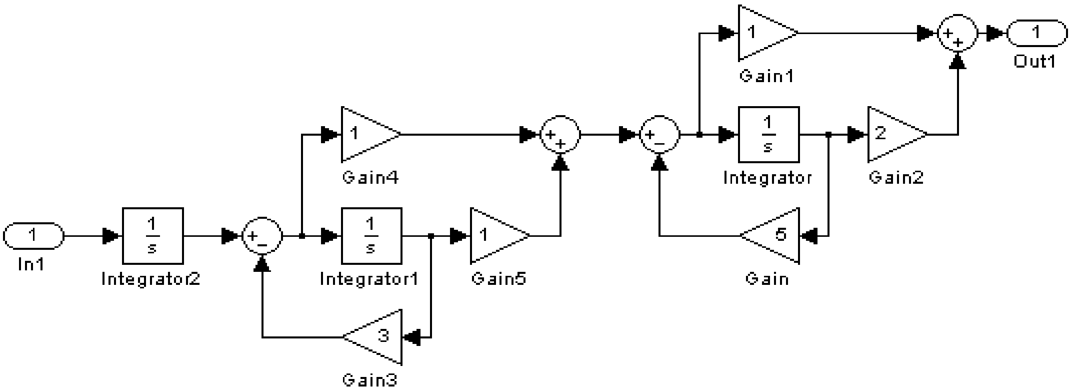

Figure 10.4. Diagram for Guillemin Form State Representation.

10.5 State Transformations between Different Forms

Given a state space representation then one can evaluate the corresponding transfer function and use this to obtain a different state space representation. In some cases, however, it is more convenient if one can obtain the specific state transformation, *T*, discussed in section 10.3, that will do this directly. It can be shown that any *A* matrix can be transformed to a diagonal form by its own eigenvector matrix. Eigenvectors only define directions, however, so that such a matrix is not unique with a scalar multiplier being allowed on any column vector t_i of *T*. The eigenvector, t_{i}, corresponding to a particular eigenvalue, s_{i}, of a matrix *A* is found from $(s_i I - A)t_i = 0$ for *i* = 1…*n*. For example, the A matrix $A = \begin{pmatrix} 2 & 1 \\ -12 & -5 \end{pmatrix}$ has a characteristic equation of $(s-2)(s+5)+12=0$, giving $s^2+3s+2=(s+2)(s+1)=0$, so has eigenvalues of -1 and -2. The corresponding eigenvectors are obtained from $(-I-A)t_1=0$ and $(-2I-A)t_2=0$, yielding $t_1=\begin{pmatrix} a & -3a \end{pmatrix}^T$ and $t_2=\begin{pmatrix} b & -4b \end{pmatrix}^T$ where *a* and *b* are any constants. Thus taking $T=\begin{pmatrix} a & b \\ -3a & -4b \end{pmatrix}$ then the transformation $T^{-1}AT$ will yield the diagonal matrix $A_\Lambda = \begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$ whatever the choice of *a* and *b*. However, if the transformation was applied to a state space representation (*A,B,C,D*) the resulting (A_Λ,B_Λ,C_Λ,*D*), would have different results for B_Λ and C_Λ dependent on the choice of *a* and *b*. When the given *A* matrix is in controllable canonical form then it can be shown that the column eigenvectors t_{i} of *T* are given by $t_i = \begin{pmatrix} 1 & s_i & s_i^2 & .... & s_i^{(n-1)} \end{pmatrix}^T$, which is known as a Vandermonde matrix.

10.5.1 Transforming to Controllable Canonical Form

If it is required to find the controllable canonical form of a state space representation (*A*,*B*,*C*,*D*) then this can be achieved by a unique transformation as not only is the A_c matrix of a specific form but so also is the vector B_c. From the two equations

$$T^{-1}AT = A_c \text{ and } TB_c = B$$

it can be shown that the column vectors t_i of *T* are given, for *i* = 1…*n*, by

$$t_n = B\,,\ At_1 = -a_0 t_n\,,\ At_2 = t_1 - a_1 t_n\,,\ At_3 = t_2 - a_2 t_n\,, \text{ etc.}$$

Here the $a_{i,}$ *i* = 0…..(*n*-1), are the coefficients of the characteristic equation of *A*, which of course form the last row of A_c. Some algebraic manipulations on these equations show that the transformation matrix *T* can be written as

$$T = \begin{pmatrix} B & AB & A^2B & \dots. & A^{n-1}B \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \dots & a_{n-1} & 1 \\ a_2 & \dots.. & a_{n-1} & 1 & 0 \\ : & : & : & : & : \\ a_{n-1} & 1 & 0 & . & 0 \\ 1 & 0 & \dots & . & 0 \end{pmatrix} \tag{10.17}$$

10.6 Evaluation of the State Transition Matrix

There are several ways to evaluate the state transition matrix $\varphi(t) = e^{At}$ and some of these are outlined below.

10.6.1 Direct Expansion

This is tedious and involves calculating powers of *A*, substituting them in equation (10.5) and finding the exponential series which give each term in the summed matrix expression.

10.6.2 The Inverse Laplace Transform

This involves finding $\varphi(t)$ from the inverse of equation (10.10), that is $\varphi(t) = \mathcal{L}^{-1}(sI-A)^{-1}$. This is straightforward but very laborious for calculating the required matrix inversion, except for low order matrices, *A*. One then has to find the inverse Laplace transforms of the individual matrix terms which are functions of *s*.

10.6.3 Use of a Diagonal Transformation

If the matrix T is a transformation which diagonalises the A matrix to Λ then it can be shown that $\varphi(t) = e^{At} = Te^{\Lambda t}T^{-1}$. Thus, once T and its inverse have been found this approach requires evaluation of the product of three n x n matrices.

10.6.4 Use of the Cayley Hamilton Theorem

This theorem states that a matrix satisfies its own characteristic equation. Thus, if the matrix A has a characteristic equation $\Delta(s) = s^n + a_{n-1}s^{n-1} + ...a_1 s + a_0 = 0$, then $\Delta(A) = A^n + a_{n-1}A^{n-1} + ...a_1 A + a_0 = 0$. This means that A^n can be calculated in lower powers of A and that any infinite series of $A, f(A) = c_0 I + c_1 A + c_2 A^2 + c_3 A^3 ....to.\infty$, can be expressed as $f(A) = \gamma_0 I + \gamma_1 A + \gamma_2 A^2 + \alpha_3 A^3 ....\gamma_{n-1}A^{n-1}$ for a matrix of order n. Further all eigenvalues of A must also satisfy this equation. Thus when the function of the matrix A of order n is the exponential its eigenvalues and the matrix must satisfy

$$e^{s_i t} = \sum_{k=0}^{n-1} \alpha_k s_i^k \quad \text{and} \quad e^{At} = \sum_{k=0}^{n-1} \alpha_k A^k \tag{10.18}$$

The first equation when used for all the eigenvalues provides n equations which can be solved for the n coefficients α. Substituting these in the second equation enables the state transition matrix $\varphi(t) = e^{At}$ to be found.

10.7 Controllability and Observability

Consider a state space representation (A,B,C,0) with

$$A = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix}, \; B = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{ and } C = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$$

Then, in block diagram form this consists of the four modes at -1, -2, -3, -4 which are connected respectively to the input and output, output only, input only and to neither input or output. The transfer function from input to output is simply 1/(s+1) as the -1 mode is the only one connected to both the input and output. Since the -1 mode is connected to both input and output it is said to be both controllable and observable. The -2 mode is said to be uncontrollable and observable being connected to the output only; the -3 mode controllable and unobservable being connected to the input only; and the -4 mode is said to be uncontrollable and unobservable being connected to neither the input or the output. Given a state space description it is desirable, as will be seen in the next chapter, to know which modes are in the different situations exemplified by the four above modes. A system is said to be controllable if all the states are controllable, and observable if all the states are observable. The formal definitions are given below. From the above example it is clear that only those modes which are controllable and observable appear in the transfer function between input and output. Thus, if a system with an $n \times n$ A matrix is controllable and observable the denominator of its transfer function will be of order n (i.e. it will have n poles).

10.7.1 Controllability

A system is controllable if there exists an input *u* which transfers the initial state $x(0)$ to the zero state $x(t) = 0$ in a finite time *t*. Given any SISO system, A $(n \times n)$ and B $(n \times 1)$ matrices then it can be shown that the system will be controllable if the $(n \times n)$ controllability matrix $\mathrm{X} = (\mathrm{B}\ AB\ A^2B\ ....\ A^{n-1}B)$ has rank *n*. It will be noticed that this matrix is the first part of the transformation matrix for *T* in equation (10.17) and, as a consequence, a system can only be put into controllable canonical form if it is controllable. Or, alternatively, a system which has a controllable canonical form state space representation is controllable.

10.7.2 Observability

A system is observable if the initial state $x(0)$ can be uniquely determined by observing the output over a finite time *t*. Given any SISO system, A $(n \times n)$ and C $(1 \times n)$ matrices then it can be shown that the system will be observable if the $(n \times n)$ observability matrix $\mathrm{O} = \begin{pmatrix} C \\ CA \\ CA^2 \\ :: \\ CA^{n-1} \end{pmatrix}$ has rank *n*.

Again it can be shown that a system can only be put into observable form if it is observable.

10.8 Cascade Connection

In previous chapters on control system design significant attention has been given to cascade compensation and the effect on the open loop frequency response locus of adding a compensator. If the compensator and plant are given in state space form then it may be desirable to obtain a state space representation for their cascade combination. Thus, let the compensator $G_c(s)$ with state *z*, input *e*, and output *u* have the state space representation $(\mathrm{A}_1,B_1,C_1,\mathrm{D}_1)$ and the plant $G(s)$ with state *x*, have input *u*, and output *c* have the state space representation (A_2,B_2,C_2,D_2), then

$$\dot{z} = A_1 z + B_1 e\,,\quad u = C_1 z + D_1 e$$

and $\dot{x} = A_2 x + B_2 u$, $c = C_2 x + D_2 u$

Writing a combined state vector $(\mathrm{z},\ \mathrm{x})^{\mathrm{T}}$ one can write $\begin{pmatrix} \dot{z} \\ \dot{x} \end{pmatrix} = \begin{pmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{pmatrix}\begin{pmatrix} z \\ x \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix} e$ and $c = \begin{pmatrix} D_2C_1 & C_2 \end{pmatrix}\begin{pmatrix} z \\ x \end{pmatrix} + D_2D_1 e$ which gives a state space representation (A,B,C,D) with

$$A = \begin{pmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{pmatrix},\ B = \begin{pmatrix} B_1 \\ B_2D_1 \end{pmatrix},\ C = \begin{pmatrix} D_2C_1 & C_2 \end{pmatrix} \text{ and } D = D_2D_1. \qquad (10.19)$$

11 Some State Space Design Methods

11.1 Introduction

The previous chapters on controller design have mainly concentrated on introducing the compensator in the forward path, but use of a simple compensator in the feedback path has been discussed. Also feedback compensation has been mentioned with respect to the PI-PD controller and velocity feedback in a position control system. Both these two cases can be regarded as feedback of two states, namely, the output to form the error and the derivative of the output. It is therefore appropriate to look in general at how the performance of a control system can be changed by the feedback of state variables. If this is to be done in practice then the state variables have to be available either as measured values or estimates. Obtaining measurements can be costly because of the requirement for additional sensors so in many cases the variables are estimated using estimation methods. This is a topic outside the scope of this book but it will suffice to say that estimation methods have become relatively easy to implement with the use of modern technologies employing microprocessors with significant software included to do the required computations. In the next section results are derived for full state variable feedback and this is followed by a discussion of the linear quadratic regulator problem. The problem of direct closed loop transfer function synthesis, or standard forms, is looked at again in terms of using state variable feedback to achieve such a design. Finally as an example of the benefits of using a state variable feedback design the problem of controlling a plant having a transfer function with lightly damped complex poles, considered initially in section 9.5 is reconsidered.

11.2 State Variable Feedback

Consider a SISO system, *G*, with a state space representation (A,B,C,0). Assume state feedback is used so that the system input $u = K_c(v - k^T x)$, as shown in Figure 11.1. Here the row vector k^T, is given by $k^T = (k_1 \quad k_2 \quad k_3 \quad . \quad . \quad . \quad . \quad k_n .)$, which means that the signal fed back and subtracted from v is $k_1 x_1 + k_2 x_2 ......k_n x_n$. The thick line is used to show that it represents more than one signal, in this case the state x which has n components.

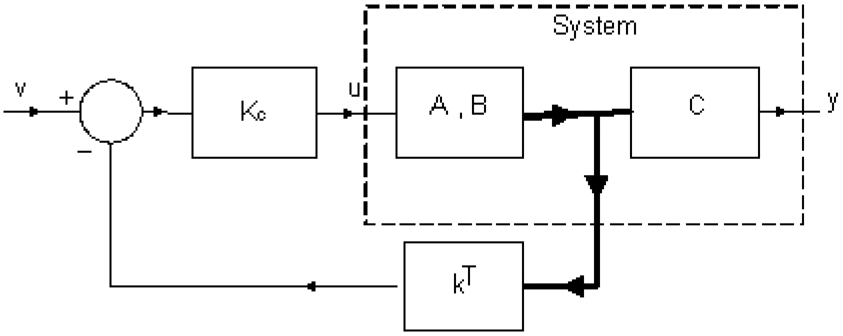

Figure 11.1 Block diagram of state feedback

The new system, with input *v*, is

$$\dot{x} = Ax - BK_c k^T x + BK_c v \tag{11.1}$$

which can be written

$$\dot{x} = A_f x + B_f v \tag{11.2}$$

where the matrices

$$A_f = A - BK_c k^T \text{ and } B_f = BK_c \tag{11.3}$$

Now suppose the original system was in controllable canonical form so that

$$A = A_c = \begin{pmatrix} 0 & 1 & 0 & . & . & . & . & 0 \\ 0 & 0 & 1 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 1 & 0 & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & 0 & 1 & 0 \\ . & . & . & . & . & . & . & 1 \\ -a_0 & -a_1 & -a_2 & . & . & . & . & -a_{n-1} \end{pmatrix} \quad B = B_c = \begin{pmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ 1 \end{pmatrix} \tag{11.4}$$

then

$$A_f = \begin{pmatrix} 0 & 1 & 0 & . & . & . & . & 0 \\ 0 & 0 & 1 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 1 & 0 & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & 0 & 1 & 0 \\ . & . & . & . & . & . & . & 1 \\ -a_0 - K_c k_{c1} & -a_1 - K_c k_{c2} & -a_2 - K_c k_{c3} & . & . & . & . & -a_{n-1} - K_c k_{cn} \end{pmatrix} \tag{11.5}$$

as the matrix $BK_c k_c^T$ is all zeros apart from the last row. The gain vector has been subscripted by *c* to denote that it has state inputs from the controllable canonical form.

Thus the characteristic equation of the system with state feedback is

$$s^n + (a_{n-1} + K_c k_{cn})s^{n-1} + (a_{n-2} + K_c k_{c(n-1)})s^{n-2} + \ldots\ldots..(a_1 + K_c k_{c2})s + a_0 + K_c k_{c1} = 0 \tag{11.6}$$

and in principle the poles can be placed anywhere by choice of the components of k_c^T . Larger values of the components of k_c^T will speed up the system response but in practice this will not be possible due to physical limitations on the magnitudes of signals for linear operation. The gain K_c is basically redundant, however, it is useful to include it as the structure might, as is clear from Figure 11.1, be a resultant closed loop system with K_c the controller gain. In this case the controller input will include the error and for this to be the case when the state x_1 is the output, k_1 will be equal to one. If the system is not in controllable canonical form then the coefficient terms in the characteristic equation will not each involve a single feedback gain. This means that simultaneous equations need to be solved to find the required feedback gains to give a specific characteristic equation. One way to avoid this is to transform the original system to controllable canonical form, determine the required feedback gains for this representation and then transform these gains back to the required feedback values from the original states. The system must be controllable to do this transformation and it can be shown that this is a required condition to be able to place the poles in desired locations. Thus, if the calculated state feedback gain vector is k_c^T from the controllable form states x_c and the transformation from the original states x is $x = Tx_c$ then the required vector k^T for the original states, x, is obtained from the relationship $k^T = k_c^T T^{-1}$. Several algorithms are available in MATLAB which calculate the required feedback gain vector k^T for a given system (A,B) to give specified pole locations.

The feedback signal $k^T x$ can be written in transfer function terms as $k^T\Phi(s)BU$ and the output $Y(s) = C\Phi(s)BU$ so that in terms of the classical block diagram of Figure 5.1 the state feedback is equivalent to a feedback transfer function of $H(s) = \dfrac{k^T\Phi(s)}{C\Phi(s)}$.

11.3 Linear Quadratic Regulator Problem

It can be shown [11.1] for a state space representation with matrix A and column vector B that if a performance index

$$J = \int_0^\infty [x^T(t)Qx(t) + Ru^2(t)]dt \tag{11.7}$$

is to be minimised then the required control signal, $u(t)$, is given $u(t) = -k^T x(t)$, a linear function of the state variables. Further the value of the feedback gain vector is given by $k^T = R^{-1}B^T P$ where P is the unique positive definite symmetrical matrix solution of the algebraic Riccati equation (ARE)

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{11.8}$$

Obviously the solution for k^T depends upon the choice of the positive scalar, *R*, and the matrix *Q* which must be at least semi-positive definite. Although this is a very elegant theoretical result, the solution depends on the choice of the weighting matrix *Q* and scalar *R*. No simple method is available for choosing their values so that the closed loop performance meets a given specification. A further point is that whatever values are chosen then the open loop frequency response will avoid the circle of unit radius centred at (-1,0) on the Nyquist plot [11.2]. This means a phase margin of at least 90° for a typical control system open loop transfer function, which makes the design very conservative. The command **[x,l,g] = care(A,B,c$^{\mathbf{T}}$*c,R)** in MATLAB will return the solution *P* for the ARE in *x*, where the vector *c* defines the matrix *Q* by $Q = c^T * c$.

11.4 State Variable Feedback for Standard Forms

To show how state variable feedback can be used to achieve a standard form step response design consider a fourth order all-pole system transfer function *G(s)* with one integrator given in phase variable canonical state space form with

$$A_c = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -a_1 & -a_2 & -a_3 & -a_4 \end{pmatrix}, \; B_c = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } C_c = (K_p\ 0\ 0\ 0\ 0) \tag{11.9}$$

Using state variable feedback the new state space description has a state variable representation with

$$A_f = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -k_{c1}K_c & -a_1 - k_{c2}K_c & * & * & -a_4 - k_{c5}K_c \end{pmatrix}, \; B_f = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

and $C_f = (K_p \quad 0 \quad 0 \quad 0 \quad 0)$. (11.10)

The corresponding closed loop transfer function is

$$\frac{K_p K_c}{s^5 + (a_4 + k_{c5}K_c)s^4 + ... + (a_1 + k_{c2}K_c)s + k_{c1}K_c} \tag{11.11}$$

Dividing by $K_p K_c$ and setting $k_{c1} = K_p$, gives the standard form

$$\frac{1}{s_n^5 + d_4 s_n^4 + d_3 s_n^3 + d_2 s_n^2 + d_1 s_n + 1}, \tag{11.12}$$

where the time scale factor by which the transfer function is 'faster' than the normalised one is $\alpha = (K_p K_c)^{1/5}$ and $d_i = (a_i + k_{c(i+1)} K_c) / \alpha^{(5-i)}$ for $i = 1$ to 4. Thus time scaling is achieved by varying K_c and then the standard form accomplished by choosing the values of k_{c2} to k_{c5} to give the required values of d_1 to d_4. Speeding up of the response can be done by increasing K_c or by increasing the feedback gains but this also increases the magnitude of the control signal. Thus, in practice limiting values will normally exist for these quantities. Trade offs are of course possible if there is some flexibility in the allowable response time. It may, for example, be possible to choose the time scale so as to require no feedback from one state or to realise an almost standard form with no feedback from more than one state.

If the plant has a zero then C_f will be of the form $C_f = (K_p \; K_2 \; 0 \; 0 \; 0)$. Proceeding as above the only change is that the numerator of the transfer function will be $K_c K_p + K_c K_2 s$ which in normalised form is $1 + c_1 s_n$, with $c_1 = \alpha K_2 / K_p$ and the parameters d_i now need to be chosen for the chosen value of c_1, which depends on the time scale factor.

For a plant transfer function with no integration term the design can be achieved using a PI controller. In this case again assuming the plant transfer function is in controllable canonical form

$$A_c = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & -a_4 \end{pmatrix}, B_c = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } C_c = (K_p \; 0 \; 0 \; 0 \; 0). \tag{11.13}$$

If state feedback is applied according to $u = r - k_c^T x_c$, where r is the output of the PI controller with transfer function $(K_1+K_2s)/s$, then the closed loop transfer function is

$$\frac{K_p(K_1+K_2s)}{s^6+(a_4+k_{c5})s^5+....+(a_0+k_{c1})s+K_p(K_1+K_2s)} \qquad (11.14)$$

The transfer function is normalised by dividing by K_pK_1 to give

$$\frac{1+c_1s_n}{s_n{}^6+d_5s_n{}^5+d_4s_n{}^4+d_3s_n{}^3+d_2s_n{}^2+d_1s_n+1} \qquad (11.15)$$

where $d_i = [a_{(i-1)}+k_{ci}]/\alpha^{(6-i)}$ for i = 2 to 5, $d_1 = (a_0+k_{c1}+K_pK_2)/\alpha^5$, $c_1 = \alpha K_2/K_1$, and the time scale factor $\alpha = (K_pK_1)^{1/6}$. Thus, in principle K_1 can be chosen to select the time scale, K_2 to select the zero and the feedback gains to get the correct values of the d coefficients for the chosen zero. Again trade offs are possible if there is flexibility in the choice of the response speed. To illustrate the procedure two examples are given below.

Example 1

A system with a plant transfer function having the state space representation

$$A=\begin{pmatrix}0 & 13 & 6\\ 0 & 1 & 1\\ 0 & -12 & -6\end{pmatrix}, \quad B=\begin{pmatrix}-1\\ 0\\ 1\end{pmatrix} \text{ and } C = (10\ 10\ 10).$$

is considered.

If x_c denotes the state for the controllable canonical form and x that for the original system, then the required transformation $x=Tx_c$ has

$$T=\begin{pmatrix}1 & 1 & -1\\ 0 & 1 & 0\\ 0 & -1 & 1\end{pmatrix}$$

and the controllable canonical form is

$$A_c=\begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & -6 & -5\end{pmatrix}, \quad B_c=\begin{pmatrix}0\\ 0\\ 1\end{pmatrix} \text{ and } C_c = (10\ 0\ 0).$$

The corresponding system transfer function is $(10s+10)/(s^3+5s^2+6s)$, which is seen to contain an integration term. Designs are carried out for the ISTE and IST^{2}E criteria, with $\alpha=1$ and 2. First for the ISTE cases with $\alpha = 1$, $K_c = 0.1$, $c_1 = 1$, $d_1 = 2.32$, $d_2 = 1.80$ giving $K_c k_c^T = (1\ -3.68\ -3.20)$ and $K_c k^T = (1\ -6.88\ -2.20)$ and secondly with $\alpha = 2$, $K_c = 0.8$, $c_1 = 2$, $d_1 = 2.95$, $d_2 = 2.20$ giving $K_c k_c^T = (8\ 5.80\ -0.60)$ and $K_c k^T = (8\ 5.20\ 7.40)$. For the IST^{2}E cases with $\alpha = 1$, $K_c = 0.1$, $c_1 = 1$, $d_1 = 2.43$, $d_2 = 2.14$ giving $K_c k_c^T = (1\ -3.58\ -2.86)$ and $K_c k^T = (1\ -6.43\ -1.86)$ and secondly with $\alpha = 2$, $K_c = 0.8$, $c_1 = 2$, $d_1 = 3.09$, $d_2 = 2.61$ giving $K_c k_c^{\mathrm{T}} = (8\ 6.36\ 0.22)$ and $K_c k^T = (8\ 6.58\ 8.22)$. The responses for the four cases are shown in Fig.11.2.

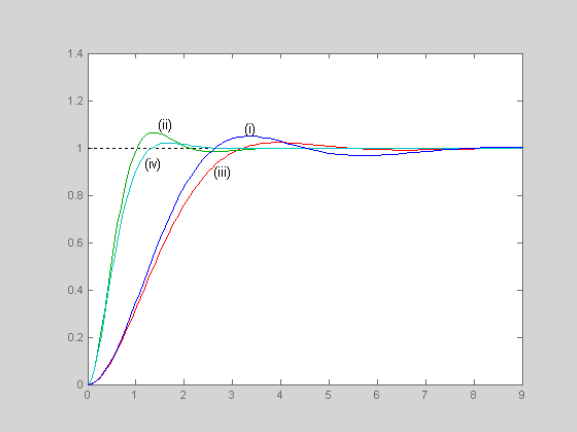

Fig.11.2 Responses for example 1 for ISTE (i) and (ii); IST^{2}E (iii) and (iv)

Example 2

Consider a plant with state space representation

$$A = \begin{pmatrix} 3.5 & 14.5 & 6.5 \\ -0.5 & 0.5 & 0.5 \\ -3 & -13 & -7 \end{pmatrix}, \quad B = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \text{ and } C = (10\ 0\ \text{-}10).$$

The transformation $x = Tx_c$ to put the representation into controllable canonical form has

$$T = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \text{ and gives } A_c = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -10 & -7 & -3 \end{pmatrix}.$$

Using the state feedback $u = r - k_c^T x_c$ and the PI controller $(K_1+K_2)/s$ gives the closed loop transfer function

$$\frac{10(K_1 + K_2 s)}{s^4 + (3 + k_{c3})s^3 + (7 + k_{c2})s^2 + (10 + k_{c1} + 10K_2)s + 10K_1},$$

which can be put into normalised form by dividing by $10K_1$.

The time scale factor $\alpha = (10K_1)^{1/4}$ and the coefficients of the normalised form are $c_1 = \alpha K_2/K_1$, $d_3 = (3+k_{c3})/\alpha$, $d_2 = (7+k_{c2})/\alpha^2$ and $d_3 = (10+k_{c1}+10K_2)/\alpha^3$. From examination of these values it can be seen that time scaling by two should give reasonable feedback gain values. Fig 11.3 shows the coefficients required as functions of c_1 for ISE, ISTE and IST^{2}E designs, from which it can again be seen that the coefficients increase with increasing values of c_1 and are larger for a given c_1 the higher the time weighting in the performance index. Designs are done to minimise the ISTE with $\alpha = 2$ and $c_1 = 1$ and 4. For these two cases the required values of the feedback vector k_c^T are (4.0 6.92 0.64) and (-0.08 16.0 2.04), respectively. From the original system states the required vectors are $k_c^T T^{-1}$ and are (1.68 9.24 2.32) and (-1.06 17.0 0.98), respectively. The resulting output responses for the two cases are shown in Fig 11.4, where the one with the faster rise time is for $c_1 = 4$.

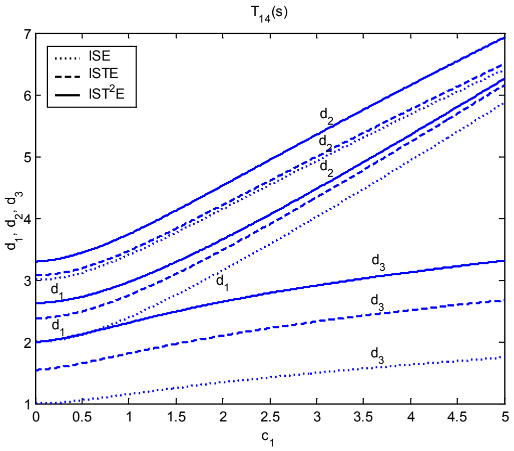

Fig 11.3 Optimum coefficients for $T_{14}(s)$.

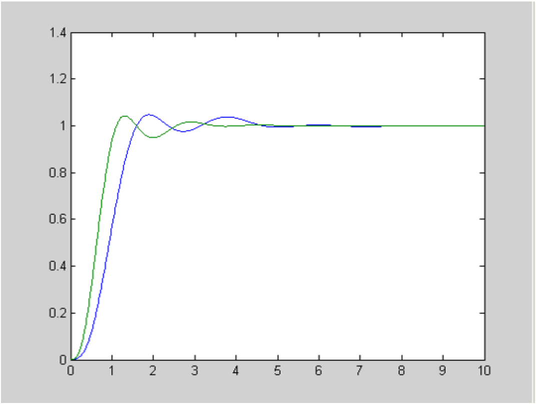

Fig 11.4 Closed loop step responses for example 2.

11.5 Transfer Function with Complex Poles

The design of a state feedback compensator is considered for the plant transfer function with complex poles, $G(s) = \frac{0.1}{s(s^2+0.2s+1)}$ discussed in section 9.5. So that the sensitivity of the response to changes in the damping and natural frequency of the lightly damped poles can be seen the transfer function is taken as $G(s) = \frac{K_p\omega_o^2}{s(s^2+2\zeta s+\omega_o^2)}$, where the nominal values of K_p, $\omega_{o,}$ and ς are 0.1, 1 and 0.1, respectively. In controllable canonical form, with a scaling on $\dot{x}_2$ and the plant gain taken at the input, the state space matrices are

$$A_c = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \omega_o \\ 0 & -\omega_o & -2\zeta\omega_o \end{pmatrix},\ B_c = \begin{pmatrix} 0 \\ 0 \\ K_p \end{pmatrix} \text{ and } C_c = (1 \quad 0 \quad 0). \tag{11.16}$$

The closed loop transfer function with the state feedback is then

$$\frac{K_pK_c\omega_o^2}{s^3+(2\zeta\omega_o+k_{c3}K_pK_c\omega_o)s^2+(\omega_o^2+\omega_o^2k_{c2}K_pK_c)s+\omega_o^2k_{c1}K_pK_c} \tag{11.17}$$

It can be seen from the transfer function that without feedback the coefficient of the s^2 term can be much smaller than that of the s term. With k_{c1} = , the nominal parameters substituted and the controller gain chosen as 10 the closed loop transfer function is $G(s) = \frac{1}{s^3 + (0.2 + k_{c3})s^2 + (1 + k_{c2})s + 1}$, which is in normalised form. Doing an ISTE design requires k_{c3} = 1.272and k_{c2} = 1.042. Closed loop step responses are shown for this design in Figure 11.5. The responses marked 0.8 and 1.2 are obtained with the natural frequency of the plant at these values rather than the nominal one of unity. The three responses shown, all marked 1.0, are for the nominal natural frequency of unity and with damping ratios equal to 0.05, 0.1 the nominal value, and 0.15, respectively, for the plant. It can be seen from these results that a much better performance can be achieved by state feedback than by the series compensator used in section 9.5 if the plant parameters change.

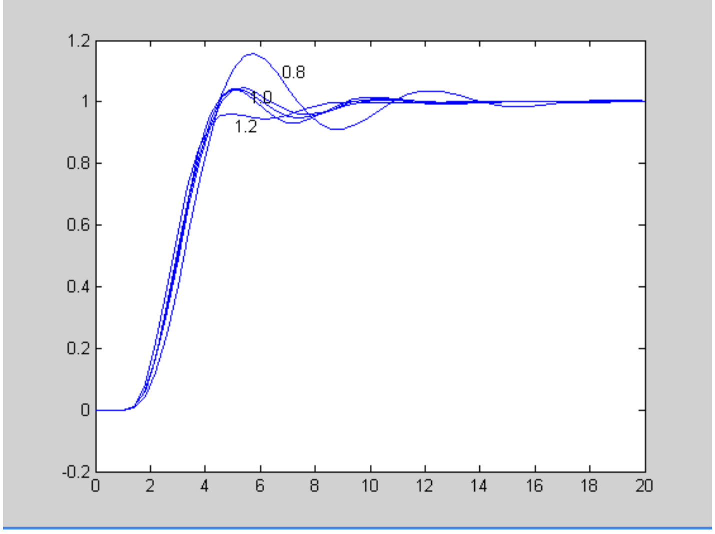

Fig. 11.5 Closed loop step responses for the system having a plant with complex poles

12 Appendix A

The Laplace Transformation

The defining integral of the transformation is

$$F(s) = \int_0^\infty e^{-st} f(t)dt$$

where *s* is the Laplace complex variable, often denoted by, with units seconds^{-1}. In shorthand form the notatio $\mathscr{L}$ is often used to denote the transformation that is

$F\ (s) = \mathscr{L}[f(t))]$ and for the inverse $f(t) = \mathscr{L}^{-1}[F(s)]$.

f(*t*) and *F*(*s*) are referred to as transform pairs and have a unique correspondence, that is, for a given *f*(*t*) or *F*(*s*) there is a corresponding unique *F*(*s*) or *f*(*t*). The three 'most basic' results are:

1) $\mathscr{L}[e^{-at}] = 1/(s+a)$
2) $\mathscr{L}[sin\ wt] = w/(s^2+w^2)$
3) $\mathscr{L}[cos\ wt] = s/(s^2+w^2)$

The result of (1) is easily shown from the defining integral. It will also be seen on doing the integral that the solution can only exist provided Re(s) > -*a*, so that the integrand tends to zero as *t*→∞. Mathematical rigour such as this and the fact that a contour integral exists to evaluate *f(t)* from *F(s)* do not normally need to be considered for control engineering applications. Results (2) and (3) are then easily obtained from (1) by writing sin *wt* and cos *wt* in terms of complex exponentials, i.e. and using the superposition theorem (i) below. Useful theorems are:

i) Superposition theorem

$$\mathcal{L}\left[af_1(t)+bf_2(t)\right]=aF_1(s)+bF_2(s)$$

ii) Complex shifting theorem

$$\mathcal{L}\left[e^{-at}f(t)\right]=F(s+a)$$

iii) Derivative theorem

$$\mathcal{L}\left[\frac{df}{dt}\right]=sF(s)-f(0)$$

$$\mathcal{L}\left[\frac{d^n f}{dt^n}\right]=s^nF(s)-s^{n-1}f(0)-s^{n-2}\frac{df(0)}{dt}\ldots\ldots-\frac{d^{n-1}f(0)}{dt^{n-1}}$$

iv) Integral theorem

$$\mathcal{L}\left[\int_0^t f(\tau)d\tau\right]=\frac{1}{s}F(s)+\frac{p(0^+)}{s}$$ where

v) *t* multiplying (or complex differentiation) theorem

$$\mathcal{L}\left[tf(t)\right]=-\frac{dF(s)}{ds}$$

vi) Real shifting theorem

$$\mathcal{L}\left[f(t-a)u_0(t-a)\right]=e^{-as}F(s)$$

where $u_0(t)$ is the unit step at $t = 0$.

All other transforms which are normally required can be easily obtained from the above information. For example, using (ii) with (2) and (3) gives

4) $$\mathcal{L}\left[e^{-at}\sin\omega t\right]=\frac{\omega}{(s+a)^2+\omega^2}$$

5) $$\mathcal{L}\left[e^{-at}\cos\omega t\right]=\frac{s+a}{(s+a)^2+\omega^2}$$

Also noting the unit step $u_0(t)$ for $t > 0$ $\lim_{a\to 0} e^{-at}$ it follows that

6) $\mathcal{L}\left[u_0(t)\right] = \frac{1}{s}$.

The unit impulse, $d(t)$, equals $\frac{d}{dt}\left[u_0(t)\right]$, so that from (6) and (iii)

7) $\mathcal{L}\left[\delta(t)\right] = 1$

The inverse transforms corresponding to higher order denominators with repeated roots in $F(s)$ can be found by repeated use of (v), for example,

8) $\mathcal{L}\left[tf(t)\right] = \frac{1}{(s+a)^2}$

9) $\mathcal{L}\left[t\right] = \frac{1}{s^2}$

Additional useful theorems are:

vii) Final value theorem

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

viii) Initial value theorem

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

ix) Convolution theorem

$$\mathcal{L}\left[\int_0^t f_1(\tau) f_2(t-\tau)\,d\tau\right] = F_1(s)F_2(s)$$

Transforming a Differential Equation

Consider the differential equation:

$$\ddot{y} + a\dot{y} + by = u$$

where $\dot{}$ denotes the derivative w.r.t. time. Taking transforms of both sides of the equation and denoting the transforms of *y(t)* and *u(t)* by *Y(s)* and *U(s)*, as is a standard convention, use of the theorem in (iii) above gives

$$s^2Y(s) - sy(0) - \dot{y}(0) + a(sY(s) - y(0) + bY(s) = U(s)$$

that is

$$(s^2 + as + b)Y(s) - y(0)\,(a+s) - \dot{y}\,(0) = U(s).$$

Note that to solve the equation, since it is second order, two initial conditions are required. When the initial conditions are zero the equation becomes

$$(s^2 + as + b)\,Y(s) = U(s)$$

and when written in the form

$$\frac{Y(s)}{U(s)} = \frac{1}{s^2 + as + b}$$

the resulting expression is known as the transfer function between the input, *u*, and output, *y*. Systems are often described by their transfer functions, often denoted by a single symbol such as *G(s)* or *F(s)*.

To obtain $\mathscr{L}^{-1}[F(s)]$ one first puts $F(s)$ into partial fractions. This requires determination of the roots of the denominator of $F(s)$ which will be real or complex pairs. If (4) and (5) are used then the complex roots may be left as a quadratic factor.

Finding the Inverse of *F(s)*

As an example consider the transfer function

$$F(s) = \frac{2s^2 + 4s + 3}{s^3 + 2s^2 + 2s + 1}$$

The denominator can be factored into $(s + 1)$ $(s^2 + s + 1)$ and $F(s)$ written as

$$F(s) = \frac{1}{s+1} + \frac{s+2}{s^2 + s + 1}$$

Partial fraction expansions can be obtained using MATLAB but it separates the quadratic into the two complex poles (s + 0.5 + j0.866) and (s + 0.5 – j0.866). The command to do this is **[r,p,k]=residue(b,a)**. The required inputs are vectors *b* and *a*, the coefficients of the numerator and denominator polynomials in descending order. The residues are returned in r, the poles in p and any direct gain in k. Thus for the above transfer function one obtains:-

```
>> [r,p,k]=residue([2 4 3],[1 2 2 1])

r =

1.0000
0.5000 – 0.8660i
0.5000 + 0.8660i

p =

-1.0000
-0.5000 + 0.8660i
-0.5000 – 0.8660i

k =

[]
```

The quadratic factor can then be split into the forms of (4) and (5), that is

$$F(s) = \frac{1}{s+1} + \frac{s + 1/2}{(s + 1/2)^2 + \left(\sqrt{3}/2\right)^2} + \frac{\sqrt{3}/2}{(s + 1/2)^2 + \left(\sqrt{3}/2\right)^2} . \sqrt{3}$$

so that from (1), (4) and (5) one obtains

$$f(t) = e^{-t} + e^{-t/2}\cos\frac{\sqrt{3}t}{2} + \sqrt{3}e^{-t/2}\sin\frac{\sqrt{3}t}{2}$$

Note that $f(0) = 2$ and

$$\lim_{s\to\infty} sF(s) = \lim_{s\to\infty} \frac{2s^3 + 4s^2 + 3s}{s^3 + 2s^2 + 2s + 1} = 2$$

in agreement with the initial value theorem.

Also $f(\infty) = 0$ and

$$\lim_{s\to 0} sF(s) = \lim_{s\to 0} \frac{2s^3 + 4s^2 + 3s}{s^3 + 2s^2 + 2s + 1} = 0$$

in agreement with the final value theorem.

13 Appendix B

The Routh-Hurwitz Criterion.

Research on the roots of a polynomial was an important area of work in mathematics in the 19th century. The work of Routh and later Hurwitz led to the Routh-Hurwitz criterion which is useful for investigating the stability of linear feedback systems. Given the polynomial

$$F(s) = f_n s^n + f_{(n-1)} s^{(n-1)} + \dots . f_1 s + f_0 \text{ with } f_0 > 0 \tag{B1}$$

The following table is constructed from the polynomial coefficients

Row label				
s^n	f_n	f_{n-2}	f_{n-4}	…..
s^{n-1}	f_{n-1}	f_{n-3}	f_{n-5}	…..

It is then continued as follows

Row label				
s^n	f_n	f_{n-2}	f_{n-4}	…..
s^{n-1}	f_{n-1}	f_{n-3}	f_{n-5}	…..
s^{n-2}	b_1	b_2	b_3	…..
s^{n-3}	c_1	c_2	c_3	……
.	.	.	.	
.	.	.	.	
s^0	g_1			

with the new values found from

$$b_1 = -\frac{1}{f_{n-1}}\begin{vmatrix} f_n & f_{n-2} \\ f_{n-1} & f_{n-3} \end{vmatrix}, \quad b_2 = -\frac{1}{f_{n-1}}\begin{vmatrix} f_n & f_{n-4} \\ f_{n-1} & f_{n-5} \end{vmatrix}, \quad c_1 = -\frac{1}{b_1}\begin{vmatrix} f_{n-1} & f_{n-3} \\ b_1 & b_2 \end{vmatrix}, \quad c_2 = -\frac{1}{b_1}\begin{vmatrix} f_{n-1} & f_{n-5} \\ b_1 & b_3 \end{vmatrix}$$

and so on until one number is reached in the first column, here denoted by g_1. The number of sign changes of the numbers in the first column, that is f_n, f_{n-1},b_1, c_1, …… g_1, gives the number of roots in the right hand side of the s plane.

Thus for a stable polynomial, for $f_0 > 0$, they will all be positive. Two special cases arise (i) when a number in the first column is zero and (ii) when all the elements of a row are zero. These are discussed with examples in many texts. In case (i) the zero is replaced by a small positive value ε, the table completed, and the sign changes examined as $\varepsilon \to 0$. Case (ii) arises when a pair of roots give a term of the form ($s^2 \pm \alpha^2$). In this case the table is restarted by forming a polynomial from the coefficients in the row above the row with the zeros.

14 Appendix C

Pade Table.

As mentioned in section 6.2 a result was obtained by Pade which gives approximations for e^{-x} as a ratio of polynomials. Different orders can be chosen for the numerator, n, and denominator, m, polynomials as given in the following table for n, $m \le 3$.

m – n	0	1	2	3
0	$\frac{1}{1}$	$\frac{1-x}{1}$	$\frac{1-x+\frac{x^2}{2}}{1}$	$\frac{1-x+\frac{x^2}{2}-\frac{x^3}{6}}{1}$
1	$\frac{1}{1+x}$	$\frac{1-\frac{x}{2}}{1+\frac{x}{2}}$	$\frac{1-\frac{2x}{3}+\frac{x^2}{6}}{1+\frac{x}{3}}$	$\frac{1-\frac{3x}{4}+\frac{x^2}{4}-\frac{x^3}{24}}{1+\frac{x}{4}}$
2	$\frac{1}{1+x+\frac{x^2}{2}}$	$\frac{1-\frac{x}{3}}{1+\frac{2x}{3}+\frac{x^2}{6}}$	$\frac{1-\frac{x}{2}+\frac{x^2}{12}}{1+\frac{x}{2}+\frac{x^2}{12}}$	$\frac{1-\frac{3x}{5}+\frac{3x^2}{20}-\frac{x^3}{60}}{1+\frac{2x}{5}+\frac{x^2}{20}}$
3	$\frac{1}{1+x+\frac{x^2}{2}+\frac{x^3}{6}}$	$\frac{1-\frac{x}{4}}{1+\frac{3x}{4}+\frac{x^2}{4}+\frac{x^3}{24}}$	$\frac{1-\frac{2x}{5}+\frac{x^2}{20}}{1+\frac{3x}{5}+\frac{3x^2}{20}+\frac{x^3}{60}}$	$\frac{1-\frac{x}{2}+\frac{x^2}{10}-\frac{x^3}{120}}{1+\frac{x}{2}+\frac{x^2}{10}+\frac{x^3}{120}}$

Table C Pade expansions for e^{-x}.

15 Appendix D

Table of Integrals

As mentioned in section 8.1 the integral $J_0 = \int_0^\infty e^2(t)dt = \frac{1}{2j\pi}\int_{-j\infty}^{j\infty} E(s)E(-s)ds$ can be evaluated in the s-domain when $E(s) = \frac{c(s)}{d(s)}$ and $c(s)$ and $d(s)$ are the polynomials

$c(s) = c_{n-1}s^{n-1} + c_{n-2}s^{n-2} + ...c_1 s + c_0$ and $d(s) = d_n s^n + d_{n-1}s^{n-1} + ...d_1 s + d_0$. A short table is given below where the integral is denoted by I_n for $d(s)$ of order n.

$I_1 = \frac{c_0^2}{2d_0 d_1}$
$I_2 = \frac{c_1^2 d_0 + c_0^2 d_2}{2d_0 d_1 d_2}$
$I_3 = \frac{c_2^2 d_0 d_1 + (c_1^2 - 2c_0 c_2) d_0 d_2 + c_0^2 d_2 d_3}{2d_0 d_3 (d_1 d_2 - d_0 d_3)}$
$I_4 = \frac{c_3^2 (d_0 d_1 d_2 - d_3 d_0^2) + (c_2^2 - 2c_1 c_3) d_0 d_1 d_4 + (c_1^2 - 2c_0 c_2) d_0 d_3 d_4 + c_0^2 (d_2 d_3 d_4 - d_1 d_4^2)}{2d_0 d_4 (d_1 d_2 d_3 - d_0 d_3^2 - d_4 d_1^2)}$