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.

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.

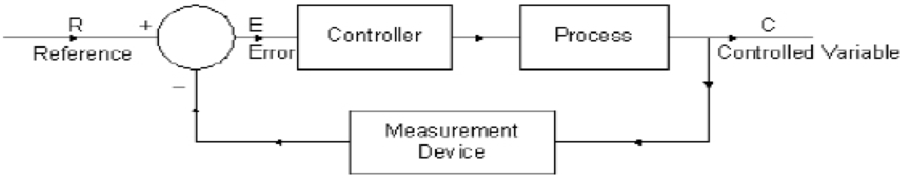


Figure 1.1 Basic Feedback Control Structure

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

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

$$A(D) = D^n + a_{n-1}D^{n-1} + a_{n-2}D^{n-2}.........a_1D + a_0 \tag{2.2}$$

$$B(D) = D^m + b_{m-1}D^{m-1} + b_{m-2}D^{m-2}.......b_1D + b_0 \tag{2.3}$$

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

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 $\mathcal{L}$ denotes the Laplace transform then one may write $\mathcal{L}[f(t)] = F(s)$ and $\mathcal{L}^{-1}[F(s)] = f(t)$. 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)

$$A(s) = s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2}.........a_1s + a_0 \tag{2.10}$$

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

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

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

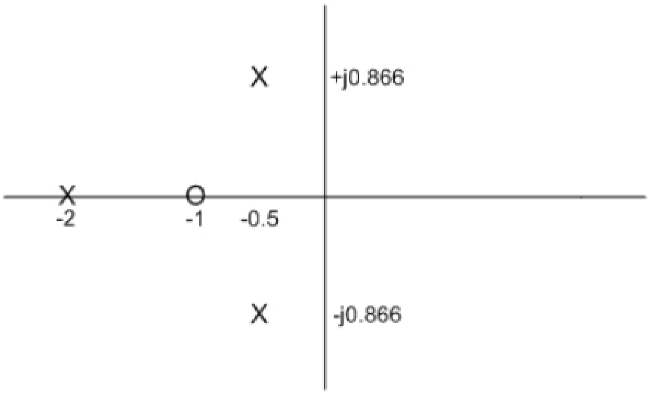


Figure 2.1 Zero-pole plot.

2.3 State space representations

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

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

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

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

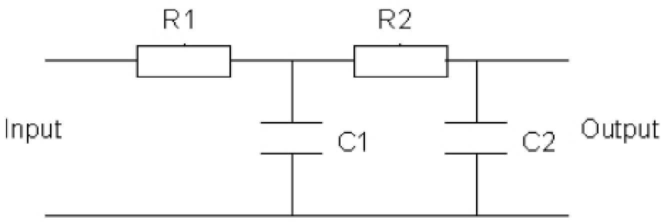


Figure 2.2 Simple *R-C* circuit.

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

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

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

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

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

In matrix form the state and output equations can be written

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

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

```
>> 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.

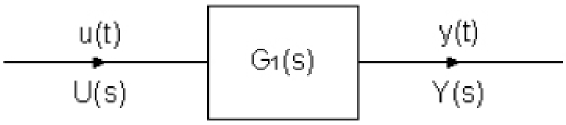


Figure 2.3 Block representation of a transfer function

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.

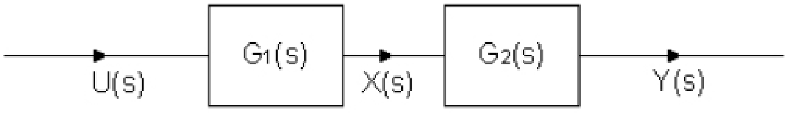


Figure 2.4 Series (or cascade) connection of blocks

If two 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 $\mathbf{G=G_1+G_2}$

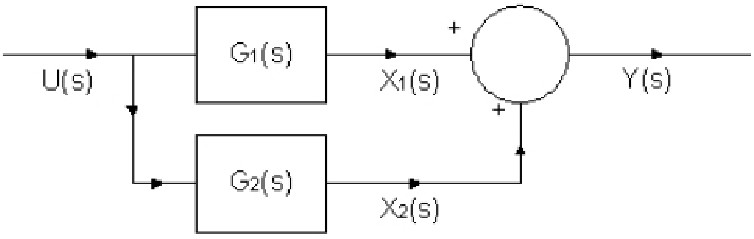


Figure 2.5 Parallel connection of blocks

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

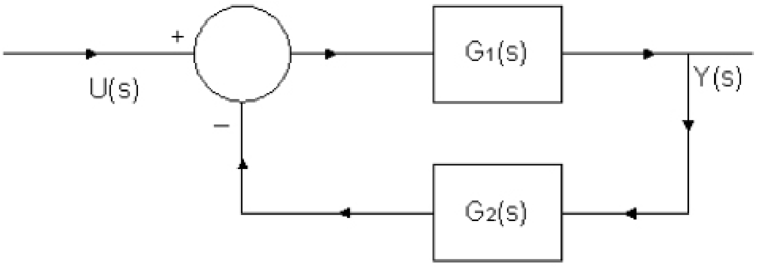


Figure 2.6 Feedback connection of blocks.

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

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

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

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

Taking the inverse Laplace transform gives the result

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

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.

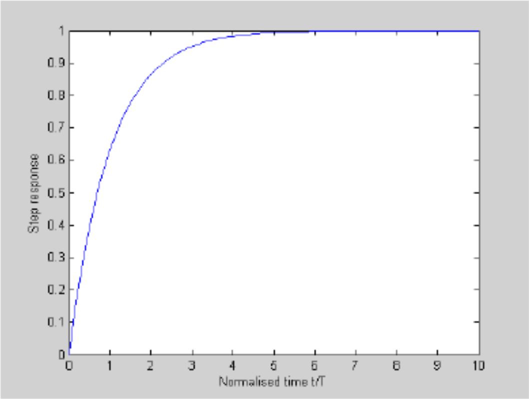


Figure 3.1 Normalised step response for a single time constant transfer function.

3.1.2 Two Complex Poles

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

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

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

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

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

3.1.2 Two Complex Poles

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

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

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

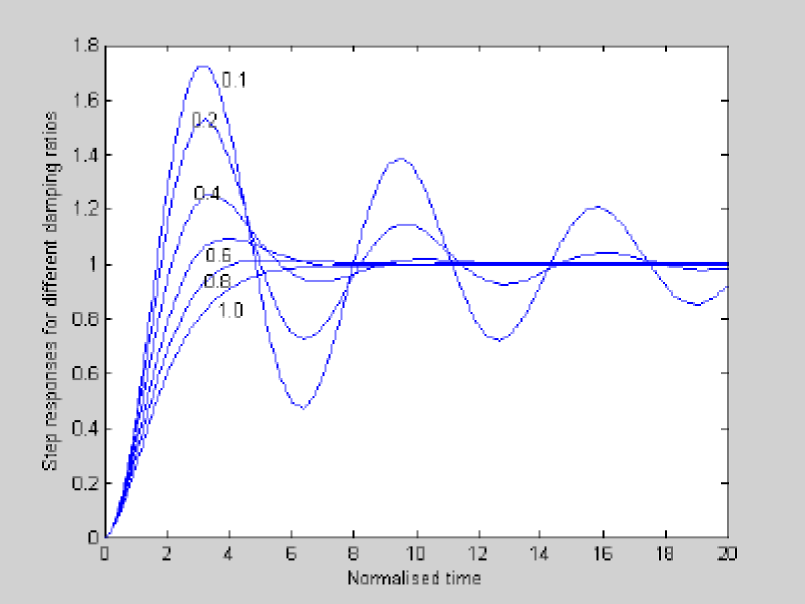


Figure 3.2 Normalised step response of second order system for different ς

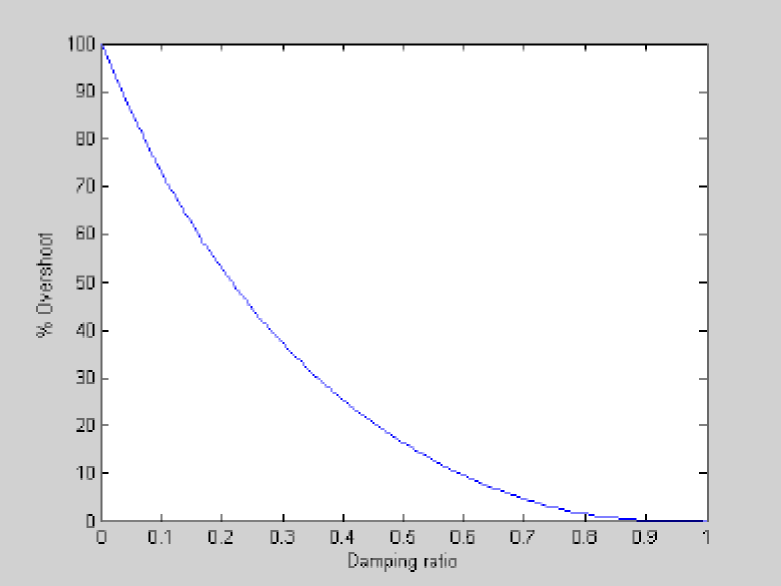


Figure 3.3 Graph of % overshoot as a function of the damping ratio.

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 $G_0(\text{s})$ to a unit step input is $y_0(\text{t})$ then the response of $G(\text{s})$ to a unit step input can be written as

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

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.

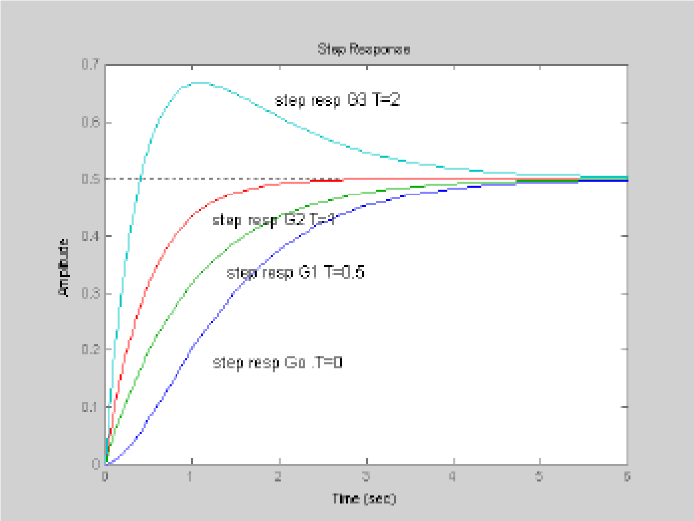


Figure 3.4 Step responses for various values of *T*

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

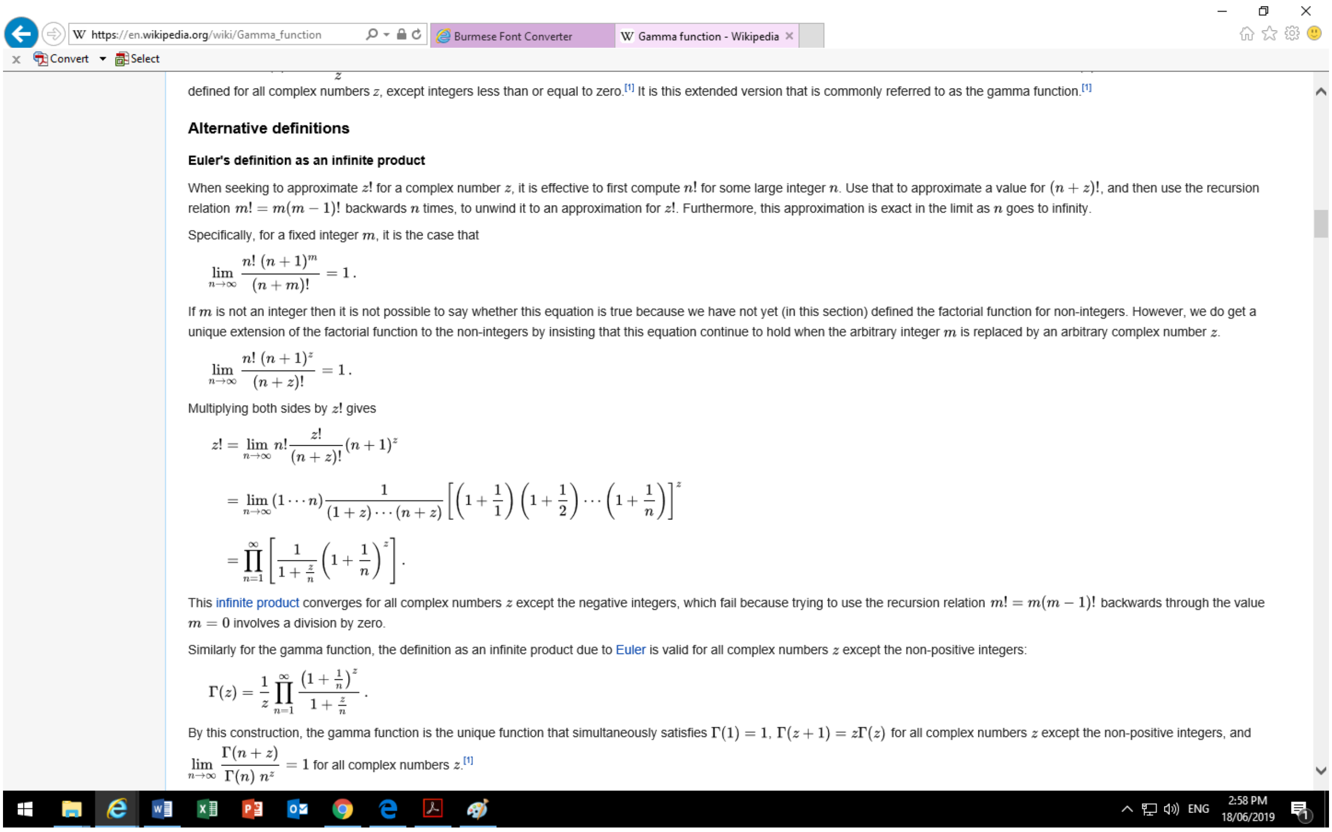
$$G_2(s) = \frac{0.5}{s^2 + 0.2s + 1} + \frac{2.5}{s + 5}.$$

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.

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

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



https://en.wikipedia.org/wiki/Gamma_function
Burmese Font Converter
Gamma function - Wikipedia
Convert
Select
ENG
2:58 PM
18/06/2019

defined for all complex numbers z, except integers less than or equal to zero.[1] It is this extended version that is commonly referred to as the gamma function.[1]

Alternative definitions

Euler's definition as an infinite product

When seeking to approximate $z!$ for a complex number z, it is effective to first compute $n!$ for some large integer n. Use that to approximate a value for $(n+z)!$, and then use the recursion relation $m! = m(m-1)!$ backwards n times, to unwind it to an approximation for $z!$. Furthermore, this approximation is exact in the limit as n goes to infinity.

Specifically, for a fixed integer m, it is the case that

$$\lim_{n\to\infty} \frac{n!\,(n+1)^m}{(n+m)!} = 1\,.$$

If m is not an integer then it is not possible to say whether this equation is true because we have not yet (in this section) defined the factorial function for non-integers. However, we do get a unique extension of the factorial function to the non-integers by insisting that this equation continue to hold when the arbitrary integer m is replaced by an arbitrary complex number z.

$$\lim_{n\to\infty} \frac{n!\,(n+1)^z}{(n+z)!} = 1\,.$$

Multiplying both sides by $z!$ gives

$$\begin{aligned} z! &= \lim_{n\to\infty} n! \frac{z!}{(n+z)!}(n+1)^z \\ &= \lim_{n\to\infty} (1\cdots n)\frac{1}{(1+z)\cdots(n+z)}\left[\left(1+\frac{1}{1}\right)\left(1+\frac{1}{2}\right)\cdots\left(1+\frac{1}{n}\right)\right]^z \\ &= \prod_{n=1}^{\infty}\left[\frac{1}{1+\frac{z}{n}}\left(1+\frac{1}{n}\right)^z\right]. \end{aligned}$$

This infinite product converges for all complex numbers z except the negative integers, which fail because trying to use the recursion relation $m! = m(m-1)!$ backwards through the value $m = 0$ involves a division by zero.

Similarly for the gamma function, the definition as an infinite product due to Euler is valid for all complex numbers z except the non-positive integers:

$$\Gamma(z) = \frac{1}{z}\prod_{n=1}^{\infty}\frac{\left(1+\frac{1}{n}\right)^z}{1+\frac{z}{n}}\,.$$

By this construction, the gamma function is the unique function that simultaneously satisfies $\Gamma(1) = 1$, $\Gamma(z+1) = z\Gamma(z)$ for all complex numbers z except the non-positive integers, and $\lim_{n\to\infty} \frac{\Gamma(n+z)}{\Gamma(n)\,n^z} = 1$ for all complex numbers z.[1]

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 PB_j and 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 = 3j 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 ω = 3, has

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

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

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

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

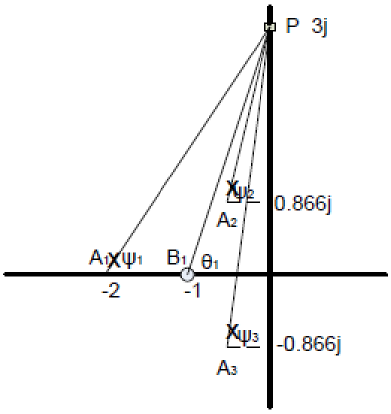


Figure 3.6 Graphical evaluation of a frequency response from the zero-pole plot.

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 $G(j\omega)$ was introduced in the last chapter. As $G(j\omega)$ 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 $G(s)$.

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

$$M = M_1M_2 \text{ and } \varphi = \varphi_1 + \varphi_2 \tag{4.3}$$

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

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

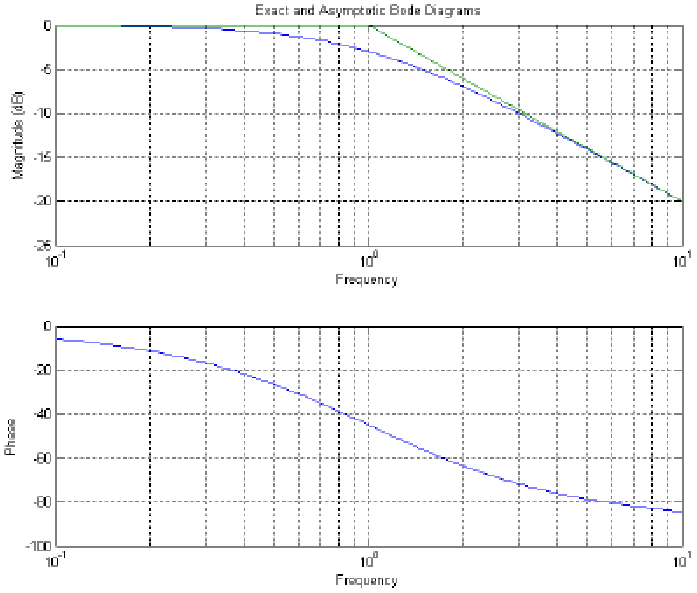


Figure 4.1 Bode exact and approximate magnitude curves, and phase curve, for a single time constant.

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

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

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

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

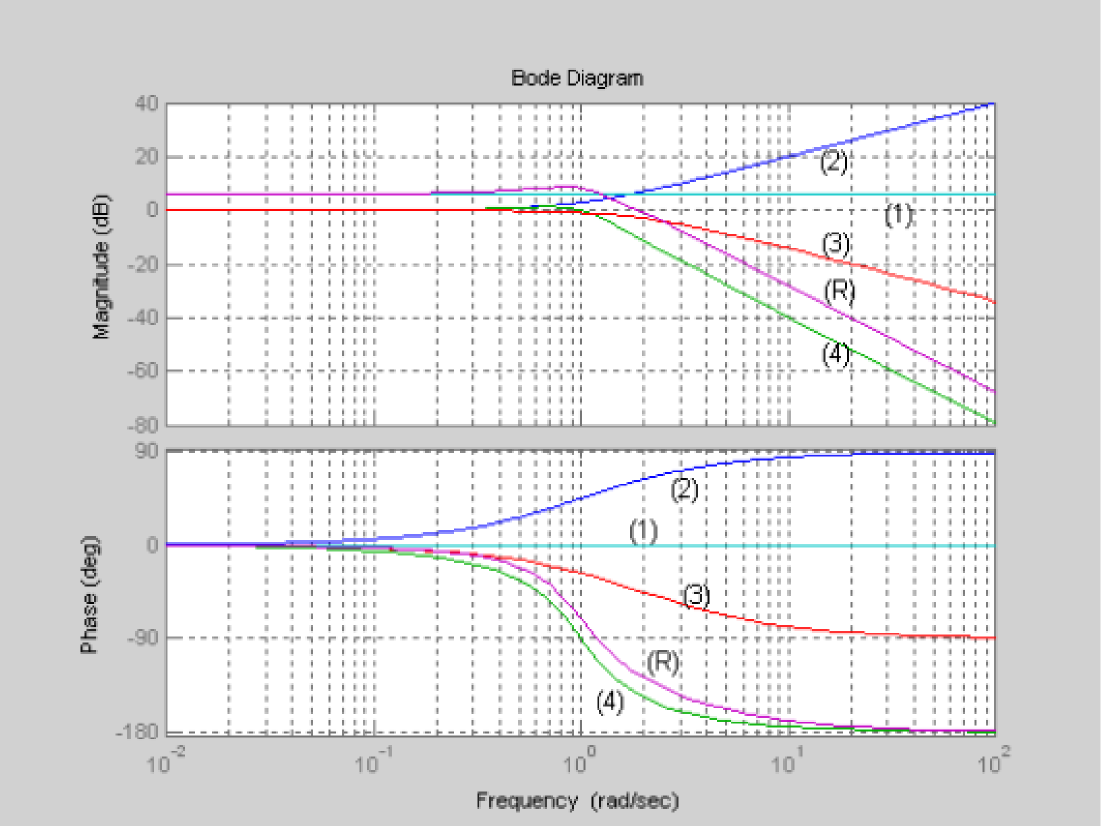
4.2.4 An Example Bode Plot

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

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



4.3 Nyquist Plot

Since

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

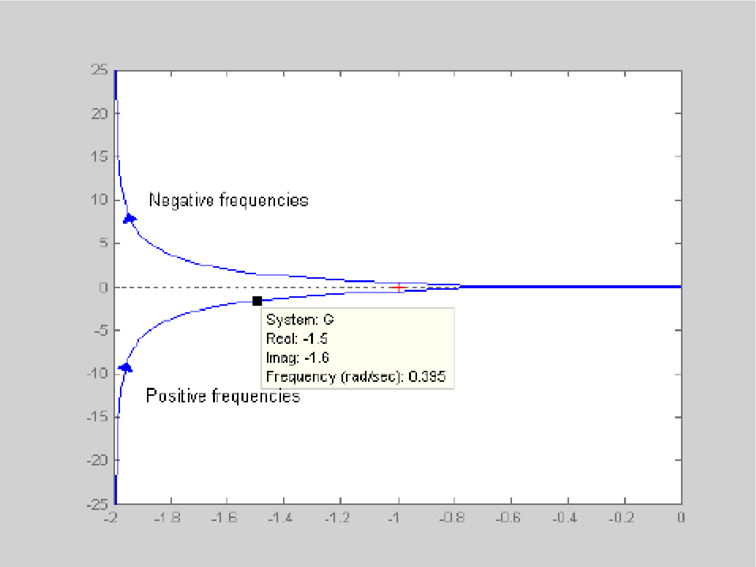
every choice of ω gives a point in a complex plane either plotted in polar coordinates for the M, φ form or in rectangular coordinates in X, Y form. Joining the points together as ω is varied produces a locus with ω as a parameter which is known as a polar or Nyquist plot. To obtain analytical results 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.

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

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 $-6\tan^{-1}\omega$.



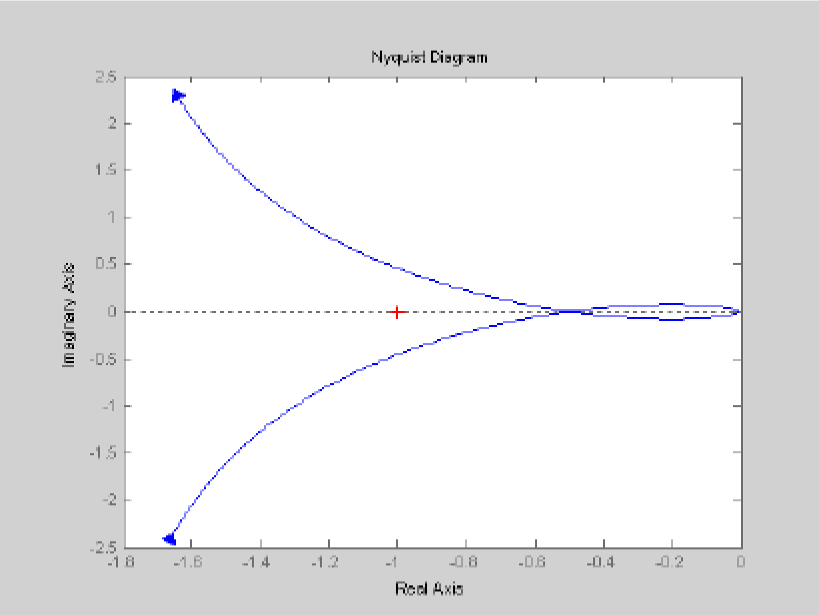


Figure 4.5 Nyquist plot with new *ω* vector.

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°.

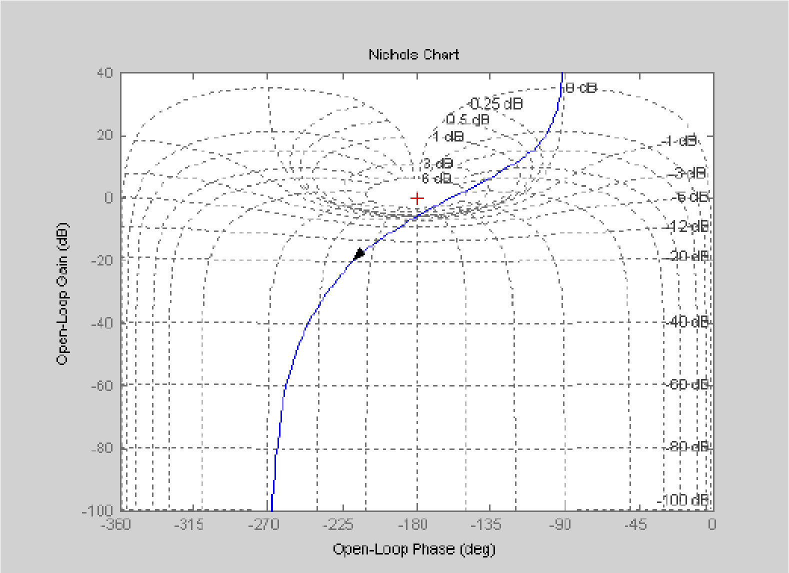


Figure 4.6 Nichols plot of $1/s(s + 1)^2$.

5.1 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.

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.

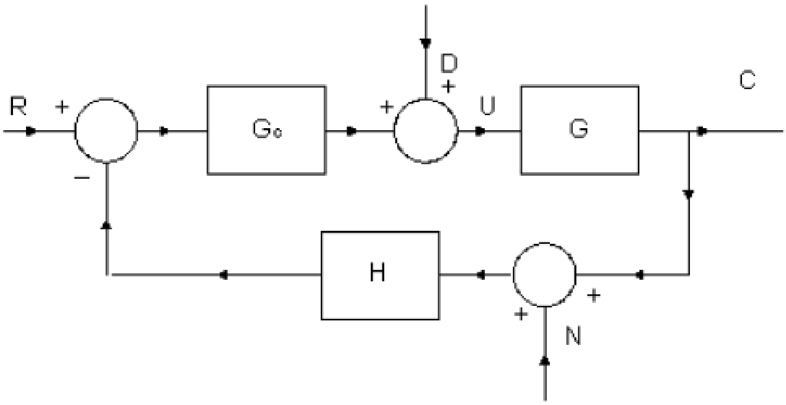


Figure 5.1 Basic Block Diagram of Feedback Control System.

5.2 The Closed Loop

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

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

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

$$G_{ol}(s) = G_c(s)G(s)H(s) \qquad (5.2)$$

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

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

Its poles are the roots of

$$F(s) = 1 + G_c(s)G(s)H(s) = 1 + G_{ol}(s) = 0 \qquad (5.4)$$

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

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

then the closed loop transfer function

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

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 $1/H$.

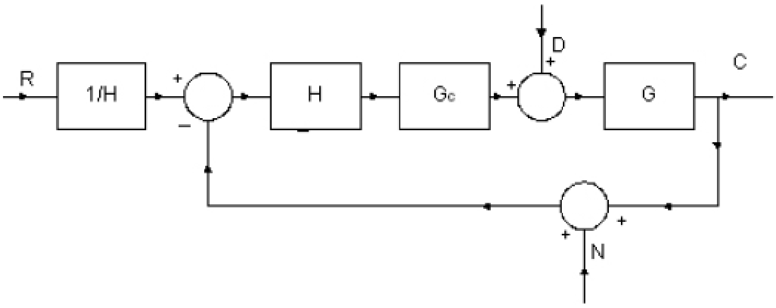


Figure 5.2 Equivalent block diagram to Figure 5.1.

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

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

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.

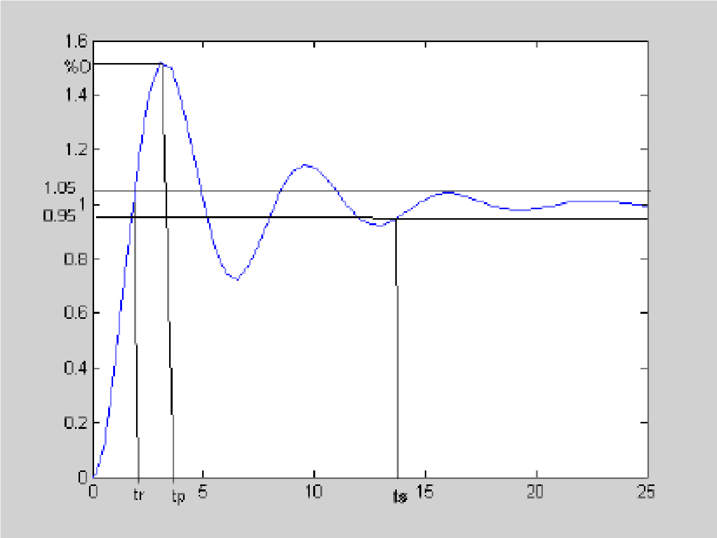


Figure 5.3 Typical step response for specifications.

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*(*jω*) 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.

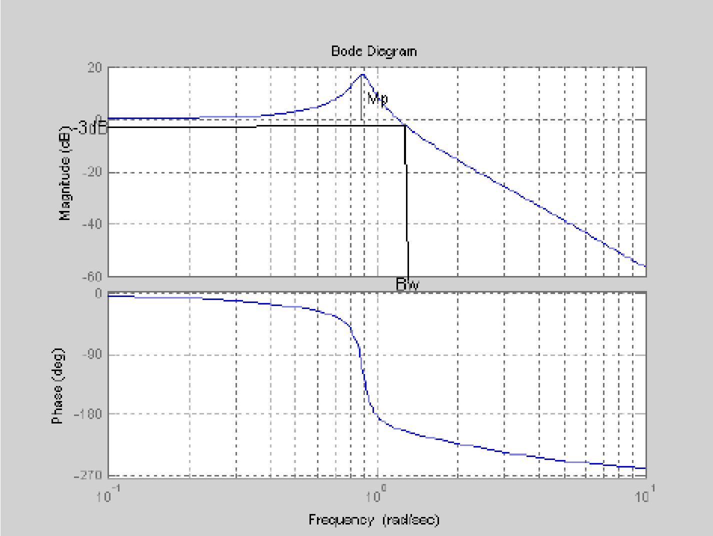


Figure 5.4 Typical frequency response for specifications.

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

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

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*ω) 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

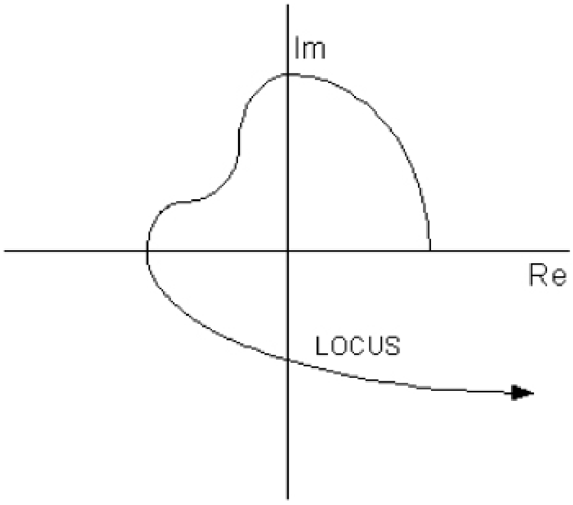


Figure 5.5 Illustration of Mikhailov Criterion for Stable Fourth Order *F*(*j*ω).

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*ω) 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:-

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

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

$$\text{zeros of } F(s) \text{ in rhs} = N_o + \text{poles of } F(s) \text{ in rhs.} \tag{5.10}$$

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

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

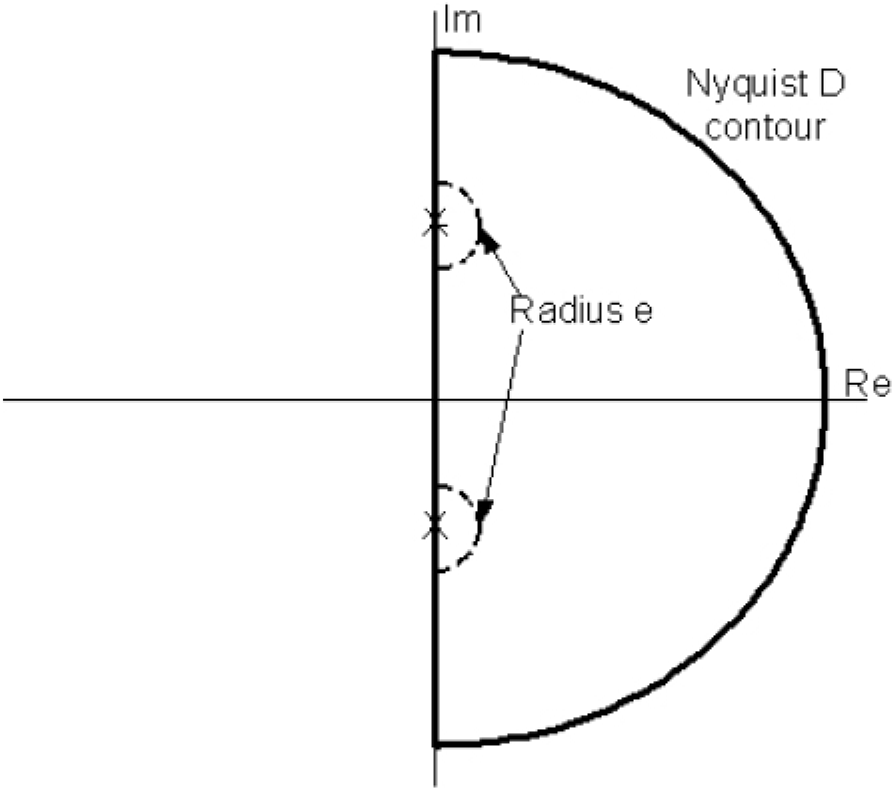


Figure 5.6 Nyquist D contour.

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 $e^{-s\tau}$.

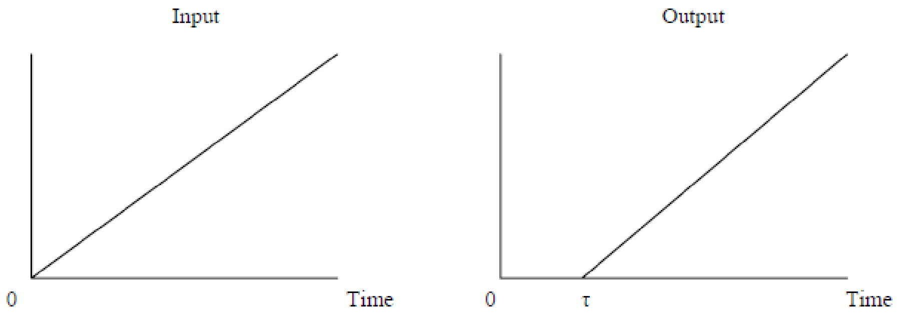


Figure 6.1 Illustration of a time delay

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

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

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]))
```

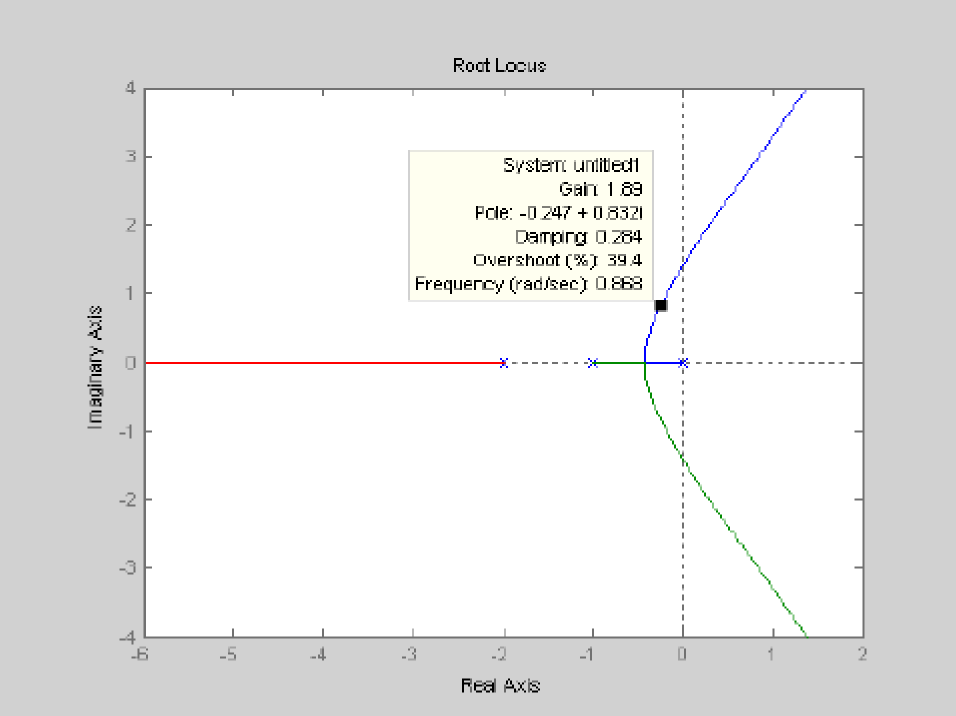


Figure 6.3 Illustrative root locus plot.

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.

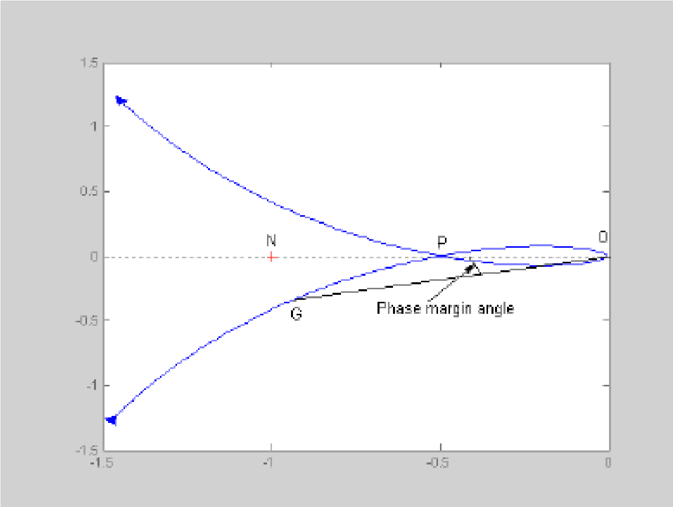
Obviously since instability results from a pole entering the rhs *s*-plane, the nearer a pole in the lhs s-plane is to the imaginary axis the nearer the system being studied will be to instability. Thus in the previous root locus example the nearer the gain to the value of 6 the nearer the two complex 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

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°



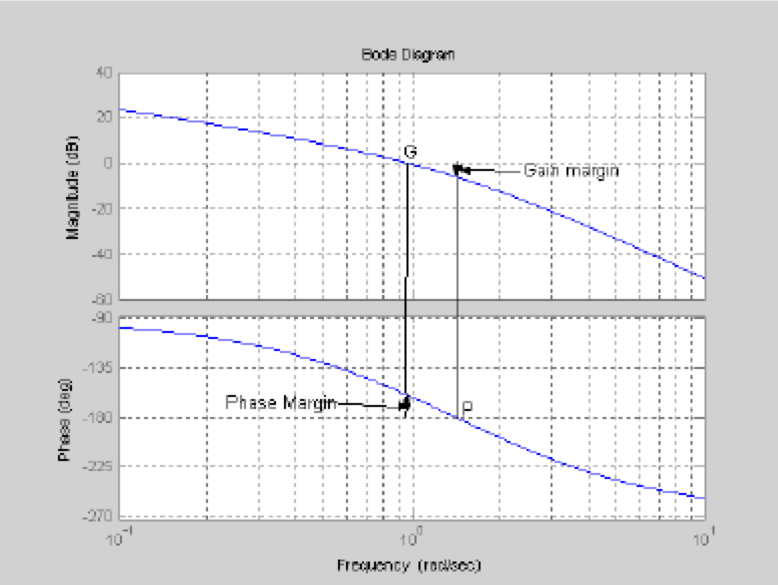


Figure 6.5 Bode plot illustrating gain and phase margins.

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

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

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.

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 $\alpha < 1$.

	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.

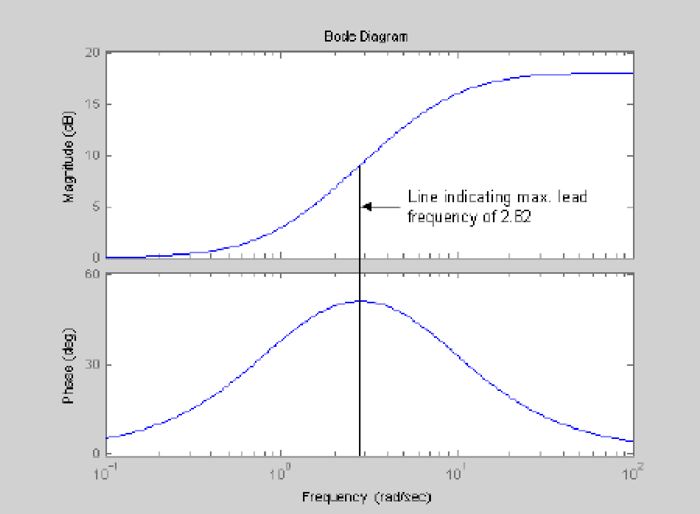


Figure 7.1 Bode diagram of phase lead with $T = 1$ and α = 1/8.

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).

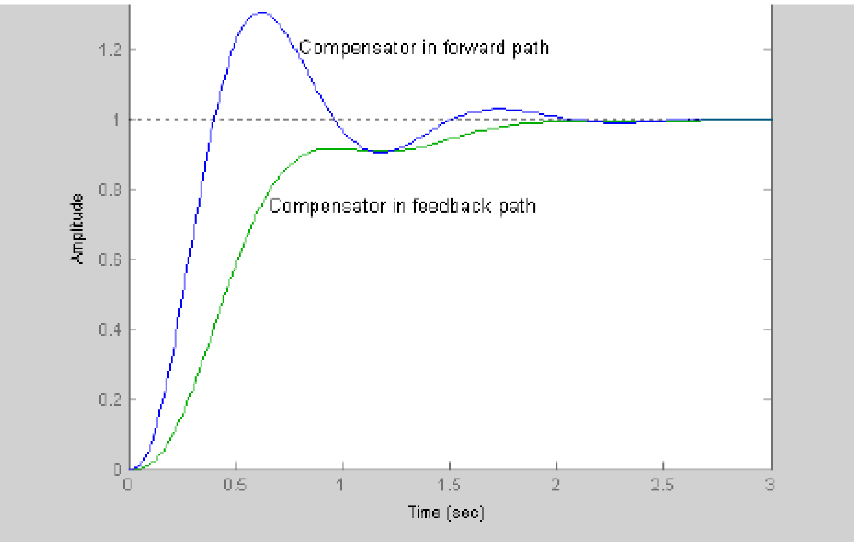
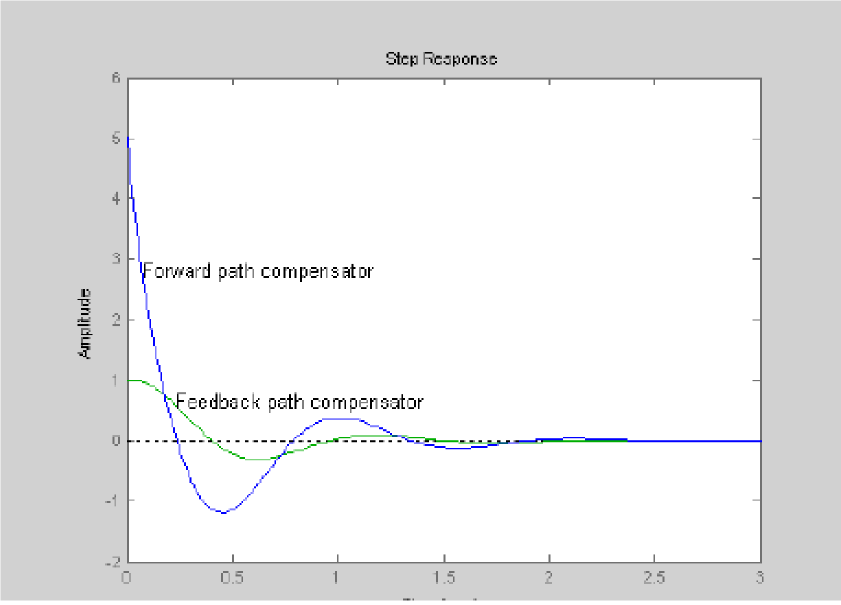


Figure 7.5 Comparison of step responses for forward path and feedback path locations of the compensator.



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 δ = 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, α = 3.12 and the required compensator transfer function is $G_c(s) = \frac{1+6.62s}{1+20.7s}$.

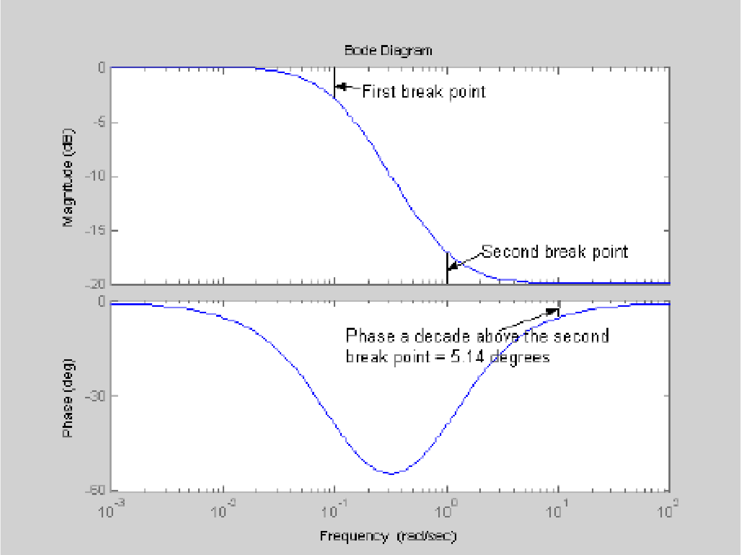


Figure 7.8 Bode diagram of lag compensator.

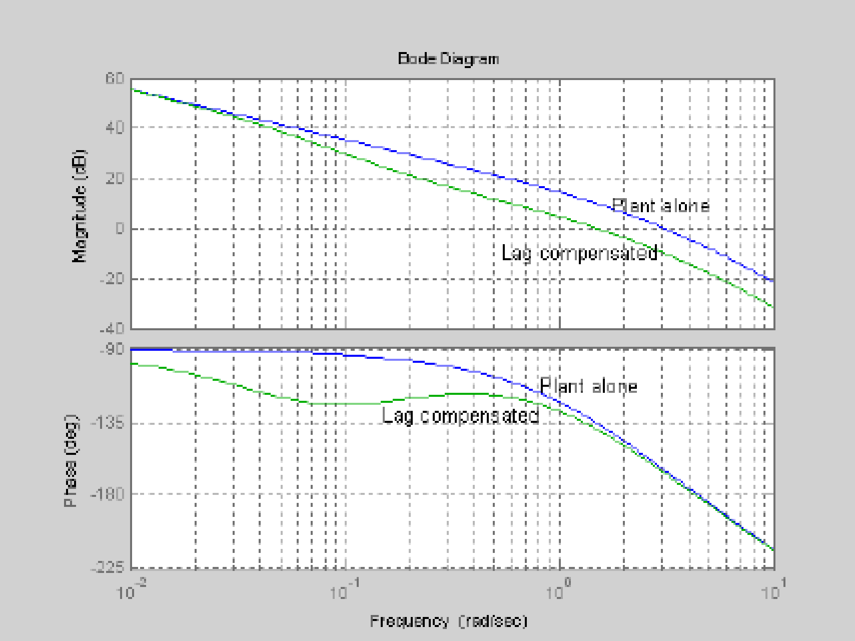


Figure 7.9 Bode diagrams for the plant alone and the lag compensated system

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

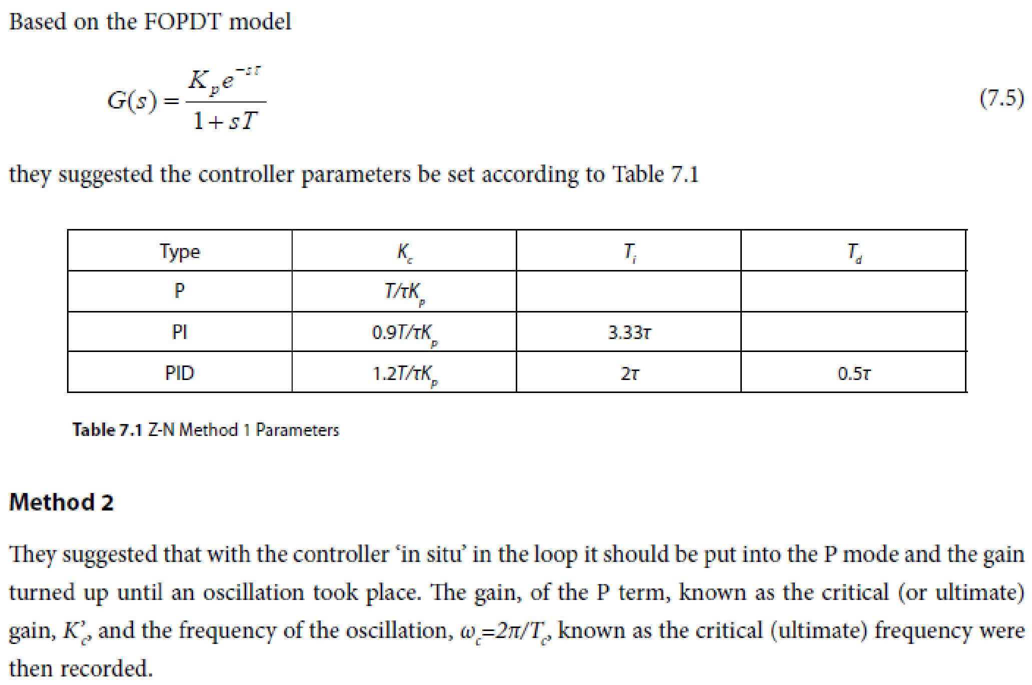
$$G_c(s) = K_c(1 + sT_d + [1/sT_i]) \tag{7.3}$$

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

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

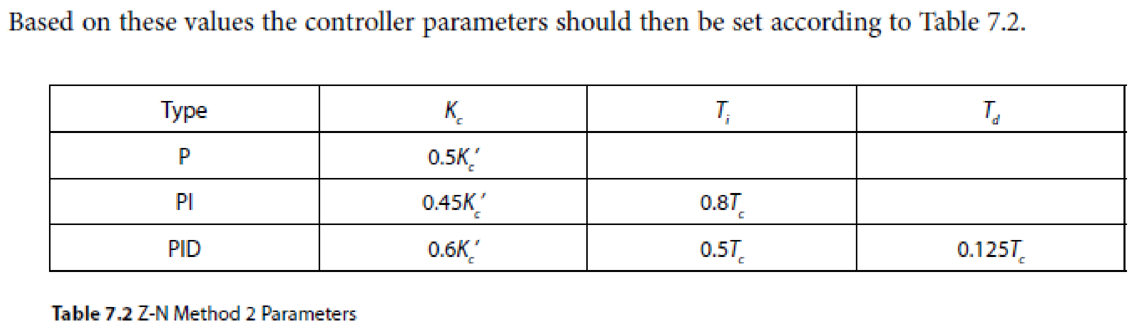
they suggested the controller parameters be set according to Table 7.1

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.

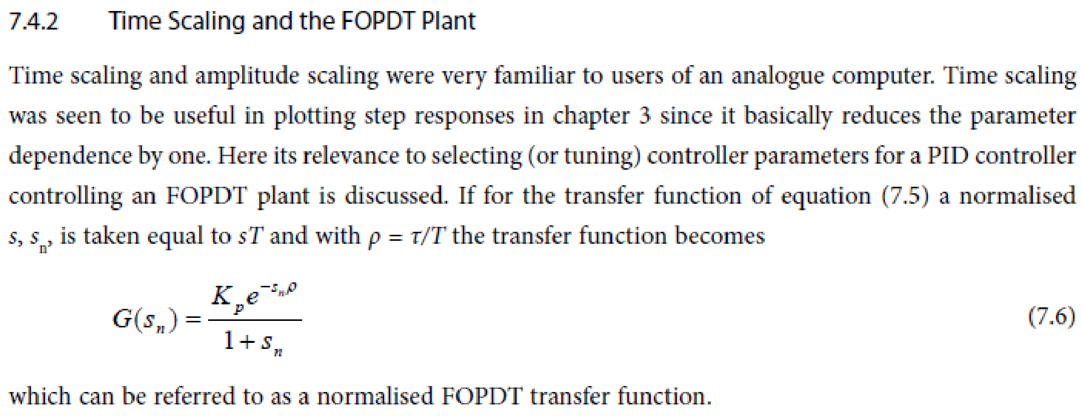


Based on these values the controller parameters should then be set according to Table 7.2.

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

Read 7.4.2 Time Scaling and the FOPDT Plant Page 81



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

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

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 $T_d = Tf_3(\rho)$.

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 $T_d = Tf_3(\rho)$.

	$f_1(\rho)$	$f_2(\rho)$	$f_3(\rho)$
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.

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.

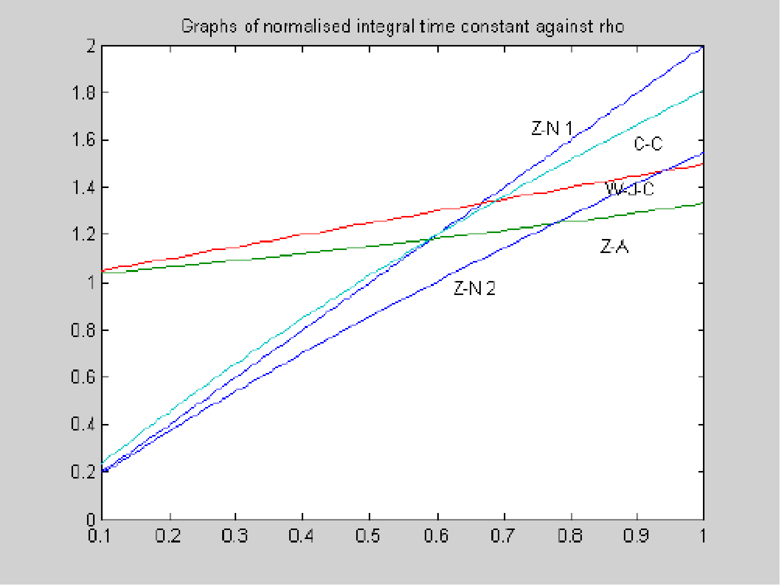


Figure 7.12 Graph of normalised integral time against rho

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)°.

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

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

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

$$G(s)=\frac{320(s+1)}{(s+0.5)(s+2)(s+4)(s+8)} \qquad (7.1$$

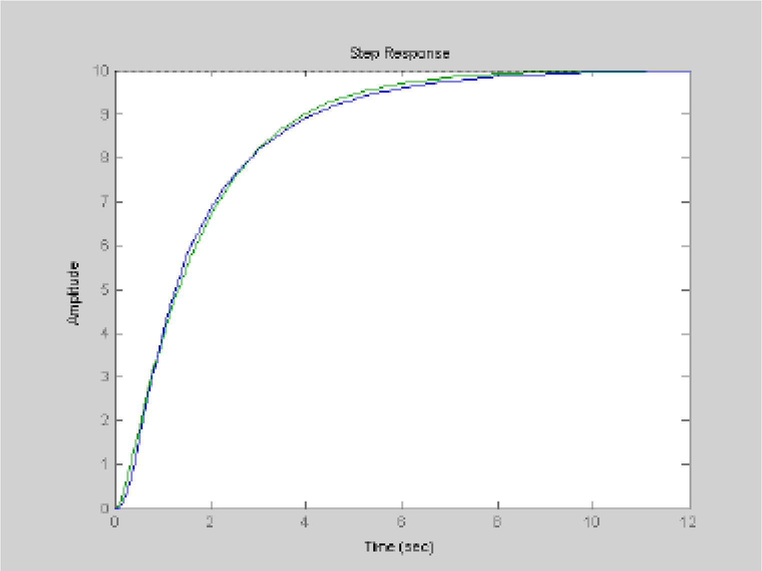
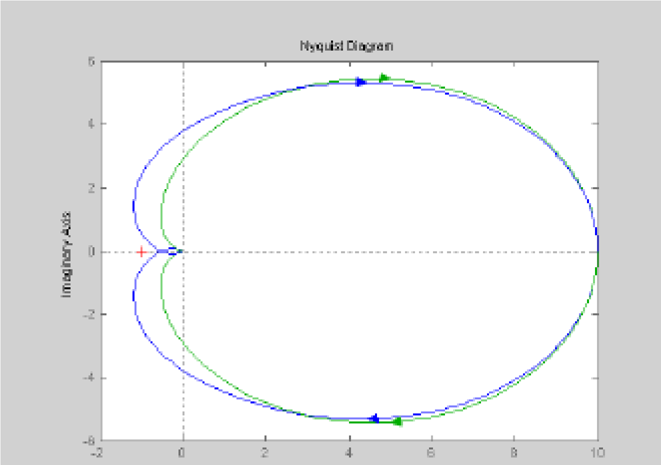


Figure 7.14 Step responses of *G*(*s*) and $G_s(s)$.



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.

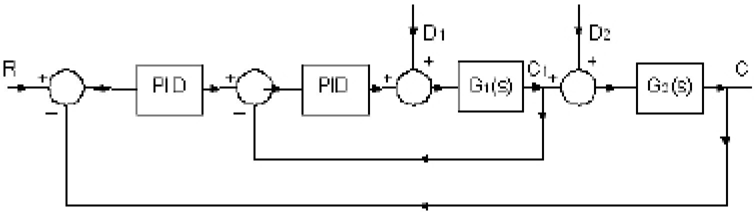


Figure 7.16 Block Diagram of Cascade Control.

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
$\lvert e(t)\rvert$	Integral absolute error – IAE
$t\lvert e(t)\rvert$	Integral time absolute error – ITAE
$t^2\lvert e(t)\rvert$	Integral time squared absolute error – IT²AE
$e^2(t)$	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

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

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

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 = $1/2K$. 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

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

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

$$I = (1 + \lambda^2 K^2)/2K \qquad (8.5)$$

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.

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.

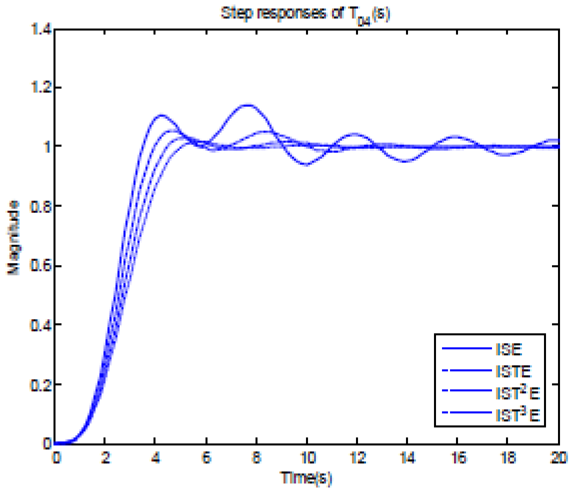


Figure 8.1 Step responses for $j = 4$

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

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

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.

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.

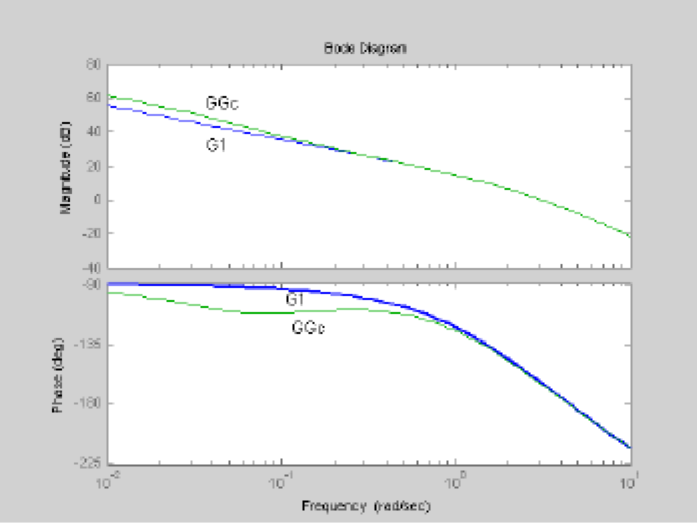


Figure 9.1 Bode diagrams for the example.

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

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

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

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

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

$$T(s) = \frac{K_cK}{s^3T + s^2 + sK_cKT_1 + K_cK} \tag{9.13}$$

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 **b**, *C* a $1 \times n$ row vector, often denoted by $\mathbf{c}^T$, 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 $e^{at}x(0)$ where $x(0)$ is the initial value of $x(t)$. Differentiating this expression gives

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 $e^{at}x(0)$ 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 $x(t) = e^{at}f(t)$, 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

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

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

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

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

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

10.7.1 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 $\mathrm{X} = (\mathrm{B}\ AB\ A^2B \ldots\ A^{n-1}B)$ 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

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

Read Chapter 11