EXTENSIONS TO THE ROOT LOCUS METHOD

by

S. E. Williamson, BSc, CEng, MIEE, MIERE

A thesis submitted for the degree of Doctor of Philosophy in the Faculty of Engineering, Department of Electronic and Electrical Engineering of the University of Surrey.

April 1971.
SUMMARY

Analytical conditions are derived which enable the root locus shapes of many simple systems to be obtained directly from the relative positions of their open loop poles and zeros. It is also shown that the pole-zero patterns of these systems can often be normalised in order to display all possible forms of their root loci on a single map. Many previously unknown locus shapes are given and these are considered in relation to the corresponding new forms of closed loop response.

A computer program is developed which enables root loci to be plotted for systems which include pure time delay or distributed lag. This extension to the method makes it possible to predict the closed loop responses of pure time delay systems directly from their root loci. It also enables these systems to be designed to achieve specified closed loop performance and provides a method of obtaining their open loop dynamics from measurements of closed loop response.

The speed of the new computer algorithm makes it feasible to use root locus methods for the design of high order multiloop systems. This improved capability is illustrated for aircraft control systems. A general pole-zero method is developed which enables control laws to be chosen for a desired aircraft response throughout the complete flight envelope.

The pure time delay capability of the program is then used to determine human pilot transfer functions from flight records of the pilot-aircraft response. Finally the results of this analysis are used to design improved forms of manual aircraft control systems.
ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Professor W. F. Lovering for his active interest in this work and for his constructive comments regarding its presentation. I would also like to thank the Hawker Aircraft Company and the Royal Aircraft Establishment for providing the opportunities to apply root locus methods to problems arising in the control and stability of aircraft.

Finally, I wish to thank Professor D. R. Chick for providing the time and facilities which enabled this work to be carried out in his Department.
### CONTENTS

<table>
<thead>
<tr>
<th>Summary</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>3</td>
</tr>
<tr>
<td><strong>CHAPTER 1</strong></td>
<td><strong>INTRODUCTION</strong></td>
</tr>
<tr>
<td><strong>CHAPTER 2</strong></td>
<td><strong>ANALYTICAL PROPERTIES OF THE ROOT LOCI OF SIMPLE SYSTEMS WHICH OCCUR FREQUENTLY IN ENGINEERING DESIGN</strong></td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>13</td>
</tr>
<tr>
<td>2.2 General Properties of Root Loci</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1 Defining equations</td>
<td>13</td>
</tr>
<tr>
<td>2.2.2 Salient features of root locus shapes</td>
<td>15</td>
</tr>
<tr>
<td>2.2.3 Invariant properties of pole-zero patterns</td>
<td>16</td>
</tr>
<tr>
<td>2.3 Relationships between the root locus shapes of simple systems and the relative positions of their open loop poles and zeros</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1 Introduction</td>
<td>17</td>
</tr>
<tr>
<td>2.3.2 The root loci of second order systems</td>
<td>18</td>
</tr>
<tr>
<td>2.3.3 Root loci of three pole systems</td>
<td>19</td>
</tr>
<tr>
<td>2.3.4 Third order systems with one zero</td>
<td>23</td>
</tr>
<tr>
<td>2.3.5 The root loci of four pole systems</td>
<td>27</td>
</tr>
<tr>
<td>2.4 The Root Loci and Closed Loop Responses of Systems in which the closed loop poles occur at or vertically in-line with the C of G of the open loop poles</td>
<td>32</td>
</tr>
<tr>
<td>2.4.1 Introduction</td>
<td>32</td>
</tr>
<tr>
<td>2.4.2 Zero sensitivity three pole systems</td>
<td>32</td>
</tr>
<tr>
<td>2.4.3 Third order system with a zero</td>
<td>36</td>
</tr>
<tr>
<td>2.5 Root Locus Shapes Associated with Specified forms of Closed Loop Response</td>
<td>40</td>
</tr>
<tr>
<td>2.5.1 Introduction</td>
<td>40</td>
</tr>
<tr>
<td>2.5.2 Root locus shapes required for third order systems</td>
<td>41</td>
</tr>
<tr>
<td>2.5.3 Four pole systems</td>
<td>50</td>
</tr>
<tr>
<td>2.6 Summary</td>
<td>53</td>
</tr>
</tbody>
</table>
CHAPTER 3 AUTOMATIC ROOT LOCUS PLOTTING

3.1 Introduction

3.2 Review and Criticism of Present Techniques
  3.2.1 Analogue methods
  3.2.2 Digital computer programs
  3.2.3 Special Purpose Computers

3.3 The Author's New Search Vector Method
  3.3.1 Introduction
  3.3.2 Basic Concept of the algorithm
  3.3.3 Refinements to improve speed and accuracy

3.4 Comparison with classical techniques
  3.4.1 Introduction
  3.4.2 Computation of the slope of the chord joining two points on the locus
  3.4.3 Analytical expression for the heading correction
  3.4.4 Comparison with the author's algorithm

3.5 Modification of the basic algorithm to deal with particular locus singularities
  3.5.1 Break points
  3.5.2 Unusual locus properties in the region of certain zeros

3.6 Plotting Constant Gain Contours
  3.6.1 Introduction
  3.6.2 Plotting procedure

3.7 The Root Loci of Systems which include Pure Time Delay
  3.7.1 Phase correction representation of pure time delay
  3.7.2 Extension of the author's algorithm to plot the primary branches of the locus
  3.7.3 Location of the extra branches of the locus
  3.7.4 Constant gain contours

3.8 Summary
CHAPTER 4  THE USE OF ROOT LOCI FOR THE ANALYSIS OF CLOSED LOOP PURE TIME DELAY SYSTEMS

4.1 Introduction

4.2 Methods of Approximating the Closed Loop System
  4.2.1 Statement of the problem
  4.2.2 Methods of approximating the time delay
  4.2.3 The author's method using root locus plots

4.3 Prediction of Closed Loop Transient Response
  4.3.1 Introduction
  4.3.2 Truncated Maclaurin Series approximations
  4.3.3 Use of Pade functions to represent the delay
  4.3.4 The author's root locus method
  4.3.5 Comparison of results

4.4 Identification of the dynamics of closed loop linear systems which include pure time delay
  4.4.1 Introduction
  4.4.2 Determination of systems transfer function from its measured impulse response
  4.4.3 Extension to pure time delay systems

4.5 Methods of compensating for the effects of pure time delay
  4.5.1 Use of minor loop feedback
  4.5.2 Inverse root locus methods

4.6 The use of inverse root locus methods to determine the optimum step response parameters of a pure time delay third order type I system
  4.6.1 Optimum performance without time delay
  4.6.2 Optimum performance including the effect of pure time delay

4.7 Summary

88  88  88  91  93  93  94  94  96  99  99  99  100  101  105  105  108  109  109  111  117
CHAPTER 5 APPLICATION OF ROOT LOCUS TECHNIQUE TO THE CONTROL AND STABILITY OF AIRCRAFT

5.1 Introduction 118

5.2 Aircraft Transfer Functions 118
5.2.1 Generalised equations of motion 118
5.2.2 Stick fixed stability 119
5.2.3 Method of including generalised auto pilot control laws 122

5.3 Longitudinal Channel 126
5.3.1 Stability Polynomials for elevator control of pitch attitude and height 126
5.3.2 Root locus equations and order of loop closure 128

5.4 Design of an autopilot to control pitch attitude and height 129
5.4.1 General requirements and basic system 129
5.4.2 Choice of autopilot control laws for a typical aircraft 131

5.5 Summary 139

CHAPTER 6 ROOT LOCUS DETERMINATION OF IN-FLIGHT HUMAN PILOT TRANSFER FUNCTIONS 140

6.1 Introduction 140

6.2 Formulation of a mathematical model for the human pilot 140
6.2.1 Linear Continuous Pilot transfer functions 141
6.2.2 Sampled data model 142
6.2.3 Comparison of discrete and continuous pilot model for predicting human pilot control of unstable VTOL in hover. 143

6.3 Determination of Pilot Transfer Function for a specific flight manoeuvre 148
6.3.1 Measurement of pilot-aircraft response 148
6.3.2 The effect on the closed loop performance of varying the parameters of the pilot transfer function 150
6.3.3 Determination of the pilot transfer function 151
6.4 Pilot induced stability during elevation tracking

6.4.1 Prediction of the oscillation frequency using root locus

6.4.2 Verification of the pilot transfer function from measured flight records

6.5 Summary

CHAPTER 7 ROOT LOCUS OPTIMISATION OF HUMAN PILOT CONTROL SYSTEMS

7.1 Introduction

7.2 Manual Control of Aircraft Flight Path

7.2.1 The pilot control loop

7.2.2 Flight vector response to movement of the elevator

7.2.3 The tracking problem

7.3 Methods of Improving Performance

7.3.1 Modification to the Aircraft Dynamics

7.3.2 Shaping and deflection of the aiming symbol in the head-up display

7.4 Human Pilot Control of Apparent Flight Vector

7.4.1 Apparent flight vector response of the aircraft movement of the elevator

7.4.2 Root locus optimisation of the control loop

7.4.3 The effect on performance of using non-linear stick to elevator gearing

7.5 Simulator verification of the root locus design

7.6 Summary

CHAPTER 8 DISCUSSION AND CONCLUSIONS

APPENDIX 1 THE AUTHOR'S DIGITAL SIMULATION LANGUAGE

A1 Introduction

A2 Choice of Integration Algorithm

A3 Transfer Function Simulation
CHAPTER 1

INTRODUCTION

Although the stability of closed loop systems can be determined from Nyquist diagrams, Bode Plots, or Nichols Charts, these techniques deal only with the sine wave response of the system. They fail to give any direct indication of the response to other forms of deterministic input and it is often difficult to see how the system should be compensated in order to achieve a desired improvement in closed loop performance.

The root locus method is an alternative design technique which was first introduced by Evans in 1948. It is based on the fact that the closed loop transient response of a system depends on the poles and zeros of its characteristic equation \( 1 + KG(s) = 0 \).

By writing this equation in the form \(-1 = KG(s)\) the variation of the closed loop pole positions with loop gain \(K\), can be plotted as a locus in the complex plane. This is achieved by searching for points in the plane where the net phase angle of \(KG(s)\) is \(180^\circ\).

Having produced a root locus, the designer can readily see the different types of transient response which are available for the closed loop system. He can also obtain gain and phase margins for a particular value of \(K\) and can determine the frequency response of the system by direct measurement to points on the \(j\omega\) axis.

If all the possible forms of transient response are unacceptable he can then reshape the locus with compensating poles and zeros in order to achieve a closed loop pole zero pattern which is known to have the required performance. Until recently root loci have been
plotted manually, which is a very tedious and time consuming process. There has also been a lack of analytical data showing how the general shape of the locus depends on the relative positions of its poles and zeros. Both these factors have seriously limited the use of root locus methods in engineering design and many of the unique capabilities of the technique have not yet been fully exploited.

In the present work the author derives analytical conditions which enable the root loci of simple systems to be sketched directly from the relative positions of their poles and zeros. Many locus shapes are given which were previously unknown and the significance of transitional forms of the locus are considered in relation to closed loop performance. It is also shown that the pole-zero patterns of many simple systems can be normalised in order to display all possible forms of their loci on a single map.

The various methods which have been proposed for automatic root locus plotting are all shown to have very serious limitations. In many cases it has been found that they were unable to plot even simple loci. From a study of the reasons why these methods fail to plot root loci, the author has developed a new reliable computer program which can also deal with the root loci of systems which include pure time delay or distributed lag.

The ability to plot these loci enables the closed loop performance of pure time delay systems to be obtained directly from the primary and extra delay branches of the locus. It also makes possible the identification of the open loop dynamics of pure time delay systems from measurements of their closed loop response. The speed and accuracy of the new plotting procedure now makes it practical to
use root locus methods for the design of high order multiloop systems. To illustrate this extension to the technique, the author has developed a general pole-zero method for the design of aircraft control systems.

This new method coupled with the ability to plot pure time delay loci has now made it possible to identify human pilot transfer functions from measured flight records of the pilot-aircraft response.

The determination of a reliable human pilot transfer function has enabled the author to design improved manual aircraft control systems. And in particular to solve the problem of aircraft oscillation caused by the pilot during flight vector elevation tracking.

Although this work has been carried out on an aircraft control loop, the original techniques which have been developed could readily be applied to the solution of similar man-machine control problems arising in other fields of cybernetics.
2.1 Introduction

The root locus technique is now widely used for the design and synthesis of control systems. Loci are normally plotted by means of a spirule which is used to locate the points in the complex plane where the net phase angle is 180 degrees. The task of plotting the locus is simplified by applying a standard set of rules (1) which enable the designer to obtain the asymptotes, axis crossings, and the set-off directions from the open-loop poles.

Unfortunately these rules do not show how the many different forms of the locus depend on the relative positions of the open loop poles and zeros. Unless the designer has some prior knowledge of the general shapes that can arise for a particular system, then plotting its locus can often be a very tedious and time consuming process. This is especially true when the open-loop poles and zeros are such that small changes in their relative positions cause a radical change in the shape of the locus.

Although many practical systems can be represented by quite simple pole-zero patterns, very little analytical information has been available showing how the locus shape depends on the relative positions of the poles and zeros.

2.2 General Properties of Root Loci

2.2.1 Defining equations

The root locus of a unity negative feedback control system having an open loop transfer function KG(s) is defined by the equation

\[-1 = KG(s)\]
which is equivalent to the two conditions

\[ \arg G(s) = (2r + 1) \pi \quad 2.2.2 \]

and

\[ |G(s)| = 1/K \quad 2.2.3 \]

The root locus plot is therefore a graphical solution of the characteristic equation \( 1 + KG(s) = 0 \) and shows how the closed loop poles of the system depend on the loop gain \( K \).

The shape of the locus can be plotted from the angle condition by searching for points on the complex plane where the net phase angle \( KG(s) \) is \( 180^\circ \), while the closed loop poles for a particular value of \( K \) are obtained from the points on the locus which also satisfy the modulus condition equation (2.2.3).

There also exists a general cartesian equation to a root locus which was first discovered in 1958 by Mikhailov (2).

He has shown that by writing equation (2.2.1) in the form

\[ -1 = K \frac{Z(s)}{P(s)} \quad 2.2.4 \]

it is possible to obtain a general cartesian equation for the shape of the root locus in terms of the derivatives of \( Z(x) \) and \( P(x) \) with respect to \( x \).

Applying his expression to a third order system for example, the general cartesian equation of the locus is given by

\[ y[P(x) Z'(x) - Z(x) P'(x)] - \frac{1}{3!} [P(x) Z'''(x)] + \frac{P'(x) Z''(x)}{2!} - \frac{P''(x) Z'(x)}{2!} - \frac{P'''(x) Z(x)}{3!} = 0 \]

\[ 2.2.5 \]

The Mikhailov expression greatly reduces the algebraic manipulation required to obtain the cartesian equation and has been particularly useful for evaluating the analytical properties of simple loci.
2.2.2 Salient features of root locus shapes

Certain features of a root locus are of particular interest and are normally examined prior to making a general sketch of its shape.

The centre of gravity

Writing equation 2.2.1 in the form

\[ -1 = k \sum_{i=1}^{i=M} (s + z_i) / \sum_{j=1}^{j=N} (s + p_j) \]

the centre of gravity of the open loop pole-zero pattern is given by

\[ \chi = \frac{1}{(N-M)} \left[ \sum_{i=1}^{i=M} z_i - \sum_{j=1}^{j=N} p_j \right] \]

This point on the real axis is the position of the \((N-M)\) th order pole which is equivalent to the system for large values of \(S\).

Asymptotic behaviour of the locus

The asymptotic shape of the locus will therefore be defined by the \((N-M)\) straight line branches from the equivalent pole at the centre of gravity. These branches can be readily constructed and radiate from the \((N-M)\) th order pole at angles of \((2r+1)/(N-M)\) where \(r = 0, 1, 2 \ldots N-M-1\).

Break Points

Most root loci have branches which are partly real and partly complex. The transition occurs at a coincident pair of closed loop poles and is known as a break point.

Break points can therefore be located directly from the Mikaililov equation of the locus from the values of \(\chi\) at which the complex branches cross the real axis.

Alternatively they can be located from the fact that at a break point \(K\) reaches a stationary value with respect to \(\chi\).
Hence the break points can also be obtained from the solution of

\[ \frac{\partial K}{\partial x} = 0 \]  

(2.2.8)

The exit points are in fact the maxima while the minima of the

function \( K(x) \) occur at the re-entry points. The nature of the break

point can therefore be determined from the sign of \( \frac{\partial^2 K}{\partial x^2} \).

Many loci have more than one break point and the pole-zero

pattern for which they coincide normally gives rise to a transitional

form of the locus. Since the remaining basic shapes of the locus arise

between the transitional forms, then these existence conditions are

very useful for relating the general shape of the locus to the

relative positions of the poles and zeros.

2.2.3 Invariant properties of pole-zero patterns

The general shape of a root locus depends only on the relative

positions of the open loop poles and zeros. Hence any property of the

closed loop pole-zero pattern which is independent of loop gain \( K \), will

have direct application in determining the shape of the locus.

The position of the centre of gravity

The closed loop poles of the system for a particular value of \( K \)

are the roots of the characteristic equation

\[ \prod_{j=1}^{N} (s + p_j) + K \prod_{i=1}^{M} (s + Z_i) = 0 \]  

(2.2.9)

which can also be written as

\[ \prod_{j=1}^{N} (s + \alpha_j) = 0 \]  

(2.2.10)

For systems in which \( (N-M) > 2 \) the open and closed loop pole positions

are therefore related by the equation \( \sum \alpha_j = \sum p_j \).
Hence from equation (2.2.7) it can be seen that the centre of gravity of the closed loop poles is independent of $K$ and remains fixed at the centre of gravity of the open loop poles.

The product of the vectors from the poles to one of the zeros

Transferring the origin of co-ordinates to the zero at $s = -Z_k$ by means of the substitutions $s' = s + Z_k$, $p_j' = p_j - Z_k$, and $Z_i' = Z_i - Z_k$, the characteristic equation for the closed loop poles becomes

$$\prod_{j=1}^{N} (s' + p_j') + K \prod_{i=1}^{i=k-1} (s' + Z_i') \prod_{i=k+1}^{i=M-k} (s' + Z_i') = 0 \quad 2.2.11$$

Hence provided $(N-M) \geq 1$, the product of the vectors from the closed loop poles to the zero at $s = -Z_k$ will be independent of $K$ and is given by $p_j'$. This is a particularly useful invariant property of the closed loop pole-zero pattern and has enabled the author to directly establish the root locus topology of many simple systems.

2.3 Relationships between the root locus shapes of simple systems and the relative positions of their open loop poles and zeros

2.3.1 Introduction

While Truxal's Rules give a clear indication of the asymptotic shape of a locus, they fail to show how its many different general forms depend on the relative positions of the poles and zeros. Simple analytical relationships between locus shape and pole-zero pattern do, however, exist for many systems up to fourth order, and enable the designer to obtain a very rapid sketch of the root locus.

Attempts to establish the root locus topology of higher order systems have been made by Teodorchik and Bendrikov (3) and more recently by Power (4). Although their work undoubtedly contributes to our knowledge of the geometry of higher plane curves, it is difficult to see how it can be used
to assist in the sketching of a root locus. In general root loci of systems above fourth order should be plotted using a computer as explained in Chapter 3.

The author has, however, found that there are some simple analytical results which apply to important special cases such as type 3 systems, and these are given in a supporting publication ref (5).

2.3.2 The root loci of second order systems

The general equation for the root locus of a second order system can be written in the form

\[
-1 = \frac{K(s + a)(s + b)}{(s + c)(s + d)} = K \frac{Z(s)}{P(s)}
\]

(2.3.1)

where the zeros a and b and/or the poles c and d, can be either real or complex.

In this case the Miklailov general expression for the cartesian equation of the locus reduces to

\[
y [P(x) Z'(x) - Z(x) P'(x)] + y^3 \left[ \frac{P'(x) Z''(x)}{2!} - \frac{Z'(x) P''(x)}{2!} \right] = 0
\]

(2.3.2)

From which

\[
y = 0 \text{ or }
\]

\[
[ (c+d) - (a+b)] x^2 + \left[ (c+d)-(a+b) \right] y^2 + 2 (cd-ab) x
\]

\[
- (a+b) cd - (c+d) ab = 0
\]

(2.3.3)
Hence the complex root locus branches of a second order system form part of a circle whose centre and radius can be obtained from equation (2.3.3). In the case of a system with only one zero, the centre and radius can also be determined directly from the invariant vector product property of root loci given in section (2.2.3).

Analytical relationships between locus shape and pole zero pattern have been established for many second order systems and these are summarised in a series of supporting publications see refs (5, 6, 7).

2.3.3 Root Loci of Three Pole Systems

In this case all possible shapes of the locus can be investigated by considering the general equation of the type 0 system. This can be written in normalised form as follows:

\[
-1 = \frac{K}{(s+A)(s^2+2\zeta s+1)} \tag{2.3.4}
\]

The case of three real poles then corresponds to \( \zeta > 1 \), while the type I system is the special case \( A = 0 \). There are 9 different general shapes of root locus for the type 0 system and these are shown in Figs (2.1) and (2.2).

It should be noted that the following analysis which was used to establish these shapes also includes the 3 basic forms of the locus previously discovered by Meadows (8) for the special case \( (A = 0, \zeta > 1) \).

**Cartesian equation**

The Miklailov general expression in this case reduces to

\[
y [P(x) Z'(x)] - y^3 \left[ \frac{Z(x)}{s!} \right] = 0 \tag{2.3.5}
\]

from which

\[
y = 0 \tag{2.3.6}
\]

or

\[
y^2 = 3x^2 + 2(A+2\zeta) x + 1 + 2A \zeta \tag{2.3.7}
\]
Real axis break points

An initial subdivision of locus shapes can be obtained by determining the pole patterns which give rise to complex branches intersecting the real axis.

For this to occur the roots of the equation

\[ 3x^2 + 2(A + 2\zeta)x + 1 + 2AC = 0 \]  

must be real and also satisfy the equation of the locus. The pole positions are therefore restricted to values of \( A \) and \( \zeta \) for which

\[ 4\zeta > A + \sqrt{3(4-A^2)} \]

Hence the locus has two break points for \( \zeta < 1 \) (see Fig 2.1 f), and one break point for \( \zeta > 1 \) (see Fig 2.2 a).

Transitional shapes of the locus

The next step towards identifying the many different forms of the locus is to determine the pole patterns which give rise to the transitional shapes. These occur when two real axis break points coincide, i.e. when

\[ 4\zeta = a \pm \sqrt{3(4-A^2)} \]

Under these conditions the cartesian equation of the locus reduces to a pair of straight lines which pass through the centre of gravity of the poles, i.e.

\[ y^2 = 3 \left[ x + \frac{2\zeta + A}{3} \right]^2 \]

The locus then coincides with its asymptotes as shown in Figs (2.1 b,e) and Fig (2.2 c).
Fig 2.1 The Root Loci of a Three Pole Type 0 System for $\zeta < 1$. 

(a) $\zeta \omega_n < a$

$\zeta = \frac{a - \sqrt{3(\zeta \omega_n^2 - a^2)}}{\zeta \omega_n}$

$\omega_1 = \sqrt{\omega_n(\omega_n + 2\zeta a)}$

$K_m = 2\zeta \omega_n(a^2 + 2\zeta a_0 + \omega_n^2)$

$P_1 = -\frac{1}{2}(2\zeta \omega_n + a)$

(b) $\zeta \omega_n > a$

$\zeta = \frac{a + \sqrt{3(\zeta \omega_n^2 - a^2)}}{\zeta \omega_n}$

$\omega_1 = \sqrt{\omega_n(\omega_n + 2\zeta a)}$

$K_m = 2\zeta \omega_n(a^2 + 2\zeta a_0 + \omega_n^2)$

$P_1 = -\frac{1}{2}(2\zeta \omega_n + a)$

(c) $\frac{K}{s+\alpha}[(s^2 + 2\zeta \omega_n s + \omega_n^2)]$

$\zeta \omega_n < a$

$\zeta = \frac{a - \sqrt{3(\zeta \omega_n^2 - a^2)}}{\zeta \omega_n}$

$\omega_1 = \sqrt{\omega_n(\omega_n + 2\zeta a)}$

$K_m = 2\zeta \omega_n(a^2 + 2\zeta a_0 + \omega_n^2)$

$P_1 = -\frac{1}{2}(2\zeta \omega_n + a)$

(d) $\frac{K}{s+\alpha}[(s^2 + 2\zeta \omega_n s + \omega_n^2)]$

$\zeta \omega_n > a$

$\zeta = \frac{a + \sqrt{3(\zeta \omega_n^2 - a^2)}}{\zeta \omega_n}$

$\omega_1 = \sqrt{\omega_n(\omega_n + 2\zeta a)}$

$K_m = 2\zeta \omega_n(a^2 + 2\zeta a_0 + \omega_n^2)$

$P_1 = -\frac{1}{2}(2\zeta \omega_n + a)$

(e) $\frac{K}{s+\alpha}[(s^2 + 2\zeta \omega_n s + \omega_n^2)]$

$\zeta \omega_n < a$

$\zeta = \frac{a - \sqrt{3(\zeta \omega_n^2 - a^2)}}{\zeta \omega_n}$

$\omega_1 = \sqrt{\omega_n(\omega_n + 2\zeta a)}$

$K_m = 2\zeta \omega_n(a^2 + 2\zeta a_0 + \omega_n^2)$

$P_1 = -\frac{1}{2}(2\zeta \omega_n + a)$

(f) $\frac{K}{s+\alpha}[(s^2 + 2\zeta \omega_n s + \omega_n^2)]$

$\zeta \omega_n > a$

$\zeta = \frac{a + \sqrt{3(\zeta \omega_n^2 - a^2)}}{\zeta \omega_n}$

$\omega_1 = \sqrt{\omega_n(\omega_n + 2\zeta a)}$

$K_m = 2\zeta \omega_n(a^2 + 2\zeta a_0 + \omega_n^2)$

$P_1 = -\frac{1}{2}(2\zeta \omega_n + a)$

$P_2 = -\frac{1}{2}(2\zeta \omega_n + a) + \sqrt{4\zeta^2 \omega_n^2 + a^2 - 3\omega_n^2 - 2\zeta \omega_n a}$

$P_3 = -\frac{1}{2}(2\zeta \omega_n + a) - \sqrt{4\zeta^2 \omega_n^2 + a^2 - 3\omega_n^2 - 2\zeta \omega_n a}$
\[
\frac{K}{(s+a)(s+b)(s+c)}
\]

\[
\omega_1 = \sqrt{a^2 + b^2 + c^2}
\]

\[
K_m = (a+b)(a+c)(b+c)
\]

\[
P_1 = -\frac{1}{2}(a+b+c)
\]

\[
P_2 = -\frac{1}{2}(a+b+c - \sqrt{a^2 + b^2 + c^2 - (ab+bc+ac)})
\]

Fig 2.2 The Root Loci of a Three Pole Type 0 System for \( \zeta \geq 1 \).
The remaining locus shapes then occur for pole positions which lie on either side of the transitional forms of the locus and are shown in Figs 2.1 and 2.2. Further assistance in sketching a particular general shape is also given in the form of analytical expressions for the real axis break points, imaginary axis crossings and the maximum gain for closed loop stability.

The author has also obtained the root locus shapes associated with many other commonly occurring type 0 systems and these are given in a supporting publication ref (5).

**Three pole type I systems**

The three pole Type I system arises very frequently in practice and is for example the basic pole pattern of a simple remote position control system. The restriction of placing one of the poles at the origin greatly reduces the number of locus shapes that can arise compared with the type 0 system. It also gives very much simpler analytical relationships for the relative pole positions corresponding to transitional forms of the locus.

There are in fact only four different general locus shapes in this case and these are as shown in Fig 2.3.

### 2.3.4 Third order systems with one zero

The general equation of the locus in this case is

$$-1 = \frac{K(s + b)}{s(s^2 + 2ζs + 1)}$$

and the introduction of a zero into the open loop system gives rise to 16 different general shapes of the locus, details of which are given in the author's supporting publication ref (6).
Fig 2.3 The Root Loci of a Three Pole Type I System.
Perhaps the most important practical system in this group is the type 2 system whose locus is defined by the equation

\[ -1 = \frac{K(s+b)}{s^2 (s+a)} \]  

2.3.12

It arises for example in the phase advance stabilisation of a V.T.O.L. aircraft and is also the basic pole-zero pattern for the proportional plus integral compensation of a simple position control system.

The cartesian equations for the locus in this case are

\[ y = 0 \]  

2.3.13

or

\[ y^2 = \frac{2x^3 + (a + 3b) x^2 + 2abx}{(b-a-2x)} \]  

2.3.14

Hence provided \( x \neq \frac{1}{2}(b-a) \) i.e. the centre of gravity of the pole-zero pattern, then the complex branches of the locus cross the real axis when

\[ x = 0, \text{ or } x = -\frac{1}{2}[a+3b \pm \sqrt{a^2 + 9b^2 - 10ab}] \]  

2.3.15

The break points will coincide and give rise to a transitional locus shape when

\[ a^2 - 10ab + 9b^2 = 0 \]  

2.3.16

From which \( a = b \) or \( a = 9b \).

Since \( a \neq b \), then the 9:1 ratio applies in this case and the closed loop system can have a triple pole at \( x = -3b \).

Four general locus shapes arise for this system and these are shown in Fig 2.4.
Fig 2.4 Root Loci of a Third Order Type 2 System with one real zero.
Having successfully established the relationship between the locus shape of simple systems and the relative positions of their poles and zeros, it was decided to investigate the possibility of displaying all possible root locus shapes of a system on a single map. The different general forms of the locus would then be associated with various regions of the map, and the transitional locus shapes would occur on the boundaries between these regions.

This new concept in root locus theory can be applied to any system in which the relative pole-zero positions can be expressed in terms of two parameters. The main point which seems to have been missed by other workers in this field is the fact that the shape of a root locus is independent of its origin and scale. This means that any pole or zero can be arbitrarily chosen as the origin and the remaining pole-zero positions can then be normalised in order to minimise the number of independent variables.

**Normalised equations**

The root locus equations of any four pole system which has two or more real poles can be expressed in the form

\[-I = \frac{K}{S(S+A)(S^2+2\zeta S+1)}\]  \hspace{1cm} 2.3.17

From which the corresponding cartesian equations are given by

\[Y = 0\]  \hspace{1cm} 2.3.18

and

\[Y^2 = \frac{4X^2+3(A+2\zeta)X^2+2(1+2A\zeta)X+A}{(4X+A+2\zeta)}\]  \hspace{1cm} 2.3.19

Hence all possible shapes of the locus can therefore be investigated in terms of the normalised open loop pole parameters A and \(\zeta\).
In the case of a system with two pairs of complex poles the locus equation can be written in the form

\[-1 = \frac{K}{(S^2+\omega_r^2)(S^2+2\zeta S+1)}\]  \hspace{1cm} 2.3.20

and the cartesian equation is given by

\[\gamma^2 = \frac{2X^3+2\zeta X^2+(1+\omega_r^2)X+\zeta \omega_r^2}{(2X+\zeta)}\] \hspace{1cm} 2.3.21

The two normalised parameters are then \(\omega_r\) and \(\zeta\).

**Transitional locus shapes**

Detailed investigation of the roots of the cartesian equation gives the relationships between the normalised parameters corresponding to transitional forms of the locus. When these relationships are plotted in the \((A, \zeta)\) and \((\omega_r, \zeta)\) planes they form boundaries which represent the relative pole positions where the locus undergoes a radical change in its general shape.

Once these boundaries had been established, the remaining locus shapes were readily obtained and all possible forms of the locus were then displayed on the two maps shown in Figs 2.5 and 2.8. As a further aid to sketching each general form of the locus, analytical expressions were also produced for break points, axis crossings, and the maximum values of \(K\) for closed loop stability. Examples of this data for some of the shapes in the \((A, \zeta)\) map are shown in Figs 2.6 and 2.7.

For full details of the analysis used to establish the two normalised maps, the reader is referred to the author's recent paper "The Root Loci of Four Pole Systems" ref (9).
Fig 2.5 The relationship between the root locus shapes of a four pole system and the values of the normalised parameters A and $\xi$. 
\[
P_1 = \left( -\frac{A}{2}, 0 \right); \quad P_2 = \left( -\frac{A + \sqrt{(A^2 - 2)}}{2}, 0 \right); \quad P_3 = \left( -\frac{A - \sqrt{(A^2 - 2)}}{2}, 0 \right);
\]
\[
P_4 = -\frac{A}{2} \left[ \sqrt{(2 - A^2)} \right] \quad P_5 = -\frac{A}{2}, -\sqrt{(2 - A^2)}.
\]

Fig 2.6 Loci for the symmetry condition \( A = 2\zeta \).

\[
P_1 = -\frac{(A + 2\zeta)}{4}, 0; \quad P_2 = -\left[ \frac{2\zeta(1 + A^2 + 2\zeta) - 5A}{3A^2 - 4A\zeta + 12\zeta^2 - 8} \right], 0;
\]
\[
P_3 = -\frac{9(A + 2\zeta)^3 - 64\zeta(2 + 4\zeta - 12\zeta^2)}{4(3A^2 - 4A\zeta + 12\zeta^2 - 8)}, 0; \quad P_4 = -\left( \frac{A + \sqrt{(A^2 - 2)}}{2} \right), 0;
\]
\[
P_5 = -\left( \frac{A - \sqrt{(A^2 - 2)}}{2} \right), 0.
\]

Fig 2.7 Locus shapes for \( \zeta \) in the range \( \sqrt{3}/2 \geq \zeta \geq \sqrt{2} \).
Fig 2.8 Variation of locus shape with $\omega_r$ and $\zeta$. 
2.4 The Root Loci and Closed Loop Responses of Systems in which the Closed Loop Poles occur at or vertically in-line with the centre of gravity of the open loop poles

2.4.1 Introduction

Many previously unknown root locus shapes were given in section 2.3, and these in turn give rise to the possibility of many new forms of pole-zero pattern for the closed loop systems. Of particular interest are those locus shapes which enable the designer to place the closed loop poles at or vertically in-line with the centre of gravity of the open loop poles. A selection of these locus shapes is considered and the significance of the in-line pole distribution is discussed in relation to the transient response of the closed loop system.

The time responses given in this thesis were obtained using a new digital simulation language developed by the author. Details of the structure of the language together with its extension to the simulation of pure time delay systems is given in Appendix No. 1.

2.4.2 Zero sensitivity three pole systems

An early attempt to connect root locus shape with closed loop performance was made by Hannoc Ur (10) who defined closed loop sensitivity to gain change as

$$ S_K^G = \frac{1}{G(s)} \frac{dy}{dx} $$

From this definition the sensitivity is zero at non-singular points on the locus where the gradient is zero.

Although this definition has not been widely adopted in systems analysis, the author has found that closed loop poles placed at zero gradient positions on the locus often give rise to very interesting forms of closed loop response.
Consider for example the case of a Three Pole Type 0 system.

From the root loci of this system which are given in Figs 2.1 and 2.2, it can be seen that the zero gradient condition arises for the case \( \zeta > a \) and \( 4\zeta < a + \sqrt{3(4-a^2)} \) (Fig 2.1c).

Under these conditions the complex branches have a minimum point which occurs when

\[
\frac{dy}{dx} = 6x + 2(a + 2\zeta) = 0 \tag{2.4.2}
\]

From which

\[
x = -\frac{1}{3} (a + 2\zeta) \tag{2.4.3}
\]

the corresponding value of \( y \) is given by

\[
y = \left[ 1 - \frac{4\zeta^2 + a^2 - 2a\zeta}{3} \right]^\frac{1}{2} \tag{2.4.4}
\]

Hence the minimum point is situated vertically above the centre of gravity of the open loop poles.

Since the centre of gravity of the closed loop poles remains invariant in this case, then the third closed loop pole will be located on the real axis at \( x_3 \) given by

\[
x_3 = 2 \left[ \frac{a + 2\zeta}{3} \right] = -(a + 2\zeta) \tag{2.4.5}
\]

From which

\[
x_3 = -\frac{1}{3} [a + 2\zeta] \tag{2.4.6}
\]

Hence all the closed loop poles are vertically in-line.

To achieve this closed loop pole distribution the gain \( K \) must be set to a value \( K_0 \) given by

\[
K_0 = \frac{2}{3} (\zeta - a) \left[ 1 - \frac{(4\zeta - a)(2\zeta + a)}{9} \right] \tag{2.4.7}
\]
when the complex closed loop poles are aligned with the third pole at
the centre of gravity, the closed loop transfer function can be
expressed in the form
\[
\frac{C}{R}(s) = \frac{K_0}{(s+a)(s^2+2\zeta\omega_n s+\omega_n^2)}
\]
where
\[
a = \frac{a+2\zeta}{3} \quad \text{and} \quad \omega_n^2 = 1 - \frac{2}{9}(4\zeta-a)(\zeta-a)
\]

The closed loop impulse response can then be expressed directly
in terms of the root locus pole positions shown in Fig 2.9(b) and is
given by
\[
c(t) = \frac{|P_0P_1|^2 |AP_1| e^{-|OA|t}}{|P_1P_2|^2} (1-\cos (P_2P_1 t))
\]

Hence the in-line closed loop pole distribution gives rise to a non-
negative impulse response, examples of which are shown in Fig 2.10.
The limiting case of the non-negative response occurs when
\[4\zeta = a + \sqrt{3(4-a^2)}\]. In this case the locus has the transitional shape
shown in Fig 2.9(a), and all three closed loop poles are located at the
centre of gravity. The closed loop system then has a Benomial response
given by
\[
c(t) = \frac{K_0}{2} t^2 e^{-at}
\]

The third order type I system

The non-negative impulse response is of course associated with a
non-overshooting step response. Its main advantage will therefore
occur for the type I system which also has a zero steady state error
for a step input.

In this case the analytical results are greatly simplified and the
value of K required to obtain the in-line closed loop poles is given
Fig 2.9 Root Locus Shapes and Closed Loop Pole Distributions which give rise to a non-negative impulse response.

Fig 2.10 Non-negative Impulse Response of a Third Order type 0 system for $\omega_n = 1$ and $\alpha = 0.1$. 
Families of step and impulse responses for the system under these conditions are given in Figs 2.11 and 2.12. From these results it can be seen that there exists an optimum value of $\zeta$ corresponding to the fastest response.

Further empirical tests on the system showed that this value of $\zeta$ was 0.62, which is very close to the value of $\zeta$ that gives the maximum value of $K_0$ in equation (2.4.11).

The normalised open loop transfer function for the fastest non-overshooting closed loop step response is therefore given by

$$G(s) = \frac{0.273}{s(s^2 + 1.24s + 1)} \quad 2.4.12$$

It will be shown in section 2.5 that this system also has a shorter settling time than those designed according to the normally accepted closed loop performance criteria.

2.4.3 Third order system with zero

Important practical forms of closed loop response associated with the position of the centre of gravity of the open loop poles are by no means confined to all-pole systems. Consider for example the third order system with an open loop transfer function

$$G(s) = \frac{K(s + b)}{s(s^2 + 2\zeta s + 1)} \quad 2.4.13$$

In this case there are three basic root locus shapes as shown in Fig 2.13, which enable the closed loop poles to be placed at the centre of gravity of the open loop poles.
Fig 2.11 Non-negative Impulse Responses of a Third Order Type I System

Fig 2.12 Step response corresponding to the impulse responses given in Fig 2.1
Under these conditions the closed loop transfer function can be expressed in the form

\[ \frac{C(s)}{R} = \frac{\alpha^3(s + 1)}{(s + \alpha)^3} \quad 2.4.14 \]

where \( \alpha \) is the position of the centre of gravity of the open loop poles normalised with respect to the open loop zero at \( s = -b \).

The step response is then given by

\[ C(t) = 1 - e^{-\alpha t} \left[ 1 + \alpha t - \alpha^2 \left( \frac{\alpha - 1}{2} \right) t^2 \right] \quad 2.4.15 \]

and the corresponding impulse response is

\[ g(t) = \alpha^3 e^{-\alpha t} t \left[ 1 - \left( \frac{\alpha - 1}{2} \right) t \right] \quad 2.4.16 \]

Since \( t \neq 0 \), and \( e^{-\alpha t} \neq 0 \), the step response overshoot occurs at time \( t \) given by

\[ t = \frac{2}{(\alpha - 1)} \quad 2.4.17 \]

The case of \( \alpha < 1 \) applies to the locus shapes shown in Fig 2.13 (a)(b), which arise when the open loop zero is to the left of the centre of gravity of the open loop poles.

The closed loop step response is then faster than the Bionomial response of a three pole system and is aperiodic with no overshoot.

When \( \alpha > 1 \), the locus shape is as shown in Fig 2.13 (c) and the height of the overshoot in the step response is given by the equation

\[ H = (2\alpha - 1) e^{-\frac{2\alpha}{(\alpha - 1)}} \quad 2.4.18 \]

The step response is again aperiodic but in this case it has the very unusual feature of an overshoot with no undershoot.
\[ s(\omega_1^2 + 2\omega_n s + \omega_n^2) \]
\[ \sqrt{\frac{27}{32}} \geq |\xi| \geq \frac{\sqrt{2}}{2} \]
\[ b = \frac{8}{9} \frac{\omega_n}{(4\xi^2 - 3)} \]
\[ P_1 = \frac{1}{3} \{ b - 2\xi\omega_n \} \]
\[ \omega_m = \sqrt{\frac{27}{32}} - 3\xi^2 \]
\[ K_m = \frac{9\omega_n^2 (4\xi^2 - 3)}{(27 - 32\xi^2)} \]
\[ P_2 = -\frac{2}{3} \xi\omega_n \]

---

Fig 2.13 Relative pole-zero positions which enable the closed loop poles to be located at the C of G of the open loop poles.

---

\[ \frac{K(s+b)}{s(s+a)(s+c)} \]
\[ a > 2c \]
\[ b = \frac{1}{9} \left[ \frac{1}{(a+c)^2 - 3ac} \right] \]
\[ P_1 = -\frac{a+c}{2} \left[ \frac{8(a+c)^2 - 7ac}{9(a+c)^2 - 27ac} \right] \]
\[ P_2 = -\frac{a+c}{3} \]
\[ P_3 = -\frac{ac(a+c)}{2((a+c)^2 - 3ac)} \]

---

Fig 2.14 Relationship between the % overshoot of the step response and the position of the C of G of the open loop poles.
shows that all possible forms of this locus shape arise for values of $\alpha$ in the range $1 \leq \alpha \leq 3$.

The case of $\alpha = 1$ occurs when $b = c$ and corresponds to a critically damped second order system. The upper limit $\alpha = 3$ occurs when $c = 0$ and corresponds to the third order type 2 system shown in Fig 2.4(c).

The relationship between the step response percentage overshoot and the position of the centre of gravity of the open loop poles of these systems is shown in Fig 2.14.

2.5 Root Locus Shapes Associated with Specified Forms of Closed Loop Response

2.5.1 Introduction

Since a detailed specification of closed loop transient response is not normally required, most control systems are designed according to some form of overall index of performance. The system could for example be adjusted to have minimum integral squared error for a step input. This would then ensure minimum energy consumption by the controller. Another widely used criterion first proposed by Graham and Lathrop (11) is that of minimum ITAE. This has the advantage that it disregards the unavoidable initial error in the step response and penalises long duration transients. Fortunately both these criterion normally give rise to acceptable forms of closed loop response. If, however, a very precise transient response is specified then the required system can be obtained using Konwerski's synthesis method ref (12).
An alternative method of achieving acceptable closed loop performance is to regard the control system as a low pass filter. It can then be designed to have a Binomial, Butterworth, or Chebyshev response and will also have good noise rejection characteristics.

Although these design techniques are widely used in practice, the root locus shapes associated with specified forms of closed loop response are not generally known even for simple systems. The object of this section is to develop global maps for a selection of simple systems, to show how the general shapes of their root loci are related to specified forms of closed loop response. Also given on these maps are regions showing the root locus shapes associated with optimum forms of the closed loop system.

2.5.2 Root Locus shapes required for third order systems

The closed loop transfer functions of the ITAE, Binomial and Butterworth systems can be written in the form

\[ \frac{C(s)}{R} = \frac{1}{(s^3 + as^2 + bs + 1)} \] 2.5.1

Once each set of values of a and b has been determined the corresponding normalised parameters \( K \) and \( \zeta \) of the open loop system can be obtained from the expression

\[ \frac{C(s)}{R} = \frac{K}{(s^3 + 2\zeta s^2 + s + K)} \] 2.5.2

It is then possible to obtain a direct comparison of the root locus shapes, closed loop pole distributions and step responses corresponding to each specified form of the closed loop system. The results of this analysis are shown in Figs 2.15 and 2.16, together with the corresponding data for the optimum non-negative impulse response system discussed in section 2.4.2.
Fig 2.15 Root Locus Shapes associated with specified forms of closed loop response.

Fig 2.16 Step responses corresponding to the closed loop pole distributions shown in Fig 2.15.
It is interesting to note that the design criteria so far considered are all associated with the same general shape of the root locus. The step responses of these systems also confirm that the optimum NNI design does in fact have a shorter settling time than any of the normally accepted closed loop systems.

Also considered were the root locus shapes associated with the Chebyshev systems. These have a gain-frequency characteristic defined by the equation

$$\left| \frac{C(j\omega)}{R} \right|^2 = \frac{1}{1 + \varepsilon^2 T_N^2(\omega)}$$  \hspace{1cm} 2.5.3

From which the corresponding spectral factors are given by

$$\frac{C(s)}{R} \cdot \frac{C(-s)}{R} = \frac{1}{1 + \varepsilon^2 T_N^2(s/j)}$$  \hspace{1cm} 2.5.4

For a third order system

$$T_3^2(s/j) = -16s^2(s^2 + \frac{3}{4})^2$$  \hspace{1cm} 2.5.5

Hence the closed loop poles are therefore the roots of the equation

$$s^6 + 1.5s^4 + 0.6625s^2 - 0.0625\varepsilon^2 = 0$$  \hspace{1cm} 2.5.6

The three closed loop poles in the left hand half plane were obtained for values of \( \varepsilon^2 \) in the range \( 1 \geq \varepsilon^2 > 0 \).

In each case the resulting cubic was then normalised to the form

$$s^3 + 2\zeta s^2 + s + K = 0$$  \hspace{1cm} 2.5.7

to obtain the corresponding values of \( K \) and \( \zeta \).

The values of \( \zeta \) for these systems lie in the range \([\frac{1}{\sqrt{2}} \geq \zeta \geq 0]\), so that they all have the same general shape of root locus. Examples of this shape together with the corresponding closed loop pole distributions and step responses of the system are shown in Figs 2.17 and 2.18.
Fig 2.17 Root Locus Shapes associated with closed loop Chebyshev Poles of a Third Order Type I System.

Fig 2.18 Chebyshev responses of a Third Order Type I System.
From the step responses it can be seen that the system becomes very oscillatory for values of $\zeta < 0.5$. Since low values of $\zeta$ give the best noise rejection in this case, then the final design must be a compromise between noise rejection and transient overshoot.

The many different forms of closed loop response which arise for a third order type I system can now be summarised in a global $(K,\zeta)$ map as shown in Fig 2.17. The map also shows the values of $\zeta$ which correspond to transitional forms of the locus and indicates the range of values of $K$ and $\zeta$ which give rise to generally preferred designs for the closed loop system. When used in conjunction with the analytical data given in Fig 2.3, the map also enables the designer to see how the different forms of closed loop response depend on the general shape of the root locus.

It has been found that the small changes in $\zeta$ which produce a radical change in the locus shape on a transition boundary, do not give rise to a corresponding radical change in the closed loop response. The relationship between sudden change in locus shape and the change in the time response has also been investigated for many other systems and the same effect has been observed in all cases. Although there is a radical redistribution of the closed loop poles in these cases, the shape transition gives rise to an equivalent closed loop system which has virtually the same time response.

**The third order type 2 system**

In this case the root locus shapes are shown in Fig 2.4, and the only standard design criterion which can be used is that of minimum ITAE. It is therefore important to develop relationships between locus shapes and closed loop performance for other practical forms of the closed loop system.
Fig 2.19 Relationships between root locus shape, loop gain and the closed loop responses of a Third Order Type I System.
This can be achieved by writing the open loop transfer function in the normalised form

\[
G(s) = \frac{K(s + 1)}{s^2(s + A)}
\]  \hspace{1cm} 2.5.8

The locus shapes are then determined by the value of \(A\), while the closed loop response for each shape depends on the value of \(K\). The condition for the closed loop response to be aperiodic with overshoot but no undershoot can be deduced from the type I system previously discussed in section 2.43 and occurs when \(A = 9\), and \(K = 27\). Fortunately it has also been possible to determine the locus shapes and closed loop pole positions which give rise to the maximum damping ratio \(\zeta\) for the complex closed loop poles.

The required relationships were deduced from the Miklailov cartesian equation of the locus and are given by

\[
A = (1 + 2\zeta)^2
\]  \hspace{1cm} 2.5.9

and

\[
K = \omega_n^2 (1 + 2\zeta)
\]  \hspace{1cm} 2.5.10

where \(\omega_n = (1 + 2\zeta)\) is the undamped natural frequency of the complex closed loop poles.

Since \(\zeta \leq 1\), then \(K\) and \(A\) are related by the equation \(K = A\sqrt{A}\), for \(A \leq 9\). It is interesting to note that the constant gain contour for this value of \(K\) is a circle, so that the third closed loop pole is therefore located on the real axis at \(\alpha = -\omega_n^*\).

The designer can therefore specify the required bandwidth and speed of response by means of \(\omega_n\), and then obtain the locus shape and value of \(K\) for maximum damping from equations 2.5.9 and 2.5.10.

Many other relationships between the general shape of the root locus, the loop gain and the closed loop response have been obtained and these are summarised in the \((K,A)\) map shown in Fig 2.20.
Fig 2.20 Relationships between root locus shape, loop gain and closed loop response of a Third Order Type 2 System.
Maximum damping condition $K = A \sqrt{A}$

Minimum ITAE response $A = 5.7$, $K = 34.5$.

Fig 2.21 Closed loop responses of a third order Type 2 System.
2.5.3 Four pole systems

In this case the closed loop transfer functions of the ITAE, Binomial, Butterworth and Chebyshev systems can be expressed in the form

\[
\frac{C(s)}{R(s)} = \frac{1}{(s^4 + as^3 + bs^2 + cs + 1)}
\]

And the values of a, b and c can be determined to give the required closed loop performance. Factorisation of the resulting quartics showed that each design criterion was associated with the same general shape of root locus in the (A,ζ) plane. The values of K, A and ζ for each system were obtained by writing the closed loop transfer function in the normalised form

\[
\frac{C(s)}{R(s)} = \frac{K}{S^4 + (A + 2)S^3 + (1 + 2Aζ)S^2 + AS + K}
\]

It is then possible to obtain a direct comparison of the root locus shapes, closed loop pole distributions, and transient responses of these systems. The results of this analysis are shown in Figs 2.22 and 2.23.

Many other relationships between closed loop response and root locus shape have been investigated and these are summarised in the (A,ζ) map shown in Fig 2.24.
Fig 2.22 Closed Loop Pole positions and root locus shapes for specified closed loop performance.

Fig 2.23 Closed loop step responses of Four Pole Systems.
Fig 2.24 Root Locus Shapes of Four Pole Systems Associated with specified forms of closed loop response.
2.6 Summary

The work of this chapter and its supporting publications has helped to establish the basic root locus shapes of many simple systems which occur frequently in engineering design. In each case considered analytical data has been produced which enables the general shape of the root locus to be identified directly from the relative positions of the open loop poles and zeros. It has also been shown that suitable choice of normalising factors for simple pole-zero patterns can often lead to the display of all possible forms of their root loci on a single map. This new concept in root locus topology has been illustrated for the root loci of four-pole systems.

Many locus shapes are given which were previously unknown and these have been considered in relation to the transient response of the closed loop system. Transitional forms of root loci were of particular interest together with those which gave rise to closed loop poles at or vertically in-line with the C of G of the open loop poles. These systems often had unusual forms of closed loop response which in some cases were superior to standard forms of closed loop design.

Finally global maps were developed for a selection of simple systems to show how specified forms of closed loop performance are related to the general shapes of their root loci.

The author believes that the bulk of the work in this Chapter is original and represents a significant advance in our present knowledge of the analytical properties of simple root loci.
3.1 Introduction

The analytical data produced in Chapter 2 has now made it possible for the designer to obtain a very rapid sketch of the root loci of simple systems. Unfortunately it may not always be possible to approximate a system by one of these simple forms and it then becomes necessary to produce an accurate plot of the locus.

From a careful study of the methods available for plotting root loci it is shown that they all have very serious limitations. The author has now written a computer program which overcomes these difficulties and also enables loci to be obtained for systems which have pure time delay or distributed lag. The algorithm is very rapidly convergent and takes an average of less than two computations per point to compute the locus to a phase accuracy better than 20 seconds of arc.

The author claims that the method is original and is superior to any other known technique. Following extensive tests and considerable operating experience in conjunction with industry, the program is now able to plot the root locus of ANY linear system.

3.2 Review and Criticism of Present Methods

3.2.1 Analogue Techniques

Many analogue methods have been proposed all of which are either a variation of Levine's Steepest Ascent technique (13), or the D'azzo and Hoppis servo loop (14). Both of these methods require many resolvers, inverse resolvers, and multipliers. Such equipment is very
expensive and is not normally available in any quantity in a general purpose computer. Apart from the obvious limitations of the hardware such as low accuracy at small signal levels, there are three serious disadvantages which were not pointed out by the authors in their original papers.

The first of these is the fact that each branch of the locus must start on one of the open-loop poles. This means that pure-time delay loci cannot be obtained since they have branches which start at minus infinity.

In both methods the distance of the trial point from the open loop pole must increase monotonically as the computer traces out the branch of the locus. There are, however, many loci which have branches that curve back towards the open-loop pole as can be seen from the examples in Chapter 2. In such cases the computer leaves a gap in the locus branch until it finds a point at a greater distance from the open loop pole.

The third limitation is their inability to locate break points. This is due mainly to their low phase accuracy which is normally not much better than one degree. Unfortunately the phase accuracy required near a break-point is often seconds or arc, with the result that the computer either gives an incorrect locus shape or else it stops computing and has to be manually reset.

For loci which do not have pure time delay, break points, or branches curving back towards the poles, these methods give a fairly good quality plot which can be displayed on a C.R.O. or X-Y plotter. The D'azzo and Hoppis type methods also have the advantage that they can produce other phase angle loci simply by adjusting the set point of the servo loop.
3.2.2 Digital Computer Programs

The present digital computer programs are based either on the direct factorisation of the characteristic equation or a grid search technique in the S plane.

Direct factorisation was first suggested by Doda (15) who as usual failed to point out its practical disadvantages. Perhaps the most serious of these is the fact that all known factorisation techniques are unable to factorise polynomials which have two or more repeated roots. Since break points are in fact multiple closed loop poles, then these importance features of the locus are not available to the designer.

The direct method also has the disadvantage that all the roots of the characteristic equation must be determined simultaneously, while in practice the designer is usually more interested in only one or two branches of the locus.

Finally the method can only be extended to pure time delay systems using approximations such as Pade or truncated Maclaurin series, all of which give rise to erroneous results as will be shown in Chapter 4.

The first of the grid search methods was proposed in 1968 by Cook and Cook (16). In their method a trial point scans the S plane and tests the phase error at a large number of equally spaced points. The only new feature of their method is the use of a phase error function f(E) defined by the equation

$$ f(E) = \left( \frac{E}{2\pi} \right) - \text{ENTER} \left( \frac{E}{2\pi} \right) - 0.5 \ \text{sgn} \left( \frac{E}{2\pi} \right) $$

This has the effect of producing a sign change in f(E) as the phase error E passes through odd multiples of $\pi$. A point is assumed to lie on the locus if the magnitude of f(E) is less than 0.25 and its sign is different from that of its four closest neighbours.
The resulting plot consists of a pattern of crosses placed at these points and gives only a vague indication of the general shape of the locus. Another disadvantage is the large amount of storage required for the sign comparison required at the trial point.

A more recent grid search technique was reported by Aird and Moseley at the 1969 International Conference on Computer Aided Design (17). By simply testing the sign of the phase error at each consecutive point on the grid they claim to have plotted root loci using a small PDP8 computer with an 8K store. Unfortunately their plots are even more vague than those of Cook and Cook.

Both grid search methods are very inefficient since less than 1% of the points tested actually lie on or near to the locus. The author feels that these methods are a classic example of the misuse of a computer to overwhelm the problem rather than solve it.

3.2.3 Special Purpose Computers

Apart from the many amusing creations consisting of pieces of string wound around pulleys and potentiometers, there has only been one commercially available root locus computer called EASIAC (18). It was first introduced in 1960 and sold for about £6000.

Although it is called an automatic plotter, it is in fact only semi automatic and relies for its operation on a skilled operator who has to search for points in the S plane where the phase error is zero. In practice it is very difficult to use especially in regions of high sensitivity such as break points.

Another disadvantage of this apparatus is the fact that the loci are plotted on a log scale, which means that the designer finds it difficult to relate pole position to closed loop transient response.
3.3 The Author's New Search Vector Method

3.3.1 Introduction

From a careful study of the present methods available for plotting root loci, it became obvious that they could not be used as a serious research tool either for the study of root locus properties or for the extension of the technique to the design of systems with pure time delay. The author therefore decided to adopt a new approach to the problem and develop a method in which the trial point tracked the locus in a similar fashion to the guidance system of a beam rider missile. This technique has proved to be very successful and has resulted in a computer program which can plot the root loci of any linear system including those with pure time delay (19). The algorithm is very rapidly convergent and takes an average of less than two computations per point to achieve a phase accuracy better than 20 seconds of arc.

3.3.2 Basic Concept of the Algorithm

The computing action from an open loop pole is illustrated in Fig 3.1 which shows a typical branch of the locus arising from one of the complex poles of a pole-zero pattern KG(s).

The tangent to the locus at the open loop pole makes an angle $\psi_{10}$ with the $\sigma$ axis given by

$$\psi_{10} = \arg G(\sigma_i + j\omega_i) - \pi$$

This angle is then brought into the range $-\pi$ to $+\pi$ and a search vector of length $\Delta s$ sets off in the direction $\psi = \psi_{10}$ to give the first trial point at $(\sigma_t, \omega_t)$ where
Fig 3.1 Basic computing action from an open loop pole.
\[ \sigma_t = \sigma_i + \Delta s \cos \psi_t \quad 3.3.2 \]

\[ \omega_t = \omega_i + \Delta s \sin \psi_t \quad 3.3.3 \]

A multiple pole of order \( N \) at \( (\sigma_i, \omega_i) \) would of course give rise to \( N \) distinct branches of the locus, each with a different set-off direction. In this case each branch of the locus is computed separately and the set-off directions are obtained from the expression

\[ (\psi_{10})_r = (\psi_{10} + 2\pi r)/N \quad \text{for } r = 0 \text{ to } N-1 \quad 3.3.4 \]

Having located the first trial point, the phase error at this point is calculated from the equation

\[ \phi_t = \text{arg } G (\sigma_t + j\omega_t) - \pi \quad 3.3.5 \]

The heading of the search vector is then changed to

\[ \psi_t := \psi_t + \phi_t \quad 3.3.6 \]

which brings the trial point towards the locus. This process is repeated until the phase error at the trial point is less than 20 seconds of arc.

The need for such high accuracy is illustrated in Fig 3.2 which shows some of the phase angle loci of the system \( K(s+1)/s^2(s+9) \). Having located the first point on the locus, the plotter then joins the open pole to this point by means of a straight line.
Fig 3.2 Phase angle loci of the system $\frac{K(s+1)}{s^2 (s+9)}$ plotted to a phase accuracy of 20 seconds of arc.
The first point on the locus is then used as an anchor point from which the search vector \( \Delta s \) sets off in a direction \( \psi_{20} = \psi_{1i} \). Where \( \psi_{1i} \) is the final heading of the search vector from the previous anchor point which in this case is the open loop pole. The trial point then iterates towards the locus where it becomes the next anchor point.

The value of \( K \) is computed at each successful point and the whole process is repeated until \( K \) exceeds some specified value. Another branch of the locus may then be plotted by starting the computation at the appropriate open loop pole.

Unfortunately the convergence of the basic algorithm is very poor for the required phase accuracy of 20 seconds of arc. This is illustrated in Table (3.1) which shows the number of computations required at each of the first 31 points on the locus branch given in Fig 3.3. The average number of computations is 45 rising to a maximum of 146 in the region where there is a rapid change in curvature.

3.3.3 Refinements to improve Speed and Accuracy

Since most locus branches have fairly uniform curvature along small sections of the curve, it was decided to try the effect of setting off from the \( i \)th anchor point in a direction

\[
\psi_i(0) = 2\psi_{i-1}(0) - \psi_{i-1}(0)
\]

where \( \psi_{i-1}(0) \) is the final successful direction of the search vector to that point from the previous anchor point.
Fig 3.3 Root locus branch of the system $K/s (s^2 + 1.6s + 1)$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$K$</th>
<th>$G$</th>
<th>$C$</th>
<th>OSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.8000</td>
<td>0.6000</td>
<td>0.0000</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.7757</td>
<td>0.5682</td>
<td>0.0449</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.7507</td>
<td>0.5370</td>
<td>0.0840</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.7248</td>
<td>0.5065</td>
<td>0.1177</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.1462</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.6690</td>
<td>0.4493</td>
<td>0.1702</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.6382</td>
<td>0.4238</td>
<td>0.1900</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.6045</td>
<td>0.4022</td>
<td>0.2062</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.5674</td>
<td>0.3874</td>
<td>0.2196</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.5278</td>
<td>0.3830</td>
<td>0.2316</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.4886</td>
<td>0.3906</td>
<td>0.2433</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.4525</td>
<td>0.4077</td>
<td>0.2579</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.4197</td>
<td>0.4305</td>
<td>0.2750</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.3895</td>
<td>0.4568</td>
<td>0.2959</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.3613</td>
<td>0.4852</td>
<td>0.3211</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.3346</td>
<td>0.5149</td>
<td>0.3510</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.3089</td>
<td>0.5456</td>
<td>0.3861</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.2841</td>
<td>0.5770</td>
<td>0.4268</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.2600</td>
<td>0.6089</td>
<td>0.4734</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.2364</td>
<td>0.6412</td>
<td>0.5264</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.2132</td>
<td>0.6738</td>
<td>0.5861</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.1904</td>
<td>0.7066</td>
<td>0.6530</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.1679</td>
<td>0.7397</td>
<td>0.7274</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.1457</td>
<td>0.7729</td>
<td>0.8097</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.1236</td>
<td>0.8063</td>
<td>0.9002</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.1018</td>
<td>0.8398</td>
<td>0.9994</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.0801</td>
<td>0.8734</td>
<td>1.1077</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.0585</td>
<td>0.9071</td>
<td>1.2254</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.0371</td>
<td>0.9409</td>
<td>1.3529</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.0158</td>
<td>0.9748</td>
<td>1.4907</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.0055</td>
<td>1.0087</td>
<td>1.6390</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.0266</td>
<td>1.0426</td>
<td>1.7983</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The results of using this strategy to compute the locus branch shown in Fig 3.3 are given in table 3.2. Its effect is to reduce the average number of computations per point to 20 which represents a 100% improvement in efficiency. Despite this improvement the algorithm was still very slowly convergent and it was therefore decided to allow the system to overshoot to obtain a more rapid response.

The rate of change of phase error can vary considerably at different points along the branch of a locus as can be seen from Fig 3.2. In order to make the heading iteration loop adapt to these changes, an autogain parameter $G$ was introduced into the algorithm as follows

$$\psi_t := \psi_t + G \phi_t$$  \hspace{1cm} (3.3.8)

The autogain is initially set to unity at the open loop pole and is changed only if more than two computations are required to bring the trial point onto the locus. If the trial point has not reached the locus, then $G$ is increased by unity, else if it has overshot then the value of $G$ is multiplied by 0.75.

Autogain produced a dramatic improvement in the overall efficiency of the algorithm as can be seen from table 3.2. The average number of computations per point decreased to 2.5 and when curvature prediction was also included this was further reduced to 1.9 per point.

Extensive tests were then carried out on many different forms of locus and it was found that the combined effects of autogain and curvature prediction always gave an average of less than 2 computations per point for a phase accuracy of 20 seconds of arc.
## TABLE 3.2

The Results of Using Different Algorithms to Compute the Locus Shown in Fig 3.3.

<table>
<thead>
<tr>
<th>CO-ORDINATES OF POINTS ON THE LOCUS</th>
<th>CURVATURE PREDICTION ONLY</th>
<th>AUTOGAIN FUNCTION ONLY</th>
<th>CURVATURE PREDICTION AND AUTOGAIN FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>V</td>
<td>K</td>
<td>G</td>
</tr>
<tr>
<td>-0.8000</td>
<td>0.6000</td>
<td>0.0000</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.7757</td>
<td>0.5682</td>
<td>0.0449</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.7507</td>
<td>0.5370</td>
<td>0.0840</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.7268</td>
<td>0.5065</td>
<td>0.1177</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.1462</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.6690</td>
<td>0.4493</td>
<td>0.1702</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.6382</td>
<td>0.4138</td>
<td>0.1910</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.6045</td>
<td>0.3822</td>
<td>0.2062</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.5674</td>
<td>0.3574</td>
<td>0.2196</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.5278</td>
<td>0.3330</td>
<td>0.2316</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.4886</td>
<td>0.3008</td>
<td>0.2438</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.4525</td>
<td>0.2978</td>
<td>0.2579</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.4197</td>
<td>0.4307</td>
<td>0.2750</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.3895</td>
<td>0.4569</td>
<td>0.2959</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.3613</td>
<td>0.4353</td>
<td>0.3211</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.3346</td>
<td>0.5150</td>
<td>0.3511</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.3049</td>
<td>0.5457</td>
<td>0.3862</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.2841</td>
<td>0.5771</td>
<td>0.4209</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.2600</td>
<td>0.6090</td>
<td>0.4735</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.2364</td>
<td>0.6413</td>
<td>0.5265</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.2152</td>
<td>0.6739</td>
<td>0.5863</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.1904</td>
<td>0.7468</td>
<td>0.6532</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.1679</td>
<td>0.7393</td>
<td>0.7275</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.1457</td>
<td>0.7731</td>
<td>0.8098</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.1236</td>
<td>0.8064</td>
<td>0.9044</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.1018</td>
<td>0.8399</td>
<td>0.9996</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.0301</td>
<td>0.8735</td>
<td>1.1079</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.0585</td>
<td>0.9072</td>
<td>1.2257</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.0371</td>
<td>0.9410</td>
<td>1.3532</td>
<td>1.00</td>
</tr>
<tr>
<td>-0.0158</td>
<td>0.9749</td>
<td>1.4910</td>
<td>1.00</td>
</tr>
<tr>
<td>0.0054</td>
<td>1.0088</td>
<td>1.6393</td>
<td>1.00</td>
</tr>
</tbody>
</table>
If the first two headings of the search vector fail to give a point on the locus, then the phase errors at these points can be used to calculate the average rate of change of phase w.r.t. heading. Assuming a linear phase variation, the change in heading required to bring the search vector onto the locus is given by

$$\Delta \psi = \phi \frac{\partial \psi}{\partial \phi} = \frac{G \phi_1 \phi_2}{(\phi_2 - \phi_1)}$$  \hspace{1cm} (3.3.9)

where \( \phi_1 \) and \( \phi_2 \) are the phase errors at the first and second trial points respectively.

A detailed print out of the effect of using heading prediction for the computation of the locus branch shown in Fig 3.2 is given in table 3.3. The results show that there is a fantastic increase in phase accuracy which for three computations is often better than a millisecond of arc. The overall effect for a larger number of points is shown in table 3.4 from which it can be seen that heading prediction also increases the average number of computations per point by 30%.

It was therefore decided that this extra refinement was not justified for normal loci which are sufficiently well defined by a phase accuracy of 20 seconds of arc.

An example of the quality of locus produced by the author's new algorithm is shown in Fig 3.4. All 5 branches of the locus were computed in less than 15 seconds and when plotted on-line the complete plot was obtained in 65 seconds.
### TABLE 3.4

The Effect of Heading Predictor on the Number of Computations

"C" per point required for the Locus shown in Fig 3.3.

<table>
<thead>
<tr>
<th>POINT</th>
<th>COUNT</th>
<th>LOCUS CO-ORDINATES</th>
<th>PLUS HEADING PREDICTOR</th>
<th>NORMAL ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X</td>
<td>Y</td>
<td>G</td>
</tr>
<tr>
<td>0</td>
<td>0.00</td>
<td>-0.8000</td>
<td>0.6000</td>
<td>1.0000</td>
</tr>
<tr>
<td>1</td>
<td>1.00</td>
<td>-0.7757</td>
<td>0.5682</td>
<td>0.5625</td>
</tr>
<tr>
<td></td>
<td>2.00</td>
<td>-0.7757</td>
<td>0.5682</td>
<td>0.5625</td>
</tr>
<tr>
<td></td>
<td>3.00</td>
<td>-0.7757</td>
<td>0.5682</td>
<td>0.5625</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>-0.7507</td>
<td>0.5370</td>
<td>0.5370</td>
</tr>
<tr>
<td></td>
<td>2.00</td>
<td>-0.7507</td>
<td>0.5370</td>
<td>0.5370</td>
</tr>
<tr>
<td></td>
<td>3.00</td>
<td>-0.7507</td>
<td>0.5370</td>
<td>0.5370</td>
</tr>
<tr>
<td>3</td>
<td>1.00</td>
<td>-0.7247</td>
<td>0.5062</td>
<td>0.5062</td>
</tr>
<tr>
<td></td>
<td>2.00</td>
<td>-0.7247</td>
<td>0.5062</td>
<td>0.5062</td>
</tr>
<tr>
<td></td>
<td>3.00</td>
<td>-0.7247</td>
<td>0.5062</td>
<td>0.5062</td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.4771</td>
</tr>
<tr>
<td></td>
<td>2.00</td>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.4771</td>
</tr>
<tr>
<td></td>
<td>3.00</td>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.4771</td>
</tr>
</tbody>
</table>

**The Effect of Heading Predictor on Phase Accuracy**

<table>
<thead>
<tr>
<th>COUNT</th>
<th>LOCUS CO-ORDINATES</th>
<th>PLUS HEADING PREDICTOR</th>
<th>NORMAL ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>-0.8000</td>
<td>0.6000</td>
<td>-0.9272952141</td>
</tr>
<tr>
<td>1.00</td>
<td>-0.7757</td>
<td>0.5681</td>
<td>0.0024923410</td>
</tr>
<tr>
<td>2.00</td>
<td>-0.7757</td>
<td>0.5681</td>
<td>0.0016200164</td>
</tr>
<tr>
<td>3.00</td>
<td>-0.7757</td>
<td>0.5682</td>
<td>-0.0000000456</td>
</tr>
<tr>
<td>1.00</td>
<td>-0.7507</td>
<td>0.5369</td>
<td>0.003349256</td>
</tr>
<tr>
<td>2.00</td>
<td>-0.7507</td>
<td>0.5370</td>
<td>0.003044155</td>
</tr>
<tr>
<td>3.00</td>
<td>-0.7507</td>
<td>0.5370</td>
<td>-0.0000000048</td>
</tr>
<tr>
<td>1.00</td>
<td>-0.7247</td>
<td>0.5065</td>
<td>0.002975872</td>
</tr>
<tr>
<td>2.00</td>
<td>-0.7247</td>
<td>0.5065</td>
<td>0.002882752</td>
</tr>
<tr>
<td>3.00</td>
<td>-0.7247</td>
<td>0.5065</td>
<td>-0.0000000086</td>
</tr>
<tr>
<td>1.00</td>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.0030606171</td>
</tr>
<tr>
<td>2.00</td>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.003023602</td>
</tr>
<tr>
<td>3.00</td>
<td>-0.6977</td>
<td>0.4771</td>
<td>0.0000000196</td>
</tr>
<tr>
<td>1.00</td>
<td>-0.6690</td>
<td>0.4493</td>
<td>0.003219589</td>
</tr>
<tr>
<td>2.00</td>
<td>-0.6690</td>
<td>0.4493</td>
<td>0.003201382</td>
</tr>
<tr>
<td>3.00</td>
<td>-0.6690</td>
<td>0.4493</td>
<td>0.000448627</td>
</tr>
</tbody>
</table>
Fig 3.4 Root Locus of an aircraft pitch attitude control loop computed and plotted on line in 65 seconds.
3.4 Classical Techniques

3.4.1 Introduction

Following the high accuracy which was obtained using only a simple heading predictor, the author decided to compare his non-linear adaptive algorithm with a more orthodox analytical approach. Having once decided to track the locus by driving the search vector to zero phase error, it would appear that classical techniques could be used to advantage to predict the direction of the search vector to the next point on the locus. The object of the following analysis is to show that these techniques lack the elegant simplicity of the author's method and furthermore that they are also very much less efficient.

3.4.2 Computation of the slope of the Chord joining two points on the locus

Assuming that the point \((x,y)\) lies on the locus, then the phase error \(\phi\) at the point \((x+\Delta x, y+\Delta y)\) is given by

\[
\phi(x + \Delta x, y + \Delta y) = \phi(x,y) + \frac{\partial}{\partial x} \phi(x,y) \cdot \Delta x
+ \frac{\partial}{\partial y} \phi(x,y) \cdot \Delta y + O(|\Delta s|^2) \ldots
\]

where \(\Delta s = \Delta x + j \Delta y\), is the search vector from the point \((x,y)\) on the locus.

If the point \((x + \Delta x, y + \Delta y)\) also lies on the locus then

\[
\phi(x + \Delta x, y + \Delta y) = \phi(x,y) = 0
\]

The heading \(\psi_1\) of the search vector joining these two points is therefore given by

\[
\psi_1 = \arctan \left( \frac{\Delta y}{\Delta x} \right) = \arctan \left[ \frac{\partial \phi(x,y)}{\partial x} / \frac{\partial \phi(x,y)}{\partial y} \right]
\]
Now $\phi(x,y)$ is known analytically from the positions of the open loop poles and zeros i.e.

$$\phi(x,y) = \sum_{j=1}^{i=M} \arctan \left( \frac{y - Y_j}{x - X_j} \right) - \sum_{i=1}^{i=N} \arctan \left( \frac{y - Y_i}{x - X_i} \right) - (2k + 1)\pi$$  \hspace{1cm} 3.4.4

where $Y_j$ is the ordinate of the $j$th zero and $X_i$ is the abscissa of the $i$th pole.

Hence analytical expressions can be obtained for $\frac{\partial}{\partial x} \phi(x,y)$ and $\frac{\partial}{\partial y} \phi(x,y)$ by differentiation of equation (3.4.4) i.e.

$$\frac{\partial}{\partial x} \phi(x,y) = - \sum_{j=1}^{i=M} \frac{(y - Y_j)}{(x - X_j)^2 + (y - Y_j)^2} + \sum_{i=1}^{i=N} \frac{(y - Y_i)}{(x - X_i)^2 + (y - Y_i)^2}$$  \hspace{1cm} 3.4.5

and

$$\frac{\partial}{\partial y} \phi(x,y) = \sum_{j=1}^{i=M} \frac{(x - X_j)}{(x - X_j)^2 + (y - Y_j)^2} - \sum_{i=1}^{i=N} \frac{(x - X_i)}{(x - X_i)^2 + (y - Y_i)^2}$$  \hspace{1cm} 3.4.6

3.4.3 Analytical expression for the heading correction

A heading function $H(\psi)$ can now be defined by the equation

$$H(\psi) = \phi(x + a \cos \psi, y + a \sin \psi)$$  \hspace{1cm} 3.4.7

which is zero when $\psi$ is the correct heading of the next point on the locus.

Due to the non-linear nature of the function $\phi(x,y)$ and the fact that high order terms were neglected, the initial heading $\psi_1$ will be such that $H(\psi_1) \neq 0$. 
An estimate of the change in heading $\Delta \psi$ that is required to bring the trial point onto the locus can be obtained by expanding the function $H(\psi)$ about the heading $\psi_1$ as follows:

$$H(\psi_1 + \Delta \psi) = H(\psi_1) + \frac{\partial}{\partial \psi} H(\psi_1) \Delta \psi + O(|\Delta s|^2)$$ 3.4.8

Assuming that $H(\psi_1 + \Delta \psi) = 0$ i.e. that this is now the correct heading, then the required change $\Delta \psi$ is given by:

$$\Delta \psi = -\frac{H(\psi_1)}{\frac{\partial}{\partial \psi} H(\psi_1)}$$ 3.4.9

Now $\frac{\partial}{\partial \psi} H(\psi_1) = \left(\frac{\partial \phi}{\partial x} \frac{dx}{d\psi}\right)_{\psi=\psi_1} + \left(\frac{\partial \phi}{\partial y} \frac{dy}{d\psi}\right)_{\psi=\psi_1}$ 3.4.10

which when simplified reduces to:

$$\frac{\partial}{\partial \psi} H(\psi) = -\frac{\partial}{\partial x} \phi(x + \Delta s \cos \psi_1, y + \Delta s \sin \psi_1) \Delta s \sin \psi_1$$

$$+ \frac{\partial}{\partial y} \phi(x + \Delta s \cos \psi_1, y + \Delta s \sin \psi_1) \Delta s \cos \psi_1$$ 3.4.11

3.4.4 Comparison with the Author's Algorithm

While the classical method will eventually give an accurate point on the locus, it is less efficient than the author's method for the following reasons.

(1) A complex calculation involving the ARCTAN of the sum of a set of partial derivatives obtained from each of the open loop poles and zeros is required to even obtain a first approximation to the correct heading of the search vector. In the author's method this direction is obtained by a very simple calculation involving only the difference between the initial and final angles of the search vector at the previous point.
(2) Both methods then compute the phase error at the first trial point. If this value lies outside the tolerance band, the classical method would now have to perform another complex calculation in order to approximate the heading change required to bring the trial point onto the locus. The author's method computes the required change in heading using a simple adaptuon auto-gain parameter.

The success of the author's algorithm over all other known techniques is not only due to its simplicity but also to its adaptive nature which enables it to take advantage of experience gained at the other points on the locus.

3.5 Modifications of the Algorithm to deal with particular locus singularities

3.5.1 Break points

These occur on the locus when two or more of its branches coalesce at a single point. Unlike previous methods of automatic locus plotting, the author's algorithm is able to locate these points very accurately.

The method used is based on the fact that the value of K must increase continuously along the locus and reaches a stationary value at the break point. A break point has occurred if the value of K at a point on the locus is less than that at the previous point. The position of the break point is then located more accurately by approaching it from the overshoot position using a smaller vector length.

In practice it is found that location of a break point to a positional accuracy of 0.1% of the width of the plot has a negligible effect on the time taken to plot the locus.
Since all other known techniques either fail completely at break points or else give incorrect results, the author decided to investigate the ultimate accuracy that could be achieved using his new algorithm.

The break point is located by comparing the value of $K$ with that at the previous point on the locus. If $K$ has decreased then the search vector is rotated through $180^\circ$ and its length is reduced by a factor of 10. Normally only one reversal of direction is employed, but for the purpose of this experiment seven crossings of the break point were used giving a final vector length of $10^{-7} \Delta s$.

The change of direction of the locus branch at the break point is $(\pi/N)$ where $N$ is the number of branches entering or leaving that point. Since the value of $N$ cannot be known a-priori, the algorithm tests a sequence of values of $N = 2, 3, 4 \ldots$ etc. until a direction is found where the value of $K$ is greater than that at the break point. The vector length is then restored to its original value $\Delta s$ and the algorithm returns to normal tracking procedure.

The first example chosen is the four pole system shown in Fig 3.5 which illustrates the case when $N = 4$. A detailed print out of the location of the break point at (-1, 0) is given in table 3.5, which starts when the vector length has been reduced to $10^{-3}$. This is a particularly stringent test for the method due to the extremely low rate of change of $K$ near to the break point and in this case the final positional accuracy is only 1 part in $10^3$.

From table 3.5 it can be seen that the limiting factor is the effect of the rounding errors in the calculation of the value of $K$ which is only correct to eleven significant figures. The reason why all other techniques either diverge or give inaccurate results in this case is probably due to the very low rate of change of $K$. 
### TABLE 3.5

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.999000000000</td>
<td>-0.0000000020</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.000000000000</td>
<td>-0.0000000023</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.001000000000</td>
<td>-0.0000000025</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.000900000000</td>
<td>0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.000910000000</td>
<td>0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.000930000000</td>
<td>0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.000919000000</td>
<td>0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.000918000000</td>
<td>0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.000918100000</td>
<td>0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.0009180901</td>
<td>0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-1.0009181229</td>
<td>0.1000000000</td>
<td>0.9999999999563155</td>
</tr>
<tr>
<td>-0.9302087389</td>
<td>0.0707119951</td>
<td>1.00009743320978</td>
</tr>
<tr>
<td>-0.8594975772</td>
<td>0.1414221896</td>
<td>1.001579226067406</td>
</tr>
</tbody>
</table>

### TABLE 3.6

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.9999599999</td>
<td>-0.0000000000</td>
<td>0.9999999999214146</td>
</tr>
<tr>
<td>-1.9999699999</td>
<td>-0.0000000000</td>
<td>0.9999999999570718</td>
</tr>
<tr>
<td>-1.9999800000</td>
<td>-0.0000000000</td>
<td>0.9999999999803549</td>
</tr>
<tr>
<td>-1.9999899999</td>
<td>-0.0000000000</td>
<td>0.9999999999963620</td>
</tr>
<tr>
<td>-1.9999999999</td>
<td>-0.0000000000</td>
<td>1.000000000012732</td>
</tr>
<tr>
<td>-2.0000099999</td>
<td>-0.0000000000</td>
<td>0.9999999999963620</td>
</tr>
<tr>
<td>-2.0000099999</td>
<td>-0.0000000000</td>
<td>0.9999999999963620</td>
</tr>
<tr>
<td>-2.0000091000</td>
<td>-0.0000000000</td>
<td>0.9999999999956344</td>
</tr>
<tr>
<td>-2.0000090899</td>
<td>-0.0000000000</td>
<td>0.9999999999956344</td>
</tr>
<tr>
<td>-2.0016258677</td>
<td>0.0099869295</td>
<td>1.004997071677280</td>
</tr>
<tr>
<td>-2.009345753</td>
<td>0.1996737006</td>
<td>1.019975495173639</td>
</tr>
<tr>
<td>-2.0227222519</td>
<td>0.2988003125</td>
<td>1.044905017743076</td>
</tr>
</tbody>
</table>
The second example given in Fig 3.6 is more typical of the type of break point which arises in root locus plotting and illustrates the case when $N = 2$. A detailed print out of the location of the break point $P_1$ at (-2,0) is shown in table 3.6 which starts when the vector length has been reduced to $10^{-4}$. In this case the final positional accuracy achieved was 1 part in $2 \times 10^5$.

### 3.5.2 Unusual locus properties in the region of certain zeros

During the early stages of development of the algorithm a very interesting phenomena occurred on several loci as the search vector approached one of the zeros. Although the vector length was often as small as 0.02 inches and the maximum value of $K$ was less than 10 the tracking procedure suddenly became completely unstable as shown in Fig 3.7.

Detailed investigations into the cause of this instability were carried out by plotting constant $K$ contours in the region of the zero. It was found that these contours were approximately circular and that the value of $K$ increased rapidly from 10 to infinity within a radial distance of 0.01 inches of the zero. The instability was therefore due to the search vector stepping over the zero before the maximum value of $K$ had been reached.

Various techniques of overcoming this difficulty were tried such as computing the distance of the trial point from each zero, and calculating the values of $K$ at radial distances $\Delta s$ around the zeros. Although these methods were successful in preventing zero step over, they considerably increased the time taken to plot the locus. Detailed print out of the tracking procedure showed that the vector normally steps over the zero onto or very near to the $0^\circ$ locus. This means that the magnitude of the phase error suddenly increases to
Root locus of

\[-1 = \frac{K (s+0.148)(s+0.27\pm j 0.349)}{s (s+1)(s+1.667)(s+0.2)(s+0.203\pm j 0.725)}\]

Fig 3.7 Locus showing the effect of zero step over on the basic tracking procedure.
about 3 radians. Since phase error is required for normal tracking, then testing its magnitude has a negligible effect on the computing time and it proved to be a very useful method for detecting and preventing zero step over.

3.6 Plotting Constant Gain Contours

3.6.1 Introduction

The success of the author's curve following algorithm has now opened up the possibility of automatic plotting of a wide variety of useful contours. In particular it should now be possible to obtain the constant gain contours of a pole-zero pattern.

When these are taken in conjunction with the root locus plot of the system they provide the designer with a complete global picture showing how the closed loop pole positions vary with loop gain $K$.

3.6.2 Plotting procedure

Constant gain contours of a pole-zero pattern are generally more difficult to plot than the corresponding phase angle loci since they do not start or terminate on any of the singularities of the open loop transfer function. Furthermore a single value of $K$ can often give rise to more than one closed loop contour.

The search for a constant gain contour begins at the pole $P$ which has the largest negative real part. A first trial point is taken at a point $T$ on the negative real axis which is large compared with $P$. The value of $K$ is then computed at $T$ and a point $M$ which is mid-way between $T$ and $P$. From these calculations the algorithm decides if the first point on the locus lies in the range $PM$ or $MT$. The correct range is then halved and the process is repeated until a point is found which has the specified value of $K$. 
If the algorithm has failed to locate the value of \( K \) after 100 iterations, it is assumed that the contour does not intersect that direction of the search line. A new direction is then tried and the process is repeated until the value of \( K \) is located.

Once the first point has been located a small search vector sets off in a direction which is orthogonal to the search line. The trial point then iterates towards the contour using the same algorithm as that described for plotting constant phase angle loci. In this case, however, the error is the difference between the gain at the trial point and the value of \( K \) for that contour.

Unlike the 180° root locus, the tolerance band must be adjusted for each new value of \( K \). In order to achieve similar speed and positional accuracy to the phase angle loci, the tolerance band is set to \( 10^{-2}\% \) of the nominal value of \( K \). The plotting of a contour continues until it becomes closed or reaches one of the boundaries defining the size of the plot.

Another contour is then plotted around the same pole and the process is repeated for all the required values of \( K \). The algorithm then searches for contours around the remaining poles and plots those contours which have not previously been plotted. In this way the complete set of constant gain contours can be obtained as illustrated in Fig 3.8.

The 19 closed contours for this system were computed and plotted on-line in less than 300 seconds. While this is considerably longer than that for phase angle loci, the extra time is entirely due to the additional search procedure required to locate the starting point of each locus.
Fig 3.8 Constant Gain Contours and Root Locus of the System $K(s+1)/s^2(s+9)$
Apart from the few cases derived by Blackman where gain contours can be plotted by evaluating analytical expressions (20), such curves have not previously been available to the designer.

3.7 The Root Loci of Systems which include Pure Time Delay

3.7.1 Phase correction representation of pure time delay

To plot an accurate root locus of a pure time delay system one must locate points in the S plane which satisfy the equation.

\[-1 = Ke^{-st} G(s)\]  \hspace{1cm} 3.7.1

The angle condition then becomes

\[\arg G(s) + \phi_c = (2k+1)\pi\] \hspace{1cm} 3.7.2

where \(\phi_c = -\omega\tau\) is the phase correction.

Curves of constant \(\phi_c\) are therefore straight lines parallel to the real axis as shown in Fig 3.9.

The value of \(K\) at any point on the locus is given by

\[K = \frac{\exp(\sigma \tau)}{|G(s)|}\] \hspace{1cm} 3.7.3

A great advantage of the phase correction technique is that it can also be used to represent distributed parameter elements such as a heat exchange. In this case the transfer function is

\[e^{-\sqrt{s\tau}}\] and the phase correction \(\phi_c\) is given by

\[\phi_c = -\sqrt{\frac{\tau}{2}} \sqrt{\sigma^2 + \omega^2 - \sigma}\] \hspace{1cm} 3.7.4

Curves of constant \(\phi_c\) are then confocal parabolae as shown in Fig 3.10 and are defined by the equation

\[\omega^2 = 4 \frac{\phi_c^2}{\tau} \left[ \sigma + \frac{\phi_c^2}{\tau} \right]\] \hspace{1cm} 3.7.5
Fig 3.9 Phase Angle Representation of the effect of a Pure Time Delay $e^{-s\tau}$.

Fig 3.10 Phase Angle Representation of a Distributed Lag $e^{-\sqrt{s\tau}}$.
The value of $K$ at any point on the locus can be calculated from the expression

$$K = \frac{1}{|G(s)|} \exp \left( \frac{\tau}{2} \sqrt{\sigma^2 + \omega^2} + \sigma \right) \quad 3.7.8$$

Manual plotting of distributed lag and pure time delay loci using a spirule can take up to three or four hours even for simple pole zero patterns. Chu(21) has suggested that one should first plot a family of phase angle loci and then superimpose these on the phase correction lines shown in Fig 3.10. The main disadvantage with this approach is that even if the designer first plots the phase angle loci he still has to add these to the phase correction curves and search for points where the net phase angle is $180^\circ$.

Although these techniques give the exact modes in the closed loop transient response, the work involved in constructing the loci has so far made them impractical for the design of control systems.

3.7.2 Extension of the author's algorithm to plot the primary branches of the locus

Unlike all previous methods of automatic plotting, the author's algorithm can be used to plot these loci with great speed and accuracy. It can also be used to obtain the corresponding constant gain contours.

The phase error at the trial point $(\sigma_t, \omega_t)$ now becomes

$$\phi_t = \arg G(\sigma_t + j\omega_t) - \pi + \omega_t \tau \quad 3.7.9$$

where $\tau$ is the pure time delay.

And the value of $K$ at each successful point is calculated from equation 3.7.8.
Since the author's original method is based on phase angle error, the slightly more complex expressions for $\phi_t$ has a negligible effect on the time taken to plot the locus. Using this method each branch is an accurate locus of one of the roots of the infinite order polynomial.

Although there are an infinite number of branches, the designer can now select a sufficient number of accurate roots to obtain a close approximation to the closed loop transient response.

3.7.3 Location of the extra branches of the locus

The effect of pure time delay on the root locus of the system $K/s$ is shown in Fig 3.11 from which it can be seen that the extra branches of the locus do not start at the open loop pole, but originate from the infinite order pole at infinity.

The algorithm first plots the branches from the open loop poles and then moves the trial point to the bottom left hand corner of the plot where it initiates a "LOCK-ON" procedure which operates as follows.

The trial point moves vertically upwards in steps of 0.1 As until it detects a change in the sign of the phase error. This occurs when the trial point has passed through a 0° or 180° branch of the locus. To distinguish between these two cases, the algorithm also tests the magnitude of the phase error, which for the 180° locus will be less than 1.5 radians, see Fig 3.12.

When the correct branch has been located, the search vector is rotated through -90° and the trial point moves onto the locus using the normal tracking procedure. Once the first anchor point has been established the search vector is restored to its original length and the remainder of the branch is then plotted.
Fig 3.11 Root Locus of the Pure Time Delay System

\[ s = \frac{1}{K} e^{-st} \]  
where \( \tau = 4 \) seconds.

Fig 3.12 Variation of Phase Error \( \phi_e \) along the \( j\omega \) axis for the time delay locus shown in Fig 3.11.
The next delay branch is located by starting the trial point at the left hand side of the plot with a vertical displacement equal to the peak value detected on the previous branch. The process is then repeated until 5 extra branches have been plotted or the trial point has exceeded the automatic cut out boundaries which define the size of the plot.

Accurate pure time delay loci such as that shown in Fig 3.11 are the first to be plotted using a computer. The author claims that the ability to plot these loci at a rate of 4 seconds per branch to an accuracy of 20 seconds of arc, represents a considerable extension to the design capability of the root locus technique.

3.7.4 Constant gain contours

In order to calibrate the pure time delay root loci and determine where the closed loop poles are insensitive to gain, the constant gain contour program was also modified.

In this case the value of $K$ at the trial point is calculated from the expression

$$K = \frac{e^{\sigma \tau}}{|G(\sigma + j\omega)|}$$

Unlike the root locus, there are no extra branches produced in this case. The general effect is to distort the shape of the contours so that they are no longer closed except near to a pole or zero.

The effect of a 0.1 second delay on the circular gain contours of the system $K/s$ is shown in Fig 3.13. An interesting feature of this system is the fact that the value of $K$ at the break point is now the maximum value of gain which gives rise to a closed contour.
Fig 3.13 Constant Gain Contours and Primary Root Locus Branches of the system $K \frac{e^{-\tau s}}{s}$ where $\tau = 0.1$ seconds.
When these contours are combined with the root locus of a pure time delay system they provide a complete picture of the movement of the closed loop poles with variation in loop gain.

3.8 Summary

A detailed study of present methods of automatic root locus plotting has shown that they all have very serious limitations and are often unreliable even for simple loci. The author's new method has overcome these difficulties and can also be used to obtain root loci of systems which have pure time delay or distributed lag. It has also been used successfully to obtain the first computer plots of constant gain contours. These are a particularly useful addition to the normal root locus since they enable the designer to obtain the position of the closed loop poles.

The author claims that the method is original and is superior to any other known technique. It has been extensively tested in conjunction with industry and is now able to plot the root locus of ANY linear system. An interesting feature of the new algorithm is its ability to locate break points with great speed and accuracy. This means that it could be used in reverse to solve the age-old problem of factoring polynomials when they have one or more repeated roots.

The success of the method has made possible the development of a low cost (£1500) Hybrid Computer capable of displaying root locus on a CRO. This computer is at present being built in the Department by Mr. Mansi who holds joint patents on the machine with the author and the National Research and Development Corporation. (22)
4.1 Introduction

One of the main advantages of the author's search vector method is that it can be extended to give accurate root loci of systems which contain pure time delay or distributed lag. This means that the designer can now obtain the exact positions of the closed loop poles and can therefore accurately predict the transient response. The ability to plot these loci now removes the need to approximate the effects of time delay by means of Pade functions or truncated Maclaurin series. It also enables the designer to specify a desired transient response and obtain the required open loop system by plotting the 0° locus which passes through the open loop poles.

The author also shows how these loci can be used to identify the dynamics of a pure time delay system for the transient modes contained in its closed loop impulse response.

4.2 Methods of Approximating the Closed loop system

4.2.1 Statement of the problem

From the block diagram of the general pure time delay system shown in Fig 4.1, it can be seen that the closed loop transfer function is of the form

\[ \frac{C(s)}{R(s)} = \frac{KG(s)}{e^{sT} + KG(s)} \] 4.2.1

Since there are an infinite number of closed loop poles in this case, the response of the system can only be obtained by approximating the characteristic equation by a finite order polynomial. This is normally achieved using some form of approximation to the time delay.
The accuracy of such an approximation depends on the order of the approximating polynomial used. An accurate prediction of the transient response requires the use of a high order polynomial with consequent extensive calculation to evaluate the response.

The author presents an alternative approach in which the system is represented by a linear closed loop system associated with a modified delay. The appropriate closed loop poles are obtained directly from the root locus of the system.

4.2.2 Methods of approximating the time delay

Series Expansion

The Maclaurin series expansion for the pure time delay function \( e^{-st} \) given by

\[
e^{-st} = \sum_{N=0}^{\infty} (-1)^N \frac{(st)^N}{N!}
\]

This can be truncated by expanding the corresponding series for \( 1/e^{st} \) namely

\[
\frac{1}{e^{st}} = \sum_{N=0}^{\infty} \frac{N!}{(st)^N}
\]

or else it can be expressed in the form

\[
e^{-st} = \text{Limit}_{N \to \infty} \frac{1}{(1 + \frac{st}{N})^N}
\]

in which case it can be regarded as an Nth order real pole at \( s = -N/\tau \).

The general form of the equivalent closed loop system arising from the use of truncated Maclaurin series approximation to the delay is shown in Fig 4.2.
Fig 4.1 Unity negative feedback control system with a pure time delay of $\tau$ seconds in the forward path.

Fig 4.2 Approximation to the delay by a Truncated Maclaurin Series Expansion of $1/e^{s\tau}$.

Fig 4.3 Approximation to the delay using Non-Minimum Phase Padé Functions for $e^{-s\tau}$. 
Pade Functions

Perhaps the most widely used approximations to time delay are the non-minimum phase functions due to Pade. The first order approximations based on this principle is given by

\[ e^{-st} \approx \frac{(s - 2/\tau)}{(s + 2/\tau)} = 1 - st + \frac{s^2\tau^2}{2!} - \frac{s^3\tau^3}{4} \] 4.2.5

Higher order approximations which are valid for a greater range of \( \omega t \) are given by the equation

\[ e^{-st} = \prod_{r=1}^{N} \frac{(s^2\tau_r^2 - 2\tau_r\tau_s + \tau_s \tau_r \ s + 1)}{(s^2\tau_r^2 + 2\tau_r\tau_s + \tau_s \tau_r \ s + 1)} \] 4.2.6

where \( \tau = \sum_{r=1}^{N} \frac{4 \tau_r \tau_s}{\tau_r} \) 4.2.7

Provided these functions are of sufficiently high order to produce less than 1° phase error over the normal \( \omega t \) band of the system, then they give a fairly good approximation to the closed loop response. They do, however, have the serious disadvantage that they predict an output for the system prior to the delay.

The equivalent closed loop system obtained using Pade Functions for the time delay is shown in Fig 4.3.

4.2.3 The Author's method using root locus plots

Since equation (4.2.1) could also have been written in the form

\[ \frac{C(s)}{R} = \frac{e^{-st} KG(s)}{1 + KG(s) e^{-st}} \] 4.2.8

then the closed loop system could also have been approximated as shown in Fig 4.4 by the delayed linear system

\[ \frac{C(s)}{R} = e^{-st'} K'G'(s) \] 4.2.9
Fig 4.4 Equivalent delayed Linear System in which the poles and zeros of $K'G'(s)$ are obtained from the root locus plot of $-1 = KG(s) e^{-s\tau}$.

Fig 4.5 Second order Type I system with a 1 second pure time delay in the forward path.
where the zeros of $G'(s)$ are those of the original system $G(s)$ and the poles are the roots of the closed loop polynomial. The roots of the complete closed loop polynomial are, of course, infinite in number but in general only a few of the roots dominate the response. An inspection of the root locus plot of an actual system with finite delay shows that the dominant poles are associated with the primary branches of the locus. Other poles which may significantly contribute to the response are located on the extra branches which are nearest to the primary branches.

In general inspection of the locus will show whether, as is often the case, a reasonable approximation to the response can be obtained using only the primary poles. If, however, a more accurate approximation is required then the extra modes can readily be obtained and included in the closed loop polynomial.

Since $G'(s)$ is a finite order polynomial it contributes to the total delay in the system so that the finite delay should be modified to

$$\tau' = \tau - \tau_c$$

where $\tau$ is the true delay and $\tau_c$ is a delay correction. The value of $\tau_c$ depends on the order of $G'(s)$ and it has been found empirically that $\tau_c$ should be chosen as the value of $\tau$ at which the step response of $G'(s)$ reaches 2% of its final value.

4.3 Prediction of Closed Loop Transient Response

4.3.1 Introduction

Consider for example the pure time delay system shown in Fig 4.5. We will now obtain the closed loop step response predicted by each of the methods discussed in section (4.2) and compare the results with the accurate response of the system.
To obtain the accurate response the author has extended his
digital simulation language to include pure time delay and details
of the method used are given in Appendix (1).

4.3.2 Truncated Maclaurin Series Approximations

Truncating the Maclaurin Series to give third and fourth order
approximations to the closed loop transfer functions in this case
gives

\[ \frac{C(s)}{R} = \frac{1.4}{(s+2.4)(s^2+0.59s+0.58)} \]  

and

\[ \frac{C(s)}{R} = \frac{2.8}{(s^2+0.36s+0.702)(s^2+3.64s+3.91)} \]

The step responses obtained using these approximations are shown
in Fig 4.6 together with the accurate response of the system.

4.3.3 Use of Pade Functions to represent the delay

When Pade Functions are used in this case the third and fourth
order approximations to the closed loop transfer functions become:

\[ \frac{C(s)}{R} = \frac{-1.4 (s-2)}{(s+3.48)(s^2+0.516s+0.814)} \]  

and

\[ \frac{C(s)}{R} = \frac{1.4 (s^2-6s+12)}{(s^2+0.353s+0.765)(s^2+7.64s+21.9)} \]

The step responses obtained using these transfer functions are
shown in Fig 4.7 together with the accurate response of the system.
Closed loop step responses of the system shown in Fig 4.5 obtained using Truncated Maclaurin Series approximations to the pure time delay.

Closed loop step response obtained using Pade Functions to approximate the pure time delay.
4.3.4 The Author's root locus method

The infinite number of closed loop poles which arise in this case are defined by the root locus equation

\[-1 = \frac{Ke^{-s}}{s(s+2)} \]  

This was plotted using the author's computer program to obtain the primary and first extra delay branches of the locus as shown in Fig 4.8. Also given are the closed loop pole positions corresponding to a value of \( K = 1.4 \).

From the locus the second and fourth order approximations to the closed loop transfer functions are given by

\[
\frac{C(s)}{R(s)} = \frac{0.8075 e^{-s\tau'}}{(s^2+0.45s+0.8075)}
\]

and

\[
\frac{C(s)}{R(s)} = \frac{33.6 e^{-s\tau'}}{(s^2+0.45s+0.8075)(s^2+6.5s+41.7)}
\]

The step responses obtained using the second order approximation and the true time delay of \( \tau = 1 \) second is shown in Fig 4.9 together with the accurate response of the system.

From the predicted response the value of \( \tau_c \) was found to be 0.12 seconds, so that the corrected closed loop transfer function becomes

\[
\frac{C(s)}{R(s)} = \frac{0.8075 e^{-0.88s}}{(s^2+0.45s+0.8075)}
\]

The predicted response obtained using this closed loop approximation is now very close to the true response of the system as can be seen from Fig 4.10. When the corrected fourth order approximation was used in this case it was not possible to obtain any measurable improvement
Fig 4.8 Root Locus of the pure time delay system

$$K \frac{e^{-st}}{s(s+2)}$$, where $\tau = 1$ second
Fig 4.9 Response predicted from the primary branches of the root locus and delayed by $\tau = 1$ second.

Fig 4.10 Step response shown in Fig 4.9 including a delay correction of $\tau_c = 0.12$ seconds.
4.3.5 Comparison of Results

The results of these tests show that the closed loop response can also be predicted from the root locus of the system and there is now no need to approximate the effect of the delay by large numbers of poles and zeros. Although the delay was almost 20% of the closed loop period in this case it was found that the transient response could be predicted using only a second order approximation obtained from the primary branches of the locus. To obtain comparable accuracy using Pade or Maclaurin series for the delay it was necessary to use at least a fourth order approximation to the closed loop transfer function.

The success of the root locus method is due to the fact that it uses the accurate closed loop poles of the system and makes no attempt to predict the response prior to the delay.

Since the closed loop poles which occur on the pure time delay root loci are in fact the true modes in the transient response, it should now be possible to use the technique in reverse to identify the dynamics of the system from measurements of closed loop performance.

4.4 Identification of the Dynamics of Closed Loop Linear Systems which include Pure Time Delay

4.4.1 Introduction

The starting point of the method is the measured impulse response of the system. This has been chosen since it can be obtained in practice using Lee's (23) input-output correlation techniques and also gives a direct measure of the pure time delay.

The work of this section shows how the author has been able to combine impulse response data with accurate root locus plots to give the first successful identification of the dynamics of a system which includes a pure time delay.
4.4.2 Determination of the system transfer function from its measured impulse response

The impulse response $h(t)$ is first mapped into the frequency domain using the Fourier Transform

$$F(j\omega) = \int_{0}^{\infty} h(t) e^{-j\omega t} \, dt$$

This can also be written as

$$F(j\omega) = R(\omega) + jI(\omega)$$

where

$$R(\omega) = \int_{0}^{\infty} h(t) \cos \omega t \, dt$$

and

$$I(\omega) = \int_{0}^{\infty} h(t) \sin \omega t \, dt$$

Using correlation techniques, the impulse response is only available at discrete points. Hence these integrals must be evaluated in terms of the measured values $h(t_i)$ at the sampling instants $t_i$.

Approximating the impulse response using isosceles triangles centred on the sampling instants, the real and imaginary parts of the Fourier Integral become:

$$R(\omega) = \sum_{i=1}^{N} 2 h(t_i) \Delta t \frac{\cos(i\omega \Delta t)(1 - \cos\omega \Delta t)}{(\omega \Delta t)^2}$$

and

$$I(\omega) = -\sum_{i=1}^{N} 2 h(t_i) \Delta t \frac{\sin(i\omega \Delta t)(1 - \cos\omega \Delta t)}{(\omega \Delta t)^2}$$

where $N$ is the number of samples used to represent the impulse response, and $2\Delta t$ in the base length of the triangles.

The accuracy of these approximations was checked by mapping the impulse response of known systems and comparing the results with
those obtained from direct evaluation of their frequency response.
To achieve an accuracy better than 1% it was found that

(1) At least 20 points were required per cycle of the response
and

(2) Sufficient points had to be taken to ensure that the response
had decreased to less than 1% of its peak value.

Having obtained \( R(\omega) \) and \( I(\omega) \) for the system, the transfer function
is then determined from the Bode diagram using Konwerski's computer
program of Levy's least square curve fitting technique (24).

### 4.4.3 Extension to pure time delay systems

As an example of the author's method of identifying the dynamics
of closed loop pure time delay systems, we will now consider the
system \( G(s) = Ke^{-ST}/s(s+2) \) which was previously discussed in section
(4.3).

The impulse response of the system for \( K = 1.40 \) and \( \tau = 1 \), was
obtained by digital simulation and is shown in Fig 4.11. The resulting
waveform consists of a zero output up to \( t = 1 \) seconds due to the
delay, followed by a transient which is determined by the delay and
the pole-zero pattern of the open loop system.

When the complete response was mapped into the frequency domain
it was found that the phase angle was 180° at 1.06, 6.6 and 12.6
radians, and passed through 0° at 3.65 and 9.6 radians. These fre-
quencies correspond to the \( j\omega \) axis crossings of the 180° and 0° locus
branches and agree to within 1% with those obtained from the root locus
plot. It was, therefore, rather surprising to find that Levy's curve
fitting method gave rise to completely erroneous results for the pole-
zero pattern of the system.
Fig 4.11 The impulse response of the pure time delay system shown in Fig 4.5.

Fig 4.12 $0^\circ$ Root Locus arising from the closed loop poles determined from the impulse response shown in Fig 4.11.
gave the following closed loop transfer functions

Second order approximation

\[
\frac{C}{R}(s) = \frac{0.807}{(s^2 + 0.455s + 0.8)}
\]

Fourth order approximation

\[
\frac{C}{R}(s) = \frac{35.6}{(s^2 + 0.467s + 0.817)(s^2 + 5.4s + 43.3)}
\]

We will first assume that there are two closed loop poles on primary branches of the locus and that the higher frequency modes arise from poles on the extra pure time delay branches.

To determine the possible positions of the open loop poles, the 0° locus including the 1 second delay was plotted from these closed loop poles and is shown in Fig 4.12. Since the impulse response tends to zero as \( t \) tends to infinity then the open loop response must contain a pole at the origin, and a real pole at \( s = -a \). While the 0° and 180° phase angle loci of pure time delay systems are reversible, this is not the case for the constant gain contours. In general, the designer must therefore plot families of 180° loci to determine the combinations of open loop pole positions which cause the primary branches of the locus to pass through the measured closed loop poles. The correct combination is then determined by reference to the higher frequency modes which arise from the extra delay branches of the locus.

The position of the pole at \( s = -a \) in this example can, however, be determined directly from the geometry of the system as shown in Fig 4.13. It was found that \( a = 2.05 \) and the value of \( K \) which gave the measured closed loop poles was \( K = 1.38 \).
Fig 4.13 Determination of the position of the pole $s = -a$ to give the measured closed loop poles.

\[ a = 0.225 + 0.87 \cot (180^\circ - 0.87 - \phi_1) \]

Fig 4.14 Root locus of the system $Ke^{-st}/s(s+\tau)$ where $a = 2.05$, and $\tau = 1.0$ seconds.
The complete locus for \( a = 2.05 \) was then plotted and is shown in Fig 4.14. From this plot it can be seen that the branches of the locus should cross the \( j\omega \) axis at \( \omega = 1.05 \) and 6.6 radians/sec. And for \( K = 1.38 \), the transient response should also contain a mode \((s^2 + 6.52s + 42.1)\).

These results are all confirmed from measurements of the closed loop impulse response and therefore the open loop transfer function is given by

\[
G(s) = \frac{1.38 e^{-s}}{s(s + 2.05)}
\]

This result agrees with the known dynamics of the system to an accuracy of 2%.

If, however, the extra locus branches had failed to confirm the open loop pole-zero pattern, the designer must then assume that the high order modes are due to closed loop poles on primary branches of the locus. The iterative scheme is then repeated until the open loop pole-zero pattern obtained gives rise to extra locus branches which have closed loop poles that correspond to the modes in the measured impulse response.

Further examples of the method are given in Chapter 6 where it is used to identify the transfer function of a human pilot from measured flight records.

4.5 Methods of Compensating for the effects of Pure Time Delay

4.5.1 Use of Minor Loop Feedback

This technique was first proposed by Professor O. J. Smith (25) in 1959 and is illustrated in the block diagram shown in Fig 4.15. Applying Mason's rules (26) to the corresponding signal flow diagram shown in Fig 4.16, the closed loop transfer function is given
Fig 4.15 Use of minor loop feedback $H(s)$ to compensate for the effects of pure time delay $\tau$ in the Plant $P(s) = KG(s) e^{-sT}$

Fig 4.16 Signal Flow Graph for the system shown in Fig 4.15.

Fig 4.17 Output error due to unit step disturbance for the case when $P(s) = \frac{4e^{-s\tau}}{s}$, $M(s) = 10$, and $\tau = 1$ second.
\[
\frac{C}{R}(s) = \frac{KM(s)G(s)e^{-s\tau}}{1 + M(s)[H(s) + KG(s)e^{-s\tau}]} \tag{4.5.1}
\]

From which it can be seen that the effect of pure time delay on closed loop stability can be removed using a minor loop feedback \(H(s) = KG(s)[1 - e^{-s\tau}]\). Under these conditions the closed loop transfer function reduces to

\[
\frac{C}{R}(s) = \frac{KM(s)G(s)e^{-s\tau}}{1 + KM(s)G(s)} \tag{4.5.2}
\]

It should be noted that this form of compensation does not remove the delay in the closed loop response, but simply attempts to eliminate its effect on closed loop stability.

Taking a specific example of \(G(s) = 1/s\) and \(M(s) = K_1\), equation 4.5.2 becomes

\[
\frac{C}{R}(s) = \frac{K K_1 e^{-s\tau}}{(s + K K_1)} \tag{4.5.3}
\]

This system has zero steady state error for a step input, and it would appear that the minor loop feedback has achieved a simple method of producing a satisfactory closed loop system. Unfortunately Professor Smith did not consider the effect of his minor loop feedback on the steady state error produced by external disturbances applied to the plant.

The transfer function relating the controlled variable \(C\) to a disturbance \(D\) applied to the plant is given by

\[
\frac{C}{D}(s) = \frac{P(s)[1 + M(s)H(s)]}{1 + M(s)[H(s) + P(s)]} \tag{4.5.4}
\]
using the form of minor loop feedback \( h(s) \) suggested by Professor Smith this expression becomes

\[
\frac{C}{D}(s) = \frac{K M(s) G(s) e^{-s\tau}}{[1 + K M(s) G(s)]} \cdot KG(s) [1 - e^{-s\tau}] + \frac{1}{M(s)} \quad 4.5.5
\]

Applying this result to the system \( G(s) = 1/s, M(s) = K_1 \), the steady state output produced by a unit step disturbance is given by

\[
(C)_{ss} = \lim_{s \to 0} \left[ s \cdot \frac{1}{s} \cdot \frac{K e^{-s\tau}}{(s + K K_1)} \cdot \left[ 1 + K \left( 1 - e^{-s\tau} \right) \right] \right] \quad 4.5.6
\]

Although this limit is indeterminate it can however be evaluated using L'Hopital's rule to give

\[
(C)_{ss} = K \tau + \frac{1}{K_1} \quad 4.5.7
\]

Hence the price paid for removing the destabilising effect of the delay is to increase the steady state error by an amount \( K \tau \).

The effect is illustrated in Fig 4.17, which shows that when \( K = 4, K_1 = 10 \) and \( \tau = 1 \), the error due to a unit offset is increased from 0.1 to 4.1.

4.5.2 Inverse Root Locus Methods

Since the effect of pure time delay cannot be removed from the output, the best that can be achieved is to design a closed loop system with a desired transient response. It has been shown in section (4.2) that the closed loop system can be regarded as a delayed linear system whose poles and zeros can be obtained directly from the root locus. Hence once the closed loop pole-zero pattern to achieve a particular response has been specified, the corresponding open loop poles can then be obtained using the author's root locus plotting program as follows.
Let $T(s) = \text{the closed loop pole-zero pattern which gives the desired response.}$

$KG(s) = \text{the open loop pole-zero pattern excluding the pure time delay } \tau.$

The closed loop transfer function of the delayed linear system is given by

$$T(s) = \frac{KG(s)}{1 + KG(s) e^{-\tau s}} \quad 4.5.8$$

From which

$$KG(s) = \frac{T(s)}{1 - T(s) e^{-\tau s}} \quad 4.5.9$$

The poles of $KG(s)$ can then be obtained using the author's search vector method to plot the accurate $0^\circ$ pure time delay locus defined by the equation

$$+1 = T(s) e^{-\tau s} \quad 4.5.10$$

To illustrate this technique we will now consider the design of a third order type I system to achieve a specified transient response.

### 4.6 The use of Inverse Root Locus Methods to Determine the Optimum Step Response Parameters of a Pure Time Delay Third Order Type I system

#### 4.6.1 Optimum performance without time delay

Consider the general third order type I pure time delay system shown in Fig 4.18. It has been shown in Chapter (2) that the fastest possible step response of a third order type I system with $\tau = 0$, occurs when all three closed loop poles are vertically in-line and the complex poles have a damping ratio of 0.63.

Under these conditions the open loop transfer function normalised to $\omega_n = 1$ is given by
Fig 4.18 Third Order Type I pure time delay system.

Fig 4.19 Step responses of the system shown in Fig 4.18 for the case when $\tau = 0$ and $\omega_n = 1$. 
and the corresponding closed loop transfer function becomes

\[
\frac{C(s)}{R(s)} = \frac{0.26}{(s + 0.47)(s^2 + 0.94s + 0.56)}
\]

The step response which is shown in Fig (4.19), then has a shorter settling time and less overshoot than any of the normally accepted designs.

4.6.2 Optimum performance including the effect of pure time delay

Since \( \omega_n \) is simply a time scale factor, the open loop transfer function including the pure time delay can also be normalised with respect to \( \omega_n \) to give.

\[
G(S) = \frac{K_N e^{-S\tau_N}}{S(S^2 + 2\zeta S + 1)}
\]

where \( \tau_N = \tau \omega_N, \quad K_N = K/\omega_n^3 \) and \( S = s/\omega_n \).

Hence provided we can determine the values of \( K_N, \tau_N \) and \( \zeta \) which give the required dominant closed loop pole pattern, then the resulting system will have the fastest possible step response.

The method used to obtain these values was to plot the 0° root locus for a particular value of \( \tau \), starting at the closed loop poles \( s = -0.47, \quad s = -0.47 \pm j 0.583 \). The open loop pole positions corresponding to a type I system were obtained and the complex poles were again normalised to \( \omega_n = 1 \) as shown in Fig 4.2.0. Having normalised the open loop poles to \( \omega_n = 1 \), the complex closed loop poles then lie along the damping line \( \zeta = 0.63 \).
Fig 4.20 Family of Normalised Root Loci for the system given in Fig 4.23, showing the open loop pole positions and values of delay $\tau_N$ which give rise to the desired set of in-line closed loop poles.
Fig 4.21 Impulse responses of the system given in Fig 4.23 for the case when $\zeta = 0.3$, and $\tau_N = 2.05$ seconds.

Fig 4.22 Step responses of the system given in Fig 4.23 for $\zeta = 0.3$, and $\tau_N = 2.05$ seconds.
In order to verify that the dominant mode approach gives the correct time response, the closed loop system was simulated using the values $\tau_N = 2.05$, $\zeta = 0.3$, and $K_N = 0.165$ obtained from the root locus. The results of the simulation are shown in Figs 4.2.1 and 4.2.2 which also illustrate the effect of using values of $K_N = 0.1$ and $K_N = 0.25$. From these records it can be seen that the value of $K_N = 0.165$ does in fact give the predicted non-negative impulse response and the system has the fastest possible step response.

It was found that the aligned closed loop pole condition gave a valid approximation to the transient response for values of $\tau_N$ up to about 2.5 seconds. And for the normalising factors used in this case the maximum delay represents approximately 24% of the period of the dominant closed loop poles.

The corresponding values of $K_N$ and $\zeta$ which give the fastest step response for this range of values of $\tau_N$ were obtained from the root loci and are plotted in the $(K_N, \zeta)$ plane given in Fig 4.2.4. For values of $\tau_N > 2.5$, the response contains a large proportion of the higher frequency modes and is no longer dominated by the three closed loop poles on the primary branches of the locus.

Although the author's method is only valid for a limited range of values of $\tau_N$ it does, however, give the required closed loop performance without increasing the steady state error produced by external disturbances. And if necessary the range of values of $\tau_N$ could be extended by specifying the optimum closed loop pole-zero pattern including the poles on the first delay branch of the locus.

An estimate of the range of values of $\tau$ for which a pure time delay system response will be dominated by the modes on the primary branches of the locus can be obtained as follows.
Fig 4.23 Normalised Third Order Type I system with a delay of $\tau_N$ seconds in the forward path.

Fig 4.24 Relationship between $K_N$, $\tau_N$ and $\zeta$ for the optimum step response of the system shown in Fig 4.
From the general equation of a pure time delay root locus

\[-1 = K e^{-s \tau} G(s)\]  \hspace{1cm} \text{4.6.4}

it can be shown that the horizontal asymptotes of the extra delay branches are given by

\[\omega = \frac{\pi}{\tau} \left[(2k + 1) - (N-m)\right]\]  \hspace{1cm} \text{4.6.5}

where \((N-m)\) is the excess of finite open loop poles to zeros.

Hence for a system with an even excess the first delay branch will lie in the range

\[\frac{3\pi}{\tau} > \omega > \frac{\pi}{\tau}\]

while for a odd excess it occurs in the range

\[\frac{4\pi}{\tau} > \omega > \frac{2\pi}{\tau}\]

It has been found that the modes on these branches have a negligible effect on the transient response provided they have a frequency at least five times that of the dominant mode on the primary branch of the locus.

Applying these rules to the third order type I system considered here gives

\[\frac{2\pi}{\tau_N} > 5 \times 0.6\]

from which \(\tau_N < 2.1\) seconds.
It has been shown that the transient response of a pure time delay system can be predicted directly from its root locus, and there is now no need to represent the effect of the delay by a large number of poles and zeros. Since the closed loop poles are the true modes of the system it is now possible to identify the open loop dynamics from measurements of the closed loop response.

Standard methods of improving the performance of a pure time delay system do not remove the delay in the output, but simply try to reduce its destabilising effect on the closed loop system. While this appears to be a reasonable approach to the problem, the author has shown that these methods can also greatly increase the steady state error to external disturbances applied to the system.

A great advantage of the author's search vector method is that it can also be used to plot the $0^\circ$ root loci of pure time delay systems. Using this facility the designer can now specify the closed loop pole-zero pattern to obtain a desired transient response following the delay, and then determine the required open loop transfer function directly from the root locus. This technique has been illustrated for a third order type I system which was required to have the fastest possible step response.
5.1 Introduction

The ability to accurately include the effects of pure time delay now raises the exciting possibility that one could for instance, consider human pilot control of an aircraft. Before work of this type can begin however, it is first necessary to develop a method of representing the performance of an aircraft in terms of poles and zeros.

The aim of this chapter is to produce a general analytical technique which will enable root locus methods to be used to determine the stability and performance of an aircraft which has any form of automatic control. The technique is then used to design an automatic pilot for the pitch attitude control of a typical aircraft.

5.2 Aircraft Transfer Functions

5.2.1 Generalised Equations of Motion

The response of an aircraft to external forces and moments can be described by six simultaneous differential equations (27). These equations are only valid for small changes about the frame of reference which is normally taken as a set of body axes through the centre of gravity of the aircraft. The six degrees of freedom associated with the motion of the aircraft about these axes are illustrated in Fig 5.1 and Fig 5.2 which also show the positive sense of the externally applied forces and moments.
The coefficients of the equation of motion are aerodynamic derivatives which can either be obtained from the geometry of the airframe or from wind tunnel tests on the scaled model. More recently they have been measured directly in-flight using correlation techniques employing pseudo random signals as the input disturbance.\(^{(28)}\) The relationships between the aerodynamic derivatives and the aircraft state variables are shown in Table 5.1. Provided there is no substantial cross coupling between the symmetric and antisymmetric modes of the aircraft, the equation of motion can be divided into two groups of three which separately describe its lateral and longitudinal stability.

The Laplace transform of each group of equations can be written in the generalised form

\[
\begin{align*}
  a_{11}(s) x + a_{12}(s) y + a_{13}(s) z &= D_1(s) \\
  a_{21}(s) x + a_{22}(s) y + a_{23}(s) z &= D_2(s) \\
  a_{31}(s) x + a_{32}(s) y + a_{33}(s) z &= D_3(s)
\end{align*}
\]

where \(x, y\) and \(z\) are the components of the state variable, \(a_{ij}(s)\) are functions of the aerodynamic derivatives, and \(D_i(s)\) are the applied forces and moments.

5.2.2 Stick Fixed Stability

Assuming that the aircraft is flying straight and level with the control surfaces locked, the equations given in section 5.2.1 enable the aerodynamists to investigate the recovery of the aircraft to external disturbances such as sudden gusts or finite pressure discontinuities.
### Table 5.1 Aerodynamic Derivatives

<table>
<thead>
<tr>
<th>Air Reactions</th>
<th>Forces</th>
<th>Momenta</th>
<th>Velocities</th>
<th>Accelerations</th>
<th>Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>p</td>
<td>q</td>
</tr>
<tr>
<td>X</td>
<td>X_u</td>
<td>X_v</td>
<td>X_w</td>
<td>X_q</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>Y_u</td>
<td>Y_v</td>
<td>Y_p</td>
<td>Y_r</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>Z_u</td>
<td>Z_v</td>
<td>Z_p</td>
<td>Z_q</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>L_u</td>
<td>L_v</td>
<td>L_p</td>
<td>L_r</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>M_u</td>
<td>M_v</td>
<td>M_p</td>
<td>M_q</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>N_u</td>
<td>N_v</td>
<td>N_p</td>
<td>N_q</td>
<td></td>
</tr>
</tbody>
</table>

- Derivative zero due to symmetry.
- Blank space indicates that the derivative is usually neglected.
The response of the aircraft in \( z \) say to a disturbance of \( D_i(s) \) is given by

\[
z(s) = \sum_{i=1}^{i=3} \frac{D_i(s) P_i(s)}{F(s)}
\]

where \( P_i(s) \) are the numerator polynomials and \( F(s) \) is the characteristic polynomial formed from the aerodynamic derivatives.

Since the corresponding time function is obtained from the inverse Laplace Transform it can be seen that the roots of \( F(s) \) determine the modes of the response and hence the stick fixed stability of the aircraft.

The aircraft manufacturer is only interested in stick fixed stability; since evaluation of \( F(s) \) throughout the operational range of the aircraft enables him to obtain the stability envelope. While this information is useful to the control engineer, he is however, concerned with the complete aircraft response which also requires a knowledge of the numerator polynomials \( P_i(s) \). There are many of these polynomials each relating an output variable to one of the input forces or moments. Prior to 1961 these polynomials had not been determined algebraically and they were first published by the author in 1962 (29). They are now widely used by Government Research Establishments and Industrial firms specialising in the design of autopilots and autostabilisers.
5.2.3 Methods of Including Generalised Autopilot Control Laws

Following the development of the aircraft numerator polynomials the author produced an original method of introducing generalised autopilot control laws into the driving functions $D_i(s)$. This resulted in a general technique which gave the characteristic equation in a form which enabled root locus techniques to be used directly to design the multiloop system.

The method used was to write $D_i(s)$ as

$$D_i(s) = G_i(s) + C_{i1}(s)x + C_{i2}(s)y + C_{i3}(s)z$$  \hspace{1cm} (5.2.5)

where $G_i(s)$ are externally applied forces or moments such as gusts etc., and $C_{ik}(s)$ are generalised control terms.

The control functions are then transferred and added to the coefficients of the relevant state variables. Hence the response of the system in $z$ say, now becomes

$$z(s) = \frac{1}{\Delta_0(s)} \sum_{i=1}^{3} P_i(s)G_i(s)$$  \hspace{1cm} (5.2.6)

And the stability of the closed loop system is then determined by the polynomial $\Delta_0(s)$ which can be expressed in the form

$$\Delta_0(s) = F(s) + \sum C_i(s)A_i(s) + \sum C_i(s)C_k(s)A_{ik}(s)$$  \hspace{1cm} (5.2.7)

Where $F(s)$ is the original stick fixed stability polynomial and $A_i(s)$ is a polynomial arising from the position of the control term $C_i(s)$ in the array of coefficients. The remaining term is due to cross-coupling between the control loops.
In practice it is found that control function interaction is either zero or exerts negligible effect on the main loops. The effect may, however, be included in the analysis if desired using an iterative technique to be described later. The polynomials $F(s)$, $A_1(s)$ and $A_{ik}(s)$ are completely independent of the control functions and a digital computer has been programmed to facilitate their numerical evaluation. Once these are established, the most suitable set of linear control functions for a specific flight condition can be determined from:

$$F(s) + C_1(s)A_1(s) + C_2(s)A_2(s) + C_3(s)A_3(s) = 0$$

5.2.8

The complete solution of the above equation is obtained by closing the control loops in a logical order, determined mainly from a knowledge of control systems and aerodynamics.

The inner loop control function $C_1(s)$ is then obtained from the root locus solution of the equation

$$F(s) + C_1(s)A_1(s) = 0$$

5.2.9

Once these poles have been determined the control law $C_2(s)$ is then obtained from the equation

$$[F(s) + C_1(s)A_1(s)] + C_2(s)A_2(s) = 0$$

5.2.10

This is solved using the root locus method by arranging it in the form:

$$\frac{C_2(s)A_2(s)}{[F(s) + C_1(s)A_1(s)]} = -1$$

5.2.11
Continuation of the above process gives the remaining control functions, which are chosen so that the response of the complete system contains suitable modes.

The analysis is repeated to cover various steady state manoeuvres and a suitable range of speed, altitude, and position of the centre of gravity. It is then possible to choose the optimum control functions for a given channel consistent with the above conditions, and determine the programme of loop gains required as functions of indicated airspeed, mach number or altitude.

The general block diagrams for the lateral and longitudinal channels expressed in terms of the author's polynomials are shown in Figures 5.3 and 5.4.

The lateral channel shows the general control loop required to maintain heading $\psi$ and roll attitude $\phi$ using rudder and aileron deflections $\zeta$ and $\xi$ respectively. In the case of the longitudinal channel the diagram illustrates the elevator control $\eta$ required to maintain desired pitch attitude $\theta$ and height $H$.

Many other forms of control could be used in both channels and these can be investigated by generating the required numerator polynomials between the appropriate control surface and the controlled variable.

As an illustration of the author's method we will now obtain the open loop transfer functions relating the changes in pitch attitude and height of an aircraft to changes in the position of the elevator.
FIG 5-3 LATERAL CHANNEL SHOWING THE FEEDBACK LOOPS REQUIRED TO CONTROL THE ROLL ATTITUDE AND HEADING OF AN AIRCRAFT USING AILERON AND RUDDER

FIG 5-4 LONGITUDINAL CHANNEL SHOWING ELEVATOR CONTROL OF HEIGHT AND PITCH ATTITUDE
5.3 Longitudinal Channel

5.3.1 Stability Polynomials for elevator control of pitch attitude and height

The non-dimensional equations of motion in the longitudinal channel are given by

\[(s-x_u)\ddot{u} - x_w\ddot{w} + \left( k_L - \frac{s\mu}{\mu_1} \right) \theta = D_x(s) \quad 5.3.1 \]

\[-x_u\ddot{u} + (s-z_w)\ddot{w} - \left[ s\left( 1 + \frac{z_q}{\mu_1} \right) \right] \theta = D_z(s) \quad 5.3.2 \]

\[-m_u\ddot{u} - \left( \frac{sm_w}{\mu_1} + m_w \right) \ddot{w} + \left( \frac{s^2I_B}{\mu_1} - \frac{sm_q}{\mu_1} \right) \theta = D_m(s) \quad 5.3.3 \]

When \( s \) is the Laplace operator normalised with respect to aerodynamic time and the incremental speed changes \( \ddot{u} \) and \( \ddot{w} \) are normalised with respect to forward speed \( U \).

From these equations the stick fixed stability polynomial expressed in real time is given by

\[ F(s) = s^4 + \left[ -(x_u + z_w) - \frac{q}{I_B} - \left( 1 + \frac{q}{\mu_1} \right) \frac{w}{I_B} \right] s^{3t-1} \]

\[ + \left[ x_w (z_u - x_u \ddot{z}_w) + \frac{m_u}{I_B} (x_u + z_w) + \frac{m_w}{I_B} \left( x_u (l + \frac{z_q}{\mu_1}) - x_w \frac{z_u}{\mu_1} - k' \right) \right] s^{2t-2} \]

\[ - \frac{m_w}{I_B} \left( \mu_1 + \frac{z_q}{\mu_1} \right) - x_u \frac{m_u}{I_B} \left[ z_w \mu_1 \right] + \left[ - \frac{m_q}{I_B} (x_u z_w - x_w z_u) \right] \]

\[ + \frac{m_{\mu_1}}{I_B} \left( k_L z_u + k' x_u \right) + \frac{m_w}{I_B} \left( x_u (u_1 + z_q) - x_w z_u - \mu_1 k' \right) \]

\[ + \frac{m_{u_1}}{I_B} \left( \mu_1 k_L - x_w (u_1 + z_q) + x_q z_w \right) \left[ s^{t-3} \right] + \left[ \frac{m_{u_1}}{I_B} (k_L z_u + k' x_u) \right] \]

\[ - \frac{m_{u_1}}{I_B} \left( k_L z_w + k' x_u \right) \left[ s^{t-4} \right] \quad 5.3.4 \]
In general the roots of $F(s)$ are complex and can be expressed in the form

$$(s^2 + 2\tau_1 \omega_1 s + \omega_1^2)(s^2 + 2\tau_2 \omega_2 s + \omega_2^2) = 0$$ 5.3.5

The quadratics are normally widely separated in frequency and determine the stability of the phugoid and short term modes of the aircraft response.

Elevator control of pitch attitude can be investigated directly by including the generalised control function $C_1(s)$ into the driving functions of the equation of motion. To achieve a suitable elevator height control function, however, the change in height must first be expressed in terms of the state variables $\theta, w$ and $U$ as follows

$$\dot{h} = W \cos \theta - U \sin \theta$$ 5.3.6

Considering only small changes in $\theta$, this equation can be written as

$$\ddot{h} = \frac{1}{s} (\ddot{w} - \theta)$$ 5.3.7

The driving functions for elevator control of height and pitch attitude are therefore given by

$$D_x(s) = X(s)$$ 5.3.8

$$D_2(s) = Z(s) + \frac{z}{m} \left[ C_1(s) - \frac{1}{s} C_2(s) \right] \theta + \frac{z}{m} \frac{1}{s} C_2(s) \dot{w}$$ 5.3.9

$$D_m(s) = m(s) + \left[ C_1(s) - \frac{1}{s} C_2(s) \right] \theta + \frac{1}{s} C_2(s) \ddot{w}$$ 5.3.10

where $C_1(s)$ and $C_2(s)$ are the generalised control functions shown in Fig 5.4. The stability of the system with both attitude and height loops closed is then defined by the characteristic equation.
\[ \dot{\theta}(s) = \frac{1}{s} \left[ \mu_1 \left( \frac{z}{m_n} + \frac{z}{m_w} \right) \right] + \frac{1}{s} \left[ \mu_1 \left( \frac{z}{m_n} \right) \right] \cdot \dot{t} \]

where

\[ K_1 \theta(s) = -\frac{1}{i_B} \left[ \mu_1 \left( \frac{z}{m_n} \right) \right] + \frac{1}{s} \left[ \mu_1 \left( \frac{z}{m_n} \right) \right] \cdot \dot{t} \]

and

\[ K_2 H(s) = \left[ -\frac{z}{m_n} \right] s^3 + \left[ -\frac{z}{m_n} \left( \frac{z}{m_n} \right) \right] \cdot \dot{t} \]

5.3.2 Root Locus Equations and Order of Loop Closure

Unlike the Lateral Channel shown in Fig 5.3, the order of loop closure in the Longitudinal channel is comparatively simple. The prime consideration in this case is to achieve a good short term pitch attitude control.

A suitable autopilot attitude control function \( C_1(s) \) can therefore be obtained by plotting root loci defined by the equation

\[ -1 = \frac{K_1 C_1(s) \theta(s)}{Kt^2 F(s)} \]

Since the pole-zero pattern of the aircraft varies with speed and attitude, etc., many root loci must be plotted in order to determine the variation of the control function parameters that are required to give an acceptable response over a wide range of flight conditions.
Once the control function $C_1(s)$ has been determined, the height lock control $C_2(s)$ is then obtained from the root locus plots defined by the equation

$$-1 = \frac{K_2 C_2(s) H(s)}{s[K_1 F(s) + K_1 C_1(s) \phi(s)]}$$  \hspace{1cm} 5.3.15$$

Here again many loci must be plotted to determine the variation required in $C_2(s)$ throughout the flight envelope.

5.4 Design of an autopilot to control pitch attitude and height

5.4.1 General requirements and basic system

The main function of the autopilot in this case is to maintain the desired short term pitch attitude of the aircraft using an acceptable transient response.

Experience has shown that passengers prefer a transient with a natural frequency of 3 to 5 rad/sec and a damping ratio of 0.5 to 0.8.

The object of controlling the long term phugoid mode (period 20-40 seconds), is normally to improve its damping. This is particularly important when the autopilot is coupled into the glide path signal to form part of an automatic landing system.

The requirement in height control is again one of well damped short term and long term transients. Height lock is particularly important to modern civil airliners which are now required to fly in air corridors at well defined altitudes.

The basic control loops used to maintain height and pitch attitude of a typical aircraft are shown in Fig 5.5. Pitch attitude is normally measured by integrating the signal from the
FIG 5.5 ELEVATOR CONTROL OF HEIGHT AND PITCH ATTITUDE
rate gyro, while height error is obtained from an aneriod capsule. The error signals from these sensors are fed into the autopilot computer and the control loops are closed via an actuator which is used to adjust the position of the elevator.

Since the gain and pole-zero patterns of the aircraft vary with speed, altitude, etc., the gain and pole-zero patterns of the control laws $C_1(s)$ and $C_2(s)$ should ideally be fully adaptive in order to maintain the desired transient performance over the complete flight envelope. Adjusting the pole-zero patterns of the autopilot is, however, very costly and cannot normally be justified for subsonic aircraft.

In practice the loop gain is varied with indicated airspeed and the pole-zero patterns of the control laws are chosen to give satisfactory transient performance over the widest possible range of flight conditions.

5.4.2 Choice of Autopilot Control Laws for a typical aircraft

We will now consider the choice of the control function $C_1(s)$ for a particular aircraft whose short period natural mode varies with air speed as shown in Fig 5.6. The design will be illustrated by considering the stability and performance of the aircraft at its cruising speed (500 ft/sec) and normal operating altitude of 10,000 feet.

Under these conditions the open loop aircraft response is given by

$$\frac{\dot{\phi}}{\eta} = \frac{67.2 s (s + 0.63)(s + 0.0183)}{(s^2 + 2.2 s + 8.48)(s^2 + 0.00193 + 0.0035)}$$ 5.4.1
Fig 5.6 Airspeed variation of the short period poles of a typical aircraft.

Fig 5.7 Required variation of pitch attitude control function gain $K_1$ with indicated airspeed.
Frequency response measurements on the actuator showed that it could be approximated by the transfer function

\[
\frac{A}{\eta} = \frac{7.7}{(s + 7.7)}
\]

By plotting families of loci to cover a wide range of flight conditions it was found that a satisfactory pitch response could be obtained using a control function

\[
C_1(s) = K_1 \frac{s + 1.6}{s} \frac{(s+3)}{(s+8)}
\]

where \(K_1\) is a gain parameter which varies with indicated air speed as shown in Fig 5.7.

The closed loop poles of the short term response are then well damped as can be seen from root locus given in Fig 5.8.

Flight trials using this control law showed that while the short term response was satisfactory the aircraft failed to hold long term pitch attitude due to the random drift in the rate gyro.

This fault was overcome without effecting the short term response by including a transientising term in the control law as follows

\[
C_1(s) = K_1 \frac{(s + 1.6)}{s} \frac{(s+3)}{(s+8)} \frac{s}{(s + 0.05)}
\]

The effect of this form of control on the damping ratio and frequency of the cruising speed pitch phugoid is shown in Fig 5.9.

As the airspeed decreases, the short period aircraft mode decreases in frequency and becomes less well damped. While the frequency and damping ratio of the phugoid are virtually independent of speed over the normal operating range.
Root Locus solution of

\[ -1 = \frac{K_1 (s+1.6)(s+3)(s+0.63)(s+0.0183)}{(s+8)(s+7.7)(s^2+2.2s+8.45)(s^2+0.019s+0.0035)} \]

\[ K_1 = 31.8 \]

Fig 5.8 Short term pitch attitude control at normal cruising speed (500 ft/sec) and height (10,000 ft.).
Root Locus Solution of

\[-1 = \frac{0.0036 K_1 s(s+0.0183)}{(s+0.05)(s^2+0.019s+0.0035)}\]

Fig 5.9 Long term pitch attitude control at normal cruising speed and height.
The low speed limit for the use of pitch attitude control occurs slightly above the minimum drag speed when the aircraft polynomial \( F(s) \) contains a real pole near to the origin in the right hand half plane. Under these conditions, the closed loop system also contains a divergent exponential mode and the aircraft gradually loses height. It has been found that fixed pole-zero compensation gives satisfactory performance over a speed ratio of 4:1 and this normally defines the upper speed limit for pitch attitude control. More recent aircraft such as Concord have a speed ratio of at least 8:1 and will therefore require a fully adaptive autopilot.

Having obtained satisfactory attitude control, the height lock control function \( C_2(s) \) is then determined from the root locus equation

\[
+1 = \frac{H(s) K_2 C_2(s) C_{1z}(s)}{A(s)}
\]

where \( H(s) = 0.498 (s + 15.1)(s - 13.6)(s + 0.0121) \), \( C_{1z} \) are the zeros of the control function \( C_1(s) \), and \( A(s) \) is the seventh order polynomial formed from the closed loop poles of the inner pitch attitude loop.

It should be noted that in this case we require the \( 0^\circ \) root locus due to the non-minimum phase relationship between change in height and movement of the elevator. This can also be seen from the position of the right hand half plane zero in the polynomial \( H(s) \).

Families of root loci were again plotted and it was found that an acceptable height lock could be achieved over a wide range using the control law

\[
C_2(s) = K_2 \frac{(s + 0.2)}{(s + 1)}
\]
Root Locus Solution of

\[
+1 = \frac{0.498 K_2(s+15.1)(s-13.6)(s+0.0121)(s+8)(s+0.2)(s+0.05)}{s(s^2+0.148)(s^2+0.036s+0.00096)(s^2+4.2s+9.9)(s^2+13.5s+68.1)}
\]

Fig 5.10 Short term modes of the height lock control system.
Fig 5.11 Improvement in aircraft phugoid damping due to long term height lock control.
It was found that the basic airspeed law required for \( K_2 \) could be made the same as that used for the attitude loop, provided the ratio of the static error sensitivity was set to 0.05 degrees per foot. The root loci given in Fig 5.10 and 5.11 show that while the height lock control has virtually no effect on the short term modes of the attitude loop, it does however, produce a considerable improvement in the damping of the phugoid.

The limiting case for the use of height lock occurs when the aircraft is flying below the minimum drag speed. Under these conditions the polynomial \( H(s) \) has another zero in the right hand half plane very close to the origin. This zero together with the open loop pole at the origin gives rise to a divergent closed loop mode which shows that the system is no longer able to maintain the desired altitude.

5.5 **SUMMARY**

A general analytical method has been developed which enables root locus techniques to be used for the design of multiloop aircraft control systems. When used in conjunction with the author's computer program, the root loci for the complete flight envelope of a typical aircraft, can now be plotted in about 15 minutes compared with several weeks previously required by manual methods.

Flight trials have shown that the design procedure gives satisfactory control laws and the method has now been widely adopted by autopilot manufactures and government research establishments.
6.1 Introduction

Having developed a general root locus method for aircraft control system design and a computer program which can deal with pure time delay root loci, we are now in a position to consider the stability of aircraft controlled by a human pilot. Since we are also able to identify the dynamics of pure time delay systems, it should now be possible to obtain the transfer function of a pilot from measured flight records.

Various methods of representing the control action of the pilot are considered and it is shown that a linear continuous model is the most suitable form for parameter identification. The author then explains how root locus methods can be used to identify the parameters of the model from the experimentally determined positions of the short period closed loop poles.

Finally the validity of the model is established by its ability to accurately predict the frequency of the pilot induced oscillation which occurs during elevation tracking.

6.2 Formulation of a mathematical model for the human pilot

The transfer function of the pilot in this context is the relationship between the observed error signal as the input and the displacement of the control column as the output. The transfer function cannot simply be described as linear, non-linear, discrete or even random. His output includes all of these characteristics and is also a function of his psychological and physiological condition.
When the pilot is performing a single concentrated task such as target tracking, there are however, many features of his control action which can be represented by either a continuous or a pulsed transfer function.

We will now consider both methods of representing the control action of the pilot and compare them with respect to their ability to explain human pilot control of a V.T.O.L. aircraft during hover.

6.2.1 Linear Continuous Pilot Transfer Functions

The essential features of this type of model have been investigated by McRuer and Krendel (30) who have shown that the pole-zero pattern for the pilot can be expressed in the form

\[
\begin{align*}
\frac{\text{Pilot output movement}}{\text{Pilot visual input}} &= \frac{K_P e^{-sT_T}}{(1 + sT_P)(1 + sT_N)} \\
&= \frac{K_P e^{-sT_T}}{s^2 + (T_P + T_N)s + T_P T_N}
\end{align*}
\]

The pure time delay of \(\tau\) seconds is due to the finite time taken by the pilot to react to an input signal. Its value has been measured by many different methods most recent of which is that due to Mitchell (31) who gives a value of 0.2 seconds ± 20%.

In practice the reaction time of the pilot is his main destabilising effect in a control loop and it imposes an upper limit of about 3 c/s on his ability to follow rapidly varying signals. The neuromuscular lag time constant \(T_N\), represents the time required for movement to begin in the pilots arm after the motor nerve of the muscle has been stimulated. This has again been measured by Mitchell who gives the value of \(T_N\) as 0.1 seconds ±20%.

Reaction time delay and neuromuscular lag are both characteristics of the pilots response over which he has no control. They are approximately constant for a particular pilot under normal conditions but they are known to increase slightly with fatigue.
Perhaps the most interesting feature of the pilots control action is his ability to adjust some of the parameters of his transfer function in order to optimise the overall performance of the control loop. To achieve this, the pilot varies his anticipation $(1 + ST_A^2)$ gain $K_p$, and lag $l/(1 + sT_L)$ so that the system operates as near as possible to his own satisfaction.

The ability to introduce a lead term depends on the pilots experience of the particular control task and can vary considerably from person to person. It will be shown later in this chapter that the upper limit of the lead time constant $T_A$ is about 2 seconds.

Shinners (32) has demonstrated that the lag time constant $T_L$ used by the pilot depends on the frequency of the input signals. When the input frequency is low, the pilot adjusts $T_L$ to about 20 seconds and uses the lag term as an integrator to remove steady state error. While for high input frequencies the value of $T_L$ is set to about 0.07 seconds and the pilot uses the lag term for error smoothing.

The remaining parameter is the pilot gain $K_p$. Its physical significance in this case is the ratio of the stick movement made by the pilot to his visual error signal obtained from ground reference or some form of head up display.

Laboratory experiments by Shinners (32) suggest that $K_p$ can vary by a factor of 100:1 which indicates that the pilot is capable of making significant changes in loop gain.

6.2.2 Sampled Data Models

Following the development of Z transform theory for the analysis of sampled data systems there have been several attempts to represent the control action of the pilot by means of a discrete model. The most notable of these have been the models produced by Professor Westcott (33)
and more recently that due to Lange (34). While these models vary in complexity and sophistication they are all based on Bekey's original pulsed transfer function (35)

\[ P(z) = Z \left[ \frac{1 - e^{-sT}}{s} \frac{K}{1 + sT_N} \right] \]

6.2.2

He assumes that the pilot can be represented by a sampler which operates every 200 or 300 milliseconds followed by a zero order hold. The output of the hold circuit then operates the motor nerves which introduce a neuromuscular lag of \( T_N \) seconds.

The pilots ability to adapt his control action is then attributed to his variation of gain \( K \), and his sampling frequency \((1/T)\).

We will now compare the sampled data model with the continuous pilot transfer function in a control loop which the pilot is required to maintain pitch attitude of a V.T.O.L. aircraft during hover.

6.2.3 Comparison of discrete and continuous pilot models for predicting human pilot control of an unstable V.T.O.L. in hover

The basic system used for manual control of a V.T.O.L. aircraft during hover is illustrated by the block diagram shown in Fig 6.1. In this case the pitching moments are obtained from small puffer jets mounted on the nose and tail of the aircraft. The thrust of these jets is determined by the position of the elevator which is operated from the control column in the same way as that for conventional flight.

The control system can now be analysed using either a sampled data model for the pilot as shown in Fig 6.2 or a linear transfer function as illustrated in Fig 6.3. Consider first the closed loop performance predicted using the sampled data system.
Fig 6.1 Manual Control of a V.T.O.L. aircraft during hover.

Fig 6.2 Sampled data system for the human pilot pitch attitude control loop shown in Fig 6.1.

Fig 6.3 Continuous pilot model for the V.T.O.L. control system.
Fig 6.4 Z transform root locus plots showing the effect of varying the pilot sampling frequency on the stability of the discrete human pilot control system given in Fig 6.2.
For a sampling period of 0.3 seconds, the open loop pulsed transfer function is given by

\[
P(z) = \frac{Kz(z^2 + 0.94z + 0.25)}{(z - 1)^2 (z - 0.12)(z - 0.05)}
\]

The corresponding root locus in the z plane is shown in Fig 6.4 together with the effect of varying the sampling period to 0.2 seconds.

From these loci it can be seen that the deeper concentration associated with a higher sampling rate improves the stability of the overall system. The loci also shows that even at high sampling rates, the system will be unstable at low loop gain. This agrees with practical observations since under these conditions the system is virtually open loop.

**Continuous pilot model**

The open loop transfer function for the continuous pilot model of the V.T.O.L. hover control system is given by

\[
P(s) = \frac{48.3 K(s + 1/T_a) e^{-0.2s}}{s^2 (s + 10)(s + 15)(s+7)}
\]

Where the reaction time delay and neuromuscular lag have been taken at their nominal values. It has also been assumed that under these conditions the pilot uses his lag term for error smoothing.

The effect of varying the pilots anticipation on the stability and performance of the system can be seen from the family of root loci given in Fig 6.5. These loci show that in order to fly an aircraft of this type the pilot must be capable of introducing a phase advance equivalent to a zero in the range -2.0 to -0.5. They also suggest that the aircraft will become more difficult to fly as the pilot zero moves further away from the origin due to fatigue.
Fig 6.5  Root loci for the continuous human pilot control system given in Fig 6.2, showing the effect of varying the pilot's anticipation time constant ($1/T_A$).

\[ -1 = \frac{Ke^{-0.2s}(s + \frac{1}{T_A})}{s^2(s + 7)(s + 10)(s + 15)} \]

1. $T_A = 2$ sec
2. $T_A = 1$ sec
3. $T_A = 0.5$ sec
4. $T_A = 0.16$ sec
These observations are in agreement with practical experience, and it has been found that aircraft of this type cannot be flown in hover for long periods unless they are fitted with an auto-stabiliser such as that designed by the author for the Harrier (29).

Comparison of the results

The results of these investigations show that while both models explain the pilots' ability to fly this unstable aircraft, the continuous system gives a clearer understanding of his control action. His transfer function parameter values are also more easily obtained in this case due to the simpler relationship between transient response and the position of the closed loop poles.

When sampled data techniques were applied to the high order equation of the conventional aircraft, the mathematics became very complex and difficulty was experienced in deriving the open loop transfer function. Although the author has recently been able to simplify the evaluation of the time response of these systems (36) it soon becomes clear that sampled data models were unsuitable for the analytical determination of pilot response. It was therefore decided to adopt the linear continuous model and evaluate the parameters using pure time delay root loci.

We will now consider the authors method of determining the transfer function of the pilot when flying a conventional aircraft at normal cruising speed and height.

6.3 Determination of Pilot Transfer Function For a Specific Flight Manoeuvre

6.3.1 Measurement of Pilot-Aircraft response

The basic control system used for manual control of pitch attitude in a typical aircraft is shown in Fig 6.6. It is essentially
Fig 6.6 Human pilot control of aircraft pitch attitude

A = aircraft pole or zero
P = pilot pole or zero

Note: The remaining pilot parameters are taken at their nominal value $\tau = 0.2$ sec, $T_L = 0.07$ sec, and $T_N = 0.1$ sec.
the same as the system shown in Chapter 5, except that the autopilot is now replaced by a human pilot. The position of the closed loop poles of the pilot-aircraft loop were obtained by recording the transient from the rate gyro following a sudden command to the pilot to change the pitch attitude of the aircraft by 5°. This process was repeated about 40 times using the same pilot under the same set of flight conditions. The damping ratio and natural frequency of the short period mode were then calculated from the average values of these records and were found to be 0.28 and 4.05 rad/sec respectively.

Since the performance of the aircraft under these conditions is known, the transfer function of the pilot can now be determined from the accurate pure time delay root loci of the complete system.

6.3.2 The effect on the closed loop performance of varying the parameters of the pilot transfer function

Families of root loci are now plotted to determine the possible sets of pilot parameter values which could have been used to give rise to the measured closed loop poles of the short period response. Since the most significant parameter of the pilot transfer function is his lead term we will first try to locate the approximate position of this zero assuming that the other parameters were set at their nominal values.

Pilot Lead Term

The effect of varying the pilot lead term on the possible positions of the closed loop poles is illustrated by the family of root loci shown in Fig 6.7. Only the branches from the short period aircraft poles have been plotted and these clearly demonstrate that the pilot must have used an amount of anticipation equivalent to a zero between -1.0 and -0.5.
Reaction Time

Variation of the pilots reaction time delay between its known limits of 150 and 250 milliseconds has only a small effect on the possible closed loop modes as can be seen from the loci given in Fig 6.8 (a). It was therefore taken at its nominal value of 200 milliseconds to allow for the case of an average pilot operating under conditions of strain and high work load.

Neuromuscular Lag

This parameter is also fairly well defined and the effect of varying it over the known limits of 0.09 to 0.125 seconds is shown in Fig 6.8 (b). Since this variation produces a negligible effect in the region of the measured closed loop poles, the neuromuscular lag was taken as the average value of 0.1 seconds.

Lag Term

As explained in section (6.2.1), the pilot can either use the lag term as an imperfect integrator (TL = 10 seconds) or as an error smoothing (TL = 1/15 seconds). The effect of these two extreme values on the possible positions of the closed loop poles is illustrated in Fig 6.8 (c). From these loci it can be seen that the pilot must have used the lag for error smoothing. Once this had been established the lag time constant TL was taken as (1/15 seconds).

6.3.3 Determination of the pilot transfer function

Taking the delay, smoothing error time constant, and neuromuscular lag at their nominal values, families of root loci were then plotted to obtain the accurate position of the pilots lead term zero.
Root locus equation

\[-1 = \frac{K e^{-\tau(1 + sT_A)}(s + 1.12)}{s (s^2 + 3.9s + 15.2)(s + 7)(1 + sT_L)(1 + sT_N)}\]

(a) Reaction time delay \(\tau\) seconds

(b) Neuromuscular Lag \(T_N\)

(c) Lag Time Constant \(T_N\)

Fig 6.8 The effect of varying the pilot parameters on the short period modes of the closed loop system.
From these plots it was found that the locus passed through the measured closed loop poles when the zero was at -0.5 as shown in Fig 6.9. The pilot transfer function computed from this locus was then found to be

\[
P(s) = \frac{0.16 (1 + 2s) e^{-0.2s}}{(1 + 0.1s)(1 + 0.07s)} \text{ Inches/degree} \quad 6.3.1
\]

Since most authors define the working range of the pilot gain as 1 to 100 it was decided to write equation 6.3.1 in the form

\[
P(s) = \frac{K_p}{60} \frac{(1 + 2s) e^{-0.2s}}{(1 + 0.1s)(1 + 0.07s)} \text{ Inches/degree} \quad 6.3.2
\]

So that the value of \( K_p \) for this particular flight manoeuvre is 10 and the system becomes unstable for \( K_p = 62 \). Expressing equation 6.3.2 in pole-zero form, the human pilot transfer function becomes

\[
P(s) = \frac{5 K_p (s + 0.5) e^{-0.2s}}{(s + 10)(s + 15)} \quad 6.3.3
\]

6.4 Pilot induced Instability during Elevation Tracking

6.4.1 Prediction of the oscillation frequency using root locus

If the proposed pilot model and parameter values are a reasonably accurate description of the pilot's control action, then it should be possible to predict the performance of the pilot aircraft system under the same conditions for a different flight manoeuvre. It can, for instance be seen from the root locus given in Fig 6.9, that if the pilot increases his gain to \( K_p = 62 \) the attitude control loop would oscillate at a frequency of 5.0 radians per second.
Fig 6.9 Determination of pilot gain $K_p$.

Fig 6.10 Typical pitch attitude error during elevation tracking.
6.4.2 Verification of the pilot transfer function from measured flight records

In order to try and produce this condition of high gain, the pilot was asked to track a ground target using a fixed aiming mark on the aircraft. As expected the pilot encountered some difficulty in maintaining an exact coincidence between the marker and the target.

Film records of the tracking error did in fact reveal the presence of a small oscillation as shown in fig 6.10. The average periodicity of the fundamental oscillation obtained from these records was found to be 4.9 radians per second, which agrees to within 2% with that predicted from the root locus.

6.5 Summary

The original techniques developed in this chapter show that it is now possible to use root locus methods to determine that parameters of a human pilot transfer function from measured flight records. Although the transfer function obtained is only valid for a specific set of flight conditions, the method is completely general and could for example be used to study the variation of the pilot parameters throughout the complete flight envelope.

It has been shown that the model not only provides a good qualitative explanation of the pilots control action but it can be used to accurately predict the frequency of pilot induced oscillations during elevation tracking.
CHAPTER 7

ROOT LOCUS OPTIMISATION OF HUMAN PILOT CONTROL SYSTEMS

7.1 Introduction

The ability to determine a transfer function for the human operator now enables root locus methods to be extended to the design of improved man-machine control systems. In particular it should now be possible to compensate manual aircraft control systems in order to prevent residual oscillations induced by the pilot.

This problem has recently become more acute in modern aircraft where the pilot is required to control flight vector rather than pitch attitude. For some aircraft the non-minimum phase response of the flight vector, coupled with the reaction time of the pilot, is sufficient to produce large induced oscillations which seriously affect the performance of the overall system.

In this chapter the author first develops the pole-zero pattern of the aircraft for this form of control and then shows how root locus techniques can be used to optimise the performance of the closed loop system.

7.2 Manual Control of Aircraft Flight Path

7.2.1 The pilot control loop

When the pilot flies according to pitch attitude he simply aligns distant objects seen through the windscreen with a fixed mark on the aircraft. The flight path of the aircraft will then be somewhere below the direction in which the aircraft is pointing and depends on the angle of incidence.
Changes in flight vector direction $\gamma$, angle of incidence $\alpha$, and pitch attitude $\theta$, are in fact related by the equation

$$\gamma = \theta - \alpha$$

Hence in order to control true flight vector, the pilot must not only be able to detect change in pitch attitude but must also be aware of the corresponding change in the angle of incidence.

This is achieved in practice by presenting the pilot with an aiming mark projected onto a head up display mounted in front of the windscreen. The aiming mark is focussed at infinity and is deflected with respect to the airframe datum (LFD) by an analogue signal proportional to the change in the angle of incidence. When this deflection is calibrated against the pitch attitude displacement of a distant object, the relative movement of the aiming mark with respect to target becomes $K(\theta - \alpha)$.

The error in the position of the aiming mark relative to target is then proportional to the change in flight vector direction required to make the aircraft fly directly towards the target. A schematic diagram of the system showing the relationship between the flight vector of the aircraft and the display seen by the pilot is given in Fig 7.1

7.2.2 Flight Vector response to movement of the elevator

The first step in understanding the flight vector elevation tracking problem is to obtain the aircraft transfer function $(\gamma/\eta)$. Since $\gamma$ is a function of the angle incidence $\alpha$, one must express the change in flight vector direction in terms of the state variables $\theta$, $w$, and $u$ normally used in the longitudinal equations of motion.
Fig. 7.1 Relationship between the aircraft flight vector and the visual display seen by the pilot.
Now the change in the angle of incidence $\alpha$ can be expressed in terms of $\dot{w}$ by the equation

$$\alpha = \frac{\dot{w}}{u} = w$$ \hspace{1cm} 7.2.2

where $\dot{w}$ is the incremental vertical velocity normalised with respect to the forward speed $u$.

Combining equations 7.2.2 and 7.2.1, the required transfer function can be written as

$$\frac{\gamma}{\eta} = \frac{\theta}{\eta} - \frac{\dot{w}}{\eta} = \frac{\theta(s) - \dot{W}(s)}{F(s)}$$ \hspace{1cm} 7.2.3

where $F(s)$ and $\theta(s)$ are the stick fixed and pitch attitude polynomials previously defined in Chapter 4. The remaining numerator polynomial $\dot{W}(s)$ was obtained by solving the longitudinal equations of motion for the ratio ($\ddot{w}/m$) assuming that $X(s)$ and $Z(s)$ were zero. Combining the numerator polynomial in a single expression $\Gamma(s)$, the flight vector transfer function becomes

$$\frac{\gamma}{\eta} = \frac{\Gamma(s)}{F(s)}$$ \hspace{1cm} 7.2.4

7.2.3 The Tracking Problem

To obtain some insight into the nature of the tracking problem it is instructive to examine the free aircraft response to movements of the elevator. The results of evaluating the flight vector, incidence, and pitch attitude transfer functions for a typical aircraft were as follows

$$\frac{\gamma}{\eta} = \frac{-0.278(s + 0.0202)(s + 16.07)(s - 13.6)}{F(s)}$$ \hspace{1cm} 7.2.5
Fig 7.2 Aircraft responses to a typical pilot movement of the elevator.
\[
\frac{\alpha}{\eta} = \frac{-0.278(s + 205)(s^2 + 0.019s + 0.000725)}{F(s)}
\]

\[
\frac{\theta}{\eta} = \frac{-56.278(s + 1.08)(s + 0.02)}{F(s)}
\]

where \(F(s) = (s^2 + 3.96s + 15.2)(s^2 + 0.2s + 0.005)\).

The aircraft responses to a typical pilot movement of the elevator are then as shown in Fig 7.2.

From these responses it can be seen that the flight vector initially moves in the wrong direction when the pilot moves the stick. In most aircraft the magnitude of this effect is very small although it can often last as in this case for about 0.35 seconds.

Flying according to true flight vector is, therefore, very different from fixed sight tracking in which the pilot simply points the aircraft in the required direction by changing its pitch attitude. Pilots trained on attitude control will therefore have a natural tendency to overcorrect for the error in \(\gamma\). This will then force the pilot to use a high loop gain which can result in the induced oscillation.

The pilots difficulties in controlling flight vector can also be appreciated from the root locus plot of the control loop shown in Fig 7.3.

Assuming that the control action of the pilot can be described by the transfer function

\[
P(s) = \frac{5K_p e^{-0.2s}(s + 0.5)}{(s + 10)(s + 15)}
\]
Fig 7.3 Human pilot control of aircraft flight path $\gamma$

Fig 7.4 Root locus of the human pilot flight vector control system shown in Fig 7.3.
then the root locus equation for the short term stability of the manual control loop becomes

$$\frac{+1}{s} = \frac{Ke^{-0.2s} (s + 0.5)(s + 16.07)(s - 13.61)}{s(s + 10)(s + 15)(s + 7)(s^2 + 3.96s + 15.2)}$$

Note that in this case we require the $0^\circ$ locus due to the non-minimum phase response of the aircraft flight path to movements of the elevator.

The root locus plot given in Fig 7.4 clearly shows the nature of the difficulties experienced by the pilot during elevation tracking. In this case the branch of the locus from the short period aircraft pole $P_1$, moves directly into the right hand half plane.

Tracking will, therefore, be subject to induced oscillations since any increase in loop gain caused by the pilot will result in a decrease in the damping of the closed loop system. The root locus not only provides a qualatative explanation of the problem but it also enables the designer to predict the frequency of the oscillation and the gain used by the pilot. For the particular aircraft considered here for instance, it can be seen that there will be a 3.4 rad/sec oscillation if the pilot uses a gain $K_p = 30$.

7.3 Methods of Improving Performance

While we are unable to remove the pure time delay of the pilot it is, however, possible to considerably improve the transient response of the system following the delay. As shown in Chapter 4 this can be achieved by reshaping the pure time delay root locus until the system has well damped closed loop poles.
In this case the main objective of reshaping the locus is to force the branches from the short term aircraft poles back into the left hand half plane. This can be achieved in many ways some of which will now be considered.

7.3.1 Modification to the Aircraft Dynamics

One very attractive solution to this problem would be to modify the flight vector response of the aircraft so that it responds more rapidly to movements of the elevator. This can be achieved by fitting special flaps on the wings to enable the pilot to directly control the lift.

This would have the effect of introducing a large phase advance into the control loop and would considerably improve the stability and performance of the closed loop system.

Unfortunately this modification is very expensive and also tends to degrade the flight performance of the aircraft in other manoeuvres. It does, however, become a practical solution provided direct lift control is included in the initial design.

7.3.2 Shaping the deflection of the aiming symbol in the head up display

Since we are unable to modify the true flight vector response of the aircraft, the only alternative is to modify the flight vector response seen by the pilot in the head up display. This can be achieved by shaping the signal used to deflect the aiming symbol which will then represent an apparent change in the angle of incidence $\alpha_A$. Hence the pilot will see an apparent change in the flight vector given by

$$\gamma_A = \theta - \alpha_A$$ 7.3.1
It should be noted that any form of signal shaping will, of course, introduce an error in the position of the aiming symbol. The form of shaping used must, therefore, be chosen to give only a small error which decays to zero as rapidly as possible. The object of shaping the response seen by the pilot is to give an initial movement of the aiming mark towards the target when the pilot moves the stick. This should restore his inherent anticipation and enable him to bring the aiming mark more smoothly into the desired flight path.

The simplest way to satisfy these restrictions and also introduce phase advance into the flight vector control loop is to lag the deflection of the aiming symbol to give an apparent change in the angle of incidence of

$$\alpha_A = \frac{\alpha \beta}{(s+\beta)} \quad 7.3.2$$

This form of compensation also has the advantage of reducing the noise in the loop and a block diagram of the proposed system is shown in Fig 7.5.

7.4 Human Pilot Control of Apparent Flight Vector

7.4.1 Apparent flight vector response of the aircraft to movements of the elevator

In order to use root locus methods for the optimisation of the proposed incidence lag compensated system it must first be reduced to an equivalent single loop feedback system as shown in Fig 7.6. The analysis of the equivalent system then requires the apparent flight vector response of the aircraft which can be obtained as follows.
Fig 7.5 Proposed Incidence Lag Compensation for Human Pilot Control of Aircraft Flight Path.

Fig 7.6 Single loop equivalent of the incidence lag compensated system.

Fig 7.7 Variation of the apparent aircraft zeros as a function of the compensation pole $\beta$. 
Combining equations 7.3.1 and 7.3.2 gives

\[
\frac{\gamma_A}{\eta} = \frac{\theta}{\eta} - \frac{\beta}{s+\beta} \frac{\alpha}{\eta}
\]

which can be written in terms of the authors aircraft polynomials \(F(s), \theta(s)\) and \(w(s)\) as follows

\[
\frac{\gamma_A}{\eta} = \frac{(s+\beta) \theta(s) - \beta w(s)}{(s+\beta) F(s)}
\]

Evaluating this expression for values of \(\beta\) in the range 1 to 20 it was found that the transfer function could be written in the form

\[
\frac{\gamma_A}{\eta} = \frac{-K (s+a)(s+b+jc)}{(s+\beta)(s^2 + 3.96s + 15.2)(s^2+0.02s+0.005)}
\]

where \(K = 54\), \(a = 0.02\), \(b = 0.05\), and \(c\) varies with \(\beta\) as shown in Fig 7.7.

Hence the short term response of the aircraft is therefore given by

\[
\frac{\gamma_A}{\eta} = \frac{-54(s+0.05+jc)}{s(s+\beta)(s^2+3.96s+15.2)}
\]

7.4.2 Root Locus Optimisation of the Control Loop

The root locus equation for the apparent flight vector control loop shown in Fig 7.6 is given by

\[
-1 = Ke^{-0.2s} \frac{(s + 0.5)(s + 0.05 \pm jc)}{s(s+10)(s + 15)(s + 7)(s + \beta)(s^2 + 3.96s + 15.2)}
\]

and when plotted for the case \(\beta = 10\), the locus is as shown in Fig 7.8.
Fig 7.8 Apparent Flight Vector System for $\beta = 10$.

Fig 7.9 Apparent Flight Vector System for $\beta = 5$. 
From this plot it can be seen that incidence lag compensation produces a dramatic improvement in the damping of the short period closed loop poles for normal pilot gains in the range (10 > Kp > 1). There is also a considerable improvement in the maximum gain that he can use before the control loop becomes unstable.

In order to test the effect of varying the time constant of the lag compensator, the locus was also plotted for the case when \( \beta = 5 \) as shown in Fig 7.9. This plot also shows a considerable improvement in the damping of the short period closed loop poles. Another interesting feature of this locus is the set-off direction of the branch from the aircraft pole P1. Comparison of this direction with that for the case when \( \beta = 10 \), suggests that there must be a value of \( \beta \) which produces the maximum penetration of the aircraft pole branch into the left hand half plane.

In order to obtain the optimum value of \( \beta \), a complete family of loci were then plotted as shown in Fig 7.10. From these loci the value of \( \beta \) for maximum improvement in the damping of the short period closed loop poles is seen to be 5.5, which corresponds to an incidence lag time constant of 180 milliseconds.

7.4.3 The effect on performance of using non-linear stick to elevator gearing

The root locus for the case \( \beta = 5 \) clearly shows that a further improvement in performance is possible if we can encourage the pilot to reduce the loop gain.

Typical flight records show that during landing the pilot tries to remove the final 1° of error at a maximum rate of 1° per second. From the flight vector response shown in Fig 7.2 this rate corresponds to a 0.25° elevator deflection which is equivalent
Fig 7.10 The effect on the dominant short period mode of varying the lag compensation in the range $\beta = 1$ to $\beta = 10$.

Fig 7.11 Proposed non-linear stick to elevator gearing to achieve a further improvement in closed loop performance.
a stick movement of 0.16". He therefore typically uses a gain of 0.16" of stick per degree visual error which corresponds to a value of Kp = 10.

Despite the undoubted skill of the pilot, these small stick movements indicated that he has reduced the loop gain as far as possible with the present system. Any further reduction must therefore be made by reducing the stick to elevator gearing.

The stick to elevator gearing is normally chosen so that the full stick movement (±4") produces a change in the elevator angle of ±6°. This assures that even during a landing approach the pilot is able to change the flight vector direction at a rate of 20°/sec in order to pull the aircraft out of a dive.

These conflicting requirements could, however, be readily overcome if the aircraft manufacturers would only adopt some form of non-linear gearing such as that shown in Fig 7.11. This would then give a considerable improvement in performance even without the incidence lag compensation.

7.5 Simulator verification of the root locus design

Unknown to the author, the Royal Aircraft Establishment at Farnborough had also been asked to try and improve the elevation tracking performance of the same aircraft under the same set of flight conditions. They constructed a model of the aircraft cockpit and simulated the aerodynamics using an analogue computer. The simulator was then flown by a test pilot who tried to align the aiming symbol in the head up display with a runway seen on a film projected in front of the windscreen.
As a result of these tests it was found that the most practical engineering solution to the tracking problem was to lag the deflection of the aiming mark seen in the head up display. It was also established that the pilot showed a strong preference for incidence lag time constants in the range 170 to 200 milliseconds and chose 185 milliseconds as the optimum value.

7.6 Summary

The author has shown that it is now possible to use root locus techniques for the design of improved man-machine control systems. Although the analysis has been restricted to a specific aircraft control loop, the generality of the method makes it suitable for the solution of many similar problems arising in the field of cybernetics.

The remarkable agreement between the simulator results and those predicted by the root locus analysis not only establish the validity of the design procedure but also provides additional confirmation of the accuracy of the human pilot transfer function.
CHAPTER 8

DISCUSSION AND CONCLUSIONS

The root locus method is a powerful design tool which has many advantages compared with the frequency response techniques of Nyquist and Bode. It applies directly to sampled data, pure time delay, and distributed parameter systems; and can also be used to design improved man-machine control systems.

The comparatively slow development of root locus techniques has been due to the time and effort required to obtain a sketch of the locus. Although rules exist for constructing the locus, it has been found that they fail to show how the many different forms depend on the positions of its poles and zeros. The analytical data required to obtain this information has now been produced for many simple systems which occur frequently in engineering design. Using these results the designer can now identify the general shape of the root locus directly from the relative positions of the open loop poles and zeros.

It has also been found that a suitable choice of normalising factors for simple pole-zero patterns can often lead to the display of all possible forms of their root loci on a single map. This new concept in root locus topology has been illustrated in Chapter 2 for the root locus of four pole systems.

Investigations into the analytical properties of simple loci have revealed many locus shapes which were previously unknown and these have been considered in relation to the corresponding new forms of closed loop response. Of particular interest were the locus shapes which enabled the designer to place the closed loop poles at the C of G of the open loop pole-zero pattern. These systems were found to have very
unusual forms of closed loop response which in some cases were superior to standard closed loop designs.

A detailed study of transitional root loci showed that the radical change in locus shape is not associated with a corresponding radical change in the nature of the closed loop response. Although there is a major redistribution of closed loop poles in this case, the shape transition gives rise to an equivalent closed loop system which has an almost identical time response.

Since many different pole-zero patterns can give rise to a very similar time response, it is unlikely that any general relationship will be found between locus shape and closed loop performance. A study of the locus shapes of systems up to 4th order did however show that (1) a Binomial response is possible only for loci which coincide with their asymptotes and (2) the standard closed loop responses of a system such as ITAE, Butterworth, etc. are each associated with the same general shape of its root locus.

It was also found that other relationships between locus shape and closed loop performance exist for some of the important low order systems which occur in engineering design. These results are summarised in global performance maps given in Chapter 2.

For systems above 4th order the analytical relationship between locus shape and pole-zero pattern are generally far too complicated to be of any practical use for sketching the shape of the root locus. In such cases it has been found that the locus is more easily obtained using some form of computer program or automatic root locus plotter.

A detailed study of the methods which were available for plotting root loci showed that they all had very serious limitations and could not
be used to extend the techniques of root locus design. In particular they were unable to give the correct locus shape in regions of high sensitivity such as break points, or to plot the branches of pure time loci which do not start on an open loop pole.

The author's new computer program has overcome these difficulties and can also be used to plot the loci of systems with pure time delay and/or distributed lag. The algorithm is very rapidly convergent and has been found to take an average of less than two computations per point to compute the locus to a phase accuracy better than 20 seconds of arc.

The need for such high accuracy arises frequently in root locus plotting. It occurs for example at break points and points on the locus near to multiple closed loop poles. An example of the error in locus shape which can arise due to phase error is well illustrated in the region of the triple closed loop pole of the phase angle loci shown in Fig 3.2.

The ability of the new algorithm to locate break points with very high accuracy now enables root locus methods to be used in reverse to factorise polynomials which have one or more repeated roots. A comparison of the new method with classical curve following techniques shows that its success is due to its simplicity and its ability to take advantage of experience gained at previous points on the locus.

It has been found that the algorithm applies equally well to many other contours apart from root loci. It has, for example been used to plot constant gain contours. These are a particularly useful addition to normal root locus plots since they enable the designer to obtain the position of the closed loop poles. The technique could be further extended to plot stream functions in hydrodynamics or equipotentials in electrostatics.
The significant advantages of the new algorithm have now made it practical to develop a low cost (£1500) Hybrid Computer capable of displaying root loci on a CRO. This computer is at present being built in the Department by Mr. L. S. A Mansi who holds joint patents on the machine with the author and Professor W. F. Lovering.

Extension of automatic plotting to pure time delay systems has now made it possible to predict their closed loop transient response directly from the root locus. This a considerable improvement on previous methods in which the effect of the delay had to be represented by pole-zero patterns obtained from truncated series approximations. The success of the root locus method in this case is due to the fact that it gives the true closed loop poles of the system and makes no attempt to predict the response prior to the delay.

It has been shown that pure time delay root loci can also be used in conjunction with the measured closed loop response of a system in order to identify its open loop dynamics. This useful extension to root locus has been used to identify human pilot transfer functions from aircraft flight records.

An investigation into the present methods available for improving the performance of pure time delay systems has shown that they do not remove the delay in the output, but simply try to eliminate its destabilising effect on the closed loop system. The main disadvantage of these methods is that they rely on exact pole-zero cancellation and can also greatly increase the steady state error arising from external disturbances applied to the system.

Since the effect of a pure time delay cannot be removed from the output, the best that can be achieved is to design the system to have a desired closed loop transient response without increasing the steady state error.
The ability to plot the $0^\circ$ pure time delay root locus now enables the designer to specify a closed loop pole-zero pattern which gives a desired response, and then obtain the corresponding open loop system directly from the root locus. This extension to the root locus method has been illustrated in Chapter 4 for a third order type 1 system which was required to have the fastest possible closed loop step response following the delay.

To verify the inverse root locus design procedure it became necessary to obtain accurate time domain responses of pure time delay systems. A study of the available simulation techniques showed that they could not be used for this purpose since they require the system to be expressed as a set of first order differential equations.

A new language has been developed which overcomes these difficulties and also applies to the simulation of sampled data systems. It has been extensively tested on large scale industrial problems and has been found to have many advantages over other high level languages at present being used for system simulation.

The speed and accuracy of the new automatic plotting procedure has now made it feasible to use root locus methods for the analysis of high order multiloop systems. An example of this improved capability is given in Chapter 5 in which a generalised pole-zero method is developed for the design of aircraft control systems. When used in conjunction with the author's computer program it has been found that suitable control laws can now be obtained for a typical aircraft in approximately 15 minutes. This compares very favourably with previous methods which employed manual re-shaping of the aircraft Nichols plot and took anything up to several man-months.
This extension to the root locus capability has been used successfully to design improved autopilots for several aircraft. Flight trials of these aircraft have shown that the design technique gives satisfactory control laws and the method has now been adopted by several autopilot manufacturers and government research establishments.

Combining this pole-zero method with the author's work on time delay systems it became possible to extend root locus techniques to obtain human pilot transfer functions from measured flight records. Various methods of representing the control action of the pilot were considered and it was found that the linear continuous model was the most suitable form for parameter identification. Although the pilot transfer function obtained will only be valid for a specified set of flight conditions, the method could be used to study the variation of pilot's parameters throughout the complete flight envelope. When applied to a particular aircraft it was found that the method not only gave a good qualitative explanation of the pilot's control action but it was also successful in predicting the frequency of the pilot-induced oscillation during elevation tracking.

Having established a reliable pilot transfer function it became possible to extend root locus methods to the design of an improved pilot-aircraft control loop. For the particular aircraft considered the root locus plots showed a clear optimum for the compensation function parameters required to improve the closed loop performance. The required compensation and its optimum parameters have been confirmed using normal simulation techniques at RAE, Farnborough.

The ability to obtain these results directly from root locus analysis should considerably reduce the cost of future development of manual aircraft control systems.
Although the analysis has been restricted to a specific aircraft control loop, the generality of the method will now enable root locus techniques to be used to improve the design of many similar man-machine control systems.

The future development of the root locus method lies in the further application of its computer aided design capability. It will soon become possible to display the loci on CRO terminals attached to the computer. Using this facility together with other stored programs such as the time domain simulation language it should be possible to fully exploit the extensions to the basic root locus methods given in this thesis. It is hoped that such facilities will further encourage designers to use root locus methods in control system design.
A1 Introduction

Many digital simulation languages have been developed recently most notable of which are C.S.M.P. (37), A.P.S.E. (38), and F.I.F.I. (39). Unfortunately these programs are very expensive and are not at present available in the college computer library. Most of these languages employ high order Runge-Kutta integration algorithms and are unsuitable for the simulation of systems which include pure time delay. They also have the serious disadvantage that the system to be simulated must first be expressed as a set of first order differential equations.

Since accurate time responses are required for many systems considered in this thesis, the author developed a new language in which each block of the system is simulated directly in a similar manner to the method used on an analogue computer.

Each block or transfer function is represented by an Algol Procedure and the blocks are interconnected by programming instructions. A simple form of Euler integration is used which enables the technique to be extended to pure time delay and sampled data systems.

The language has been extensively tested on industrial problems and has been used successfully to simulate large scale systems such as the autoland controls of the Comet.
Consider a first order differential equation of the form

\[ \dot{x} = f(t) \]

The solution, \( x(t) \) is the area under the \( \dot{x}(t) \) curve and can be expressed as a Taylor series in \( \Delta t \) as follows

\[ x(t + \Delta t) = x(t) + \dot{x} \Delta t + \frac{\ddot{x}}{2!} (\Delta t)^2 \quad \text{(A.2.1)} \]

where the derivatives \( \dot{x}, \ddot{x} \) etc. are evaluated at time \( t \).

Most of the digital simulation languages compute the change \( \Delta x \) in \( x \) using one of the Runge-Kutta formulae (40)(41). Perhaps the most widely used of these is the "one-third rule" which when \( \dot{x} \) is a function of \( t \) only reduces to Simpson's rule, i.e.

\[ x(t + \Delta t) = x(t) + \Delta x \quad \text{(A.2.2)} \]

where

\[ \Delta x = \frac{1}{6} [a_1 + 2a_2 + 2a_3 + a_4] \quad \text{(A.2.3)} \]

The coefficients \( a_i \) are given by

\[ a_1 = f(t) \Delta t \quad \text{(A.2.4)} \]
\[ a_2 = f(t + \frac{a_1}{2}) \Delta t \quad \text{(A.2.5)} \]
\[ a_3 = f(t + \frac{a_2}{2}) \Delta t \quad \text{(A.2.6)} \]
\[ a_4 = f(t + a_3) \Delta t \quad \text{(A.2.7)} \]

The main difficulty with this method of integration is that four computations must be made for each new value of the integral.
As a result a large amount of computer store is required even for simple low order problems. It was also found that Runge-Kutta techniques of integration are difficult to program in block form which is the most convenient method of simulating time delay and sampled data systems. 

If, however, the Taylor Series for \( x(t) \) is truncated after the first derivative, then

\[
x(t + \Delta t) = x(t) + \dot{x} \Delta t\]  \hspace{1cm} (A.2.8)

or written as an algorithm this becomes

\[
x(t) := x(t) - \dot{x} \Delta t \]  \hspace{1cm} (A.2.9)

This technique is known as Euler Integration. It assumes that the derivative \( \dot{x} \) remains constant over the interval \( \Delta t \) and that the area under the curve between \( t \) and \( t + \Delta t \) can be taken as the rectangle \( \dot{x} \Delta t \). The simplicity of the method offers many programming advantages when integration is to be implemented on a general purpose computer in a high level language. It was therefore adopted as the basic subroutine in the author's Algol procedures for direct transfer function simulation.

An initial test run is normally carried out using \( \Delta t \approx 10^{-2} \) of the shortest time constant in the system. The simulation is then repeated with smaller values of \( \Delta t \) until there is no significant change in the results.
It was found that linear systems can be conveniently simulated using only three basic transfer function blocks namely \( \frac{K}{s+a} \), \( \frac{K}{s^2+as+b} \), and \( K(s+b) \).

Consider for example the transfer function

\[
\frac{X}{f} = \frac{K}{s^2+as+b} 
\]

The corresponding differential equation is

\[
\frac{d^2x}{dt^2} + a \frac{dx}{dt} + bx = K f(t)
\]

subject to the initial conditions \( x = 0, \dot{x} = 0 \).

The first \( T_1 \) seconds of the unit step response can be completed using Euler integration as follows. Where \( X_1 = \dot{x} \), \( X_2 = x \), and \( DT \) is the incremental step length in the time variable \( T \).

```plaintext
READ:= A, B, K, T1, DTR;
F:= 1; X:= X1:= X2:= T:= DT:= 0;
L1: T:= T+DT; X2:= K*F - (B*X+A*X1);
    X1:= X1+X2*DT; X:= X+X1*DT; DT:= DTR;
    'IF' T < T1 'THEN' 'GOTO' L1;
```

The complete program can also be written as a self-contained block or Procedure as shown in Fig A1(b). The transfer function can then be regarded as a separate block in the program and the above algorithm can be replaced by the single statement

\[
X:= GRATE 2 \left( F, K, A, B, No \right);
\]
X2:= GRATE 1(X1, K, A, 1);

(b) Quadratic Lag

X2:= GRATE 2(X1, K, A, B, 1);

(c) Real Zero

X2:= DERIV(X1, A, 1);

Fig A1.1 Transfer Function Simulation Procedures
where $F$ is the input to the block, $\lambda$ in the output; and $N_0$ is the integer which labels the particular block being used. Similar procedures have been written for the transfer functions $K/(s+a)$ and $K(s+b)$. These are given in Fig A1 (a and c).

A4 Simulation of Pure Time Delay

The block structure of the language has also enabled a procedure to be developed to represent the effect of a pure time delay element with a transfer function $e^{-st}$.

The function $f(t)$ to be delayed is sampled every $T$ seconds and read into an array of $N$ elements as shown in Fig A1.2. Each element of the array is initially set to zero and the output is read from the array one element in advance of the input. The output will therefore be zero until time $NT$ when it again returns to the first element in the array. At this instant the output is $f(0)$, which is the required value of the delayed function $f(t-NT)$. As the computation proceeds, the input overwrites the values of $f(t)$ which have been read by the output and the whole process is repeated until the end of the simulation.

In practice it has been found that the sampling period $T$ should be chosen to be less than $10^{-3}$ of the shortest time constant in the delayed waveform. This means that for a delay of only one time constant the array $1:N$ must therefore contain at least 1000 elements. Although digital simulation of delay can often require large amounts of storage, it does however offer many advantages compared with the corresponding methods normally used on an analogue computer.
X2 := \text{DELAY (X1)};

\text{ALGOL PROCEDURE} \text{DELAY}

\text{'REAL'} \text{'PROCEDURE'} \text{DELAY(YDI)}; \text{'REAL'} \text{YDI};
\text{'BEGIN'} \text{DVL}[I] := \text{YDI}; \text{'IF'} \ I < \text{T1/DTR} \text{'THEN'} \text{YDO}[I] := \text{DVL}[I+1];
\text{'ELSE'} \text{'BEGIN'} \text{YDO}[I] := \text{DVL}[0]; \ I := \text{I} + \text{1}; \text{'END'};
\text{DELAY} := \text{YDO}[I] \text{'END'};

\text{PRINCIPLE OF OPERATION OF THE DELAY PROCEDURE}

\text{ARRAY [1:N]}

\text{Input} f(t)

\text{Output} f(t-NT)

\text{Fig AI.2 Extension of Transfer Function Simulation to Include the Effect of a Pure Time Delay}
Simulation of a complete system is achieved by interconnecting the appropriate transfer functions by means of simple programming instructions. An example of the simulation of a typical system is given in Fig A.1.3.

After each increase DT in T, the computation proceeds sequentially through the blocks so that the output of each block is updated once per cycle. In this way the total delay introduced into the loop is only one DT which occurs at the input comparator.

In the case of multiloop systems each loop will have a delay of DT and should be closed before computing the next section of the main loop.

The output of the system is sampled every 50 DT and the values of the response at these instants are fed to a line printer. Automatic scaling procedures have also been developed which enable a graph of the response to be obtained either on the line printer or on the X-Y plotter.

Parameter change facilities are also available so that families of responses can be obtained as illustrated in Fig A.1.3. This feature of the language has also enabled it to be used in conjunction with standard hill climbing programs in order to obtain optimum system performance. It should, however, be noted that digital hill climbing techniques are very inefficient and work of this type should where possible be carried out on a Hybrid Computer.
COMPUTER PROGRAM TO OBTAIN THE STEP RESPONSE OF THE SYSTEM FOR VALUES OF
K = 10, 20, 30, 40, and K = 50

L2: K:= K+DK
L1: T:= T+DT; F:= 1;
Y1:= F-Z1; Y2:= GRATE 1 (Y1, K, 9, 1);
Y3:= DERIV (Y2, 1, 1); Z1:= GRATE 2 (Y3, 1, 0, 0, 1);
'IF' T < TF 'THEN' 'GOTO' L1;
'IF' K < KM 'THEN' 'GOTO' L2;

INPUT DATA
K  DK  KM  DT  TF
10  10  50  0.001  5

STEP RESPONSES OF THE SYSTEM DRAWN BY THE X-Y
PLOTTER OF THE ICL 1905 COMPUTER

Fig A1.3 Digital Simulation of a Closed Loop System
The effect of Integration Step Length on the Speed and Accuracy of the Simulation

In the case of Euler Integration the accuracy of the process is very dependent on the step size $\Delta t$ and the nature of the waveform to be integrated. The effect of these errors on the transfer function simulation procedures was examined in detail for the case of a second order system in which $K = b = 25$, and $a = 3$.

The step response of this system was obtained using the GRATE 2 procedure and the results were compared with those computed directly from the analytical solution of the differential equation.

The effects of varying $\Delta t$ on the worst case error and the total solution time are shown in Fig A.1.4. From these results it can be seen that an accuracy greater than 0.2% can be achieved with a step length equal to 0.1% of the period of the oscillation. Extensive tests on many more complex systems have shown that an accuracy better than 0.5% is usually obtained provided $\Delta t$ is less than 0.1% of the shortest time constant in the response.

For these accuracies, the author's simulation language is much faster than those which employ Runge-Kutta forms of integration. When applied to the simulation of the Comet autoland for instance, it reduced the total computing time to less than three minutes compared with over 15 minutes required by C.S.M.P.
Fig A1.4 Results of Using Grater Procedure to Simulate the Step Response of the System \[
\frac{25}{(s^2 + 3s + 25)}
\]
Further Extensions and Facilities of the Simulation Language

The use of Euler Integration in the Transfer Function procedures enables problems to be simulated for negative values of $\Delta t$. This means that problems can be run backwards to determine the initial conditions of a system which give rise to specified final values of the response.

Procedures have also been written to represent the behaviour of simple non-linearities and these have been used to investigate the stability and performance of non-linear systems.

Samplers and zero-order holds are also readily programmed. These have been used to obtain the response of high order sampled data systems and considerably reduce the amount of algebra required to evaluate the responses using conventional $Z$ Transform Analysis.

The ability to include Boolean statements and standard Hill climbing routines has also enabled Hybrid computer techniques to be used for optimisation of system response.
REFERENCES

(1) Truxal J.G. "Control System Synthesis"


(3) Teodorchik K.F. and Bendrikov G.A.
    "The methods for plotting root paths of linear systems by qualitative determination of path type"

(4) Power H.M. "Approximation Theorems for establishing root locus topology"

(5) Williamson S.E.
    "Root Loci of Type II and III Systems"
    Control, July 1968, Vol 12, No 121, pp 645-647.

(6) Williamson S.E.
    "Root Loci of Type I Systems"
    Control, June 1968, Vol 12, No. 120.

(7) Williamson S.E.
    "Design Data to Assist in the plotting of Root Loci of Type 0 Systems"
    Control, May 1968, Vol 12, No 119

(8) Meadows N.G. "Generalised Root Loci of Third Order Systems"
    Control, April 1965, Vol 9, p 198.

(9) Williamson S.E. and Lovering W.F.
    "The Root Loci of Four Pole Systems"

(10) Hanock Ur "Root Locus Properties and Sensitivity Relations in Control Systems"
    Trans IRE, 1960.
(11) Graham D. and Lathrop R.C.
"The Synthesis of Optimum Transient Response;
Criteria and Standard Forms",
Trans AIEE, November 1953.

(12) Konwerski T. "Synthesis of Linear Systems for Prescribed
Time Domain Performance using Frequency Response Data"

(13) Levine L. "Analogue set-up solves polynomials, plots root
loci automatically"

(14) D'Azio J. and Hoppis C.
"Root Locus plotted using an analogue computer"

(15) Doda J.C. "Digital Computer Makes Root Locus Easy"
Control Engineering, May 1958.

(16) Cook G.E. and Cook W.E.
"Three ways to do root locus on computers"

(17) Aird R.J. and Moseley A.D.
"On line computer analysis for control system design"
International Conference on Computer aided design,
April 1969, University of Southampton.

(18) EASIAC Computer Morgan M.L. 1962
Electro-Scientific Industries Inc., USA.

(19) Williamson S.E.
"Accurate Root Locus Plotting including the
effects of pure time delay"

(20) Blackman "Pole-zero Approach to System Design"
Control Monograph 1963.
(21) Chu Y. "Synthesis of Control Systems by Phase Angle Loci"
Trans AIEE, Part (2) Appl Ind 71, 1952.

(22) Mansi, Williamson and Lovering
"Apparatus and method for root locus computation"

(23) Lee Y.W. "Statistical Theory of Communications"

(24) Levy E.C. "Complex Curve Fitting"

(25) Professor O. J. Smith
"A controller to overcome dead time"
ISA Journal Vol 6, No 2, 1959.

(26) Mason S.J. "Properties of Signal Flow Graphs"

(27) Duncan W.J. "Control and Stability of Aircraft"
Cambridge University Press 1959.

(28) Fry D.E. "The use of Cross-Correlation techniques for
the identification of the Hunter Mk12 Dynamic Response"

(29) Williamson S.E.
"Autopilot Control Systems for the Harrier"

(30) McRuer D.T. and Krendel E.S.
"The human operator as a servo system element"

(31) Mitchell M.B."System Analysis - The Human Element"
Electro Technology, April 1966.
(32) Shinners S.M. "Man Machine Control Systems"
Electro Technology, April 1967.

(33) Professor J. H. Westcott
"The Simulation of human operator tracking
using an intermittent model"

(34) Lange G.W. "Representation of the Human Operator as a
Sampled data system"

(35) Bekey G.A. "Sampled data model of the human operator in a
control system"

(36) Williamson S.E.
"Computer program to obtain the time response
of sampled data systems"

(37) System 360 "Continuous System Modelling Program"

(38) White G.T. "Simulation Languages for Process Control Problems"

(39) Sumner H.M. "F.I.F.I."

(40) McCracken D. and Dorn W.S.
"Numerical Methods"

(41) Kunz K.S. "Numerical Analysis"