AIRCRAFT SIMULATION AND ROBUST FLIGHT CONTROL SYSTEM DESIGN

Paul Patterson Aslin

Submitted in accordance with the requirements
of the award of D. Phil degree

UNIVERSITY OF YORK
DEPARTMENT OF ELECTRONICS

JUNE 1985

CONTENTS

Acknowledgements

I am indebted to the following for the help and guidance provided by them during the course of this project :

Marconi Avionics, Flight Automation Laboratory, Chatham. Marconi provided the necessary industrial funding for a CASE studentship and the data for the Machan remotely piloted vehicle. The current work was undertaken under an MoD funded project. Particular thanks are due to Mr. J. Aplin and Mr. T. Hamill who jointly supervised the progress of the project and Mr. D. Price who provided the initial contact. Thanks are also due to the many staff at Marconi who participated in meetings both at York and at Chatham.

SERC for providing a 2 year CASE studentship for this project.

RAe, Farnborough and Bedford who's involvement in an allied research area was initiated by interest in this project. In particular thanks are due to Mr. G. Butler, who provided some useful thoughts on the current work in addition to setting up an MoD project grant for work into analytical redundancy, and to Mr. S Winter for helpful suggestions in the latter stages of the work.

Dr. A. S. I. Zinober and Dr. C. M. Dorling, University of Sheffield, Dept. of Applied Mathematics, for their assistance in producing the results of Chapter 7 and for the many useful comments on the design of VSS control systems.

The staff and technicians of the Department of Electronics, York University who provided the facilities and equipment for the project. In particular thanks are due to Dr. R. J. Patton who supervised and in no small way contributed to the work embodied in this thesis and to Mr. S. W. Willcox for his help in producing some useful results during the later stages of the project.

Declaration

The following material, previously used by the author, is incorporated into this thesis :

P. P. Aslin and R. J. Patton ; "The Development of a Full Force Digital Simulation of a Remotely Piloted Vehicle for Application to Flight Control System Design and Assessment", Research Report No. YEE.1, University of York, Department of Electronics, 1983.

P. P. Aslin, R. J. Patton and C. M. Dorling ; "The Design of a Sliding Mode Controller for Dutch Roll Damping in a Non-linear Aircraft System", IEE Control 85 Conference, University of Cambridge, July 1985.

P. P. Aslin, J. V. Hatfield, A. J. A. Oxtoby and R. J. Patton ; "Real-time Simulation of a Non-linear RPV Incorporating Head-up Type Colour Graphics", 5th. Bristol International Conference, Remotely Piloted Vehicles, Bristol, September 1985.

ABSTRACT

The work covered in this thesis is based upon the design and development of flight control laws for a particular aircraft application. The aircraft chosen is a remotely piloted vehicle, the Machan, used as a development vehicle by Marconi Avionics, Chatham. The initial stages of the study are directed towards producing a fully non-linear simulation of this aircraft on a dedicated microcomputer. The aerodynamic data for this aircraft is known accurately from wind tunnel testing on the full sized vehicle. This development work led to a non-linear model and a close-to real time simulation of the aircraft with a realistic display system.

The latter parts of the work concentrate on the design of flight control systems for the Machan and the robustness aspects of a variety of closed-loop control strategies. Conventional 'classical' s-domain analysis techniques are first investigated and the likely performance of such schemes assessed. In the later stages 'modern' control philosophies are examined using state variable feedback techniques. The asymptotic properties of optimal control designs are exploited to provide an accurate method of specifying system performance via an asymptotic solution of the LQP problem and the control designs are shown to display desirable robustness properties. More recent work into variable structure systems and the sliding mode are also shown to be applicable to the design of flight control systems and have some measure of robustness.

Throughout these studies the robustness properties of the control are demonstrated by applying the control law to the fully non-linear Machan simulation.

INTRODUCTION

Over recent years it has become increasingly apparent that the design of aircraft flight control systems using classical techniques may not be entirely appropriate for modern combat aircraft. It is felt that the use of the vast body of modern control theory, which has evolved over the past decade, may allow aircraft performance to be bettered when compared with current classical control designs. This thesis covers work carried out at York in order to establish an environment in which modern control designs may be investigated and their applicability to the aircraft problem assessed. The work undertaken at York complements an already existing project at Marconi Avionics, Rochester, under which modelling and control system design for a small remotely piloted vehicle (R.P.V), the Machan (7), is being investigated. The modelling work at Marconi was preceeded by wind tunnel testing of the R.P.V. at Cranfield College of Aeronautics (8).

It was initially considered important to have available some form of aircraft simulation, preferably running in real time. The simulation facility would allow the investigators to gain an insight into the 'feel' of the problem and also allow for expansion into 'on-line' control law implementation. The aircraft environment involves a high degree of interaction between the operator, pilot, and the system hence it was considered appropriate to attempt, as far as possible, to emulate this interaction on the simulation.

In Chapter 1 of this thesis we consider the physics of the aircraft with a view to developing a realistic non-linear model for the Machan r.p.v. The aerodynamic parameters for this model are taken directly from the previously mentioned wind tunnel tests on the Machan and an accurate model of this vehicle can be proposed. Chapter 3 presents a detailed discussion of the implementation of the simulation in a laboratory environment. The principal aim of this chapter is to detail the trade off's required in the choice of a suitable simulation medium. Having made this

choice the actual task of implementing the simulation is also examined in detail. The ultimate aim was to provide a real time simulation with high quality 'head-up' type display and inputs via a joystick type of control. How closely this was achieved is covered in this chapter. In Chapter 4 some results from the open-loop simulation are provided which indicate how pilot workload can be assessed in addition to indicating how well the model matches the actual aircraft.

In the later stages of the work the development of control strategies for the Machan have been investigated in some detail. A linearised model of the Machan is derived by intuitive arguments in Chapter 2. This model has been used throughout the control design in order to facilitate the development of linear classical and state variable feedback schemes. Classical s-domain and frequency response techniques are reviewed in Chapter 5 to provide an insight into the control problem and how such simple controllers can be developed for the aircraft. Chapters 6 and 7 then examine the design of both optimal control and variable structure system schemes. These techniques rely upon state variable feedback and are examined with a view to assessing the ease of implementation and their likely performance. These schemes are employed with both linear and non-linear models and the differences in achievable performance are quantified.

The final chapter covers the robustness and sensitivity of the linear and non-linear state feedback controllers. In this context an examination is made of how closely the schemes maintain a given closed-loop system property as the system parameters vary. It is vital in the aircraft problem that the control is robust and hence these aspects of the state feedback schemes are of considerable importance.

CHAPTER 1

Vehicle Equations

In this chapter it is intended to develop the equations governing the aerodynamic behaviour of an aircraft at subsonic speeds. It will be appreciated that the aerodynamics of an airframe are complex and inherently non-linear. For the purposes of simulation studies it was found necessary to simplify some of these dynamics but retain sufficient of the non-linear properties to provide a realistic simulation. The simplifications are indicated in the text and some brief justifications are given. The interested reader is referred to the many texts available for a more detailed analysis (1,2,3,4,9,10).

1.1 Basic Concepts

An aircraft in flight may be considered to be a rigid body immersed in a fluid medium (air). The forces acting on the airframe are due to the motion of the aircraft through the air and due to the inherent properties of the airframe i.e. mass, etc. The major force required to maintain steady flight is the lift force and this is generated by the airflow over the wings, these being of an aerofoil section. The lift force must counteract the weight of the aircraft, any excess of lift acting so as to increase the vertical speed of the aircraft. In order to provide lift, an airflow must exist over the wings. This is generated by reason of the aircraft's forward speed provided by a thrust force from one or more engines of the jet or propeller type. The thrust force must balance, or be in excess of, the induced drag force on the airframe due to its forward motion. In simple terms then, when an aircraft is flying straight and level, these forces must be in equilibrium as shown in Fig. 1.1. This simple view ignores the forces acting in the planes normal and perpendicular to the so called longitudinal plane shown in Fig. 1.1 which may be generated

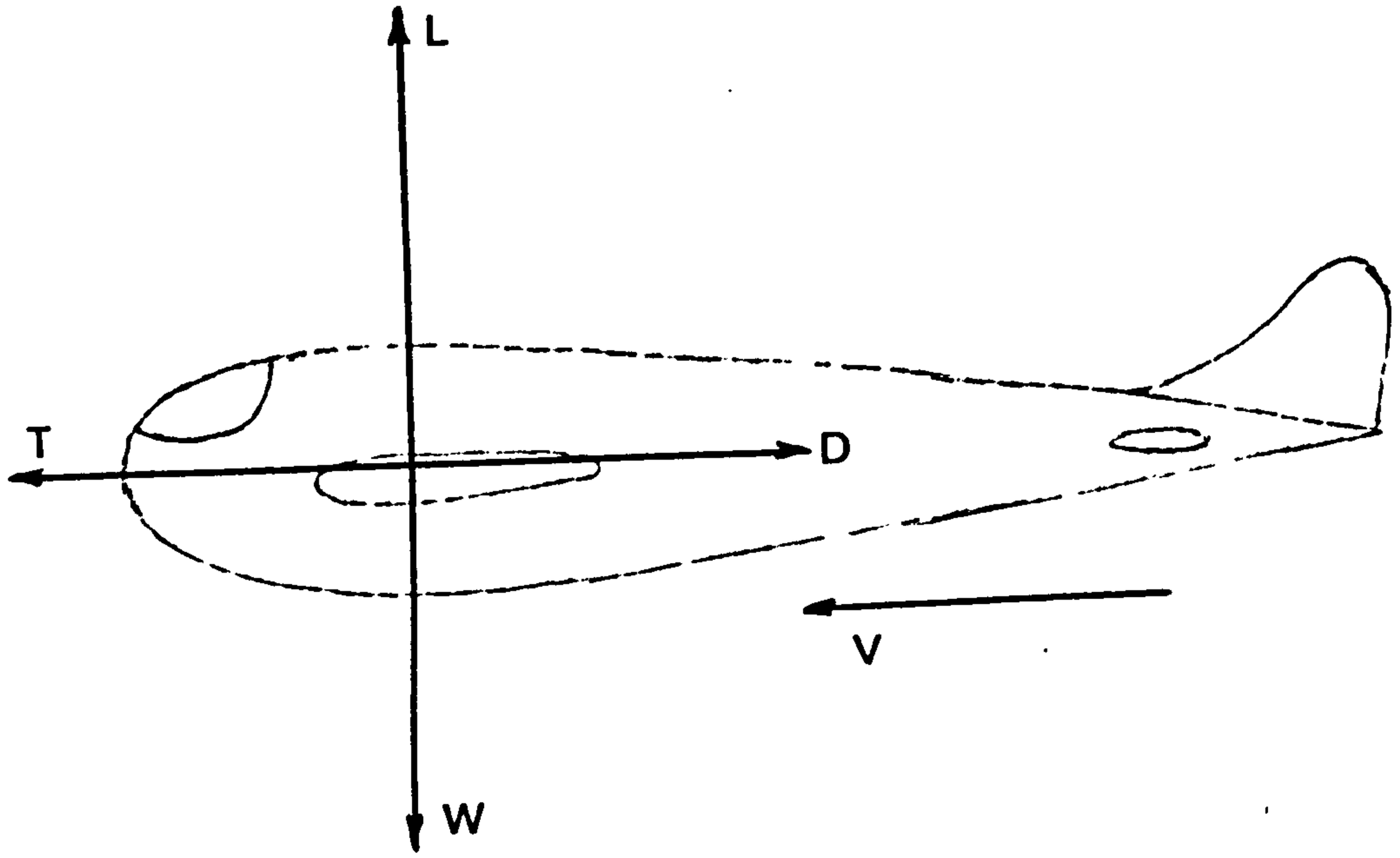


Fig. 1.1 Aircraft Forces in Steady Flight

by deflections in the aircraft's control surfaces or changes in wind speed.

In practice an aircraft contains a three axis system and forces and moments will, in general, act in all three axes, a six degree of freedom system. The two degree of freedom steady state model shown in Fig. 1.1 does, however, allow certain important aircraft performance figures to be derived.

For a realistic simulation a full force six degree of freedom model must be derived which acomodates most of the non-linearities of the basic aerodynamics.

1.2 Axes Systems

To derive a more complete picture of the aircraft in flight it is initially important to define a set of axes which will act as a reference frame around which equations of motion may be developed. Remembering that the aircraft is a free body in space its position may be defined with respect to a set of earth or gravity fixed axes which remain fixed with respect to the earth. This axis system is inconvenient for analysis, a better choice being a set of axes which remain fixed relative to the airframe and form the principal axes of inertia of the aircraft. This axis set, often called the body fixed axes, is shown diagrammatically in Fig. 1.2 and clearly remains fixed relative to the geometrical distribution of the airframe. We may now define forces and moments to act about this frame of reference. To translate between body fixed and, say, earth axes a transformation matrix may be used (see later).

It is conventional to define the nomenclature associated with the body fixed axis system with some rigour according to an agreed standard. Table 1.1 summarises this standard and we introduce some definitions below with reference to Fig 1.2.

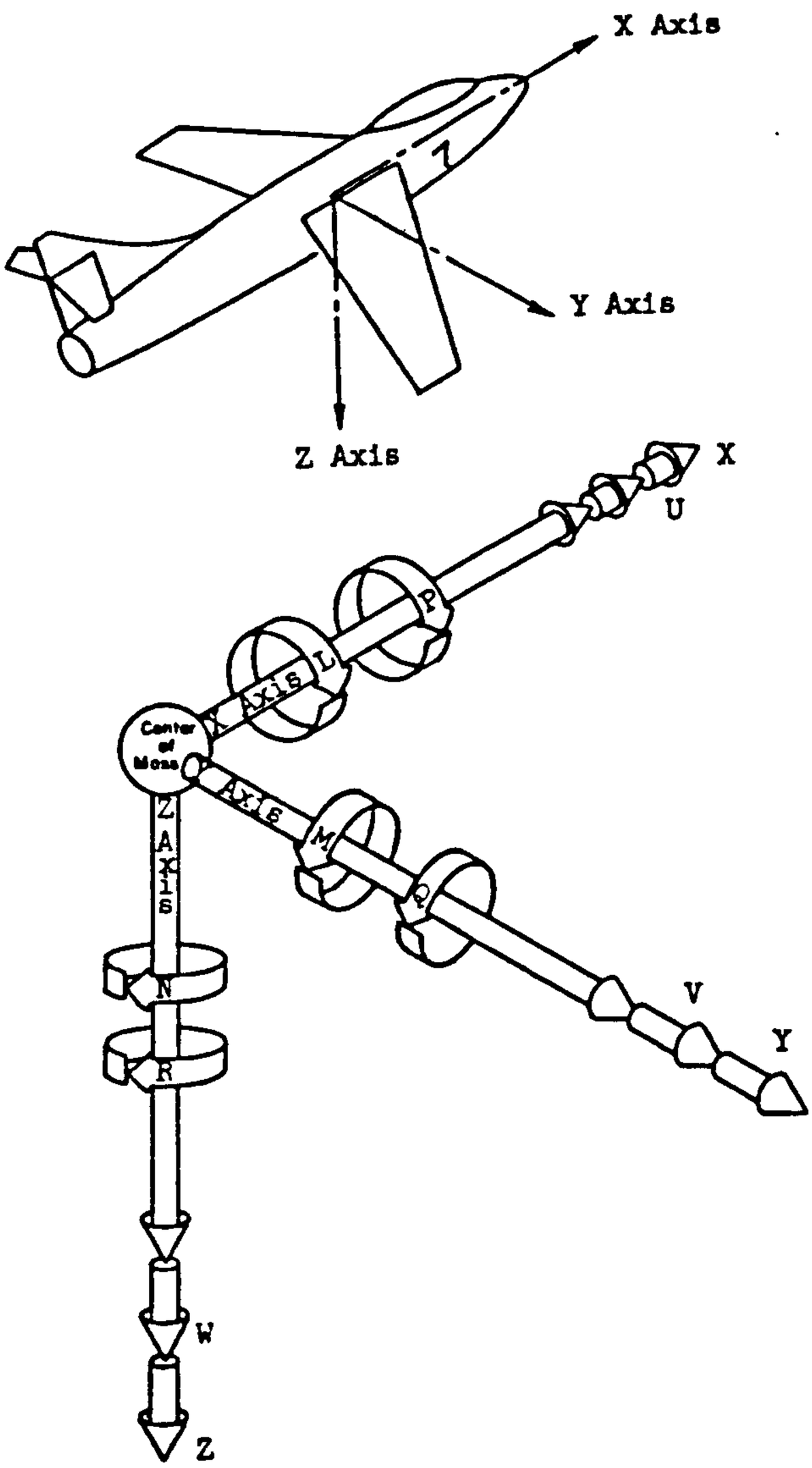


Fig. 1.2 Aircraft Axes System

Table 1.1

	Velocities	Applied forces and moments	Distances
Forward	U	X	x
Side	V	Y	y
Vertical	W	Z	z
Roll	P	L	
Pitch	Q	M	
Yaw	R	N	

a) OX is termed the longitudinal axis. Linear displacements from the steady state about this axis are defined as x metres for an aircraft having a steady state velocity along OX of U ms^{-1} and incremental velocity changes of u ms^{-1}. Force components along OX are taken as forward positive and termed X N. The roll component of aircraft motion is about OX and has velocity P rad s^{-1}, the resulting displacement being Φ rad under the action of a rolling moment L Nm taken positive in the clockwise sense looking along OX from O.

b) OY is the transverse or lateral axis. A steady state sideslip velocity, V ms^{-1}, produces a sideslip displacement y metres under the action of a sideslip force Y N. Pitching moments are generated about OY having angular velocities Q rad s^{-1}, angular displacements of Θ rad and pitching moment M Nm. The positive sense is defined as nose up or clockwise rotation about OY looking along OY from O.

c) OZ is the normal axis. An incremental change in downwards velocity of w ms^{-1} gives rise to a downward displacement of z metres with a downward force component Z N, downwards being positive. Yawing moments take place around OZ producing yawing velocities of R rad s^{-1}, an angular displacement of Ψ rad and a yawing moment of N Nm. All taken positive in the clockwise sense looking down.

Angles, moments and angular velocities are taken to be positive in the cyclyic sense and accelerations of the moving axes relative to the fixed earth axes are denoted by the dot notation.

Gravity fixed axes are a convenient reference set and are considered to be centred on the vehicle's centre of gravity. The OZ_o axis acts through the c. of g. towards the centre of the earth. The OX_o and OY_o axes lie in a plane tangiential to the earth's surface with OX_o oriented eastwards (or sometimes northwards) and OY_o oriented

southwards (or sometimes eastwards). The velocities along OX_o, OY_o and OZ_o are defined as U_o, V_o and W_o ms^{-1} respectively, note that $W_o = -dh/dt$ where h is the aircraft's height in metres, downward velocity being positive. The body-fixed axis set may be related to the gravity fixed axis set using Eulers equations. The rotational relationships between OX, OY, OZ and OX_o, OY_o, OZ_o are best understood by considering an intermediate axis set, OX_i, OY_i, OZ_i, initially coincident with OX_o, OY_o, OZ_o. The orientation of OX, OY, OZ with respect to OX_o, OY_o, OZ_o can then be built up as follows.

i) rotate OX_i, OY_i, OZ_i about the OZ_i axis by an angle Ψ(the azimuth or yaw angle).

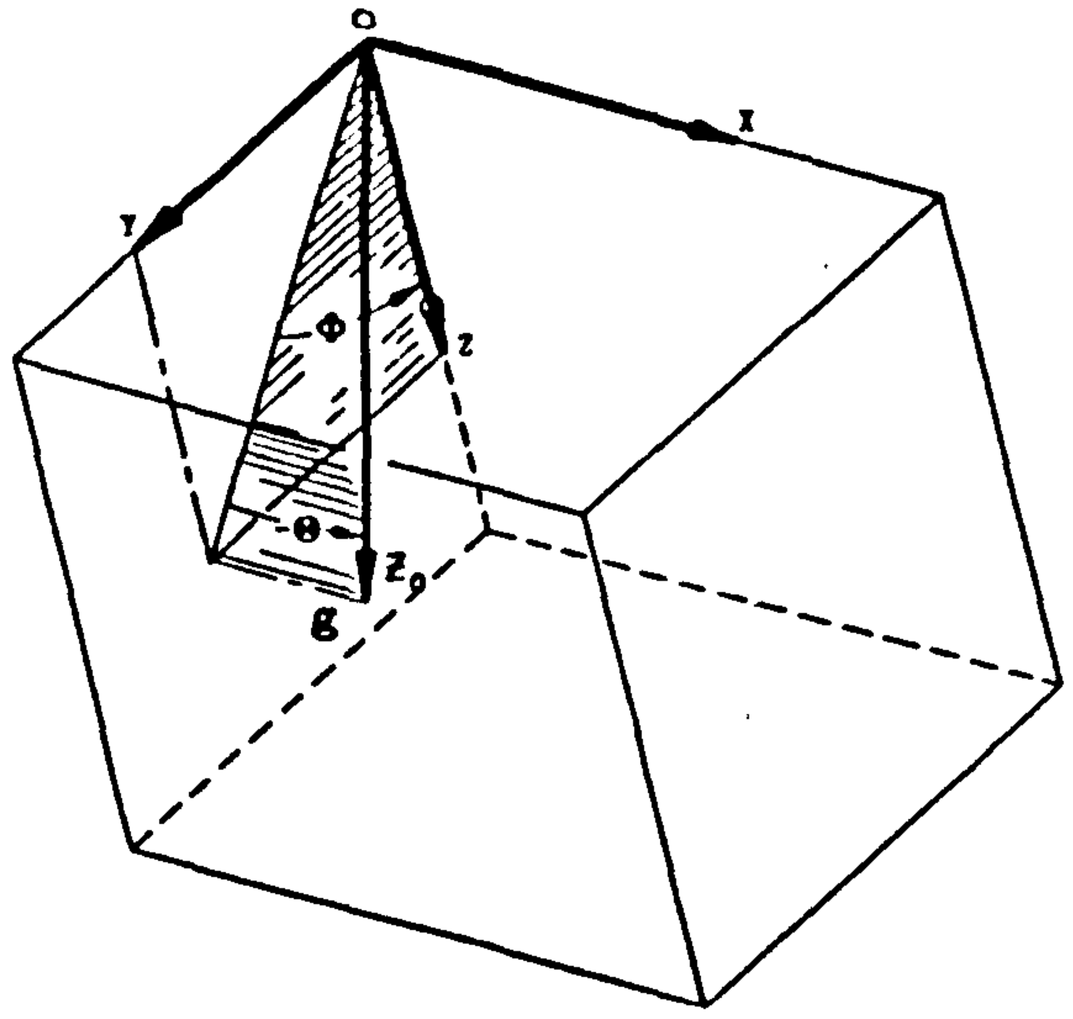
ii) rotate OX_i, OY_i, OZ_i about OY_i by an angle Θ (the pitch angle).

iii) rotate OX_i, OY_i, OZ_i about OX_i by an angle Φ (the roll angle).

OX_i, OY_i, OZ_i is now coincident with OX, OY, OZ as shown in Fig.1.3 with Ψ , Θ and Φ , the Euler angles defining OX, OY, OZ's attitude with respect to OX_o, OY_o, OZ_o.

By considering each of the rotations separately, as above, we may derive a transformation matrix, T, as a 3x3 orthogonal direction cosine array viz.

$$T = \begin{bmatrix} \cos\Psi\cos\Theta & \sin\Psi\cos\Theta & -\sin\Theta \\ \cos\Psi\sin\Theta\sin\Phi - \sin\Psi\cos\Phi & \sin\Psi\sin\Theta\sin\Phi + \cos\Psi\cos\Phi & \cos\Theta\sin\Phi \\ \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi & \sin\Psi\sin\Theta\cos\Phi - \cos\Psi\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix}$$

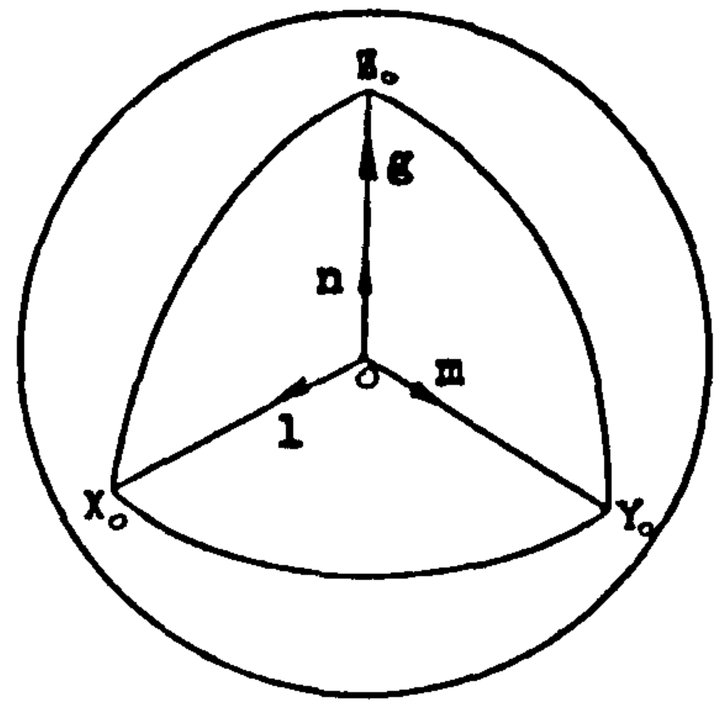


PHYSICAL SITUATION

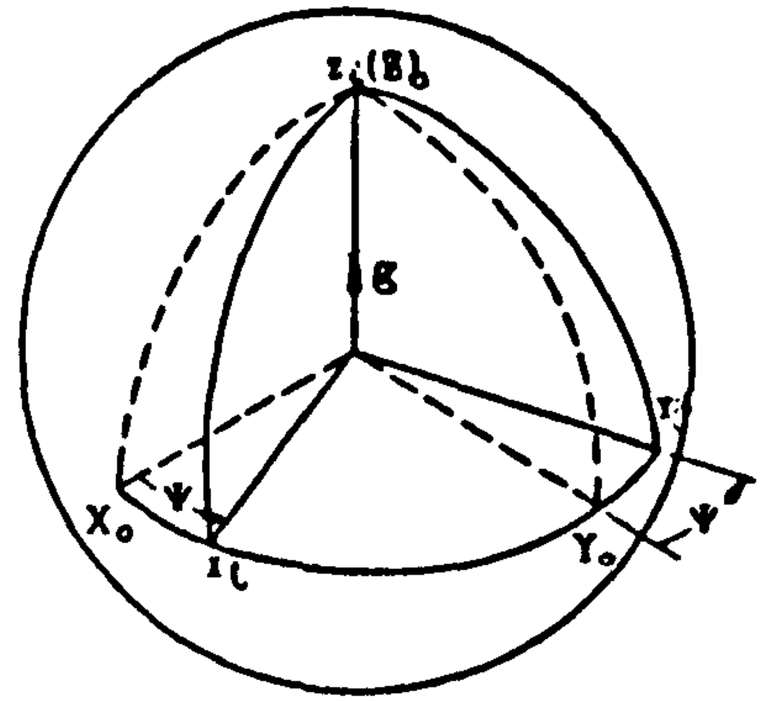
ORIENTATION OF GRAVITY VECTOR WITH XYZ BODY-FIXED AXIS SYSTEM

Θ - ANGLE BETWEEN g AND YZ PLANE; PITCH, POSITIVE NOSE UP

Φ - ANGLE BETWEEN Z AXIS AND PROJECTION OF g ON YZ PLANE; ROLL, POSITIVE RIGHT WING DOWN

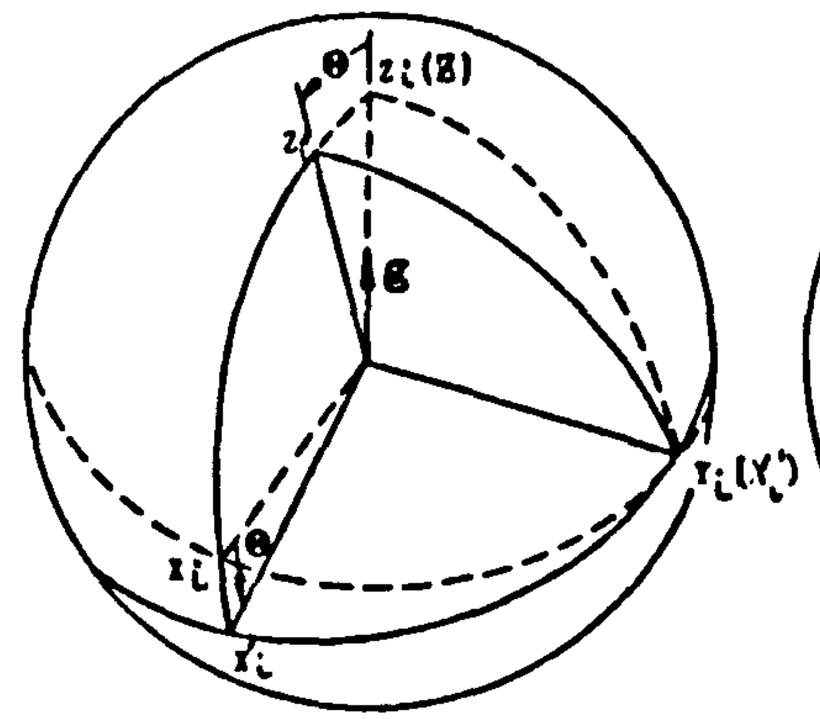


(1) VEHICLE-CENTERED GRAVITY-DIRECTED SYSTEM



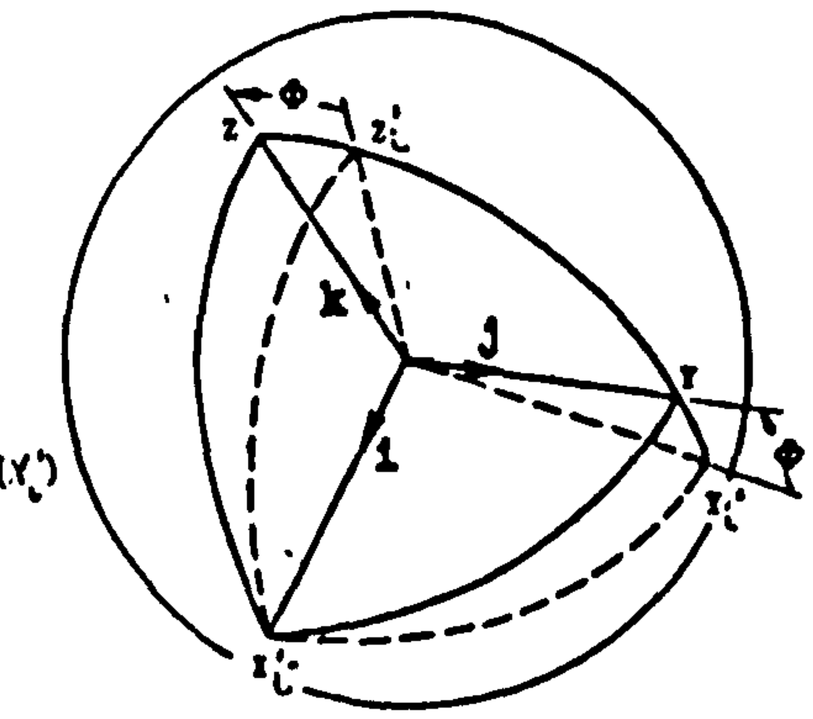
(2) HORIZONTAL FLIGHT REFERENCE SYSTEM

Derived from $X_oY_oZ_o$ by rotation about Z_o through yaw angle Ψ



(3) ROLL-RESOLVED FLIGHT REFERENCE SYSTEM

Derived from $X_i'Y_iZ_i$ by rotation about Y_i through pitch angle Θ



(4) VEHICLE BODY AXIS REFERENCE SYSTEM

Derived from $X_i'Y_i'Z_i'$ by rotation about X_i' through roll angle Φ

UNIT SPHERE DEVELOPMENT

Fig. 1.3 Axis Orientation Development

It can be shown (4) that the velocity components in the reference and inertial axes are related by

$$\begin{bmatrix} U_o \\ V_o \\ W_o \end{bmatrix} = T \begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

or

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = T^{-1} \begin{bmatrix} U_o \\ V_o \\ W_o \end{bmatrix} = T^{T} \begin{bmatrix} U_o \\ V_o \\ W_o \end{bmatrix} \qquad -1.1$$

Additionally, the pitch, roll and yaw velocities, Q,P,R, about OY,OX,OZ may be expressed in terms of the pitch, roll and yaw rates, $\dot{\Theta}$, $\dot{\Phi}$, $\dot{\Psi}$, in the gravity fixed axes since (4)

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \sin\Phi\cos\Theta \\ 0 & -\sin\Phi & \cos\Phi\cos\Theta \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \qquad -1.2$$

the inverse of equation 1.2 being

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \qquad -1.2\ a)$$

Equation 1.2 a) breaks down for $\Theta = \pm 90^{o}$ and this may be considered a disadvantage in simulation studies but this may be tolerated providing manoeuvres requiring only relatively small pitch angles are to be modelled. The 90° pitch condition could, of course, be easily trapped in any simulation if it were required to model manoeuvres leading to the establishment of larger pitch angles.

The velocity components in the body fixed axes U,V and W may be resolved into a single total velocity vector V_T whose direction is defined by the angles of sideslip, β, and incidence, α, as shown in Fig. 1.4.

It follows that

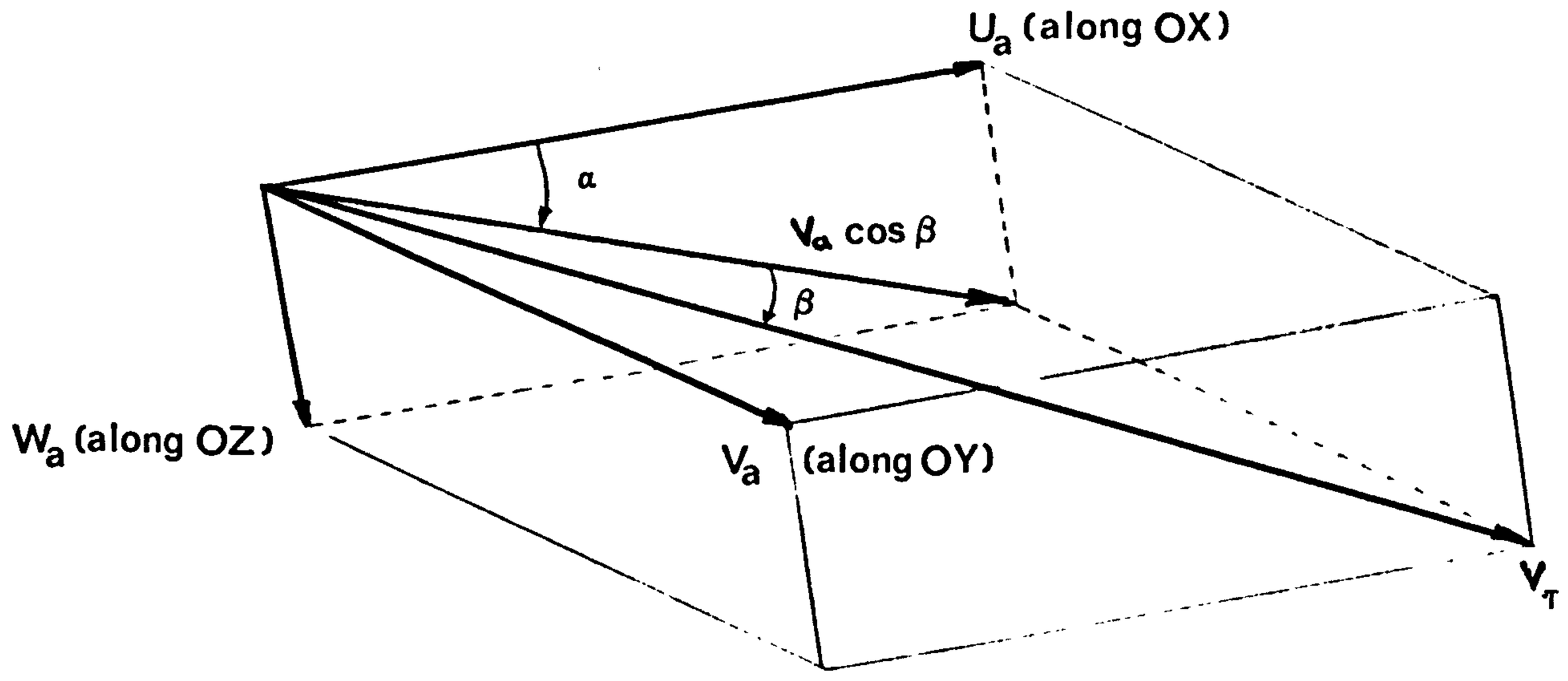


Fig. 1.4 Incidence and Sideslip Angles

$$V_T = \sqrt{U^2 + V^2 + W^2} \qquad -1.3$$

and with $\alpha = \sin^{-1} \frac{W}{\sqrt{U^2+W^2}}$; $\beta = \sin^{-1} \{V/V_T\}$ -1.4 a), b)

$$\begin{aligned} U &= V_T \cos\alpha\cos\beta \\ V &= V_T \sin\beta \\ W &= V_T \cos\beta\sin\alpha \end{aligned} \qquad -1.5$$

To proceed further with the development of the dynamics of the aircraft we must consider the aerodynamic forces and moments which act on the airframe. These forces and moments will produce changes in the aircraft's body velocities and accelerations which may be translated to velocities relative to earth axes using the equations 1.1 to 1.5. The resulting dynamic behaviour of the aircraft may then be deduced.

1.3 Euler Equations

The next stage in the analysis requires the use of the Euler equations. These equations are derived from a consideration of Newton's Second Law of motion. Two basic assumptions are required here namely :-

i) That the airframe may be considered as a rigid body i.e. that the distance between any specified points within the body does not change. All aircraft exibit some structural flexibility but for the present analysis this will be ignored.

ii) That the earth may be considered as a body fixed in space such that motion may be considered relative to the fixed earth. For most terrestrial applications this is a valid assumption but for long term navigation and extra-terrestrial flight its validity is questionable.

Given that the above assumptions are made then Newton's

Second Law is applicable within the given reference frame. This states that the time rate of change of linear momentum is the sum of all externally applied forces and that the time rate of change of angular momentum is the sum of all applied torques.

$$\text{i.e.} \quad \frac{dG}{dt} = F \quad \text{and} \quad \frac{dH}{dt} = M$$

where $\underline{G}$ is a linear momentum vector and $\underline{H}$ is an angular momentum vector both measured in the inertial co-ordinate frame. The derivation of the Euler equations also assumes that the aircraft may be considered to be moving in vacuum and is acted upon only by external forces. The mass of the displaced fluid is not considered and whilst this may be valid for some aircraft, larger aerodynamic vehicles and submarines may violate this assumption. The complete Euler equations relate the forces X,Y and Z and moments L,M and N in the OX, OY and OZ axes respectively to the angular and linear velocities in the inertial axes viz.

$$[\dot{U} + QW - RV - a_x(R^2 + Q^2) + a_y(PQ - \dot{R}) + a_z(PR + \dot{Q})]m = X$$

$$[\dot{V} + UR - PW - a_x(PQ + \dot{R}) - a_y(P^2 + R^2) + a_z(RQ - \dot{P})]m = Y$$

$$[\dot{W} + VR - QU + a_x(RP - \dot{Q}) + a_y(RQ + \dot{P}) + a_z(P^2 + Q^2)]m = Z$$

$$I_x\dot{P} + (I_z - I_y)RQ - I_{yz}(Q^2 - R^2) - I_{xz}(\dot{R} + PQ) - I_{xy}(\dot{Q} - PR) = L + Ya_z - Za_y$$

$$I_y\dot{Q} + (I_x - I_z)PR - I_{yz}(\dot{R} - PQ) - I_{xz}(R^2 - P^2) - I_{xy}(\dot{P} + QR) = M + Za_x - Xa_z$$

$$I_z\dot{R} + (I_y - I_x)QR - I_{yz}(\dot{Q} - RP) - I_{xz}(\dot{P} - RQ) - I_{xy}(P^2 - Q^2) = N + Xa_y - Ya_z$$

-1.6

Where I_x, I_y, I_z, I_{xy}, I_{yz}, I_{xz} are the moments of inertia

about the axes through the c.of g. but parallel to OX, OY and OZ, a_x, a_y, a_z are the co-ordinates of the c. of g. with respect to the origin of the OX, OY, OZ axis system and m is the mass of the aircraft. P,Q,R,U,V,W are as defined previously.

Equations 1.6 may be greatly simplified if we assume that the origin of the inertial axis system, O_{xyz}, is at the vehicle's c. of g. ($a_x, a_y, a_z = 0$) and that the vehicle is symmetric i.e. the OX, OY, and OZ axes are principal axes ($I_{xy} = I_{yz} = I_{xz} = 0$). 1.6 then reduces to

$$m(\dot{U} + QW - RV) = X$$
$$m(\dot{V} + UR - PW) = Y$$
$$m(\dot{W} + VP - QU) = Z$$

$$I_x\dot{P} + (I_z - I_y)RQ = L$$
$$I_y\dot{Q} + (I_x - I_z)PR = M$$
$$I_z\dot{R} + (I_y - I_x)QP = N$$

-1.7

The Euler equations thus allow us to define the body velocities in terms of forces and moments acting on the aircraft. We must now consider how to express these forces and moments as a function of the aircraft's aerodynamics. Firstly, however consider the effects of gravity on the aircraft.

The aircraft will, in general, be subject to gravitational forces since it will be oriented relative to the local vertical. The gravitational components of the forces and moments can be shown to be (4)

$$X_g = -mg \sin\Theta$$
$$Y_g = mg \cos\Theta \sin\Phi$$
$$Z_g = mg \cos\Theta \cos\Phi$$
$$L_g = (a_y \cos\Theta \cos\Phi - a_z \cos\Theta \sin\Phi)mg$$
$$M_g = (-a_z \sin\Theta - a_x \cos\Theta \cos\Phi)mg$$
$$N_g = (a_y \sin\Theta + a_x \cos\Theta \cos\Phi)mg$$

-1.8

Clearly, if OX, OY and OZ are taken as the principal axes, $a_x=a_y=a_z=0$, and the L_g, M_g and N_g components of equations 1.8 are zero.

1.4 Aerodynamic Forces & Moments

When a solid body moves through a fluid medium it experiences a force due to the relative motion of the solid and fluid. This force can, in general, be shown to be (1)

$$F = 1/2\ C_F\ \rho\ V_T^2\ S \qquad -1.9$$

where ρ is the density of the fluid, V_T is the velocity of the solid relative to the fluid, S is the characteristic area of the solid and C_F is a dimensionless coefficient. Considering the aircraft as a rigid body immersed in a fluid medium, air, we may write down the resulting forces and moments acting on the aircraft as

$$X_a = 1/2\ C_x\ \rho\ V_T^2\ S = \text{aerodynamic force along OX} \quad a)$$

$$Y_a = 1/2\ C_y\ \rho\ V_T^2\ S = \text{" " " OY} \quad b)$$

$$Z_a = 1/2\ C_z\ \rho\ V_T^2\ S = \text{" " " OZ} \quad c)$$

$$L_a = 1/2\ C_l\ \rho\ V_T^2\ S\ b = \text{rolling moment about OX} \quad d)$$

$$M_a = 1/2\ C_m\ \rho\ V_T^2\ S\ \bar{c} = \text{" " " OY} \quad e)$$

$$N_a = 1/2\ C_n\ \rho\ V_T^2\ S\ b = \text{" " " OZ} \quad f)$$

$$-1.10$$

remembering that a moment is generated by the product of a force and its associated moment arm. The constants S, b and $\bar{c}$ are the wing area, wing span and mean aerodynamic chord (4) respectively and clearly depend upon the aircraft geometry.

The major steady aerodynamic forces which act on the airframe are the lift and drag forces. These forces

necessarily arise in sustained flight. The lift and drag components act along and normal to the direction of the aircraft's total velocity vector since they are generated by the relative wind. Equations 1.4 a) and b) defining the incidence and sideslip angles may be used to calculate the aerodynamic components of lift and drag along the aircraft body axes OX, OY and OZ. We thus have

$$X_a = L \sin\alpha - D \cos\beta \cos\alpha$$

$$Y_a = D \sin\alpha \qquad -1.11$$

$$Z_a = - L \cos\alpha - D \cos\beta \sin\alpha$$

where α and β are the incidence and sideslip angles as given in 1.4 a) and b), L and D are the lift and drag forces respectively and are given by equation 1.9 as

$$L = 1/2\ C_L'\ \rho\ V_T^2\ S \qquad -1.12\ a)$$

$$D = 1/2\ C_D'\ \rho\ V_T^2\ S \qquad -1.12\ b)$$

where C_L' and C_D' are the lift and drag coeficients respectively and V_T, S and ρ are as previously. Before proceeding further we shall examine these lift and drag force components in more detail.

1.4.1 Lift

The lift forces acting on an aircraft are principally generated by the wings and tailplane. Control surfaces are also attached to the wings and tailplane and generate control moments by modifying the lift contribution from each flying surface. The primary control surfaces are the elevators, ailerons and the rudder. The effects of control surface deflections will be indicated below. The total lift force may be considered as being composed of the wing lift, the lift contribution from the tail plane and a tail lift contribution due to pitch rate.

The wing lift is as given in equation 1.12 a) i.e.

$$L_W = C_L \; 1/2 \; \rho \, V_T^2 \; S \qquad -1.13$$

The dimensionless coefficient, C_L, can be shown (4) to be proportional to the absolute incidence angle α and

$$C_L = C_{Lo} + a_1 \alpha \qquad -1.14$$

where a_1 is a constant of proportionality (rads s^{-1}) and C_{Lo} is the lift coefficient at zero incidence.

The lift component due to the tail plane may be considered to be made up of three components. Firstly a component due purely to the lift generated by the tail alone. Since the tail operates in the disturbed air behind the wings and fuselage the velocity of the air flow over the tail will generally be less than that over the wing section. This effect is neglected for the present but may be accomodated by defining a tail plane efficiency factor, η_t, this being basically the ratio of the relative wind speed at the tail to the total velocity V_T. The tail lift may be determined by evaluating the tail pitching moment and dividing by the tail moment arm this being given by

$$L_T' = C_{MT} \; 1/2 \; \rho \, V_T^2 \; S \; \bar{c} \; / \; l_t \qquad -1.15$$

where C_{MT} is the pitching moment coefficient from the tail and is given by

$$C_{MT} = C_{M\eta} \eta \qquad -1.16$$

where $C_{M\eta}$ is the slope of the tail pitching moment / elevator angle curve, $\bar{c}$ is the wing chord, l_t is the tail moment arm and η the elevator deflection.

The second lift contribution is due to the fuselage and is given by

$$L_{Tf} = - C_{MWBD} \; 1/2 \; \rho \, V_T^2 \; S \; \bar{c} / \; l_t \qquad -1.17$$

where C_{MWBD} is the zero pitching moment due to the body and the other symbols are as previously defined.

The final lift contribution at the tail is that due to pitch rate. A non-zero pitch rate will produce a net lift force at the tail due to the relative motion of the tailplane and the relative wind. This force component is given by

$$L_q = -\rho S V_T l_t Z_Q Q \qquad -1.18$$

where Q is the pitch rate as above and Z_Q is the lift force to pitch rate coefficient. For our applications the above three lift components are the only ones considered giving a total tail lift of

$$L_T = (C_{MT} - C_{MWBD}) -1/2 \rho V_T^2 S \bar{c}/l_t - \rho S V_T l_t Z_Q Q \qquad -1.19$$

Combining equations 1.13 and 1.19 the total lift force acting on the airframe is given by

$$L_{tot} = C_L 1/2 \rho V_T^2 S + (C_{MT} - C_{MWBD}) 1/2 \rho V_T^2 S \bar{c}/l_t - \rho S V_T l_t Z_Q Q \qquad -1.20$$

or

$$L_{tot} = 1/2 \rho V_T^2 S(C_L + (C_{MT} - C_{MWBD})\bar{c}/l_t - l_t Z_Q Q/V_T) \qquad -1.20\ a)$$

with C_L and C_{MT} given by equations 1.14 and 1.16.

1.4.2 Drag

The drag force acting on the airframe arises from a number of sources but for the remotely piloted vehicle it is satisfactory to consider a simple net drag force as given in equation 1.12 b) namely

$$D = 1/2\ C_D\ \rho\ V_T^2\ S$$

Here C_D is the drag coefficient and to a close approximation may be considered to be related to the lift coefficient C_L by

$$C_D = C_{Do} + k\ C_L^2 \qquad -1.21$$

k being a constant, the induced drag coefficient and C_{Do} is the drag coefficient at zero lift. This type of drag is termed lift dependent drag and can be shown to be (4) a close approximation to the drag acting on the complete aeroplane.

1.4.3 Pitching Moment

The pitching moment M_a is given by equation 1.10 e) as

$$M_a = 1/2\rho V_T^2\ S\ \bar{c}\ C_m$$

The coefficient C_m can be considered to be made up of two parts namely a contribution to pitching due to the wing and a contribution due to the tail. Aerodynamic moments may be considered to act at the aerodynamic centre of pressure and this is not necessarily coincident with the vehicle's centre of gravity. A moment is thus generated about the centre of gravity and pitching motion will, in general, ensue. The pitching moment due to the wing will clearly be dependent upon the wing lift and the pitching moment coefficient of the wing, C_{MW}, may be considered to be given by (4)

$$C_{MW} = C_{Mo} + C_{ML}\ C_L \qquad -1.22$$

where C_{Mo} is the aft pitching moment for zero incidence, C_{ML} is the pitching moment derivative with respect to C_L (i.e. $\partial C_M/\partial C_L$) and C_L is the lift coefficient, as before.

The contribution of the tail to the pitching moment has

already been indicated (eqn. 1.15) and is dependent upon the elevator setting. The tail pitching moment coefficient, C_{MT} is given in equation 1.16.

The total aerodynamic pitching moment is thus given by

$$M_a = 1/2\ \rho\ V_T^2\ S\ \bar{c}\ (C_{MW} + C_{MT}) \qquad -1.23$$

1.4.4 Rolling Moment

From eqn. 1.10 d) the rolling moment, L_a, is given by

$$L_a = 1/2\ C_l\ \rho\ V_T^2\ S\ b$$

C_l being the rolling moment coefficient. Rolling moments are generated by a number of factors including the deflection of the aileron. C_l will thus, in general, reflect the contributions due to each of these factors. To model the rolling moment adequately it is thus necessary to determine the rolling moment contribution due to each of the possible aerodynamic variables considered separately and sum these to produce an overall moment. This is done by employing a first order Taylor series expansion of L_a about some nominal operating point. The more important terms may then be identified either on purely intuitive grounds or by evaluation. For our purposes we shall consider the rolling moment to be made up of contributions due to pitching, yawing, sideslip and aileron deflections. C_l thus has the form

$$C_l = 1/2\ b\ L_p\ P\ /V_T\ S + 1/2\ b\ L_r\ C_L\ R\ /V_T\ S + L_v\ V\ /V_T\ S + L_\xi\ \xi\ /\ S \qquad -1.24$$

where $L_p = \frac{\partial\ L_a}{\partial\ p}$ is the rolling moment derivative w.r.t. roll rate (s rad^{-1})

$L_r = \frac{\partial L_a}{\partial r}$ is the rolling moment derivative w.r.t. yaw rate (s rad^{-1})

$L_v = \frac{L_\beta}{V_T} = \frac{1}{V_T} \frac{\partial L_a}{\partial \beta}$ is the rolling moment derivative w.r.t. sideslip (β)

$L_\zeta = \frac{\partial L_a}{\partial \zeta}$ is the rolling moment derivative w.r.t. aileron deflection ζ

the remaining symbols having their usual meaning. The above aerodynamic derivatives will normally be functions of air density, steady state velocity etc and are generally determined from wind tunnel tests on a given aircraft.

The overall rolling moment is now given by equation 1.10 d).

1.4.5 Yawing Moment

This moment is given by equation 1.10 f) as

$$N_a = 1/2\ C_n\ \rho\ V_T^2\ S\ b$$

Again, as with the rolling moment, the yawing moment coefficient, C_n, is determined by evaluating the contributions to yawing of each of the aerodynamic variables and determining the major contribution terms. For the present application only the effects on yaw due to sideslip, yaw rate, and rudder deflection are considered giving

$$C_n = N_v\ V\ /\ V_T + 1/2\ b\ N_r\ R\ /\ V_T + N_\tau\ \tau \quad -1.25$$

again N_v, N_r and N_τ are the aerodynamic derivatives relating yawing moment to sideslip velocity, yaw rate, and rudder angle (τ) respectively.

1.4.6 Side Forces

The final aerodynamic force equation required is the side force, Y_a, equation. This is given by eqn. 1.10 b) as

$$Y_a = 1/2\ C_y\ \rho\ V_T^2\ S$$

As before the value of C_y is determined by evaluating the values of the significant aerodynamic derivatives. The rudder is the control surface which contributes most to the value of this force component. For our application the sideslip, yaw rate and rudder deflection components are the only ones considered giving

$$C_y = 2\ Y_v\ V\ /V_T + b\ Y_r\ R\ /V_T + 2\ Y_\tau\ \tau \qquad -1.26$$

As before the values of Y_v, Y_r and Y_τ are determined from wind tunnel testing and are the aerodynamic derivatives relating sideforce to sideslip velocity, yaw rate and rudder angle, τ , respectively. Equation 1.10 b) then gives the sideforce components.

1.4.7 Additional Pitching Moments

A pitching moment is also produced due to the action of the wing lift and pitch rate lift about the aircraft's centre of gravity. These components are essentially given by the products of L_w and L_q (eqns. 1.13 and 1.18) and the appropriate moment arms.

The complete aerodynamic force and moment equations may now be written. These equations also include the gravitational components as given by equations 1.8 and the lift and drag components of eqns. 1.11. The value of the sideslip angle, β , is taken to be zero since, in general,

the sideslip angle is small compared with the incidence, α . We thus obtain the following equations:

$$X = X_E - D\cos\alpha + L_{tot}\sin\alpha - mg\sin\Theta \qquad a)$$

$$Y = Y_a + mg\cos\Theta\sin\Phi \qquad b)$$

$$Z = -L_{tot}\cos\alpha - D\sin\alpha + mg\cos\Theta\cos\Phi \qquad c)$$

$$L = L_E + L_a \qquad d)$$

$$M = M_a + L_w(cg - 0.25)\bar{c} - (l_t + 0.25 - cg)L_q \qquad e)$$

$$N = N_a \qquad f)$$

where the forces and moments are as defined above and cg is the position of the aircraft centre of gravity. X_E is the thrust force due to the engine whilst L_E is the rolling moment component due to the engine. The model of the engine will now be briefly examined.

1.4.8 Thrust Equations

All powered aircraft require to have at least one propulsion unit whose primary purpose is to provide a thrust force acting so as to increase (or decrease) the forward velocity of the aircraft.

This forward velocity generates the lift forces necessary to maintain flight. The engine thrust must also overcome or equal the drag force incurred due to the aircraft's motion through the air. The type of propulsion unit used varies from aircraft to aircraft but may be of the piston engined propeller, turboprop, jet, etc type. To devise a set of equations relating thrust to say pilot throttle demand (or angle) presents considerable difficulty and a generalised model is not possible due to the many types and configurations of engines used. In the present application we shall consider a specific vehicle configuration, this being the Machan (6) remotely piloted

vehicle refered to in the Introduction . The layout of this aircraft is shown schematically in Fig. 1.5 . The propulsion unit is a small piston engine driving a four bladed propeller. The propeller is housed in a nacel mounted at the rear of the aircraft and forming part of the tailplane assembly. Airflow is directed over the propeller through a duct, slightly forward of the nacel. The thrust force provided by this arrangement is thus along the longitudinal a_x axis. The engine also develops a torque, due to the airscrew rotation, about the OX body axis and this contributes to the rolling moment equation. The engine power demand is controlled by the setting of the throttle control, T_H.

It is not intended to detail the derivation of the thrust equations, but merely to quote the results. The nominal power delivered by the engine, P_{nom}, is given by

$$P_{nom} = P_{max}\ T_H \qquad -1.28$$

where P_{max} is the maximum engine power and T_H the throttle demand (0 - 100 %). The actual power delivered by the airscrew, P_{act} is given by

$$P_{act} = P_{nom}\ \eta_p \qquad -1.29$$

where η_p is the propeller efficiency.

The power delivered to the airscrew is related to the resulting thrust force, X_E, by a simple 1st. order lag viz :-

$$\dot{X}_E = \frac{(P_{act} - X_E\ U_2)}{K_e} \qquad -1.30$$

where U_2 is the air flow rate, in ms^{-1}, over the propeller, i.e. through the duct, and K_e is the engine rise rate in metres, $X_E\ U_2$ then being the power supplied. We also have

TEXT BOUND INTO

THE SPINE

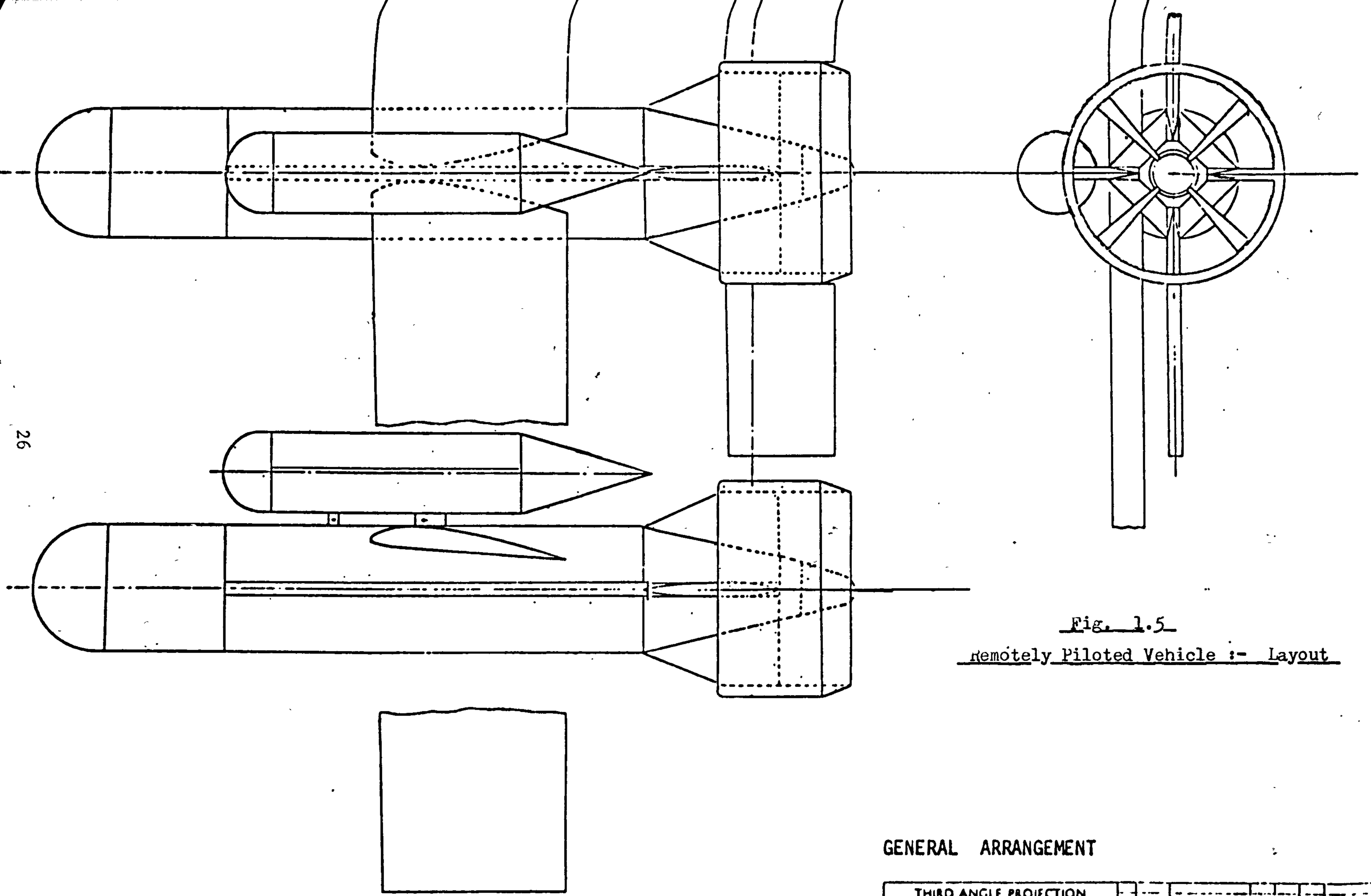


Fig. 1.5

Remotely Piloted Vehicle :- Layout

GENERAL ARRANGEMENT

THIRD ANGLE PROJECTION

that the power supplied is the rate of increase in energy of the working fluid and is given by

$$\dot{E} = 1/2\ \rho A_D\ U_2\ (U_3^2 - U^2) = X_E\ U_2 \qquad -1.31$$

where A_D is the duct area, U_3 is the flow rate in the propeller wake and ρ and U are as defined previously. Re-arranging 1.31 and making U_3 the subject of the equation

$$U_3 = (2X_E/\rho\ A_D + U^2)^{1/2} \qquad -1.32$$

The flow rate through the duct, U_2, can be derived by considering the thrust equation i.e.

$$X_E = \rho\ A_D\ U_2\ (U_3 - U)$$

Thus

$$U_2 = \frac{X_E}{\rho\ A_D\ (U_3 - U)} \qquad -1.33$$

Additionally the engine speed in rpm is given by

$$\text{rpm} = U_2/P_p \ \text{x}\ 60 \qquad \text{rev/min} \qquad -1.34$$

where P_p is the propeller pitch. The torque due to the airscrew can be derived by relating this to the nominal engine power as

$$L_E = 2\pi\ \text{x}\ \text{rpm}/60 = P_{nom}$$

$$L_E = \frac{P_{nom}\ 60}{2\pi\ \text{rpm}} \qquad -1.35$$

with L_E as in equations 1.27 .

1.5 Summary

The above discussion and analysis has described a set of non-linear equations on which to base a simulation study of the aircraft. Whilst the equations are not entirely general they are nevertheless considered satisfactory for the application proposed and relate primarily to the remotely piloted vehicle referred to in 1.4.8. For simulation purposes a digital simulation will be undertaken, the basic sequence of solution being

i) Resolve any wind forces into aircraft body axes using equations 1.1.

ii) Evaluate the individual aerodynamic forces and moment components using the current values of control angles and aircraft attitude and angular velocities using the results of section 1.3.

iii) Use equations 1.27 to evaluate the X, Y and Z forces and L, M and N moments including the gravitational components.

iv) Integrate the Euler equations 1.7 at each time step

v) Evaluate the resulting aircraft attitudes in earth co-ordinates using equations 1.2 and evaluate the new aircraft position, height, easting and northing positions using equations 1.1.

vi) Repeat for the next time step.

In the following chapters we will investigate the development of this digital simulation in particular with respect to the implementation machine and programming languages. Before this, however, we shall briefly examine the methods available for linearisation of the complete aircraft equations of motion with a view to developing a state space model of the aircraft.

CHAPTER 2

Linearisation of Vehicle Equations

The complete aerodynamic equations developed in Chapter 1 are too cumbersome to employ directly for control system design. To be useful, a set of linear equations must be developed which preserve the essential dynamics of the non-linear equations but are analytically tractable. It is inevitable that in determining such a model some gross simplifications must be made since the aircraft is inherently a very difficult system to model accurately . Indeed, some basic assumptions have already been made in determining the equations of Chapter 1 which are not necessarily true in certain flight configurations. The particular linear model which is obtained will largely be determined by the assumptions which are made in it's derivation and for this reason some care is necessary when making these assumptions. Ideally for a complete linear model the non-linear equations should be rigorously adhered to and a linearisation undertaken at each point in the flight envelope. This effectively leads to a time varying linear model although it may be found that some model parameters change little over the flight envelope or are sufficiently small so as to be negligble. Conventionally, however, a set of assumptions are first formulated, these allowing certain gross simplifications to be made in deriving a linear time invariant model. This method is perfectly valid for small perturbations when, for example, stability of the aircraft is of interest as we are only interested in how, or indeed if, the aircraft returns to its undisturbed equilibrium position after a small perturbation in a control surface position.

Such analyses lead to the design of so called stability augmentation systems and classical autopilots which aim to alleviate the pilot's workload in controlling a potentially unstable aircraft. For control of the aircraft 'in the

large' i.e. when large changes in flight variables are occurring due to possibly large deflections in one or more of the control surfaces a more complete linear model of the aircraft is needed.

In this Chapter it is intended to examine a typical classical 'small perturbation' derivation of a linear aircraft model. The analysis presented will at least serve to exemplify the characteristic aircraft modes and the nature of the resulting control problem. Some physical justification of the overall aircraft behaviour may then be made and an insight gained into possible interactions between motions.

2.1 General Techniques

The starting point for any linearisation is the appreciation that any motion may be considered as being composed of two parts. Firstly, a component which constitutes an average steady-state or trimmed condition and secondly a perturbation component about this nominal operating point. Note that the trimmed condition in an aircraft necessarily implies zero rotational and translational acceleration, although translational and rotational velocities need not be zero under these conditions. If we now perturb the motion we will, in general, simply be adding a small additional component into each of the trimmed conditions. As an example consider the velocity component along say the OX body axis, U. Using the o subscript to signify trimmed values and the lower case u to signfy a small perturbation about U_o then, in the perturbed condition

$$U = U_o + u$$

similarly $P = P_o + p$; $\Theta = \Theta_o + \theta$; etc. To obtain the relevant dynamic equations of motion due to small

perturbations in the variable we first evaluate the trimmed equations and the perturbed equations. Subtracting the former from the latter then gives the relevant dynamics. Note here that since u, p, θ etc. are small, second order and product terms in these variables are neglected whilst small angle formulae for trignometric functions can be applied (i.e. $\sin\theta = \theta$; $\cos\theta = 1$; θ small). Consider, as an example. the simplified X force component equation (eqn. 1.7 a)) with the corresponding gravitational force component (eqn. 1.8 a)) included i.e.

$$m(\dot{U} + QW - RV) + mg \sin\Theta = X \qquad -2.1$$

The trimmed form of this equation is given by

$$m(W_oQ_o - R_oV_o) + mg \sin\Theta_o = X_o \qquad -2.2$$

the perturbed form of equation 2.1 can be obtained as

$$m((\dot{u} + Q_oW_o + qW_o + Q_ow + qw) - (R_oV_o + rV_o + vR_o + rv)) + mg \sin(\Theta_o + \theta) = dX \qquad -2.3$$

Subtracting 2.2 from 2.3 and neglecting 2nd. order products

$$m(\dot{u} + W_oq + Q_ow - V_or - R_ov + \theta g \cos\Theta_o) = dX \qquad -2.4\ a)$$

Similarly the other five equations in 1.7 yield

$$m[\dot{v} + U_or + R_ou - W_op - P_ow - (g \cos\Theta_o \cos\Phi_o)\varphi + (g \sin\Theta_o \cos\Phi_o)\theta] = dY \qquad -2.4\ b)$$

$$m[\dot{w} + V_op + P_ov - U_oq - Q_ou + (g \cos\Theta_o \cos\Phi_o)\varphi + (g \sin\Theta_o \cos\Phi_o)\theta] = dZ \qquad -2.4\ c)$$

$$\dot{p}I_x + (Q_o r + R_o q)(I_z - I_y) = dL \qquad -2.4\ d)$$

$$\dot{q}I_y + (P_o r + R_o p)(I_x - I_z) = dM \qquad -2.4\ e)$$

$$\dot{r}I_z + (P_o q + Q_o p)(I_y - I_x) = dN \qquad -2.4\ f)$$

A similar analysis applied to equations 1.2 relating P, Q and R to Φ, Θ and Ψ yeilds

$$p = \dot{\varphi} - \dot{\psi}\sin\Theta_o - \theta(\dot{\Phi}_o \cos\Theta_o) \qquad -2.5\ a)$$

$$q = \dot{\theta}\cos\Phi_o - \theta(\dot{\Psi}_o \sin\Theta_o \sin\Phi_o) + \varphi(\dot{\Psi}_o \cos\Theta_o \cos\Phi_o + \dot{\Theta}_o \sin\Phi_o) + \dot{\psi}\cos\Theta_o \sin\Phi_o \qquad -2.5\ b)$$

$$r = \dot{\psi}\cos\Theta_o \cos\Phi_o - \varphi(\dot{\Psi}_o \cos\Theta_o \sin\Phi_o + \dot{\Theta}_o \cos\Phi_o) - \dot{\theta}\sin\Phi_o - \theta(\dot{\Psi}_o \sin\Theta_o \cos\Phi_o) \qquad -2.5\ c)$$

It is also possible to apply a similar analysis to the T matrix of 1.1 which yields a rather unwieldy set of equations, 2.6, shown in Fig. 2.1.

We now note that the equations 2.4, 2.5 and 2.6 are linear but represent a daunting set of equations from which useful results can be derived. Note also that the expressions for dX, dY, ..., dN have not yet been considered and these will, in general, be more complex than the above. The major reason for this complexity is the very general set of trim conditions chosen. By restricting the trim conditions to a specific flight configuration, however, the equations become somewhat more tractable.

Let us now consider the linearisation of the aerodynamic forces and moments, as mentioned above.

2.1.1 Aerodynamic Forces

The aerodynamic forces and moments acting on the aircraft may in general be considered to be functions of (U,

$$
\begin{bmatrix}
\left\{\begin{matrix}\underline{[\cos\Psi_0\cos\Theta_0]}\\ -\psi(\sin\Psi_0\cos\Theta_0)\\ -\theta(\cos\Psi_0\sin\Theta_0)\end{matrix}\right\} & \left\{\begin{matrix}\underline{[\sin\Psi_0\cos\Theta_0]}\\ +\psi(\cos\Psi_0\cos\Theta_0)\\ -\theta(\sin\Psi_0\sin\Theta_0)\end{matrix}\right\} & \left\{\begin{matrix}\underline{[-\sin\Theta_0]}\\ -\theta(\cos\Theta_0)\end{matrix}\right\} \\
\left\{\begin{matrix}\underline{[\cos\Psi_0\sin\Theta_0\sin\Phi_0-\sin\Psi_0\cos\Phi_0]}\\ -\psi(\sin\Psi_0\sin\Theta_0\sin\Phi_0+\cos\Psi_0\cos\Phi_0)\\ +\theta(\cos\Psi_0\cos\Theta_0\sin\Phi_0)\\ +\varphi(\cos\Psi_0\sin\Theta_0\cos\Phi_0+\sin\Psi_0\sin\Phi_0)\end{matrix}\right\} & \left\{\begin{matrix}\underline{[\sin\Psi_0\sin\Theta_0\sin\Phi_0+\cos\Psi_0\cos\Phi_0]}\\ +\psi(\cos\Psi_0\sin\Theta_0\sin\Phi_0-\sin\Psi_0\cos\Phi_0)\\ +\theta(\sin\Psi_0\cos\Theta_0\sin\Phi_0)\\ +\varphi(\sin\Psi_0\sin\Theta_0\cos\Phi_0-\cos\Psi_0\sin\Phi_0)\end{matrix}\right\} & \left\{\begin{matrix}\underline{[\cos\Theta_0\sin\Phi_0]}\\ -\theta(\sin\Theta_0\sin\Phi_0)\\ +\varphi(\cos\Theta_0\cos\Phi_0)\end{matrix}\right\} \\
\left\{\begin{matrix}\underline{[\cos\Psi_0\sin\Theta_0\cos\Phi_0+\sin\Psi_0\sin\Phi_0]}\\ +\psi(-\sin\Psi_0\sin\Theta_0\cos\Phi_0+\cos\Psi_0\sin\Phi_0)\\ +\theta(\cos\Psi_0\cos\Theta_0\cos\Phi_0)\\ +\varphi(-\cos\Psi_0\sin\Theta_0\sin\Phi_0+\sin\Psi_0\cos\Phi_0)\end{matrix}\right\} & \left\{\begin{matrix}\underline{[\sin\Psi_0\sin\Theta_0\cos\Phi_0-\cos\Psi_0\sin\Phi_0]}\\ +\psi(\cos\Psi_0\sin\Theta_0\cos\Phi_0+\sin\Psi_0\sin\Phi_0)\\ +\theta(\sin\Psi_0\cos\Theta_0\cos\Phi_0)\\ -\varphi(\sin\Psi_0\sin\Theta_0\sin\Phi_0+\cos\Psi_0\cos\Phi_0)\end{matrix}\right\} & \left\{\begin{matrix}\underline{[\cos\Theta_0\cos\Phi_0]}\\ -\theta(\sin\Theta_0\cos\Phi_0)\\ -\varphi(\cos\Theta_0\sin\Phi_0)\end{matrix}\right\}
\end{bmatrix}
$$

-2.6

Fig. 2.1 Steady State Direction Cosine Array

$V, W, P, Q, R, \dot{U}, \dot{V}, \dot{W}, \dot{P}, \dot{Q}, \dot{R}, U_G, V_G, W_G, U_G, V_G, W_G, \rho$) in addition to any other variables such as Mach number, Reynolds number, angle of attack and sideslip. (Note that U_G, V_G, W_G are wind gust velocity components in earth axes). If the forces are continuous functions of all of the above variables then each force or moment may be expressed as a Taylor series expansion of the force about some nominal operating point i.e.

$$F = F_o + \left(\frac{\partial F}{\partial \Lambda_1}\right)_0 \alpha_1 + \left(\frac{\partial F}{\partial \Lambda_2}\right)_0 \alpha_2 + \ldots\ldots \qquad -2.7$$

here we neglect second order or higher derivatives since these will, in general, be small for small perturbations α_i in the Λ_i's. Considering the potentially very large number of dependent variables equation 2.7 looks formidably complex. The situation is eased somewhat, however, by again specialising the trim conditions and by an intuitive knowledge of the likely values of some of the partial derivatives. These are termed the aerodynamic derivatives and are evaluated holding the remaining primary motion variables constant.

As an example consider the aircraft trimmed so as to maintain a steady incidence, α , in the XZ plane as shown in Fig. 2.2. Neglecting sideslip, i.e. $\beta = 0$, then the lift and drag components acting on the aircraft are as shown and the aerodynamic forces X and Z are given by equation 1.11 viz. :-

$$X_a = - D \cos\alpha + L \sin\alpha$$

$$Z_a = - L \cos\alpha - D \sin\alpha$$

replacing D and L by their equivalent expressions from equations 1.12 a) and b) then

$$X_a = 1/2 \ \rho V_T^2 S \ (- C_D \cos\alpha + C_L \sin\alpha)$$

$$Z_a = 1/2 \ \rho V_T^2 S \ (- C_L \cos\alpha - C_D \sin\alpha) \qquad -2.8$$

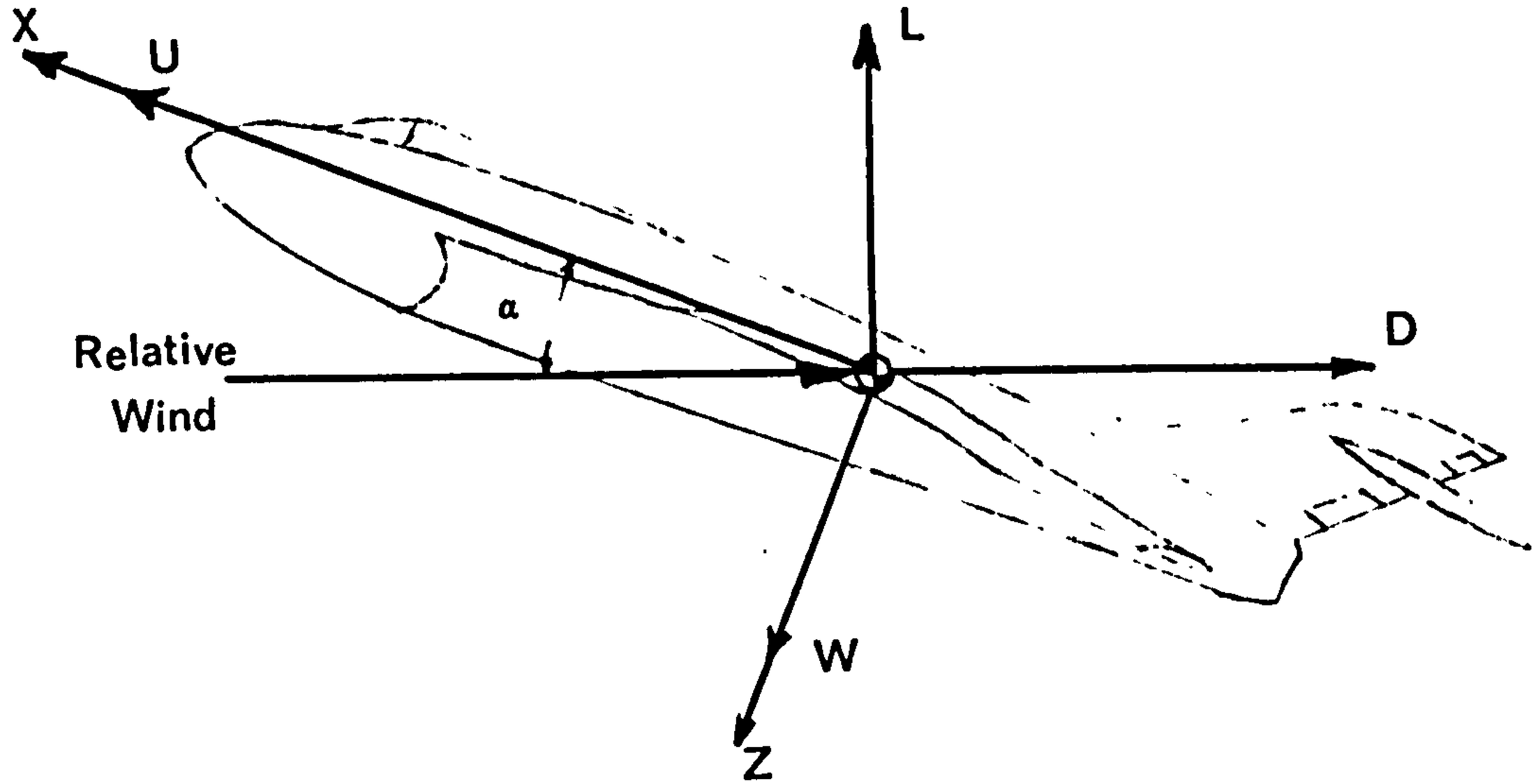


Fig. 2.2 Longitudinal Lift and Drag Forces

Assuming that C_D and C_L are functions of only Mach number, M^*, and incidence angle, α , then we may expand equations 2.8 as

$$\frac{\partial X_a}{\partial \Lambda} = \frac{\partial X_a}{\partial M}\frac{\partial M}{\partial \Lambda} + \frac{\partial X_a}{\partial V_T}\frac{\partial V_T}{\partial \Lambda} + \frac{\partial X_a}{\partial \alpha}\frac{\partial \alpha}{\partial \Lambda}$$

where Λ is the given primary motion variable. Thus

$$\frac{\partial X_a}{\partial \Lambda} = \frac{1}{2} V_T^2 S - \frac{\partial C_D}{\partial M} \cos \alpha_o - \frac{\partial C_L}{\partial M} \sin \alpha_o \frac{\partial M}{\partial \Lambda}$$

$$+ \rho V_T S (-C_D \cos \alpha_o + C_L \sin \alpha_o) \frac{\partial V_T}{\partial \Lambda}$$

$$+ \frac{1}{2} \rho V_T^2 S (- \frac{\partial C_D}{\partial \alpha} \cos \alpha_o + \frac{\partial C_L}{\partial \alpha} \sin \alpha_o + C_D \sin \alpha_o$$

$$+ C_L \cos \alpha_o) \qquad -2.9$$

A similar expression is clearly obtainable for $\frac{\partial Z_a}{\partial \Lambda}$.

The primary motion variables for this trim condition are U and W and expressions for

$$\frac{\partial M}{\partial U} ; \frac{\partial V_T}{\partial U} ; \frac{\partial \alpha}{\partial U} ; \frac{\partial M}{\partial W} ; \frac{\partial V_T}{\partial W} ; \frac{\partial \alpha}{\partial W}$$

may be obtained from equations 1.11. For the case of W constant dW = 0 and

$$\frac{\partial V_T}{\partial U} = \cos \alpha_o ; \frac{\partial \alpha}{\partial U} = - \frac{\sin \alpha_o}{V_T} ; \frac{\partial M}{\partial U} = \frac{\cos \alpha_o}{a}$$

and for U constant, dU = 0 thus

$$\frac{\partial V_T}{\partial W} = \sin \alpha_o ; \frac{\partial \alpha}{\partial W} = \frac{\cos \alpha_o}{V_T} ; \frac{\partial M}{\partial W} = \frac{\sin \alpha_o}{a}$$

-2.10

* :- Mach no. is defined as the ratio V_T/a where a is the speed of sound and V_T, the total aircraft velocity.

where a is the velocity of sound. Substituting equations 2.10 into equations 2.9 gives

$$\frac{\partial X_a}{\partial U} = \rho V_T S [(-C_D - C_{D\alpha}) \cos^2 \alpha_o$$

$$+ (1/2 (C_{D\alpha} - C_L) + C_L + C_{Lu}) \sin \alpha_o \cos \alpha_o$$

$$- 1/2(C_{L\alpha} + C_D) \sin^2 \alpha_o] \quad -2.11$$

a similar expression is obtainable for $\frac{\partial X_a}{\partial W}$ namely

$$\frac{\partial X_a}{\partial W} = \rho V_T S [1/2(C_L - C_{D\alpha}) \cos^2 \alpha_o$$

$$+ (1/2(C_{L\alpha} + C_D) - C_D - C_{D\alpha}) \sin \alpha_o \cos \alpha_o$$

$$- (C_L + C_{Lu}) \sin^2 \alpha_o] \quad -2.12$$

Note that the standard notation is used for the partial derivatives i.e. $\frac{\partial C_D}{\partial \alpha} = C_{D\alpha}$; $C_{Lu} = (M/2)(\frac{\partial C_L}{\partial M})$; etc.

A similar analysis for $\frac{\partial Z_a}{\partial U}$ and $\frac{\partial Z_a}{\partial W}$ gives

$$\frac{\partial Z_a}{\partial U} = \rho V_T S [(- C_L - C_{Lu}) \cos^2 \alpha_o$$

$$+ (1/2(C_{L\alpha} + C_D) - C_D - C_{Du}) \sin \alpha_o \cos \alpha_o$$

$$- 1/2 (C_L - C_{D\alpha}) \sin^2 \alpha_o] \quad -2.13$$

$$\frac{\partial Z_a}{\partial W} = \rho V_T S [1/2(C_L + C_D) \cos^2 \alpha_o -$$

$$(1/2(C_{D\alpha} - C_L) + C_L + C_{Lu}) \sin \alpha_o \cos \alpha_o$$

$$- (C_D + C_{Du}) \sin^2 \alpha_o] \quad -2.14$$

The above example is quoted to indicate the degree of complexity of the expressions for the aerodynamic derivatives even for this relatively specialised trim condition. The choice of $\alpha_o = 0$ does, however, simplify the above expressions considerably. A choice of $\alpha_o = 0$ clearly influences the individual contribution each of the aerodynamic derivative terms makes to the total value of $\partial X_a / \partial U$, for example.

Fig. 2.3 shows the analagous situation in the XY plane. Here a continuous sideslip angle, β , exists giving rise to a side lift force L and a drag force D. The aerodynamic derivatives for this situation may be inferred from equations 2.11 to 2.14 by replacing α_o by β_o and W with V. The values of side lift coefficient, C_L and side drag coefficient, C_D, will normally be different however. Note that in the XZ plane lift is required to maintain steady flight and a corresponding drag force is incurred. In the XY plane side lift is normally undesirable whilst manoeuvres requiring constant angles of sideslip to be set up are not normally required.

It is thus convenient to separate the two planes of motion into longitudinal (XZ) and lateral (XY) motions. It is assumed here that the lateral trim conditions are such as to render steady state values of roll rate, P, yaw rate, R, sideslip velocity, V and azimuth angle (heading) ψ of zero. Additionally, it is clear that perturbed longitudinal motion does not give rise to appreciable perturbations in the lateral motion otherwise the above assumption is violated. Also, control surface deflections do not necessarily give rise to appreciable cross coupling between lateral and longitudinal motions. Hence aileron and rudder control actuation will only affect lateral motion appreciably whilst elevator actuation on its own will affect longitudinal motion.

This separation of the dynamics simplifies analysis and will, in general, be valid for most aircraft manoeuvres. At a later stage in the current project lateral/longitudinal coupling will be considered in greater detail but it is

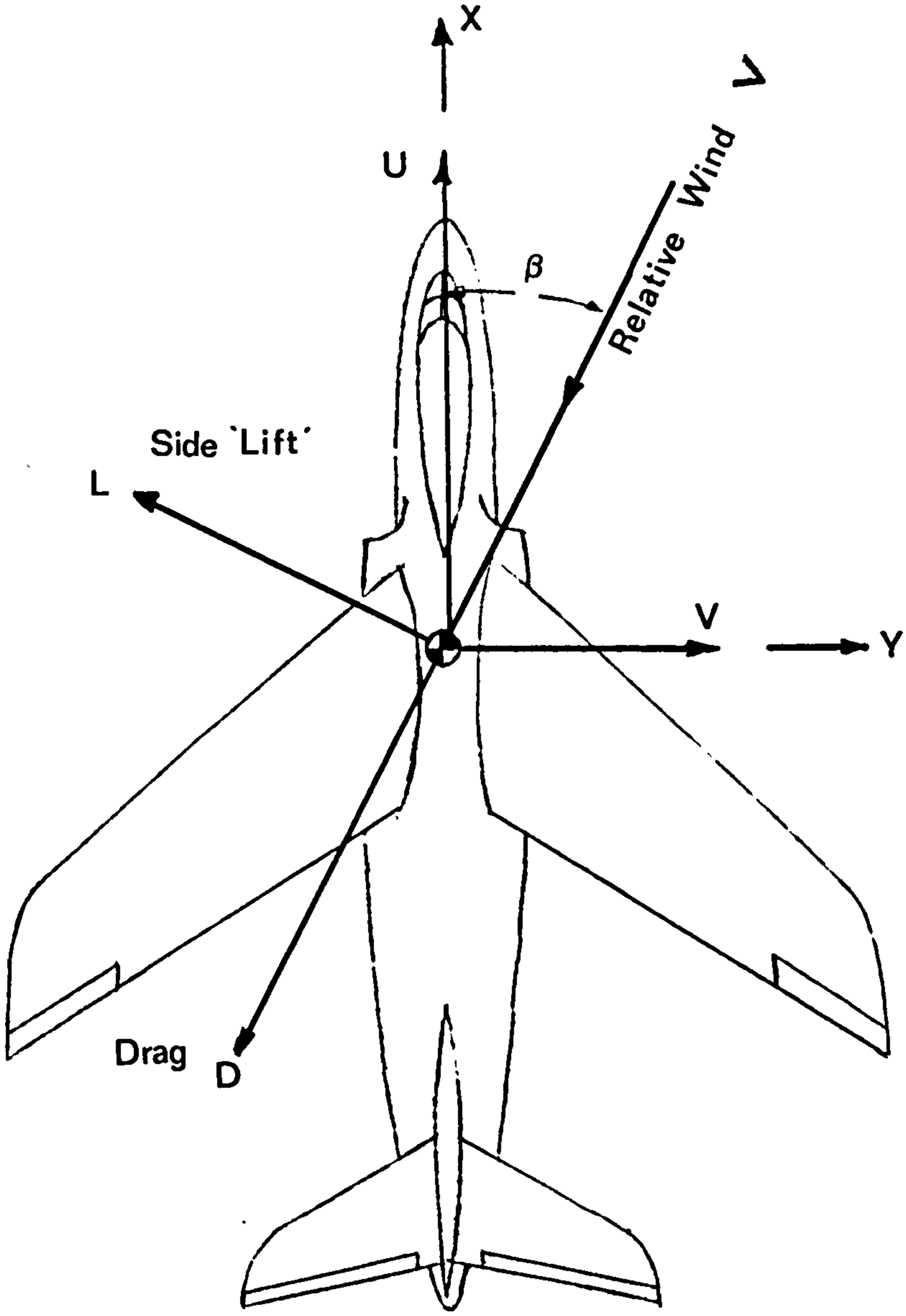


Fig. 2.3 Lateral Lift and Drag Forces

convenient to adopt the uncoupled approach in the following discussion.

2.1.2 Thrust Equations

To complete a general discussion of the linearisation of the vehicle equations we shall finally examine the effect of the thrust force acting on the aircraft.

In section 1.4.8 the equations for the thrust force specific to a given aircraft were developed. Whilst it is true that the equations for a given aircraft geometry are relatively specialised we may nevertheless consider the general case of a power plant developing a thrust force, T, along a thrust line inclined at an angle ϵ_T to the relative wind as in Fig. 2.4. Only the longitudinal motions are considered here and an initial angle of attack (incidence) α_o is assumed. The thrust vector is normally displaced from the origin of the body axes (often the aircraft's centre of gravity) by a distance e_T, the thrust eccentricity. For this situation we may write down the pertinent force and moment equations as

$$\begin{aligned} X_T &= T \cos (\epsilon_T - \alpha_o) \\ Z_T &= - T \cos (\epsilon_T - \alpha_o) \qquad\qquad -2.15 \\ M_T &= T \; e_T \end{aligned}$$

For the thrust considered to be a function of only the total forward velocity V_T, power plant demand setting δ_T and air density then the perturbed form of equations 2.15 gives

$$\delta X_T = \cos (\epsilon_T - \alpha_o) \; \frac{\partial T}{\partial V_T} \; \delta V_T + \frac{\partial T}{\partial \delta_T} \; \delta \delta_T$$

$$\text{or} \quad \delta X_T = \cos (\epsilon_T - \alpha_o) \left\{ \frac{\partial T}{\partial V_T} \left(\frac{\partial V_T}{\partial U} u + \frac{\partial V_T}{\partial W} w \right) + \frac{\partial T}{\partial \delta_T} \delta \delta_T \right\}$$

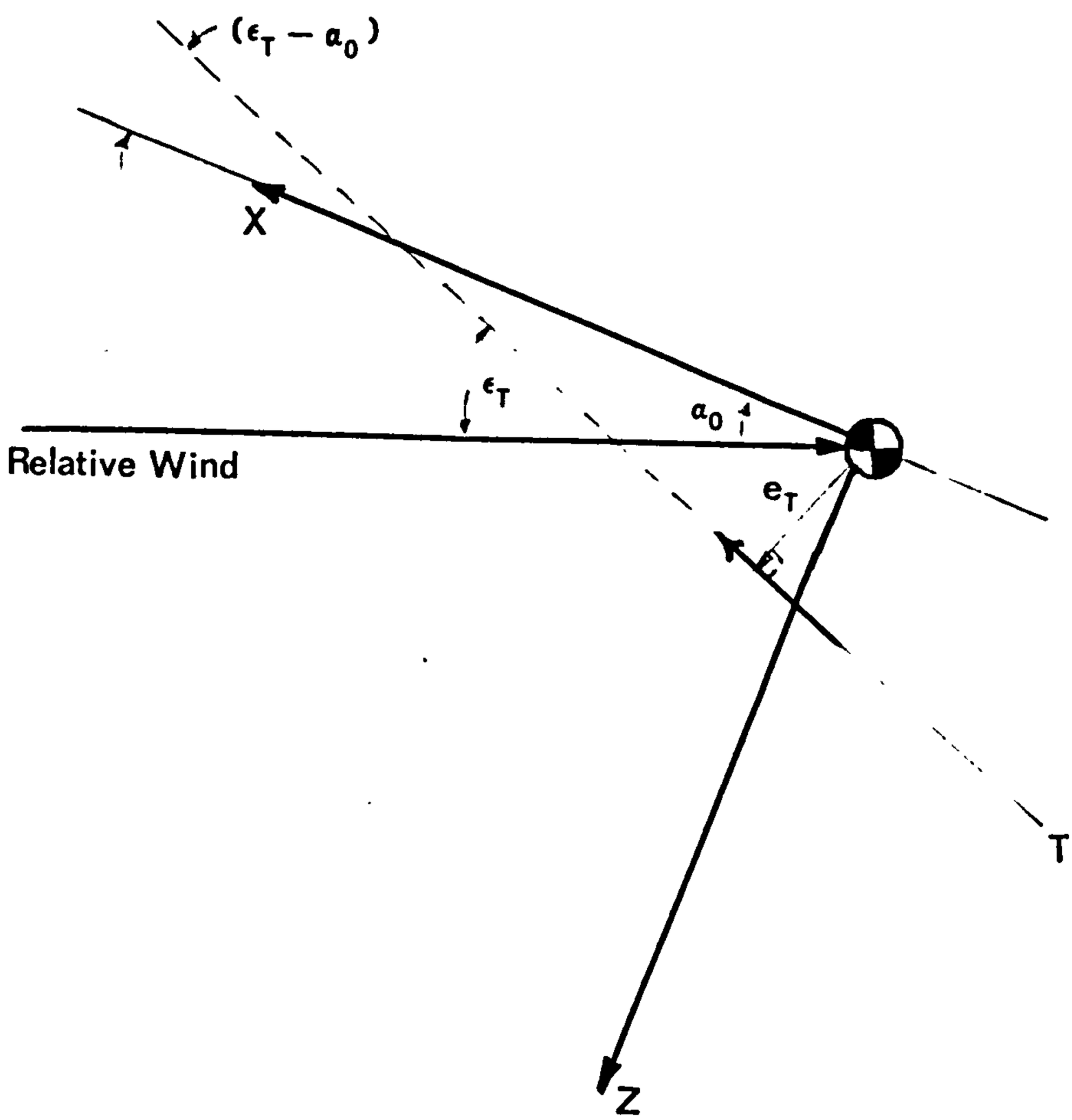


Fig. 2.4 Thrust Force Orientation

similarly

$$\delta Z_T = - \sin (\epsilon_T - \alpha_o)\left\{\frac{\partial T}{\partial V_T}\left(\frac{\partial V_T}{\partial U} u + \frac{\partial V_T}{\partial W} w\right) + \frac{\partial T}{\partial \delta_T} \delta\delta_T\right\}$$

and applying the results in equations 2.10 to determine the overall aerodynamic derivatives gives

$$\frac{\partial X_T}{\partial U} = \frac{\partial T}{\partial V_T} \cos \alpha_o \cos (\epsilon_T - \alpha_o) \quad a)$$

$$\frac{\partial X_T}{\partial W} = \frac{\partial T}{\partial V_T} \sin \alpha_o \cos (\epsilon_T - \alpha_o) \quad b)$$

$$\frac{\partial X_T}{\partial \delta_T} = \frac{\partial T}{\partial V_T} \cos (\epsilon_T - \alpha_o) \quad c)$$

$$\frac{\partial Z_T}{\partial U} = - \frac{\partial T}{\partial V_T} \cos \alpha_o \sin (\epsilon_T - \alpha_o) \quad d)$$

-2.16

$$\frac{\partial Z_T}{\partial W} = - \frac{\partial T}{\partial V_T} \sin \alpha_o \sin (\epsilon_T - \alpha_o) \quad e)$$

$$\frac{\partial Z_T}{\partial \delta_T} = - \frac{\partial T}{\partial V_T} \sin (\epsilon_T - \alpha_o) \quad f)$$

Equations 2.16 may now be combined with equations 2.11, 2.13, 2.14 and 2.12 to derive the overall value of the respective aerodynamic derivatives.

In the steady state the pitching moment due to the offset thrust line must be balanced by a corresponding aerodynamic moment due to the control surfaces in order to maintain steady state equilibrium thus,

$$M_a = T\, e_T + 1/2\, \rho\, V_T^2\, S\, \bar{c}\, C_m = 0 \quad -2.17$$

where the aerodynamic pitching moment is as given in

equations 1.23. Considering only variations in the thrust due to V_T, δ_T, and neglecting the fact that C_m varies with Mach number then

$$\delta M_a = e_T \left\{ \frac{\partial T}{\partial V_T} \left(\frac{\partial V_T}{\partial U} u + \frac{\partial V_T}{\partial W} w \right) + \frac{\partial T}{\partial \delta_T} \delta\delta_T \right\} +$$

$$\rho V_o S \bar{c} C_m \left(\frac{\partial V_T}{\partial U} u + \frac{\partial V_T}{\partial W} w \right) \qquad -2.18$$

From equation 2.17

$$\rho V_o S \bar{c} C_m = - \frac{2Te_T}{V_o} \qquad -2.19$$

Thus combining equations 2.18, 2.19, and equations 2.10 gives

$$\delta M_a = e_T \left[\left(\frac{\partial T}{\partial V_T} - \frac{2T_o}{V_o} \right) (u \cos \alpha_o + w \sin \alpha_o) + \frac{\partial T}{\partial \delta_T} \delta\delta_T \right] \qquad -2.20$$

where, as before, the o suffices refer to steady state values of the variables.

As mentioned above the use of a very general set of trim conditions leads to a very complex linear model of the aircraft. A very useful insight into typical aircraft response modes is to be gained by specialising the trim conditions. In particular, the stability aspects of the aircraft may be examined using the so called 'stability axis' transformation. In this analysis the trim conditions are chosen so as to reduce the complexity of the linearised aircraft equations sufficiently to allow classical stability augmentation systems to be designed.

Having thus guaranteed reasonable stability margins for

the closed-loop system, autopilots, which attempt to maintain given pitch rate, height, etc. demands can be implemented as an outer loop design. Stability axis analyses are generally restricted to small perturbations about the specified trim condition but have been shown to provide satisfactory performance over a relatively large range of the aircraft flight envelope. For large signal purposes, e.g. navigation, this analysis technique is of limited value.

In order to gain some insight into the stability axis analysis and hence into the general control problem applied to aircraft we shall now consider this approach in more detail.

2.2 Stability Axes

For relatively small perturbations about the steady state and for short term dynamic analysis it is often valid to assume that the aircraft is trimmed to straight and level flight over a flat earth. In addition, the previously outlined assumptions which have been used to derive the aerodynamic force and moment balance equations are taken to hold. The principal reductions in the linearised equations of motion then arise since $\Phi_o = P_o = Q_o = R_o = 0$ and correspondingly $\dot{\Psi}_o = \dot{\Theta}_o = \dot{\Phi}_o = 0$. Note that as indicated above $\Phi_o = P_o = R_o = V_o = 0$ leads to the separation of the aircraft equations into the lateral and longitudinal sets, this will be clarified in the following discussion.

To complete our stability axis model a final assumption is made that the body axes system is oriented such that the forward speed, along OX, lies along the direction of the aircraft's total velocity vector, V_T. The OX axis then lies along this vector and $V_o = W_o = 0$. It may be convenient to assume an intial orientation of the velocity vector of γ_o, as shown in Fig. 2.5 relative to the horizontal. Perturbed motion then gives rise to the establishment of an incidence angle between the OX body axis and the initial direction of the total velocity vector V_T. This initial

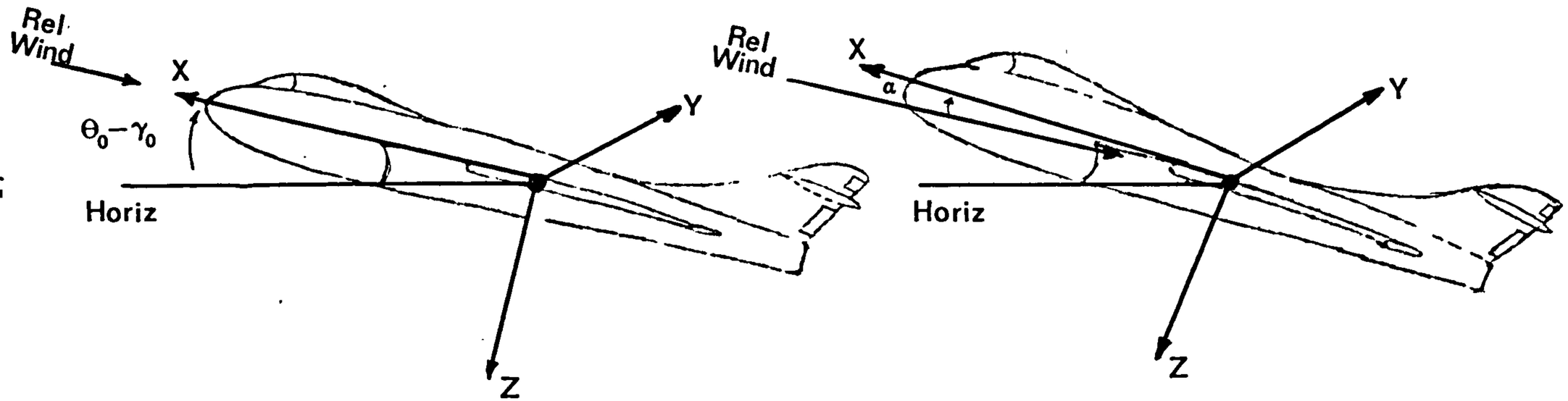


Fig. 2.5 Stability Axes Definition

orientation will change as a function of initial trim conditions but perturbed motions are still measured with respect to the body fixed axes. (Note that the initial incidence angle $\alpha_o = \sin^{-1} W_o/\sqrt{U_o^2 + W_o^2} = 0$

and that $\Theta_o = \gamma_o + \alpha_o = \Gamma_o$).

Equations 2.5 now reduce to

$$p = \dot{\varphi} - \dot{\psi}\sin\Gamma_o = \dot{\varphi} - r \tan \Gamma_o$$

$$q = \dot{\theta} \qquad -2.21$$

$$r = \dot{\psi}\cos \Gamma_o$$

An even further simplification is possible if we take $\Gamma_o = 0$ i.e. the initial velocity vector lies along the horizontal, the aircraft being in straight and level flight. Equations 2.6 then become

$$p = \dot{\varphi}$$

$$q = \dot{\theta} \qquad -2.22$$

$$r = \dot{\psi}$$

It is normal to place the origin of the stability axis system at the vehicle's centre of gravity. Note, however, that the stability axes are not necessarily the axes of symmetry of the vehicle and hence, in general, we must consider the changed moments of inertia and stability derivatives by resolving their body axis values into the stability axes. For relatively small deviations, however, it is valid to assume that this effect is small and accordingly we shall ignore it in the following discussion.

Continuing with the assumption that $\Gamma_o = 0$ and that lower case variables denote perturbed values, the linearised Euler matrix of equation 2.6 can now be reduced to

$$\begin{bmatrix} 1 & -\psi & +\theta \\ +\psi & 1 & -\varphi \\ -\theta & +\varphi & 1 \end{bmatrix} \qquad -2.23$$

Equation 1.1 now gives the velocities in earth axes relative to the stability axes velocities, with $V_o = W_o = 0$, as

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} 1 & -\psi & +\theta \\ +\psi & 1 & -\varphi \\ -\theta & +\varphi & 1 \end{bmatrix} \begin{bmatrix} U_o + u \\ v \\ w \end{bmatrix}$$

and, neglecting products of lower case variables

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} U_o + u \\ U_o + v \\ -U_o\,\theta + w \end{bmatrix} \qquad -2.24$$

In addition the incidence, sideslip and total velocity are given from equations 1.3 and 1.4 as

$$V_T = U_o \qquad -2.25$$

$$\alpha = w/U_o \quad ; \quad \beta = v/U_o$$

neglecting u, since this is small compared with U_o and taking $\sin\alpha = \alpha$, for small α .

Equations 2.4 now reduce to

$$mg\,\theta + m\,\dot{u} = dX$$

$$-mg\,\varphi + m(\dot{v} + U_o\,r) = dY$$

$$m(\dot{w} - U_o\, q) = dZ \qquad -2.26$$

$$\dot{p}\, I_x = dL$$

$$\dot{q}\, I_y = dM$$

$$\dot{r}\, I_z = dN$$

The thrust equations of 2.15 now reduce to

$$X_T = T$$

$$Z_T = -\epsilon_T\, T \qquad -2.27$$

$$M_T = T\, e_T$$

for small ϵ_T and $\alpha_o = 0$.

The vehicle height perturbations are often of interest and since height is the time integral of the downwards velocity in earth axes (downwards positive) then from equation 2.24

$$\dot{h} = W = -U_o\,\theta + w \qquad -2.28$$

We now have an almost complete model of the aircraft in stability axes. All that remains is to expand the aerodynamic forces and moments on the right hand sides of equations 2.26 in terms of the relevant aerodynamic derivatives as indicated in section 2.2. The potentially large number of such terms leads us to consider which of these derivatives are likely to have the largest influence on the aerodynamics under the assumed trim conditions, in this case with $\alpha_o = 0$. This may be done on intuitive grounds, given a knowledge of the likely values of various derivatives for the particular aircraft or by calculation and wind tunnel testing. The derivatives most often considered to be significant are

1) for dX ; X_u, X_w, X_η

2) for dY ; Y_v, Y_r, Y_τ

3) for dZ ; Z_u, Z_w, Z_q, $Z_{\dot{w}}$, $Z_{\dot{q}}$, Z_η

4) for dL ; L_v, L_p, L_r, $L_{\dot{p}}$, $L_{\dot{r}}$, L_ζ, L_τ

5) for dM ; M_u, M_w, M_q, $M_{\dot{w}}$, $M_{\dot{q}}$, M_η

6) for dN ; N_v, N_p, N_r, $N_{\dot{p}}$, $N_{\dot{r}}$, N_ζ, N_τ

where the standard notation is used viz:

$$X_u = \frac{\partial X}{\partial u} \quad ; \quad Y_v = \frac{\partial Y}{\partial v}$$

etc.

More or less of these aerodynamic derivatives may be required for a particular aircraft but we shall consider only the above for the present.

We may now combine equations 2.26 with the above aerodynamic force and moment derivatives to yield

$$\begin{aligned}
m\,\dot{u} &= X_u\,u + X_w\,w + X_\eta\,\eta - mg\,\theta \\
m\,\dot{v} &= Y_v\,v + Y_r\,r + Y_\tau\,\tau + mg\,\varphi - m\,U_o\,r \\
m\,\dot{w} &= Z_u\,u + Z_w\,w + Z_q\,q + Z_{\dot{w}}\,\dot{w} + Z_{\dot{q}}\,\dot{q} + Z_\eta\,\eta + m\,U_o\,q \\
\dot{p}\,I_x &= L_v\,v + L_p\,p + L_r\,r + L_{\dot{p}}\,\dot{p} + L_{\dot{r}}\,\dot{r} + L_\zeta\,\zeta + L_\tau\,\tau \\
\dot{q}\,I_y &= M_u\,u + M_w\,w + M_q\,q + M_{\dot{w}}\,\dot{w} + M_{\dot{q}}\,\dot{q} + M_\eta\,\eta \\
\dot{r}\,I_z &= N_v\,v + N_p\,p + N_r\,r + N_{\dot{p}}\,\dot{p} + N_{\dot{r}}\,\dot{r} + N_\zeta\,\zeta + N_\tau\,\tau
\end{aligned} \qquad -2.29$$

A natural development of the derivation of a differential equation description of the aircraft dynamics is that of finding the most appropriate state-space model representation for subsequent analysis. Whilst the choice of state variables for a given model is arbitrary, on closer inspection of equations 2.22, 2.28 and 2.29 it should be apparent that the most suitable state variables for this system model are :

$$[u, v, w, p, q, r, \theta, \psi, \varphi, h]$$

Remembering that in section 2.2 it was indicated that a natural separation of the dynamics into longitudinal and lateral motions can be made, we may re-order the state vector such that only those state variables affected by longitudinal modes are in the upper portion of the state vector. The state variables affected only by lateral motions then form the lower portion of the state vector, viz:

$$[u,\ w,\ q,\ \theta\ ,\ h\ |\ v,\ p,\ r,\ \varphi\ ,\ \psi\ ]^T$$

The input vector into the system is formed by the control surface deflections and the change in thrust setting from the power plant. This vector may again be separated into longitudinal and lateral input vectors as

$$[\ \eta,\ T\ |\ \tau\ ,\ \xi\ ]^T$$

Note here that any actuator dynamics are ignored for the present discussion but may be included by suitably augmenting the state vector. Power plant dynamics are also ignored. With this state and input vector structure the system state equation may be formulated from equations 2.29, 2.28, 2.27 and 2.22 as

$$
\begin{bmatrix}
m & 0 & 0 & 0 & 0 \\
0 & m-Z_{\dot{w}} & -Z_{\dot{q}} & 0 & 0 \\
0 & -M_{\dot{w}} & I_y-M_{\dot{q}} & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h}
\end{bmatrix}
=
\begin{bmatrix}
X_u & X_w & 0 & -mg & 0 \\
Z_u & Z_w & Z_q+mU_o & 0 & 0 \\
M_u & M_w & M_q & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & U_o & 0
\end{bmatrix}
\begin{bmatrix}
u \\ w \\ q \\ \theta \\ h
\end{bmatrix}
+
\begin{bmatrix}
X_\eta & 1 \\
Z_\eta & \epsilon_T \\
M_\eta & e_T \\
0 & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\eta \\ T
\end{bmatrix}
$$

$$
\begin{bmatrix}
m & 0 & 0 & 0 & 0 \\
0 & I_x-L_{\dot{p}} & -L_{\dot{r}} & 0 & 0 \\
0 & -N_{\dot{p}} & I_z-N_{\dot{r}} & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\varphi} \\ \dot{\psi}
\end{bmatrix}
=
\begin{bmatrix}
Y_u & 0 & Y_r-mU_o & mg & 0 \\
L_v & L_p & L_r & 0 & 0 \\
N_v & N_p & N_r & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
v \\ p \\ r \\ \varphi \\ \psi
\end{bmatrix}
+
\begin{bmatrix}
Y_\tau & 0 \\
L_\tau & L_\zeta \\
N_\tau & N_\zeta \\
0 & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\tau \\ \zeta
\end{bmatrix}
$$

-2.30

This partitioned structure of the system equations has

arisen naturally largely as a function of the many assumptions made in its derivation. Principally, the total decoupling of lateral and longitudinal modes has only been possible by specialising the trim conditions as indicated above. In some circumstances these trim conditions will not be applicable and a more complete set of equations may be required here. The structure of equation 2.30 may, however, be exploited to provide a useful insight into aircraft modal responses. Consider first the longitudinal sub-system of the state equations as follows.

2.2.1 Longitudinal Modes

The states associated with the longitudinal motion of the aircraft are u, w, q, θ and h these forming the upper portion of the state vector in equation 2.30. The equations corresponding to these states are, from equation 2.29, 2.22 and 2.28 :-

$$m\dot{u} = X_u u + X_w + X_\eta \eta - mg\theta \quad \text{a)}$$

$$m\dot{w} = Z_u u + Z_w w + Z_q q + Z_{\dot{q}} \dot{q} + Z_{\dot{w}} \dot{w} + Z_\eta \eta + m U_o q \quad \text{b)}$$

$$\dot{q} I_y = M_u u + M_w w + M_q q + M_{\dot{w}} \dot{w} + M_{\dot{q}} \dot{q} + M_\eta \eta \quad \text{c)}$$

$$\dot{\theta} = q \quad \text{d)}$$

$$\dot{h} = - U_o \theta + w \quad \text{e)}$$

-2.31

Equation 2.31 c) gives for $\dot{q}$

$$(1-m_{\dot{q}})\,\dot{q} = m_u u + m_w w + m_q q + m_{\dot{w}} \dot{w} + m_\eta \eta \qquad -2.32$$

where $m_u = M_u/I_y$; $m_q = M_q/I_y$; etc.

in 2.31 b) this gives

$$\dot{w}\,(1-z_{\dot{w}}) = z_u\,u + z_w\,w + z_q\,q + z_\eta\,\eta + U_o\,q +$$

$$z_{\dot{q}}\;\;\frac{1}{(1-m_{\dot{q}})}\;\;(m_u\,u + m_w\,w + m_q\,q + m_{\dot{w}}\,\dot{w} + m_\eta\,\eta\;)$$

-2.33

where as above $z_u = Z_u/m$; etc.

$$\dot{w}\left\{(1-z_{\dot{w}}) - \frac{z_{\dot{q}}\,m_{\dot{w}}}{(1-m_{\dot{q}})}\right\} = (z_u + \frac{z_{\dot{q}}\,m_u}{(1-m_{\dot{q}})})\;u + (z_w + \frac{z_{\dot{q}}\,m_w}{(1-m_{\dot{q}})})\;w$$

$$+ (z_q + U_o + \frac{z_{\dot{q}}\,m_q}{(1-m_{\dot{q}})})\;q + (\;z_\eta + \frac{z_{\dot{q}}\,m_\eta}{(1-m_{\dot{q}})})\;\eta$$

-2.34

similarly for $\dot{q}$ we obtain

$$\dot{q}\left\{(1-m_{\dot{q}}) - \frac{m_{\dot{w}}\,z_{\dot{q}}}{(1-m_{\dot{w}})}\right\} = (m_u + \frac{m_{\dot{w}}\,z_u}{(1-m_{\dot{w}})})\;u + (m_w + \frac{m_{\dot{w}}\,z_w}{(1-m_{\dot{w}})})\;w$$

$$+ (m_q + \frac{m_{\dot{w}}\,z_q}{(1-m_{\dot{w}})}\,(1+U_o)\;)\;q + (m_\eta + \frac{m_{\dot{w}}\,z_\eta}{(1-m_{\dot{w}})})\;\eta$$

-2.35

It is often the case that $m_{\dot{q}}$, $m_{\dot{w}}$, $z_{\dot{q}}$ and $z_{\dot{w}}$ are small and in this case the r.h.s. coefficients in 2.34 and 2.35 may be approximated by z_u, m_u, z_w, m_w ; etc. whilst the l.h.s. coefficients tend to unity. Pursuing this line of reasoning and letting $x_u = X_u/m$; $x_w = X_w/m$; etc. we may derive a state space model of the longitudinal dynamics from 2.34, 2.35, 2.31 and the thrust equations of 2.27.

$$\begin{bmatrix}\dot{u}\\ \dot{w}\\ \dot{q}\\ \dot{\theta}\\ \dot{h}\end{bmatrix} = \begin{bmatrix} x_u & x_w & 0 & -g & 0\\ z_u & z_w & z_q+U_o & 0 & 0\\ m_u & m_w & m_q & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & -1 & 0 & U_o & 0\end{bmatrix}\begin{bmatrix}u\\ w\\ q\\ \theta\\ h\end{bmatrix} + \begin{bmatrix} x_\eta & 1\\ z_\eta & z_e\\ m_\eta & m_e\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\eta\\ x_e\end{bmatrix}$$

-2.36

Note here that the thrust force T is replaced by the specific thrust $x_e = T/m$ and ϵ_T and e_T in equation 2.30 are replaced by z_e and m_e, these being normalised coefficients determined by including the thrust equation in the derivation of 2.34 and 2.35 and

$$z_e = \left\{ \frac{z_{\dot{q}}\, e_T}{(1-m_{\dot{q}})} + \epsilon_T \right\} \left\{ (1-z_w) - \frac{z_{\dot{q}}\, m_{\dot{w}}}{(1-m_{\dot{q}})} \right\}^{-1}$$

$$m_e = \left\{ \frac{m_{\dot{w}}\, \epsilon_T}{(1-m_{\dot{w}})} + e_T \right\} \left\{ (1-m_q) - \frac{z_{\dot{q}}\, m_{\dot{w}}}{(1-m_{\dot{w}})} \right\}^{-1}$$

The open-loop response of the system of equation 2.36 will be dictated by the eigenvalues of the state 'A' matrix, this being determined by inspection and comparison with the standard description of a linear system in state-space form :

$$\dot{\underline{x}} = A\, \underline{x} + B\, \underline{u}$$

The characteristic equation for the open-loop system is given by $\det(sI - A) = 0$, where s is the Laplace variable. The values of s which satisfy the characteristic equation are the eigenvalues of the A matrix. For most aircraft the longitudinal fifth order characteristic equation of 2.36 has the form

$$s(s^2 + 2\,\zeta_p\,\omega_{np}\,s + \omega_{np}^2)(s^2 + 2\,\zeta_s\,\omega_{ns}\,s + \omega_{ns}^2) = 0 \qquad \text{-2.36 a)}$$

i.e. a single pole (root) at s=0 and two complex pole pairs. The complex poles normally give rise to stable oscillatory responses but having significantly different response times. One pole pair is generally termed the phugoid mode, suffix p, the other being the short period, or quick, mode, suffix s. The single pole at s = 0 corresponds to a pure integration and arises due to the height integration action.

It is instructive to consider the approximate orders of the two oscilatory modes. To do this consider the zero input form of equation 2.36 i.e. with $\eta = x_e = 0$. We first note that the height equation, h, does not couple into any of the remaining equations. The last row and column of the A matrix may thus be removed leaving

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_u & x_w & 0 & -g \\ z_u & z_w & z_q+U_o & 0 \\ m_u & m_w & m_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \qquad -2.37$$

Short period aircraft motions always take place rapidly when compared with changes in forward speed, u. These motions may thus be considered to take place with constant speed giving $\dot{u} = 0$. The first row and column of 2.37 may thus be removed leaving

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} z_w & z_q+U_o & 0 \\ m_w & m_q & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \end{bmatrix} \qquad -2.38$$

The last row and column of 2.38 may now be deleted since θ does not couple into the w and q equations.

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} z_w & z_q+U_o \\ m_w & m_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} \qquad -2.39$$

The characteristic equation associated with 2.39 is thus

$$s^2 - (z_w + m_q)\, s + (z_w m_q - m_w z_q - m_w U_o) = 0$$

This equation is normally found to be close to the

$$(s^2 + 2\,\zeta_s\,\omega_{ns}\,s + \omega_{ns}^2)$$

factor in 2.36 a). Table 2.1 details the values of the longitudinal and lateral modes for the Machan r.p.v., evaluated using the relevent aerodynamic derivatives for an airspeed of 33 ms^{-1} and zero incidence, according to the equations derived above and in the following sections. From this table the Machan's short period mode has a natural frequency, ω_{ns}, of 3.5 rad s^{-1} and damping, ζ_s, of 0.44. These values are fairly typical of short period aircraft motions which are reasonably rapid and relatively well damped.

Returning to equation 2.37, the assumption required for the phugoid approximation is that the w and q dynamics are sufficiently fast for their respective differential equations to be instantaneously satisfied, i.e. $\dot{w} = \dot{q} = 0$. Under these conditions the 2nd. and 3rd. rows of 2.37 give

$$0 = z_u\,u + z_w\,w + (z_q + U_o)\,q$$

$$0 = m_u\,u + m_w\,w + m_q\,q$$

solving for q and w in terms of u gives

$$q = \frac{(z_u\,m_w - z_w\,m_u)\,u}{(m_q z_w - (z_q + U_o)m_w} \quad ; \quad w = -\frac{(z_u\,m_q - m_u(z_q + U_o))\,u}{(m_q z_w - (z_q + U_o)m_w)}$$

substituting these values into equation 2.37 yields

$$\begin{bmatrix} \dot{u} \\ \\ \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_u - \dfrac{x_w(z_u m_q - m_u(z_q + U_o))}{(m_q z_w - m_w(z_q + U_o))} & -g \\ \\ \dfrac{-(z_u m_w - m_u z_w)}{(m_q z_w - m_w(z_q + U_o))} & 0 \end{bmatrix} \begin{bmatrix} u \\ \\ \\ \theta \end{bmatrix} \qquad -2.40$$

Table 2.1

Machan Dynamics Evaluated at 33 ms^{-1} and Zero Incidence

Longitudinal		
Short period	-1.57 + 3.16j	(ζ = 0.44, ω_n = 3.52)
Phugoid	-0.116 + 0.73j	(ζ = 0.16, ω_n = 0.74)
Lateral		
Roll subsidence	-8.524	(τ = 0.1173 s)
Spiral mode	0.1279	(τ = 7.81 s)
Dutch roll	-0.2984 + 3.453j	(ζ = 0.086, ω_n = 3.46)

(frequencies in rad s^{-1})

The characteristic equation associated with 2.40 is now given by

$$s^2 + \left[- x_u + \frac{x_w \ (z_u m_q - m_u (z_q + U_o))}{(m_q z_w - m_w (z_q + U_o))} \right] s$$

$$- g \ \frac{\{z_u m_w - m_u z_w\}}{m_q z_w - m_w (z_q + U_o)} \qquad = 0$$

-2.40 a)

It is often found that equation 2.40 a) is a reasonable approximation to the phugoid term in 2.36 a) i.e.

$$(s^2 + 2 \zeta_p \ \omega_{np} \ s + \omega_{np}^2)$$

Again referring to Table 2.1, the Machan's phugoid mode is clearly lightly damped compared with the short period mode but of a considerably longer period, $\zeta_p = 0.16$, $\omega_{np} = 0.74$ rad s^{-1}. The phugoid mode is normally so slow that the pilot is well able to correct this long term oscillation in height/pitch attitude.

The longitudinal modes have the following simple physical interpretation :

i) The height integration mode is a natural consequence of the geometry of the aircraft. If the vertical velocity has a finite value then the aircraft's height will increase or decrease indefinitely since there is no feedback of height into the vertical velocity.

ii) The short period mode is a result of the "arrow stability" of the aircraft. This is due to the effects of the offset of the aircraft's centre of gravity from the aerodynamic centre of pressure.

iii) The phugoid mode has a quite subtle interpretation. Its fundamental origin is due to the exchange of kinetic and potential energies of the airframe. When flying in straight and level flight a lift force is generated primarily due to the airflow over the wings.

This lift force increases the aircraft's upwards velocity and consequently its height at the expense of the forward velocity. This reduces the airflow over the wings reducing lift and upwards velocity and hence reducing height. This loss of height, potential energy, is converted into kinetic energy which acts so as to increase the forward velocity of the aircraft, increasing the airflow over the wings and hence increasing lift and height, the process then repeating with a relatively long time constant and relatively light damping. The aircraft thus exhibits sinusoidal variations in height when flying straight and level, its pitch attitude clearly also varying in a sinusoidal manner.

2.2.2 Lateral Modes

The equations concerned with the lateral dynamics are those associated with the lower portion of the state vector of equations 2.30 namely the equations in v, p, r, φ and ψ. These are:

$$m\,\dot{v} = Y_v\,v + Y_r\,r + Y_\tau\,\tau + mg\,\varphi - m\,U_o\,r \qquad a)$$

$$\dot{p}\,I_x = L_v\,v + L_p\,p + L_r\,r + L_{\dot{p}}\,\dot{p} + L_{\dot{r}}\,\dot{r} + L_\tau\,\tau + L_\xi\,\xi \qquad b)$$

$$\dot{r}\,I_z = N_v\,v + N_p\,p + N_r\,r + N_{\dot{p}}\,\dot{p} + N_{\dot{r}}\,\dot{r} + N_\tau\,\tau + N_\xi\,\xi \qquad c)$$

$$\dot{\varphi} = p \qquad d)$$

$$\dot{\psi} = r \qquad e)$$

-2.41

We may treat 2.41 b) and c) as was done for the longitudinal modes to obtain

$$\dot{p}\left\{(1-l_p) - \frac{l_{\dot{r}}\, n_{\dot{p}}}{(1-n_{\dot{r}})}\right\} = l_v + \frac{l_{\dot{r}}\, n_v}{(1-n_{\dot{r}})}\, v + \left(l_p + \frac{l_{\dot{r}}\, n_p}{(1-n_{\dot{r}})}\right) p +$$

$$\left(l_r + \frac{l_{\dot{r}}\, n_r}{(1-n_{\dot{r}})}\right) r + \left(l_\tau + \frac{l_{\dot{r}}\, n_\tau}{(1-n_{\dot{r}})}\right)\tau + \left(l_\zeta + \frac{l_{\dot{r}}\, n_\zeta}{(1-n_{\dot{r}})}\right)\zeta$$

$$\dot{r}\left\{(1-n_{\dot{r}}) - \frac{n_{\dot{p}}\, l_{\dot{r}}}{(1-l_{\dot{p}})}\right\} = \left(n_v + \frac{n_{\dot{p}}\, l_v}{(1-l_{\dot{p}})}\right) v + \left(n_p + \frac{n_{\dot{p}}\, l_p}{(1-l_{\dot{p}})}\right) p +$$

$$\left(n_r + \frac{n_{\dot{p}}\, l_v}{(1-l_{\dot{p}})}\right) r + \left(n_\tau + \frac{n_{\dot{p}}\, l_\tau}{(1-l_{\dot{p}})}\right)\tau + \left(n_\zeta + \frac{n_{\dot{p}}\, l_\zeta}{(1-l_{\dot{p}})}\right)\zeta$$

-2.42

By employing a similar technique to that used in section 2.2.1 we may derive the state equation for the lateral motions as :

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\varphi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} y_v & 0 & y_r - U_o & g & 0 \\ l_v & l_p & l_r & 0 & 0 \\ n_v & n_p & n_r & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \varphi \\ \psi \end{bmatrix} + \begin{bmatrix} y_\tau & 0 \\ l_\tau & l_\zeta \\ n_\tau & n_\zeta \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tau \\ \zeta \end{bmatrix}$$

-2.43

Again proceeding as in section 2.2.1 we may derive the fifth order characteristic polynomial associated with the system of equation 2.43 as det (sI - A) = 0. For most aircraft this characteristic polynomial is factorisable into a number of modes as follows :-

$$s(s + 1/\tau_r)(s^2 + 2\,\zeta_d\,\omega_d\,s + \omega_d^2)(s + 1/\tau_s) = 0$$

We note here the presence of three open loop real poles (roots) at s=0, s=$1/\tau_r$ and s=$-1/\tau_s$. These modes are termed

heading integration, roll subsidence and spiral divergence, respectively. The final quadratic factor yields two complex poles giving rise to an oscillatory response mode termed 'Dutch Roll'.

Approximate expressions for the factors of equation 2.43 a) may be obtained as follows. Consider the zero input form of 2.43 and delete the last row and column of the 'A' matrix since the heading integration, ψ , mode does not couple into any other state. We thus obtain, for the impulse response of the lateral motions :

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} y_v & 0 & y_r - U_o & g \\ l_v & l_p & l_r & 0 \\ n_v & n_p & n_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \varphi \end{bmatrix} \qquad \text{-2.43 b)}$$

For the majority of aircraft the rolling mode decays rapidly compared with the spiral and dutch roll modes. This being the case we may assume that the states v, r, and φ are zero for short term rolling motion and hence the roll subsidence equation is approximated by

$$\dot{p} = l_p \, p$$

i.e. a first order mode of time constant $-1/l_p$ seconds. This is normally a good approximation to the roll subsidence mote of equation 2.43 a). Note that l_p is normally negative, the roll subsidence mode being stable in this case.

If the roll mode is considered to be rapid then the differential equation in $\dot{p}$ may be considered to be instantaneously satisfied giving

$$\begin{bmatrix} \dot{v} \\ 0 \\ \dot{r} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} y_v & 0 & y_r - U_o & g \\ l_v & l_p & l_r & 0 \\ n_v & n_p & n_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \varphi \end{bmatrix} \qquad -2.44$$

Solving row 2 of 2.44 algebraically gives

$$p = -\frac{1}{l_p} (l_v v + l_r r) \qquad -2.44\ a)$$

and neglecting gravitational effects the φ state will have no effect on any ensuing motion thus we obtain.

$$\begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} y_v & y_r - U_o \\ n_v - \frac{n_p l_v}{l_p} & n_r - \frac{n_p l_r}{l_p} \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} \qquad -2.45$$

Equation 2.45 has a characteristic polynomial

$$s^2 + (- y_v - n_r + \frac{n_p l_r}{l_p}) s + (y_v n_r - n_v (y_r - U_o)$$

$$- y_v \frac{n_p l_r}{l_p} - \frac{n_p l_v}{l_p} (y_r - U_o)) = 0 \qquad -2.46$$

It can be shown that equation 2.46 is a good approximation to the dutch roll mode of equation 2.43 a). Dutch roll has a similar origin to the short period dynamic of the longitudinal equations, however it is accompanied by a rolling motion owing to the coupling into roll from equation 2.44 a).

The final mode, the spiral divergence, is a slow mode

and we may thus consider the differential equations in p, v and r to be instantaneously satisfied giving $\dot{p} = \dot{v} = \dot{r} = 0$ and

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} Y_v & 0 & Y_r - U_o & g \\ l_v & l_p & l_r & 0 \\ n_v & n_p & n_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \varphi \end{bmatrix} \qquad -2.47$$

Solving this equation algebraically yeilds, for φ ,

$$\dot{\varphi} = g\varphi \left\{ \frac{Y_v(n_r l_p - l_r n_p)}{(n_r l_v - l_r n_v)} + \frac{(Y_r - U_o)(n_v l_p - l_v n_p)}{(n_v l_r - l_v n_r)} \right\}^{-1} \qquad -2.48$$

For most flight conditions the $(Y_r - U_o)$ term in 2.48 may be considered to be much greater than the Y_v term and hence

$$\dot{\varphi} \approx g\varphi \frac{(n_v l_r - l_v n_r)}{(Y_r - U_o)(n_v l_p - l_v n_p)}$$

This is often a reasonable representation of the $(s + 1/\tau_s)$ factor of equation 2.43 a) and clearly

$$\tau_s = \frac{(Y_r - U_o)(n_v l_p - l_v n_p)}{g\,(l_r n_v - l_v n_r)}$$

Note that τ_s will be large and may for some aircraft be negative leading to a divergent mode. For the Machan the lateral modes, evaluated as in the above expressions, are given in Table 2.1. Note that the spiral mode is unstable with a relatively long time constant of 7.8 secs. The Dutch

Roll is lightly damped, ζ_d = 0.086, and of reasonably short period, ω_d = 3.45 rad s^{-1}, and the roll subsidence has a time constant of 0.117 s. The divergent spiral mode is often tolerated since it is easily controlled by the pilot, it being extremly slow and a stable spiral mode may often only be attained at the expense of a less well damped Dutch Roll mode.

The physical interpretation of the lateral motion modes may be stated as :

i) As with the height integration mode the heading integration mode is the result of the aircraft's geometry, no local feedback of heading angle into side velocity taking place.

ii) The roll subsidence mode is the response of the aircraft due to aileron deflection and decays rapidly owing to the damping action of the wings.

iii) The spiral divergence mode arises due to the effect of gravity on the aircraft. In forcing the aircraft to roll gravity will induce both sideslip and yaw. For a divergent spiral mode this yaw and sideslip will further increase the roll angle, φ, leading to a further increase in yaw angle and sideslip velocity. The ensuing motion, if uncontrolled, results in ever increasing yaw and roll, the aircraft then entering a high speed spiral dive. The mode is, however, so slow that it is easily controlled by the pilot.

iv) The final mode, the Dutch Roll, arises from much the same source as the short period pitching mode in the longitudinal motion. Dutch Roll consists of a combination of roll, yaw and sideslip about some equilibrium position. The aircraft first rolls from the equilibrium inducing yaw and sideslip. This sideslip then induces a righting rolling moment which forces the aircraft back through the original equilibrium position leading to yaw and sideslip in the opposite sense. The resulting oscillatory response may be stable and die away relatively quickly. In fact, as

the Dutch Roll mode gives rise to such a physically unpleasant sensation it is important, particularly for passenger aircraft, to design the airframe carefully to provide adequate Dutch Roll damping. An alternative approach is to use a stability augmentation system (S.A.S) in the form of a control system autopilot. Both of these approaches can improve Dutch Roll but often at the expense of reduced spiral mode stability.

2.3 An Alternative Linearisation Technique

By way of a contrast to the above 'classical' aerodynamic decomposition of the response modes of the aircraft the following section presents a brief description of a possible alternative approach. This is based on the evaluation of the first partial derivatives for each of the state variables, using the non-linear equations of Chapter 1, and forming these into the appropriate (first order) state A and B matrices for the system. These expressions retain the functional dependence of the elements of the A and B matrices on the set of state variables, $\underline{x}$. The A and B matrices may then be locally linearised about the actual state trajectory of the non-linear aircraft model. Consider first the longitudinal dynamics.

2.3.1 Longitudinal Dynamics

Table 2.2 is a list of the elements of the state space matrices, A and B, written in functional form and for the longitudinal dynamics of the model. The subscript notation on the left-hand side has been used to clarify the significance of each matrix element without recourse to a standard numbering system.

It should be noted that by retaining the functional dependence of these parameters on the set of state variables $\underline{x}$, the locally linearised A and B matrices corresponding to

Table 2.2

Table of Locally Linearised Parameters for Machan (R.P.V.) model (Longitudinal Dynamics)

A, B Matrix parameters		First Variational Expressions
1)	$A(\dot{u},u)$	$-\frac{\rho S V_T}{m}(C_{Do} + kC_L^2)$
2)	$A(\dot{u},w)$	$\frac{\rho S\ V_T\ C_L}{2\ m}$
3)	$A(\dot{u},q)$	$-w\left\{1 + \rho\frac{S\ l_T\ Z_q}{m}\right\}$ $\rightarrow 0$ as $w \rightarrow 0$
4)	$A(\dot{u},\theta)$	$-g\cos\theta$ $\rightarrow -g$ as $\theta \rightarrow 0$
5)	$A(\dot{u},h)$	0
6)	$A(\dot{u},X_e)$	$1/m$
7)	$A(\dot{w},u)$	$-\frac{\rho S\ V_T}{m}\left\{C_L + \frac{\bar{c}}{l_T}(C_{MT} - C_{MWBD})\right\}$
8)	$A(\dot{w},w)$	$-\frac{\rho S\ V_T}{2m}\{a_1 + C_{Do}\}$
9)	$A(\dot{w},q)$	$V_T\left\{1 + \rho\frac{S\ l_T\ Z_q}{m}\right\}$
10)	$A(\dot{w},\theta)$	$-g\sin\varphi\cos\varphi \rightarrow 0$ as $\varphi \rightarrow 0$
11)	$A(\dot{w},h)$	0
12)	$A(\dot{w},X_e)$	0
13)	$A(\dot{q},u)$	$\rho\frac{S\ V_T\ \bar{c}}{I_y}[C_{M\eta}\eta_0 + C_{MW} + C_L(cg - 0.25)]$ (where η_o = trim elevator angle, typically 0.11 rads.)

14) $A(\dot{q},w)$

$$\frac{\rho S}{2I_y}\Big\{C_{ML}a_1\bar{c}V_T + \bar{c}(cg - 0.25)\{2wC_{Lo} + a_1V_T\}$$

$$+ 2w\bar{c}(C_{Mo} + C_{ML}C_{Lo})$$

$$+ \frac{2\,wl_T\,Z_q\,q}{V_T}\{l_T + \bar{c}(cg - 0.25)\}\Big\}$$

$$\rightarrow \rho\frac{S\,a_1\bar{c}V_T}{2I_y}\{C_{ML} + cg - 0.25\}$$

as $w \rightarrow 0$

15) $A(\dot{q},q)$

$$\rho\frac{Sl_T\,Z_qV_T}{I_y}(l_T + \bar{c}(0.25 - cg))$$

16) $A(\dot{q},\theta)$, $A(\dot{q},h)$, $A(\dot{q},X_e)$ — 0

17) $A(\dot{\theta},u)$, $A(\dot{\theta},w)$, $A(\dot{\theta},h)$, $A(\dot{\theta},X_e)$, $A(\dot{\theta},\theta)$ — 0

18) $A(\dot{\theta},q)$ — $\cos\varphi \rightarrow 1$, as $\varphi \rightarrow 0$

19) $A(\dot{h},u)$ — $\sin\theta \simeq \theta$, for small θ

20) $A(\dot{h},w)$ — $\cos\varphi\cos\theta \rightarrow 1$, for φ, θ small (sign convention for integrated downwards velocity)

21) $A(\dot{h},q)$, $A(\dot{h},h)$, $A(\dot{h},X_e)$ — $\rightarrow 0$ for small u, v, w

22) $A(h,\theta)$

$$u\cos\theta + v\sin\varphi\sin\theta - w\cos\varphi\sin\theta$$

$$\rightarrow u - \theta w \quad \text{for small } \varphi,\theta$$

23) $A(\dot{X}_e,X_e)$

$$\frac{-2X_e\rho A_D(M^* - u) + X_e^2/M^*}{\rho^2 A_D^2(M^* - u)^2 I_xK_e}$$

$$\text{where } M^* = \sqrt{\frac{2X_e + \rho A_Du^2}{\rho A_D}}$$

and X_e, u are the operating point variables

24) $A(\dot{X}_e, u)$ $\quad \dfrac{X_e{}^2(1 + u/M^*)}{I_x \rho A_D K_e (M^* - u)^2}$

25) $B(\dot{w}, \eta)$ $\quad -\dfrac{C_{M\eta}\, \rho\, S V_T{}^2 \bar{c}}{2\, m\, l_T} = \dfrac{Z_\eta V_T{}^2 \rho\, S}{m}$

26) $B(\dot{q}, \eta)$ $\quad \dfrac{1}{I_y}\left(\dfrac{\partial M_a}{\partial \eta}\right) = \dfrac{\rho\, V_T{}^2\, S\bar{c}\, C_{M\eta}}{2\, I_y}$

27) $B(\dot{X}_e, T_H)$ $\quad \dfrac{P_{max}\, \eta_p}{K_e I_x}$

Note:- $V_T = \sqrt{u^2 + v^2 + w^2}$

a linear time-varying system can be generated ; the linearisation being computed for the state vector at a given step of the simulated response. It follows also that the computation of constant A and B matrices corresponding to the given trim flight can be achieved in this manner. The elements of the A and B matrices then relate directly to the unnormalised aerodynamic derivatives, e.g. :-

$$A(u,u) = \frac{1}{m}\frac{\partial X}{\partial u} = -\frac{\rho S V_T (C_{DO} + k C_L^2)}{m}$$

$$= x_u$$

$$B(w,\eta) = \frac{1}{m}\frac{\partial Z}{\partial \eta} = \frac{Z_\eta \ V_T^2 \ \rho \ S}{m} = z_\eta$$

etc...., to first approximation.

An example of the longitudinal motion state-space matrices for an elevator trim setting of $\eta = \eta_o = 0.11$ rads. and for an airspeed of $V_T = U = 33\ ms^{-1}$ is :-

$$\dot{\underline{x}} = A_{lo}\,\underline{x} + B_{lo}\,\underline{u}$$

where $\underline{x} = x_i$, $i = 1, 2, 3, \ldots\ldots, 6$; $\underline{u} = u_j$, $j = 1, 2$

$$A_{lo} = \begin{bmatrix} -0.059 & 0.147 & 0. & -9.81 & 0. & 0.0125 \\ -0.475 & -2.93 & 32.77 & 0. & 0. & 0. \\ 0.166 & -0.416 & -0.645 & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. & 0. \\ 0. & -1. & 0. & 33. & 0. & 0. \\ -19.74 & 0. & 0. & 0. & 0. & -2.275 \end{bmatrix} ;$$

and

$$B_{lo} = \begin{bmatrix} 0. & 0. \\ 5.318 & 0. \\ -13.58 & 0. \\ 0. & 0. \\ 0. & 0. \\ 0. & 1600. \end{bmatrix}$$

The open-loop poles are -1.811 ± j 3.47, -2.17, -0.0596 ± j 0.675, 0. and the ordering of the state and input variables is as follows

$u \rightarrow x_1$ i.e. A(u,u) = A(1,1)
$w \rightarrow x_2$ A(w,u) = A(2,1), etc..
$q \rightarrow x_3$
$\theta \rightarrow x_4$
$h \rightarrow x_5$
$x_e \rightarrow x_6$
$\eta \rightarrow u_1$
$T_H \rightarrow u_2$

The above pole values should be compared with the approximations derived in section 2.2 and listed in Table 2.1. Note that the thruster pole at -2.17 is not modelled in section 2.2 and thus is not included in Table 2.1.

2.3.2 Lateral Dynamics

Table 2.3 is a list of the elements of the A and B matrices of the lateral motion dynamics using the notation of section 2.3.1.

An example of these lateral motion state-space matrices for a rudder and aileron trim setting of $\tau = 0$, $\zeta = 0$, and for V_T=33 ms^{-1} is :-

$$\dot{\underline{x}} = A_{la}\,\underline{x} + B_{la}\,\underline{u}$$

where $\underline{x} = x_i$, i = 7, 8, 9,....,11 ; $\underline{u} = u_j$, j = 3, 4

$$A_{la} = \begin{bmatrix} -0.277 & 0. & -32.9 & 9.81 & 0. \\ -0.1033 & -8.525 & 3.75 & 0. & 0. \\ 0.3649 & 0. & -0.639 & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \end{bmatrix}, \text{ and}$$

Table 2.3

Table of Locally Linearised Parameters for Machan (R.P.V.) Model (Lateral Dynamics)

	A, B Matrix parameters	First Variational Expressions
1)	$A(\dot{v},v)$	$\frac{\rho S}{V_T m}(Y_v V_T^2 + v^2 Y_v + (b/2)\ Y_r rv + 2Y_\tau \tau V_T^2)$ $\rightarrow \frac{\rho S\ Y_v V_T}{m}$, as $v,r,\tau \rightarrow 0$
2)	$A(\dot{v},p)$	$w/m \simeq 0$
3)	$A(\dot{v},r)$	$-u + \frac{\rho S\ b\ Y_r\ V_T}{2m}$
4)	$A(\dot{v},\varphi)$	$g\cos\theta\cos\varphi \rightarrow g$, as θ, φ smal
5)	$A(\dot{v},\psi)$	0
6)	$A(\dot{p},v)$	$\frac{\rho bS}{2V_T I_x}\left\{\frac{v}{2}(L_p p + L_r C_L r) + L_v V_T^2 + L_v v^2 + 2L_\zeta \zeta v V_T\right\}$ $\rightarrow \frac{\rho bSV_T L_v}{2I_x}$, as $v,w \rightarrow 0$
7)	$A(\dot{p},p)$	$\rightarrow \frac{\rho b^2 S L_p V_T}{4I_x}$
8)	$A(\dot{p},r)$	$\frac{\rho\ b^2 S V_T L_r C_L}{4I_x} + \frac{(I_x - I_z)\ q}{I_x}$ $\rightarrow \frac{\rho\ b^2 S V_T L_r C_L}{4I_x}$, as $q \rightarrow 0$
9)	$A(\dot{p},\varphi)$, $A(\dot{p},\psi)$	0
10)	$A(\dot{r},v)$	$\frac{\rho SbV_T N_v}{2I_z}$

11)	$A(\dot{r},p)$	$\frac{q(I_z - I_x)}{I_z} \rightarrow 0$, as $q \rightarrow 0$
12)	$A(\dot{r},r)$	$\rho \frac{N_r \, SV_T b^2}{4I_z}$ (where $N_r = -0.046 - 0.015C_L^2$ and $C_L = C_{Lo} + a_1\alpha$)
13)	$A(\dot{r},\varphi)$, $A(\dot{r},\psi)$	0
14)	$A(\dot{\varphi},p)$	1
15)	$A(\dot{\varphi},r)$	$\cos\varphi\tan\theta$, $\rightarrow 0$ as $\theta \rightarrow 0$
16)	$A(\dot{\varphi},v)$, $A(\dot{\varphi},\psi)$	0
17)	$A(\dot{\varphi},\varphi)$	$q\cos\varphi\tan\theta - r\tan\theta\sin\varphi$
18)	$A(\dot{\psi},r)$	$\frac{\cos\varphi}{\cos\theta} \rightarrow 1$
19)	$A(\dot{\psi},v)$, $A(\dot{\psi},p)$	0
20)	$A(\dot{\psi},\varphi)$	$q\frac{\cos\varphi}{\cos\theta} - r\frac{\sin\varphi}{\cos\theta}$
21)	$B(\dot{v},\tau)$	$\rho \frac{SV_T^2 Y_\tau}{m}$
22)	$B(\dot{p},\xi)$	$\rho \frac{V_T^2 bSL_\xi}{2I_x}$
23)	$B(\dot{r},\tau)$	$\rho \frac{S \, V_T^2 bN_\tau}{2I_z}$

Note:- $V_T = \sqrt{u^2 + v^2 + w^2}$

$$B_{la} = \begin{bmatrix} -5.432 & 0. \\ 0. & -28.64 \\ -9.49 & 0. \\ 0. & 0. \\ 0. & 0. \end{bmatrix}$$

The open loop poles are $-0.4983 \pm j\ 3.5$, -8.5579, 0.0, 0.119 and the ordering of the state and input variables is as follows :-

v	$\rightarrow$	x_7	i.e. A(v,v) = A(7,7)
p	$\rightarrow$	x_8	A(p,v) = A(8,7), etc.
r	$\rightarrow$	x_9	
φ	$\rightarrow$	x_{10}	
ψ	$\rightarrow$	x_{11}	
τ	$\rightarrow$	u_3	
ξ	$\rightarrow$	u_4	

(Note :- for Machan rudder actuation couples into yaw only whilst aileron actuation couples into roll only. The B matrix thus has zero's as the B(p,τ) and B(r,ξ) elements, c.f. equation 2.43.)

2.3.3 Cross-Coupling Dynamics

Table 2.4 gives a list of the state space elements providing the functional cross-coupling between longitudinal and lateral motions of the aircraft model. It should be noted that, whilst it is usual to neglect the effect of these terms for small perturbations about trim flight conditions, for large incidence flight or manoeuvering motions some of the cross-coupling dynamics terms become significant. This information is to be used to predict the range of parameter changes expected from an actual aircraft in flight. The study will be useful in assessing the robustness of autopilot designs and also in determining the validity of some analytical redundancy techniques for sensor failure detection. The cross-coupling terms are also likely to be used in an improved numerical integration method for

Table 2.4

Table of Locally Linearised Parameters for Machan (R.P.V) model (Cross-coupling Dynamics)

A, B matrix parameters		First Variational Expressions
1)	$A(\dot{u},v)$	r
2)	$A(\dot{u},r)$	v
3)	$A(\dot{w},v)$	$-p$
4)	$A(\dot{w},\varphi)$	$-g\cos\theta\sin\varphi$
5)	$A(\dot{q},v)$	$\frac{1}{I_y}\left(\frac{\partial M_a}{\partial v}\right) = \frac{1}{I_y}\{v\rho Sc(C_{MW} + C_{M\xi}\xi)\}$
6)	$A(\dot{q},p)$	$\frac{r}{I_y}(I_z - I_x)$
7)	$A(\dot{q},r)$	$\frac{p}{I_y}(I_z - I_x)$
8)	$A(\dot{\theta},r)$	$-\sin\varphi$
9)	$A(\dot{\theta},\varphi)$	$-g\sin\varphi - r\cos\varphi$
10)	$A(\dot{v},u)$	$-r + \rho\frac{SV_T}{m}\left\{\frac{Y_v v + (b/2)Y_r r}{V_T} + 2Y_\tau\tau\right\}$
11)	$A(\dot{v},w)$	$p + \frac{S\rho}{V_T m}\{Y_r vw + (b/2)vY_r r + 2wY_\tau\tau\ V_T\}$
12)	$A(\dot{v},\theta)$	$-g\sin\theta\sin\varphi$
13)	$A(\dot{p},u)$	$\frac{1}{m}\left(\frac{\partial L_e}{\partial u}\right) \to L_e$ as $L_e \to 0$
14)	$A(\dot{p},w)$	$\frac{\rho bS}{2I_x V_T}\left\{\frac{bw}{2}(L_p p + L_r C_L r) + L_v vw + 2L_\xi\xi\ wV_T\right\}$ $\to 0$, as $v = w = 0$
15)	$A(\dot{\varphi},q)$	$\sin\varphi\tan\theta$

16) $A(\dot{\psi}, q)$ $\dfrac{\sin\varphi}{\cos\theta}$

17) $A(\dot{\psi}, \theta)$ $\tan\theta\sec\theta(q\sin\varphi + r\cos\varphi)$

solving the non-linear model equations.

The A matrix elements of Table 2.4 give rise to the cross-coupling matrices A_1, A_2 of the partitioned state-space equations as follows :-

$$\dot{\underline{x}} = \left[\begin{array}{c|c} A_{1o} & A_1 \\ \hline A_2 & A_{1a} \end{array}\right] \underline{x} + \left[\begin{array}{c} B_{1o} \\ \hline B_{1a} \end{array}\right] \underline{u}$$

2.4 Summary

In this chapter the linearised equations of motion for the aircraft have been developed from the basic non-linear six degrees of freedom equations outlined in chapter 1. Under the very general trim conditions of a typical aircraft the resulting linearised equations are formidably complex as indicated in section 2.1. To derive an analytically tractable set of linear equations it proved necessary to specialise the trim conditions of the aircraft and to limit the analyses to small perturbations about these 'steady-state' conditions. Of particular importance is the separation of the aircraft motions into longitudinal and lateral modes, this being possible by specialising the aircraft trim. A further simplification was made by suitably aligning the initial body axes of the aircraft to straight and level flight. The resulting stability axis model is sufficiently simplified to allow a state space model of the aircraft to be derived. This state space model may then be decomposed to provide approximations to the dominant response modes in both lateral and longitudinal motions. At each stage of the derivation it was necessary to make a number of assumptions and this necessarily implies that the linear state space model is only valid under a very specialised set of aircraft flight configurations (i.e. for a given airspeed and for small perturbations about trimmed flight). For relatively short term control analysis, however, this type of analysis is valid and has been shown to provide adequate performance over a fairly wide range of

aircraft manoeuvres. It is also worth noting that the analyses presented are by no means the only methods available but nevertheless provide a useful insight into aircraft response and the potential control problem. Interested readers are referred to the standard texts (1-5,9,10) for further information on aircraft modelling.

In the following Chapter a discussion is presented of the requirements for simulation of the non-linear equations of motion of the aircraft. The problems of control system design are treated in Chapters 5, 6 and 7.

CHAPTER 3

Implementation

In the previous chapters we have developed a set of equations which are considered to be a sufficiently detailed model of an aircraft to allow them to be used for simulation studies. Perhaps, at this point, it would be useful to recap the principal requirements of an aircraft simulator for use in control system studies in a University Laboratory, as mentioned in the introduction.

i) An important aspect of the aircraft system is the high degree of interaction between the pilot and the system, the pilot often forming an integral part of the control loop. It was felt important that any simulation study should attempt to emulate this highly interactive environment in order to allow pilots to assess the handling qualities of the aircraft and any control law designs implemented. This level of interaction would also allow the investigators to gain a 'feel' for the problem. One implication here is that a close-to real time simulation is required with pilotic 'joystick' interaction. This is a severe constraint not only on the type of simulation used but also on the machine used to run the simulation.

ii) In order to emulate closely the type of environment found in a typical aircraft it was felt desirable to allow the simulation to 'stand-alone' on a dedicated machine whilst any control laws were implemented on a second dedicated machine emulating, say, a flight control computer and passing control signals to and from the simulation on a standard data link. This allows, for example, sensor noise or sensor failures to be emulated in addition to possibly allowing controller parameters to be changed 'on-line'.

iii) Since a large number of variables are available within the simulation it is important to have available adequate display facilities which provide displays of important aircraft parameters in 'real time'. One possibility is to use a 'head-up' type of display typical of those used in a cockpit. Again this aspect severely restricts the choice of simulation machine.

With the above in mind a brief specification of the 'ideal' simulation machine could be drawn up. It was, however, initially important to decide between an analogue or digital simulation. Basically, an analogue simulation was rejected on the grounds that an analogue or hybrid computer was not immediately available and that the number and non-linear nature of the aerodynamic equations would lead to unwieldy patching and scaling and probably require a reasonably powerful analogue machine. An analogue simulation would, however, have advantages for the real time aspects. Limiting the discussion then to purely digital computers our 'ideal' machine can be envisaged as follows :

A single user computer capable of being programmed in a high level language and compiling and running the resulting code under some form of executive in a realistic real time environment. The machine would also require high speed input/output facilities accessible from the high level language in order to interact with say a joystick type of pilotic input in addition to communicating with a second controller computer. The machine should also have high resolution fast access graphics in order to provide for say a headup type of operator display. A reasonable amount of main memory would also be required considering the likely size of code and backup high density file storage facilities for data and program storage. The machine must clearly also be reasonably 'fast' to handle the necessary 'number crunching' in something close to real time.

The above specification would probably be most easily met using a dedicated minicomputer such as a PDP-11, this was not ,however, a feasible solution since such a machine was not, at the time, available at York. A remote mainframe

computer could be used but the requirement for high interaction and real time computation could not easily be met. A mainframe facility is, however, useful for 'off-line' control system design and assessment (see later). A compromise thus had to be arrived at which basically made best use of the available resources whilst sacrificing some of the 'real time' requirements.

Two machines were finally decided on which could provide the facilities required in the areas of interaction, graphics and ease of programming, these being :

i) An 8-bit CP/M based TUSCAN using the Z 80 processor with disc storage, medium resolution graphics and running a compiled Pascal language.

ii) A desk-top HP 9826 microcomputer using a 16-bit microprocessor (68000) with backup disc storage, medium resolution graphics and running the interpretive BASIC language.

Both machines were fitted with 64K bytes of RAM.

To evaluate the potential usefulness of these machines two computer programs were written, one in Pascal, the second in BASIC, to run on the above machines. These programs solved the set of non-linear aerodynamic equations, outlined in Chapter 1, for the aircraft simulation. The basic structure of these programs will be outlined below. It became clear that despite the fact that the HP machine was running interpretive BASIC the execution times of the algorithm was considerably faster on this machine than on the TUSCAN. The use of the TUSCAN as a simulation machine was thus rejected and the HP machine was used as the simulation facility in all subsequent work. It is worth mentioning here that the HP machine would only run in something close to real time by making the integration step length quite large. This has some undesirable effects for reasons which will be made clear later. The HP as used is therefore by no means an ideal machine to use in this application. The addition of a compiled language to the HP, Pascal is available, would be a considerable advantage and

allow an order of magnitude improvement in execution speed to be realised. The additional overhead of graphics generation and display was later found to considerably reduce the execution speed of the HP machine, however, and this is again a disadvantage which will be discussed later.

We shall now move on to examine the development of the basic simulation program and the variations around the program to enable it to be used on the various computers indicated above.

3.1 Program Development

The fundamental requirement of the simulation program is to solve the non-linear equations of motion for the aircraft at discrete points in time. The values of the time varying variables within the program are updated at each time point, the time step length being chosen to follow adequately the dynamics of the system. The choice of step length is a crucial factor if a real time simulation was to be realised. Basically the program will require a finite time to execute a single loop through the algorithm and this time will represent the smallest practical time step achievable. An execution time (solution time) of say 100 ms. would be inadequate for real time simulation if the dynamics of the simulated system required that we sampled at say 50 ms. intervals. The choice of integration method used was allied to this requirement. A brief discussion of this problem is included below but basically a compromise between numerical stability and execution time must be met. For the initial work a simple discrete integration method was chosen namely :

$$x_{n+1} = x_n + \Delta t\ f(x_n, u_n \ldots\ldots)$$

where Δt is the integration step length. It should be noted that, for stability in any one mode :

$$|1 + \Delta t\, \lambda_i| < 1$$

Thus the overall stability of this method is guaranteed if and only if :-

$$|1 + \Delta t \lambda_{max}| < 1$$

$$\text{i.e. } |\Delta t \lambda_{max}| < 2$$

$$\text{or } \Delta t < |2/\lambda_{max}|$$

For very widely separated modes this imposes a large constraint on the maximum usable step, an anomalous requirement since the fastest modes in the system are not of interest to the control system designer. This property alone suggests that a more suitable numerical integration method be chosen for this problem. It is felt appropriate to use an A-stable method such as the Trapezoidal Rule, based on the locally updated Jacobian matrix of the non-linear system. This would guarantee left-hand half-plane stability for large step lengths with the only penalty being one of accuracy. In the later work on control system design the real time aspects were not important and a fourth order Runge Kutta Merson algorithm was used as the integration technique. A FORTRAN version of the simulation was evolved based around this algorithm.

A number of generations of program were evolved, each differing from the previous generation basically in the input and output facilites provided. As mentioned above two versions of the program were necessary, one written in Pascal for the TUSCAN and DEC-10 and one written in BASIC for the HP 9826. Additionally, a FORTRAN version of the program was developed for the 'off line' design and assessment of various control schemes and to investigate alternative integration techniques.

Chronologically, the different generations of the program evolved as follows :

Vers. 1) A Pascal program developed around a tree structure. This version had only simple input and output facilities. Input being from data stored

on a disc file, read at run time, output being numerical to a v.d.u. This allowed an assessment to be made of the capabilities and shortcomings of such a structure.

2) A BASIC program having the same structure as version 1 and similar input and output formats. This BASIC version was required for assessment of the capabilities of the HP 9826 machine.

3) A modified version of 2 which allowed improved output facilities in particular the ability to display graphical plots of a number of the simulation variables vs. time or any of a number of other simulation variables.

4) An improved version of 3 which removed much of the tree structure in favour of a modular structure. In addition the arithmetic operations were optimised such that only changing variables were updated at each time step. This optimisation was basically done to improve the speed of execution of the algorithm.

5) An identical program to V.4 except for the inclusion of better input facilities. This was achieved by interfacing simple joystick and trim controls to the HP via. the GPIB. It was thus no longer necessary to read the input data from a disc file at run time. Output was still as in version 3.

6) An identical program to version 5 but with improved output displays. Principally a typical head-up type of display was developed indicating pitch, yaw and roll attitudes in addition to height, heading, wind speed and direction and a number of other simulation variables. Input to this program was as in V.5.

7) A FORTRAN version of the simulation based on a fourth order Runge Kutta Merson algorithm from the NAG library (166). The real time aspects were not considered important for this version. Input is via disc file, output is to a disc file for later processing by a graphics generating program. This version was used for the non real time studies of control law performance.

Of the above version 6 was considered to be as close to the ideal real time simulation program as could be achieved at this stage. Its major shortcoming was found to be the execution time overhead imposed by the head-up display. It would be considered desirable to off-load this overhead onto a dedicated graphics processor for future work. A BBC computer has recently been purchased primarily for this purpose and a project on this aspect of the work has already commenced.

In order to examine the program structure in more detail we shall consider, initially, the vers. 1 program and examine how this solves the non-linear equations of motion and how data is input and output. Subsequent version numbers will then be considered and the areas in which these differ from vers. 1 indicated. Note that all except versions 1 and 7 of the program were written in BASIC to run on the HP 9826 and exploit the 9826's input and output facilities.

3.1.1 Simulation Program Vers. 1

The first generation of the simulation program is written in Pascal and a listing is provided in Appendix 1. We may decompose the program into a tree type structure consisting of a main program part which calls various procedures. Each of these procedures may then call a number of other procedures thus implementing a specific function. This structure is shown diagrammatically in Fig. 3.1. In the following discussion the function of each of the

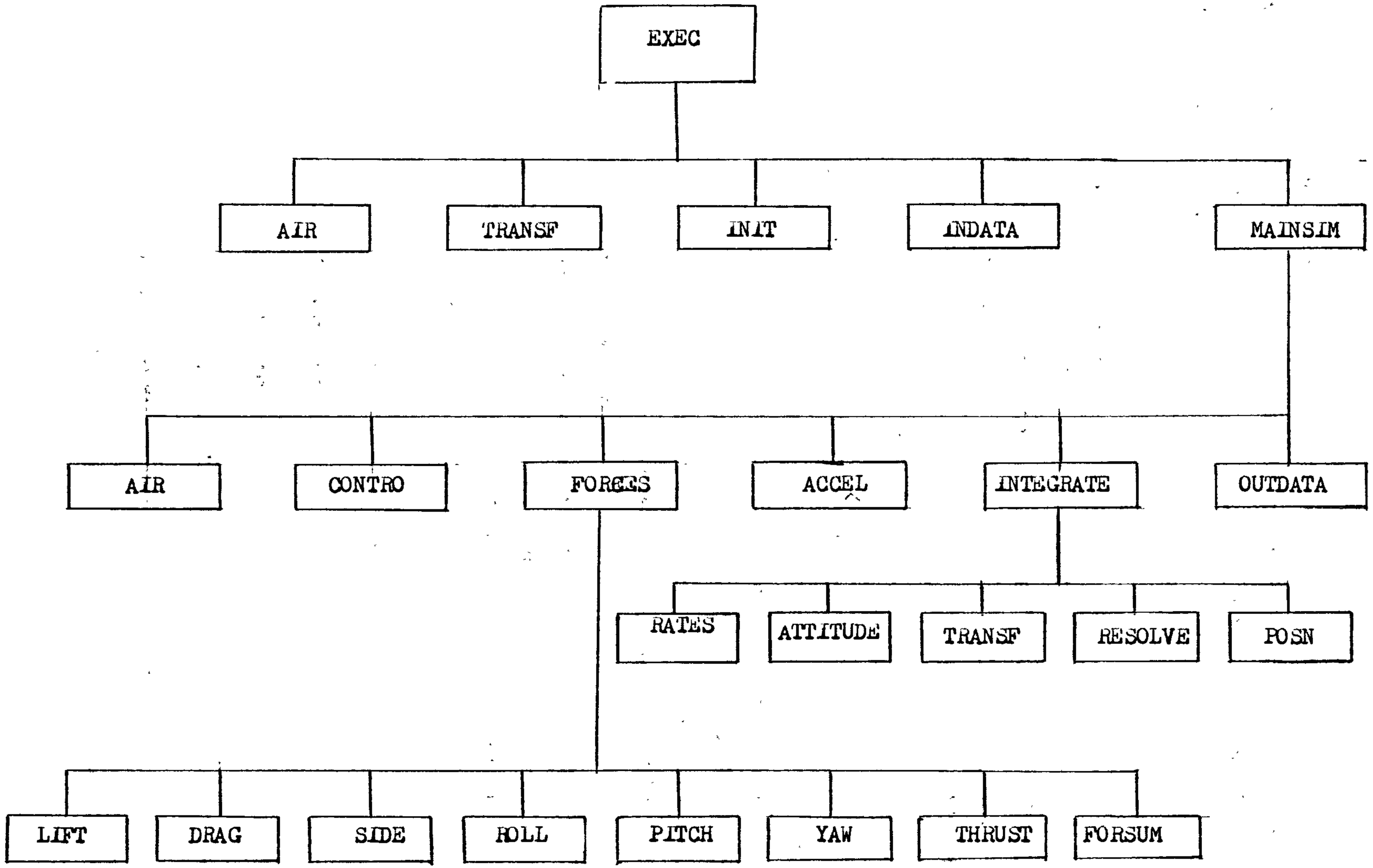


Fig 3.1 Procedure Architecture

procedures will be briefly indicated starting with the main program and working down. Line numbers referred to in the text are those given in Appendix 1. Internal variable names are fully described in the Pascal variable declaration part of Appendix 1.

The Pascal main program, lines 682-688, calls four data initialisation procedures before handing over control to the MAINSIM procedure. The initialisation procedures function as follows.

i) Indata :- The pertinent simulation data are stored on three disc files (VAR1-3.DAT). Prior to commencing the simulation these data are read from file by this procedure and used to initialise the relevant variables. The run time files contain the following data.

   a) VAR1.DAT :- This file holds the initial values of a number of run time variables. The total simulation time, integration step length, and printing time interval are followed by the vehicle c. of g. position, wind speed and direction, initial earth axis velocities, initial pitch, roll and yaw angles and initial height. These variables will normally be set to reflect the initial conditions pertinent to the given simulation run. The sequence of data which should be provided in this file are as follows :

length of simulation	secs.
Integration step	secs.
print step	secs.
c. of g. position	fraction of wing chord
wind speed	m s^{-1}
wind direction	degrees (0° = headwind)

initial	northerly speed	m s^{-1}
"	easterly "	m s^{-1}
"	vertical "	m s^{-1} (down = positive)
"	roll attitude	degrees
"	pitch "	"
"	yaw "	"
"	heading	degrees from north
"	height	meters

b) VAR2.DAT :- This file contains the sequence of pilotic inputs for the particular simulation run. Each sequence basically contains the time point for the change in pilot demand, throttle, elevator, aileron and rudder demands. The file is read repeatedly and each sequence stored for use in the simulation. This file may be changed by the user to simulate pilotic commands for a given run. For example a launch simulation may require data such as

time = 0 secs. 0., 1., 7., 0., 0.

" = 2 secs. 2., 1., 5., 0., 0.

The initial time (time =0 secs.) settings are always required and in this case command full throttle (=1.0), elevator angle of 7.0 degrees, zero aileron and zero rudder. The elevators are backed off by two degrees to 5 degrees after 2 seconds. Up to 10 of these sequences may be input, although this may be extended by modifying the program.

c) VAR3.DAT :- This file contains all of the fixed aerodynamic and physical variables for the particular aircraft and these may be set prior to running the simulation.

The Indata procedure reads each of these files sequentially and stores each of the values read in the appropriate variables. The VAR2.DAT file is read 'n' times until the end-of-file marker is found. The value of 'n' is used as an indicator of the number of changes in pilotic demand which are required. The initial values of aircraft attitude, control surface deflections and wind direction are also converted to radians for later use.

2) Transf :- This procedure simply evaluates the direction cosine matrix of equation 1.1 (transform) using the current values of pitch, roll and yaw angles (theta, phi and psi). The inverse i.e. transpose of this matrix is also evaluated and stored in transpose.

3) INIT :- This procedure simply initiates a number of simulation variables to zero in addition to setting up the value of air density (rho) and gravitational acceleration (gravity). The initial inertial body velocities are also evaluated using the direction cosine matrix of equation 1.1 and the initial values of earth axis velocities (Ug, Vg and Wg). Note that a variable stall is used to indicate a stalled aircraft configuration. This variable is initially set to zero but may be set to 1, indicating stall, by the lift procedure. Note also that the inertial body velocities (Ub, Vb and Wb) are not identical to the aircraft axial airspeeds (Uc, Vc and Wc) the airspeeds being related to the ground speeds by the direction cosine matrix in the Air procedure.

4) Air :- The relevant body airspeeds are evaluated here using the d.c.m. and the current ground speed values given by (Ua, Va and Wa) these being evaluated from Ug, Vg and Wg and the appropriate headwind and vertical gust velocities. The incidence (alpha) and sideslip (beta) angles are also evaluated using equations 1.4.

The aircraft's total velocity is evaluated using the Uc, Vc and Wc velocities in equation 1.3.

After calling the above four procedures the simulation proper is commenced by calling the MAINSIM procedure.

The body of the MAINSIM procedure consists of a REPEAT/UNTIL loop which calls the procedures required to implement the simulation at each integration time step. A loop counter is used to increment the simulation time counter (T) and a decision is made as to whether data should be written to the v.d.u. using OUTDATA. Data are written only at fixed time points determined by the value of the 'tprint' variable. The loop is terminated only if the total elapsed simulation time exceeds the simulation end time, tend, or if the aircraft stalls, indicated by the stall variable being equal to 1. Before entering the main simulation loop two routines are called, CONTRO and OUTDATA, which set up the initial control surface deflections and throttle demand and write out a header sequence to the screen as indicated below. The procedures called from MAINSIM which have not been described above function as follows :

5) CONTRO :- This procedure simply reads the values of the control surface deflections and the throttle setting from the arrays set up by Indata. These data will only be changed when the current time equals or is in excess of the time at which the control settings are to be changed indicated by the elements of the 'time' array.

6) Forces :- This procedure is used to call a number of procedures which evaluate the aerodynamic forces acting on the aircraft as indicated in chapter 1. These are as follows :

a) Lift :- Evaluates the wing and pitch rate lifts using equations 1.13 and 1.18. Additionally the lift coefficient, C_L, is determined using equation 1.14 and if this is above a value of 1.2 the

aircraft is considered to be stalled, 'stall' then being set to 1.

b) Drag :- Evaluates the aircraft's drag force using equation 1.12 b) and the drag coefficient, C_D, value given by equation 1.21.

c) Side :- Determines the side force, Y_a, using equation 1.10 b).

d) Roll :- Evaluates the rolling moment L_A as given by equation 1.10 d).

e) Pitch :- Evaluates the pitching moment M_A and the taillift given by equations 1.10 d) and 1.15. The pitching moment coefficients C_{MT} and C_{MW} are as given in equations 1.16 and 1.22.

f) Yaw :- Determines the yawing moment N_A as given by equation 1.10 f). The N_R aerodynamic derivative is given by

$$N_r = N_{ro} + N_{rl}\ C_L^2$$

where N_{ro} is the yawing moment coefficient at zero lift, N_{rl} is the yawing moment derivative w.r.t. C_L, the lift coefficient.

g) Thrust :- The thrust force X_E is given by equation 1.33. This procedure evaluates X_E as per section 1.3.8. The procedure also evaluates L_E according to equation 1.35.

h) Forsum :- This procedure simply evaluates the total forces (X, Y and Z) and moments (L, M and M) according to equations 1.27 in addition to evaluating the total lift force acting on the aircraft, (ltotal).

7) Accel :- The time derivatives of the body inertial

velocities (UBDOT, VBDOT, WBDOT) and the pitch, roll and yaw rates (PDOT, QDOT, RDOT) are evaluated by this procedure according to equations 1.7. These values are clearly the respective linear and angular accelerations of the aircraft in the inertial axes.

8) Integrate :- This procedure calls a number of other procedures which together perform an integration on the time derivatives of body velocities and rates. These procedures also transform the resulting body velocities into the earth co-ordinate system and integrate the resulting ground speeds to give heading, downrange and height values. The particular procedures called here are as follows :

a) Rates :- This procedure integrates the body pitch, roll and yaw accelerations (QDOT, RDOT, PDOT) using the simple lst. order integration method of equation 3.1. The procedure also converts the radian measures of these rates into degrees.

b) Attitude :- The pitch, roll and yaw angles in earth axes (Theta, phi and psi) are evaluated using equation 1.2 a) and integrating the resulting time derivatives (Thetadot, Phidot and Psidot). These angles are also evaluated in degrees (Thetadeg, Phideg and Psideg).

c) Resolve :- This procedure integrates the linear accelerations in inertial body axes and evaluates the linear velocities in inertial axes (UB, VB and WB). These velocities are then resolved into the ground axes to give the groundspeeds (UG, VG and WG) using the direction cosine matrix of equation 1.1, updated by the Transf procedure.

d) Posn :- This procedure simply integrates the ground velocities to give the ground positions namely XN, northing position, YE, easting position

and Height, the aircraft height. Note that height is measured as the negative integral of up/down velocity, WG.

e) Outdata :- This procedure outputs the values of various variables to the currently assigned output device, in this case a v.d.u. An integer parameter is passed by the calling program which indicates which one of three possible output format options is to be used. The pass parameter controls the output as follows.

i) option = 1 :- In this case a header sequence is output followed by the initial values of total simulation run time, tend, time integration step length, Tstep, print time increment, c. of g. position, initial aircraft attitude and control surface positions. This option is used only once on the initial pass through the simulation loop.

ii) option = 2 :- A header sequence for the run time output variables is output for this option. The variables currently output are the current time, easterly air speed, total air speed, northerly airspeed, heading, roll rate and roll angle. These output variables and headings may be chosen by editing the program.

iii) option = 3 :- This option simply outputs the variables as indicated above to the v.d.u.

The Outdata routine is called at the print interval time points from the MAINSIM procedure.

The body of the MAINSIM procedure is reasonably self-explanatory and simply consists of a REPEAT/UNTIL loop which calls the appropriate procedures for the simulation in

addition to controlling the outputting of data via. Outdata. The simulation is terminated by the simulation time exceeding the defined end time or by the aircraft stalling (indicated by the stall variable).

3.1.2 Simulation Prog. Vers. 2-6

For the initial TUSCAN and DEC-10 studies the above describe Pascal program proved adequate as an evaluation tool. For the HP 9826 machine a BASIC program was necessary and this formed the version 2 generation of the program. The original version 2 program which was developed was very similar in structure to the Pascal version 1 program described above. In order to exploit the many graphics features of the HP machine a version 3 form of the program was evolved. Additionally, it was decided, at this point, that the execution speed differential between the TUSCAN and the HP machines was such that the HP machine should be used in all subsequent simulation work.

Versions 4 to 6 of the simulation program were thus all written in BASIC for the HP machine and all share a similar main simulation structure. The principal areas in which the four versions differ is in the type of input and output facilities available. These differences are apparent from the listings of versions 3, 5 and 6 given in the appendices. Note that the limited memory available on the HP required that various subroutines be loaded from disc at run time and deleted after use to allow the main simulation program to reside in the available memory at run time. This restriction requires a fairly elaborate driver program to manage the loading and unloading of the memory. This driver segment thus forms the only program loaded and run by the user.

We shall now briefly examine the extended input and output facilities afforded by the HP machine and how these have been exploited in version 4, 5 and 6 of the simulation.

i) Facilities for graphical output of data :-

Appendix 2 details the structure of version 4 of the program. It will be noted that the main simulation has the same general structure as the Pascal version 1. The procedures used by 'Forces' (see 6 a) - d) above) are , however, now concatenated into one large subroutine 'Forces'. This was done in order to increase the speed of execution of the program by eliminating a number of CALL statements. As noted previously a limited amount of memory was available on the HP machine. For this reason a menu is provided by the PREAMBLE routine which allows the user to set up the axes and scaling and define the plot channels for a particular run. Two files "AUTOPLT3" and "DEFXY" are loaded from disc to provide these facilities and are subsequently deleted to allow the main simulation program, in files "MACH1", "MACH2" and "INITIALISE", to be loaded into the memory at run time. The graphics screen of the HP is entirely separate from the alpha screen and hence a simple DRAW instruction, called at each time step in MAINSIM, plots out the required data on the graphics screen assuming that this has previously been initialised via. a call to Gset or Setxy in AUTOPLT and DEFXY. An indication of the channels for plotting are stored on disc in a file "XYCHAN" which is intialised by the subroutines contained in AUTOPLT3 and DEFXY and read by MAINSIM at run time.

Using this system a two dimensional plot of any of 10 predefined simulation variables against any of the same 10 variables is achievable with interactive user axis scaling. Some results of simulation runs using this system are given in the following chapter. Note that runs may be repeated using the same axes and plot channels to provide multiple plots on the same axes. The graphics screen is only erased after a call to Gset or Setxy and hence re-initialising the graphics or re-defining plot channels will erase the graphics

data.

A number of further points are worthy of mention here which relate to specific features of the HP machine and HP BASIC. Firstly, unlike Pascal, variables in HP BASIC are local unless declared otherwise. For this reason a COM (common) statement is provided which allows variables to be communicated 'globally' in a similar manner to the FORTRAN COMMON statement. Named common blocks are thus extensively used in the simulation programmes, these simply allocating data storage space at pre-run for all of the 'commoned' variables. The HP machine allows for programming of 'soft keys' on the keyboard and the execution of subroutines on depression of these programmed keys via. the ON KEY statement. This facility is used extensively to provide the interactive graphics user operation. The alpha display on the HP is still used by the Outdata subroutine and hence alpha and/or graphics data may be displayed during a particular run. I/O path names are used for all data transfers to and from the disc. This basically increases the speed of transfer.

For further specific information on the HP 9826 the reader is refered to references 161, 162, 163 and 164.

ii) Facilities for Joystick type Interactive inputs :-

In versions 1 through 4 of the program control surface demands are define 'off-line' and read from a data file "VAR2.DAT" or "VAR2.BDAT" as described above. An interactive facility using a typical joystick type of input was required for vesions 5 and 6 of the simulation. This facility was provided by a joystick potentiometer providing elevator and aileron commands in addition to trim pots. for rudder, aileron, elevator and throttle demands. The interface for these analogue (d.c.) signals was provided by an eight channel A/D

converter (RS type 305-545) of which only six channels are used in this application. The A/D was scanned through the six analogue inputs by the HP 9826 via. the GPIB interface bus. Data from the A/D was also read by the HP via. this bus. A circuit diagram and brief description of the A/D unit is given in Appendix 3. Scaling is applied to all of the A/D outputs by the simulation program as it is read from the GPIB. Version 5 of the program exploits this feature and the primary area in which the program differs from version 4 is in the Contro routine. This routine is listed in Appendix 4. It will be recalled that the Contro routine assigns values to the control surface positions and updates these at each time step. Whereas before these values were preset by the contents of a data file "VAR2.BDAT" the values are now assigned by scanning the A/D on the joystick module, reading the resulting data and applying appropriate scaling to each of the elevator, aileron, rudder and throttle settings. The simulation program now uses these settings as previously. Note that the GPIB interface is device 12 on the HP and this is assigned to the I/O path @Gpio_path.

iii) Facilities for Head-up type output display :-

To provide a more comprehensive display0of simulation variables a head-up type of cockpit display was devised for the version 5 program. This display uses the graphics facilities of the HP to provide a 'moving horizon', height and heading dials and displays of the various aircraft attitudes, control deflections and airspeeds. A typical output format is shown in Fig. 3.2. Version 6 of the simulation is provided with this output format via. routines held in a disc file "HEADUP". This version also has the joystick type input format as version 5. The headup facility is provided primarily by a suite of routines on two disc files, "HEADUP" and "UPDATE". The routines within

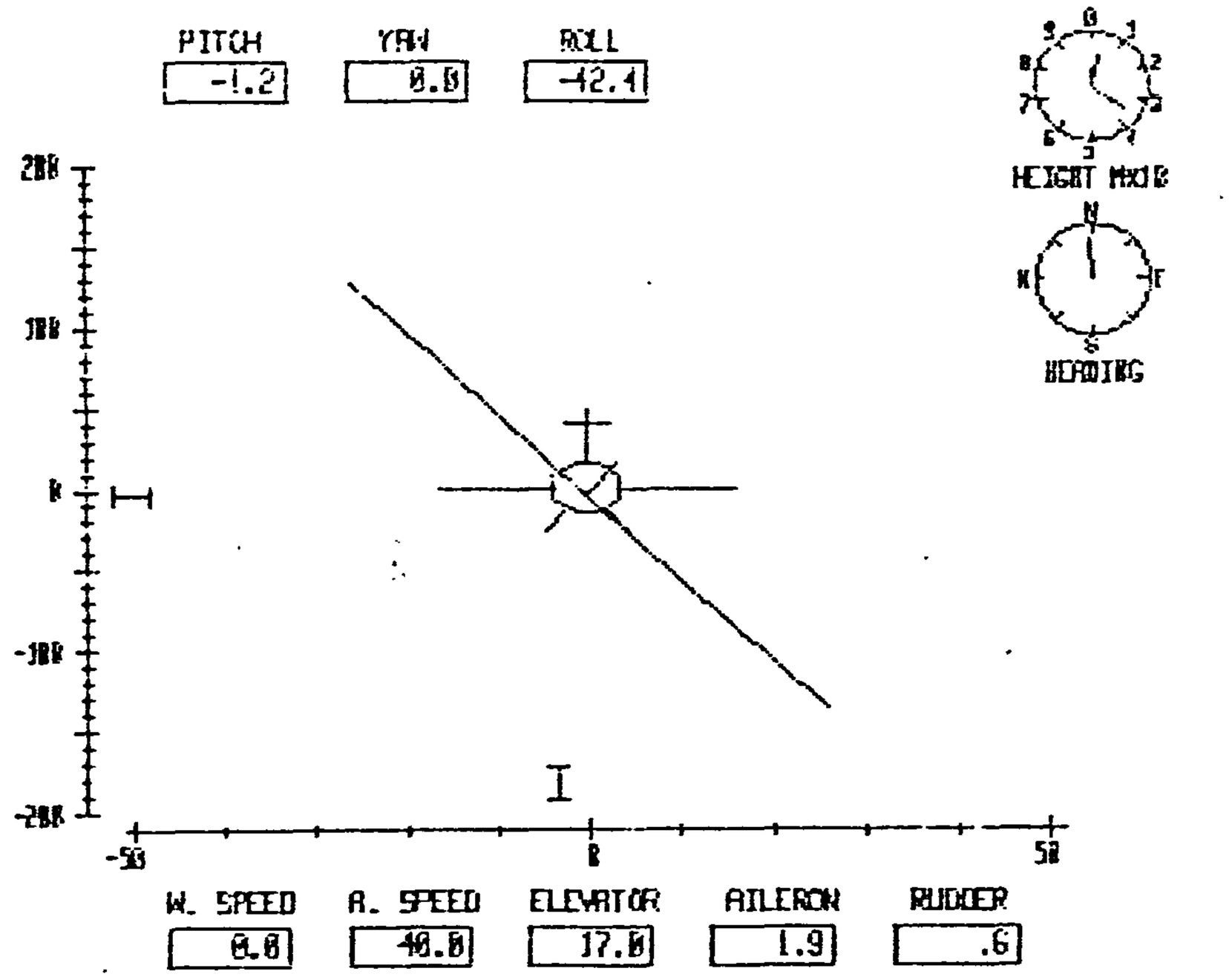


Fig. 3.2 Head-up Display Format

"HEADUP" basically initialises the graphics display for the run by drawing the horizon, dials and boxes and labelling these accordingly (see fig 3.2). In order that the MAINSIM routine may update the horizon position, boxes and dials a suite of routines in "UPDATE" are provided. These routines first have to undraw any updated line positions on the display, based on the old positions of the lines and then draw any new lines at the positions indicated by the new values of the variables passed to these routines in a common block. A listing of these files is provided in Appendix 5 along with the modified MAINSIM and driver routines. The MAINSIM subroutine is now modified so that the new simulation variables are written into a 'New' array before updating the display. The previous values of these variables are held in an array 'Temp' which is updated to the values in the 'New' array after re-drawing the display. The main body of the simulation remains the same as that in version 4 and the joystick input facility in retained.

3.1.3 Simulation Prog. Vers. 7

This version of the simulation was developed principally in order to carry out investigations into the performance of various closed-loop control laws. To this end the real time aspects were largely neglected in favour of an improved integration technique. The program was intended to run on the remote DEC-10 mainframe computer and a listing of this version is provided in Appendix 6. The investigation of a different integration technique was desirable to provide a comparison with the previously used Euler technique and in order to ensure accurate modelling of any fast modes brought about by the closed-loop feedback. The particular integration method chosen was a fourth order Runge Kutta Merson algorithm from the FORTRAN NAG library (D02 BBF). Employing this routine imposes a set structure on the simulation which is basically of the top down form of vers. 1. Referring to the listing of Appendix 6 we note

that the routine, D02 BBF, is called from the driver routine which initially establishes from the user the simulation length and time step size between successive 'samples' of the system 'states'. The 'states', X(I)'s, and derivatives, XDOT(I)'s, are also initially set to values read from a data file on disc (INSTAT.DAT) by the INITAL subroutine. This routine also reads and sets up the aerodynamic constants from a disc file (CONT.DAT) in addition to reading the pilotic demand data (DEMAND.DAT) and the state feedback gains (FEEDBK.DAT). Note that in these closed-loop studies we are normally concerned with state feedback structures and hene we require two state feedback gain matrices defining the state and reference feedback gains. The data files INSTAT, DEMAND and CONT are of roughly the same format as the VAR1, VAR2 and VAR3 files in previous versions of the simulation.

The call to D02 BBF specifies the initial time, T, end time, TEND, number of states, N, and an overall error bound on the itteration, TOL. The routine also requires the calling program to nominate two subroutines, FCN and OUTPUT which D02 BBF calls at each user specified time step, TSTEP. FCN must contain the computation necessary in order to determine the values of the XDOT(I) variables at each integration step. For this example FCN calls the appropriate routines to evaluate the states, X(I)'s, from the non-linear equations of motion for the aircraft using much the same subroutines as for the Pascal and BASIC versions. The CONTRL routine provides for closed-loop control by evaluating the control surface demands via the set of state and reference feedback gains. The DIFFS routine then evaluates the XDOT(I) values from the set of state variables, X(I)'s. The OUTPUT routine provides a means of outputting simulation data at each time step. In this case the values of the states are written into a large array, RESULT, whose column count is updated at each step by the loop counter RESCNT. On exit from D02 BBF this result array is written to a disc file, nominated by the user, for later processing by a graphics program which provides the state responses in graphical form.

3.2 Summary & Further Work

In this Chapter the implementational aspects of the non-linear aircraft model have been considered. Initially it was felt that a close to real time simulation was required in order to allow an 'on-line' assessment of the aircraft performance to be made by interacting with the simulation via joystick type controls. A digital simulation technique was chosen and a number of machines were investigated with a view to providing this function. Ultimately, a dedicated desktop machine was chosen, an HP 9826, which provided a close to real time simulation. The addition of improved graphics to this machine, however, severely compromised the real time aspects. It is felt that the addition of a compiled language to the HP 9826, namely Pascal, and the off loading of the display generation to, for example, a BBC micro would provide greatly enhanced performance. A third computer may also be required to carry out any closed-loop control and interface to the pilotic joystick. The Pascal version 1 of the simulation running on the 9826 would then provide a real time simulation facility, closer to the initial objectives of section 3.1.

The problem of integration method was briefly mentioned above. A simple first order Euler approximation is currently used in the real time work so as not to compromise the execution speed. In later work a fourth order Runge Kutta Merson routine has been used with a FORTRAN version of the simulation, version 7, to provide an improved integration technique but this clearly degrades the real time aspects. Version 7 of the simulation has, however, been used to provide 'off-line' assessments of various closed-loop controller configurations prior to implementing these in a real time environment. With improved real time computing power it may be possible to employ a somewhat more accurate integration method based on, for example, the trapeziodal rule or a low order Runge Kutta. Inevitably, however, a compromise must be found with respect to step length vs. computational time and stability of the integration routine. These aspects are currently under

review as improvements in the real time simulation are being developed.

CHAPTER 4

Open-Loop Simulation :- Results

In this chapter the results of a number of simulation runs using version 4 of the program and the graphical output facilities are detailed. The pilotic demand signals are generated by editing the VAR2.BDAT file whilst initial conditions are defined via. the VAR1.BDAT file.

The particular simulation runs considered emulate the landing and take off manoeuvres. The sequence of control inputs required was determined by repeated simulation and editing of the pilotic input sequence. In this way an appropriate sequence of pilot demands was developed for each of the runs. Of particular interest was the landing and take off behaviour since these would allow the pilotic workload during these manoeuvres to be easily assessed. Two situations will therefore be considered these being :

i) Take off from launch to steady flight

ii) Landing with regard to the flare and float times

The results are presented for each of the above and some indication made of the degree to which these results compare with actual telemetered results from the real aircraft.

i) Take Off

The remotely piloted vehicle which formed the basis of the simulation is launched using a pneumatic launcher which, it is assumed, raises the aircraft's forward velocity to the stall speed of 25 m s^{-1} and initial nose up pitch attitude of 7 degrees. The aircraft would normally be launched under full throttle.

To assess the potential pilot workload on launch, a

series of runs were undertaken for a range of elevator settings, zero wind speed and full throttle. The simulation step length was chosen as 0.01 s and the print increment as 0.1 s. The results of these simulations are shown in Figs 4.1. The time-height histories demonstrate that a launch elevator setting of about 7 degrees may be tolerated before the aircraft stalls on launch. This corresponds to about 1 degree forward of the trim setting of 6 degrees for 25 m s^{-1}. Attempting to climb the aircraft from launch (elevator settings of < 6 degrees) resulted in stall on launch. Attempting to hold the aircraft's nose down on launch (elevator settings > 6 degrees) resulted in the launch trajectories of Fig 4.1 a) for elevator settings from 7 to 12 degrees. An elevator setting of, say, 7 degrees would thus result in a condition in which the aircraft could probably be pulled into the air 4 seconds after launch. The variation in air speed with an elevator angle of 7 degrees is also shown in Fig. 4.1 b).

For the condition of launch with a headwind and full power the time-height trajectories are shown in Fig 4.2. Fig 4.2 a) shows launch with a 5 m s^{-1} headwind, b) shows the results for a 10 m s^{-1} headwind. In both cases a launch elevator angle of 5 degrees or less now results in a stall at take off. The aircraft will tolerate a smaller stick forward demand and still climb to altitude. As expected the aircraft climbs more rapidly with the larger headwind value.

Power loss on take off was also examined for a range of headwind values. These results are shown in Figs 4.3 a) and b) for elevator settings of 6 and 8 degrees respectively. We note that with 6 degrees of elevator the aircraft stalls for zero wind speed but climbs adequately for wind speeds of 5 and 10 m s^{-1}. For 8 degrees of elevator (4.3 b)) the aircraft remains landable even for zero power and zero headwind. For 5 and 10 m s^{-1} the trajectories show that the pilot could have landed the aircraft relatively easily after about 4 seconds.

The above results indicate that launch is a relatively easily controlled manoeuvre demanding little of the pilot.

Even for the case of zero power on launch the aircraft would appear to be recoverable. The results also correlate with those provided by Marconi Avionics who flew the remotely piloted vehicle at Bedford. This would thus indicate that the simulation provides a relatively good model of the launch manoeuvre.

ii) Landing

The landing manoeuvre requires that the aircraft looses altitude by diving, levelling out to a trimmed condition close to zero altitude and finally descending the remaining distance to the ground. This manoeuvre may be broken down into the flare and float, the flare being the period required to recover from the descent from cruise altitude, the float being the subsequent descent to touchdown. To simulate such a manoeuvre requires a resonably complex sequence of elevator commands. For our simulation initial values of 30 m s^{-1} forward air speed and 30 meters height were chosen, all initial attitudes being zero and zero headwind. Fig. 4.4 a) shows a typical landing profile. Figs. 4.4 b) through f) detail the pitch angle, pitch rate, height, elevator angle and airspeed time histories respectively. The points to note here are that the aircraft was trimmed for a power off glide with an initial elevator angle of 8 degrees , this being slighty forward of trim at 30 m s^{-1} and the aircraft thus pitches nose down after an initial transient. The aircraft loses height and the flare manoeuvre is initiated after 7 seconds by the pilot pulling back on the stick, an elevator angle of 7 degrees being demanded. The float period for the run then ensues after approximately 10 seconds, lasts for 6 seconds and touchdown occurs after 16.3 seconds. The elevators are trimmed during the float and the aircraft lands with a slight nose up pitch angle of 2 degrees at an airspeed of 25 m s^{-1}. Significantly, Fig 4.4 a) shows that the flare starts at about 220 m downrange finishing at about 300 m, the float then lasts for a further 130 m, touchdown occuring at 460 m

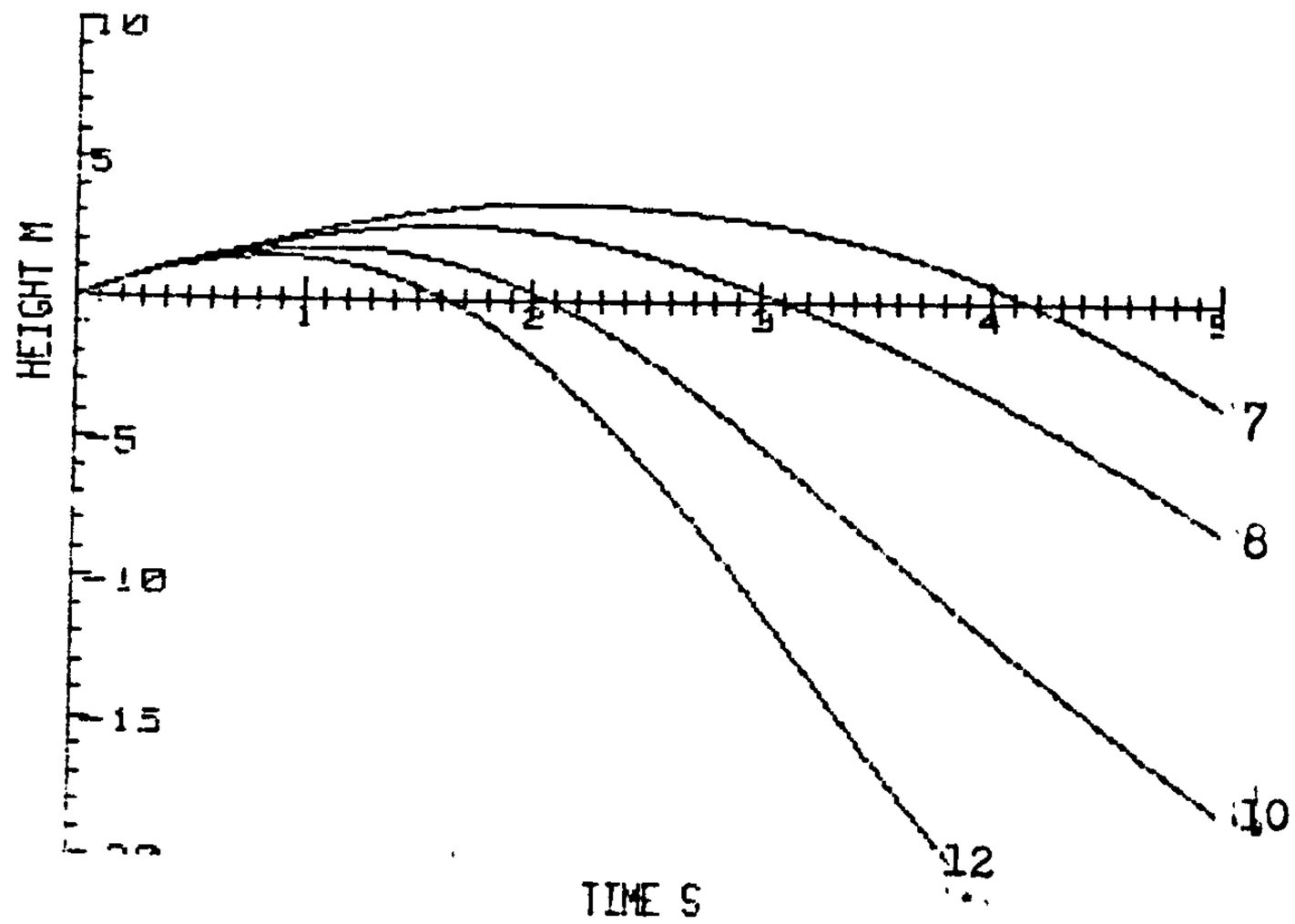
downrange.

By contrast, Figs. 4.5 a) through f) show the effects of the pilot attempting to 'dive off height' from the same initial settings. The pilot initially demands a 10.5 degree elevator setting which is well forward of the trim condition for 30 m s^{-1}. The flare manoeuvre now begins after approximately 4 seconds, the flare lasting 4 s and the float a further 5 s Again the landing profile of Fig. 4.4 a) shows that the flare must be started at a height of 15 m and lasts for approximately 100 m. The float lasts for some 200 m with touchdown occuring at 420 m downrange after a 14.2 second flight and at an airspeed of 25 m s^{-1}.

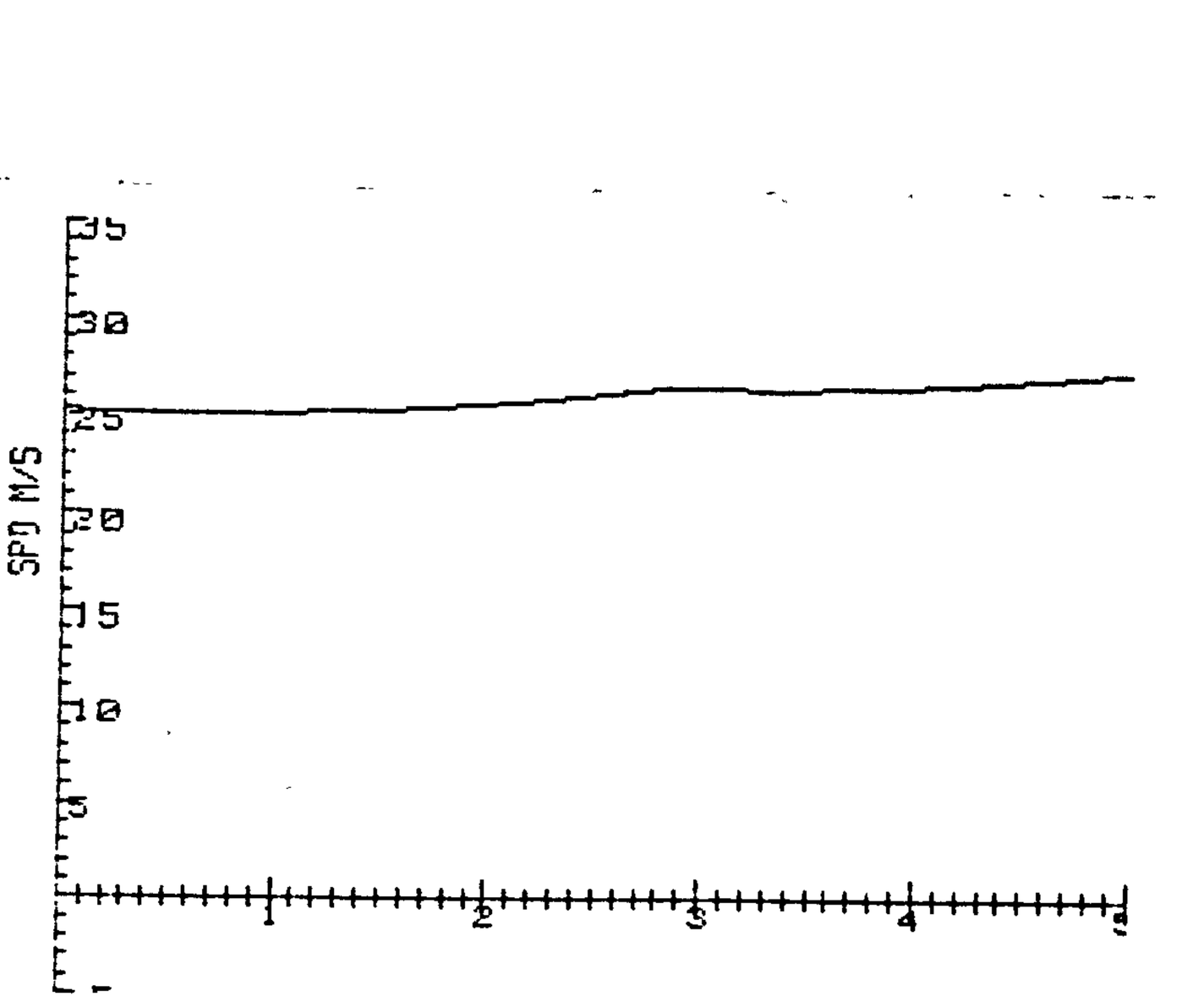
The 'dive off height' philosophy may thus be considered to increase the pilot workload significantly, this being evident from the larger number of pilotic elevator commands of Fig 4.4 e).

The results obtained above show good correspondence with the flight trials undertaken by Marconi (6).

Fig. 4.1 Launch Trajectories, full throttle, zero headwind

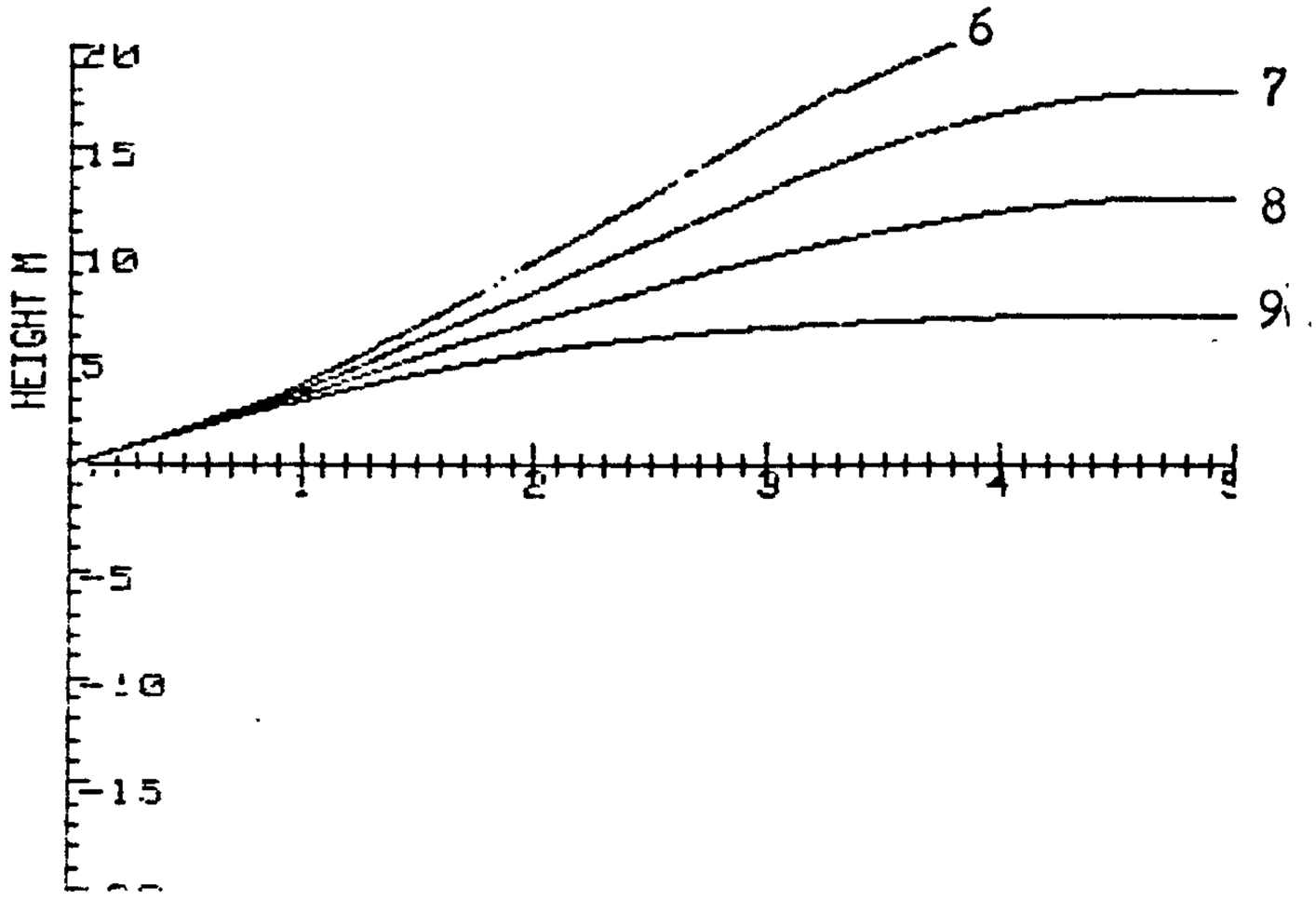


a)

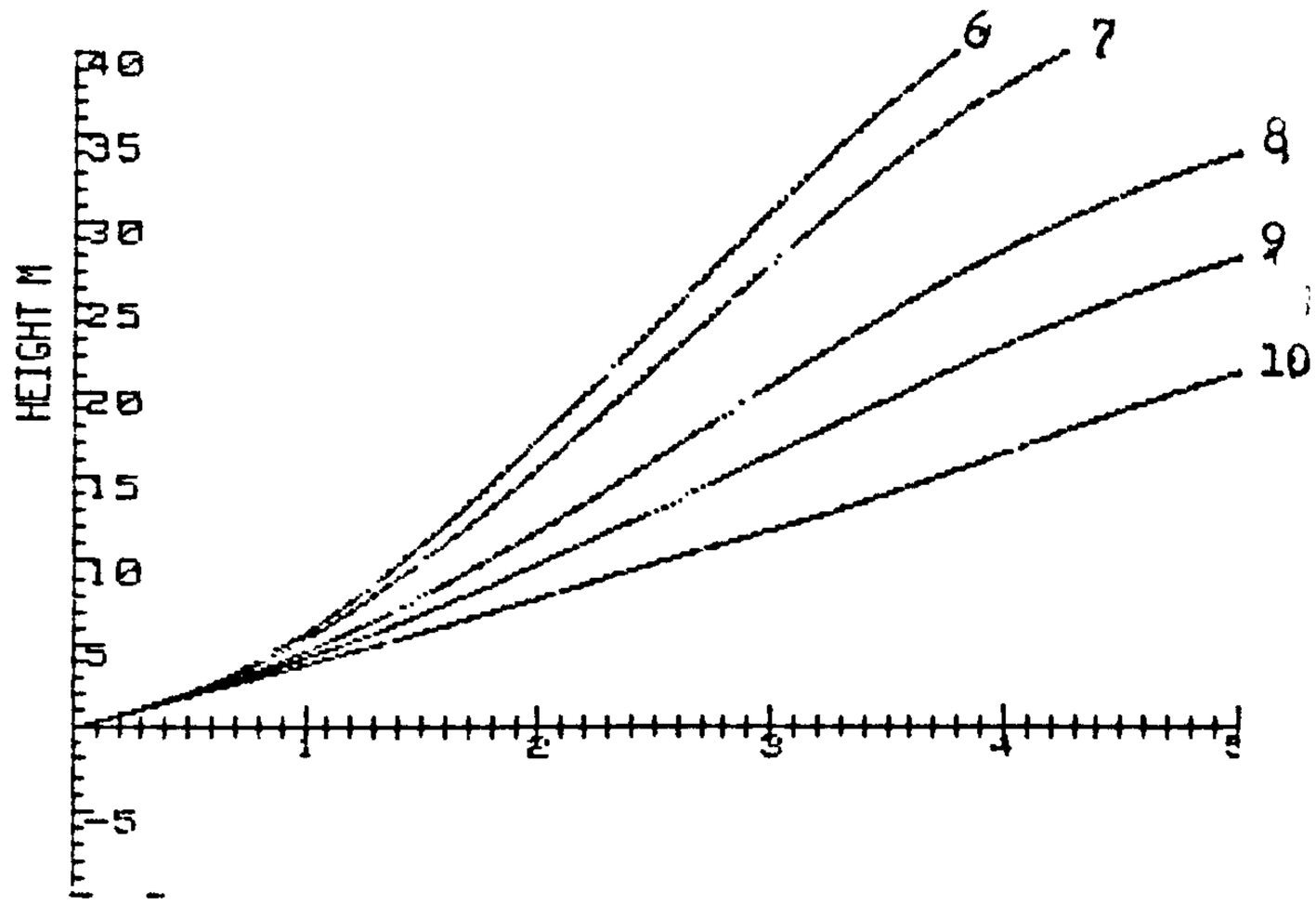


b)

Fig. 4.2 Launch Trajectories, full throttle,

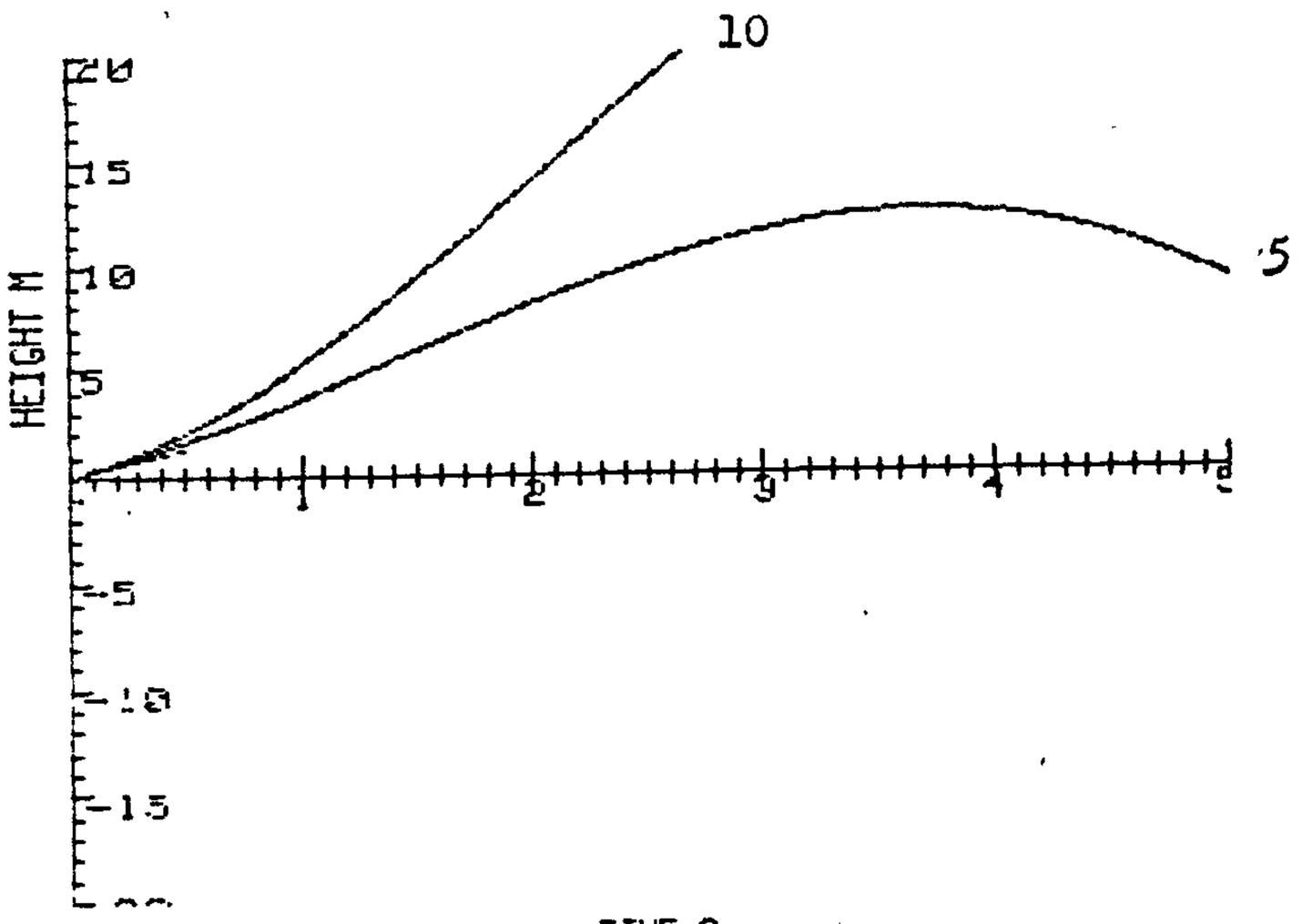


a) 5 ms^{-1} headwind

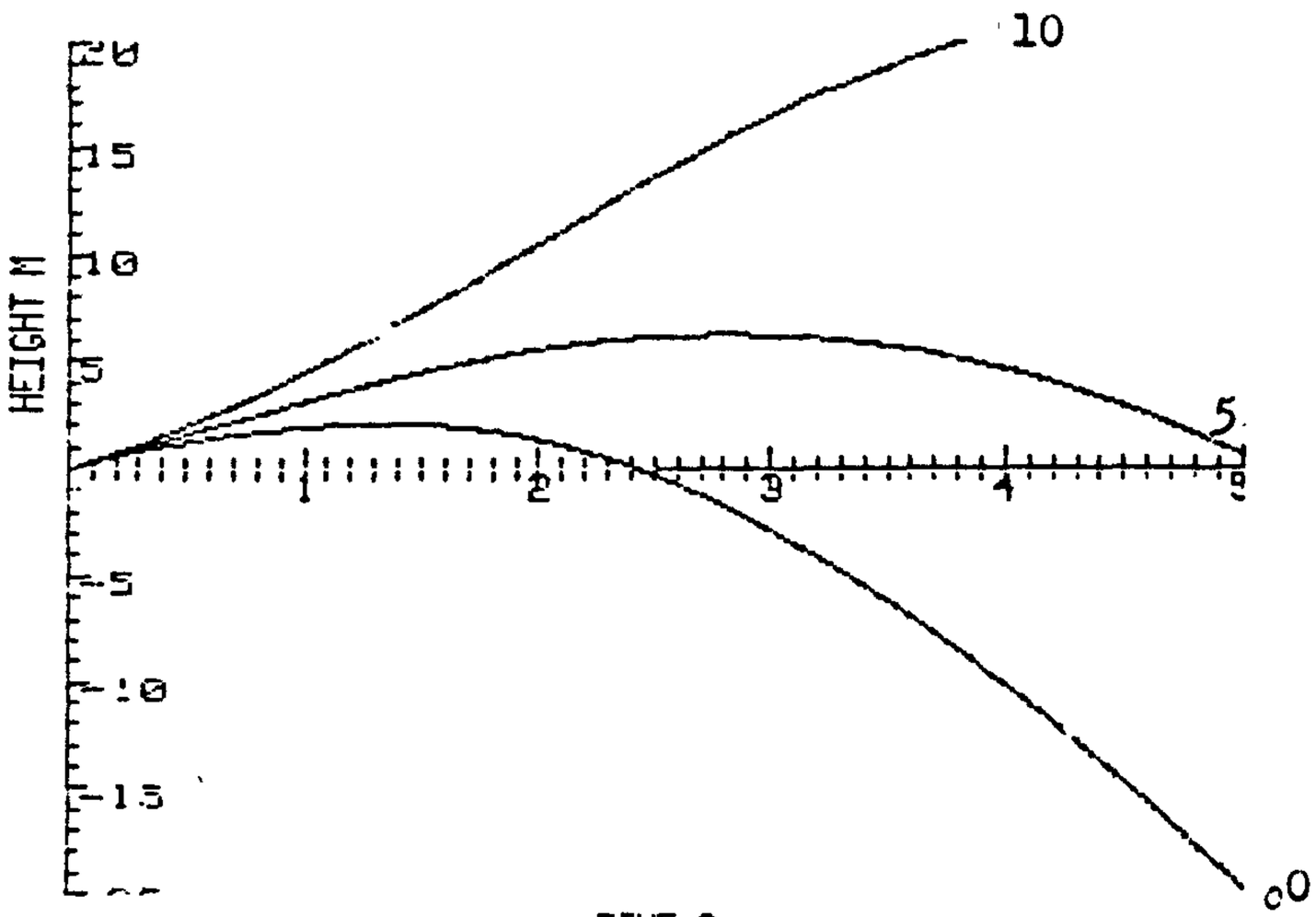


b) 10 ms^{-1} headwind

Fig. 4.3 Launch Trajectories, no power



a) Elevator 6 degrees



b) Elevator 8 degrees

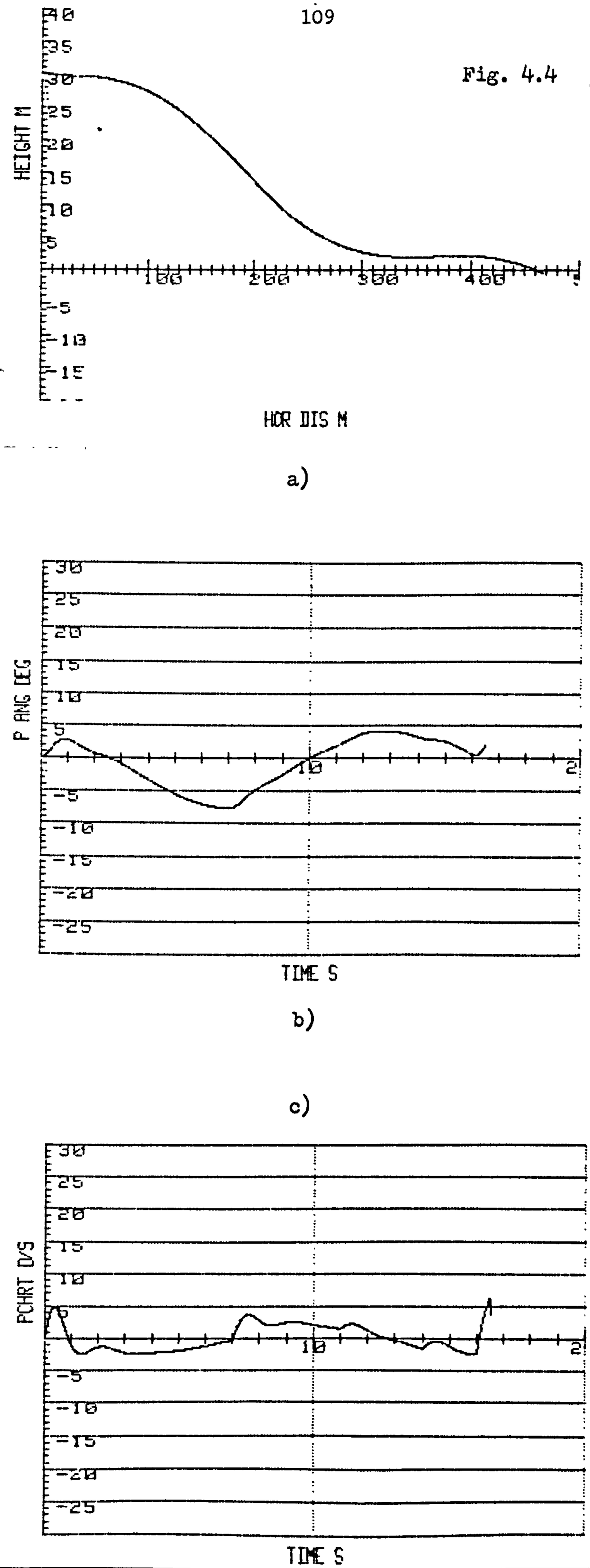
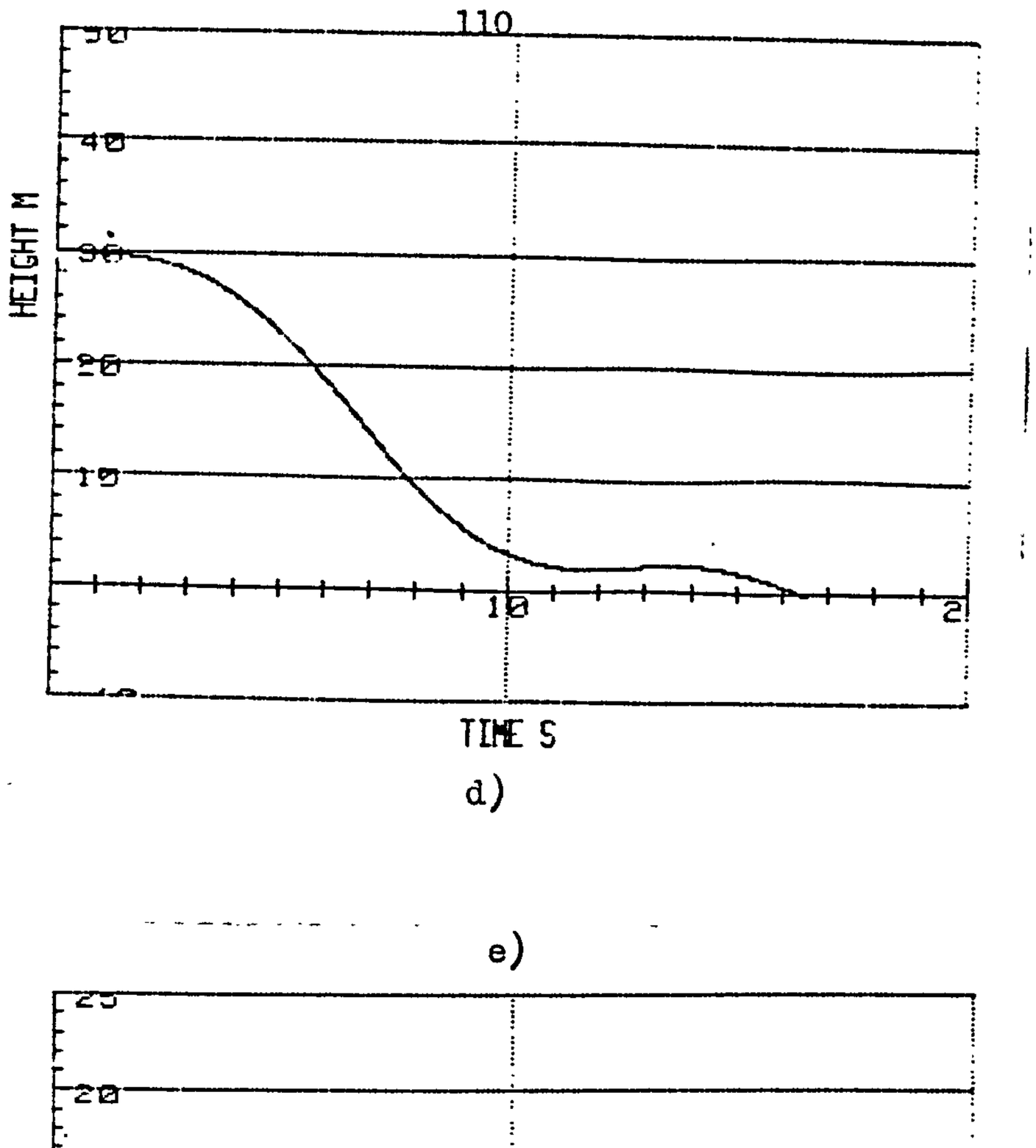


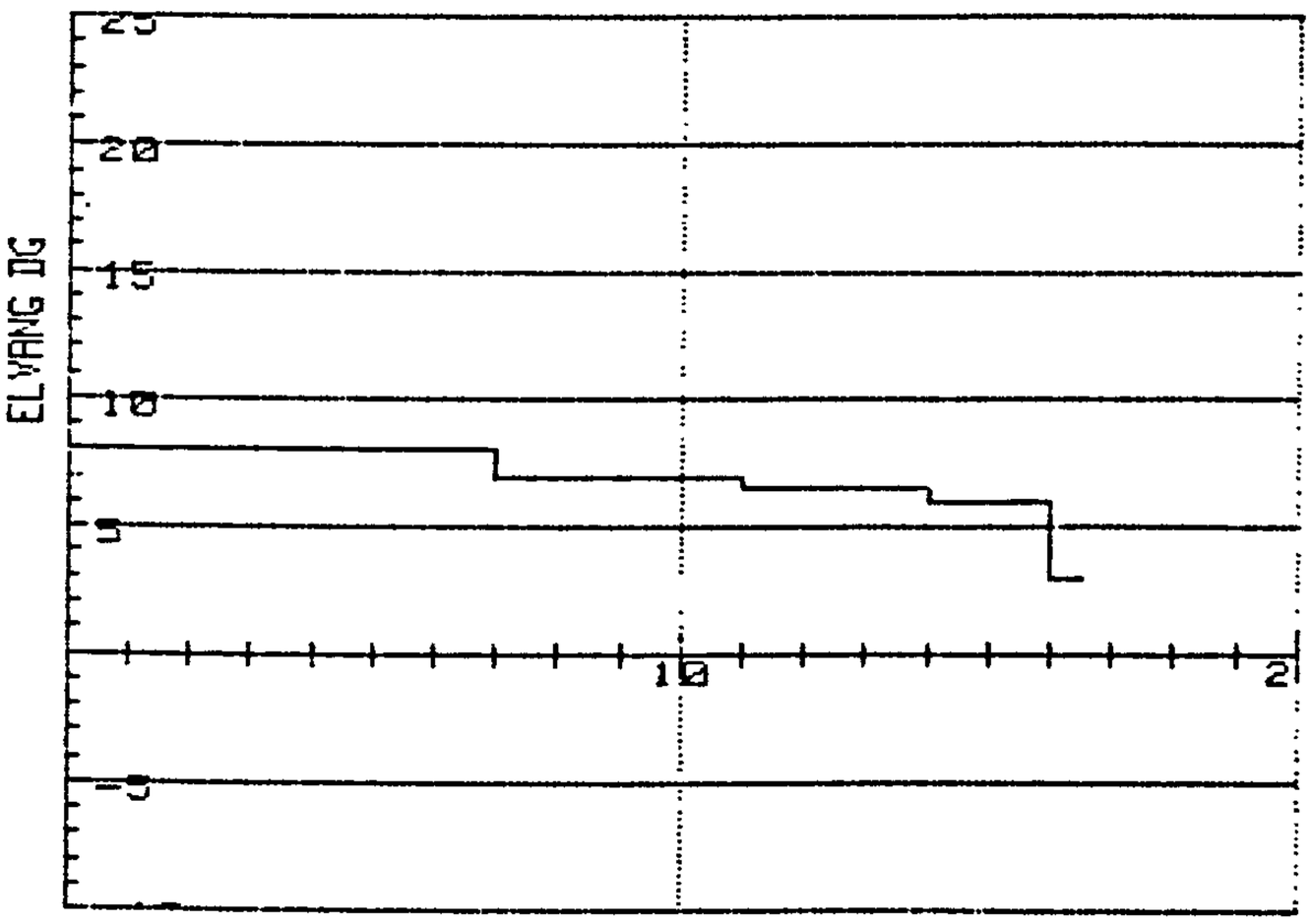
Fig. 4.4 Landing Trajectories

Fig. 4.4 cont.

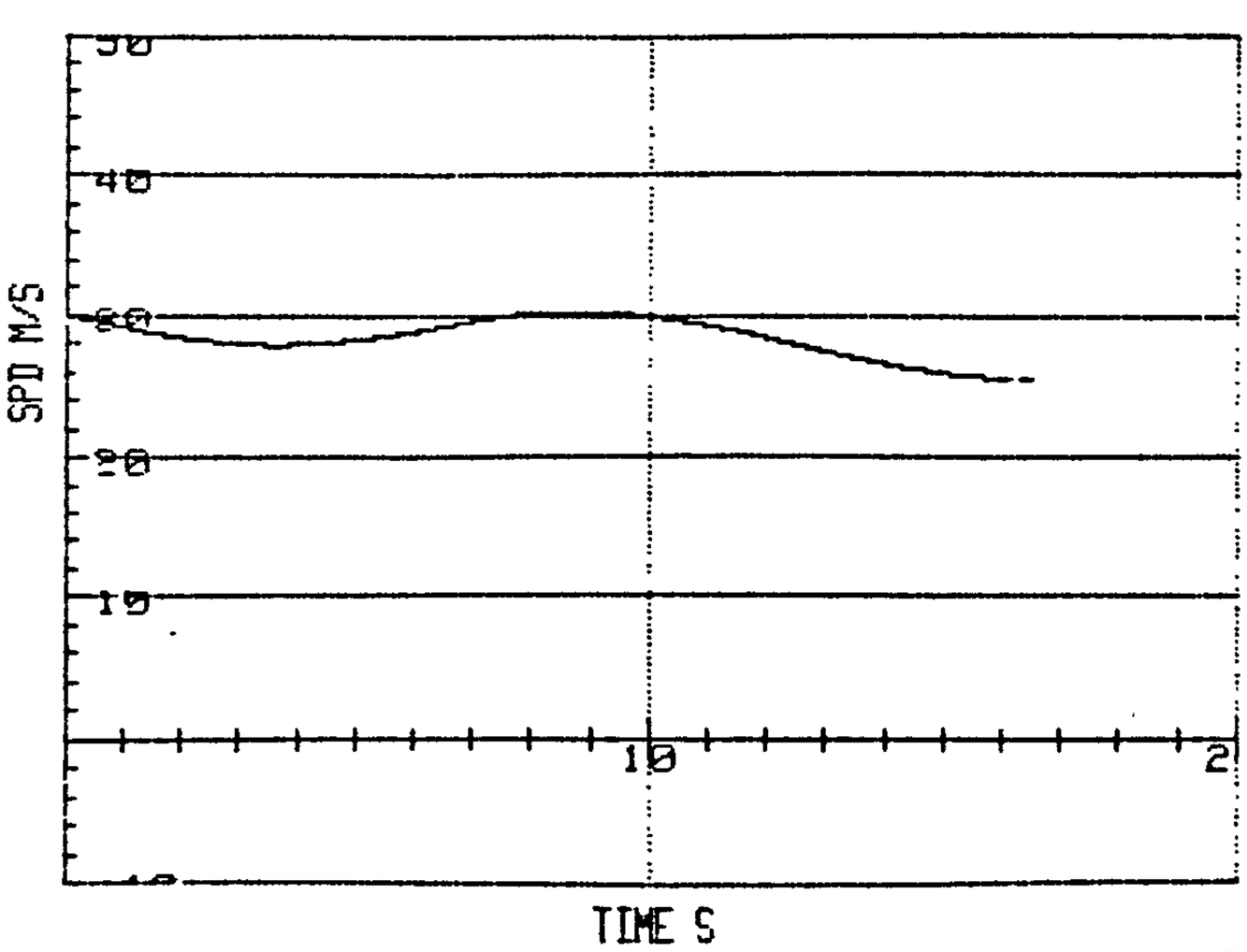


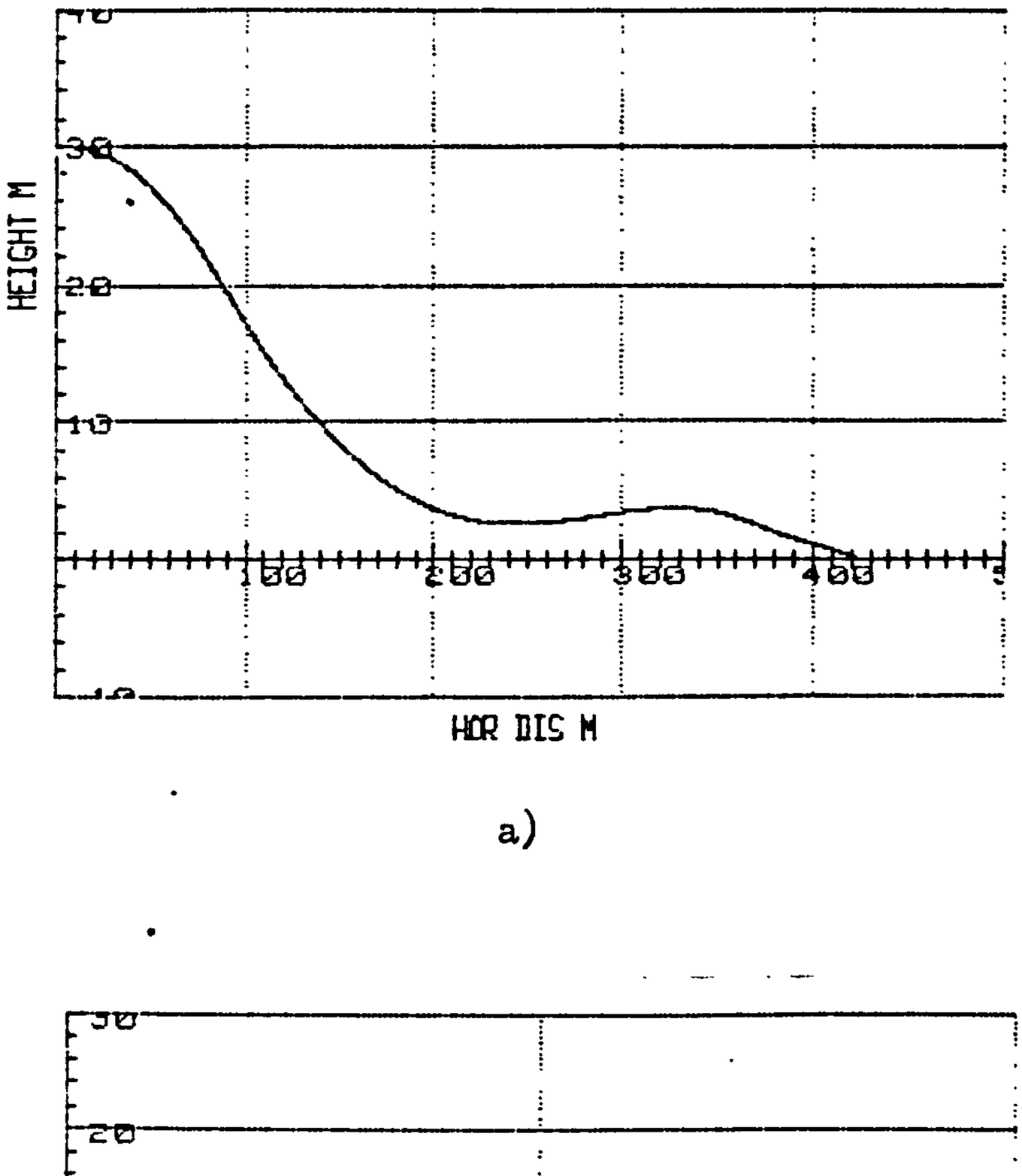
d)

e)



f)



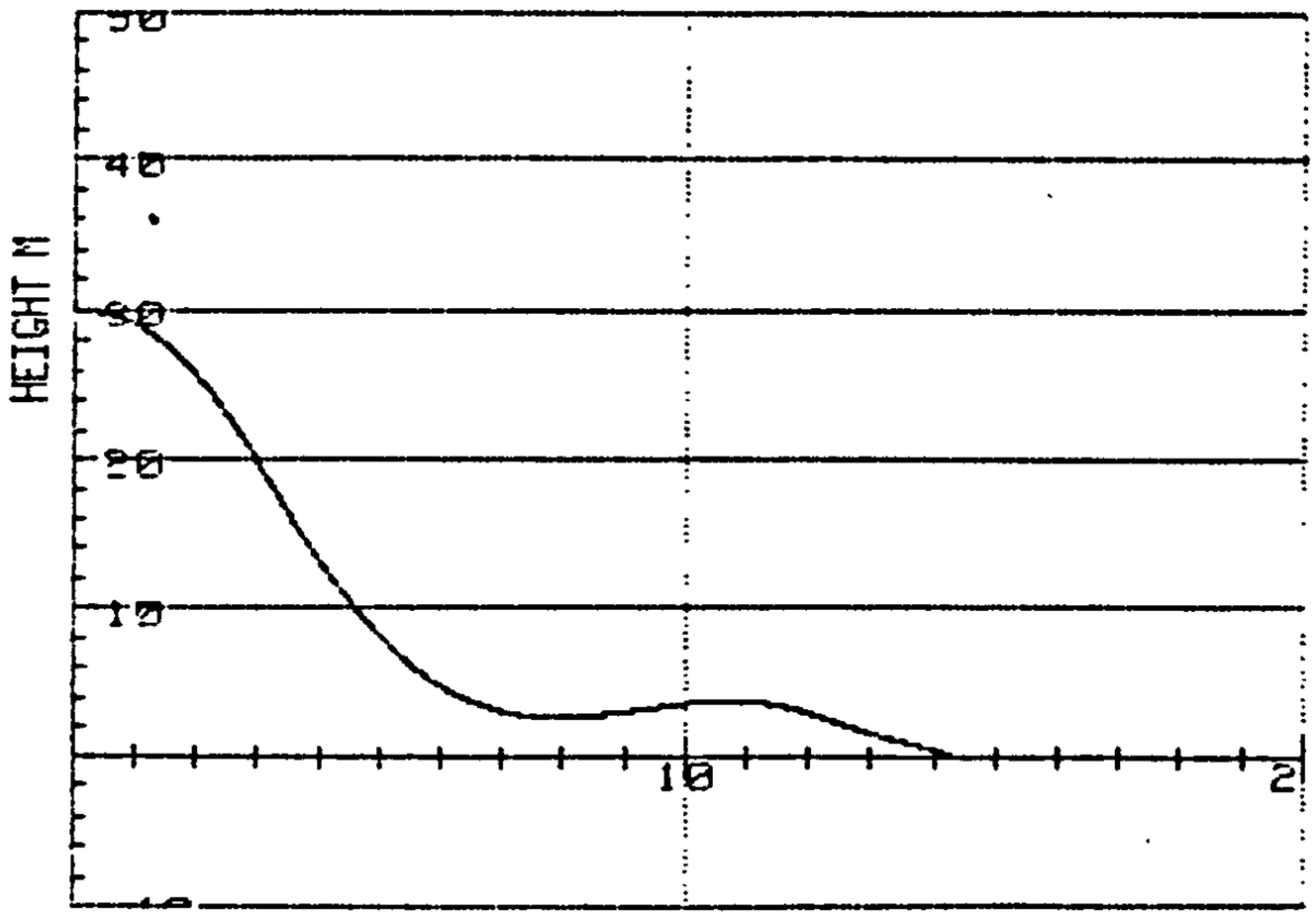


c)

Fig. 4.5

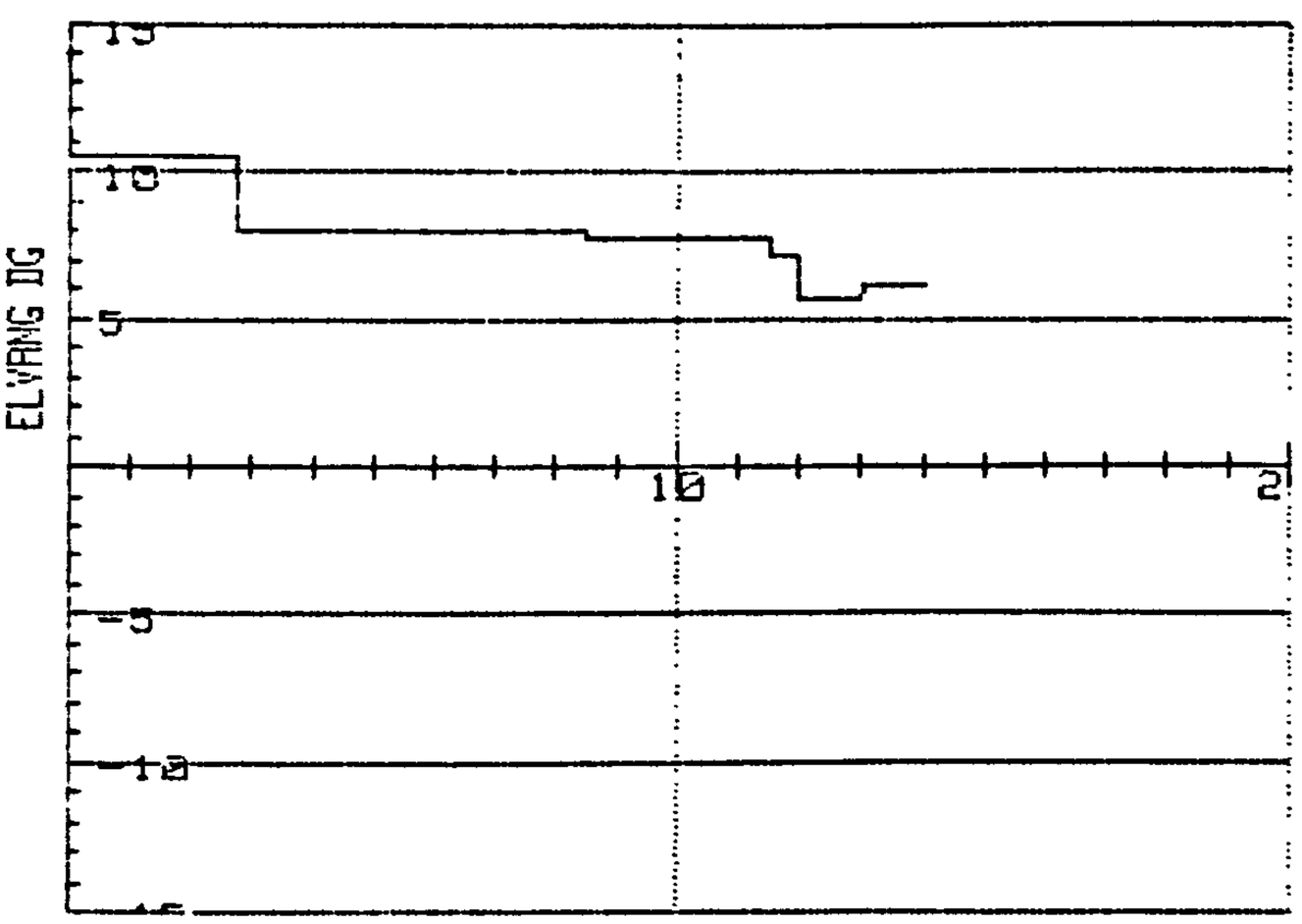
Landing Trajectories

Fig. 4.5 cont.

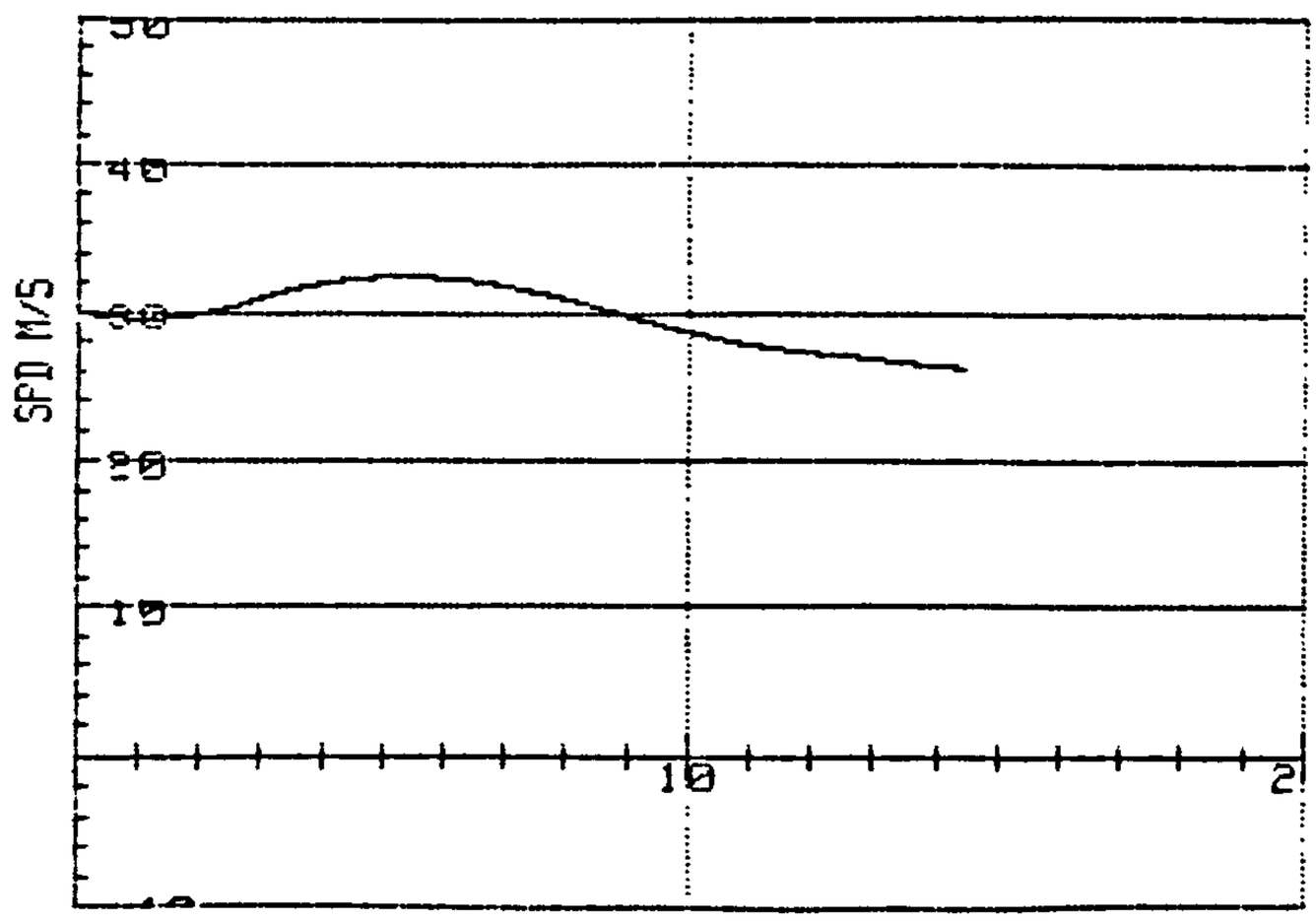


d)

e)



f)



CHAPTER 5

Closed-Loop System Design - Classical approach

5.1 Introduction

Thus far the development and implementation of an open-loop simulation of an aircraft has been considered. This simulation has been shown, in Chapter 4, to allow pilotic workload to be assessed under various manoeuvres. It was also demonstrated that the simulation was a valid one by comparing the results obtained with those from remote telemetry on the actual aircraft. Attention must now be turned to the likely requirements of a closed-loop control system when used with the aircraft in addition to surveying the techniques available for such closed-loop control.

In this Chapter 'classical' control systems design techniques are investigated, particularly with regard to differentiating between so called stability augmentation systems (SAS) and autopilot designs. This discussion will then be broadened in Chapters 6 and 7 into a comparison of 'modern' control schemes and their application to flight control systems.

5.2 Classical Control

One of the earliest applications of control systems theory to the aircraft problem was in the provision of stability enhancing or augmenting systems (139,158,169). This type of system was found to be necessary as a result of the increasing demands for high performance aircraft and the consequent reduction of the inherent stabilty of the airframe. Reduced airframe stabilty can impose a heavy burden on the pilot since he may be required to correct a poorly damped or unstable aircraft mode. SAS systems provide improved stabilty by automatically compensating for these poorly damped or unstable modes. In effect this compensation is provided by feeding back aircraft motions into the control surface deflections, possibly via. filters.

Typical of this type of SAS would be a yaw damper which corrects for possibly unstable lateral modes (e.g. spiral divergence) and ultimately assists the pilot in executing a turn, for example. A requirement also developed for attitude hold systems which essentially act as autopilots by maintaining say a given pitch attitude or heading (yaw) angle. Whilst the requirements for both systems are similar, it is often the case that the SAS design is undertaken as an inner-loop design whilst the autopilot conventionally acts as an outer-loop controller. The trend today is towards a more unified approach with a single unit providing SAS and autopilot functions, the design then being undertaken as a 'single' step process. (This will be discussed in more detail in Chapters 6 and 7). It is also worth noting that a simple SAS system will consider the pilot's actions as a disturbance and hence will reject these. The autopilot is normally arranged so as to allow the pilot to interact with the aircraft by setting required values of heading, etc. Having thus identified the desirability of some form of automatic flight control, from the pilot's viewpoint, it is important to establish what performance criteria can be applied to the design of flight controllers and what a pilot might expect from such systems.

5.3 Performance Criteria

The aircraft environment is one in which a close interaction normally occurs between the pilot and the dynamic system, i.e. the aircraft. This type of "pilot-in-the-loop" system can impose a heavy burden on the pilot particularly when faced with a poorly behaved aircraft. As mentioned above the addition of an SAS may assist the pilot in his task but may also reduce the 'feel' of the aircraft since pilotic demands will normally be interpreted and executed by the control system. This situation may be alleviated by reducing the authority of the SAS and allowing the pilot more direct contact with the control surfaces. The SAS then only operating at perhaps the extremes of the aircraft's performance envelope. This does however detract from the inherent ability of an SAS to provide markedly

better performance than the pilot in addition to providing additional facilities such as gust alleviation, decoupling of structural modes, etc. The foregoing remarks raise the issue of the role of the pilot in the aircraft system and this has been the subject of much work (11,12,13). If a somewhat more pragmatic view is adopted and the likely requirements of a 'stand-alone' flight controller are considered then a number of points become clear.

i) The aircraft is a complex dynamical system, additionally, the parameters of the dynamic equations of motion change as a result of changes in flight condition. Any control system must be able to cope with these changes without an undue difference in the aircraft 'feel'.

ii) The system must operate successfully within the constraints imposed by the aircraft environment. Control surface demands, for example, must be within the capabilities of the servo-actuators and must have sufficient freedom to allow for additional attitude command signals. The limited measurements available must also be borne in mind with regard to the achievable system performance.

iii) Any automatic system is capable of failure and hence the pilot must be able to effect a successful recovery from potentially dangerous system failures. Alternatively the system must be made highly reliable by using analytical or hardware redundancy.

iv) Along with iii) the degree of overide provided for the pilot is important as is the authority given to the automatic system in manoeuvring and 'stick-fixed' regimes.

v) A somewhat more detailed aspect is the actual hardware and, increasingly today, the software implementation and display systems provided for the pilot.

Whilst the avionics system designer must be aware of the above factors and will, ultimately, be influenced by them, the principal function of the control system will be

to provide adequate 'performance' from the overall system. How we define 'performance' will, to some extent, depend upon the designer and his view of the system and the constraints imposed by the specification. Traditionally, flight control systems have been developed using servo-mechanism theory and the analytical tools of frequency response plots (Bode diagrams) and s-domain plots (root-loci) and hence traditionally derived figures of merit are applied to the design, for example, peak-overshoot, 10-90% rise-time, phase margin, gain margin, etc. These techniques generally rely upon single-loop design, the loop being formed between a measurable output variable, which it is desired to control, and a control surface deflection. Where interaction between loops occurs this may be removed by introducing appropriate compensation into the loop or by viewing the cross-coupling as a disturbance.

The effects of the changing dynamics of the aircraft over the flight envelope are normally accounted for by using root-locus techniques and investigating the likely effect of the movement of the poles and zeros of the appropriate transfer function in terms of the effects on the locus. If potentially unstable or unsuitable regions are identified then the system closed-loop gain, for example, may be changed by linking this to say the current airspeed or air density. What is effectively being done here is to modify some of the aerodynamic derivative terms, which determine the pole/zero locations, in such a way as to retain stability of the closed-loop system (whence stability augmentation). It is therefore quite possible to adopt these traditional performance criteria and ensure that, for example, damping of the aircraft Dutch roll mode is adequate over all possible flight conditions. This does not of course guarantee that the pilot will favour the feel of such a control law but the system may be tuned 'on-line' by flight trials.

With the advent of modern control techniques and their application to flight control we may be led to investigate the applicable performance figures of merit when using these systems. The model reference technique, for example,

requires that the system performs in an arbitrarily close correspondence to an idealised model (14-17). In later chapters it will be indicated how the techniques to be discussed perform initially with regard to traditional figures of merit. It may be useful, therefore, to investigate a typical control system design for our particular application i.e. a remotely piloted vehicle.

5.4 Requirements

The requirements for a control system for a remotely piloted vehicle are, to some extent, unique in that the aircraft may spend a considerable amount of time flying with no pilotic intervention. This type of situation is typical, for example, when blind flying via. a radar table. The automatic control system must thus be capable of implementing long term attitude control, e.g. height hold, in addition to well controlled manoeuvre demand systems, e.g. turn rate and pitch rate control. The performance of the automatic control must thus be quite stringent since the 'pilot' will have little direct feedback of the aircraft attitude. There are also constraints placed upon the measurement devices carried on-board. Normally a rate gyro pack is carried but no positional data is available since no accelerometers are carried on-board. The rate gyro data is also of rather poor quality hence simple integration to give positional data is subject to long term drift. The use of flight control quality rate gyro's does however provide a relatively inexpensive flight control system implementation.

The Machan typically requires the following manoeuvre and attitude hold systems.

i) Pitch rate control
ii) Roll rate (turn rate) control
iii) Height hold
iv) Yaw damping

These systems must produce adequate performance in terms of responsiveness and stability. By way of an example of controller design consider cases i), ii) and iv) above.

5.4.1 Pitch Rate Controller

The aircraft pitch attitude is controlled directly by elevator deflections and the open-loop response is largely determined by the longitudinal set of equations derived in Chapter 2. The pilot would normally demand a given pitch attitude is set up by forward or backward movement of the stick. The control system then establishes the elevator deflection required. As indicated earlier a very powerful technique for control system synthesis is the single-loop feedback type of structure as in Fig 5.1.

The design of the compensator element is of interest since it is this which largely determines the closed loop response of the system as shown. One fairly simple approach which has been shown to be acceptable in practice is to make $G_c(s)$ a purely proportional gain K_{qc}, hence

$$\frac{\eta_d}{q_E} = K_{qc} \text{ deg/deg/s}$$

$$\text{and } q_e = (q_d - q)$$

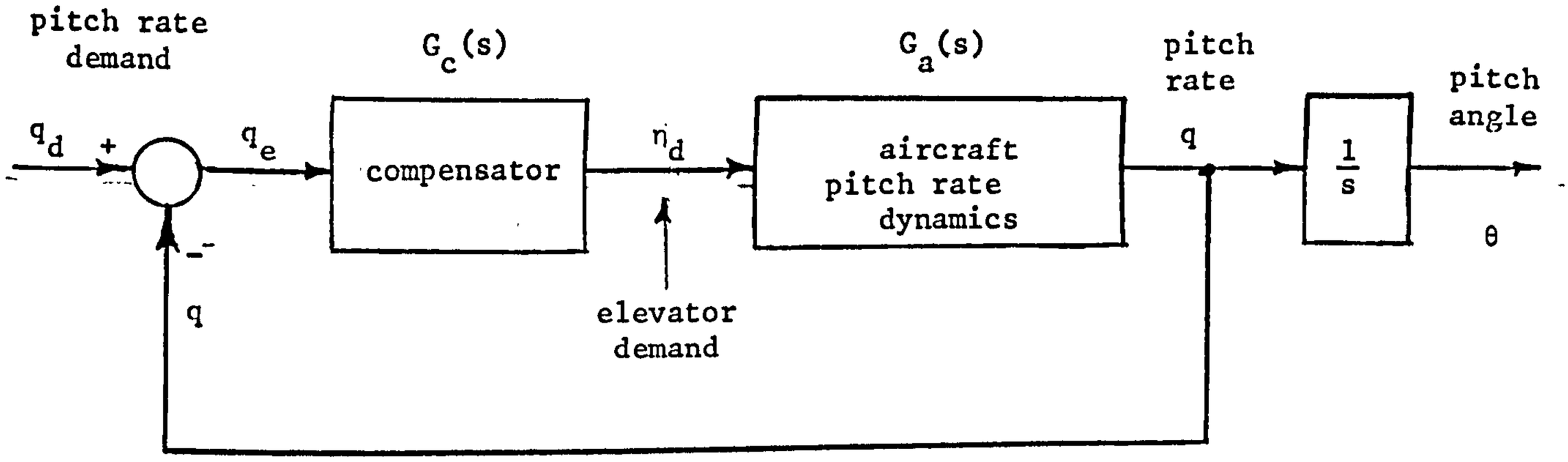
In this case q_d is taken to be the pitch rate demand from the autopilot system. The pilot's demand would normally have higher authority and be capable of overiding the autopilot. Taking the above as a basis for analysis some idea of the $G_a(s)$ transfer function is required. This may be derived directly from equations 2.36 by applying

$$G_a(s) = C'_{lo}(sI - A'_{lo})^{-1}B'_{lo}$$

with the modified matrices at $33ms^{-1}$ airspeed

$$A'_{lo} = \begin{bmatrix} -0.059 & 0.147 & 0. & -9.81 \\ -0.475 & -2.93 & 32.77 & 0. \\ 0.166 & -0.416 & -0.645 & 0. \\ 0. & 0. & 1. & 0. \end{bmatrix}$$

Fig 5.1 Archetypal Pitch Rate Controller



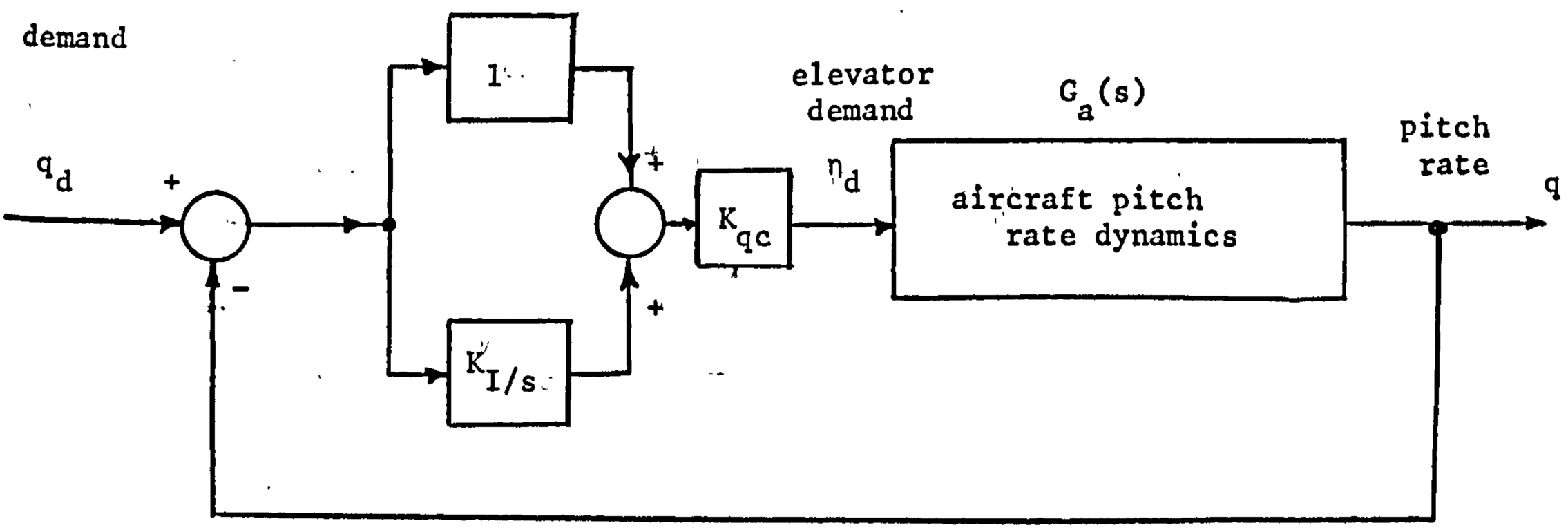


Fig 5.4 P+I Controller for Pitch Rate Autostabiliser

$$B'_{lo} = \begin{bmatrix} 0. \\ 5.318 \\ -13.58 \\ 0. \end{bmatrix}$$

$$C'_{lo} = [\ 0. \quad 0. \quad 1. \quad 0.\ ] \qquad -5.1$$

ignoring the height integration and thrust equations and any actuator dynamics and taking the state vector as :

$$\begin{aligned} u &\rightarrow x_1 \\ w &\rightarrow x_2 \\ q &\rightarrow x_3 \\ \theta &\rightarrow x_4 \end{aligned}$$

The input is η , the elevator deflection. Thus :

$$G_a(s) = \frac{K_q\ s(s + 1/T_1)(s + 1/T_2)}{(s^2 + 2\ \zeta_s\ \omega_s s + \omega_s{}^2)(s^2 + 2\ \zeta_p \omega_p s + \omega_p{}^2)} \qquad -5.2$$

Note that the fourth order dynamics give rise to the two complex pole pairs, the short period and the phugoid and three real zeros at s=0, $s=-1/T_1$ and $s=-1/T_2$. The zero at s=0 is a natural consequence of the derivative action required between elevator demands (degrees) and pitch rate (degrees/sec). The two remaining zeros for the Machan are at :

$$1/T_1 = 0.08 \text{ and } 1/T_2 = 3.07$$

and K_q = 13.58 deg/deg/s

for a stick-fixed configuration at an airspeed, V_T, of $33 ms^{-1}$.

A root locus plot for the above system with the loop closed with unity negative feedback and a proportional forward path controller is shown in Fig 5.2. Note from this that :-

i) The phugoid poles move towards the real zeros at s=0 and $s=-1/T_1$. For very large gains the damping of the phugoid mode can be made large however, this may be undesirable since it may lead to poor short period response. For the relatively low gain cases normally considered the phugoid mode is little affected.

ii) The short period poles move onto the real axis for relatively low values of gain. To maintain short period response, therefore, requires the use of relatively low gain systems since the short period motion will normally be required to have a specified damping and natural frequency.

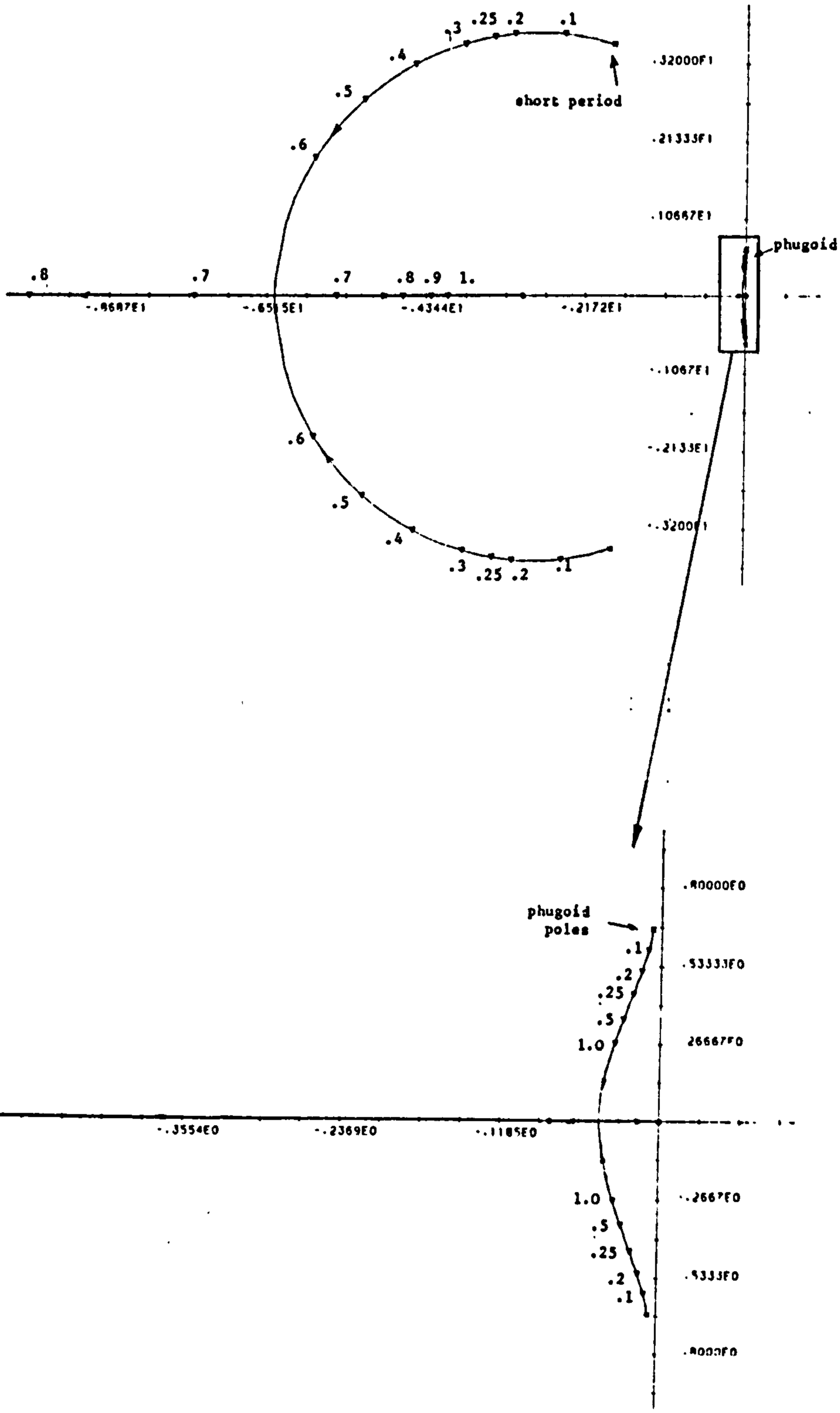
The simple proportional pitch rate control law can thus be seen to have a desirable effect on the damping of the short period dynamics but little effect on the phugoid mode for low proportional gains. Using this type of feedback system is analagous to increasing the m_q stability derivative term and from equation 2.39 note that this will directly affect the damping of the short period motions since :

$$2\,\zeta_s\,\omega_s = -(z_w + m_q)$$

A choice of $K_{qc} = 0.25$ gives approximately critical damping of the short period mode, as shown in Fig 5.2. A typical system step response for a demanded pitch rate of 1 degree/sec is shown in Fig 5.3 and from this note that whilst the short period dynamic is well damped, the phugoid mode is still evident as a long term oscillation about the steady state. Another disadvantage of this controller is that the steady state value of pitch rate is zero due to the zero at s=0. Since it is normally required to have attitude control, i.e. pitch angle control, this configuration would seem undesirable since no feedback of pitch attitude is used and no direct control is thus available over this variable.

It is normal to provide feedback of both q and θ either directly or by the introduction of integral action into the controller as shown in Fig 5.4. The P+I controller introduces a pole at s=0 and a zero at $s=-K_I$. This modifies the root locus for the system so that the phugoid poles now

Fig 5.2 Root Loci - Pitch Rate Loop



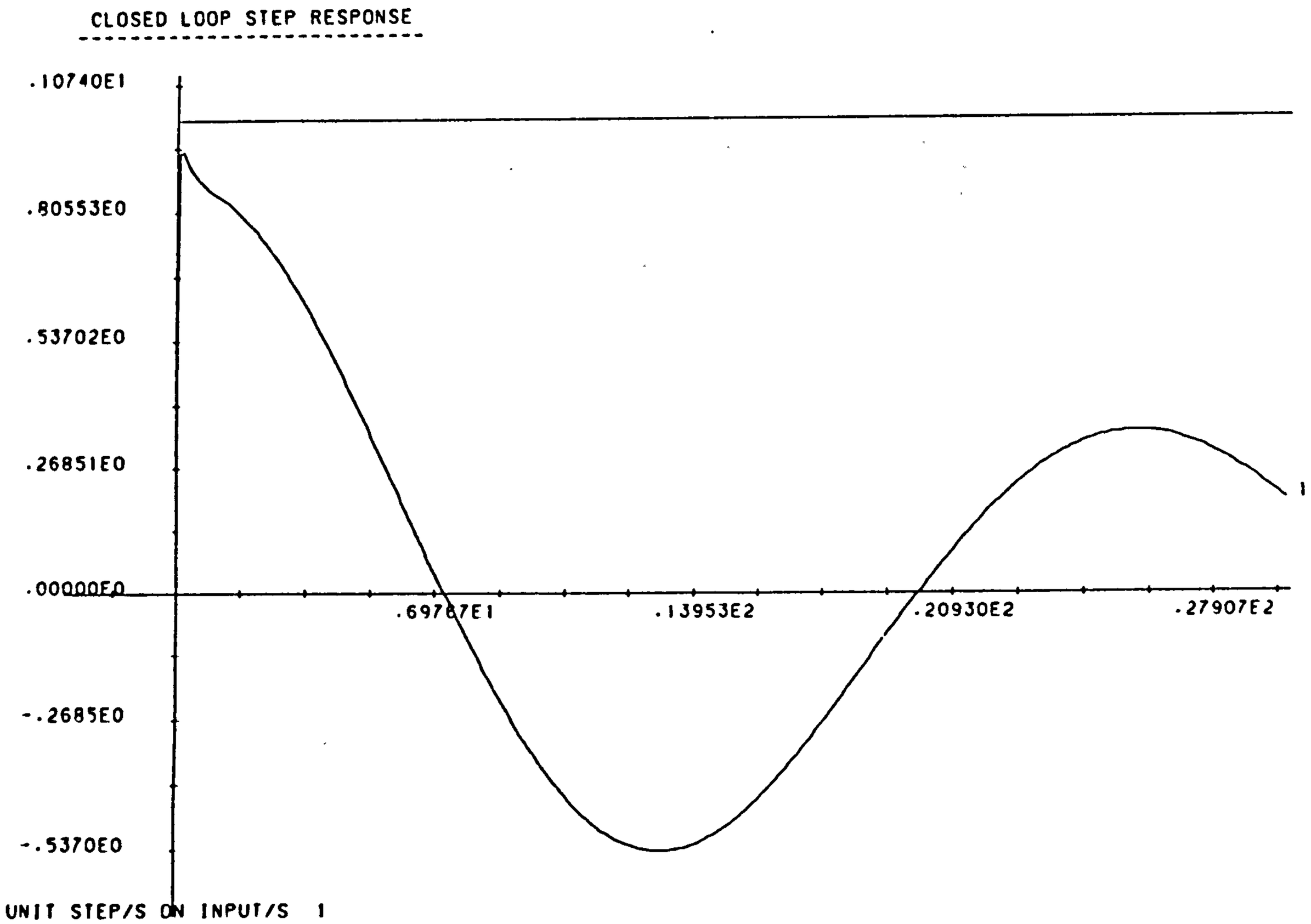


Fig 5.3 Pitch Rate Response

move rapidly to the two zeros at $s=-1/T_1$ and $s=-1/T_2$ as shown in Fig 5.5. The zero at s=0 is cancelled by the pole from the controller. Fig 5.5 has $1/K_I=0.1$ and for moderately low values of gain the phugoid mode may be made well damped. The short period mode can also be arranged to have a reasonable degree of damping.

Typical system step responses for a closed-loop gain of 1.0 are shown in Fig. 5.6 and the suppression of the phugoid mode is clear along with the rapid decay of the short period mode. In the steady state the pitch rate approaches the demanded value although a slight offset is still evident.

The pitch rate controller described above is typical of an inner-loop stability augmentation system and provides acceptable performance from the pitch dynamics. An autopilot system could now be used to provide the q_d signal, the magnitude of which may be made proportional to the error in pitch attitude, for example.

Now consider the effects on the system designed above of a change in say the airspeed of the aircraft. For 'stick-fixed' operation at $50ms^{-1}$ the aerodynamic derivatives will change and as a result the A'_{lo} and B'_{lo} matrices of equations 5.1 will be modified. Evaluating these matrices gives :-

$$A'_{lo} = \begin{bmatrix} -0.0965 & 0.2685 & 0. & -9.81 \\ -0.8664 & -3.81 & 49.7 & 0. \\ -0.531 & -0.503 & -0.9387 & 0. \\ 0. & 0. & 1. & 0. \end{bmatrix}$$

$$B'_{lo} = \begin{bmatrix} 0. \\ 12.2 \\ -52.8 \\ 0. \end{bmatrix}$$

The open-loop pitch rate dynamics of equation 5.2 now give the approximate pole/zero locations

phugoid poles $s = +0.176 \pm 0.716j$

short period poles $s = -2.37 \pm 4.79j$

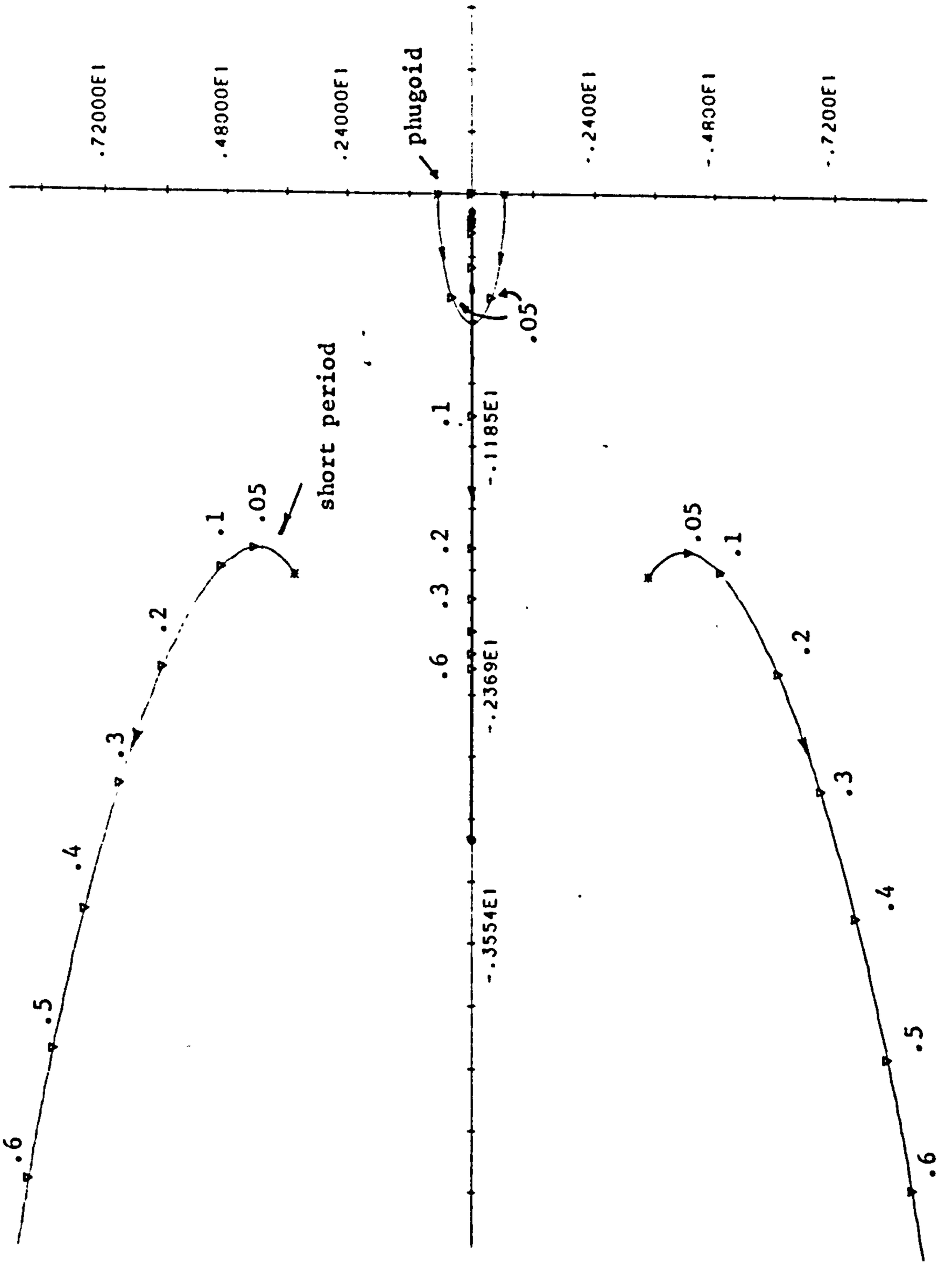


Fig 5.5 Root Locus - P+I Pitch Rate Controller

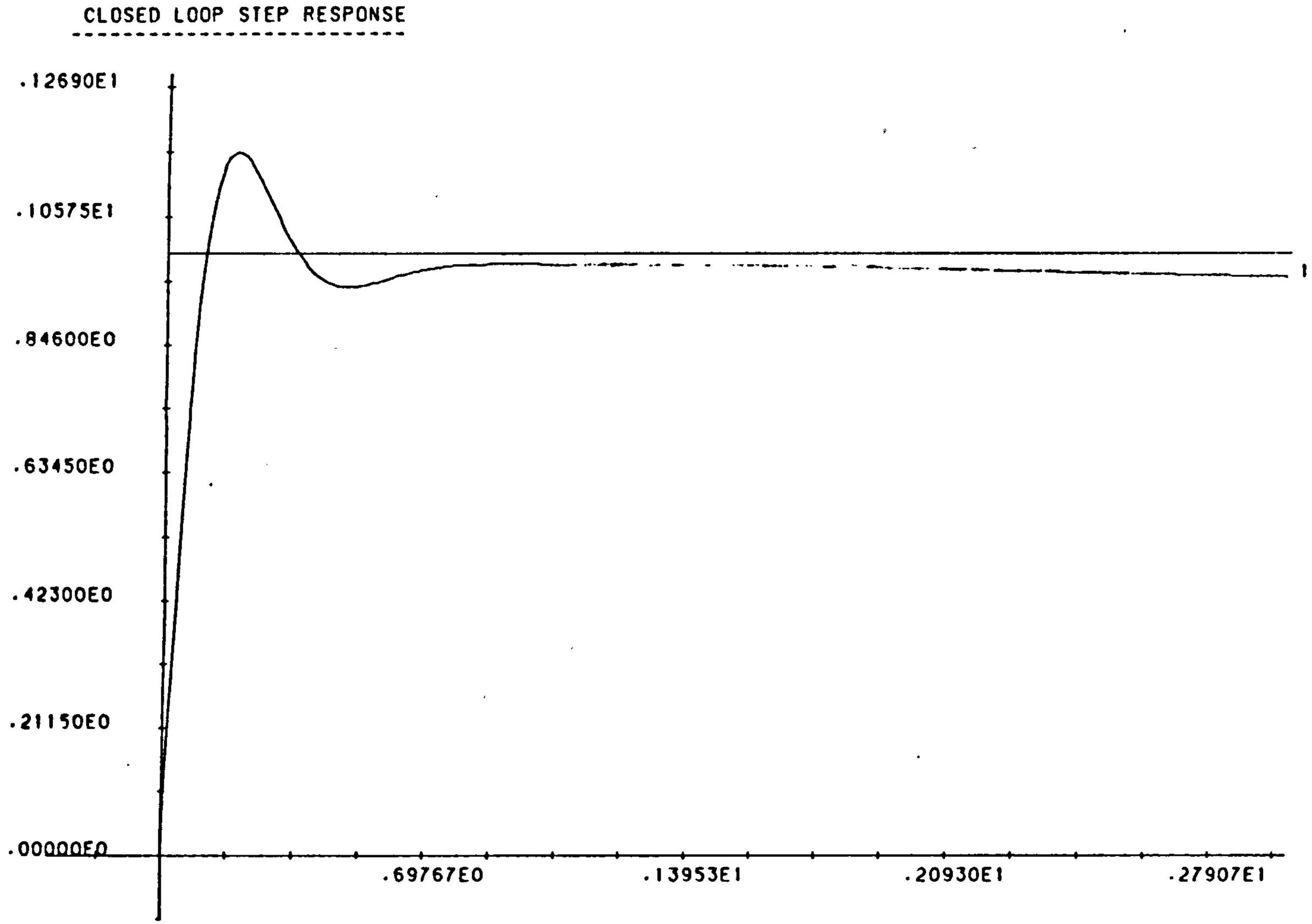


Fig 5.6 Response - P+I Pitch Rate Loop

system zeros $s = 0.0$, $s = -3.849$ and $s = -0.16$

system static gain, $K_q = 52.8$

Note the increased static system gain indicating the increased effectiveness of the elevator at higher airspeeds. Note also that the phugoid mode is now open-loop unstable and hence some form of compensation is needed.

By employing the P+I pitch rate controller described above a closed-loop system which retains reasonable damping of the phugoid and the short period modes is obtained. The dynamic response of the system is, however, somewhat different from the $33ms^{-1}$ case as shown in by the step response of Fig. 5.7. To regain a similar dynamic response, a simple change in the controller gain could be made which would provide a modified response. Over the entire flight envelope a gain vs. airspeed curve could be defined which gave similar performance as the aerodynamic derivatives changed in value. Such gain-scheduling schemes are commonly employed to provide adequate system stability and performance.

A secondary consideration of such pitch rate autopilots is vertical gust alleviation. Since gusts will normally be considered to be a system disturbance, the closed-loop pitch autopilot will, for a high gain, tend to reject any gust disturbances and hence render the pitch attitude insensitive to gusts. This, however, is not sufficient to ensure that the height, for example, will not be affected adversely by vertical gusts. In manoeuvres requiring tight control of the vehicle's height a pitch rate controller can do little but improve short period and phugoid damping and an outer-loop closure of angle of attack or normal acceleration to elevator will be required (see section 5.4.3)

It is normal to include a short time constant filter in the pitch rate loop in order to remove any noise due to the pitch rate gyro, in addition, the elevator gain and transfer function may be included giving the complete block diagram of Fig. 5.8.

As stated above pitch attitude/rate autopilots give desirable improvements in short period and phugoid damping

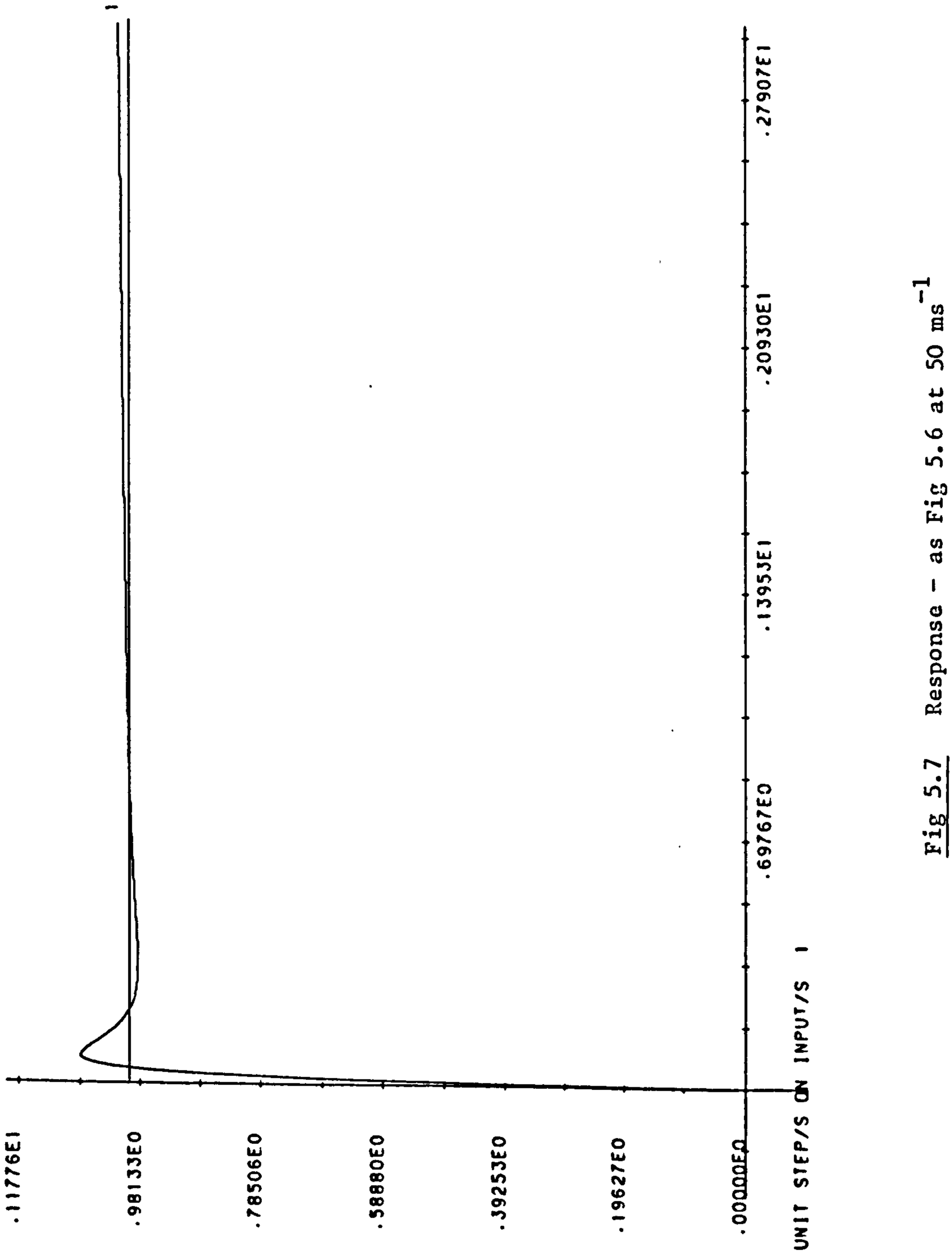


Fig 5.7 Response - as Fig 5.6 at 50 ms^{-1}

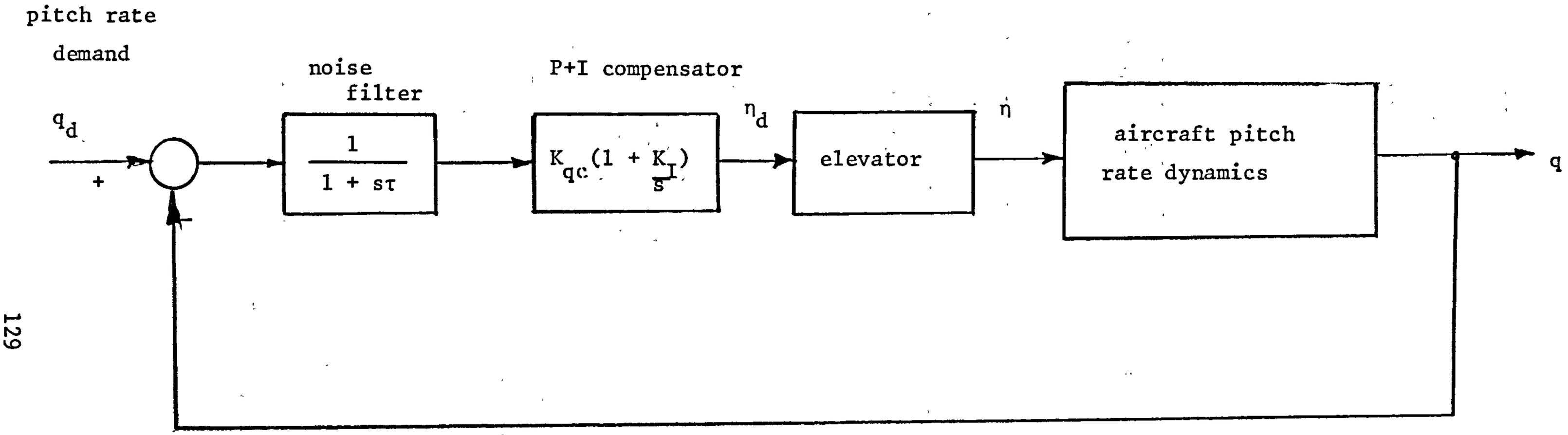


Fig 5.8 Overall Pitch Rate Autostabiliser Block Diagram

in addition to providing improved pitch 'stiffness' and normally form an integral part of an inner-loop SAS design.

5.4.2 Lateral SAS Systems

Aircraft lateral motions consist primarily of yawing (about the OZ axis) and rolling (about the OX axis). Motions about these axes give rise to changes in the aircraft's heading and bank angles. In executing a turn manoeuvre, for example, the aircraft is banked into the turn by aileron deflection and the ensuing sideslip gives rise to a yawing moment which turns the aircraft about the OZ axis onto the new heading. The aircraft is then returned to neutral or zero bank angle flight. The rudder may also be used to assist the turn but is often used as a stabilising surface to counteract the possibly unstable lateral modes, principally the spiral mode, and hence improve directional stability. This so called turn co-ordination system is normally provided automatically. Two control surface deflections must thus be considered in addition to the two motions, roll and yaw. Attention is given first to the control of bank angle.

5.4.2.1 Roll Control

A tight and well controlled roll attitude loop is required in order to provide adequate response in turning manoeuvres. To this end it is normal to provide direct feedback of roll attitude to the ailerons. Referring again to the Machan model system the transfer function $\varphi(s) - \zeta(s)$ may be derived by evaluating :

$$G_{la}(s) = C'_{la}(sI - A'_{la})^{-1}B'_{la} \qquad - 5.3$$

using the modified lateral matrices A'_{la}, B'_{la} and C'_{la} from equation 2.43, namely :-

$$A'_{la} = \begin{bmatrix} -0.277 & 0. & -32.9 & 9.81 \\ -0.1033 & -8.525 & 3.75 & 0. \\ 0.3649 & 0. & -0.639 & 0. \\ 0. & 1. & 0. & 0. \end{bmatrix}$$

$$B'_{la} = \begin{bmatrix} 0. \\ -28.64 \\ 0. \\ 0. \end{bmatrix}$$

$$C'_{la} = [\ 0. \quad 0. \quad 0. \quad 1.\ ]$$

- 5.4

for stick fixed flight at 33 ms^{-1} with the state vector :

$$v \rightarrow x_1$$
$$p \rightarrow x_2$$
$$r \rightarrow x_3$$
$$\varphi \rightarrow x_4$$

Note that the above ignores the yaw (ψ) integration, assumes that the rudder angle is zero and the actuator dynamics are neglected. Application of equation 5.3 now gives :

$$\frac{\varphi}{\xi}(s) = \frac{K_\varphi\ (s^2 + 2\zeta_z\ \omega_z s + \omega_z{}^2)}{(s + 1/T_S)(s + 1/T_R)(s^2 + 2\zeta_d\ \omega_d s + \omega_d{}^2)} \quad - 5.4\ a)$$

Using the results of section 2.2.2 the Dutch roll, spiral and roll subsidence modes have values of :

$$T_R = 0.189 \text{ secs.} \quad T_S = -8.55 \text{ secs.}$$

$$\text{Dutch roll modes} \quad s = -0.5 \pm 3.5j$$

- 5.5

The numerator polynomial has roots :

$$s = -0.458 \pm 3.46j$$

and $K_\varphi = 28.64$

From the above it can be noted that the open-loop Dutch roll poles lie in close proximity to the two zeros. These two quadratic factors thus approximately cancel giving :

$$\frac{\varphi}{\xi}(s) = \frac{K_\varphi}{(s + 1/T_S)(s + 1/T_R)} \quad - 5.6$$

to a good approximation. Note that the spiral mode is

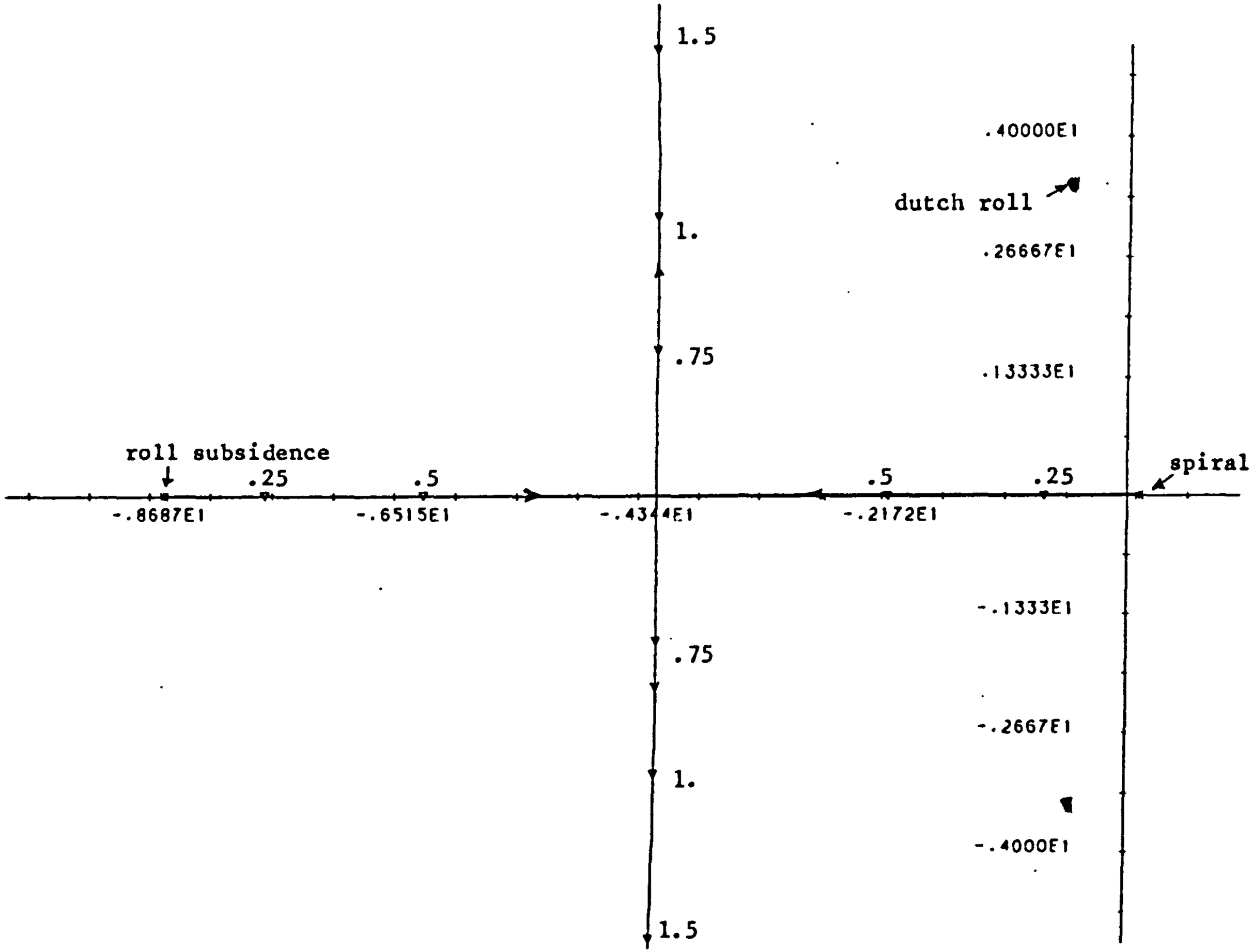


Fig 5.9 Root Locus - Roll Attitude Loop

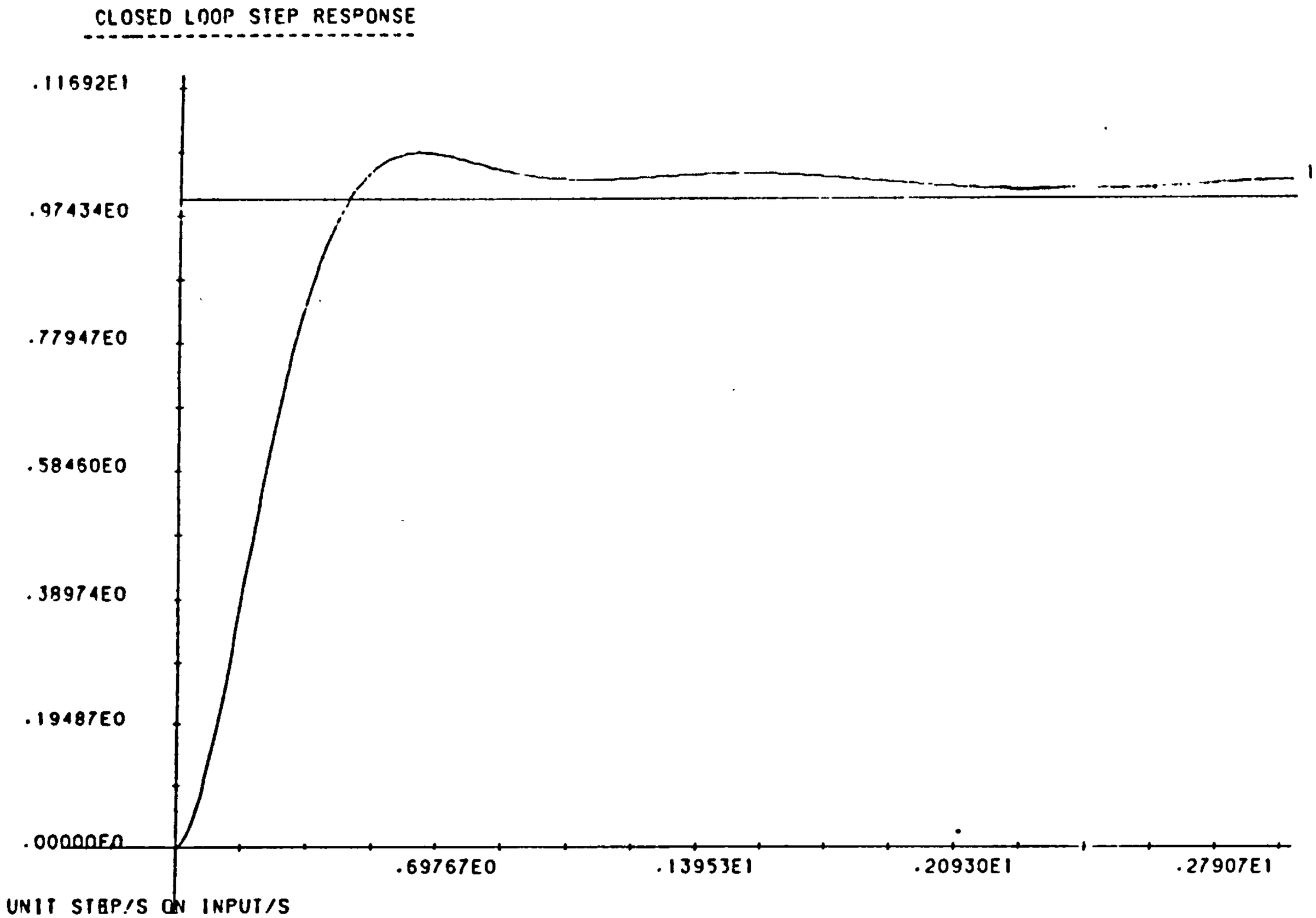


Fig 5.10 Response - Roll Loop

normally close to the origin and for most analyses may be considered a pure integration term giving a simplified dynamic

$$\frac{\varphi}{\xi}(s) = \frac{K_{\varphi}}{s(s + 1/T_R)} \qquad - 5.7$$

The Dutch roll mode may however be excited by gusts or rudder deflections.

A complete root locus diagram for the transfer function of equation 5.4 a) is shown in Fig 5.9. From this note that the Dutch roll poles lie in close proximity to the complex zeros and hence low amplitude Dutch roll motions are to be expected. The two remaining poles at $s=-1/T_R$ and $s=-1/T_S$ form a quadratic pair at reasonably low values of gain. For relatively high values of closed-loop gain a satisfactory performance in terms of roll demand, roll angle may be expected. A typical system response to a unit step demand in roll angle for a closed loop gain of 1.5 is shown in Fig. 5.10. The absence of spiral instability and Dutch roll modes is clear. A purely proportional controller would thus seem to be capable of providing reasonable performance in a roll attitude loop as in Fig. 5.11.

In the above analysis any actuator lags have been ignored and these may, in some circumstances, give rise to less than satisfactory performance. Additionally, the position of the complex zeros of equation 5.4 a) may, for some aircraft and flight conditions, be different from the 'well behaved' aircraft considered. This may lead to poorer Dutch roll damping in addition to possible instability. With this in mind it is normal to introduce rate feedback into the controller such that the controller transfer function becomes

$$G_C(s) = K_{\varphi C} \quad (s + 1/\tau_{\varphi}) \qquad - 5.8$$

Alternatively, integral action may be introduced into a roll-rate SAS system as was done for the pitch rate system of section 5.4.1. This also yields desirable changes in the

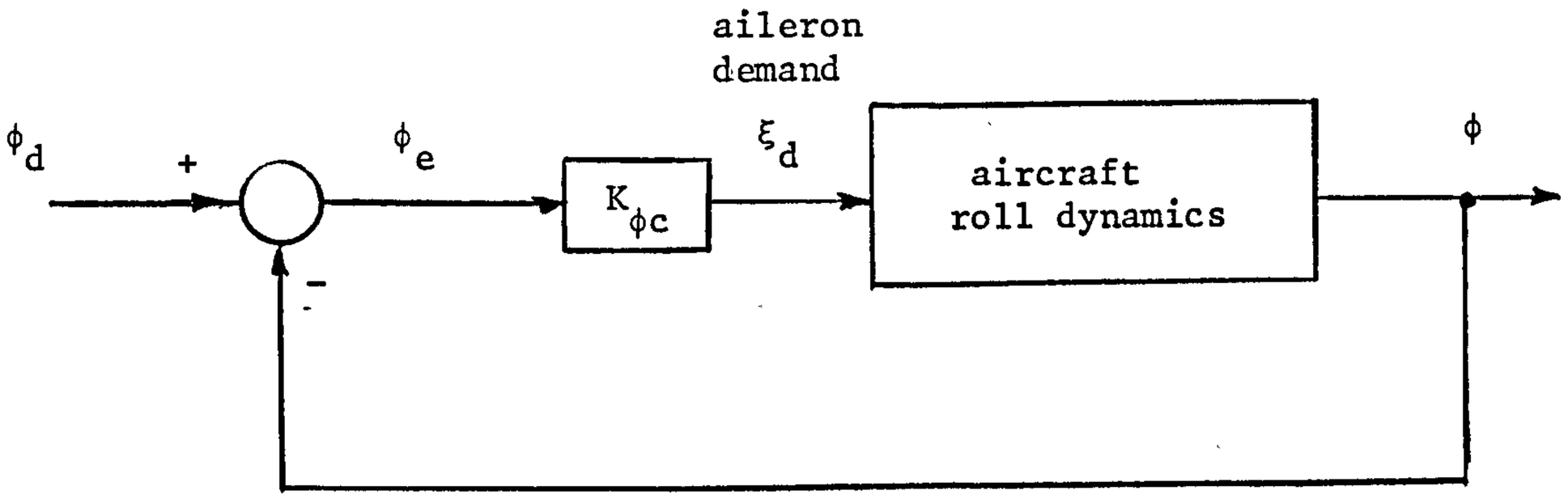


Fig 5.11 Proportional Roll Attitude Loop

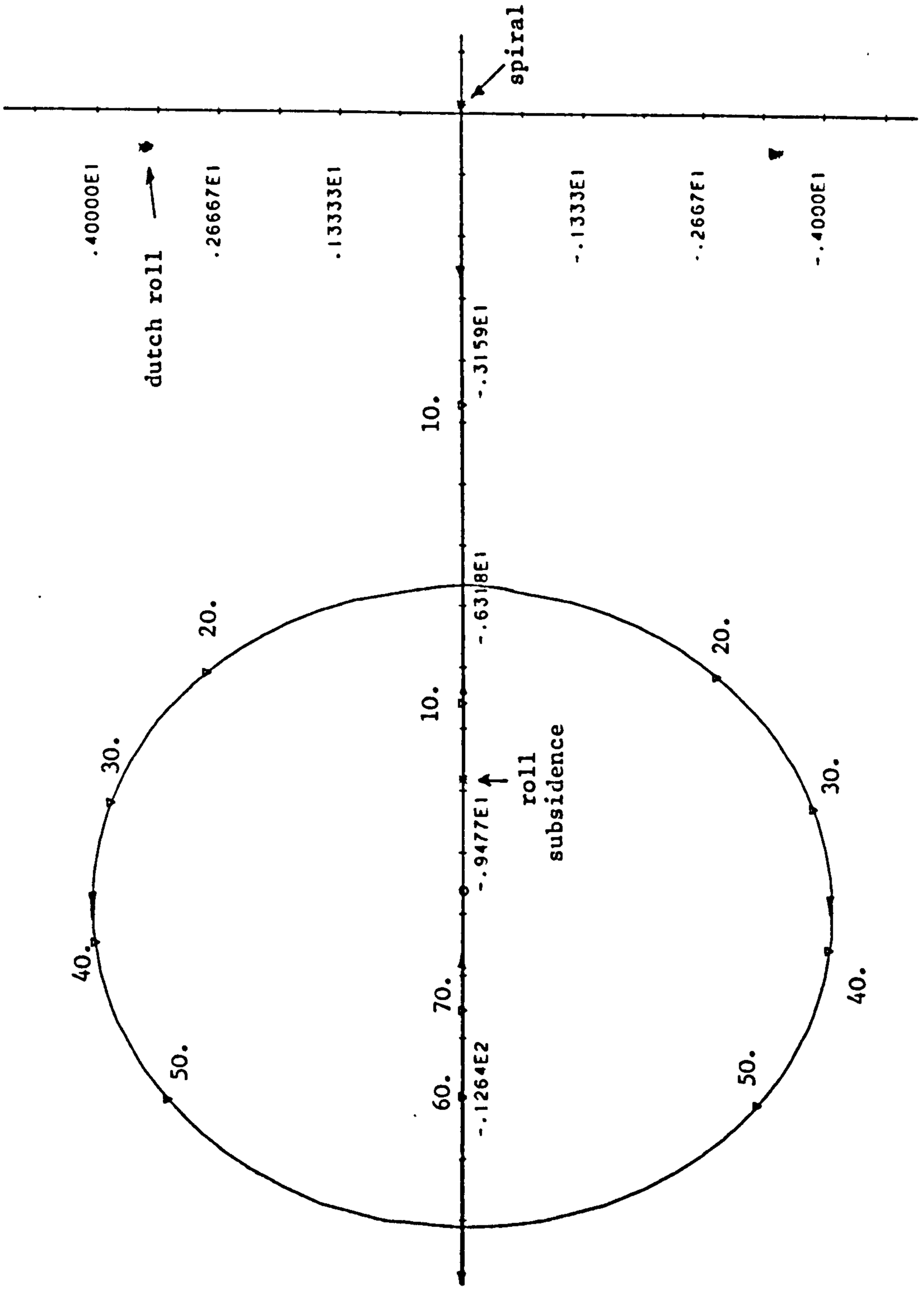


Fig 5.12 Root Locus - Roll Rate Loop

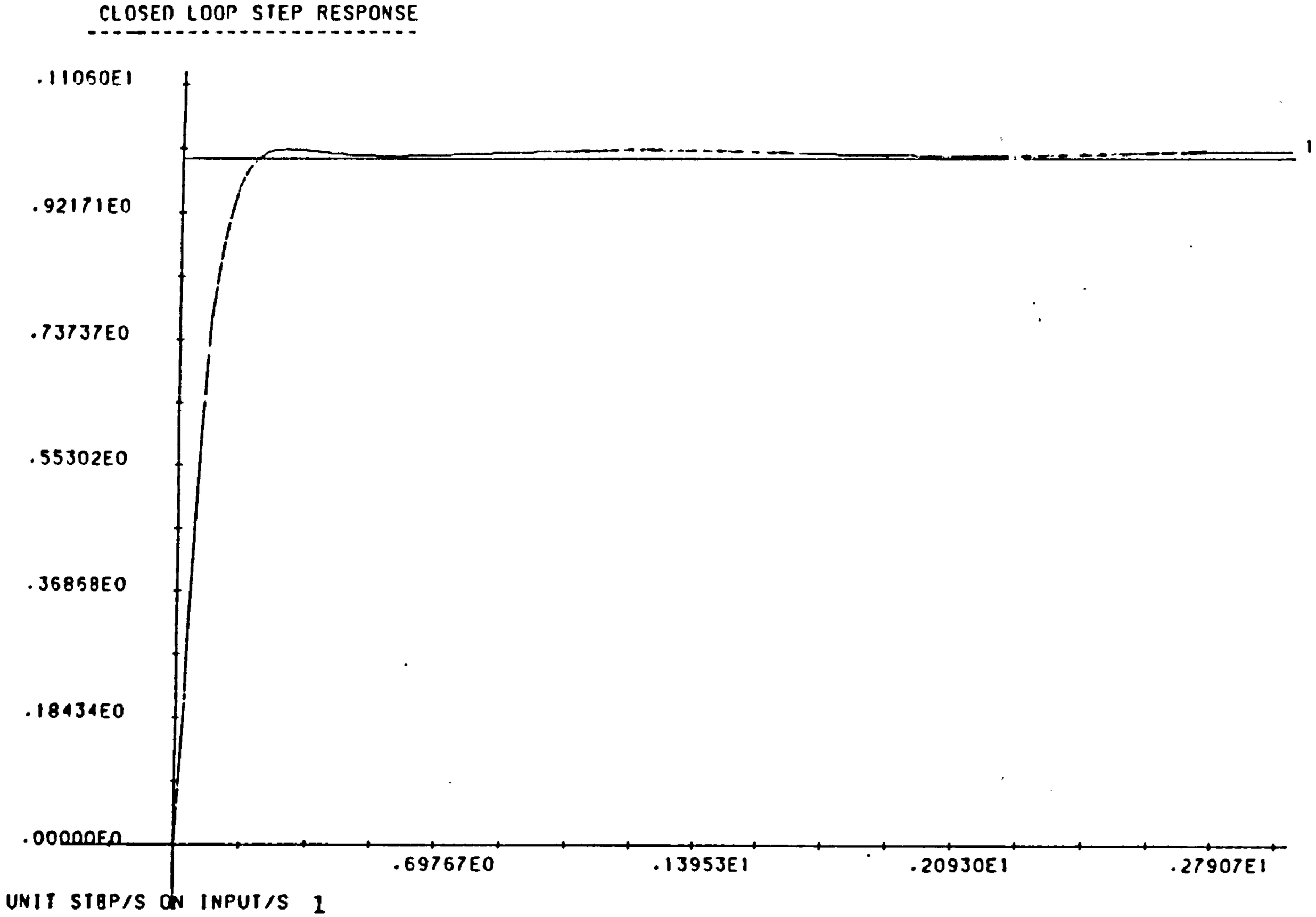


Fig 5.13 Response - Roll Rate Loop

achievable system performance by allowing higher gains to be employed and improving the responsiveness of the loop. Fig. 5.12 shows a complete root locus for the closure of the roll loop with a controller of the form of equation 5.8 with τ_φ =0.1 secs. Note that the spiral and roll subsidence poles now move to the zeros at $-1/\tau_\varphi$ and $-\infty$ and an improvement in system response speed and damping may be expected. This is confirmed by the step response of Fig. 5.13 for a unit step demand in roll attitude and controller gain, $K_{\varphi C} = 50.0$.

The effects on the system of a change in aircraft flight configuration can be demonstrated by considering say a change in airspeed from $33ms^{-1}$ to $50ms^{-1}$. The modified matrices of equation 5.4 now become :-

$$A'_{la} = \begin{bmatrix} -0.419 & 0. & -49.98 & 9.81 \\ -0.1565 & -12.92 & 5.68 & 0. \\ 0.553 & 0. & -0.968 & 0. \\ 0. & 1. & 0. & 0. \end{bmatrix}$$

$$B'_{la} = \begin{bmatrix} -12.47 & 0. \\ 0. & -65.75 \\ -21.62 & 0. \\ 0. & 0. \end{bmatrix}$$

$$C'_{la} = [\ 0. \quad 0. \quad 0. \quad 1.\ ] \qquad - 5.9$$

and the poles and zeros move such that :

T_R = 0.079 secs. : T_S = -12.94 secs.

Dutch roll mode = $-0.723 \pm 5.28j$

zeros at s = $-0.69 \pm 5.25j$

and K_φ = 65.75.

For the proportional roll attitude controller developed above the root locus now becomes as shown in Fig. 5.14. This locus is of the same general form as that of Fig 5.9 for the $33ms^{-1}$ case but the closed loop roots have reduced damping owing to the increased $\zeta - \varphi$ static sensitivity K_φ. This is borne out by the step response of Fig. 5.15 which

exhibits slight overshoot when compared with Fig 5.10. Some form of gain-scheduling would clearly be necessary here in order to preserve the dynamic response of the roll channel as the airspeed changed. Parameterising the gain $K_{\varphi C}$ by 1/U may be possible, for example.

For the case of a lead compensator controller, as in equation 5.8, the roll attitude root locus for $50ms^{-1}$ airspeed is shown in Fig. 5.16. From this note that the Dutch roll poles are almost exactly cancelled by the two zeros, the spiral mode moves to the compensator zero at $s=-10\ s^{-1}$ and the roll subsidence mode moves to the left along the negative real axis. For high gains the roll subsidence mode will dominate the response whilst the spiral and Dutch roll modes are moved close to the compensator and system zeros. For a closed loop gain, $K_{\varphi C} = 50.$, as before, a typical system step response is shown in Fig. 5.17. This shows a very rapid response, compared to Fig. 5.13, owing to the increased roll channel static sensitivity, K_{φ} . The presence of a second order type overshoot comes from the subsidence pole joining with a pole from the lead compensator which has the exact form :

$$G_c(s) = \frac{K_{\varphi C}\ (s + \alpha_1)}{(s + \alpha_2)} \qquad -5.10$$

with $\alpha_1 = 10\ s^{-1}$; $\alpha_2 = 100\ s^{-1}$.

Again a reduction in compensator gain may be desirable in order to maintain the roll channel response. The closure of the roll channel loop does, however, provide stabilisation of the unstable spiral mode for reasonably high loop gains over a range of flight configurations.

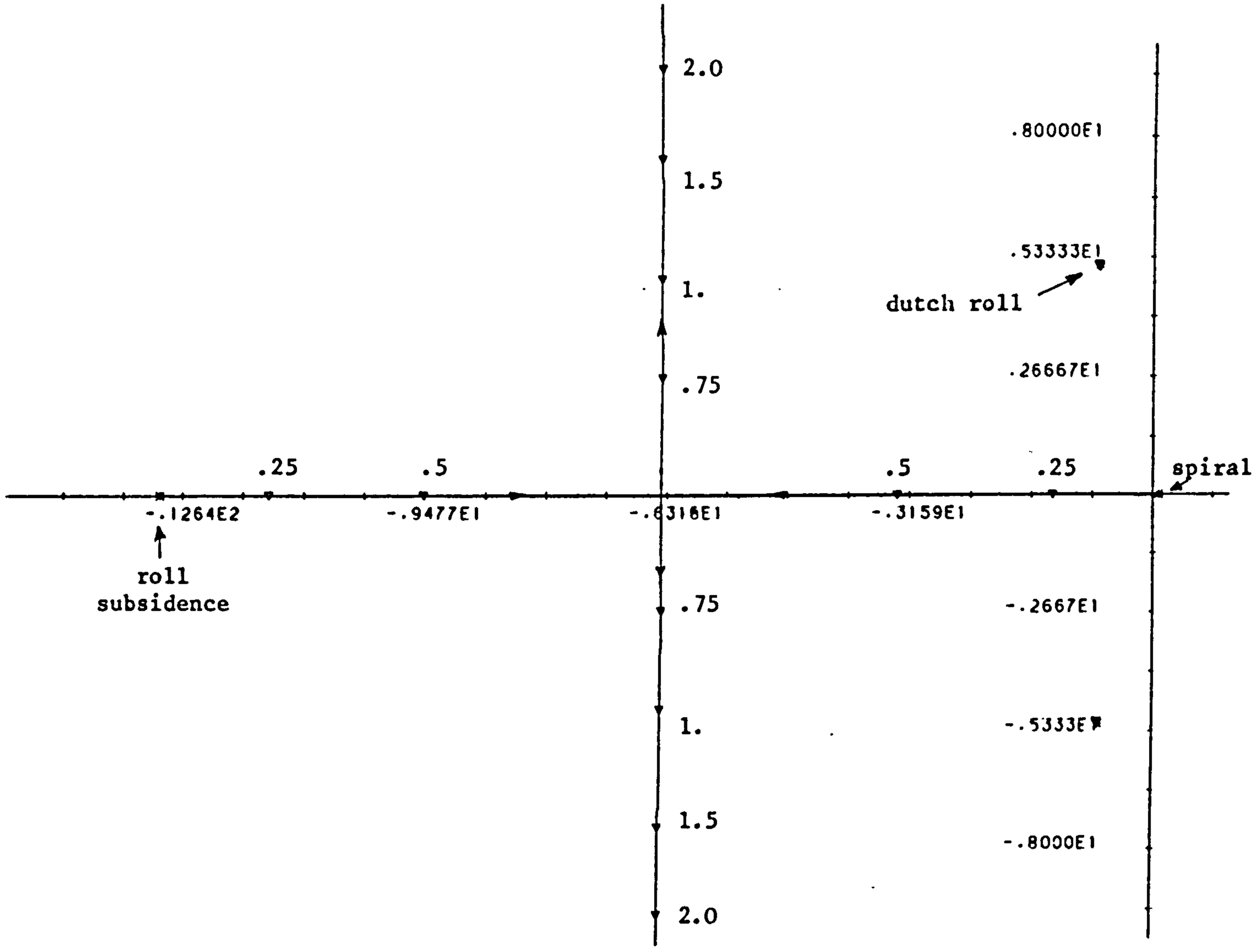


Fig 5.14 Root Locus - Roll Loop, 50 ms^{-1} dynamic

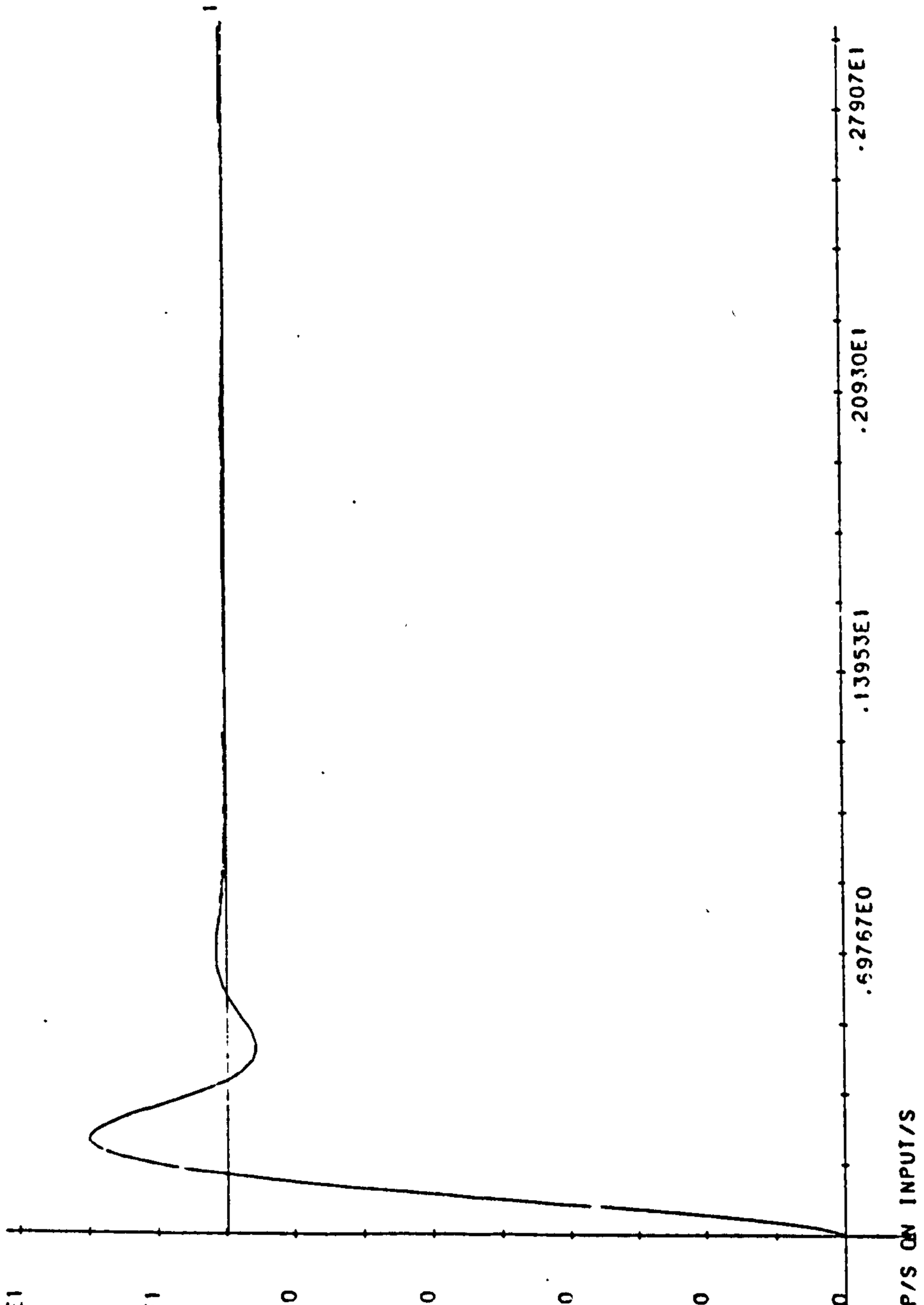


Fig 5.15 Response - Roll Loop, 50 ms^{-1}

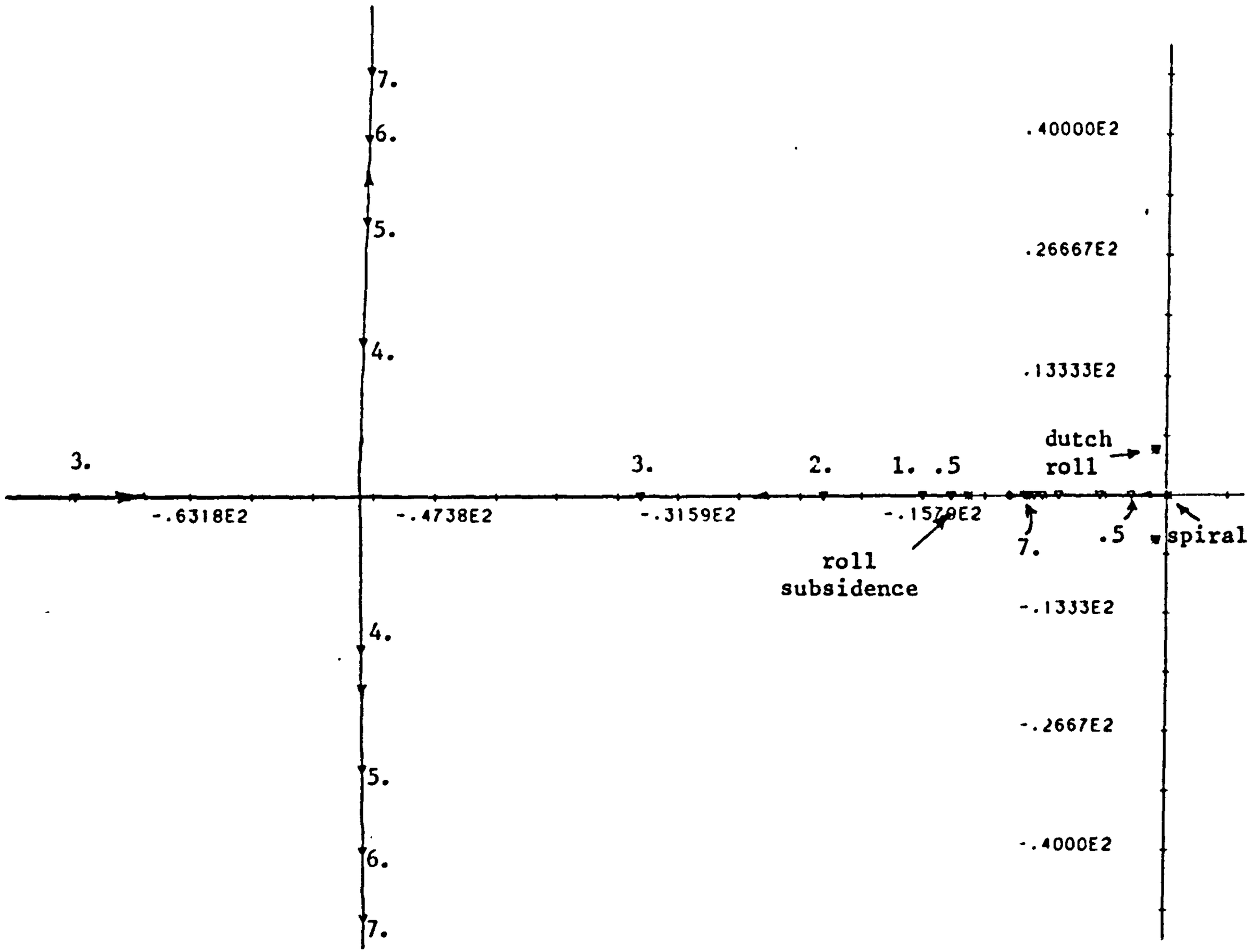


Fig 5.16 Root Locus - Roll Rate Loop, 50 $\dot{m}s^{-1}$ dynamic

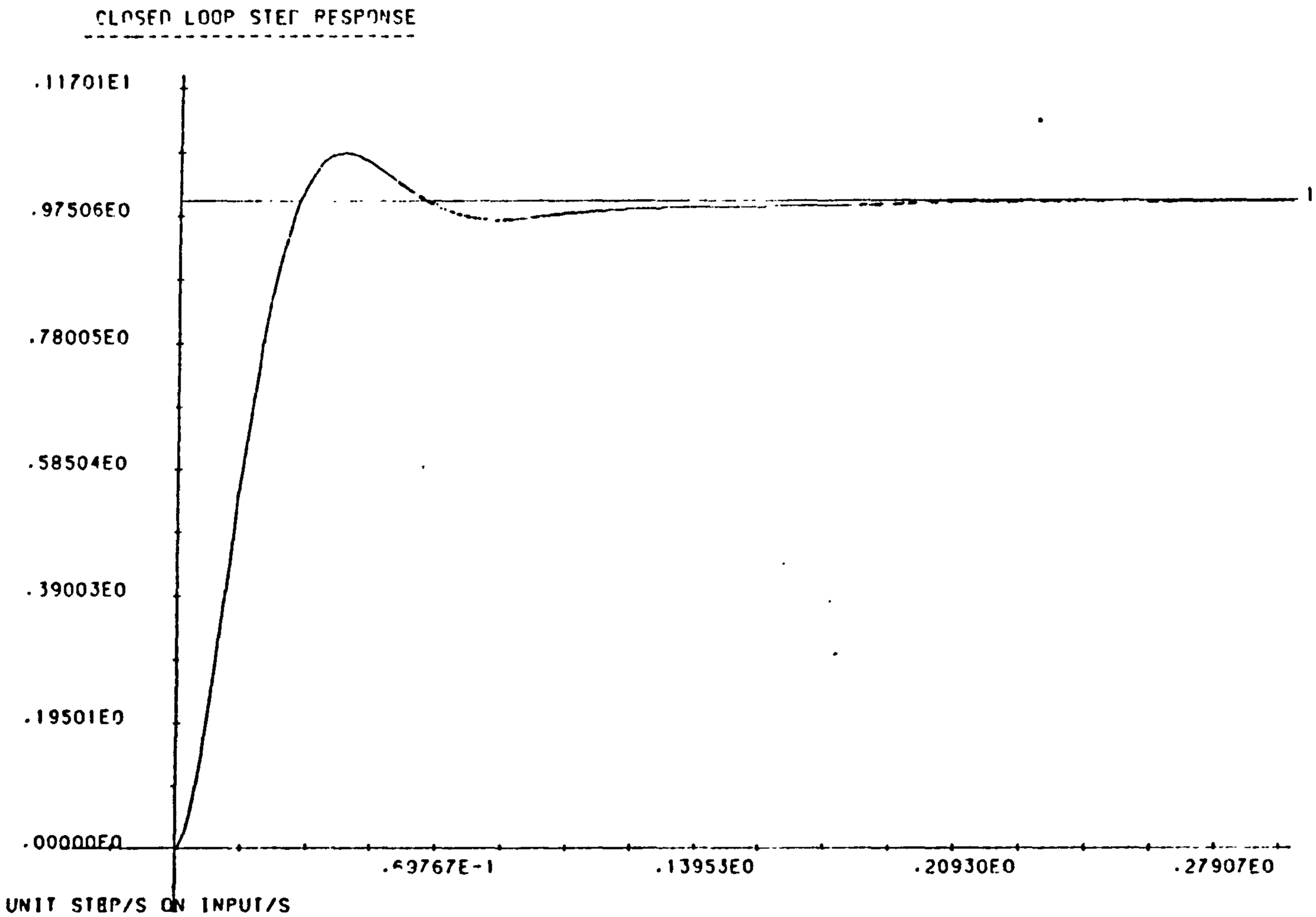


Fig 5.17 Response - Roll Rate Loop, 50 ms^{-1} dynamic

5.4.2.2 Yaw Control

The provision of yawing control on an aircraft is important since yaw angle (heading angle) is used as a means of steering the aircraft. Any yaw controller must thus provide accurate control of yaw angle in addition to removing any tendency to excite undesirable modes when executing a turning manoeuvre. It is important to provide adequate Dutch roll damping and removal of any spiral instability in yaw since rudder deflections in turns or side gusts can excite these modes. One might therefore consider closing the loop from yaw angle, ψ, to rudder deflection, τ. For the Machan at 33 ms^{-1} trimmed flight the appropriate transfer function may be derived from equation 5.3 with :

A'_{1a} (as eqn. 5.4) ; $B'_{1a} = \begin{bmatrix} 5.43 \\ 0. \\ 9.49 \\ 0. \end{bmatrix}$

and $C'_{1a} = [\ 0. \quad 0. \quad 1. \quad 0.\ ]$

noting that the yaw rate to yaw angle is a simple integration and that yaw rate does not couple into the other lateral states and again ignoring actuator dynamics, gives :-

$$\frac{\psi}{\tau}(s) = \frac{K_\psi\ (s + 1/T_\psi)(s^2 + 2\zeta_z\ \omega_z\ s + \omega_z{}^2)}{s(s + 1/T_R)(s + 1/T_S)(s^2 + 2\zeta_d\ \omega_d\ s + \omega_d{}^2)} \qquad - 5.11$$

For the Machan the roll subsidence, spiral and Dutch roll modes are as in equations 5.5 whilst the zeros become :-

$$s = -8.55 \quad ; \ s = 0.236 \pm 0.287j$$

and $K_\psi = 9.49$ - 5.12

A root locus plot for the above loop is given in Fig 5.18. From this we note that the roll subsidence pole moves to the real zero at s=-8.55, the Dutch roll poles move out to follow second order asymptotes and the spiral mode and the

free 's' term come together in the right half plane to form a complex pair which move to the complex zeros in the left half plane. Clearly, to ensure stability the loop gain must be sufficiently high to ensure that this complex pole pair move onto the left half-plane but high gains will inevitably reduce the damping associated with the Dutch roll mode. Nevertheless, for adequately damped open-loop Dutch roll modes a low gain loop can be employed provided that the slow and marginally stable complex pair can be tolerated. For aircraft having poor inherent Dutch roll damping such a scheme may be undesirable in addition to the relatively poor response from the loop. The feedback of yaw angle with a proportional control element does have disadvantages although augmenting the control action by introducing additional feedback of bank angle has been shown to have desirable effects.

A far more satisfactory performance can be achieved by employing direct feedback of yaw rate, r, to the rudder. In this case the free 's' term in the denominator of equation 5.11 is removed giving :

$$\frac{\psi}{\tau}(s) = \frac{K_\psi\ (s + 1/T_\psi\ )(s^2 + 2\zeta_z\ \omega_z\ s\ +\omega_z{}^2)}{(s + 1/T_R)(s + 1/T_S)(s^2 + 2\ \zeta_d\omega_d\ \ s + \omega_d\ {}^2)}$$

with the roots of the numerator factors as in 5.12 for the Machan.

A root locus plot for the above transfer function is shown in Fig. 5.19. Note that the absence of the pole at s=0 removes the complex pair formed by this and the spiral mode. The Dutch roll poles now move rapidly onto the negative real axis and split, one to combine with the spiral mode, the other moves to the zero at $s=-1/T_\psi$. The roll subsidence pole now moves rapidly to the left along the negative real axis. The pole-pair formed by the spiral and one Dutch roll pole leave the real axis and move to the two complex zeros for reasonably high values of gain. For low values of gain the Dutch roll damping can thus be improved, however, the spiral mode may still be divergent. For higher values of gain the response will be dominated by the complex

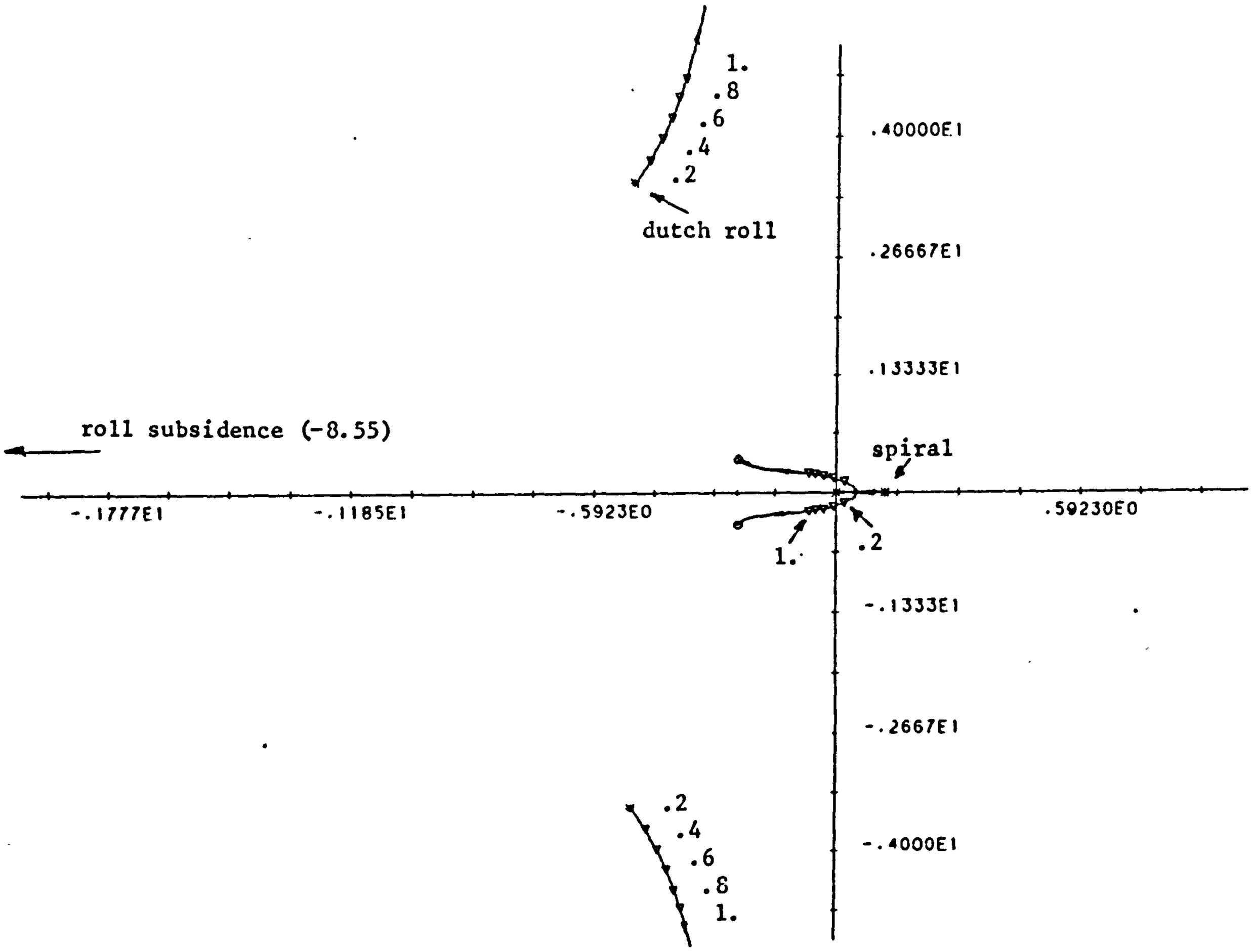


Fig 5.18 Root Locus - Yaw Damper, yaw feedback

pair formed by the spiral and one Dutch roll pole. This implies that a heavily damped and slow response is likely owing to the close proximity of the dominant complex pair to the origin. A closed-loop step response for a proportional controller gain of 10.0 is shown in Fig. 5.20 from which it can be noted that the response is typically slow and heavily damped. This value of gain gives approximately 0.9 damping in the dominant complex pair (see Fig. 5.19). This type of feedback can thus be seen to have a very significant effect on the Dutch roll mode and this may be made heavily damped at relatively low values of gain. The spiral mode however is troublesome in that it moves only very slowly into the left half-plane and high gains are thus required to provide an adequately stable spiral mode. Even then the closed-loop response may be poor and exhibit large steady state offsets. Some form of compensation may thus be desirable.

In steady turns when the pilot may demand that a given rudder angle be set up the yaw damper control system described above will tend to oppose this action. For this reason many yaw dampers include a washout term in the forward path. A washout term essentially provides zero output in the steady state and has a transfer function of the form :

$$G_w(s) = \frac{s}{(s + 1/T_w)} \qquad - 5.13$$

Introducing this in cascade with the aircraft yaw rate dynamics gives an overall yaw channel transfer function

$$\frac{\psi}{\tau}(s) = \frac{K_\psi\, s(s + 1/T_\psi)(s^2 + 2\zeta_z\, \omega_z\, s + \omega_z{}^2)}{(s + 1/T_w)(s + 1/T_S)(s + 1/T_R)(s^2 + 2\,\zeta_d\, \omega_d s + \omega_d{}^2)}$$

The modified root locus for this system is shown in Fig. 5.21 for a washout pole placed at s=-5. From this note that the Dutch roll poles move to the complex zeros, the washout pole moves to the real zero at $s=1/T_\psi$, the roll subsidence moves left along the negative real axis and the spiral mode moves to the zero at s=0. For reasonably small values of

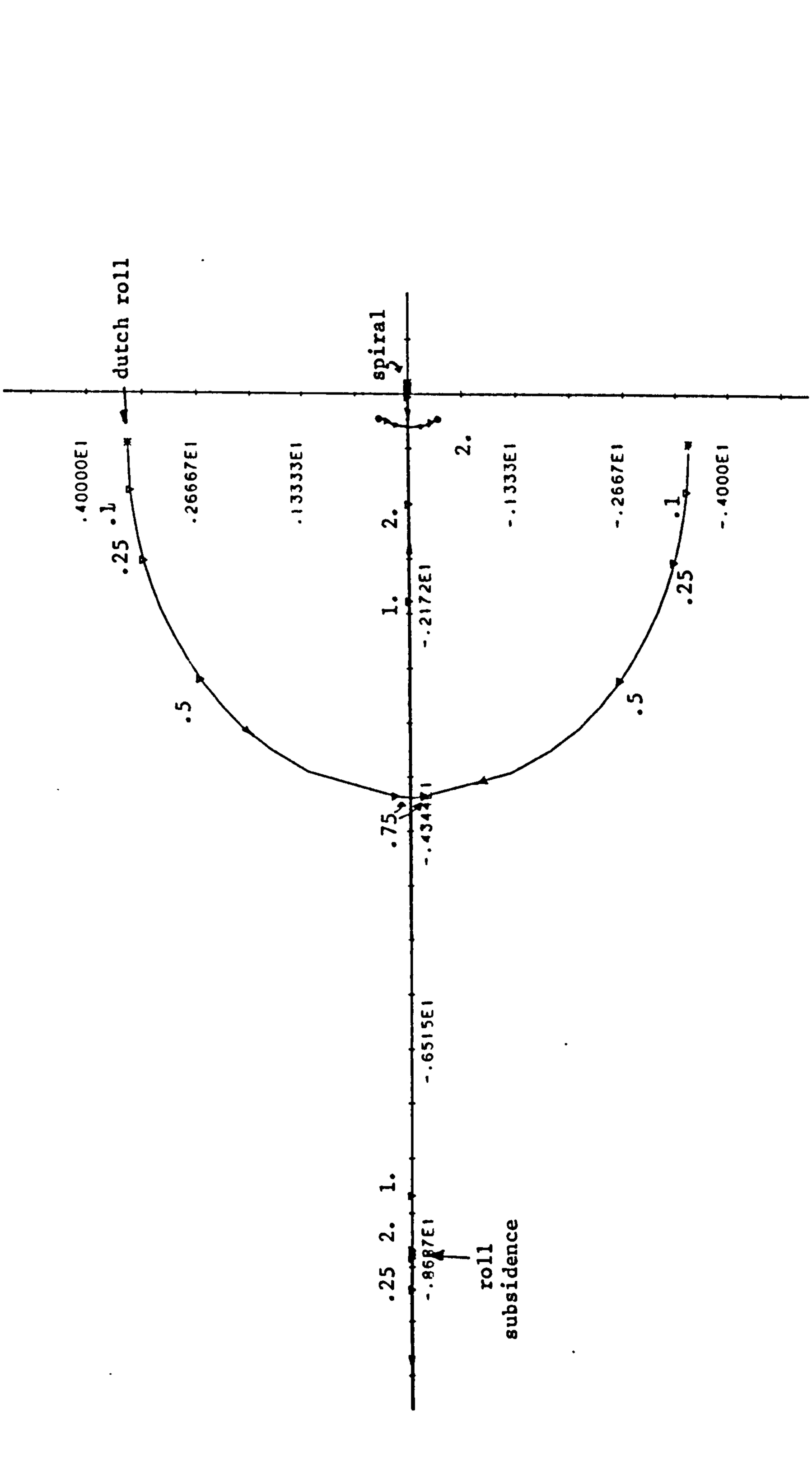


Fig 5.19 Root Locus - Yaw Damper, yaw rate feedback

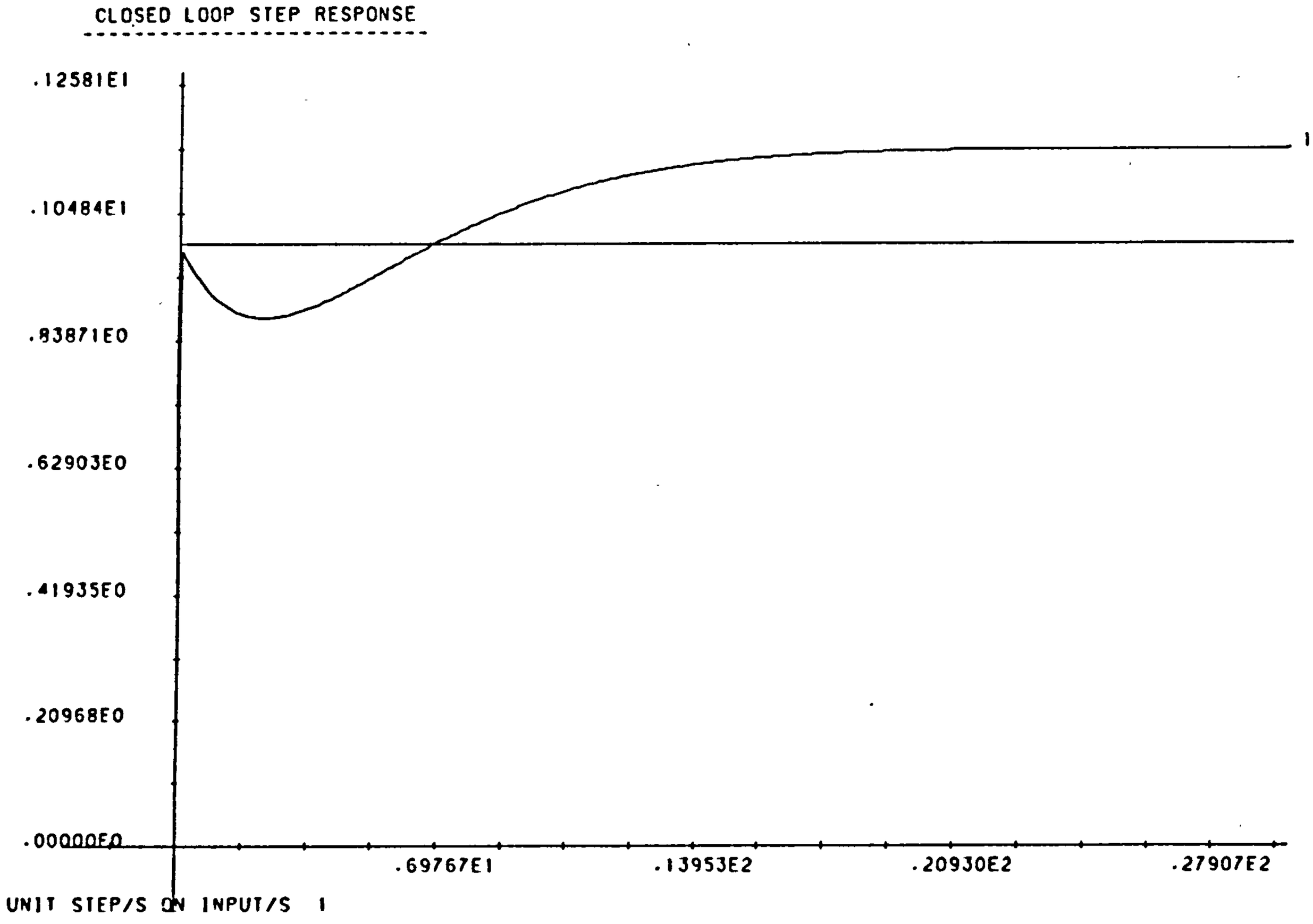


Fig 5.20 Response - Yaw Rate Loop

gain the Dutch roll damping can be considerably improved however the spiral mode remains on the right half plane for all values of gain and gives rise to a slow divergence. For high gain this divergence can be made a very slow root. To provide adequate Dutch roll damping however smaller gains are required and the spiral mode is likely to provide difficulties as is the slow root from the washout pole.

Broadly speaking then the presence of a divergent spiral mode provides a difficult control problem when using either $\psi - \tau$ or $r - \tau$ feedback. The situation is aggravated by the need to include a washout term to preclude the possibility of yaw feedback opposing steady turns. It would therefore seem to be difficult to provide a stable, responsive performance from a yaw damper. Alternative feedback schemes have been tried such as sideslip and lateral acceleration feedbacks but these still suffer from poor spiral stability. If the spiral instability can be tolerated however, and in many cases this slow divergence is easily corrected by the pilot, then significant improvements in Dutch roll damping may be gained from a washed out r - τ loop for relatively low values of closed-loop gain. The provision of a roll damper also assists in stabilising the spiral mode since it effectively provides augmentation of the l_p stabiltiy derivative and changes the disposition of the open-loop poles and zeros. It is also normal to provide precise turn coordination by introducing a rudder to aileron cross-feed term, normally via. a lag/lead type compensation element.

To illustrate the likely effects of a change in flight conditions on the yaw damper described above consider a change in airspeed from $33ms^{-1}$ to $50ms^{-1}$ as was done for the roll damper in 5.4.2.1. Using the results of equations 5.9 with :

$$A'_{la} \text{ (as in 5.9)} \quad ; \qquad B'_{la} = \begin{bmatrix} -12.47 \\ 0. \\ -21.62 \\ 0. \end{bmatrix}$$

$$C'_{la} = [\ 0. \quad 0. \quad 1. \quad 0.\ ]$$

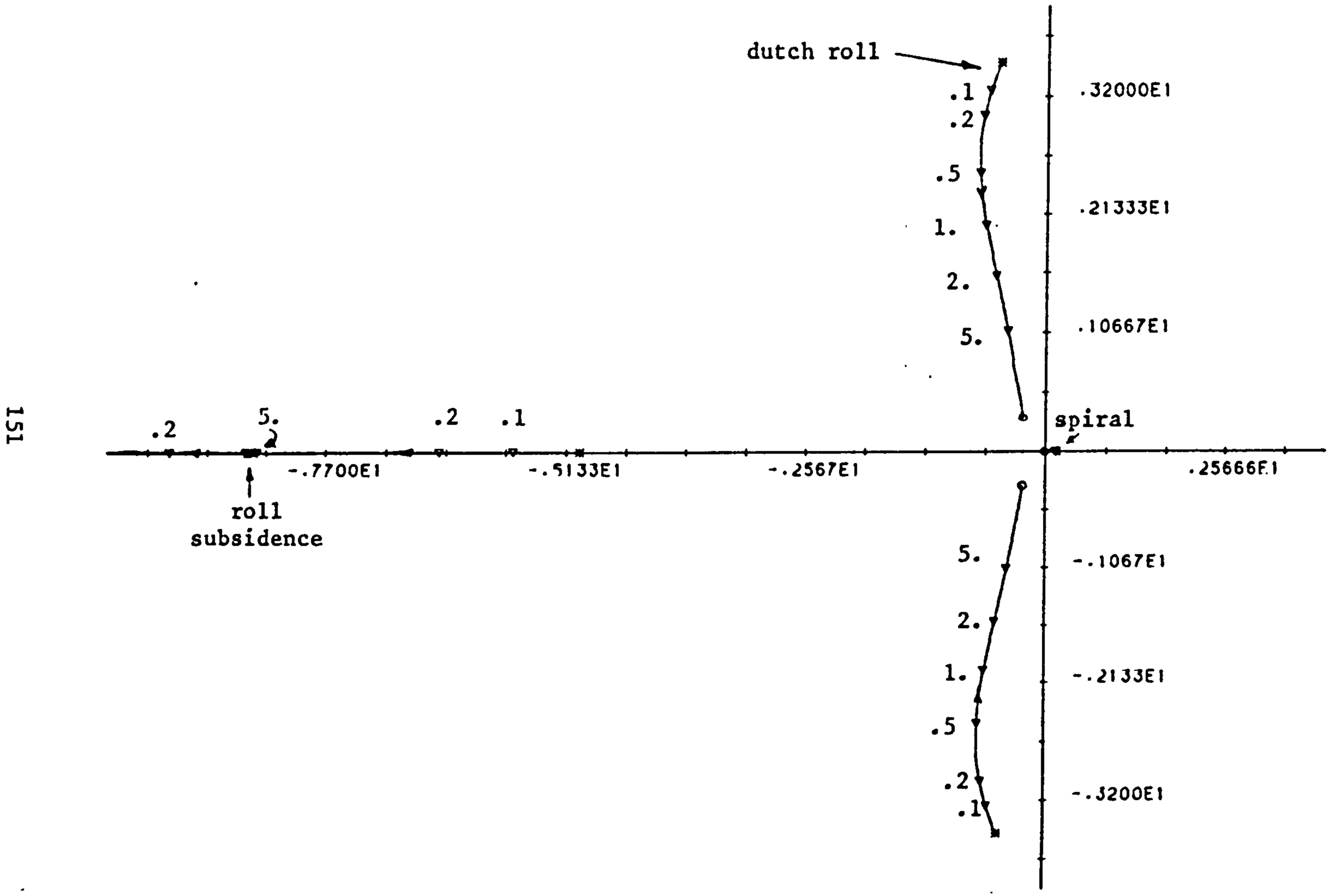


Fig 5.21 Root Locus - Washed Out Yaw Damper

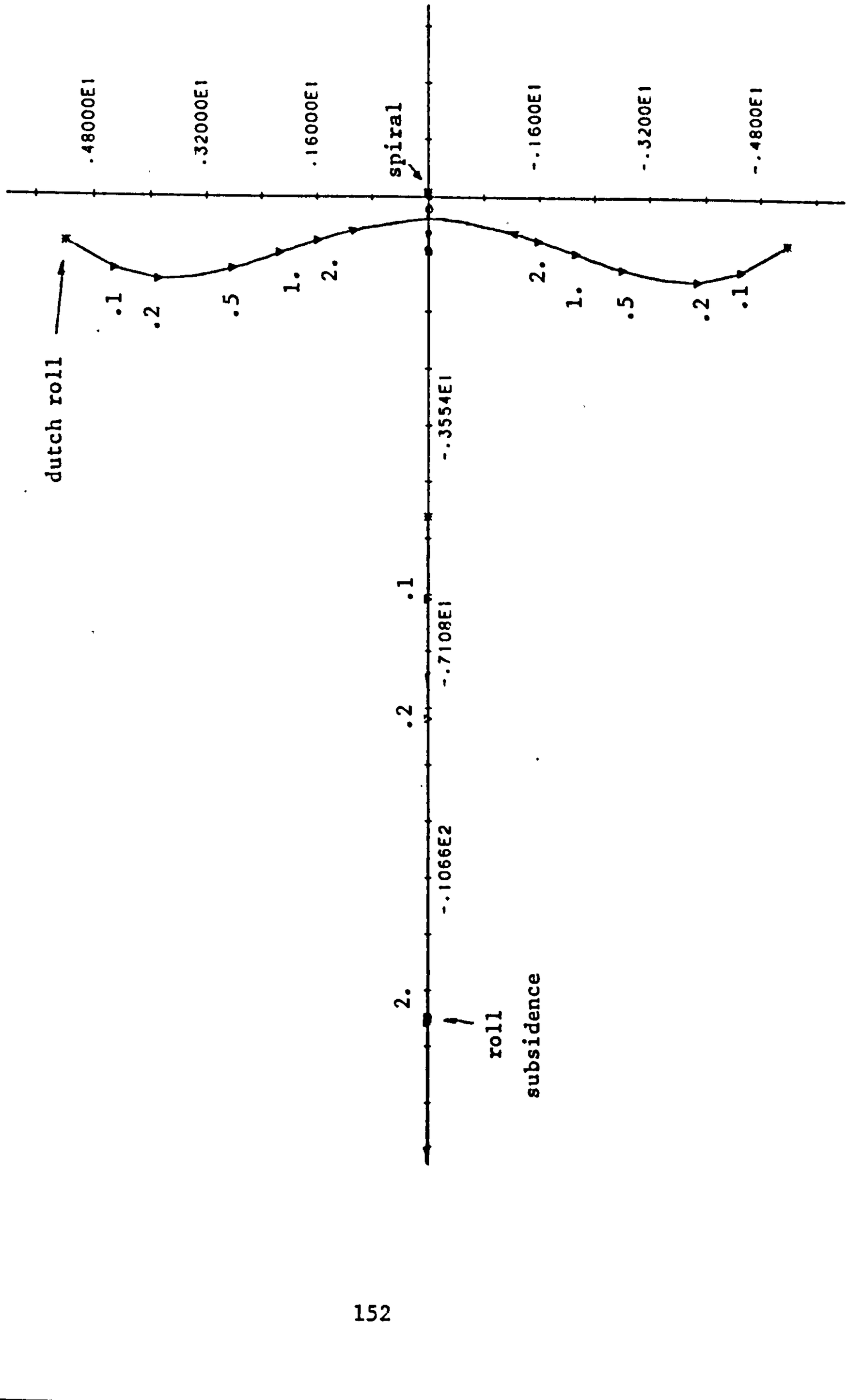


Fig 5.22 As Fig 5.21, 50 ms^{-1} dynamic

gives :

$$\frac{\psi}{\tau}(s) = \frac{K_\psi \; s \; (s + 1/T_\psi)(s + 1/T_1)(s + 1/T_2)}{(s + 1/T_w)(s + 1/T_R)(s + 1/T_S)(s^2 + 2\,\zeta_d\,\omega_d s + \omega_d{}^2)}$$

where the roll subsidence, spiral and Dutch roll modes are as in equations 5.9 and the zeros are at :

$$s = 0.0 \; ; \; s = -0.139 \; ; \; s = -0.867 \; ; \; s = -12.652$$

and $K_\psi = 21.62$

A root locus plot for the above is given in Fig. 5.22 and is broadly speaking similar to Fig. 5.21. Note however that the Dutch roll poles now move to the zeros at s=-0.139, s=-0.867, this for very high values of gain. The damping of the Dutch roll mode can still be improved for relatively low gains but again the spiral mode remains divergent for all gain values. The increased aircraft gain does however imply that gain-scheduling is required in order to provide reasonably constant Dutch roll damping with varying airspeed.

5.4.3 Height Hold

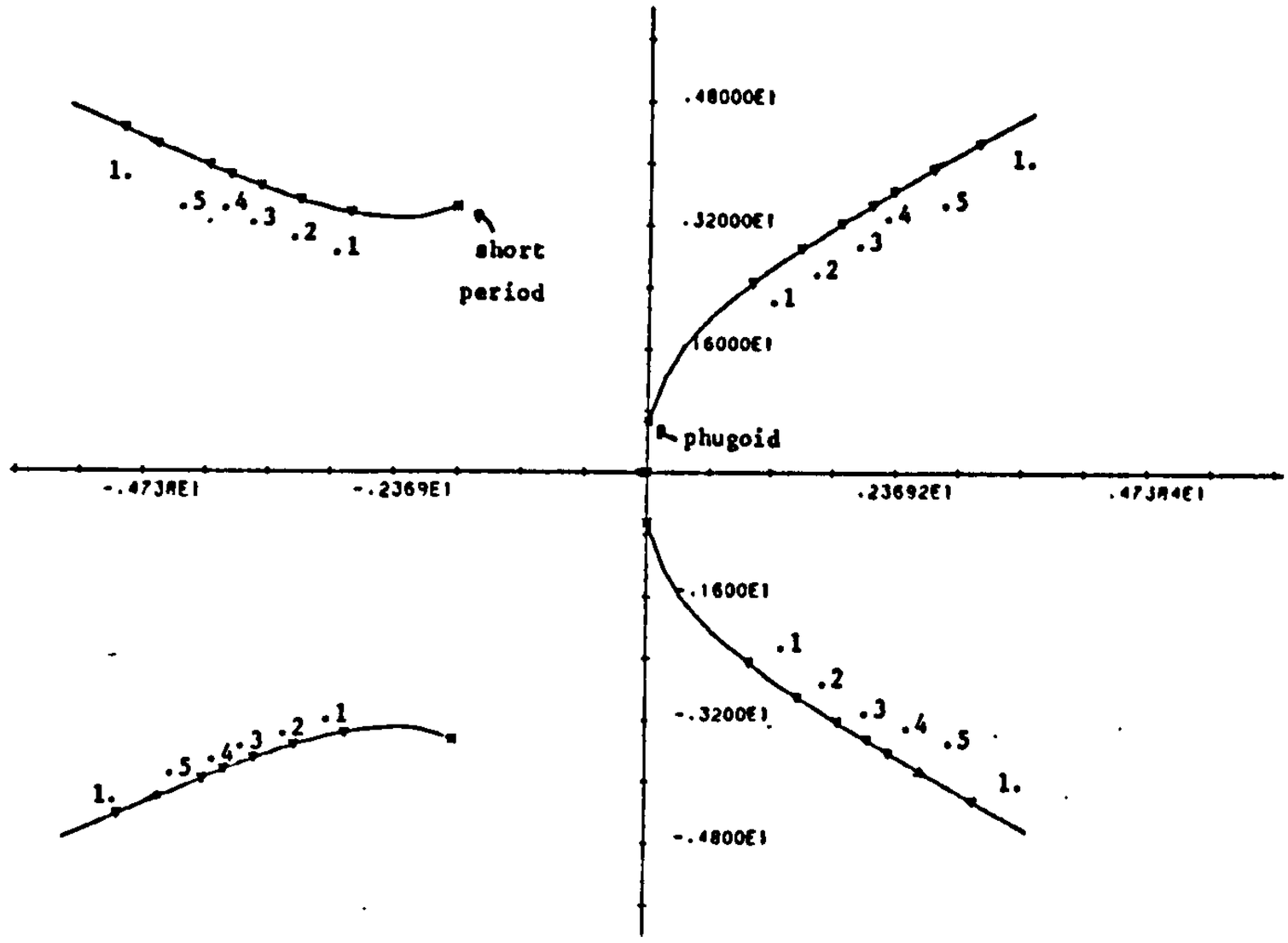
Returning briefly to the longitudinal control problem a height hold function is desirable in vehicles like the Machan in order to provide long term height following. The feedback loop here could be formed from height to elevator but the closure of this loop normally drives the phugoid mode unstable for all but very low values of proportional gain. A potentially more suitable scheme is the feedback of height rate to the elevator. For the Machan the appropriate transfer function can be evaluated as in equations 5.1 for $33 ms^{-1}$ stick fixed flight and with

$$C'_{lo} = [\; 0. \quad 1. \quad 0. \quad 0. \;]$$

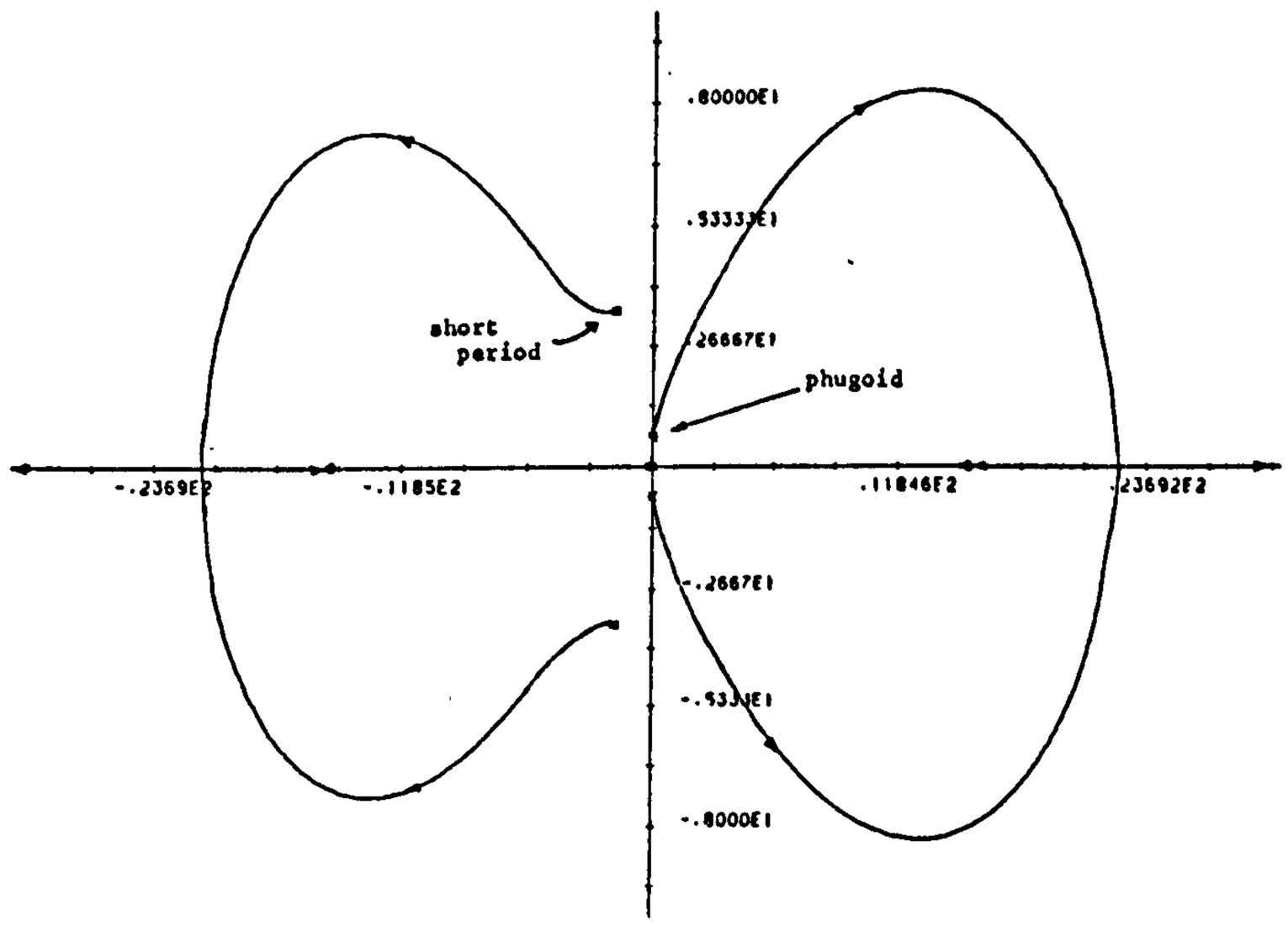
treating $\dot{h} = w$ i.e. zero pitch attitude, thus:

$$\frac{\dot{h}}{\eta}(s) = \frac{K_h \quad (s + 1/T_1)(s + 1/T_2)(s + 1/T_3)}{(s^2 + 2\zeta_s \omega_s s + \omega_s{}^2)(s^2 + 2\zeta_p \omega_p s + \omega_p{}^2)}$$

Fig 5.23 Root Locus - Height Hold Loop



Extended Version of above



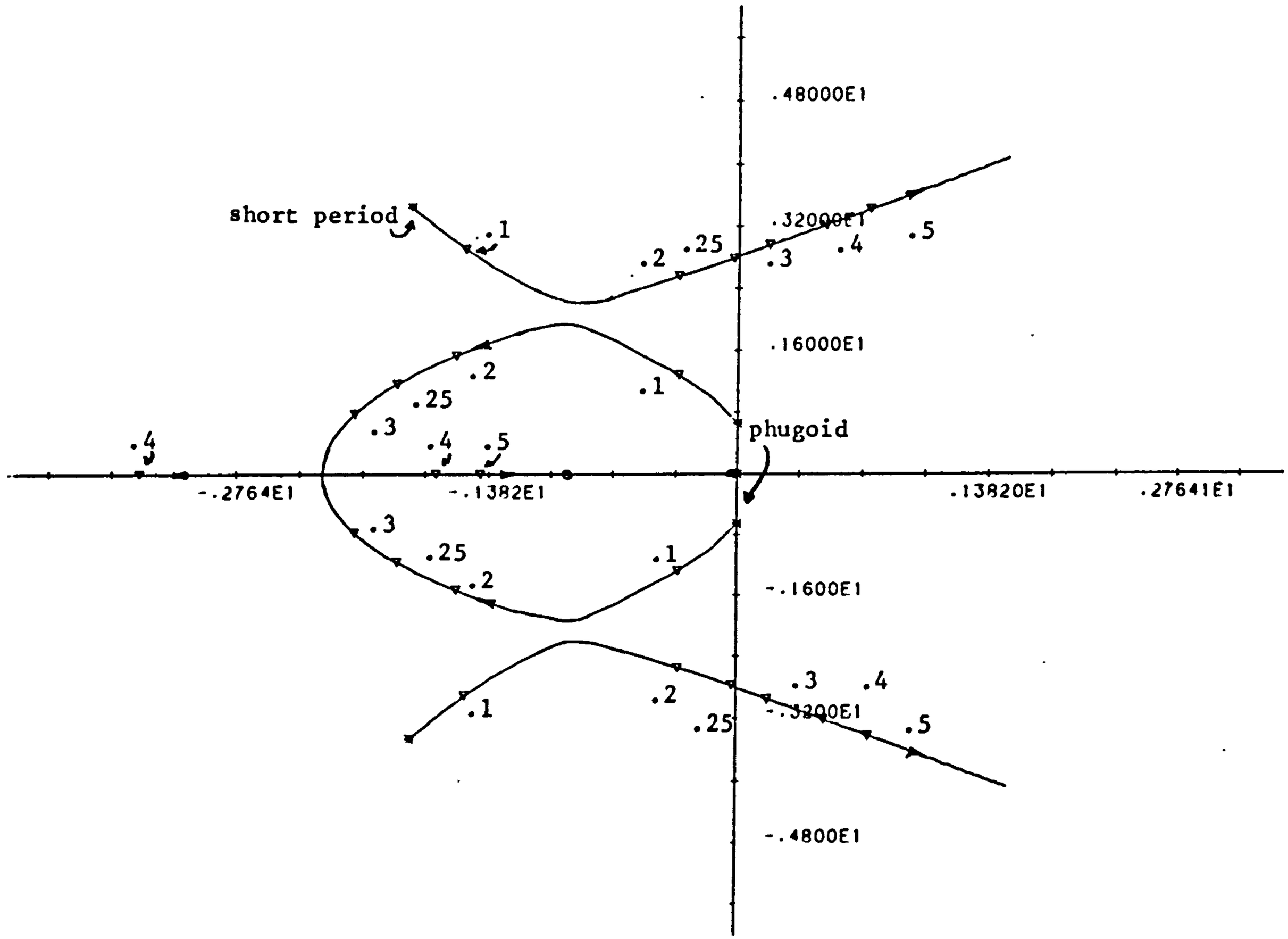


Fig 5.24 Root Locus - Height Hold Loop, compensated

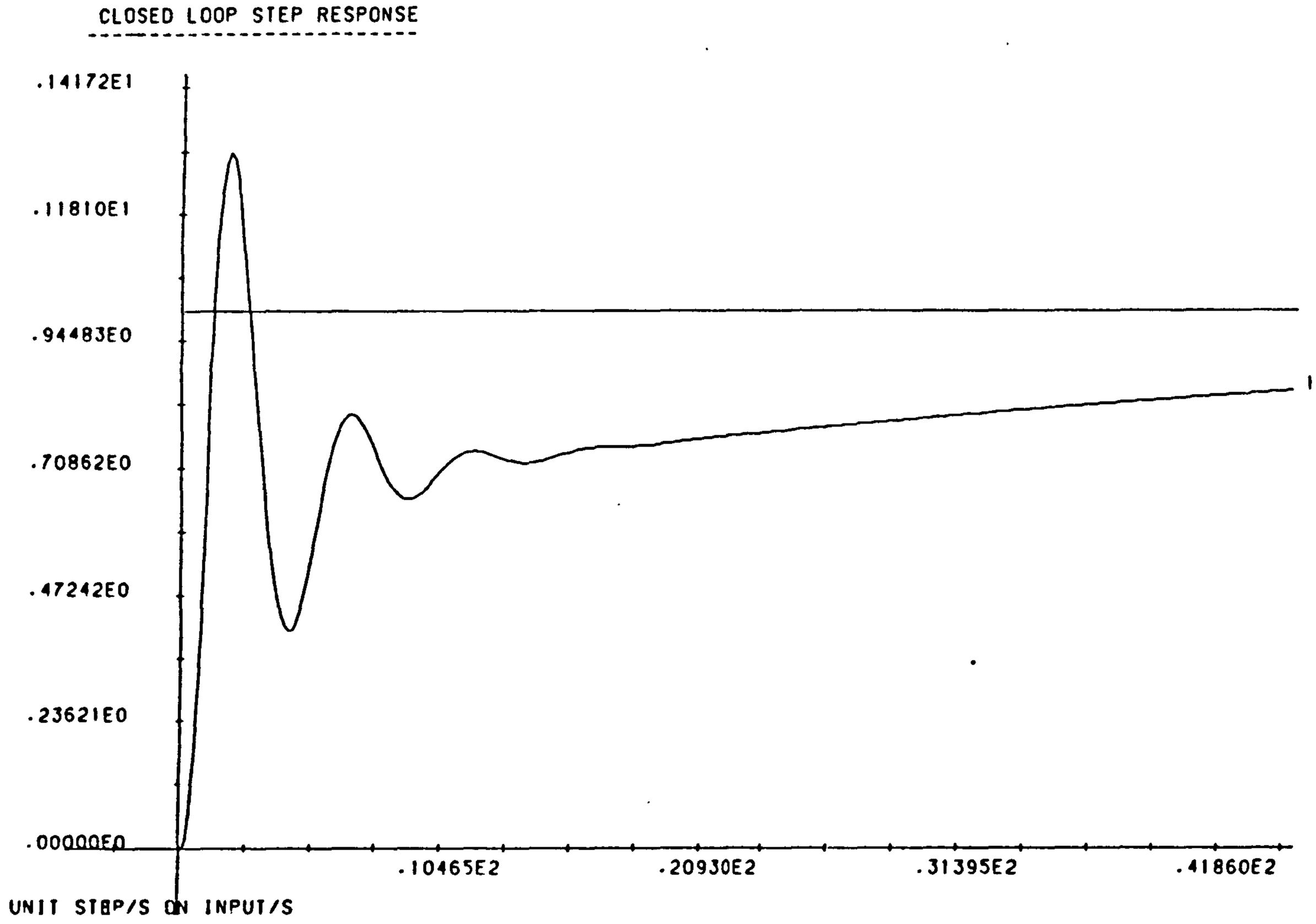


Fig 5.25 Response - Height Hold Loop, compensated

The zeros are at s=-0.027, s=-15.34 and s=15.12 and K_h = 5.318

The height to elevator transfer function can then be obtained by the introduction of a pole at s=0 giving :

$$\frac{h}{\eta}(s) = \frac{K_h\,(s + 1/T_1)(s + 1/T_2)(s + 1/T_3)}{s(s^2 + 2\zeta_s\,\omega_s s + \omega_s{}^2)(s^2 + 2\zeta_p\,\omega_p\,s + \omega_p{}^2)}$$

A root locus for the above is shown in Fig. 5.23 from which it is clear that the phugoid poles move onto the right half plane for very low values of gain and that the pole at s=0 is approximately cancelled by the zero at s=-0.027. The introduction of a lead network in the controller would, however, improve the system performance.

Consider then the introduction of a compensator of the form

$$G_C(s) = \frac{K_C\,(s + \alpha_1)}{(s + \alpha_2)}$$

with α_1 = -1., α_2 = -10.

The modified root locus plot is now shown in Fig. 5.24 and from this note that it is now the short period mode which moves into instability. The phugoid damping may also be slightly improved at lower gain values. For a closed-loop controller gain, K_C, of 0.1 a step response is shown in Fig. 5.25. The lightly damped short period mode is clearly evident along with the slight steady state offset. Higher gain values than K_C=0.26 will drive the short period mode into instability and hence are not desirable. Changes in flight condition, e.g. forward speed changes, will necessitate the scheduling of the controller gain in order to avoid the effects of the unstable short period mode.

The above discussion indicates that the use of the elevator to control height is generally undesirable, although achievable. Rate of climb systems using airspeed or Mach number feedbacks are normally more effective.

5.5 Alternative Schemes

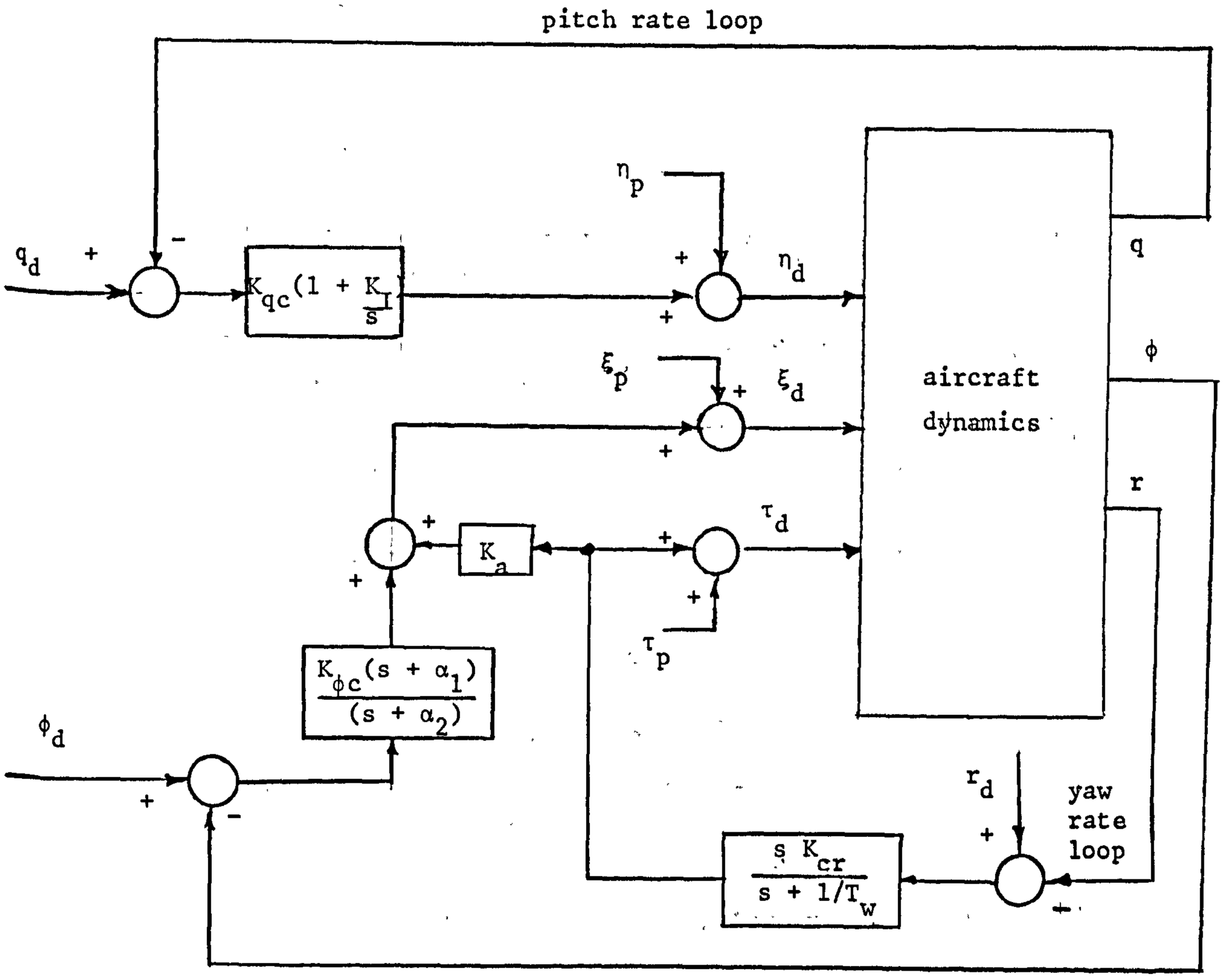
It is fair to say that the feedback of almost every measurable flight variable to the control surfaces has been investigated at some time or another. The most effective loops are generally retained since these are most likely to provide improvements in performance. The discussion above has indicated the control problems associated with some of the Machan systems, bearing in mind the limited measurements available on this vehicle. The particular loops chosen are by no means exhaustive. The lack of acceleration data precludes the use of acceleration feedback. This, however, has been shown to have desirable features in many aircraft. Equally, the feedback of angle of attack or sideslip to the appropriate control surface can be shown to yield improvements in performance. Airspeed hold is also very useful in manoeuvres requiring accurate control of airspeed, e.g. autoland, etc. The reader is referred to the many texts available on this subject for further information (1-4,167). To complete this section a brief examination will be made of the performance of the Machan with the above designed controllers.

5.6 Perfomance Assessment

In order to assess the potential usefulness of the classically designed control schemes derived above they were incorporated into the non-linear Machan simulation which was fully described in Chapters 1 and 3. The results were obtained using the FORTRAN version of the simulation running on the DEC-10 mainframe. This simulation includes actuator dynamics along with the appropriate rate and saturation limits.

The loops which have been considered are pitch rate control and roll and yaw dampers. These loops are closed around the elevator, aileron and rudder control surfaces and are shown pictorially in Fig. 5.26. No control is applied to the throttle although this could be provided by an airspeed - throttle loop. Furthermore, no provision has

Fig 5.26 Loop Closures on Machan



q_d, ϕ_d, r_d - autopilot demands

η_p, ξ_p, τ_p - pilotic demands

been made for the height hold loop since this presents some problems, as noted in 5.4.3 above. The controller parameters were largely as designed above and are summarised as ;

i) pitch rate K_{qc} = 0.074 rad/rad/s ; $1/K_I$ = 1.0 s^{-1}

ii) roll damper $K_{\varphi c}$ = 1.75 rad/rad ; α_1 = 10 s^{-1}
α_2 = 100 s^{-1}

iii) yaw damper K_{cr} = 0.5 rad/rad/s ; $1/T_W$ = 5 s^{-1}

No attempt has been made to provide gain-scheduling since the airspeed variations over the simulation are slight.

Fig. 5.27 gives the time histories of the aircraft's principal flight variables over a 30 second simulation with the non-linear model. The initial conditions imposed on the simulation were zero other than the following :

forward velocity, u = 30, 40 and 50 ms^{-1} for the three cases
height , h = 30 m
thrust , X_E = 50 N
elevator angle , η = 11°
side velocity , v = 1.0 ms^{-1}
roll rate , p = 0.5 rad s^{-1}
throttle , T_H = 100%

In addition a step pilotic demand in elevator of -3° is provided after 2 seconds.

In the longitudinal axis, Figs. 5.27 a) to g), the initial forward airspeed is lost to give a steady state of 30 ms^{-1} for all three initial airspeeds, this being an equilibrium condition with the given steady state thrust of roughly 130 N. This initial energy is largely transferred into increasing the aircraft's height since no height hold loop has been employed. The pitch rate loop tends to drive the pitch rate, q, to zero after an initial transient and a transient due to the change in elevator demand. Note that the fast short period mode is evident in the q response and is well damped. The phugoid mode is also apparent as a long

Fig 5.27 Primary Flight Variable Responses

(Non-linear Machan model, classical control)

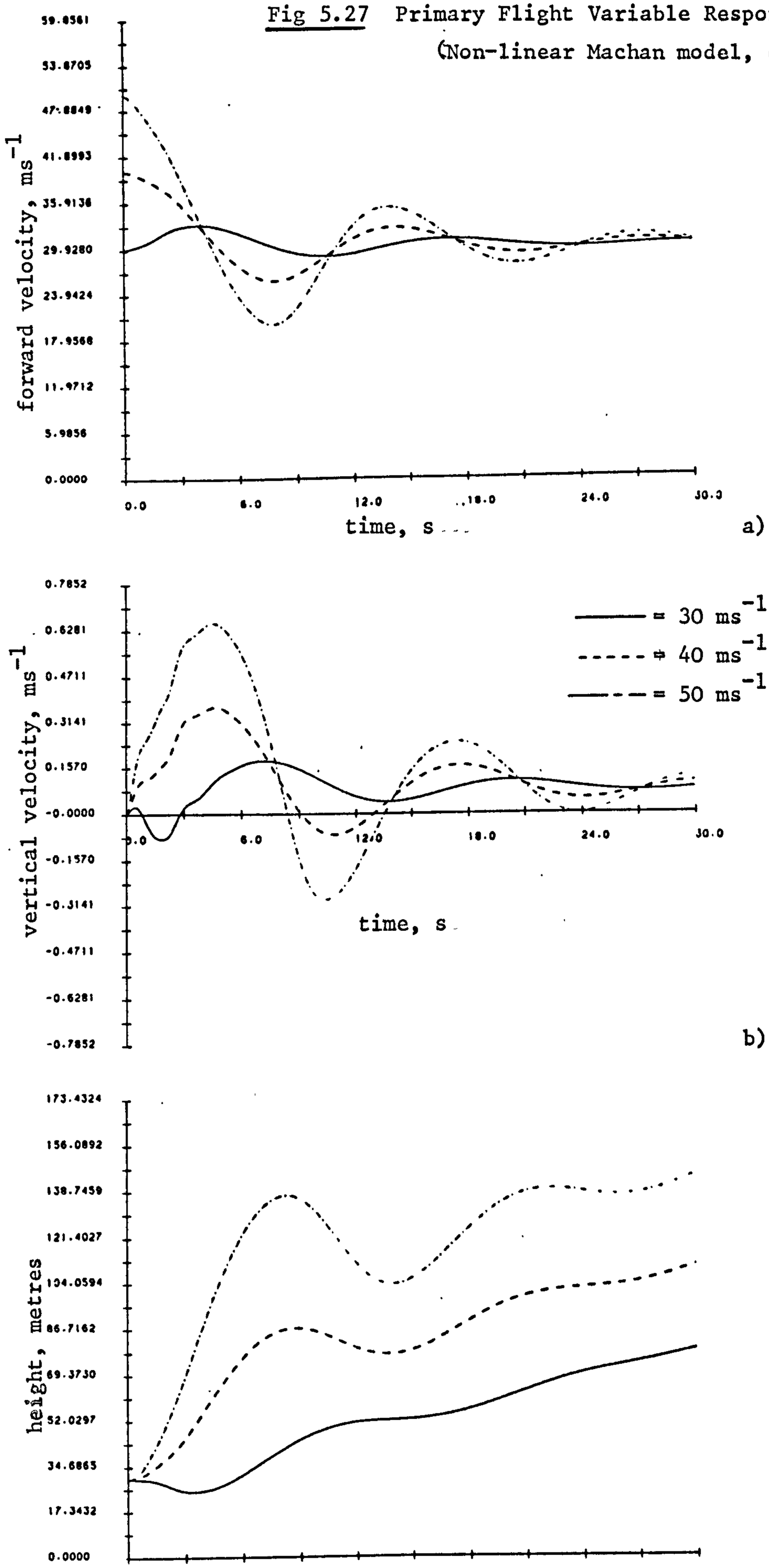
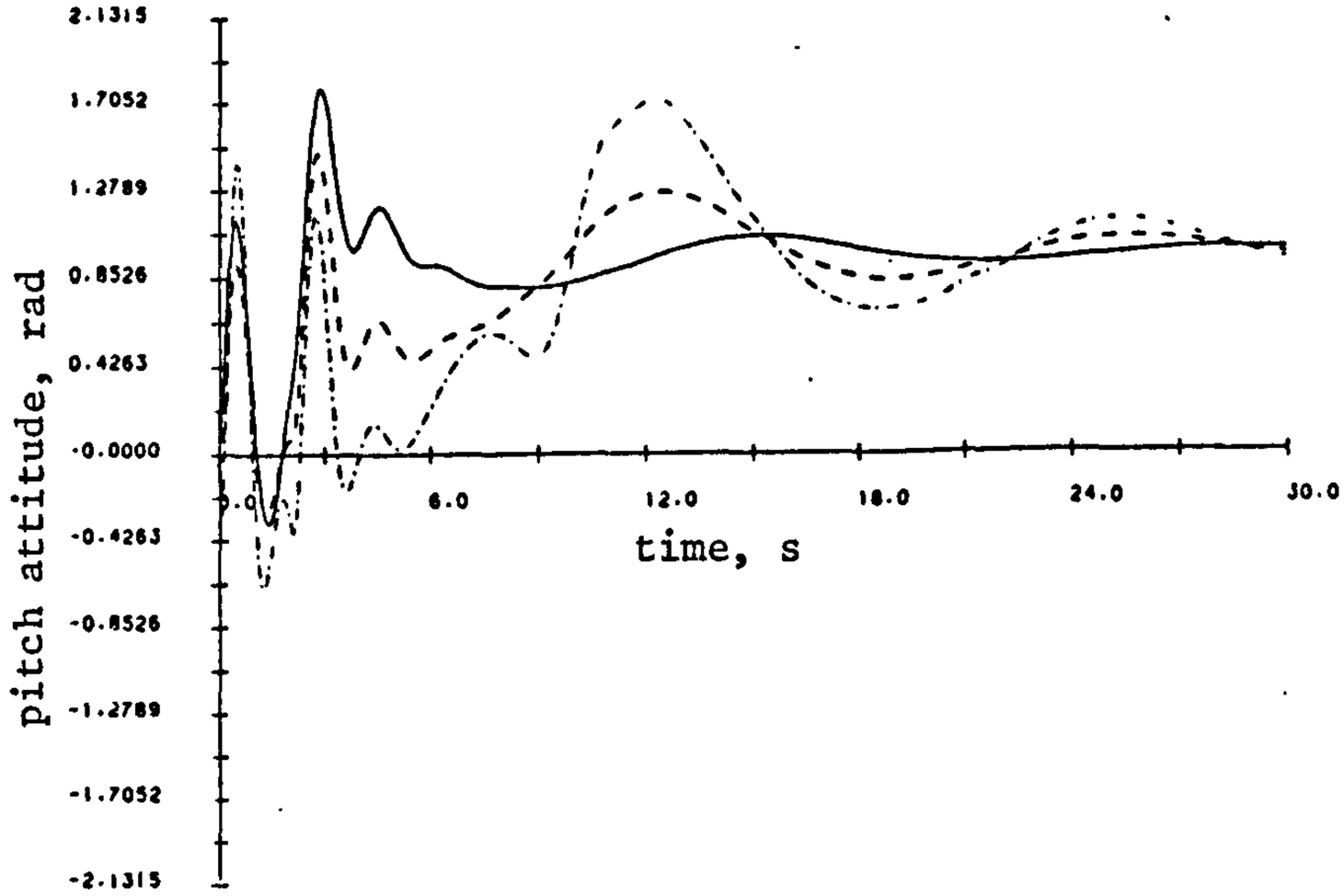
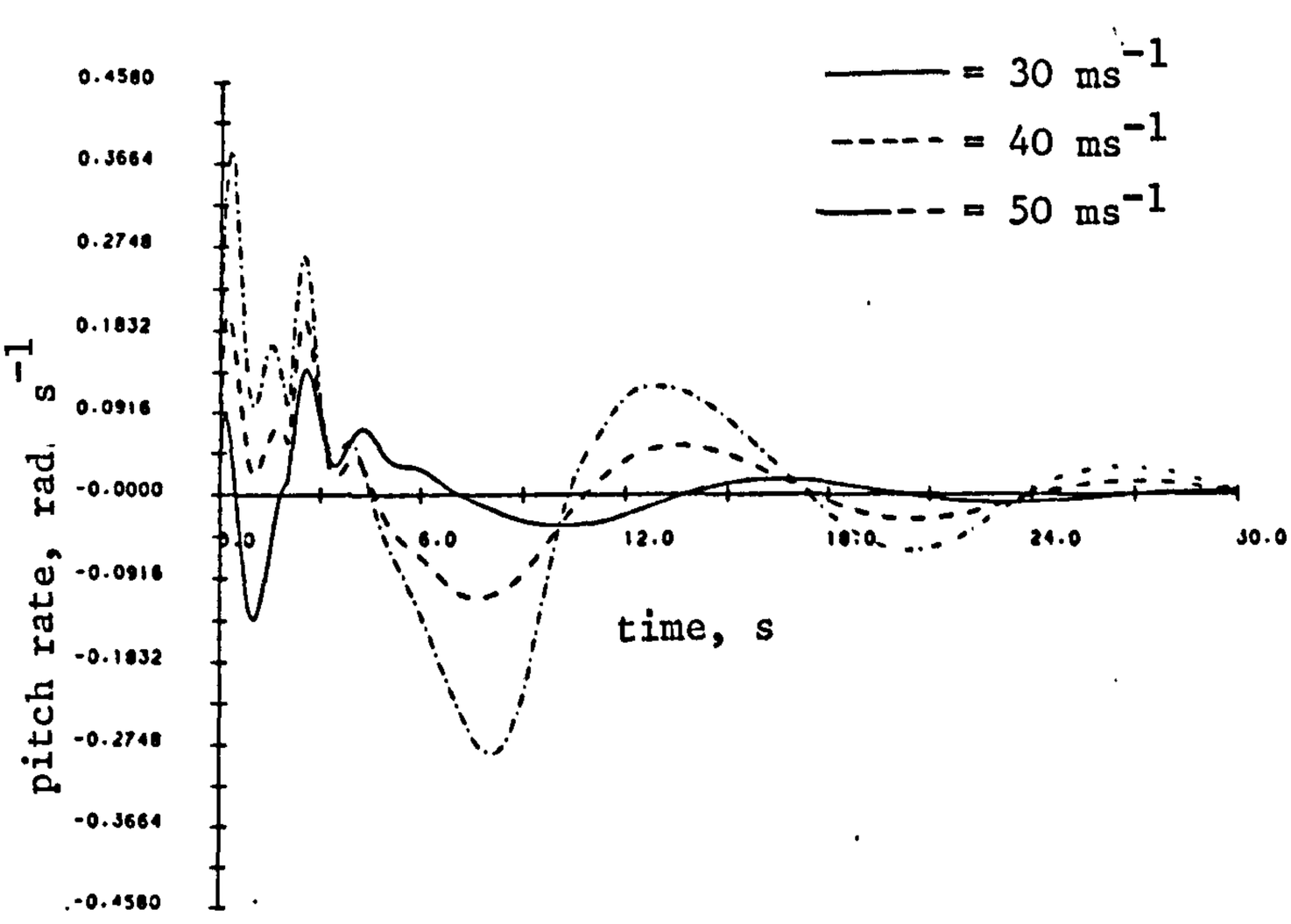


Fig.5.27 cont.

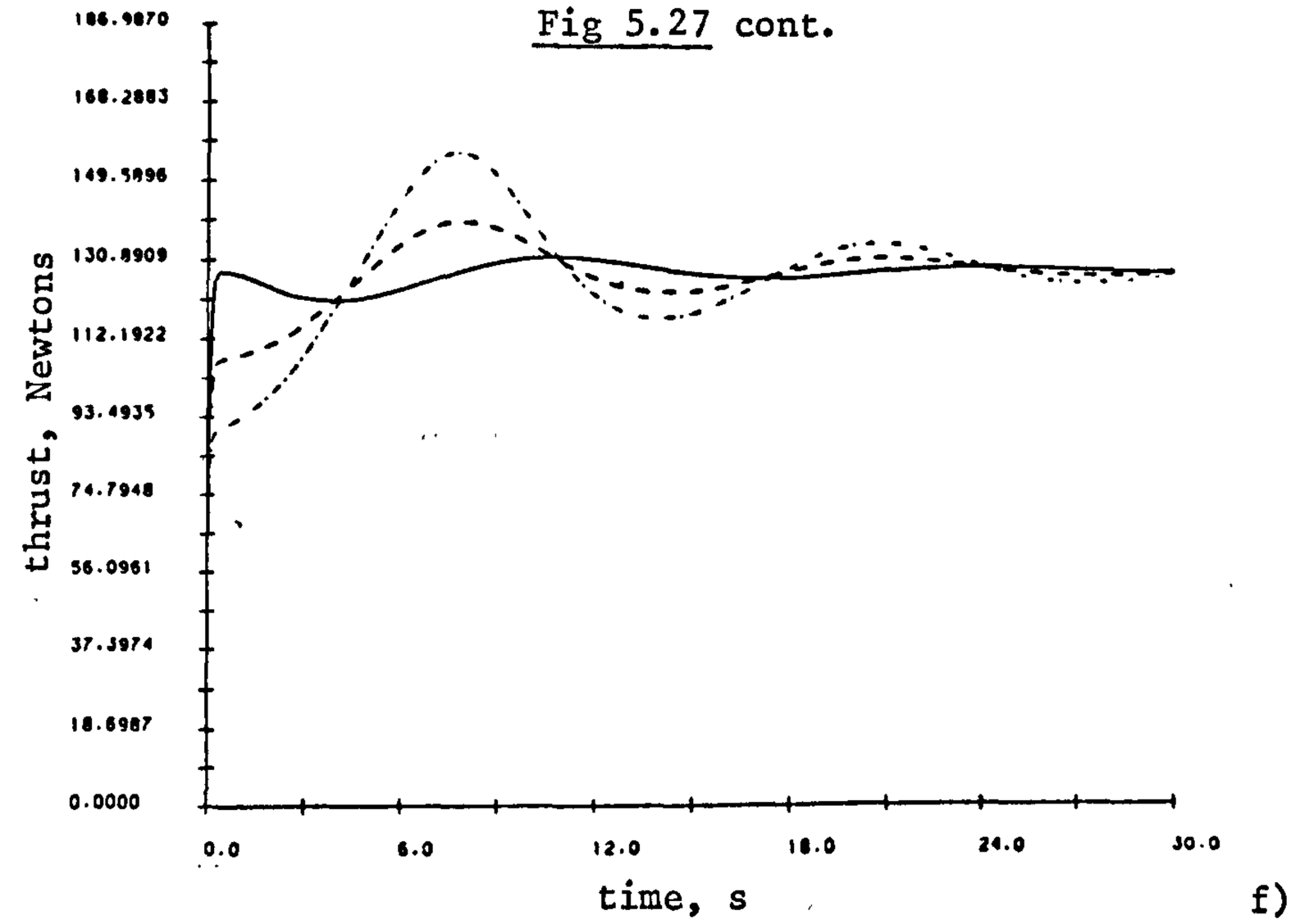


d)

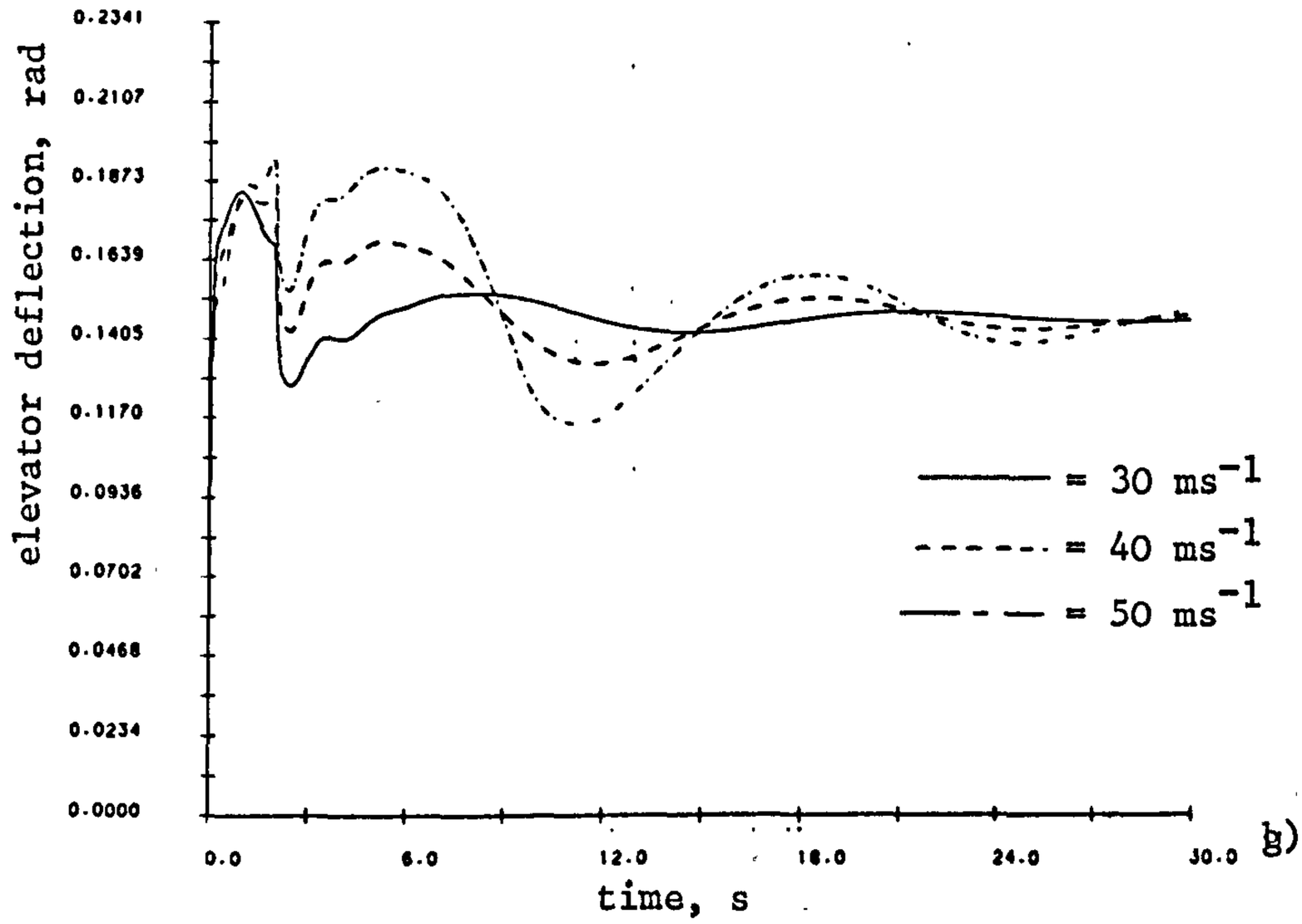


e)

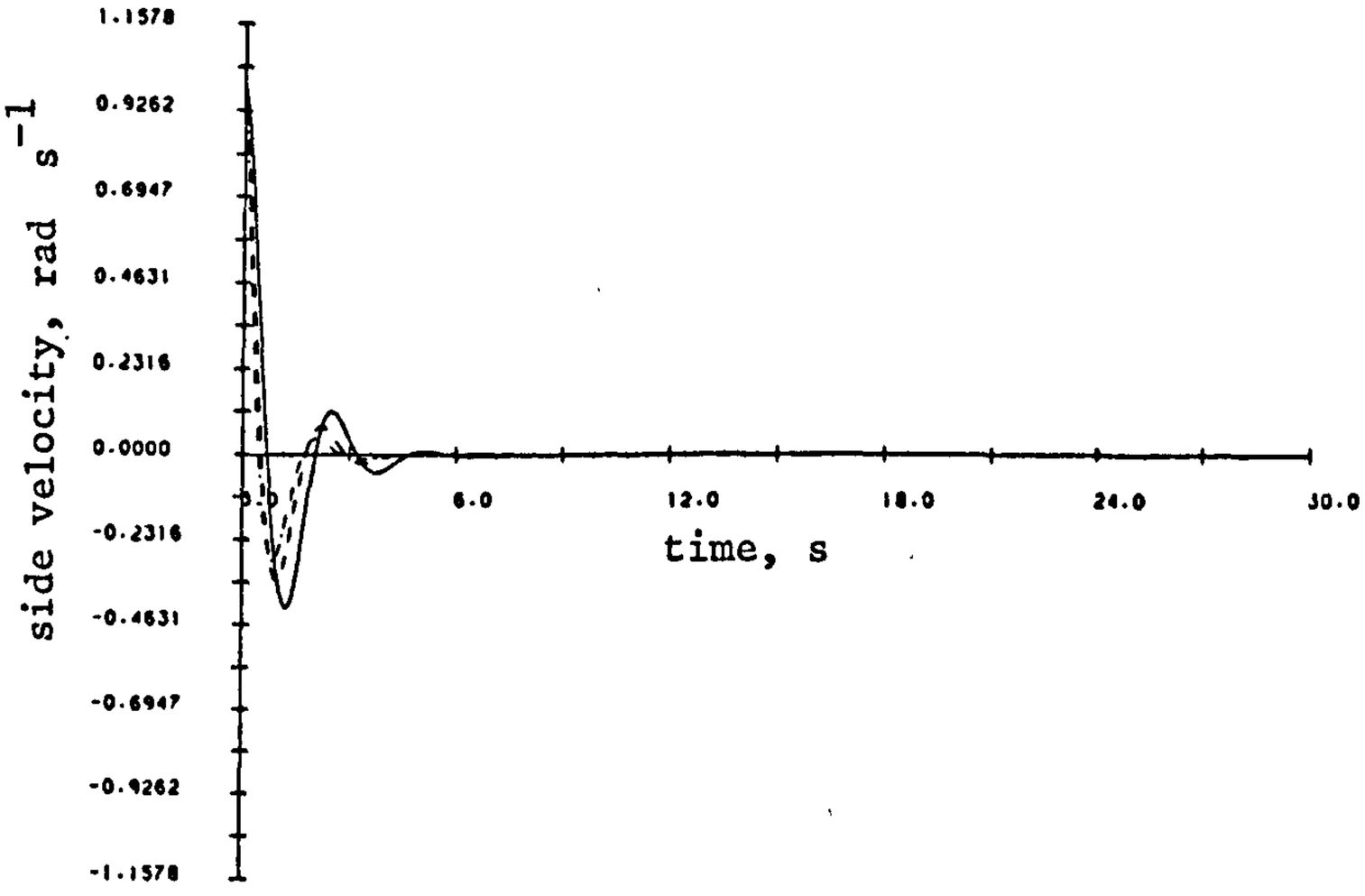
Fig 5.27 cont.



f)

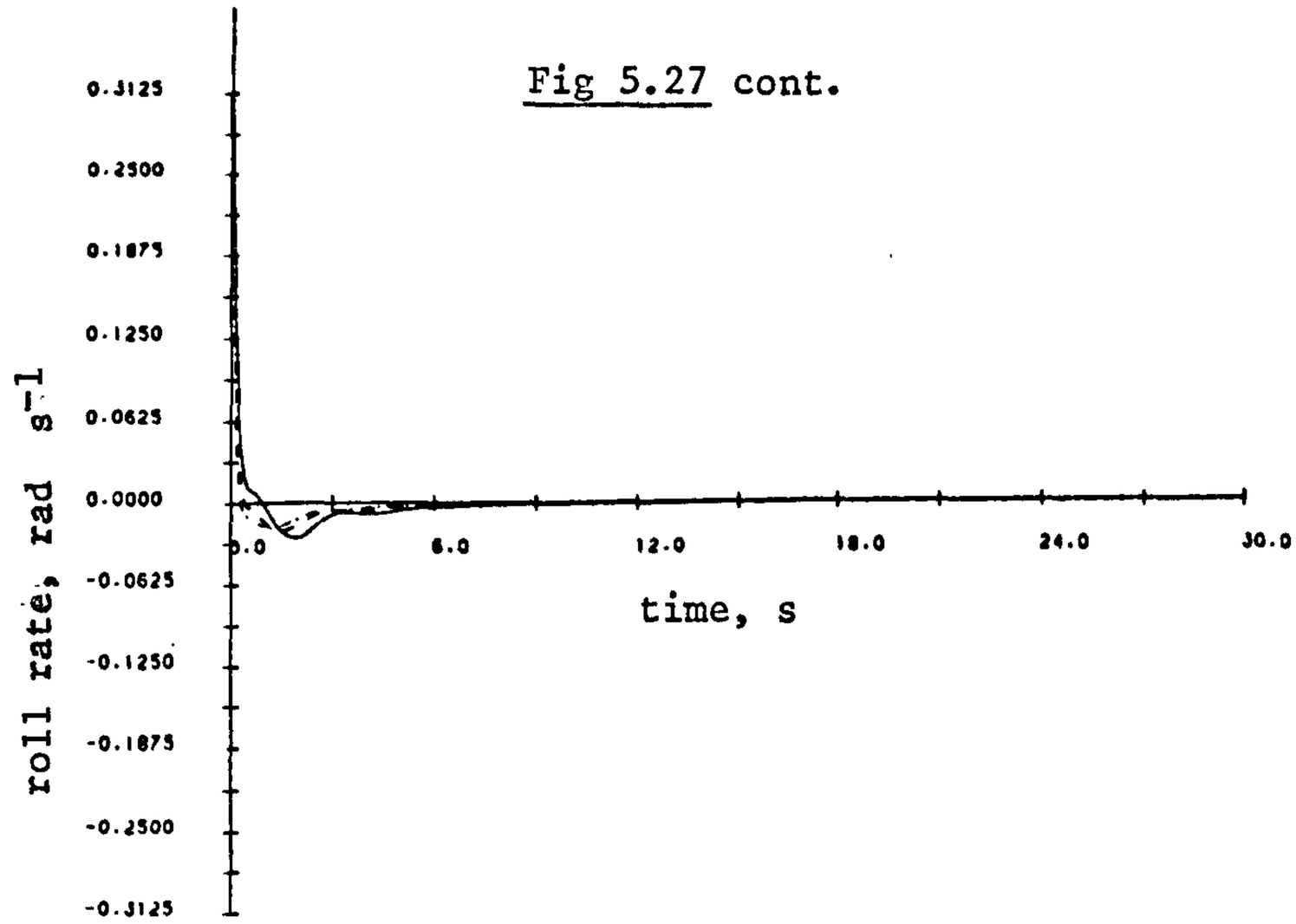


g)

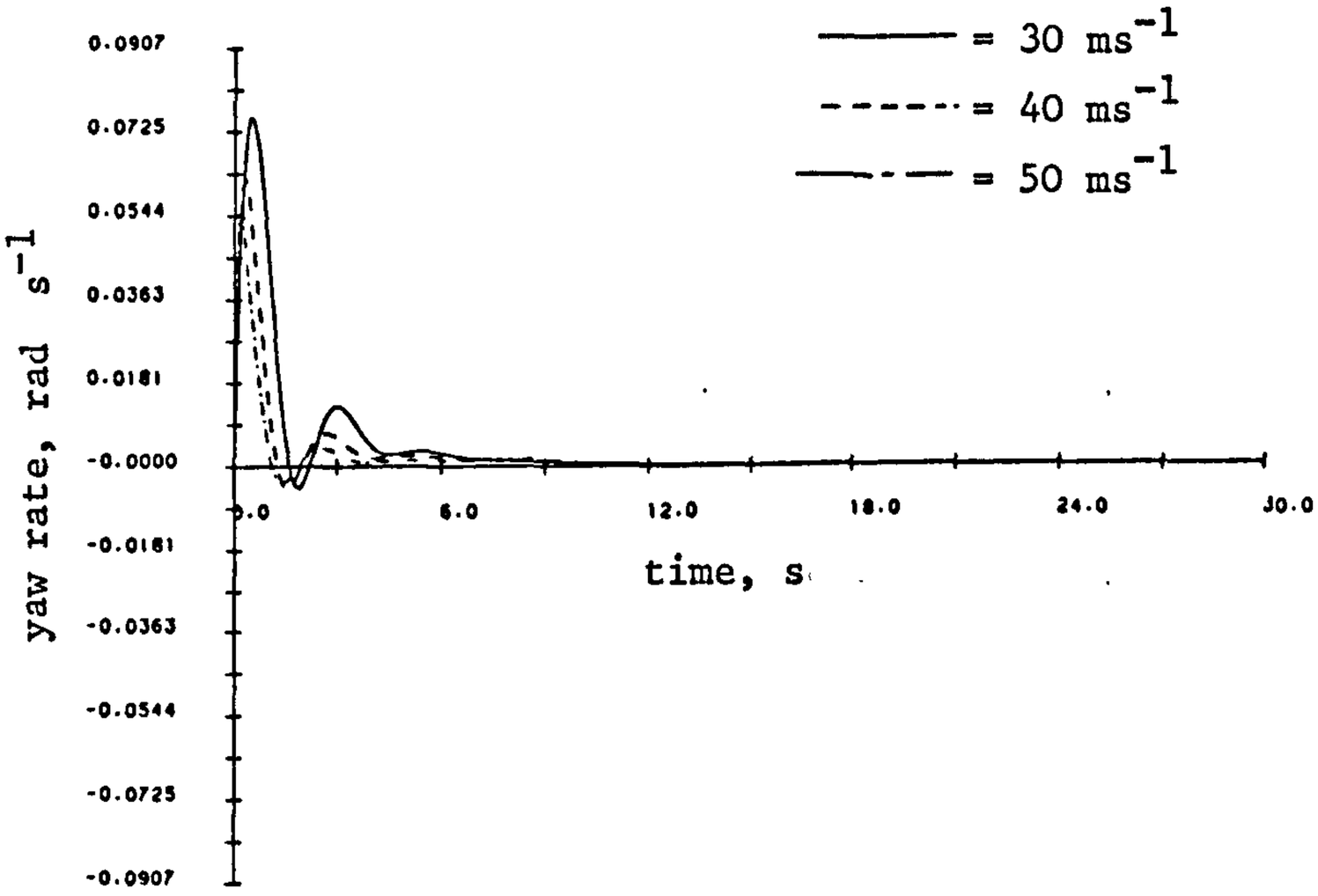


h)

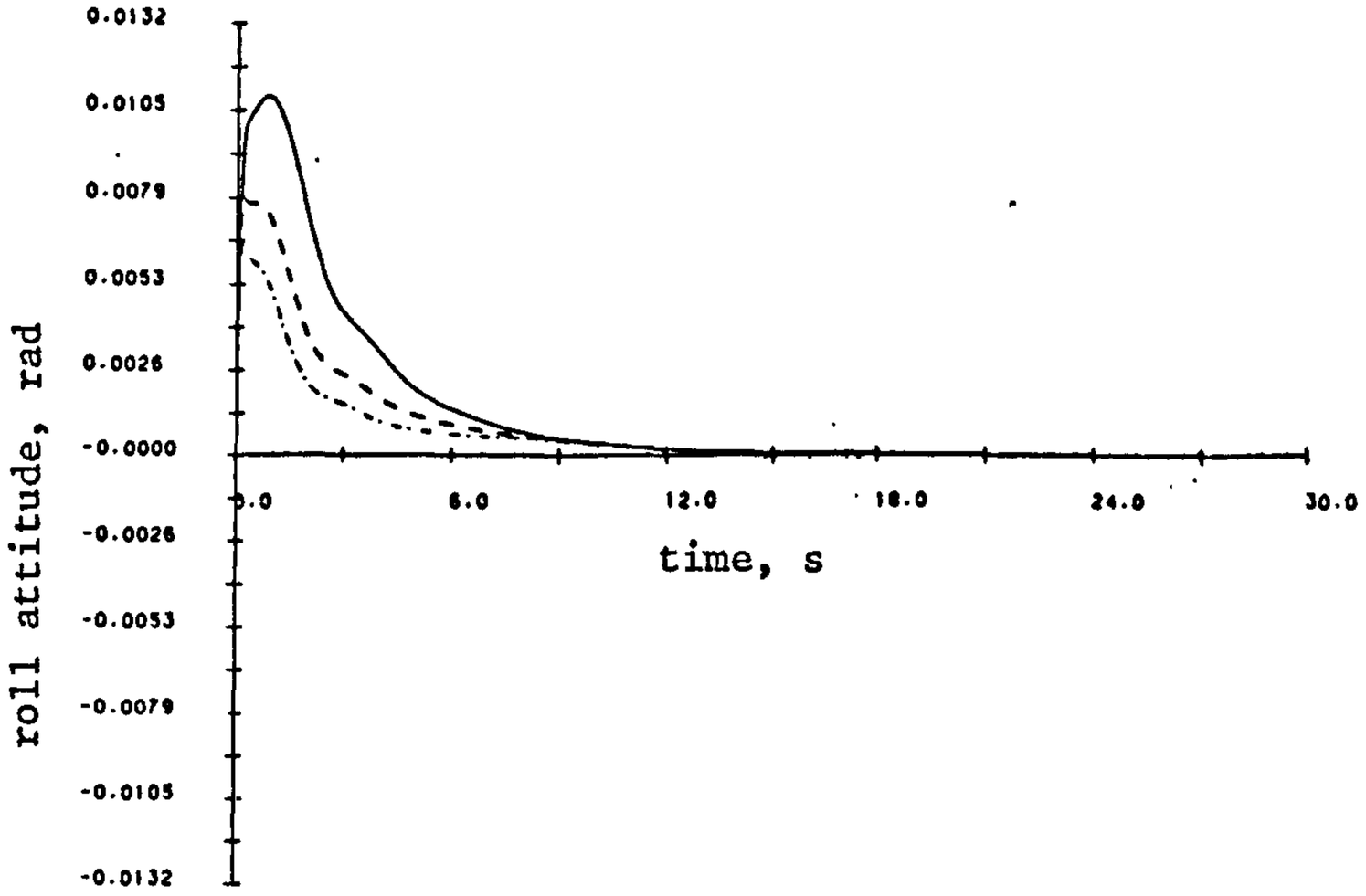
Fig 5.27 cont.



i)

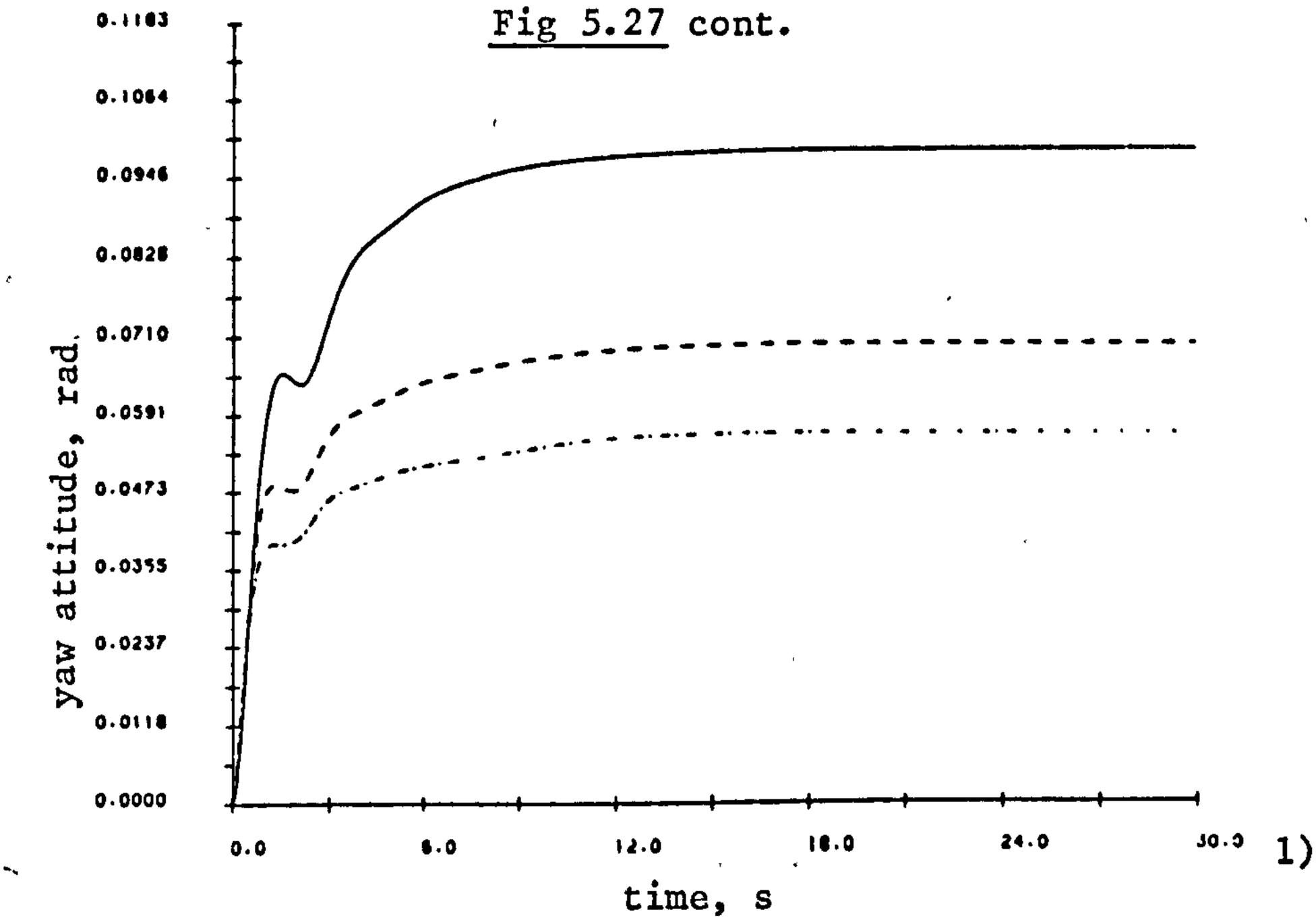


j)

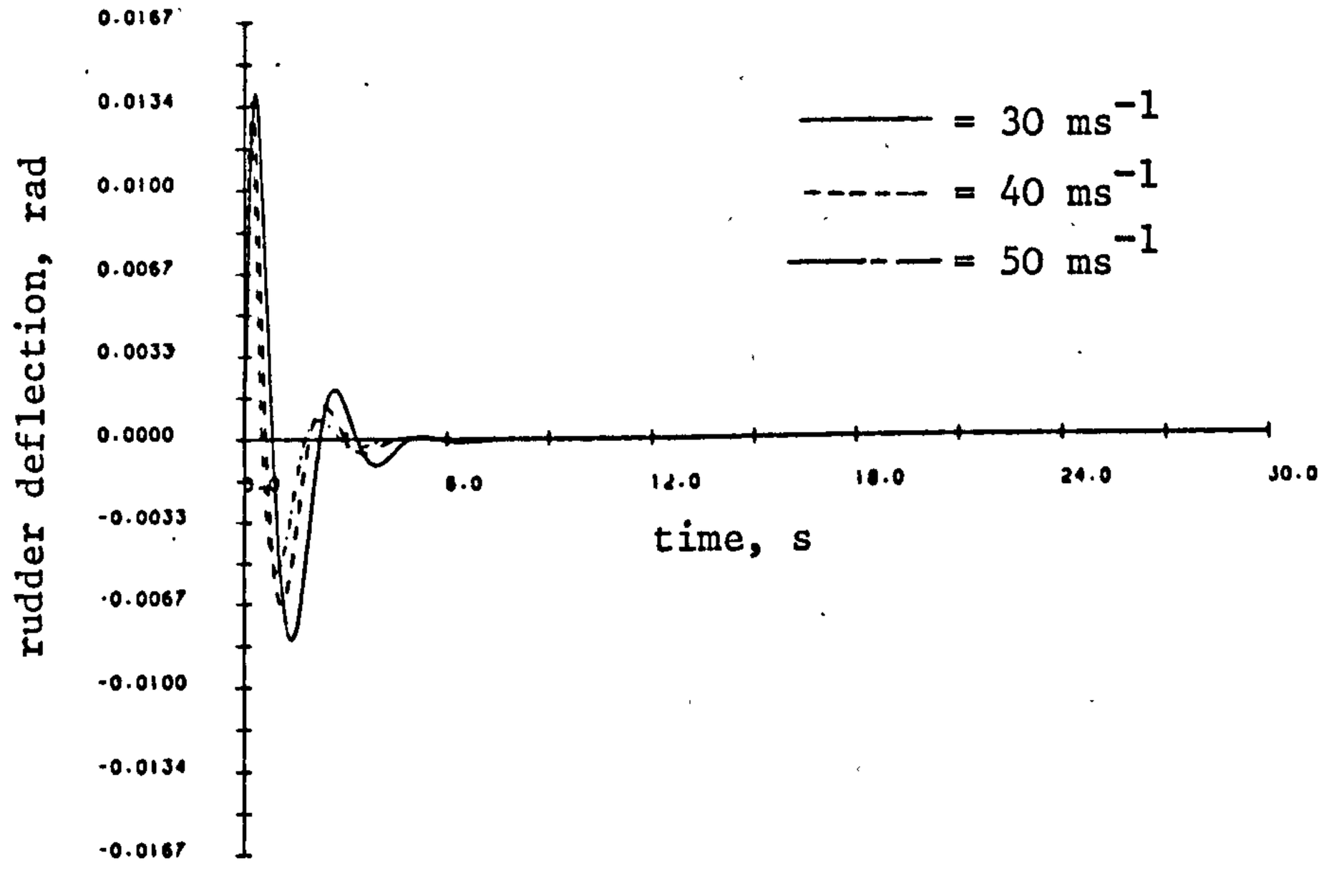


k)

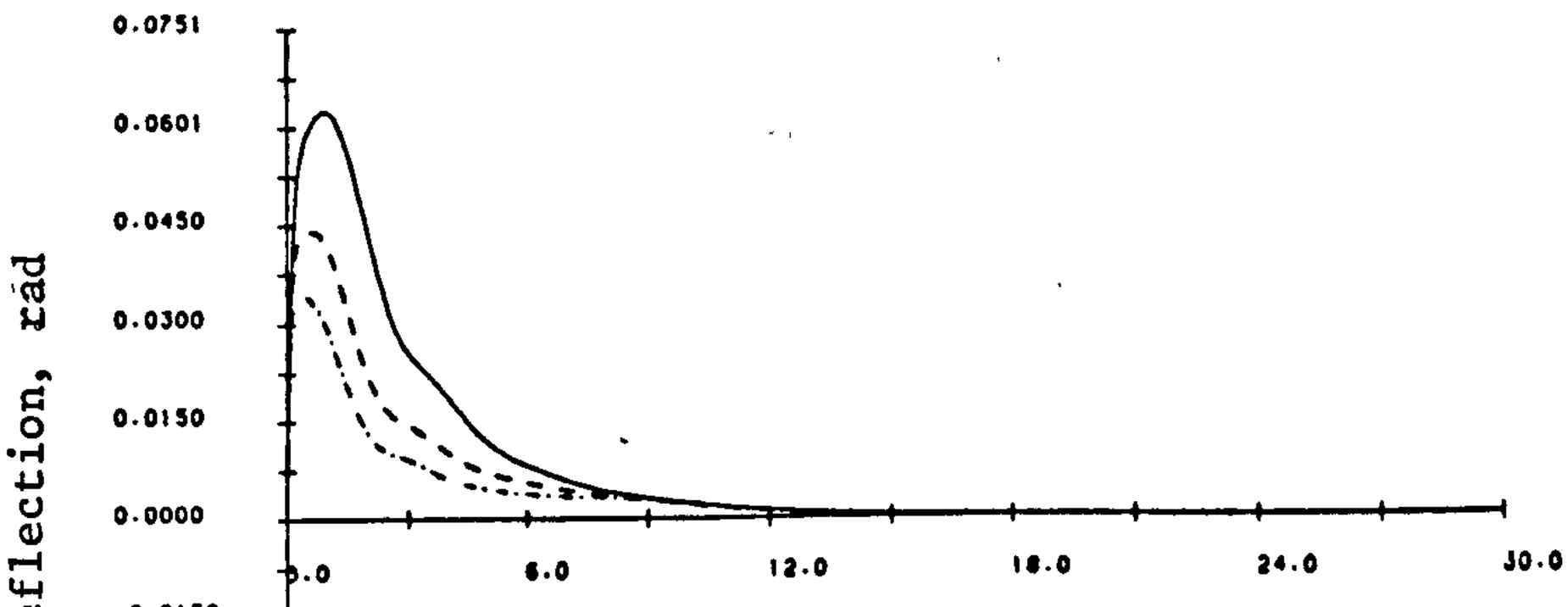
Fig 5.27 cont.



l)



m)



n)

term oscillation in q and the damping of this mode is poorer for larger initial airspeeds. The phugoid mode also shows strongly in the pitch, θ, height, h, forward velocity, u, and vertical velocity, w. The demands placed on the elevator are clearly small and the aircraft adopts a steady state elevator trim offset of approximately 8^{o}, as desired. With no height hold loop the pitch rate autopilot can do little other than improve the phugoid and short period characteristics. The aircraft will inevitably trade off forward speed against height for fixed elevator trim. The pitch rate loop does, however, provide reasonably stable short period and phugoid responses.

In the lateral axis, Fig. 5.27 h) to n), the provision of both roll and yaw damping yeilds a stable and rapid response in these loops. The initial roll rate, p, is thus rapidly removed with little overshoot and the roll angle, φ, transient is reduced to zero with a fairly long time constant, this being the now stable long term spiral mode. The yaw loop also provides a well damped and fast response in the Dutch roll mode as evidenced by the side velocity, v, and yaw rate, r, responses. The provision of spiral stabilisation in roll clearly assists the yaw loop to provide fast and stable Dutch roll responses. The yaw angle exhibits a steady-state offset since no attempt has been made to control yaw attitude directly. Actuator responses are small, Figs 5.27 m) and n), with the rudder dominantly controlling Dutch roll in yaw and the aileron dominantly controlling the roll channel.

5.7 Summary

This Chapter has dealt with the provision of improved aircraft performance by employing closed-loop feedback of primary aircraft flight variables to the control surfaces. The discussion has demonstrated that the problem can be treated as a number of single-input single-output control loops, each of which can be designed separately using classical frequency or s-domain linear analysis. The system performance may be improved in terms of stability, time

response, etc., by the inclusion of appropriate compensation in each of the loops, again designed using a classical linear systems approach. The need for gain-scheduling in such a scheme has also been demonstrated since many of the aircraft's aerodynamic derivatives change with flight condition. Such SAS systems can, and indeed do, provide acceptable performance from the airframe and are analytically a relatively simple design problem.

In the following chapters a demonstration of the way in which modern control theory may be applied to the aircraft problem is presented and the likely benefits of employing such control schemes described.

CHAPTER 6

Optimal Control and its Application

6.1 Introduction

In the following chapters it is intended to broaden the discussion of closed-loop control of aircraft in flight to include so-called modern control techniques. In this context state variable feedback techniques of various forms are investigated with application to the aircraft flight control system design problem. It is hoped that the use of such techniques will provide significant advantages compared with the classical approaches discussed in Chapter 5. In particular, the ease in handling the system's variations with changes in flight condition will be demonstrated. This is clearly an important design feature. The adaptive form of control has previously been considered for aircraft flight control applications (14,17-20), however, gain-scheduling along with state feedback should provide parameter insensitive control action. In addition, the reconstruction of unmeasurable states in the aircraft is important although this aspect has not been considered in the current work. In this Chapter the use of optimal control strategies will be investigated and the problems of using such schemes will be discussed.

6.2 Optimal Control

The classical control designs developed in the previous Chapter provided, to some extent, a system response which fulfils a given set of performance criteria. Principally, these criteria relate to the stability, response time, steady-state error, etc. of the closed-loop system. It has also been indicated that in aircraft flight control these criteria would normally be based upon considerations of the likely flight envelope in addition to the manoeuvre demands placed on the aircraft. The use of classical control techniques enables us to design controllers which, to some

degree, satisfy these performance criteria although some engineering trade off's may be necessary (110). The controller thus developed will normally be non-unique and will probably owe much to the skill and expertise of the designer. The fact that many aircraft fly very well with controllers designed using classical tools is therefore something of a tribute to the ingenuity and skill of the design engineer.

In recent years the problems of satisfying a given performance requirement with a unique controller has received considerable attention (21-30,135). This goal is desirable since such design techniques rely less on the engineering compromises than, for example, classical techniques do. It is to this end that much modern control theory development has been directed.

6.3 Performance and Objectives

The performance of any given system can be assessed by its ability to fulfil a chosen set of performance criteria. These criteria being defined by a consideration of the likely demands placed on the system. It is the function of any control system to modify the system performance such that these criteria are met more accurately. In classical control it is the designers task to evaluate the system performance and assess how well it fulfils the chosen criteria and subsequently modify the system, by the introduction of compensators, in order to improve its performance. To provide a more unified approach one may define performance criteria and employ mathematical techniques to produce a unique controller design capable of meeting these criteria exactly (109,111,132-134). This is fundamentally the approach of so-called optimal control techniques since an attempt is made to minimise a chosen performance measure as closely as possible. Classical designs are generally non-unique and may thus, in this sense, be considered sub-optimal. The price to be paid in order to achieve such optimal policies is that the controller requires much more information about the system than an equivalent classical design. The ability to provide

greatly improved performance is, however, leading to the adoption of optimal control in many areas (123,124).

An underlying concept common to all optimal control schemes is that of state feedback. This requires that all system states are measured and fed-back, via. some form of feedback gains, to the input. With the aircraft environment in mind it is often difficult, if not impossible, to sample some of the system states, e.g. thrust, sideslip, etc., although measurements of axial accelerations, body rates, etc. are commonly available. This problem may however be overcome by employing observer techniques. In the following discussion it is therefore assumed that all system states are measureable or, at worst, reconstructable.

The desirable objectives of any automatic flight control system (139) are summarised as follows :

i) To 'optimise' the aircraft by providing stability augmentation in addition to allowing precision manoeuvres to be executed.

ii) Reduction of the effects of external disturbances on the aircraft principally, gust alleviation and the reduction of the effects of velocity and flight configuration variations.

iii) Reduction of structural modes due to the non-rigid airframe.

iv) Robust control when faced often with very large changes in plant (aircraft) dynamic performance.

The classical control design of Chapter 5 sought to satisfy i), ii) and iv) using a fairly pragmatic approach. Using optimal control the above would be satisfied by careful choice of a performance index (PI). It is worth noting that by using state feedback the so called regulator problem is evolved in which each state is forced to zero. The problem encountered in i) is that many system states should not be forced to zero as some states might be required to follow a pilotic demand e.g. in a turn manoeuvre. This latter problem requires a tracking ability and the control design

is based on the more complex servomechanism or model-following control problems. Initially the application of optimal control and its likely performance is considered with reference to a particular example.

6.4 Optimal Control Theory (25,27,32,33)

Consider the n th. order linear system described by the state equations :

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) + G\,\underline{w}(t) \qquad - a)$$

$$\underline{y}(t) = C\,\underline{x}(t) + W\,\underline{v}(t) \qquad - b)$$

- 6.1

where $\underline{x}(t)$, $\underline{y}(t)$, $\underline{w}(t)$ and $\underline{v}(t)$ are state, output, input disturbance and output disturbance vectors of dimensions (n x 1), (m x 1), (ℓ x 1) and (p x 1) respectively. A, B, C, G and W are contant coefficient matrices of orders (n x n), (n x m), (m x n), (n x ℓ) and (p x m), the system having n states, m inputs, m outputs, ℓ input disturbances and p output disturbances. For deterministic systems the G and W matrices are zero giving :

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \qquad - a)$$

$$\underline{y}(t) = C\,\underline{x}(t) \qquad - b)$$

- 6.2

This is a proper system representation i.e. the output equation, 6.2 b), contains no contribution from the input vector, $\underline{u}(t)$. In the normal state feedback schemes we aim to provide control such that :

$$\underline{u}(t) = f(\underline{x}(t)) \qquad - 6.3$$

where the function f is chosen so as to minimise a performance index

$$J = \int_0^{t_f} h(\underline{x},\underline{u},t_f)\,dt \qquad - 6.4$$

The function h represents a given function of the states, $\underline{x}(t)$, inputs $\underline{u}(t)$ and the final time, t_f and requires to be chosen to reflect the constraints imposed by and on the system. We may for example require that the deviations from zero of some or all of the system states are penalised heavily i.e. a control action is initiated which will rapidly drive the state or states to zero. It would also be normal to ensure that excessive control actions are not demanded by weighting the controls, $\underline{u}(t)$, heavily. Since we require to minimise J it is clear that we normally drive both states and controls to zero. Such schemes are thus admirable for aircraft SAS systems where demand following is not required (regulatory action). Where, for example, the system must follow a pilotic demand other techniques may be used (see section 6.6).

The choice of h in equation 6.4 clearly affects the way in which the system responds. To provide analytically tractable solutions a quadratic form is often chosen for h thus :

$$J = \int_0^{t_f} (\underline{x}^t(t)\ Q\ \underline{x}(t) + \rho\ \underline{u}^t(t)\ R\ \underline{u}(t))\ dt \qquad - 6.5$$

where Q and R are (n x n) and (m x m) real symmetric weighting matrices and ρ is a scalar design parameter. Considerable freedom exists in the choice of Q and R although diagonal forms are often used giving $\underline{x}^tQ\underline{x}$ and $\underline{u}^tR\underline{u}$ the form of a weighted sum of squares. Fortunately, this choice of cost function, J, is not only mathematically tractable but also reasonably practical. The final time, t_f, can, for many problems including the aircraft, be chosen as infinity.

The use of this formulation gives rise to a control law of the form

$$\underline{u}(t) = - K\ \underline{x}(t) \qquad - 6.6$$

where the constant (m x n) matrix K is given by

$$K = P\ B\ R^{-1}\ /\rho \qquad - 6.7$$

and the (n x n) matrix P is the solution of the algebraic Ricatti equation

$$\dot{P} = Q + P\ A + A^T\ P - P\ B\ R^{-1}\ B^T\ P\ /\rho \qquad - 6.8$$

In the infinite time problem, $t_f = \infty$, the steady state solution of equation 6.8 is sought, i.e. with $\dot{P} = 0$. In most 'long' term control problems this is indeed the case but in fixed end point problems, e.g. target tracking, t_f must be made finite. Even with $\dot{P} = 0$ the solution of equation 6.8 is difficult for higher order systems (n>3) but several relatively simple methods are available for implementation on a digital computer (31,112).

In order to fully investigate the application of optimal control schemes of the form discussed above to the aircraft problem an interactive CAD package was developed. This package allows optimal feedback laws of the form of equation 6.6 to be evaluated and the time response of the resulting closed-loop linear system to be assessed. This is particularly valuable when chosing the elements of the Q and R weighting matrices of equation 6.5. The algorithm used in the package is more fully detailed in Appendix 7 but basically employs an eigenvector solution of the steady state Ricatti equation (6.8 with $\dot{P} = 0$). (Note that to guarantee a unique solution to 6.8 the (A,B,C) triple is assumed controllable and observable, R must be positive definite and Q at least positive semi-definite, symmetric).

To illustrate the application of the above synthesis techniques consider an optimal control design based on the Machan r.p.v. as described previously. The type of control initially considered is an optimal state regulator arranged such that all states are driven to zero subject to certain control constraints. In flight control this is often undesirable, however, considering only the lateral motion model and ignoring longitudinal/lateral cross-coupling then

it may be desirable to drive all lateral states to zero for SAS purposes. This is clearly not the case in longitudinal motion where pitch, height, etc. are required to have finite non-zero steady state values. Considering then the linearised lateral state space model for $33ms^{-1}$ stick fixed flight we have :

$$A_{la} = \begin{bmatrix} -0.277 & 0. & -32.9 & 9.81 & 0. \\ -0.1033 & -8.525 & 3.75 & 0. & 0. \\ 0.3649 & 0. & -0.639 & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \end{bmatrix}$$

$$B_{la} = \begin{bmatrix} -5.432 & 0. \\ 0. & -28.64 \\ -9.49 & 0. \\ 0. & 0. \end{bmatrix}$$

$$C_{la} = \begin{bmatrix} 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \end{bmatrix}$$

Note that, in this case, the actuator dynamics have been ignored.

The measurement vector is chosen as roll and yaw rates since these are normally available via. rate gyros. In order to derive a set of state feedback gains (K matrix of equation 6.6) we must initially choose the Q and R weighting matrices of equation 6.5.

Several techniques for deriving appropriate Q and R's to satisfy specified closed-loop modal responses have been proposed (22,34-38,113,114). In particular the work of Harvey & Stein (34) and Grimble (36) provide simple methods for deriving these weighting matrices. The eigenvalue/eigenvector assignment techniques of Harvey and Stein and others (28,37-44) will therefore initially be examined and then a somewhat simpler approach based on output assignment.

6.5 The Choice of Q and R

Eigenvalue/Eigenvector Techniques (22,34)

An inherent part of the optimal control design is the need to employ state feedback. It is this technique which gives rise to one of the virtues of optimal control in as much as full state feedback allows arbitrary closed-loop eigenvalue (pole) placement along with eigenvector assignment. This result was established by Moore (45,46). It is relatively simple to demonstrate this result by considering the m-input, m-output system of equation 6.2 a) and with full state feedback as in equation 6.6. If the closed-loop system has n distinct complex eigenvalues, λ_i, $1 \leqslant i \leqslant n$, and associated eigenvectors, $\underline{e}_i$, then by definition ;

$$(A + B K)\ \underline{e}_i = \lambda_i\ \underline{e}_i \qquad - 6.10$$

rearranging 6.10 gives ;

$$(\lambda_i I - A)\ \underline{e}_i = - B K\ \underline{e}_i$$

$$\text{thus } (\lambda_i I - A)^{-1} B\ \mu_i = \underline{e}_i \qquad - 6.11$$

where $\mu_i = - K\ \underline{e}_i$

For equation 6.11 to hold we assume λ_i is not an eigenvalue of the matrix A. It is therefore clear that the m vector μ_i acts so as to distribute the associated modal response, λ_i, between the states, the x's, and the outputs. A more rigorous proof of 6.11 is presented in reference (45). Further, it follows that

$$T(\lambda_i)\ \underline{v}_i = 0 \qquad i = 1,2,\ldots,n \qquad - 6.12$$

where $T(s) = (I - G(s))$ is the system return difference operator with $G(s) = - K (s I - A)^{-1} B$. This being the case then it is easily shown that

$$\mu_i = \underline{v}_i \qquad i = 1,2,\ldots,n$$

after appropriate normalisation.

Recall that it is required to evaluate values of Q and R in equation 6.5 which satisfy arbitrary closed-loop pole placement and eigenvector adjustment. The following Lemmas due to Harvey & Stein (34) are required :

Lemma 6.1 The quadratic function of equation 6.5 is expressed equivalently as

$$J = \int_0^{t_f} (\underline{y}^T \underline{y} + \rho \underline{u}^T R \underline{u})\, dt \qquad - 6.13$$

with $\underline{y}$ as in equation 6.2 b) defined by the (m x n) matrix C. Equivalence in this case denoting that the two criteria produce the same gain matrix, K, as solutions of their respective quadratic optimization problems. We note from this that only Q matrices of the form $C^T C$ need be considered. It is also possible, in general, to rearrange states and normalise C such that

$$C = W C_o \qquad - 6.14$$

where $C_o = [C_{11} \quad I_m]$ and W is a non-singular (m x m) matrix.

Lemma 6.2 Consider the linear quadratic regulator problem, equations 6.6, 6.7, 6.8, with criterion 6.13 normalised as in 6.14 and assume that

i) Rank (CB) = m ;
ii) the roots of det{C (s I - A)$^{-1}$ B} are distinct, have negative real parts and do not belong to the spectrum of A ;

then the optimally controlled system has the following properties :

i) Asymptotically finite modes :

As ρ -> 0 there are (n-m) eigenvalues of the form

$$s_i(\rho) \rightarrow s_i^o$$
$$|s_i^o| < \infty \qquad i=1,2,\ldots,n-m$$

with associated eigenvectors

$$\underline{e}_i(\rho) \rightarrow (s_i^o - A)^{-1} B \underline{v}_i^o$$

where s_i^o and $\underline{v}_i^o$ are defined by

$$C(s_i^o - A)^{-1} B \underline{v}_i^o = 0 \qquad - 6.15$$

ii) Asymptotically infinite modes

As $\rho \rightarrow 0$ there are m eigenvalues of the form

$$s_j(\rho) \rightarrow s_j^\infty / \sqrt{\rho}$$

$$|s_j^\infty| < \infty \qquad j=1,2,\ldots,m$$

with associated eigenvectors

$$\underline{e}_j(\rho) \rightarrow B \underline{v}_j^\infty \qquad - 6.16\ a)$$

where s_j^∞ and $\underline{v}_j^\infty$ are defined by

$$R = N^{-T} S^{-2} N^{-1} \qquad - 6.16\ b)$$

$$W^T W = (C_o B)^{-T} N^{-T} N^{-1} (C_o B)^{-1} \qquad - 6.17$$

with

$$N = [\underline{v}_1^\infty , \underline{v}_2^\infty , \ldots\ldots , \underline{v}_m^\infty]$$

$$S = \text{diag}\{s_1^\infty , s_2^\infty , \ldots\ldots , s_m^\infty\}$$

The s_j^∞'s are real and negative.

The above lead us to the following proposition. As $\rho \rightarrow 0$, then (n-m) eigenvalues of the closed-loop system will tend asymptotically to finite values, s_i^o. These values correspond to the 'transmission' zeros of the response matrix

$$\Phi(s) = C\ (s\ I - A)^{-1}\ B$$

Additionally, the associated eigenvectors ($\underline{e}_i^o$, $\underline{v}_i^o$) distribute the response between the states and outputs. Equation 6.15 then provides a complete specification for C_o via. 6.14. The asymptotically inifinite modes are associated with the remaining m eigenvalues and eigenvectors and the parameter ρ. Choice of these modal directions then yields the R and W matrices of equations 6.16 b) and 6.17.

The value of the ρ parameter thus allows a trade off to be made between control weighting and the degree to which the asymptotically finite modes are achieved.

The design technique thus relies on the development and construction of C_o in order to satisfy given closed-loop pole positions and eigenvector directions. In addition, the R and W matrices must be defined giving a complete solution to the Linear Quadratic Performance problem. It then remains to evaluate the K matrix by a suitable algorithm. A technique for the development of a suitable C_o matrix has been proposed by Harvey & Stein and essentially is as follows :

1) 'Choose' a B matrix structure such that B has the partitioned form

$$B = \begin{bmatrix} 0 \\ \hline B_{21} \end{bmatrix} \begin{matrix} (n-m) \\ m \end{matrix}$$

This is possible using a similarity transform from the original system into so called 'controllable' form with the (m x m) matrix B_{21}, non-singular.

Alternatively, the B_{21} rows may be chosen as the actuator rows for the particular problem. This is often possible in many control systems where the actuators can be considered essentially as input 'filters'. An advantage of such an approach is that the resulting actuator modes fall within the range space of B and hence form the asymptotically infinite modes (follows from equation 6.16 a)). The physical constraints placed upon actuator responses may thus be borne in mind when choosing the ρ parameter and the R and W matrices such that excessive demands are not placed on these devices and any cross-coupling exactly defined.

2) Select values for the (n-m) asymptotically finite modes, s_i^o, consistent with the design objectives. The associated eigenvectors must also be defined such that a desired distribution in the state space may be

achieved. It is implicitly assumed that the designer can specify these vectors but note that these objectives may not be achievable by linear quadratic design since we are constrained to eigenvectors which satisfy equation 6.11 viz.

$$\underline{e}_i^{o} = (s_i^{o}\, I - A)^{-1}\, B\, \underline{v}_i^{o}$$

It may also be that only a few components of the desired eigenvectors are specified, the rest being arbitrary. Given this and taking the desired eigenvector to be $\underline{e}_i^{*}$, the mode distribution in state space will be given by

$$\underline{x}(t) = \underline{e}_i^{*}\, e^{(s_i^{o}\, t)} \qquad i=1,2,\ldots,n-m$$

and if only some of the elements of the $\underline{e}_i^{*}$ are to be specified we may reorder $\underline{e}_i^{*}$ and partition to give

$$\{\underline{e}_i^{*}\}^{R_i} = \begin{bmatrix} \gamma_i^{*} \\ \hline v \end{bmatrix} \qquad - 6.18$$

where R_i indicates a row reordering, γ_i^{*}, is the specified sub-vector and the v's are the unspecified elements.

3) Determine the achievable set of eigenvectors $\underline{e}_i^{o}$ from those specified. Since we know from 6.11 that the achievable eigenvectors are given by

$$\underline{e}_i^{o} = (s_i^{o}\, I - A)^{-1}\, B\, \underline{v}_i^{o}$$

then using 6.18 we obtain

$$\begin{bmatrix} \gamma_i^{o} \\ \hline v \end{bmatrix} = \{(s_i^{o}\, I - A)^{-1}\, B)\}^{R_i}\, \underline{v}_i^{o}$$

$$= \begin{bmatrix} L \\ \hline M \end{bmatrix} \quad \underline{v}_i^{o} \qquad - 6.19$$

we are free to choose $\underline{v}_i^{o}$ so as to best 'fit' γ_i^{o} to γ_i^{*}. One possible solution is to approximate γ_i^{*} with a least squares fit corresponding to the normal equations :

$$\underline{v}_i^{o} = [(\bar{L})^{T}L]^{-1}(\bar{L})^{-T}\, \gamma_i^{*} \qquad - 6.20$$

This solution may be ill-conditioned for some systems and in these cases alternative techniques such as singular valued decomposition (22) may be more appropriate.

4) Form the (n x (n-m)) eigenvector matrix E from the $\underline{e}_i^o$'s such that

$$E = [\underline{e}_1^o, \underline{e}_2^o, \ldots\ldots, \underline{e}_{n-m}^o] \qquad - 6.21$$

and form the (n x n) projection matrix P via.

$$P = I_n - E[(\bar{E})^T E]^{-1}(\bar{E})^T$$

The P matrix is partitionable into the sub-matrices

$$P = \left[\begin{array}{c|c} P_{11} & P_{12} \\ \hline P_{12}^T & P_{22} \end{array}\right] \qquad - 6.24$$

where P_{11} is ((n-m) x (n-m))
P_{12} is ((n-m) x m)
P_{22} is (m x m).

5) The (m x n) C_o matrix is now obtained via.

$$C_o = [P_{22}^{-1} P_{12}^T \quad I_m] \qquad - 6.25$$

where it can be shown that the P_{22} matrix is non-singular for non-singular B_{21}.

It now remains to consider the m asymptotically infinite modes and select the eigenvalues and eigenvectors of the asymptotically infinite modes. Equations 6.16 b) and 6.17 then give the R and W matrices. The required Q is then given from equation 6.14 as

$$Q = C^T C = C_o^T W^T W C_o \qquad - 6.26$$

The above procedure whilst being fairly complex is easily undertaken on a digital computer and has several advantages, e.g.:

i) The eigenvalues and associated eigenvector directions uniquely specify the closed-loop system response.

Hence the designer can achieve decoupling of the modes in addition to providing pole placement.

ii) The asymptotically infinite modes may be assigned and manipulated. This may be of value since these modes may be made to correspond to the actuator dynamics and hence specific bandwidth and cross-coupling constraints may be included in the design.

iii) Unique values of the Q and R matrices are obtained in one step with little iterative design required.

A number of shortcomings are equally apparent viz. :

i) The system is assumed to be minimum phase and both controllable and observable (i.e. rank(CB) = m) ; these restrictions may not hold for many systems and the technique is thus not directly applicable. The case of r(CB) < m is mentioned by Harvey & Stein since this implies that there may be less than or greater than m asymptotically infinite modes and the first order asymptotic behaviour is not guaranteed. This detracts from the inherent visibility of the method.

ii) The technique relies on state feedback and hence the desired output map does not enter into the formulation. It may thus be difficult to guarantee the output response of the system.

iii) The design relies on the choice of (n-m) asymptotically finite eigenvectors of the closed-loop system. It may be possible, in some systems, to completely specify these but in many cases it may be difficult, particularly where output regulation is desired (see ii) above).

iv) The resulting structure of the Q and R matrices may be quite complex.

Despite these shortcomings the technique does allow unique specification for Q and R and is a powerful method of providing solutions to the LQP optimal control problem. In later sections some possible extensions into the output regulator problem are examined.

To provide an assessment of the method it will initially be applied to the design of a closed-loop SAS system for the lateral motion model of the Machan aircraft. It is required to demonstrate the likely performance of the resulting controller with regard to adequate Dutch roll damping, reduced roll/yaw cross-coupling and realisable actuator demands. The 'robustness' aspects of the resulting controller will also be investigated by applying the control to the non-linear model with varying airspeeds. Firstly, consider the linear design.

6.5.1 Lateral SAS Design

The lateral state space model for the Machan for $33ms^{-1}$ 'stick fixed' flight was given in equations 6.9. Note that the actuator dynamics have not been included however step i) in the design procedure requires that the B matrix has the form :-

$$B = \begin{bmatrix} 0 \\ \hline B_{21} \end{bmatrix} \quad \begin{matrix} ((n-m) \times m) \\ (m \times m) \end{matrix}$$

For the present purpose two first order actuators for the rudder and aileron have been introduced :

$$\frac{\tau_a}{\tau_d} = \frac{20}{(s + 10)} \quad ; \quad \frac{\zeta_a}{\zeta_d} = \frac{10}{(s + 5)} \qquad - 6.27$$

These are comparable with the actuator dynamics included in the non-linear model which has unity gain first order actuators of time constant 1/20 and 1/10 s for the rudder and aileron, respectively. The modified state matrices are thus :

$$A'_{la} = \begin{bmatrix} -0.277 & 0. & -32.9 & 9.81 & -5.432 & 0. \\ -0.1033 & -8.325 & 3.75 & 0. & 0. & -28.64 \\ 0.3649 & 0. & -0.639 & 0. & -9.49 & 0. \\ 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & -10. & 0. \\ 0. & 0. & 0. & 0. & 0. & -5. \end{bmatrix}$$

$$B'_{la} = \begin{bmatrix} 0. & 0. \\ 0. & 0. \\ 0. & 0. \\ 0. & 0. \\ 20. & 0. \\ 0. & 10. \end{bmatrix} \qquad - 6.29$$

In this form the (A,B) pair is controllable. Since we are dealing with state feedback, as opposed to output feedback we shall assume that a C matrix can be chosen which makes the (A,C) pair observable with r(CB) = m, and m=2 in this case. Note that this choice of C will not necessarily be that resulting from the physical measurement system and that C_o must be given by

$$C = W\, C_o$$

with W a non-singular (m x m) matrix as in equation 6.17. This implies that the measurement vector, $\underline{y}$, may contain combinations of the state responses and may not necessarily provide 'good' output regulation, as noted earlier, although state regulation will be 'optimal'.

Given that the system is now in the desired controllable form we must choose values for the (n-m) (4 in this case) asymptotically finite eigenvalues and eigenvectors. To do this we shall, for convenience, adopt the approach of Harvey & Stein and consider the desirable closed-loop mode distribution from MIL spec. F8785B (47) for the lateral dynamics of

Roll subsidence	-4.0	
Dutch roll mode	-0.63 + 2.42j -0.63 - 2.42j	- 6.29
Spiral mode	-0.05	

(Note that this military specification has been superseded by MIL spec. F8785C (1980), the principal change being the provision of improved Dutch roll damping. The minimum acceptable damping ratio for this mode being raised from

0.25 to 0.4). The desired eigenvectors for the above modes must also be defined. For this particular example the elements of the desired eigenvectors are easily defined since the states are related to physical variables within the system whose modal coupling can be defined. For example, consider the roll subsidence mode. In this case the mode should appear dominantly on roll rate and not on yaw rate or side velocity. Banking manoeuvres should thus not incur yaw or side velocity penalties. Hence turn entries, which are normally initiated by banking, will not give rise to side velocities. The ordering of the states in equation 6.28 is as follows :

x_1	-	v	side velocity
x_2	-	p	roll rate
x_3	-	r	yaw rate
x_4	-	φ	roll angle
x_5	-	τ	rudder deflection
x_6	-	ζ	aileron deflection

hence a suitable choice of eigenvector for the subsidence mode would be :

$$\underline{e}_1^* = [0 \quad 1 \quad 0 \quad v \quad | \quad v \quad v]$$

where the v's indicate that the magnitude of the element is unimportant. The corresponding eigenvalue is s_1^o=-4.0. Note that the eigenvector is effectively normalised to give a '1' in the appropriate mode.

The Dutch roll mode consists of a complex pair of eigenvalues correspondingly we would anticipate two complex conjugate eigenvectors to be required to define the modal coupling. Dutch roll is a phenomenon associated with the v,r sub-system of the lateral equations (see section 2.2.2). Consequently, we require no coupling of Dutch roll into either roll rate or roll angle. Considering first the real part of the eigenvector and remembering that the vectors occur in conjugate pairs, then a desired eigenvector would be :

$$\underline{e}_2^* = [1 \quad 0 \quad v \quad 0 \quad | \quad v \quad v]$$

and the corresponding imaginary part would be

$$\underline{e}_3^* = [v \quad 0 \quad 1 \quad 0 \mid v \quad v]$$

again the v's denote arbitrary elements. This eigenvector pair constitutes the real and imaginary parts of the eigenvector associated with one of the Dutch roll modes, e.g. $s_2^o = -0.63 + 2.42j$, the conjugate eigenvector is then associated with the conjugate eigenvalue, i.e. $s_3^o = -0.63 - 2.42j$. Note here that the Dutch roll does not couple into the roll rate, p, and bank angle, φ, states, as desired. This choice of eigenvectors may not be an exactly achievable set

The final mode, the spiral, is associated with roll angle only and should not appear on side velocity in order to avoid sideslip in steady turns. The mode may, of course, appear in yaw and roll rates. The eigenvector to achieve this would be

$$\underline{e}_4^* = [0 \quad v \quad v \quad 1 \mid v \quad v]$$

The associated eigenvalue being $s_4^o = -0.05$. Note that in all cases the mode will occur on the actuators since we are using these states to control the mode.

On the basis of these intuitive arguments a set of desirable eigenvectors associated with the specified eigenvalues has thus been derived. It now remains to evaluate the achievable eigenvectors which best fit those desired. Using the results of equations 6.19 and 6.20 we can obtain the following achievable eigenvectors ($\underline{e}_i^{o}$'s) :

$$\underline{e}_1^o = [0.007 \quad 0.995 \quad -0.069 \quad -0.249 \quad -0.0243 \quad -0.159]$$

$$\underline{e}_2^o = [0.995 \quad -0.0016 \quad -9.145 \quad 0.002 \quad -0.209 \quad -1.2 \quad]$$

$$\underline{e}_3^o = [124.99 \quad -0.0047 \quad 1.0064 \quad -0.0002 \quad 2.472 \quad -0.318]$$

$$\underline{e}_4^o = [0.0 \quad -0.05 \quad 0.3 \quad 1.0 \quad -0.0187 \quad 0.054]$$

Note that in all cases the objectives have been closely achieved. Some iteration was required in order to 'fit' the elements of the complex eigenvector associated with the Dutch roll mode due to the two degrees of freedom available in the choice of the v's in rows 1 and 3.

The E matrix of equation 6.21 is now formed and hence C_o derived through the partitioned projection matrix, P, of equation 6.24. This gives

$$C_o = \begin{bmatrix} -0.0196 & 0.0297 & -0.25 & 0.0277 & 1. & 0. \\ 0.0036 & 0.149 & -0.13 & -0.0073 & 0. & 1. \end{bmatrix} \quad - 6.30$$

The final step in the design is to choose the asymptotically infinite modes and directions. Recall that in the example considered here these modes are associated with the actuators. It is required to decouple these modes asymptotically such that at high gains no aileron to rudder (or vice-versa) coupling occurs. It is desirable also to maintain a 2:1 ratio of aileron to rudder bandwidth as dictated by the physical system (recall the bandwidth ratio is 20:10). To this end choose :

$$\underline{e}_1^{\infty} = [0 \quad 0 \quad 0 \quad 0 \mid 1 \quad 0]$$

$$\underline{e}_2^{\infty} = [0 \quad 0 \quad 0 \quad 0 \mid 0 \quad 1]$$

and the ratio $s_1^{\infty} : s_2^{\infty} = 2:1$, say $s_1^{\infty} = 1$, $s_2^{\infty} = 0.5$. Thus from equation 6.16 a), 6.16 b) and 6.17

$$\underline{v}_1^{\infty} = [1/20 \quad 0]$$
$$\underline{v}_2^{\infty} = [0 \quad 1/10]$$

and
$$N = \begin{bmatrix} 1/20 & 0 \\ 0 & 1/10 \end{bmatrix}$$

$$S = \text{diag}\{s_1^{\infty}, s_2^{\infty}\} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$R = N^{-T} S^{-2} N^{-1} = 400\ I_2$$

and
$$W^T W = (C_o B)^{-T} N^{-T} N^{-1} (C_o B)^{-1} = I_2$$

since $W^TW = I_2$ the Q matrix becomes

$$Q = C_o{}^T C_o$$

and this gives

$$Q = \begin{bmatrix} 0.0004 & -0.00005 & 0.00002 & -0.00057 & -0.0196 & 0.0036 \\ -0.00005 & 0.023 & -0.02 & -0.00026 & 0.0297 & 0.149 \\ 0.00002 & -0.02 & 0.0178 & 0.00026 & -0.025 & -0.131 \\ -0.00057 & -0.00026 & 0.00026 & 0.00082 & 0.0277 & -0.0073 \\ -0.0196 & 0.0297 & -0.025 & 0.0277 & 1. & 0. \\ 0.0036 & 0.149 & -0.131 & -0.0073 & 0. & 1. \end{bmatrix} \quad - 6.31$$

With the Q and R matrices thus finalised the ρ parameter is now chosen to satisfy the achievable actuator bandwidths of 20 and 10 rads s^{-1}. This is considered best done interactively by evaluating the closed-loop eigenvalues for a range of ρ values. In this case a computer package was written (see Appendix 7) which allows this interactive design to be achieved and a value of ρ of 0.00275 gives closed-loop actuator poles of -13.0 and -21.2 with the remaining closed-loop poles at

$$-0.81 \pm 2.72j, \quad -4.39, \quad -0.1$$

Note that these are close to the desired values.

The state feedback gains for the case of $\rho = 0.00275$ are

$$\begin{matrix} \tau \\ \xi \end{matrix} \begin{bmatrix} 0.016 & -0.025 & 0.0432 & -0.023 & -0.6 & -0.0211 \\ 0. & -0.057 & 0.0724 & 0.035 & -0.01 & -0.413 \end{bmatrix}$$

these being evaluated from the infinite time solution of the algebraic Ricatti equation (equation 6.8).

The above state feedback gains were used with a linear model in order to assess the degree of modal decoupling and responsiveness achieved. Figs. 6.1 a) to f) give the state responses for initial conditions of 0.5 rad s^{-1} in roll rate (p) and 0.1 ms^{-1} in side velocity (v). It should be noted that :

i) The roll rate response has the desired first order transient with approximately a 1/4 second time constant. There is little evidence of Dutch roll cross-coupling.

ii) The v,r responses are dominated by the Dutch roll mode which has approximately the desired damping of 0.25 and period of around 2.5 s. Both responses contain evidence of the long term spiral mode although this is far less evident in the v response.

iii) The roll response comprises the rapid roll rate transient along with the spiral mode which for this set of gains has a time constant of roughly 10 s. There is little evidence of Dutch roll on this response.

iv) The actuator responses are formed from a mixture of the modes, the aileron being the dominant control in the roll axis whilst the rudder, with more of the Dutch roll mode evident, is dominantly controlling the yaw/sideslip axis. This decoupling is as desired although incomplete due to actuator dynamic constraints and consequent use of relatively low gains. The actual demands placed on the actuators are within the physical limitations of the devices, however.

It is of interest to note that the Q matrix of equation 6.31 is close to that obtainable by choosing

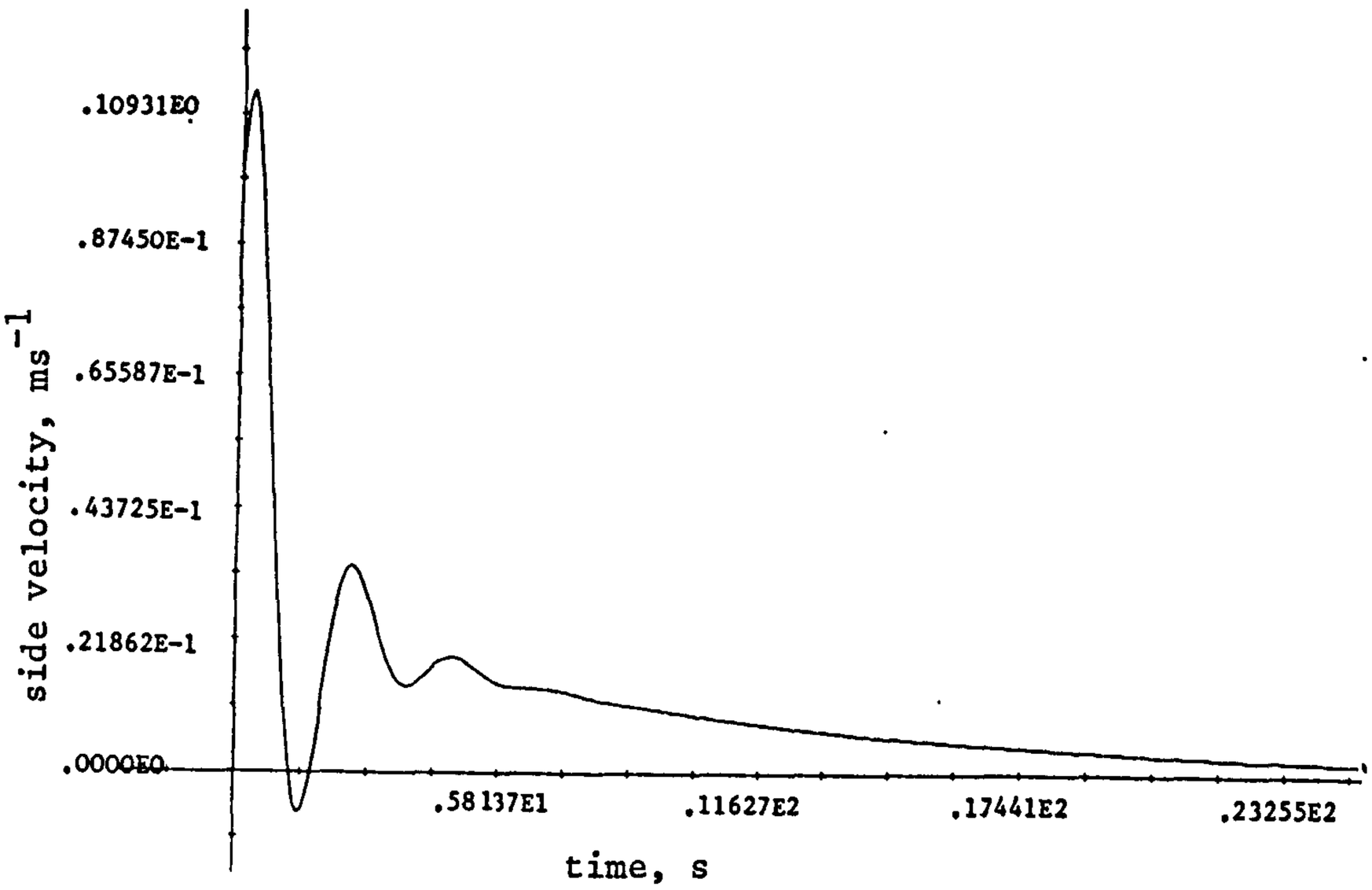
$$Q = C^T C$$

with the elements of C arranged so as to pick off the actuator states only viz.:

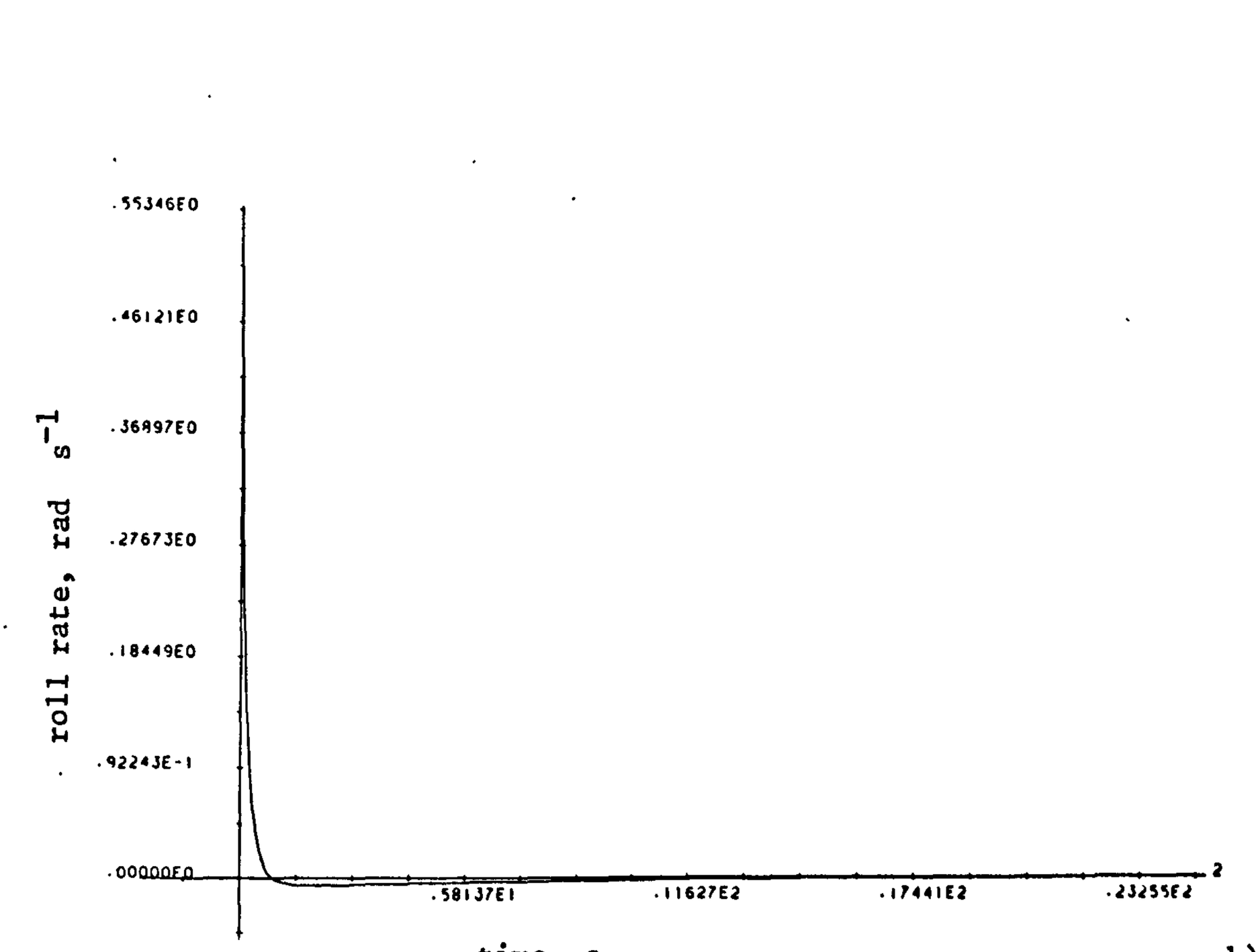
$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

the elements of the Q in 6.31 being << 1 except for elements (5,5) and (6,6). This simple choice of Q yields closed-loop eigenvalues which are close to those obtained above and is investigated further in the next section. Note, however, that this choice of C implies that the asymptotically finite eigenvectors lie within the null space of C whilst the

Fig 6.1 State Responses for Linear Lateral Model
(linear LQP controller, . 33 ms^{-1} airspeed)

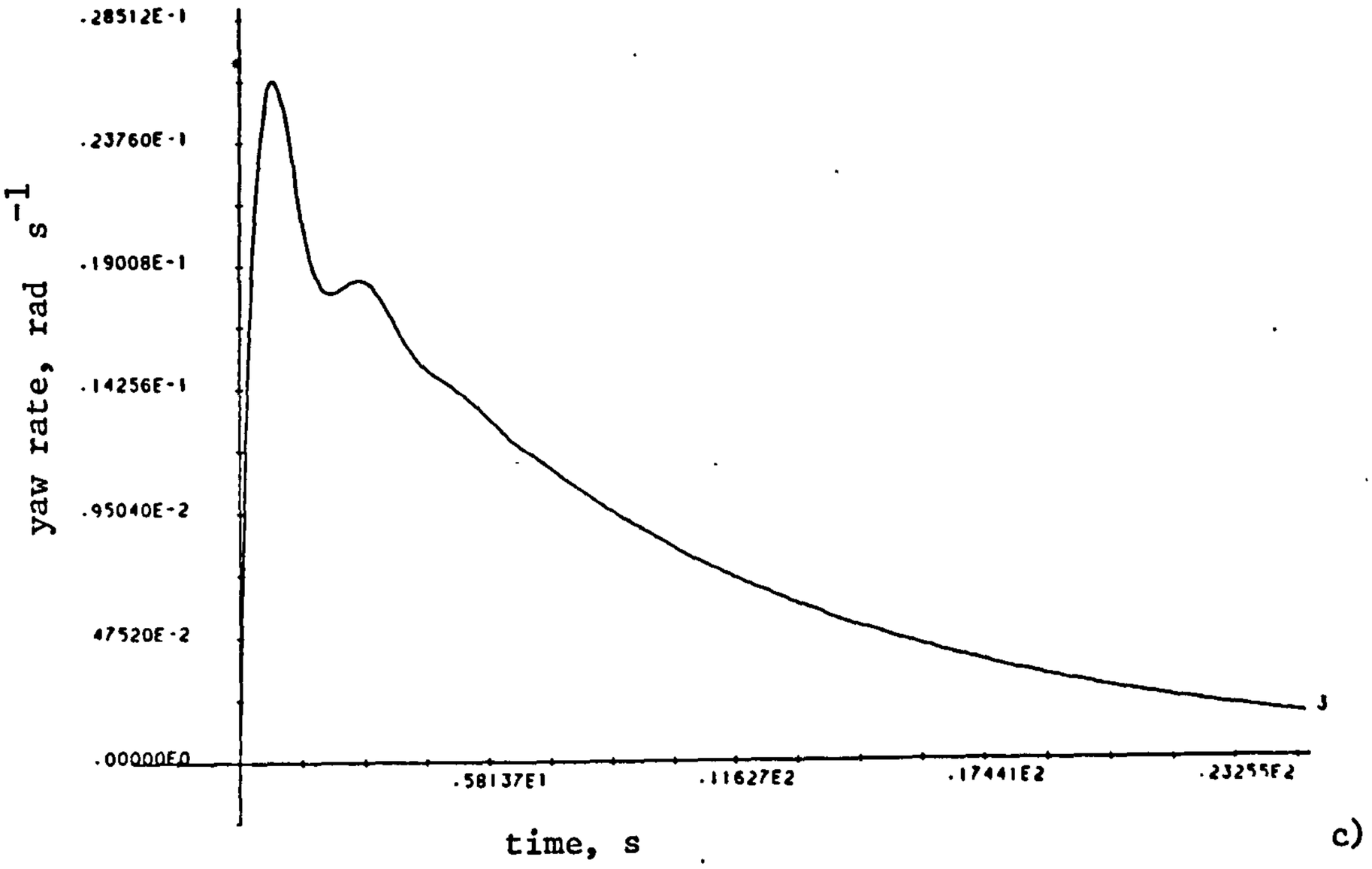


a)

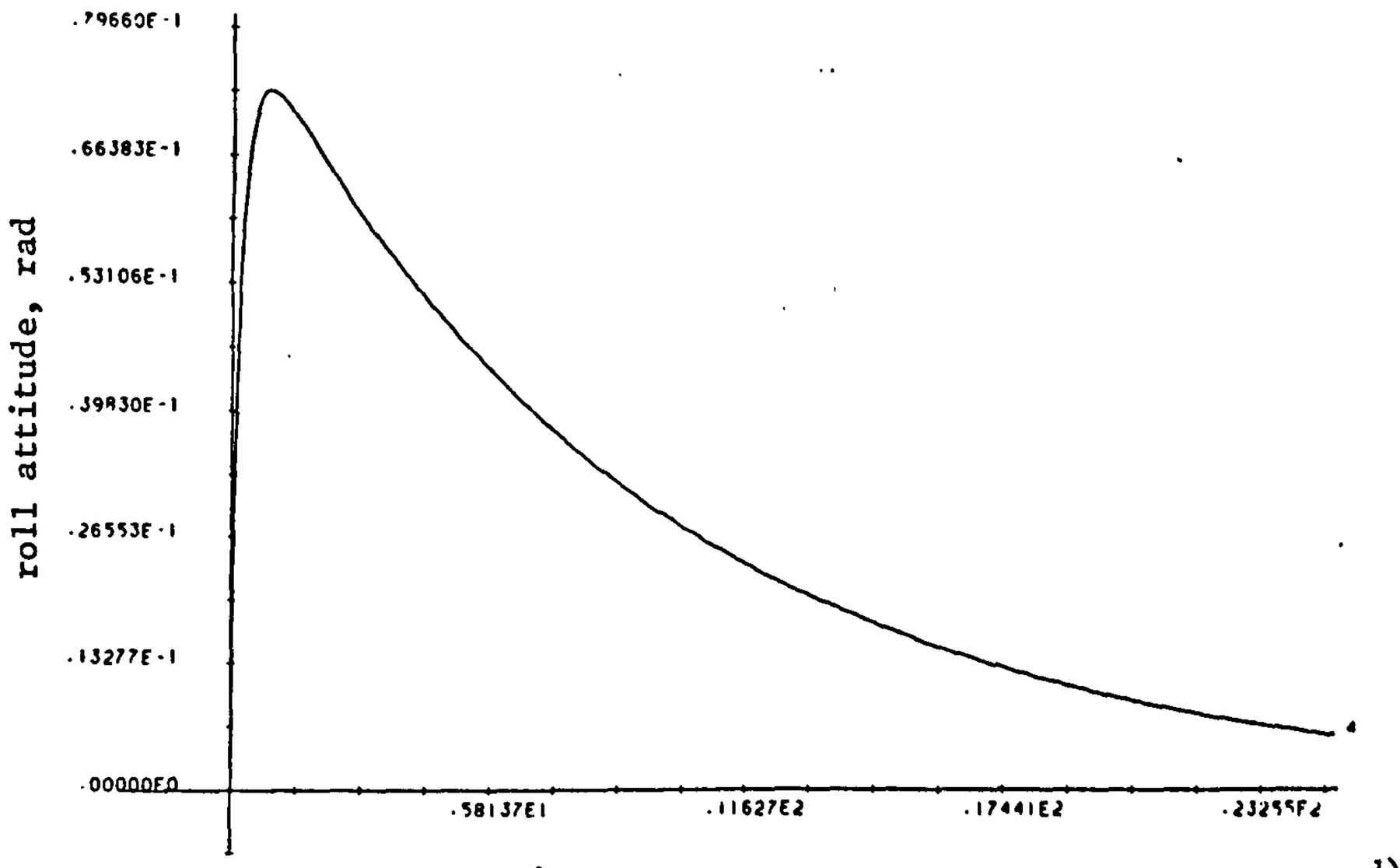


b)

Fig 6.1 cont.

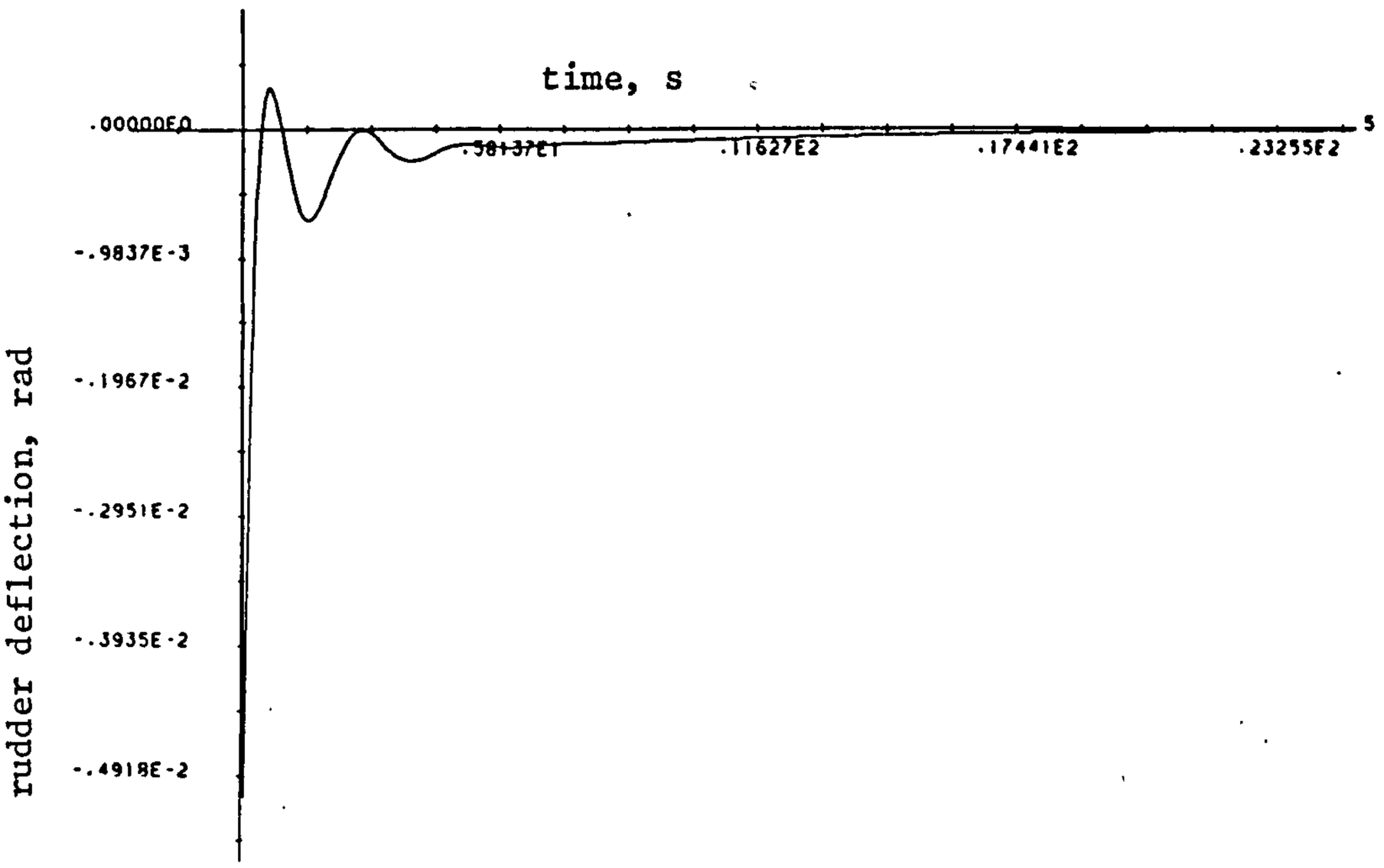


c)

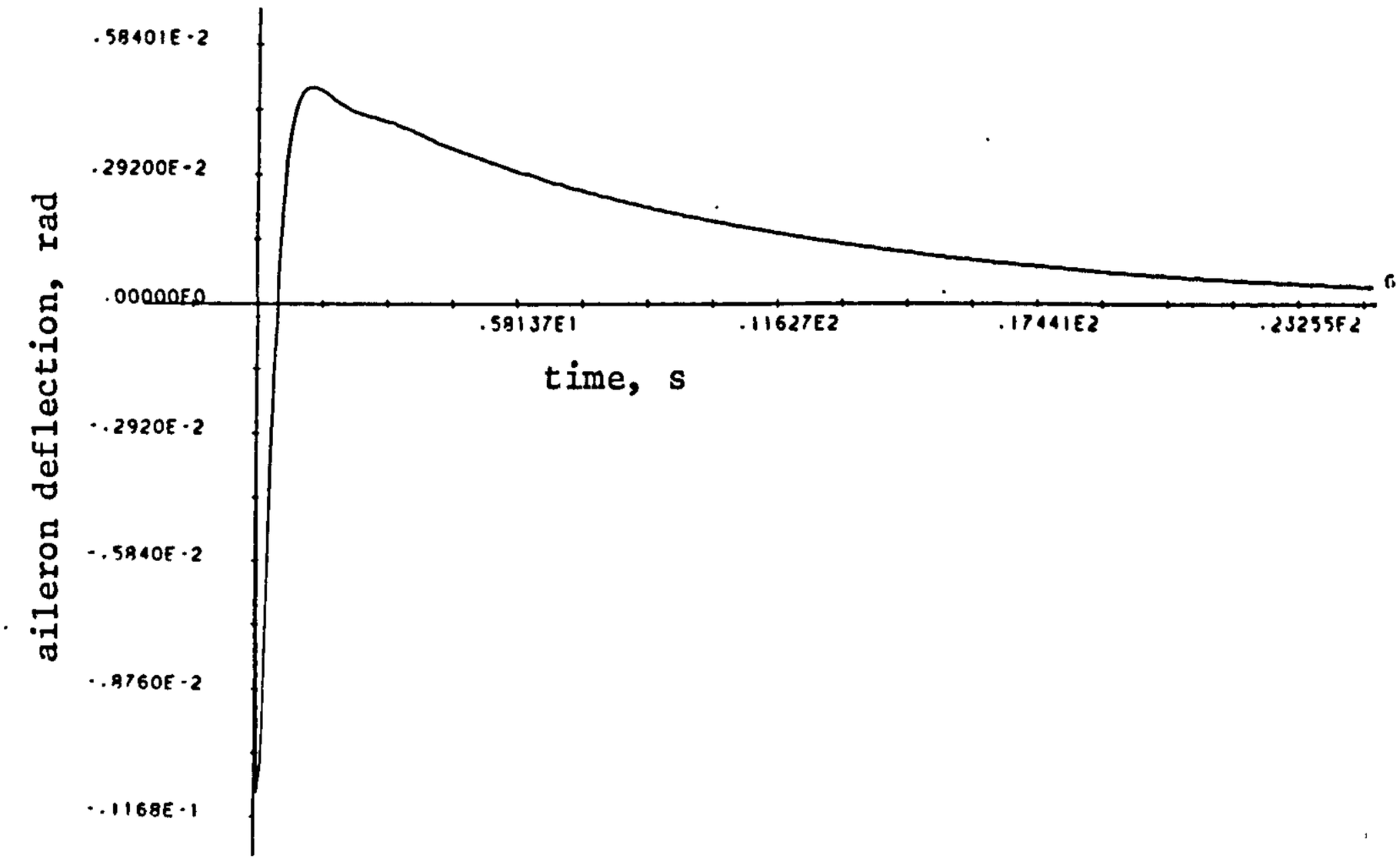


d)

Fig 6.1 cont.



e)

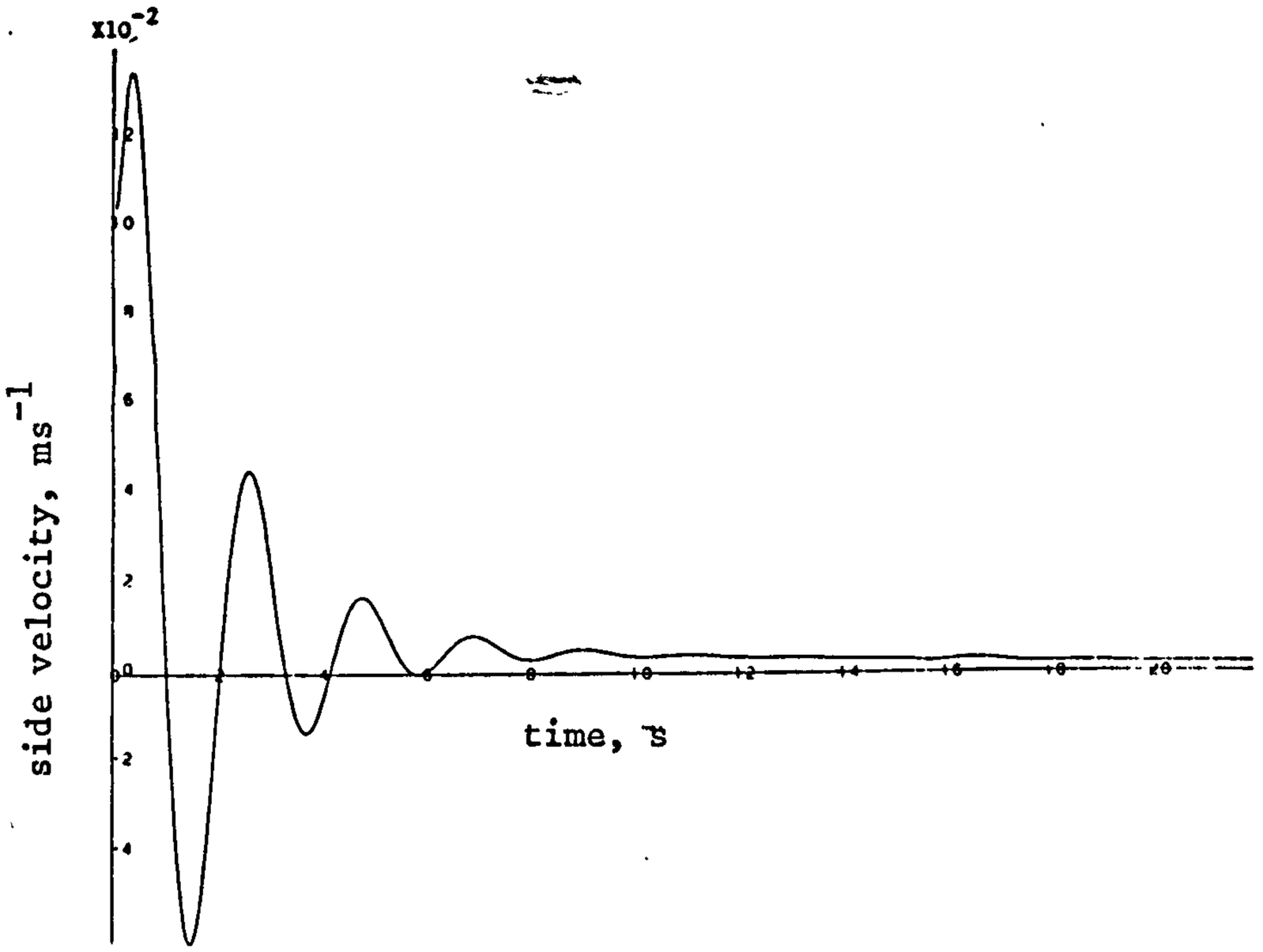


f)

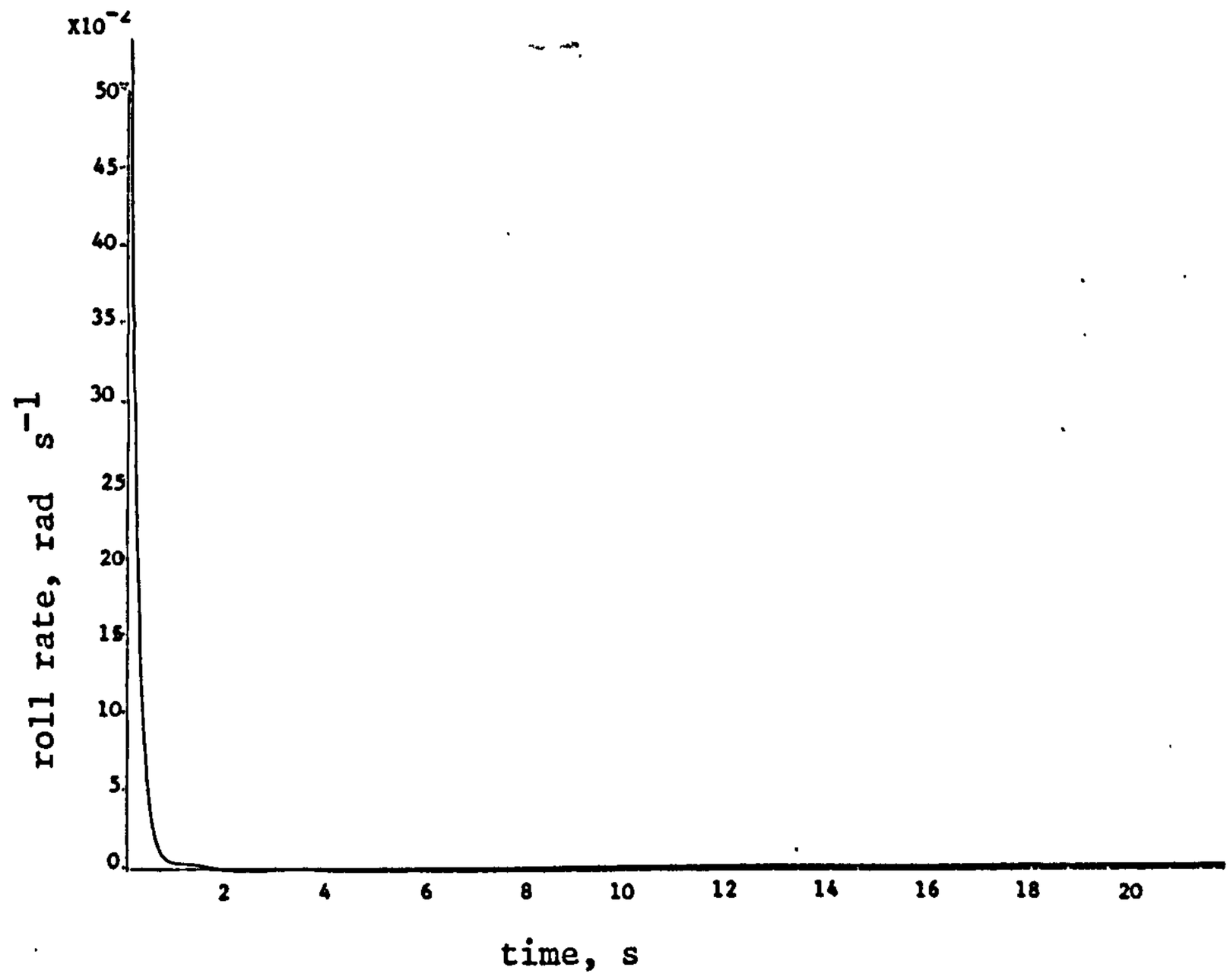
asymptotically infinite eigenvectors lie in the range space of C. This indicates that for the asymptotic case the output response will be dominated by the infinite modes which in this case would be the actuators. One may thus arrange that desired asymptotic output directions are achieved by suitable choice of the m asymptotically infinite system eigenvectors. The consequences of this observation will be discussed below.

One aspect of this design not yet considered is its robustness, i.e. how the modal distribution and couplings change as a result of changes in the system. As was demonstrated in Chapter 5 the aircraft model parameters may change drastically as the aircraft flight configuration changes. The Q and R matrices derived above lead to a set of state feedback gains which are valid for the nominal 'stick fixed' 33 ms^{-1} linear model considered and yield the desired mode distributions as has been demonstrated. This set of gains may clearly not provide the same modal distribution if, for example, the model parameters change as a result of changing airspeeds, etc. It may equally be unrealistic to expect the state feedback law to perform well when faced with the non-linear system. The robustness aspects of LQP controllers have been investigated extensively. In particular the results of Anderson and Moore (27) demonstrate that single-input LQP regulators have $\pm$ 60° phase margin, infinite gain margin and 50% gain reduction tolerance. The work of Safonov et al (23) has shown that these results may be extended to the multi-input case. It may thus be concluded that LQP control designs can exhibit improved robustness properties when compared with say a classically designed single-loop controller. To assess how well the above designed controller performs in terms of its robustness, the control was implemented on the non-linear simulation of the Machan as described in Chapter 3. The lateral motions only are considered, the longitudinal motions being held sensibly constant. For the case of 33 ms^{-1} airspeed and initial conditions of 0.5 rads s^{-1} roll rate and 0.1 ms^{-1} in side velocity, the state responses are shown in Figs. 6.2 a) to f). These may be compared with Figs. 6.1 a) to f) for the linear model. From

Fig 6.2 State Responses for Non-linear Lateral Model
(linear LQP controller, 33 ms^{-1} airspeed)

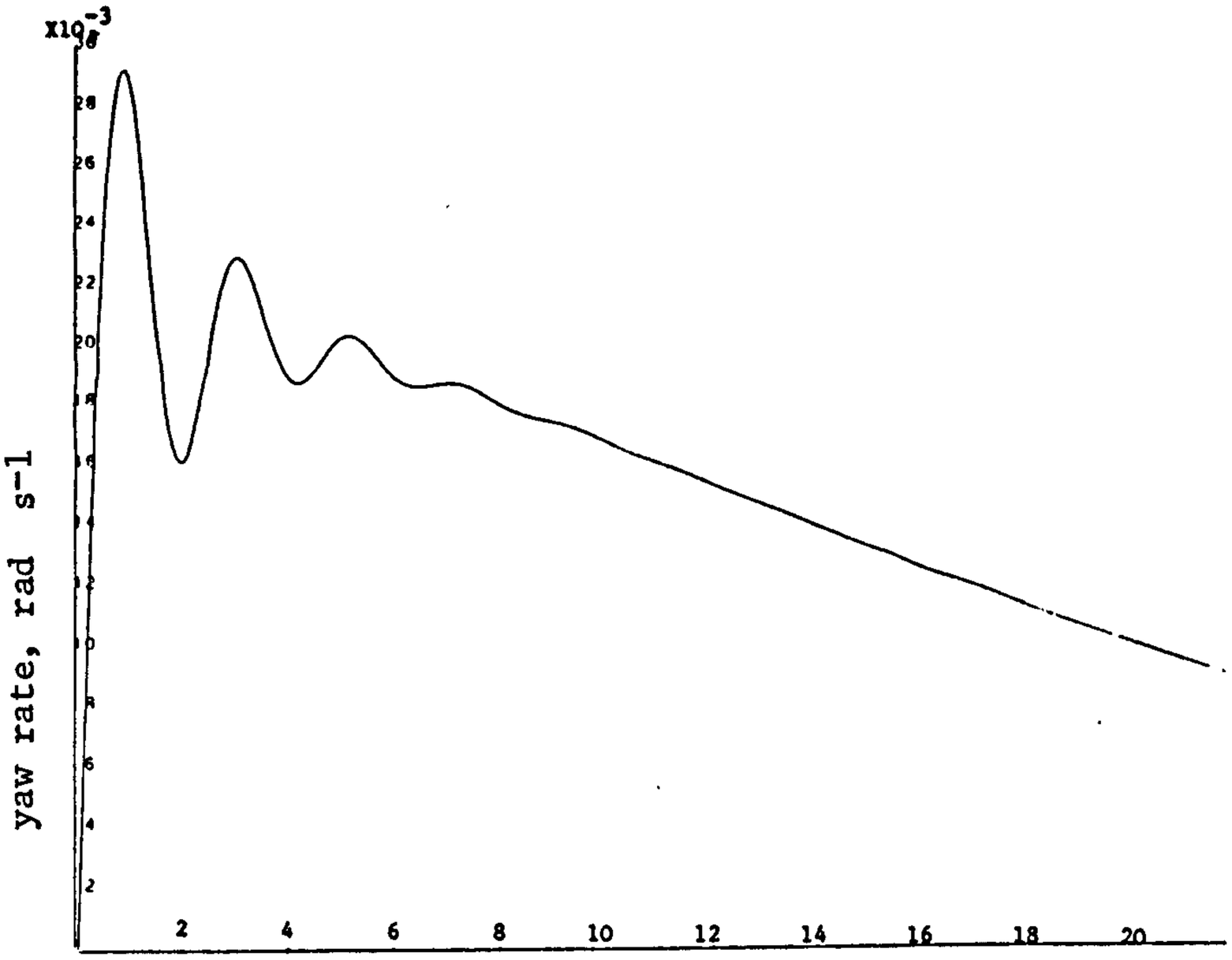


a)

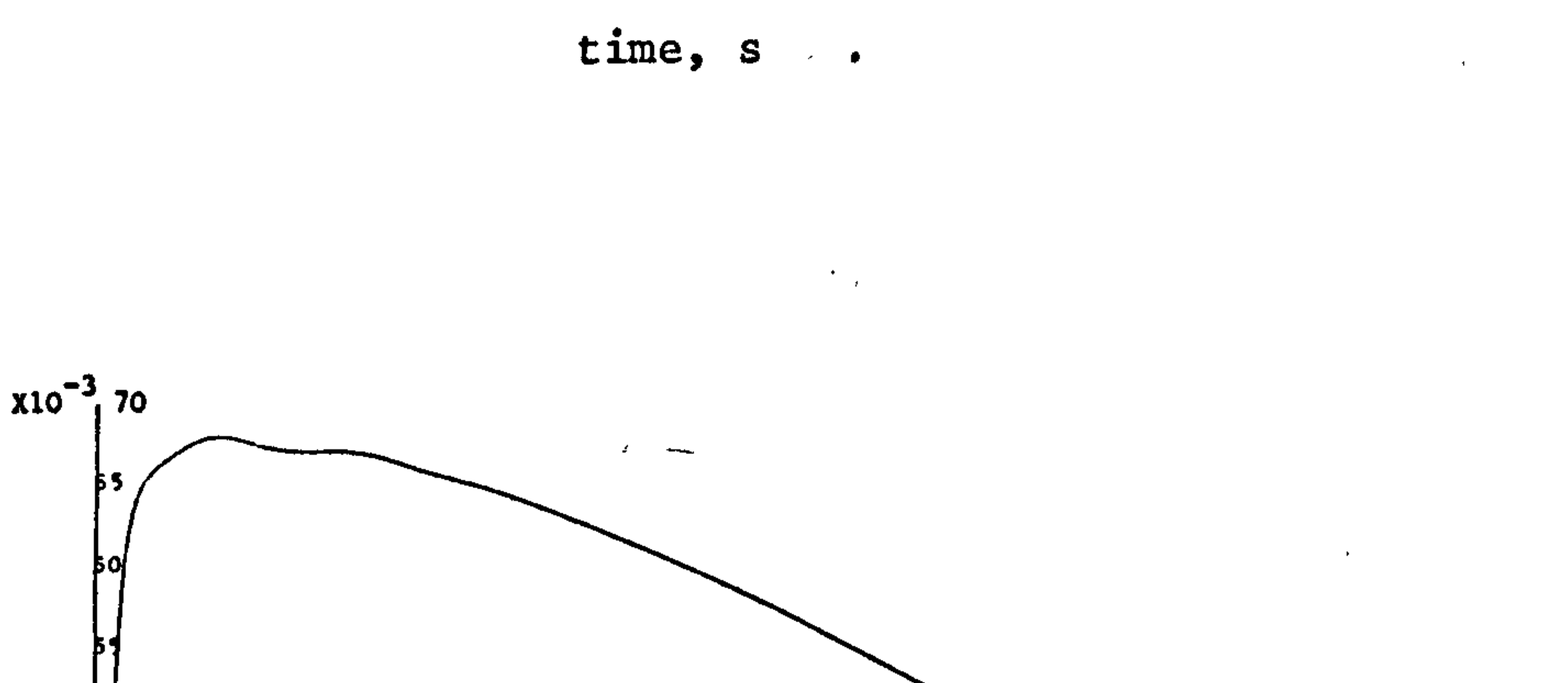


b)

Fig 6.2 cont.

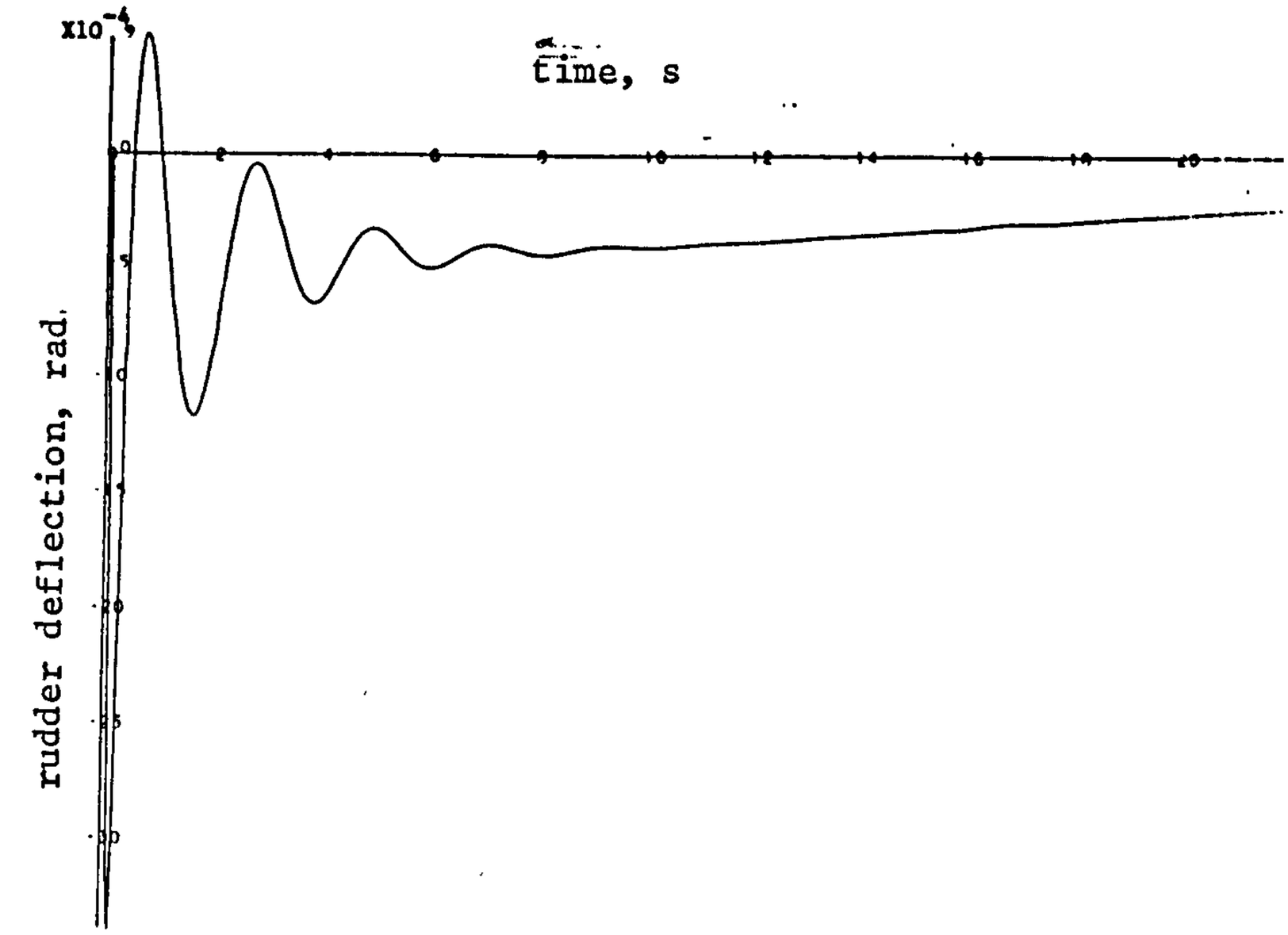


c)

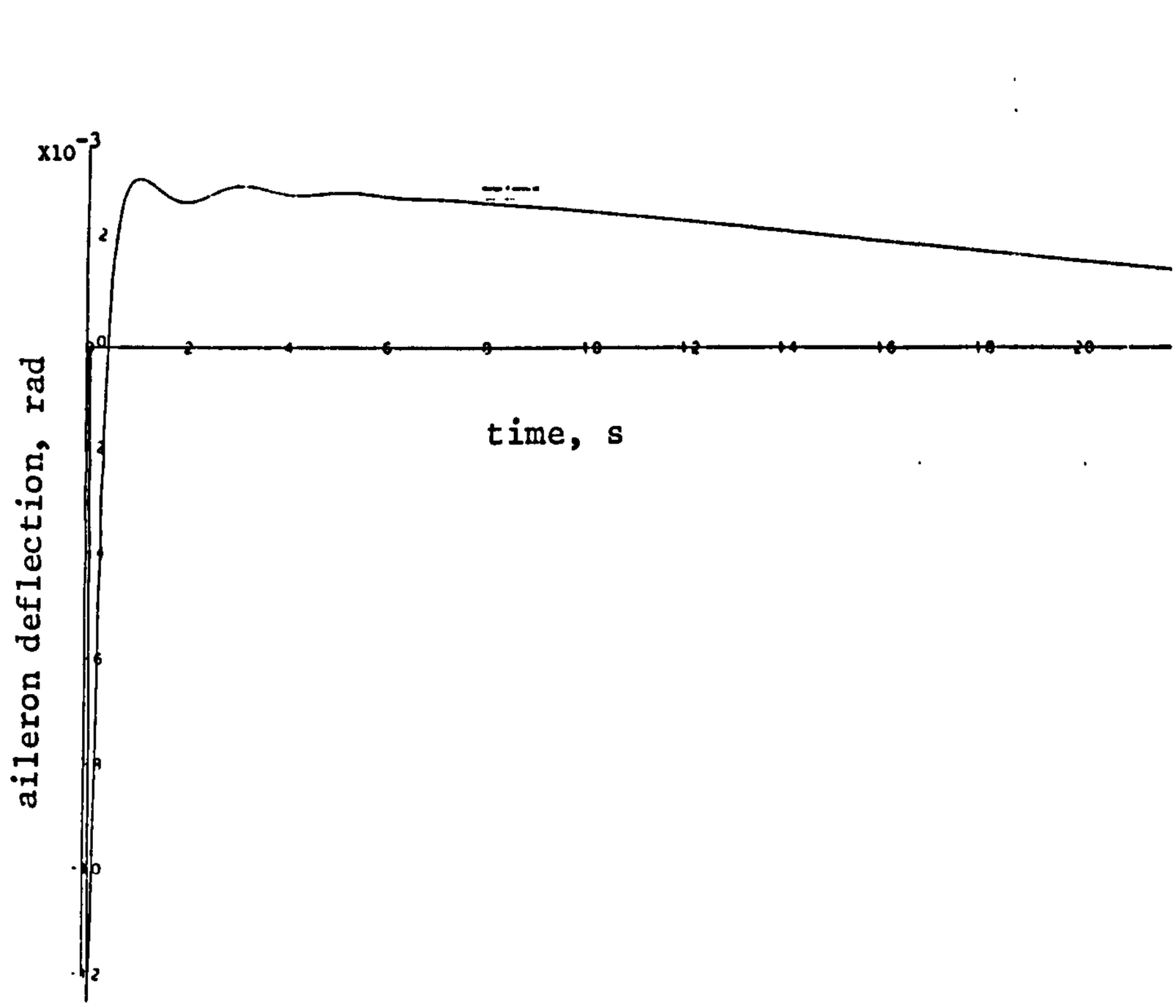


d)

Fig 6.2 cont.



e)



f)

these note that :

i) The roll rate, p, response is close to the desired first order transient with approximately the desired time constant of 0.25 s. Note that some evidence of Dutch roll is present in addition to a slight steady-state offset.

ii) The v and r responses show clear evidence of Dutch roll although the associated damping and period is somewhat different from those achievable with the linear model. The damping being somewhat lower and the period slightly shorter (approx. 2 s). The spiral mode is also evident on the r (yaw rate) response and this appears to be of larger time constant than that desired. Nevertheless this mode has clearly been stabilised.

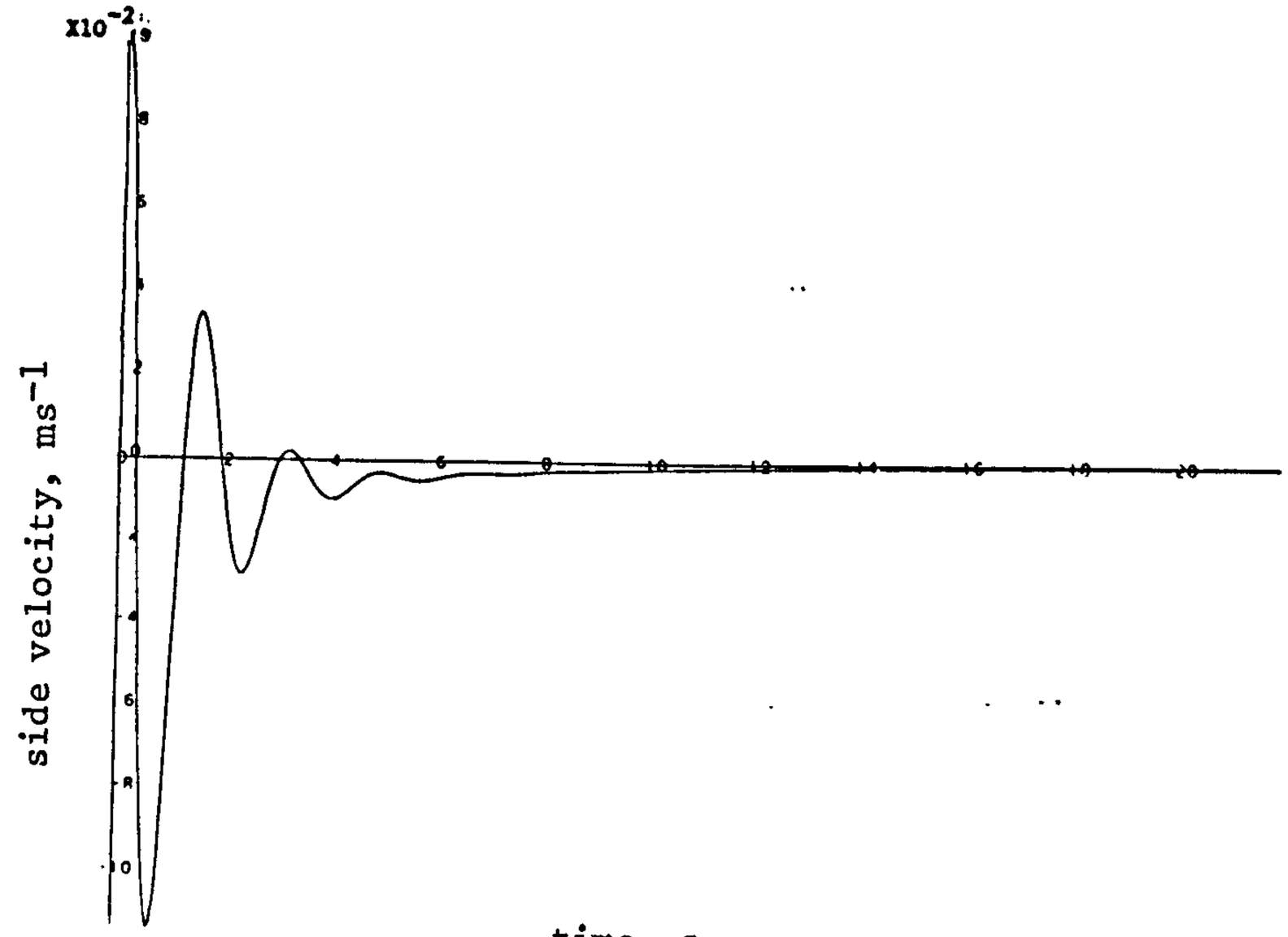
iii) The φ response shows a very slowly decaying mode along with slight Dutch roll coupling. This is considerably different from the linear model and may be due to a badly modelled spiral mode.

iv) The actuator responses are broadly similar to those for the linear model. A reasonable degree of cross-coupling suppression has been achieved. The rudder response shows evidence of the Dutch roll mode which is less well damped than desired.

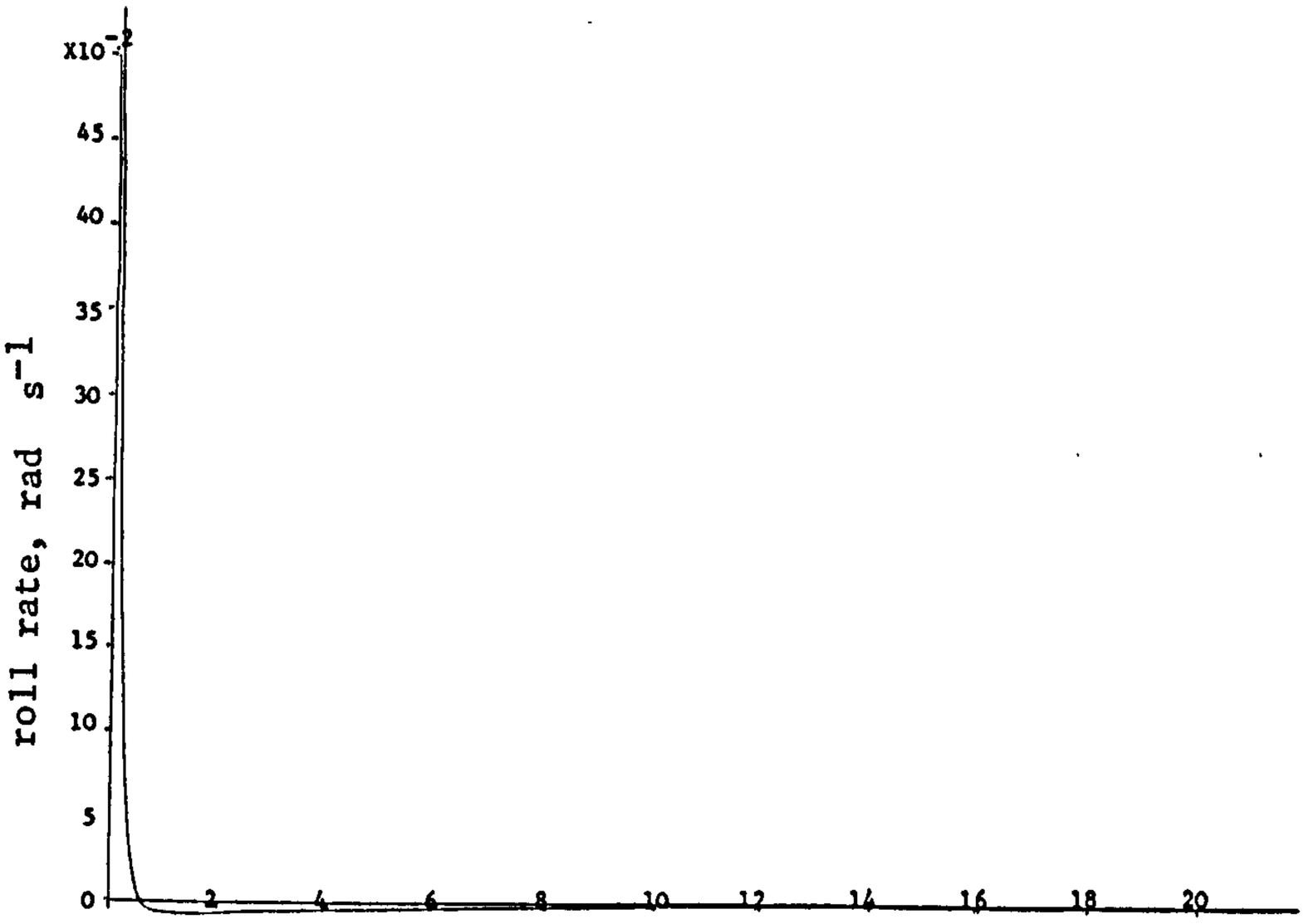
Whilst these responses do not exactly match the linearised model we still achieve a reasonable, stable response with the desired modal decoupling. Figs. 6.3 a) to f) show the state responses for the non-linear model with the same initial conditions as Figs. 6.2 but at 50 ms^{-1} airspeed. The dominant effect is to increase the Dutch roll period and slightly reduce its damping. Note, however, that the spiral mode time constant is reduced and is now quite close to that desired.

It would thus appear that the use of a single set of state feedback gains to cover all flight conditions leads to significant changes in aircraft response although, in this

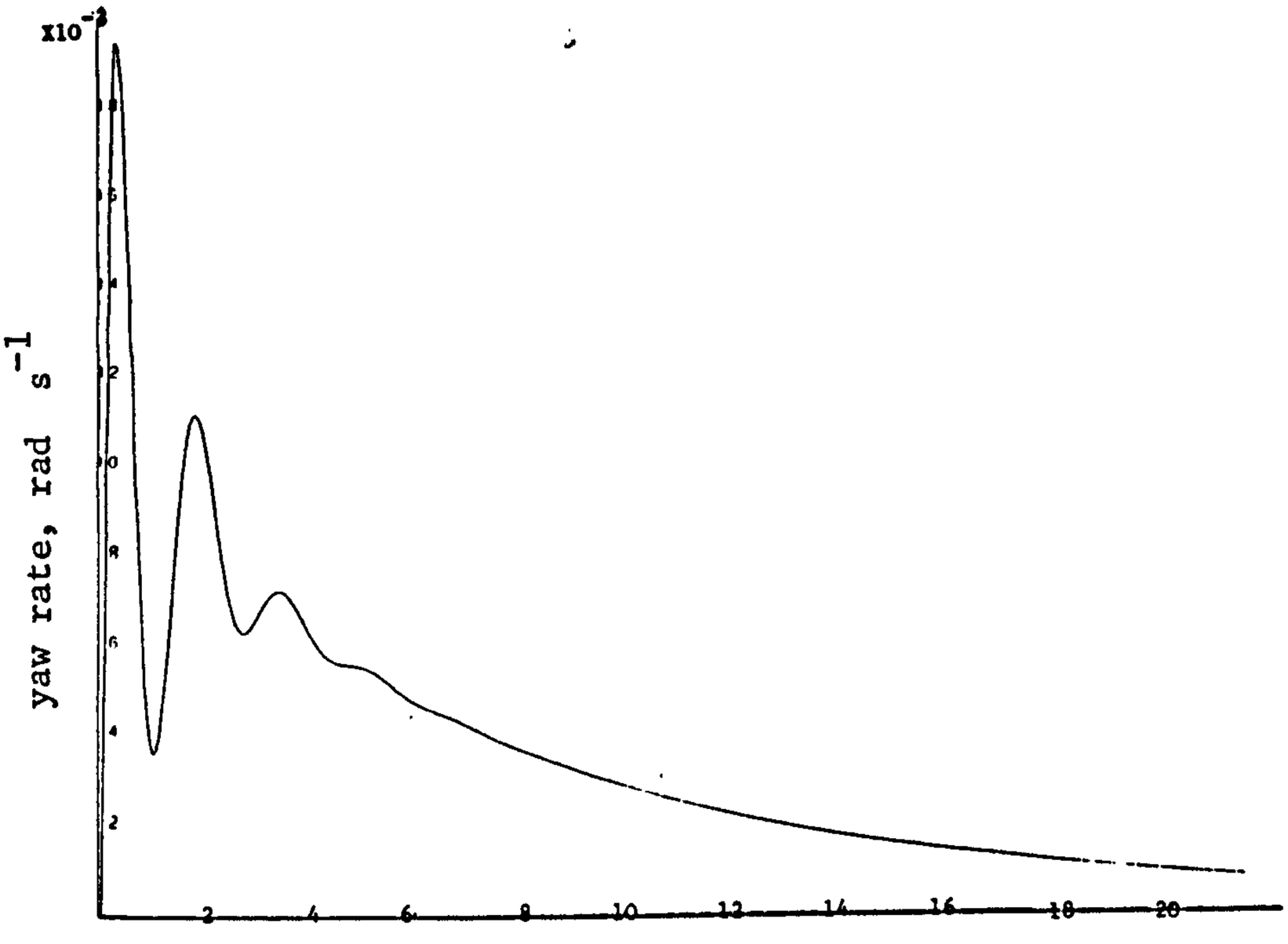
Fig 6.3 State Responses for Non-linear Lateral Model
(linear LQP controller, 50 ms^{-1} airspeed)



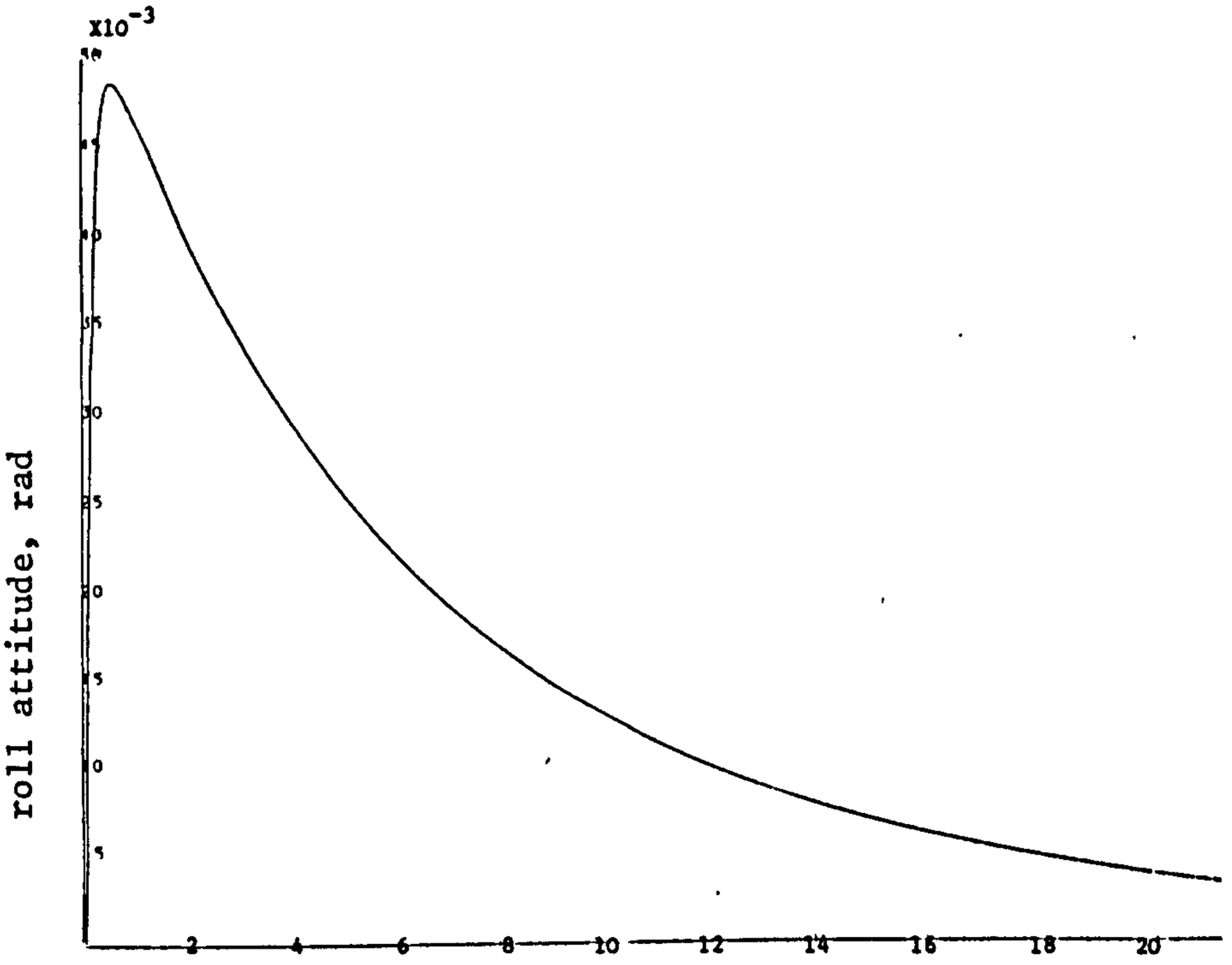
a)



b)

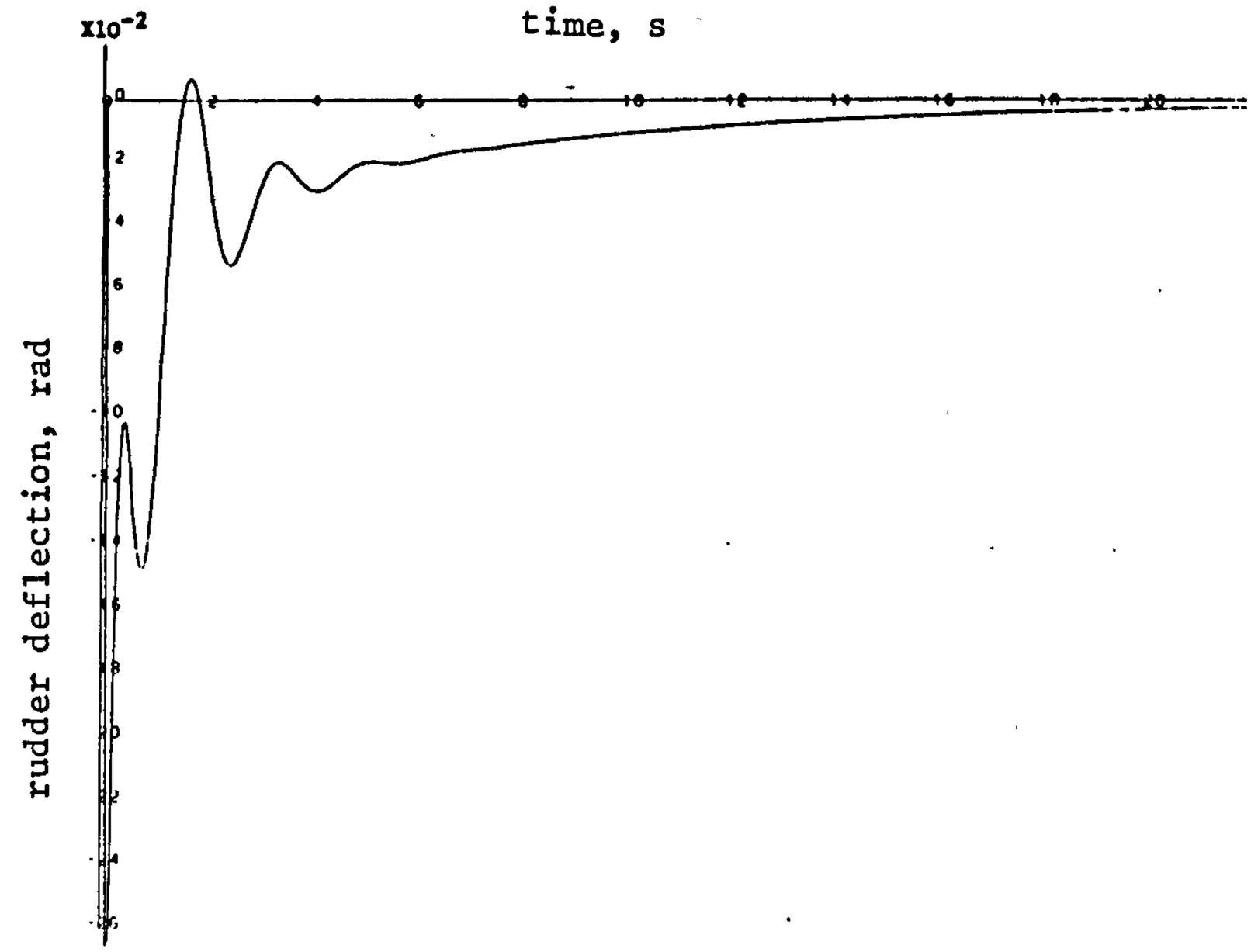

x10⁻³
yaw rate, rad s⁻¹
2
4
6
8
10
12
14
16
18
20
time, s

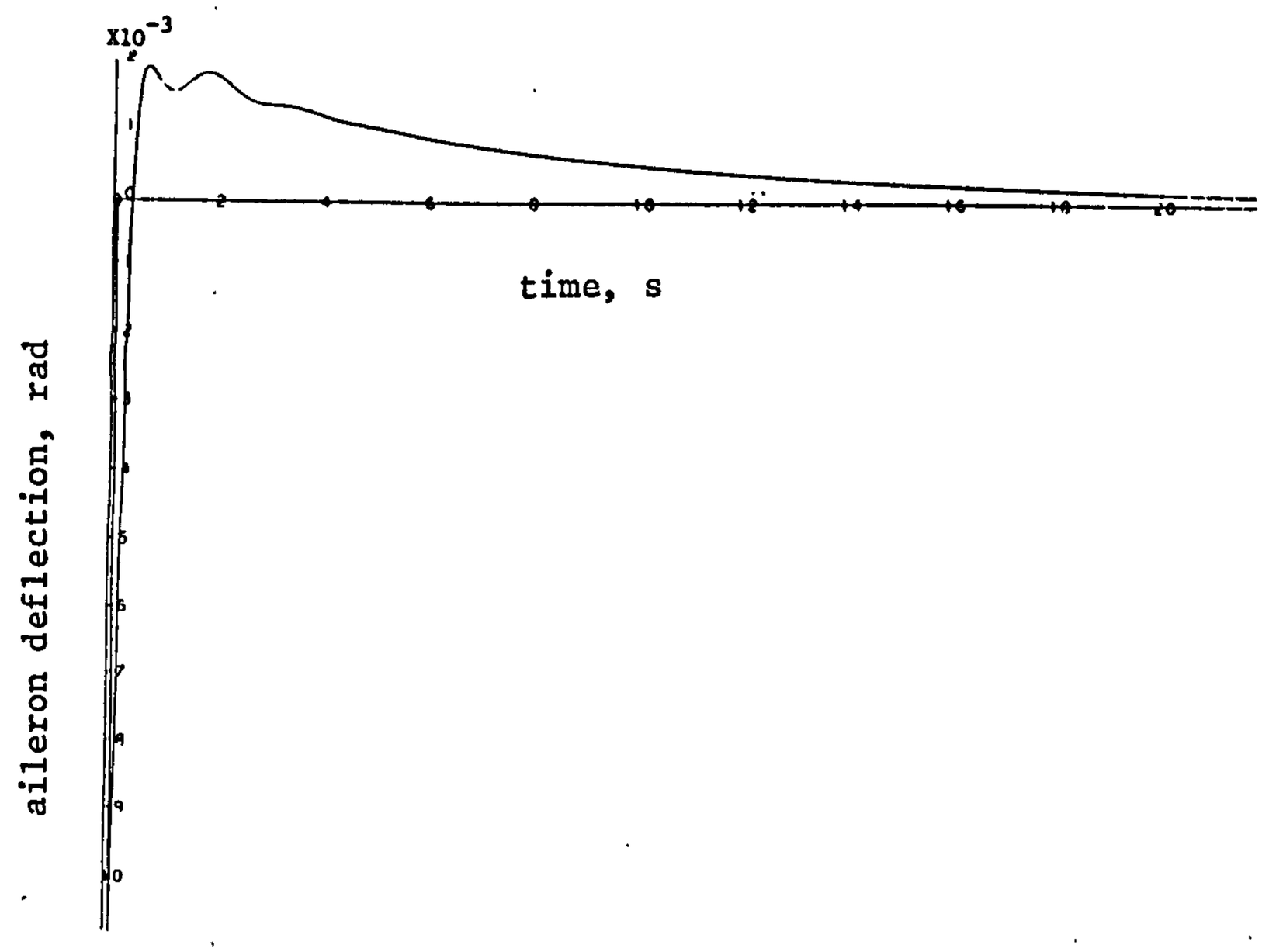
c)


x10⁻³
roll attitude, rad
2
4
6
8
10
12
14
16
18
20
time, s

d)

Fig 6.3 cont.


x10⁻²
time, s
rudder deflection, rad
e)


x10⁻³
time, s
aileron deflection, rad
f)

case, stability would appear to be maintained. Since the objective is to provide parameter-insensitive control the use of this design technique would require a gain-scheduling scheme to change the feedback gains for specific flight regimes. Nevertheless the ability to specify the desired eigenvalue/eigenvector structure may be valuable in addition to the unique specification of the Q and R matrices.

The output regulator problem mentioned above will now be investigated briefly and again the choice of Q and R to achieve desired responses indicated.

6.5.2 Output Assignment

In the previous section a useful design technique was demonstrated which allows state assignment to be achieved via the LQP problem. It was indicated that using state assignment provides no guarantee of desirable output response this being due to the ommission of the output C matrix from the design. A more desirable situation would be to provide desired output response without recourse to full state assignment (36,137,138). In the following discussion a technique due to Grimble (36) will be described which extends the work of Harvey & Stein into the output regulator problem.

Again begin by considering the linear time invariant system of equations 6.2 a) and b) subject to the quadratic cost performance index

$$J = \int_0^{t_f} (\underline{y}^t(t)\ Q'\ \underline{y}(t) + \rho\ \underline{u}^t(t)\ R'\ \underline{u}(t))\ dt \quad - 6.32$$

(c.f. with equation 6.5) where $\underline{y}(t)$ is an m vector of responses (outputs) and Q' is an appropriately dimensioned (m x m) weighting matrix. It is also required that the system be stabilisable (48) and square and also that r(CB) = m.

Consider, initially, the asymptotic behaviour of the system. For r(CB) = m there will be m asymptotically infinite eigenvalues which will approach infinity in m first

order Butterworth patterns (36,54,118) and associated eigenvectors (s_j^∞ , $\underline{v}_j^\infty$). The relationship between the Q' and R' matrices of 6.32 and (s_j^∞, $\underline{v}_j^\infty$) can be derived as follows.

The frequency domain version of the return difference relationship for the optimal LQ regulator is given by (49) :

$$\rho\, T^T(-s)\; R'\; T(s) = \rho\, R' + F^T(-s)\; Q'\; F(s) \qquad - 6.33$$

where $T(s) = (I - G(s))$

$$G(s) = - K\ \Phi(s)\ B$$

$$\Phi(s) = (s\ I - A)^{-1}$$

$$K = + P\ B\ R^{-1}/\rho$$

$$F(s) = C\ \Phi(s)\ B$$

The return difference operator, F(s), may be expressed as a power series in s^{-1} (50) thus :

$$F(s) = C\left[\frac{1}{s}\ (\ I + \frac{A}{s} + \frac{A^2}{s^2} + ...)\right] B \qquad - 6.34$$

Equation 6.33 may thus be expressed as :

$$\rho\, T^T(-s)\; R'\; T(s) = \rho\left[R' - \frac{1}{s'^2}(CB)^T\ Q'\ (CB) + O\ (\frac{1}{s})\right] \qquad - 6.35$$

where O(1/s) indicates terms in 1/s, $1/s^2$,.... and $s'=s\sqrt{\rho}$. In the asymptotic case, i.e. as $\rho \to 0$ then for a given finite frequency s', $|s| \to \infty$ and

$$T^T(-s)\; R'\; T(s) \to \left(R' - \frac{1}{s'^2}(CB)^T\ Q'\ (CB)\right) \qquad - 6.36$$

Now we have from equation 6.12 that

$$T(s_i)\ \underline{v}_i = 0 \qquad i=1,2,....,n$$

Pre multiplying by $T^T(-s_i)$ R' gives

$$T^T(-s_i)\ R'\ T(s_i)\ \underline{v}_i = 0 \qquad - 6.37$$

Hence from 6.36 for the asymptotically infinite eigenvalues s_i^∞ and associated eigenvectors $\underline{v}_i^\infty$ we have

$$\left\{ R' - \frac{1}{(s_i^\infty)^2} (CB)^T\ Q'\ (CB) \right\} \underline{v}_i^\infty = 0$$

Factoring the Q' matrix such that $Q'=W^TW$ with W a non-singular (m x m) matrix then

$$\left\{ R' - \frac{1}{(s_i^\infty)^2} (WCB)^T\ (WCB) \right\} \underline{v}_i^\infty = 0$$

$$\left\{ (WCB)^T \left[(WCB)^{-T}\ R'\ (WCB)^{-1} - \frac{1}{(s_i^\infty)^2} \right] (WCB) \right\} \underline{v}_i^\infty = 0$$

$$\left[(WCB)^{-T}\ R'\ (WCB)^{-1} \right] (WCB)\ \underline{v}_i^\infty = \frac{1}{(s_i^\infty)^2} (WCB)\ \underline{v}_i^\infty \qquad - 6.38$$

Note that equation 6.38 is in the form of an eigenvalue/eigenvector relationship with the positive definite, symmetric matrix (bracketed) having positive, real eigenvalues, $(1/s_i^\infty)^2$ and associated orthogonal eigenvectors (WCB $\underline{v}_i^\infty$). This being the case the following relationship holds :

$$(WCBN)^T\ (WCBN) = I_m \qquad - 6.39$$

where the matrix N is formed with its columns being the $\underline{v}_i^\infty$'s, i.e.

$$N = [\underline{v}_1^\infty\ ,\ \underline{v}_2^\infty\ ,\ \underline{v}_3^\infty\ ,\ \ldots\ ,\ \underline{v}_m^\infty\ ]$$

and the eigenvectors ($WCB\underline{v}_i^\infty$) are arranged to have unit magnitude. The matrix Q' then follows directly from 6.39 since

$$(CBN)^T\ W^TW\ (CBN) = I_m$$

$$Q' = W^T W = (CBN)^{-T} (CBN)^{-1} \qquad - 6.40$$

also from equation 6.38 we obtain

$$[N^T R' N^{-1}] N^{-1} \underline{v}_i^{\infty} = \frac{1}{(s_i^{\infty})^2} N^{-1} \underline{v}_i^{\infty} \qquad - 6.41$$

By choosing $W = (CBN)^{-1}$, thus

$$R' = N^{-T} S^{-2} N^{-1} \qquad - 6.42$$

where S is formed such that

$$S = \text{diag } \{s_1^{\infty} , s_2^{\infty} , \ldots\ldots, s_m^{\infty} \}$$

Also note that the asymptotically infinite eigenvalues are given by $s_i = s_i^{\infty} / \sqrt{\rho}$ and the associated eigenvectors may be established from equation 6.11, namely

$$\underline{e}_i = (\lambda_i I - A)^{-1} B \underline{v}_i$$

and $\underline{e}_i = (\frac{s_i^{\infty}}{\sqrt{\rho}} I - A)^{-1} B \underline{v}_i \rightarrow \frac{s_i^{\infty}}{\sqrt{\rho}} B \underline{v}_i$

as $\rho \rightarrow 0$ thence $\underline{e}_i^{\infty} = B \underline{v}_i^{\infty}$; provided that $\frac{s_i^{\infty}}{\sqrt{\rho}} \notin (A)$

The above establishes the results of Lemma 6.2.

The Q' and R' matrices have thus been characterised in terms of the asymptotically infinite modes of the closed-loop system. The design of the output regulator thus requires the values of $(s_i^{\infty} , \underline{v}_i^{\infty})$ to be specified and in the limiting case ($\rho \rightarrow 0$) these modes will dominate the output response. To choose $(s_i^{\infty} , \underline{v}_i^{\infty})$ Note that the asymptotic output directions, $\underline{y}_i^{\infty}$, are given by :

$$\underline{y}_i^{\infty} = C \underline{e}_i^{\infty} = CB \underline{v}_i^{\infty}$$

$$\text{or} \quad \underline{v}_i^{\infty} = (CB)^{-1} \underline{y}_i^{\infty} \qquad - 6.43$$

The s_i^{∞} may thus be chosen to satisfy given output (input) bandwidth requirements whilst the $\underline{v}_i^{\infty}$ may be chosen so as to reduce output interactions. Q' and R' then follow from

equations 6.40 and 6.42.

It now remains to determine the significance of the (n-m) asymptotically finite eigenvalues and eigenvectors (s_i^o, $\underline{e}_i^o$). It can be shown (45) that for the minimum-phase system, G(s), the asymptotically finite eigenvector directions lie within the null space of C, i.e. :

$$C\,\underline{e}_i^o = 0, \quad i \in \{1,2,\ldots,(n-m)\}.$$

Further, the (n-m) finite eigenvalues, λ_i^o, correspond to the zeros of S(A,B,C) and the vectors $\underline{e}_i^o$ correspond, except possibly for magnitude, to the zero directions ($\underline{w}_j$) of the system. The proof follows from Lemma 6.2 (51,53,54).

This result is important in that in the limiting case ($\rho \to 0$) (n-m) finite eigenvalues approach the invariant zeros of S(A,B,C) and the assosiated eigenvectors are constrained to lie within the null space of C whilst the remaining m infinite eigenvalues and assosiated eigenvectors lie within the range space of C. All uncontrollable modes will thus become unobservable (52).

Notice, however, that as $\rho \to 0$ then the closed-loop system gain increases. The demands that this may place on actuator responses, for example, may be a limiting constraint. At lower values of ρ, the degree to which the infinite modes are removed from the output response may be reduced. In many systems a compromise may be achievable but poorly behaved systems, i.e. systems with low gain sensitivities, may not be handled well by this technique. Also in the high gain case unmodelled system modes may be excited.

The discussion may be extended to cases where r(CB) = m. In these cases the calculation of the Q' and R' matrices is more complex. For a system in which the first k Markov parameters are zero Grimble (36) shows that by considering higher order terms in the Laurent series (equation 6.34) the following results are obtained :

$$Q' = (CA^kBN)^{-T}\,(CA^kBN)^{-1} \qquad - 6.44$$

$$R' = (N)^{-T}\,S_{k+1}\,N^{-1} \qquad - 6.45$$

where

$$S_{k+1} = (-1)^k \text{ diag}\{(1/s_1^{\infty})^{2(k+1)}, \ldots\ldots, (1/s_m^{\infty})^{2(k+1)}\} \qquad - 6.46$$

with the (k+1) Markov parameter, CA^kB, non-zero.

To illustrate this technique consider the lateral motion SAS design as was done in section 6.5.1.

6.5.2.1 Lateral SAS Design

Consider the linearised lateral dynamics of equation 6.9 and again choose actuator dynamics of the form given in section 6.5.1. This gives the A and B matrices as in equations 6.23. For convenience, we shall choose a somewhat impractical C matrix but one which gives r(CB) = 2. In this case choose the system outputs to be the actuator states, x_5 and x_6 and C is then given by

$$C'_{la} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For the asymptotically infinite eigenvalues and eigenvectors choose :

$$\underline{v}_1^{\infty} = [1/20 \quad 0]$$

$$\underline{v}_2^{\infty} = [0 \quad 1/10]$$

with s_1^{∞} twice as fast as s_2^{∞}, say $s_1^{\infty} = 1$, $s_2^{\infty} = 0.5$. This gives :

$$N = \begin{bmatrix} 1/20 & 0 \\ 0 & 1/10 \end{bmatrix} \quad ; \quad S = \text{diag}\{s_1^{\infty}, s_2^{\infty}\} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \qquad - 6.47$$

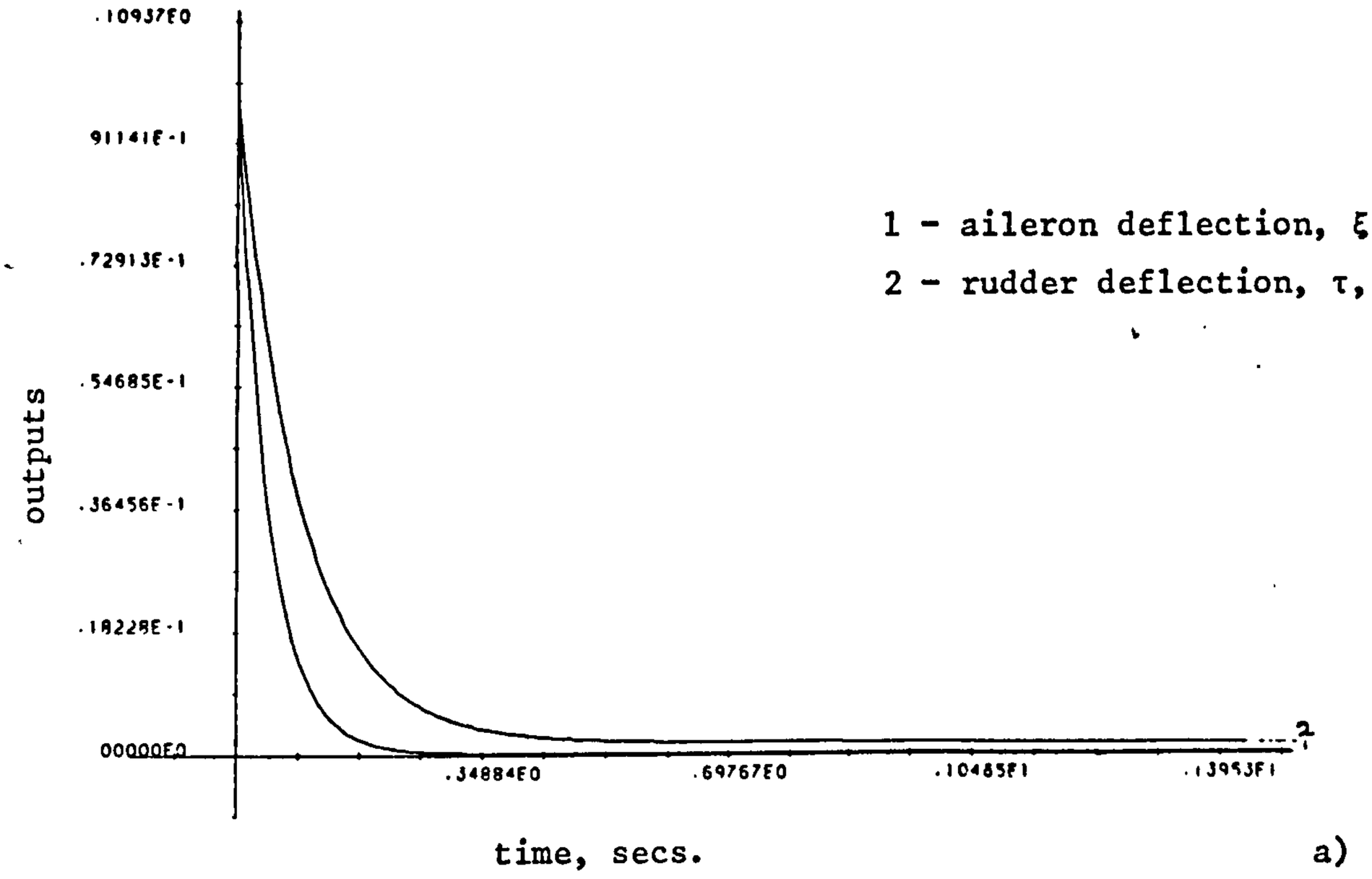
Note also that r(CB) = 2 and $CB = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix}$

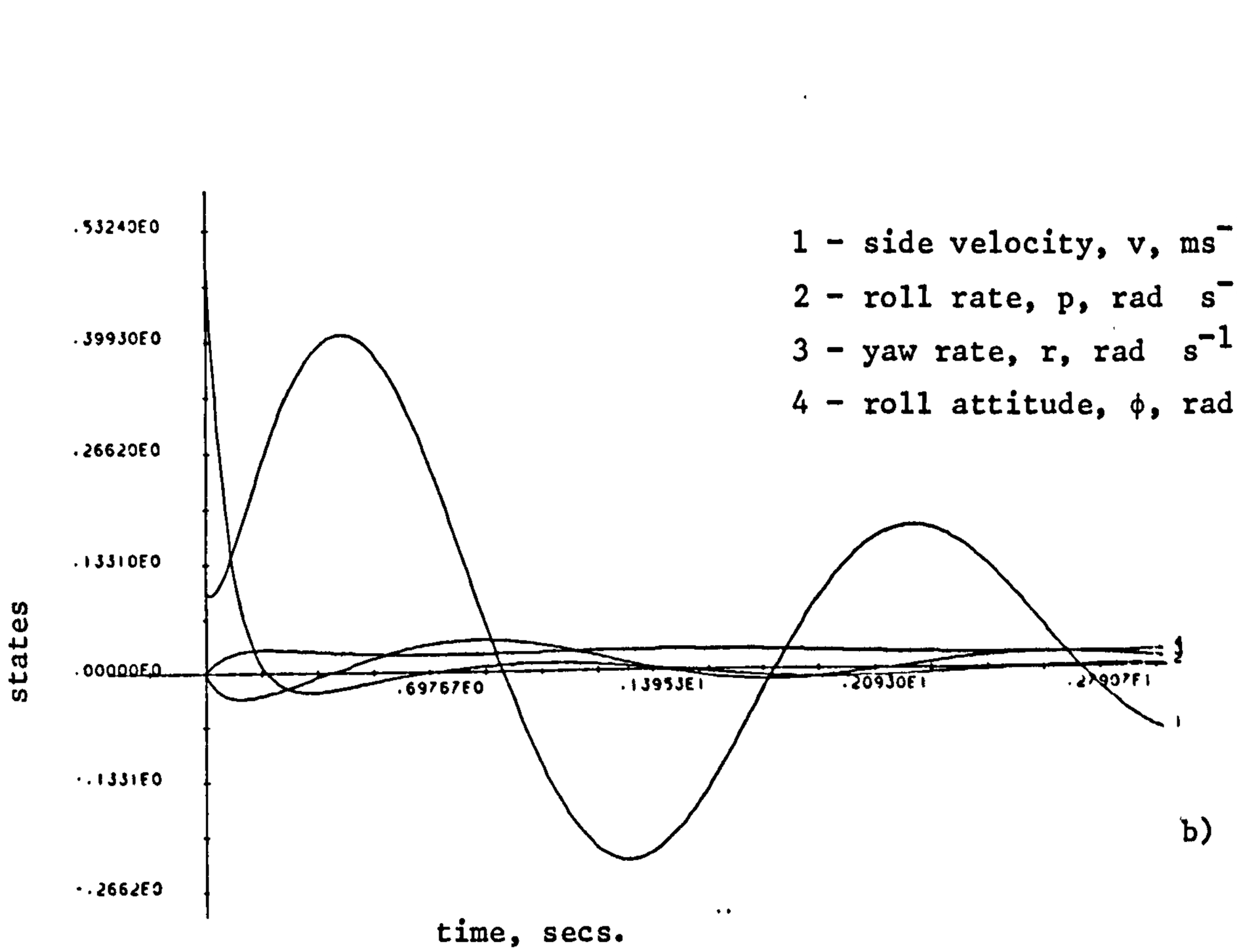
Thus $$Q' = (CBN)^{-T} (CBN)^{-1}$$
$$= I_2$$

and $$R' = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix} = 400\ I_2$$

The ρ parameter allows some flexibility in the design and

Fig 6.4 State and Output Responses for linear model at 33 ms^{-1} Airspeed (ρ = 0.0025, output assignment)





should be chosen so that, in this case, actuator bandwidths are not exceeded. Recall that the rudder and aileron actuators have 10 and 20 rad s^{-1} bandwidths, respectively. A choice of $\rho = 0.0025$ gives actuator poles at -11.18 and -20.36 s^{-1}, respectively, and disposes the remaining four closed-loop poles at

$$-8.359,\ -0.122,\ -0.5 \pm 3.5j$$

These poles correspond to the (n-m) transmission zeros of the system.

Figs. 6.4 a) and b) show the output and state responses for the optimally controlled linear system with $\rho = 0.0025$ and with initial conditions of 0.1 ms^{-1} in side velocity, 0.5 rad s^{-1} in roll rate and 0.1 rad in rudder and aileron deflections. Note that the output responses approach their desired 0.1 and 0.05 s first order time constants with no evidence of the four asymptotically finite modes, as desired.

Consider now a more realistic choice of the C matrix, as :

$$C'_{1a} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

which picks off the two aircraft lateral axial rates, i.e. p and r. These are normally available via. on-board rate gyros. Maintaining the A and B matrices as in equations 6.23, note that the first Markov parameter, CB, is not full rank but r(CAB) = 2 hence we must employ the results of equations 6.44, 6.45 and 6.46 with k=1. For convenience again employ the choice of asymptotically infinite eigenvectors so as to provide asymptotic decoupling between roll and yaw, namely

$$N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2$$

and for the asymptotically infinite eigenvalues we again choose $s_1^{\infty} = 1$, $s_2^{\infty} = 0.5$, i.e. choose output 1, roll rate, to be twice as fast as output 2, yaw rate. Now, from equation 6.46

$$S_2 = (-1)^1 \{\text{diag}\{(1/s_1^{\infty})^4, (1/s_2^{\infty})^4\}\}$$

i.e. $$S_2 = -\text{diag}\{1, 16.0\}$$

and note that $CAB = \begin{bmatrix} 0 & -286.4 \\ -189.8 & 0 \end{bmatrix}$

thus $$Q' = \begin{bmatrix} 0 & -286.4 \\ -189.8 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -286.4 \\ -189.8 & 0 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} 1.2 \times 10^{-5} & 0 \\ 0 & 2.775 \times 10^{-5} \end{bmatrix}$$

and from equation 6.45

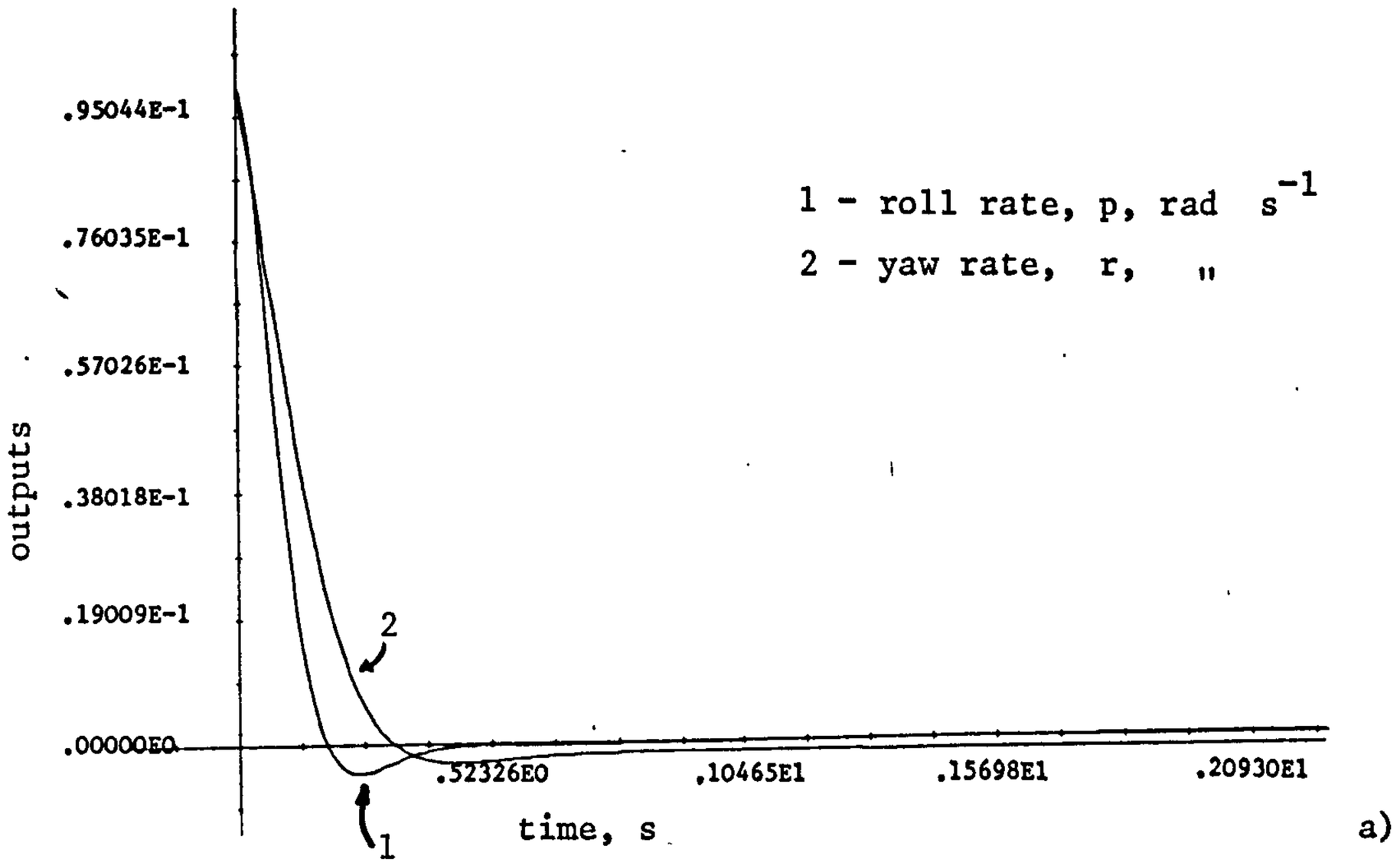
$$R' = \begin{bmatrix} -1 & 0 \\ 0 & -16 \end{bmatrix}$$

It now remains to choose the value of ρ and this is probably best done by interactive simulation or reference to a multivariable root locus diagram. Note, however, that in this case it is not as simple to decide upon the likely asymptotic behaviour since the system may have more or less than m transmission zeros. Equally, the degree to which interactions may be suppressed depends much more on the choice of ρ . There is also no guarantee of asymptotic stability in this case. This was confirmed by applying designs corresponding to successively decreasing values of ρ to the linear model.

Figs. 6.5 a) and b) show the state and output responses for $\rho = 10^{-3}$. Note that the output response is fast with approximately the 2:1 bandwidth ratio desired. The state responses are also reasonably fast, however, the actuator demands (states 5 and 6) may be considered a little excessive. Reducing ρ to 10^{-1} produces the output and state responses of Figs. 6.6 a) and b). Note the reduced degree of modal decoupling in addition to a slower response speed.

In summary then it would appear that the technique outlined above provides a useful method of specifying the Q' and R' matrices via the asymptotic behaviour of the outputs. It would, however, seem to be limited to a class of systems where r(CB) = m. In other cases some of the elegance is

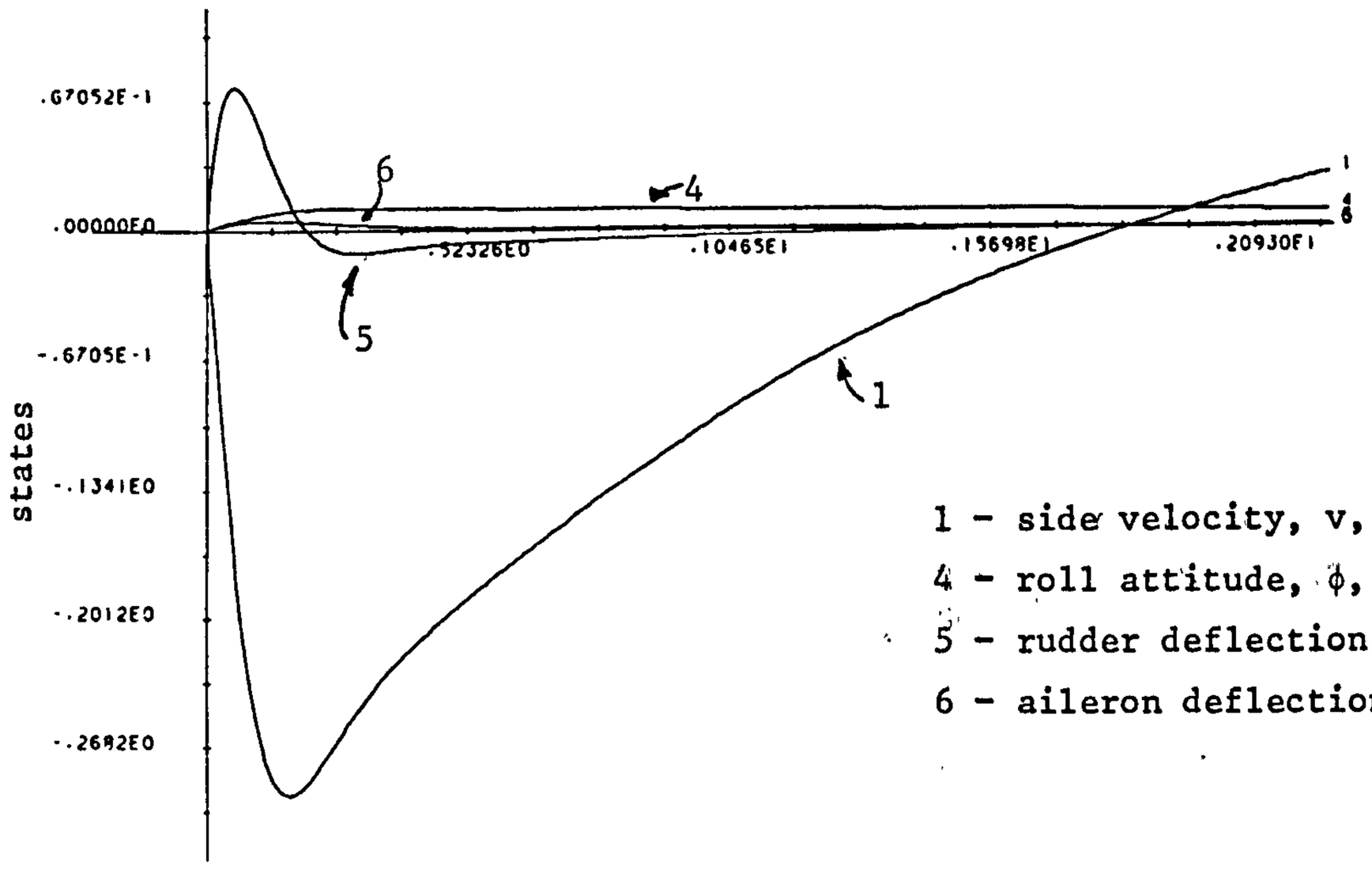
Fig 6.5 State and Output Responses for linear model at 33 ms^{-1} Airspeed (ρ = 0.001, i.c. of 0.1 in roll rate)



1 - roll rate, p, rad s^{-1}

2 - yaw rate, r, "

a)



1 - side velocity, v, ms^{-1}

4 - roll attitude, ϕ, rad

5 - rudder deflection, τ, rad

6 - aileron deflection, ξ, rad

b)

lost due to complex asymptotic behaviour. Nevertheless there is much to be said in favour of such a relatively simple design technique. In the aircraft problem it is, however, of limited value since it is desirable to maintain given modal responses since these are often felt directly by the pilot and passengers. Using output regulation will not guarantee 'ride quality' as will the techniques discussed in section 6.5.1.

6.6 Demand Following

It was indicated in the initial discussion that in the aircraft problem we frequently require that the control be designed to allow a demand signal to be followed arbitrarily closely. In the optimal control context this requires that the controller should track a reference signal. This may be the case in, for example, turning manoeuvres or where a given height must be maintained. Since the normal aim, with optimal control, is to drive all the states of the system to zero the design equations for the controller must be modified in such a way as to provide this 'following' action. There are many methods available which provide this action but here we shall consider only one (55).

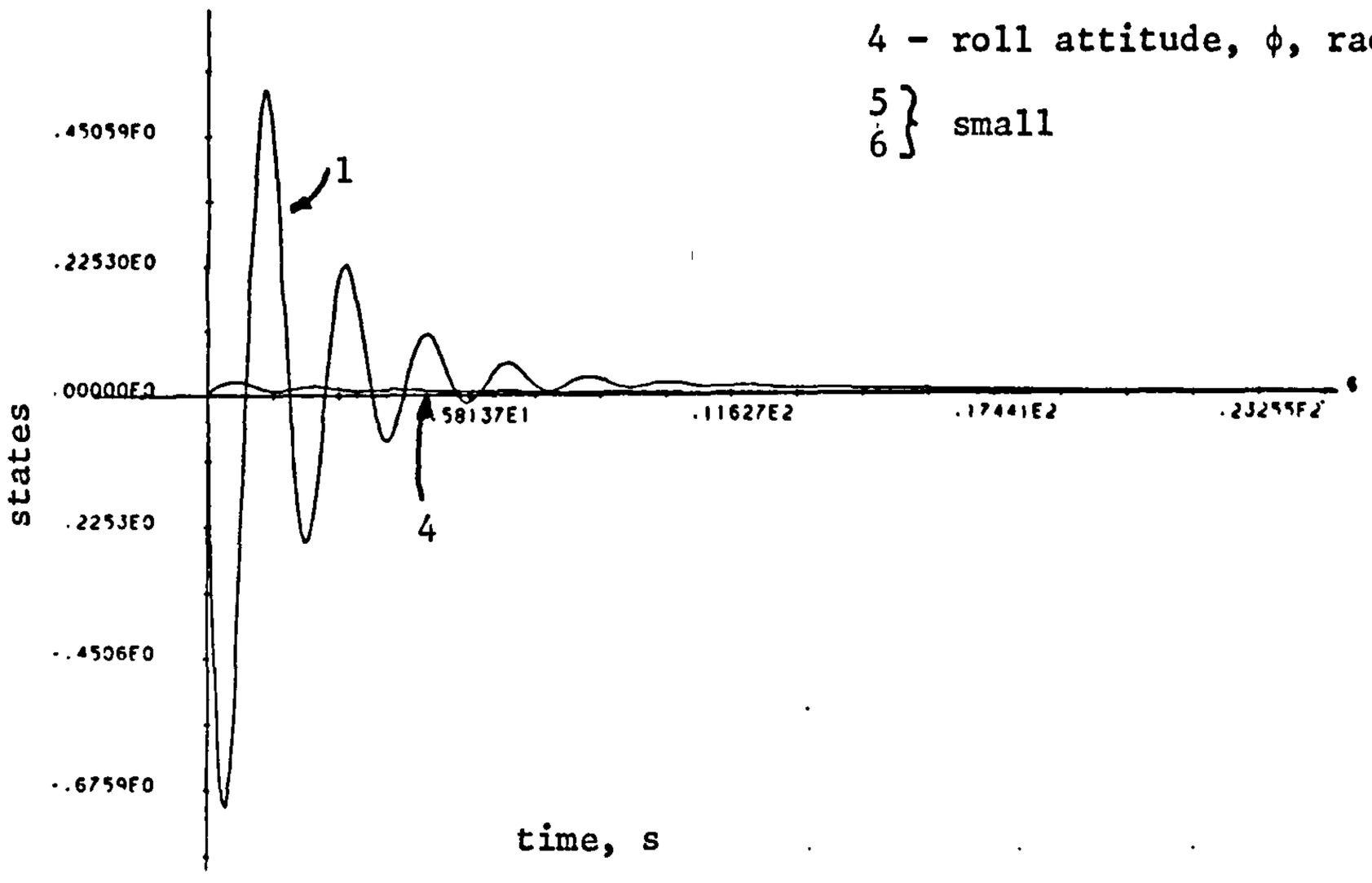
Recall that the cost function for the quadratic case is given as in equation 6.5 as

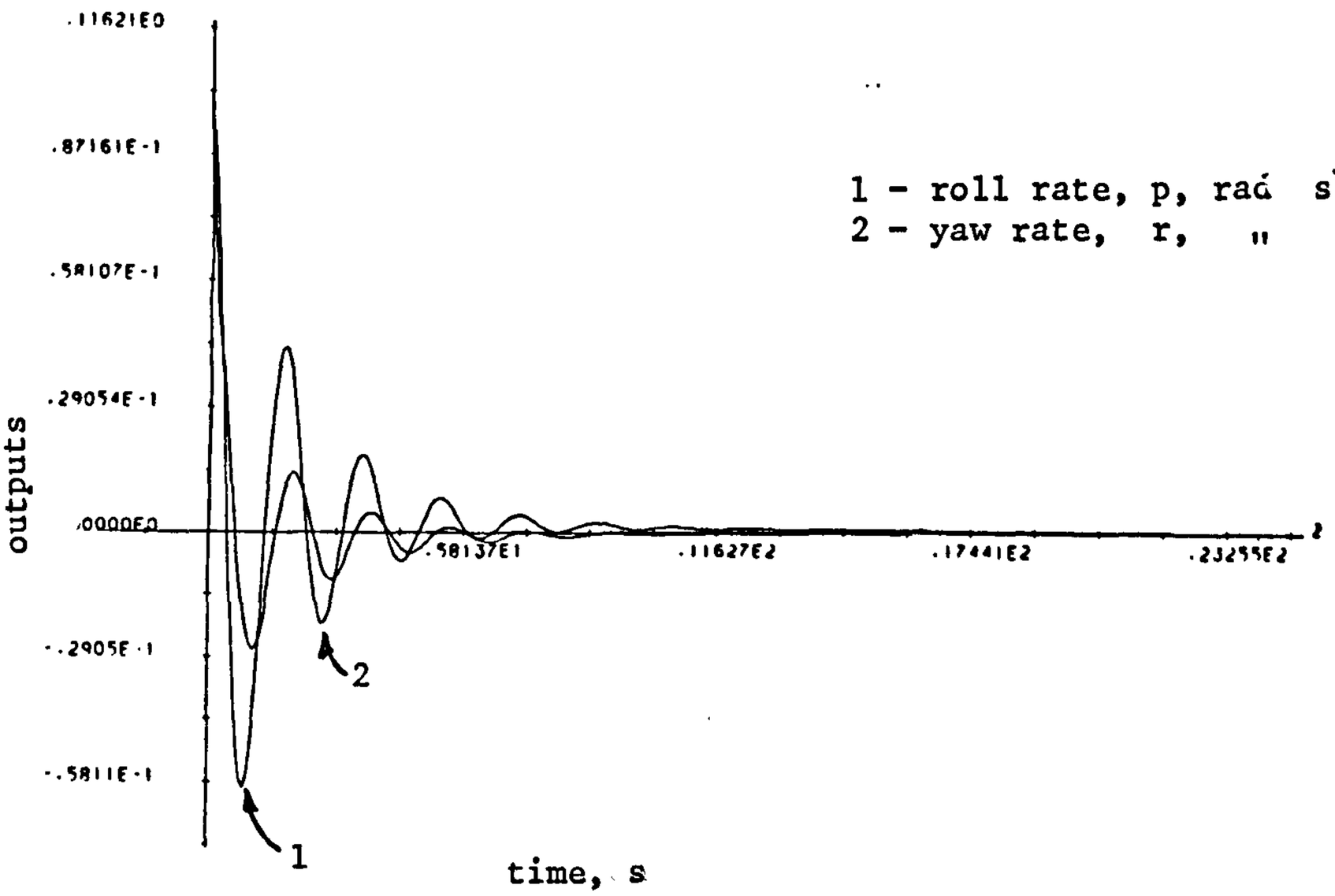
$$J = \int_0^{t_f} (\underline{x}^t \, Q \, \underline{x} + \rho \, \underline{u}^t \, R \, \underline{u}) \, dt$$

If the states, x's, in this equation are replaced by deviations, i.e. (x-r)'s, then the following is obtained

$$J = \int_0^{t_f} ((\underline{x} - \underline{r})^t \, Q \, (\underline{x} - \underline{r}) + \rho \, \underline{u}^t \, R \, \underline{u}) \, dt \qquad - 6.48$$

Fig 6.6 As Fig 6.5 with $\rho = 0.1$





where $\underline{r}$ is an n vector of reference signals. Note that now the deviations of the states from the references are driven to zero in minimum time subject to certain control constraints. This is desirable but the solution of the problem via. the Ricatti equation must be modified as follows.

Consider an augmented state vector formed from the states, $\underline{x}(t)$, and the references, $\underline{r}$, i.e.

$$\underline{z}(t) = \begin{bmatrix} \underline{x}(t) \\ \underline{r} \end{bmatrix}$$

Assuming that $\underline{r}$ is a constant, time-invariant vector then the augmented state equations may be obtained as

$$\dot{\underline{z}}(t) = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \underline{z}(t) + \begin{bmatrix} B \\ 0 \end{bmatrix} \underline{u}(t) \qquad - 6.49$$

It then follows that in the infinite time case 6.48 may be expressed as

$$J = \int_0^{t_f} \underline{z}^t(t) \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix} \underline{z}(t) + \rho\, \underline{u}^t(t)\, R\, \underline{u}(t)\, dt \qquad - 6.50$$

Recall that from equation 6.8 the solution of the infinite time LQP problem is given by

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

Substituting for our augmented state and weighting matrices of equations 6.49 and 6.50 and appropriately partitioning the P matrix yields :

$$\mathbf{0} = \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix} + \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}$$

$$- \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} B \\ 0 \end{bmatrix} R^{-1} \begin{bmatrix} 0 & B^T \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}$$

Note that P_{11} and P_{22} are (n x n) and (m x m) whilst P_{12} is (n x m). Solving this set of equations gives

$$0 = Q + P_{11} A + A^T P_{11} - P_{11} B R^{-1} B^T P_{11} \quad - 6.51$$

$$0 = -Q + A^T P_{12} - P_{11} B R^{-1} B^T P_{12} \quad - 6.52$$

$$0 = Q - P_{12}^T B R^{-1} B^T P_{12} \quad - 6.53$$

The optimal control law is given by

$$\underline{u}(t) = - R^{-1} [B^T \quad 0] \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \underline{z}(t)$$

$$\underline{u}(t) = - R^{-1} B^{-1} [P_{11} \mid P_{12}] \underline{z}(t)$$

Solving equation 6.51 for P_{11} (55) then gives P_{12} by direct substitution into 6.52, i.e.

$$P_{12} = (A^T - P_{11} B R^{-1} B^T)^{-1} Q$$

The closed-loop state equations for the system may then be expressed as

$$\dot{\underline{x}}(t) = (A + B K_1) \underline{x}(t) + B K_2 \underline{r}$$

where

$$K_1 = - R^{-1} B^T P_{11} \quad - 6.54$$

$$K_2 = - R^{-1} B^T (A^T - P_{11} B R^{-1} B^T)^{-1} Q \quad - 6.55$$

The resulting control thus comprises a closed-loop configuration as shown in Fig. 6.7 with the reference vector acting as the demand vector for the system.

To illustrate this technique consider the linearised longitudinal state equations for the Machan as given in equations 2.36 for 33 ms^{-1} 'stick fixed' flight.

Firstly, an optimal control design for the system is required and this is undertaken using the Harvey & Stein technique of section 6.5.1. For convenience the thruster dynamic is neglected and an actuator dynamic for the elevator of

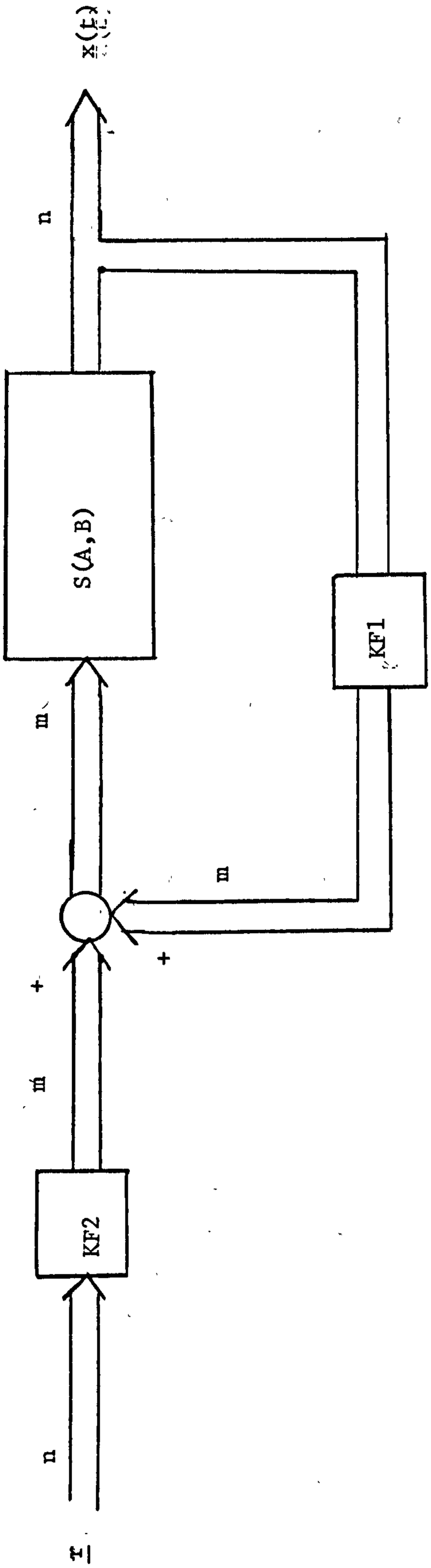


Fig 6.7 Archetypal State Feedback Demand Following Scheme

$$\frac{\eta}{\eta_d}(s) = \frac{20}{(s + 10)}$$

is introduced giving state (A,B) matrices of :

$$A_{1o} = \begin{bmatrix} -0.059 & 0.147 & 0. & -9.81 & 0. & 0. \\ -0.475 & -2.93 & 32.77 & 0. & 0. & 5.318 \\ 0.166 & -0.416 & -0.645 & 0. & 0. & -13.58 \\ 0. & 0. & 1. & 0. & 0. & 0. \\ 0. & -1. & 0. & 33. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & -10. \end{bmatrix}$$

$$B_{1o} = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 20. \end{bmatrix}$$

Note that the B_{1o} matrix has the required form and that the pair (A,B) is controllable. Since there is now only one actuator row in B_{1o} then a suitable choice of C matrix will give r(CB) = m = 1, i.e. full rank. This being the case there must be only one asymptotically infinite closed-loop eigenvalue and eigenvector with the five remaining finite eigenvalues corresponding to the transmission zeros of S(A,B,C).

Initially, these five finite closed-loop modes for the system must be specified. Recall from Chapter 2 that longitudinal motions consist of two complex-pole pairs, the phugoid and the short period. The usual aim is to improve the phugoid damping and reduce its period in addition to providing a well damped and fast short period response (57). A choice of closed-loop modal distribution was thus made such that :

Phugoid mode	$-1 \pm 1j$
Short period mode	$-2.8 \pm 5j$
Height	-0.1

This gives 45° damping of the phugoid mode and 60° damping in the short period. The height 'integration' was chosen to be first order with 10 secs. time constant. The choice of the finite asymptotic eigenvector direction set must also be made and here we may apply similar arguments to those used in section 6.5.1. The ordering of the states in this case is

x_1	-	u	forward velocity
x_2	-	w	vertical velocity
x_3	-	q	pitch rate (axial)
x_4	-	θ	pitch attitude
x_5	-	h	height
x_6	-	η	elevator deflection

For the phugoid mode recall that this is largely derived from exchanges in the kinetic and potential energies of the airframe hence it would be expected that the height, h, and forward velocity, u, responses be dominated by this mode with little coupling into pitch. A suitable desired eigenvector pair would thus be

$$\underline{e}_1^* = [v \quad v \quad 0 \quad 0 \quad 1 \mid v]$$

$$\underline{e}_2^* = [1 \quad v \quad 0 \quad 0 \quad v \mid v]$$

Since height, $h = -\int w\, dt$, it would be unrealistic to expect the mode not to appear in w.

For the short period mode note that this is a result of the 'arrow stability' of the airframe and hence is characterised by change in pitch rate, q, and vertical velocity, w. The mode should not be apparent in forward velocity but clearly will in height and pitch attitude. Hence, a suitable choice of a desired eigenvector pair would be :

$$\underline{e}_3^* = [0 \quad 1 \quad v \quad v \quad v \mid v]$$

$$\underline{e}_4^* = [0 \quad v \quad 1 \quad v \quad v \mid v]$$

The final real mode is the height integration which shows up dominantly on height with no pitch coupling and hence a suitable desired eigenvector would be :

$$\underline{e}_5^* = [v \quad v \quad 0 \quad 0 \quad 1 \mid v]$$

Note that in all cases we would expect some of each mode to show on the elevator since this is the only control.

The achievable eigenvectors, evaluated as in section 6.5.1, are as below :

$$\underline{e}_1^o = [-0.045 \quad 2.42 \quad 0.17 \quad -0.07 \quad 0.99 \quad -0.068]$$

$$\underline{e}_2^o = [\ 1.0 \quad 0.533 \quad -0.029 \quad 0.099 \quad -3.74 \quad 0.0076]$$

$$\underline{e}_3^o = [0.028 \quad 1.11 \quad 0.00035 \quad 0.025 \quad -0.047 \quad 0.027]$$

$$\underline{e}_4^o = [0.0011 \quad 0.00035 \quad 0.166 \quad -0.014 \quad 0.083 \quad 0.026]$$

$$\underline{e}_5^o = [-0.48 \quad 0.067 \quad 0.0001 \quad -0.001 \quad 1.0 \quad -0.0079]$$

Note that in the case of the phugoid and height integration modes the desired objectives can be met quite well. The short period mode poses some problems and the best achievable eigenvector is a relatively poor fit for $\underline{e}_4^*$ in particular. This reflects the relatively large degree of cross-coupling between forward velocity, u, and short term pitch variations for our particular example. A better fit may be had by a different choice of desired eigenvector. The design was completed by evaluating the C_o matrix of equation 6.25.

It remains to choose the eigenvalue and asymptotic direction of the infinite mode. In this case the task is trivial since only one control input is available. Taking R = 400 satisfies equation 6.16 b) and gives $W^TW = 1$.

A straightforward application of the optimal control design package then allows a choice to be made of the ρ parameter value so as to satisfy actuator bandwidth, etc. constraints. A value of 0.0025 gives a closed-loop actuator pole of -23.35 and this is within the physical constraints placed on the elevator actuator. The remaining closed-loop poles are then disposed at :

$$-2.86 \pm 6.3j, \quad -0.972 \pm 0.98j, \quad -0.1$$

which will be seen to be close to the desired eigenvalues

for the closed-loop system.

Recall that the objective of this exercise is to examine the performance of the demand following optimal control policy as developed in the preceding discussion. The application of the design technique gives the two state feedback gain matrices K_1, K_2 of equations 6.54 and 6.55. Note that K_1 corresponds to the standard optimal control law with no reference input. For the control design outlined above we obtain

$$K_1 = [0.432 \quad -0.03 \quad 0.459 \quad 3.77 \quad 0.2 \quad -0.875]$$
$$K_2 = [0.43 \quad -0.02 \quad 0.31 \quad 3.63 \quad 0.2 \quad -1.0 \ ]$$

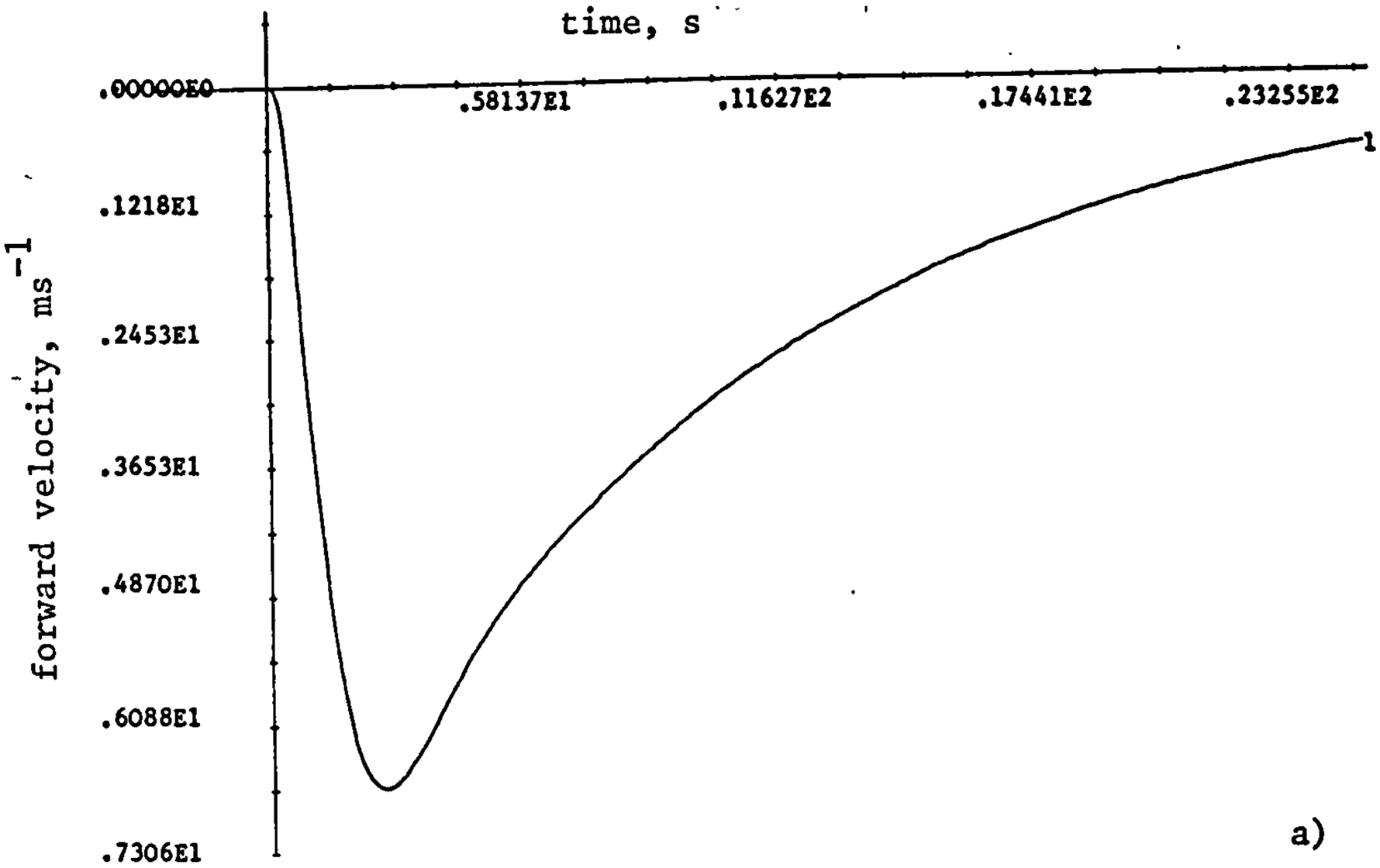
In Figs. 6.8 a) to f) the state responses of the system are shown corresponding to a change in the height state from an initial condition of 20 metres to a reference of 30 metres. This may be a typical requirement of such an SAS system, i.e. to execute a step change in height. From the figures the following can be noted :

i) The phugoid mode appears dominantly on the u and h responses. Note that this is a relatively slow mode with critical damping. Note also that the long term height integration mode appears on both u and h, as expected. The response of the height to the step change is reasonable although some offset is evident after 23 secs. This may indicate that some form of integral action is desirable although this aspect has not been considered further.

ii) The short period mode is largely distributed between the w and q states although it appears on both the pitch attitude and actuator responses. This mode is slightly underdamped and relatively fast, as desired.

iii) The overall response is acceptable but the demands placed on the actuators are quite large in terms of magnitude. This is largely to control the short period mode and hence a better choice of damping and period may be desirable. The pitch attitude change is also quite large and again a better choice of short period may be suggested.

Fig 6.8 State Responses for Linear Longitudinal Model
(33 ms^{-1} airspeed, demand following)



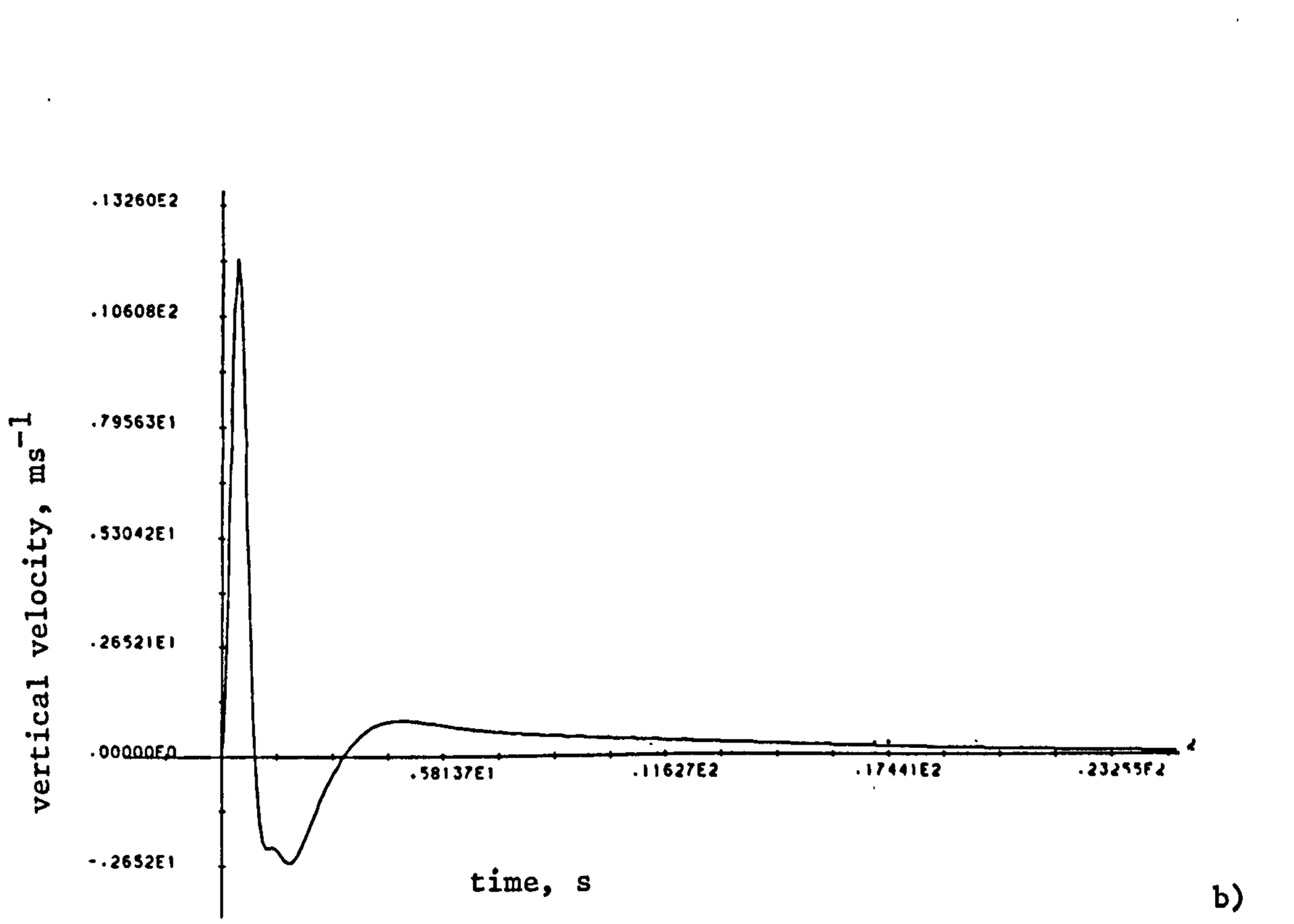
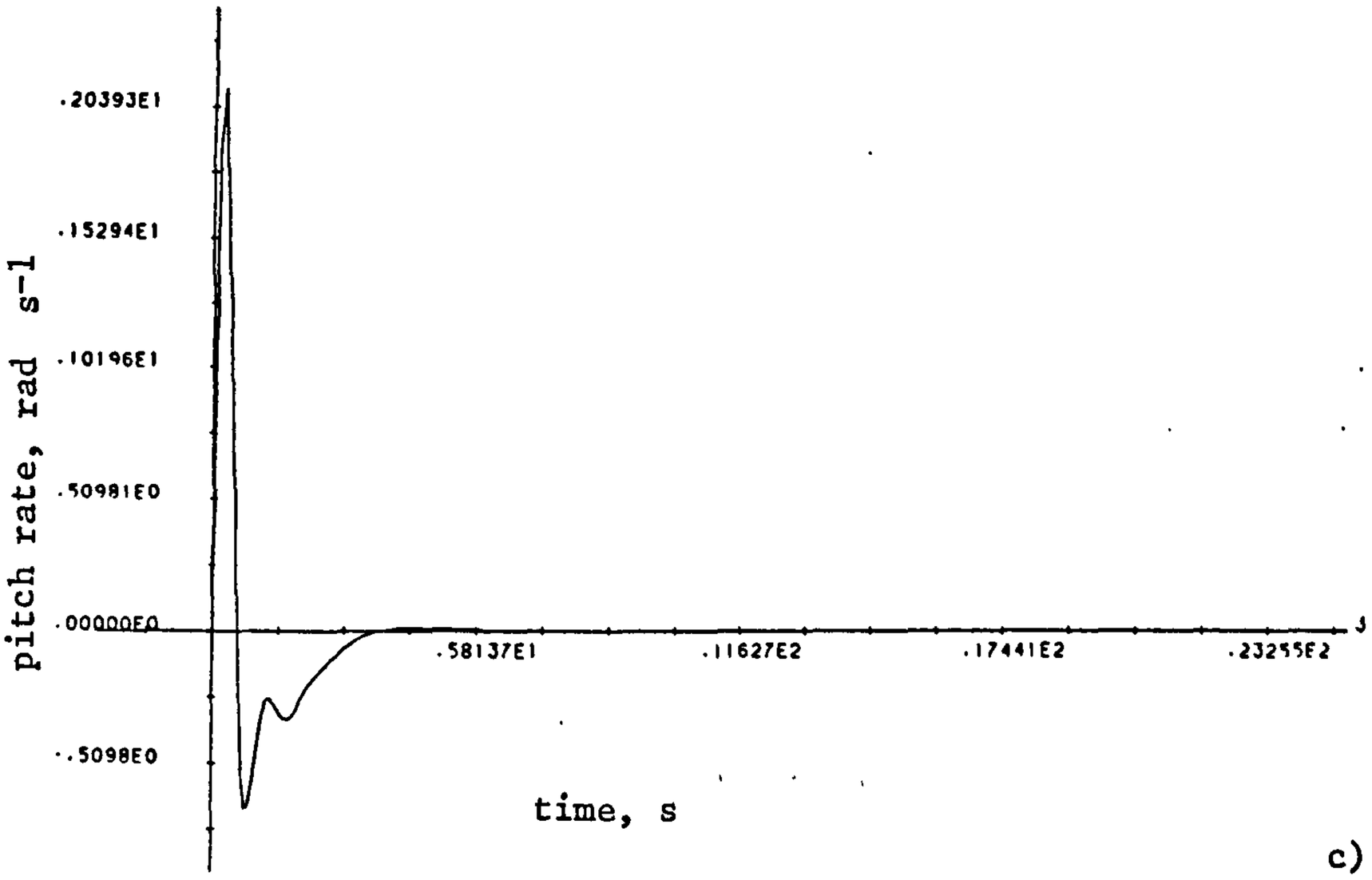
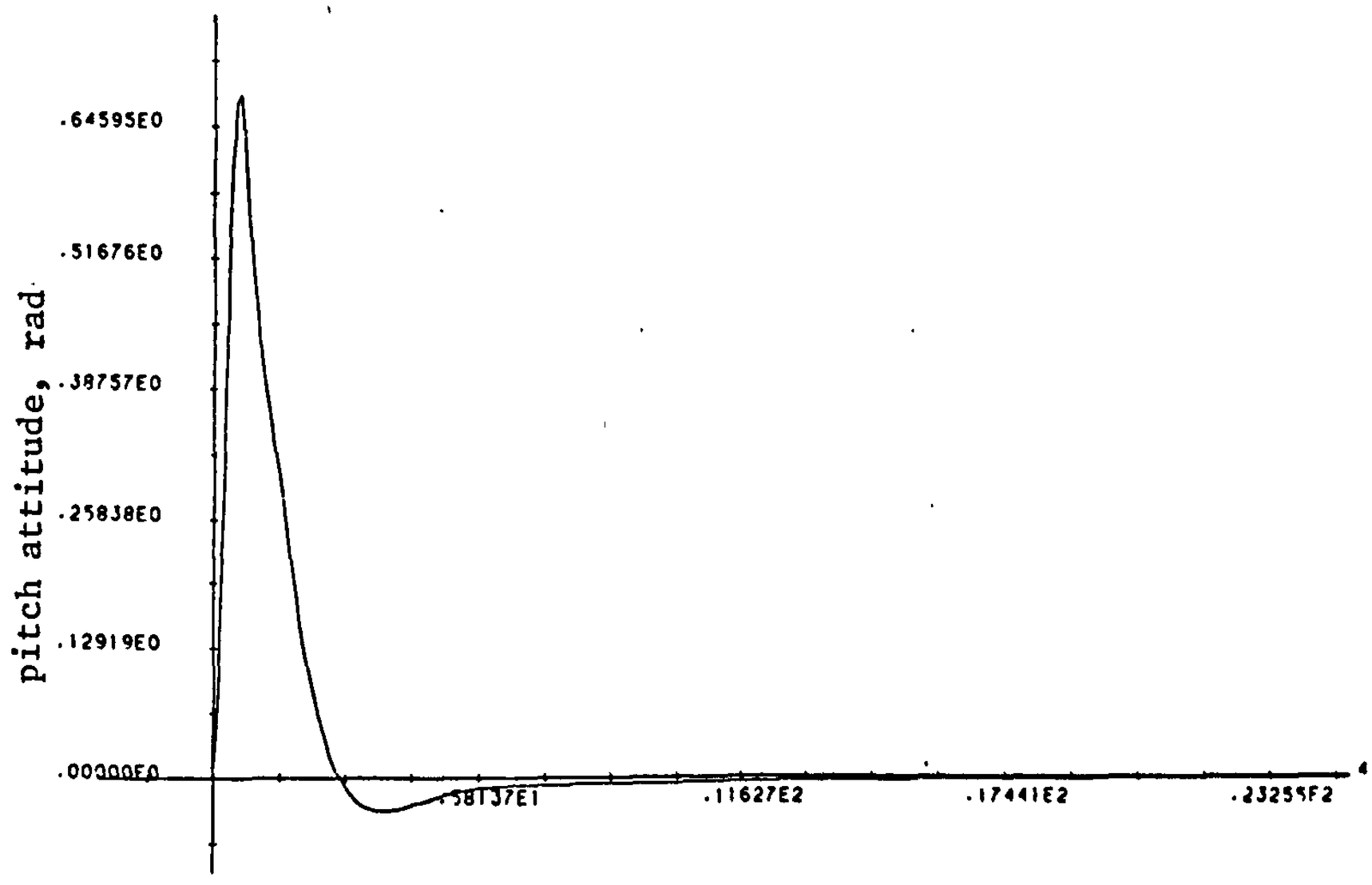


Fig 6.8 cont.

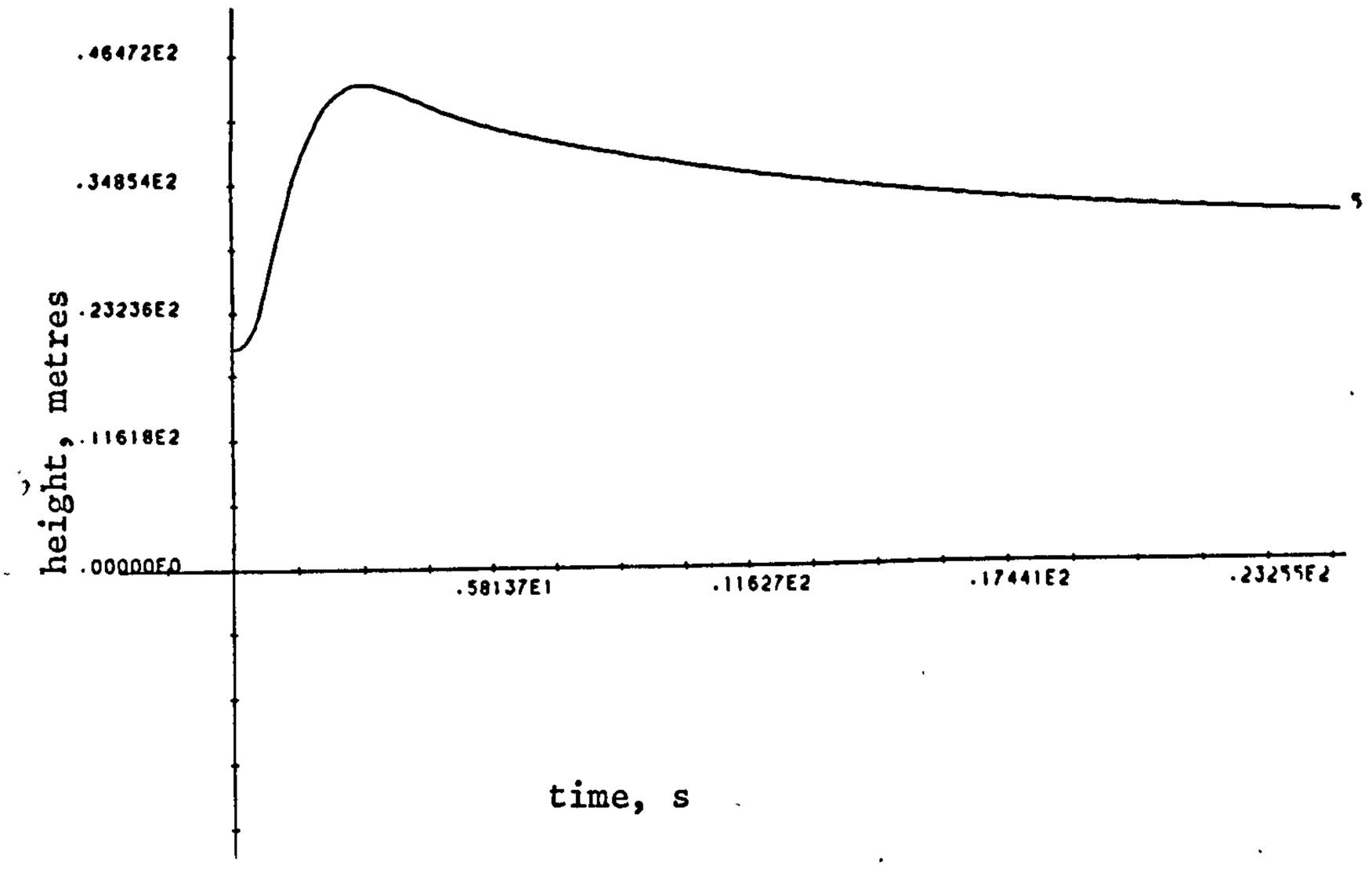


c)



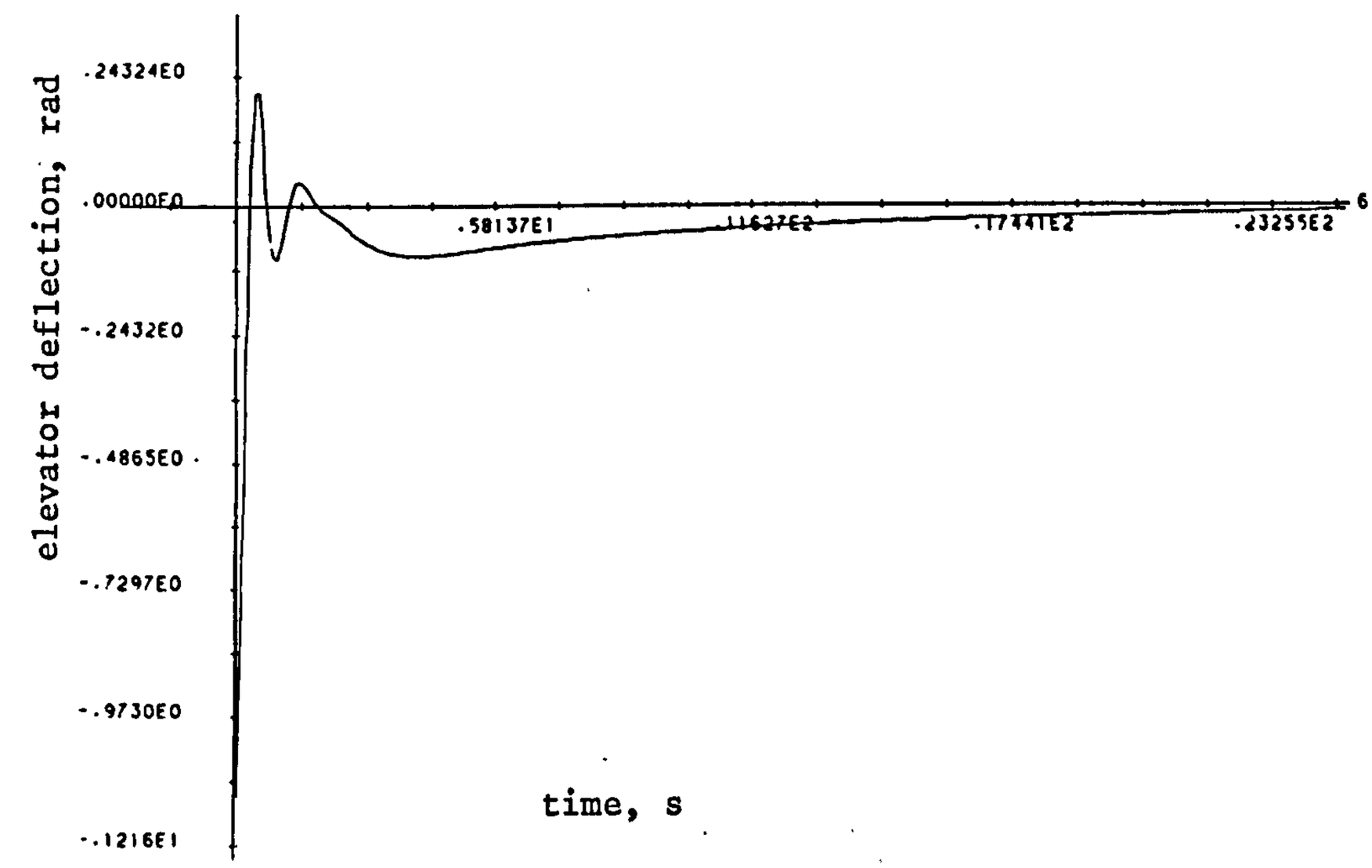
d)

Fig 6.8 cont.



time, s

e)



time, s

f)

The above example illustrates that the optimal control design technique can be extended to include demand following in the case of constant demand signals. This is particularly important in the aircraft problem when many of the system states cannot be driven to zero to fulfil physical constraints (e.g. to maintain lift, etc.). In these cases then we may still apply optimal control to provide given performance and stability. One important point is the steady state offset which may arise and was mentioned above. The provision of integral action (56,58,59) in the optimal control design is possible but has not been considered in the current work and will therefore not be discussed.

6.7 Reduced order Aircraft Models

Before leaving this section a brief discussion will be presented of the possibility of using a reduced order model for the aircraft. We are prompted to examine this possibility since, inevitably, this will lead to a less complex state feedback structure in addition to requiring fewer measurements. It may also be recalled that in Chapter 2 the linearised model of the aircraft was derived and it was shown that it was possible to decompose the state-space model into a number of weakly interconnected sub-systems which allowed us to derive approximations for the modes of the open-loop aircraft. This would seem to suggest that each sub-system could be treated independently and control provided by state feedback around each sub-system. This, of course, ignores any cross-coupling terms but these may be sufficiently small so as not to unduly degrade the control action.

To illustrate the type of scheme envisaged here we shall consider again the lateral dynamics of the Machan vehicle as discussed previously. From section 2.2.2 the parameterised lateral state equations were derived as equation 2.43 a) and are repeated below for convenience.

$$\begin{bmatrix}\dot{v}\\ \dot{p}\\ \dot{r}\\ \dot{\phi}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} y_v & 0 & y_r-U_o & g & 0\\ l_v & l_p & l_r & 0 & 0\\ n_v & n_p & n_r & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}v\\ p\\ r\\ \varphi\\ \psi\end{bmatrix} + \begin{bmatrix} y_\tau & 0\\ l_\tau & l_\zeta\\ n_\tau & n_\zeta\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\tau\\ \zeta\end{bmatrix}$$

The state variables and inputs are as defined previously. Recall that this equation has been derived from a direct consideration of the physical system equations. The approximations to the four dominant lateral modes were then obtained by intuitive arguments. Let us pursue this line of reasoning further here by considering the control of the dominant oscillatory mode, the Dutch roll, only. This motion is generated by the v and r states and can be considered slow when compared with rolling motions, hence it can be assumed that Dutch roll takes place at constant roll, i.e. $\dot{p} = 0$. Equally, neglecting gravitational effects removes the coupling of the spiral mode into Dutch roll. A simplified form of the state equations may thus be used namely :

$$\begin{bmatrix}\dot{v}\\ 0\\ \dot{r}\end{bmatrix} = \begin{bmatrix} y_y & 0 & (y_r-U_o)\\ l_v & l_p & l_r\\ n_v & n_p & n_r\end{bmatrix}\begin{bmatrix}v\\ p\\ r\end{bmatrix} + \begin{bmatrix} y_\tau & 0\\ l_\tau & l_\zeta\\ n_\tau & n_\zeta\end{bmatrix}\begin{bmatrix}\tau\\ \zeta\end{bmatrix} \qquad - 6.56$$

Solving row 2 algebraically and eliminating p from row 3 gives :

$$\begin{bmatrix}\dot{v}\\ \dot{r}\end{bmatrix} = \begin{bmatrix} y_v & (y_r-U_o)\\ (n_v-\frac{n_p l_v}{l_p}) & (n_r-\frac{n_p l_r}{l_p})\end{bmatrix}\begin{bmatrix}v\\ r\end{bmatrix} + \begin{bmatrix} y_\tau & 0\\ (n_\tau-\frac{n_p l_\tau}{l_p}) & (n_\zeta-\frac{n_p l_\zeta}{l_p})\end{bmatrix}\begin{bmatrix}\tau\\ \zeta\end{bmatrix} \qquad - 6.57$$

For the Machan both n_ζ and n_p are zero and hence the B(2,2) term is zero indicating that only the rudder may be used to control Dutch roll. Note that this is not the case in general and hence a two-input v,r sub-system would have to

be considered. Substituting in 6.57 the aerodynamic derivative values for the Machan and taking U_o=33 ms^{-1} gives :

$$\begin{bmatrix}\dot{v}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}-0.277 & -32.98\\ 0.365 & -0.319\end{bmatrix}\begin{bmatrix}v\\ r\end{bmatrix} + \begin{bmatrix}-5.432\\ -9.49\end{bmatrix}\tau \quad - 6.58$$

and introducing a single actuator dynamic for the rudder of the form of equation 6.22 a) gives :

$$\begin{bmatrix}\dot{v}\\ \dot{r}\\ \dot{\tau}\end{bmatrix} = \begin{bmatrix}-0.277 & -32.98 & -5.432\\ 0.365 & -0.319 & -9.49\\ 0. & 0. & -5.\end{bmatrix}\begin{bmatrix}v\\ r\\ \tau\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 10.\end{bmatrix}\tau_d \quad - 6.59$$

The open-loop poles are close to the actual Dutch roll poles for 33 ms^{-1} airspeed at - 0.298 ± 3.5j and an actuator pole at -5.0.

The optimal control design technique can now be applied to the state equation of 6.59 with a desired choice of closed loop pole position as in section 6.5.1 of -0.63 ± 2.42j. Note that the single actuator pole will be the asymptotically infinite eigenvalue. To maintain actuator bandwidth requirements of 10 rads s^{-1} the ρ parameter in the design can be adjusted and a choice of $\rho = 0.0025$ gives closed-loop poles at

$$-10.72 \; ; \; -0.75 \pm 2.75j$$

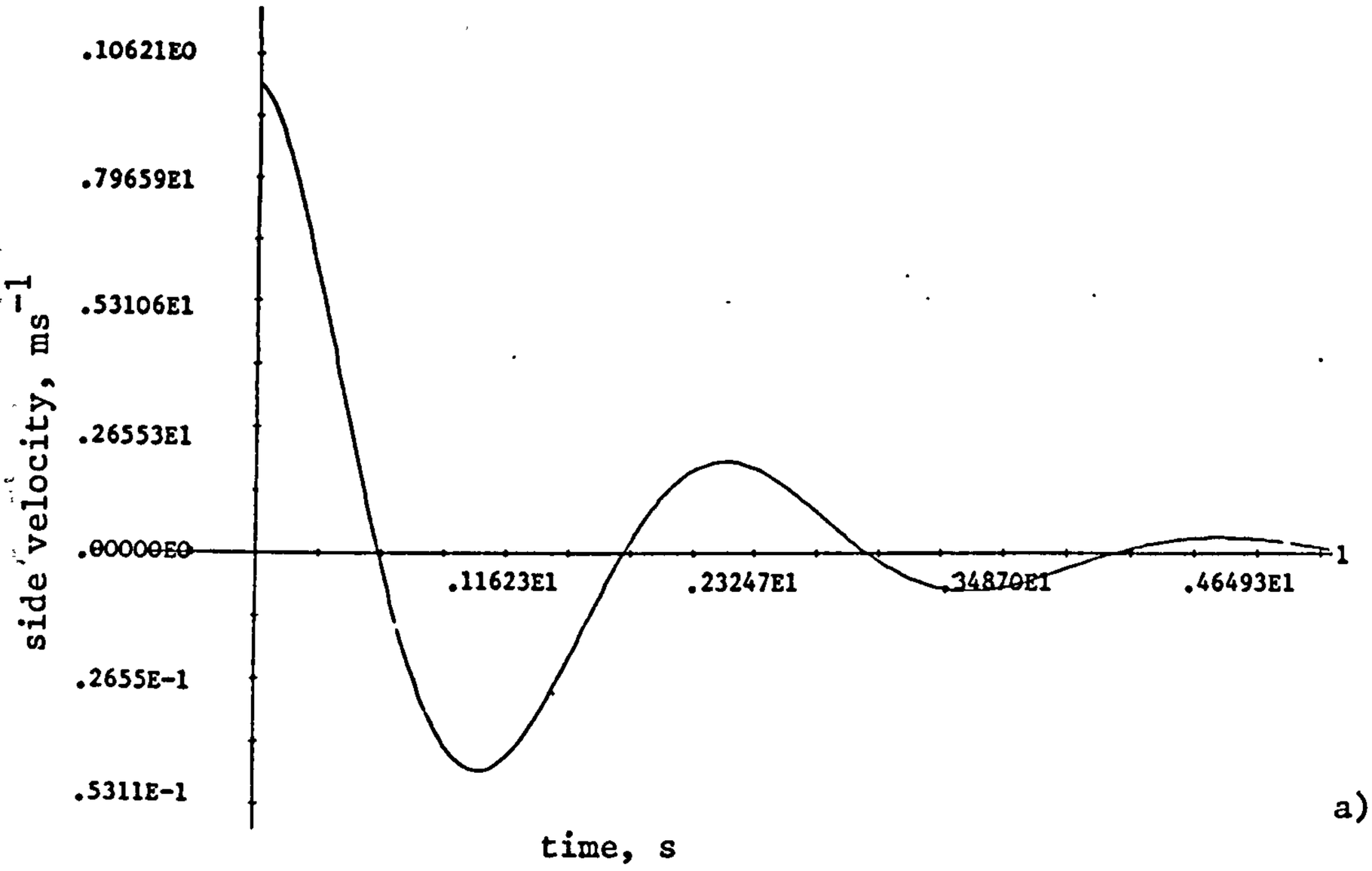
The resulting state feedback gain matrix is

$$K_1 = [0.018 \quad 0.045 \quad -0.66]$$

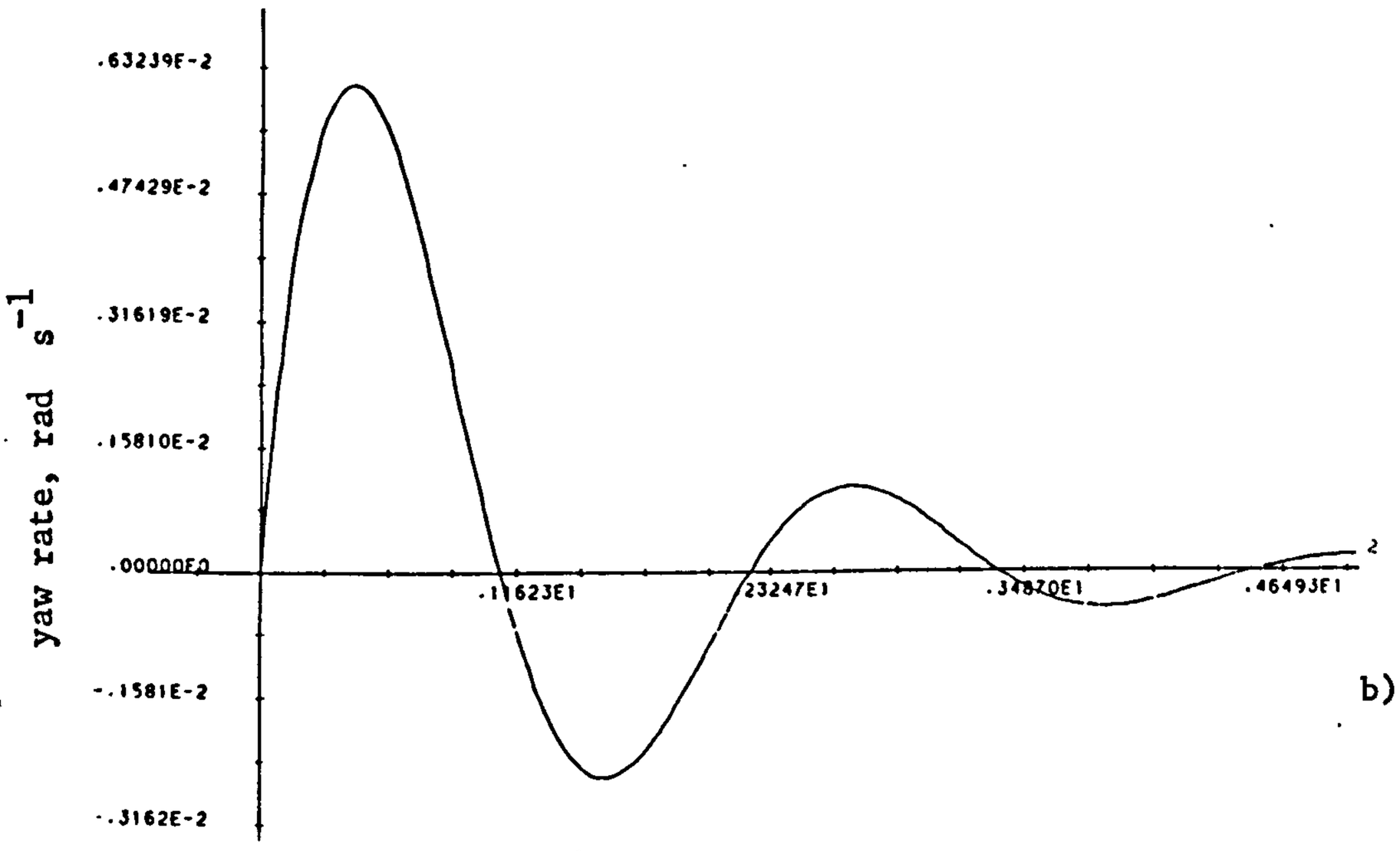
Figs 6.9 show the three state responses of the closed-loop system due to an initial condition of 0.1 ms^{-1} in v. Note that the design achieves an underdamped response with approximately 2.3 secs. period, as desired.

This third order design has thus produced a set of gains which may be employed in the non-linear system with the remaining states open-loop. If the approximations made are reasonable then the reduced order controller may be expected to perform tolerably well. Note, however, that no

Fig 6.9 Reduced Order Dutch Roll Responses
(linear model at 33 ms^{-1} , LQP controller)

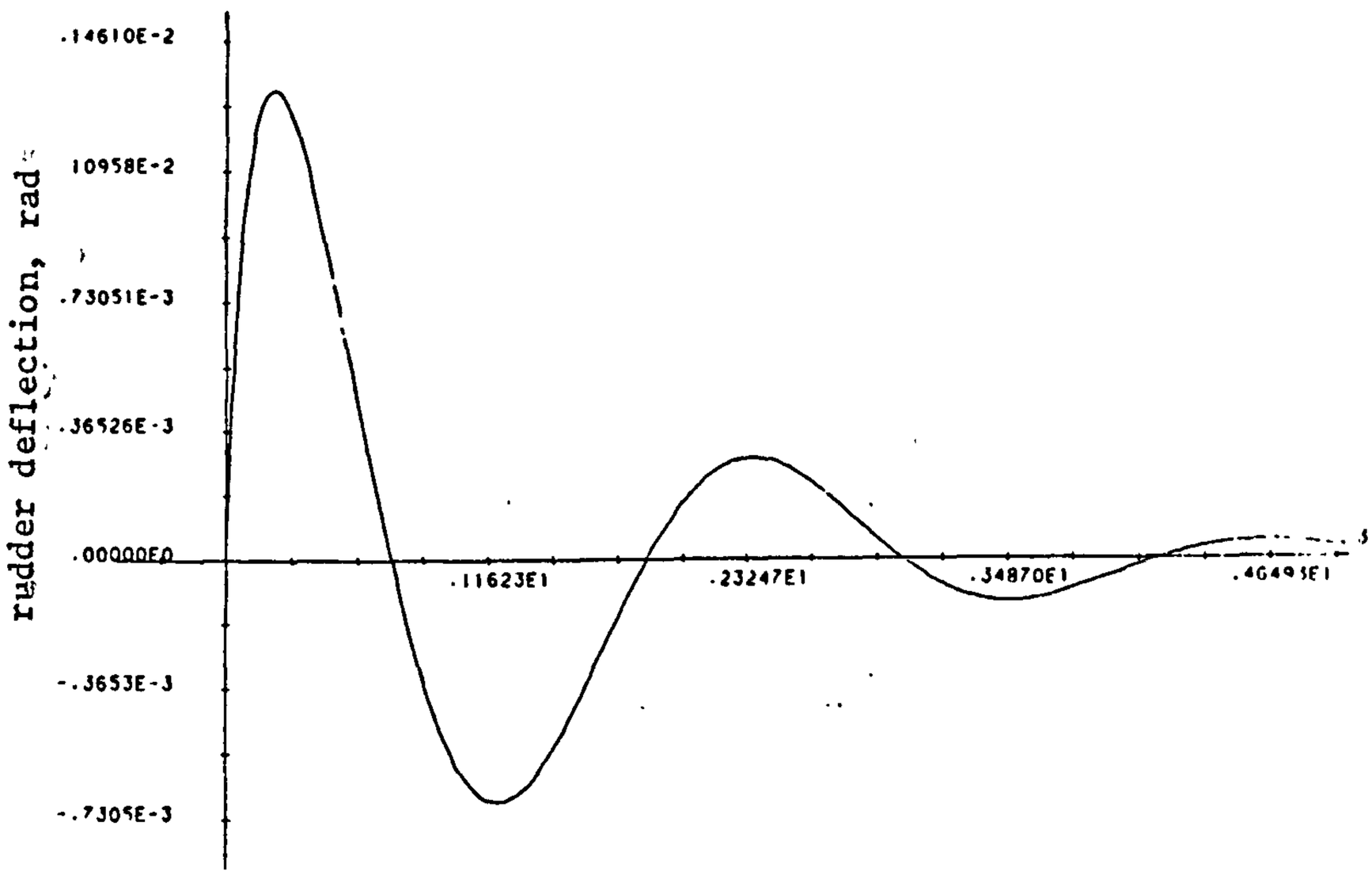


a)



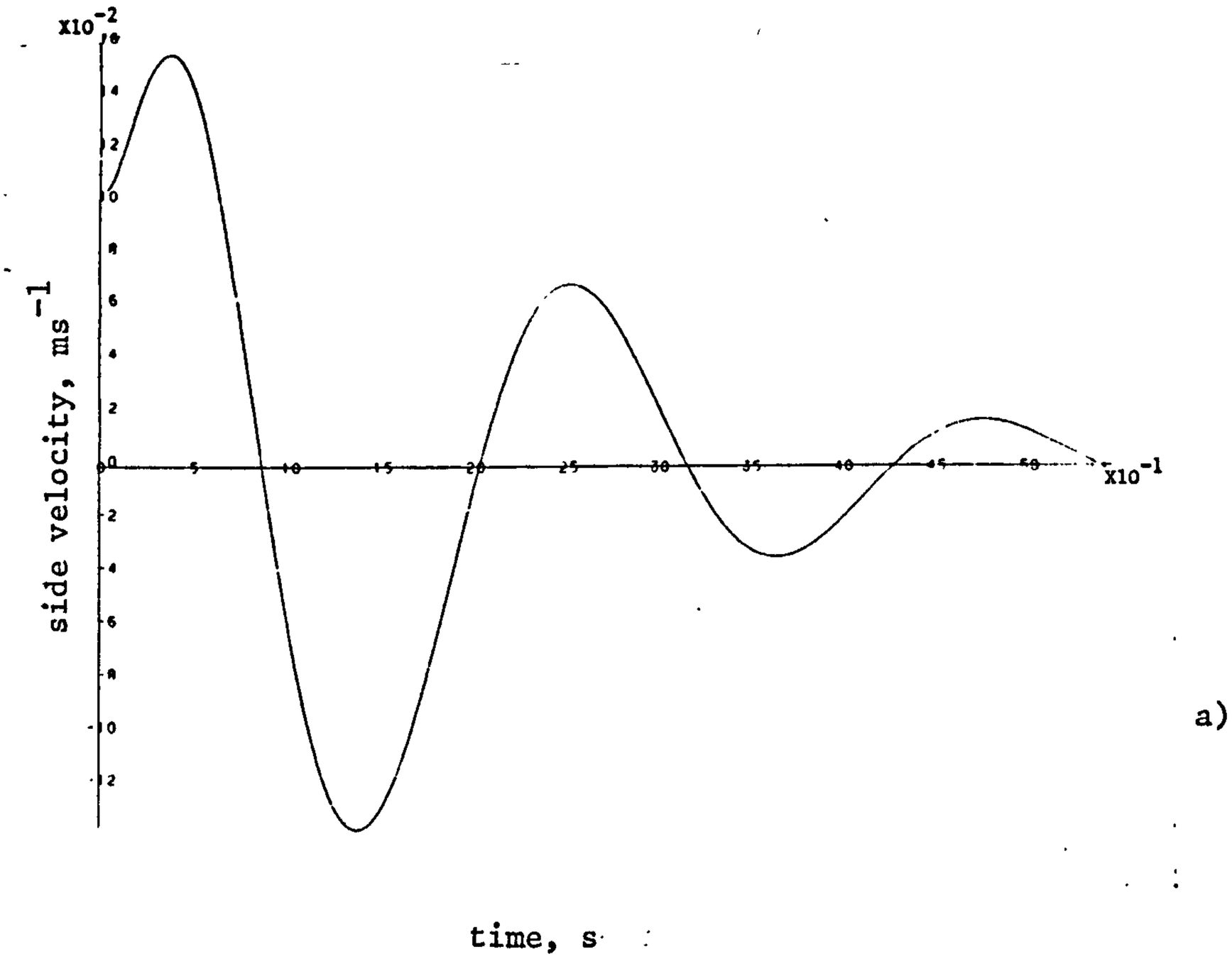
b)

Fig 6.9 cont.


.14610E-2
10958E-2
.73051E-3
.36526E-3
.00000E0
-.3653E-3
-.7305E-3
rudder deflection, rad
.11623E1
.23247E1
.34870E1
.46493E1
time, s

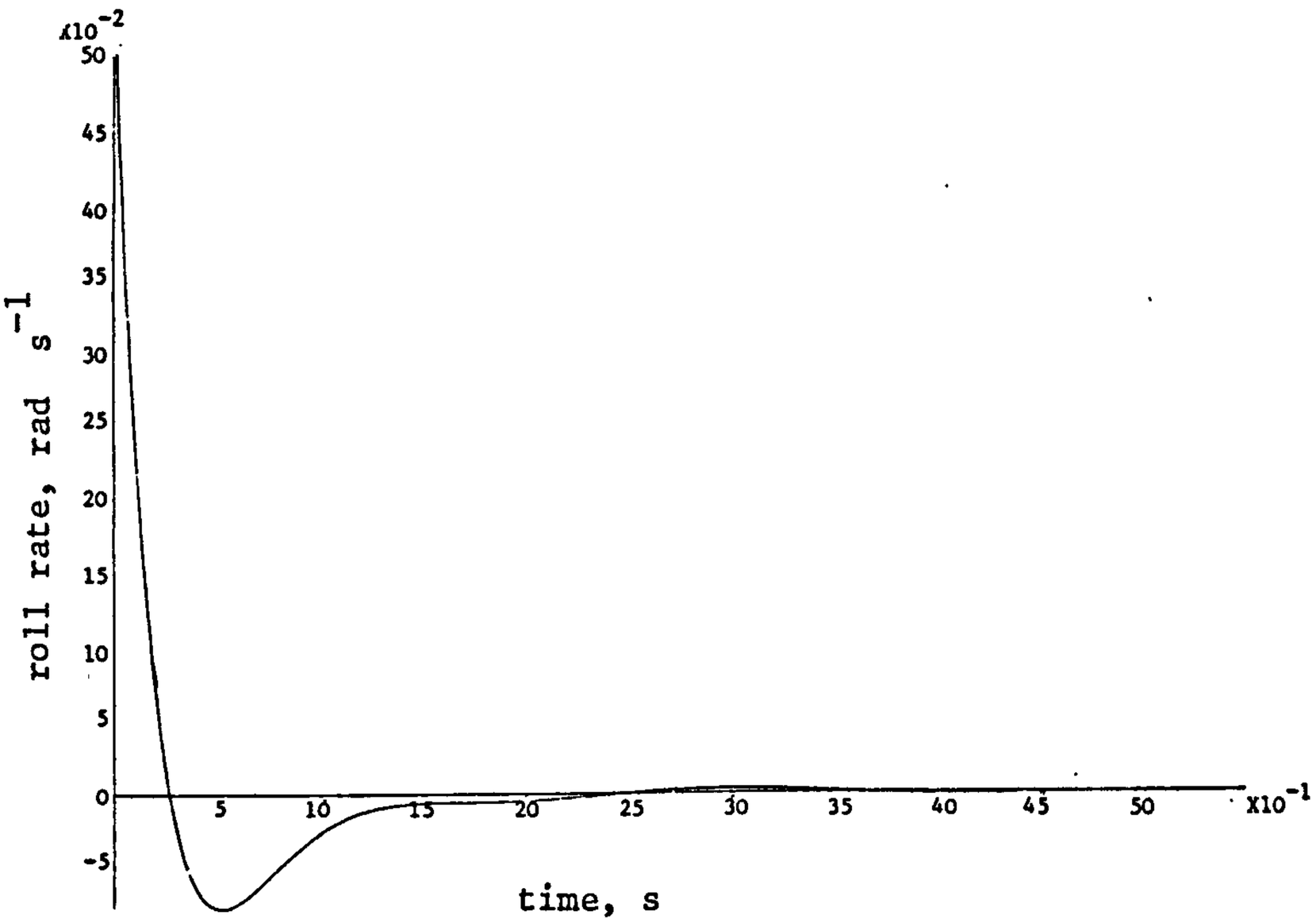
c)

Fig 6.10 Reduced Order Controller State Responses
(non-linear model at 33 ms^{-1} airspeed)



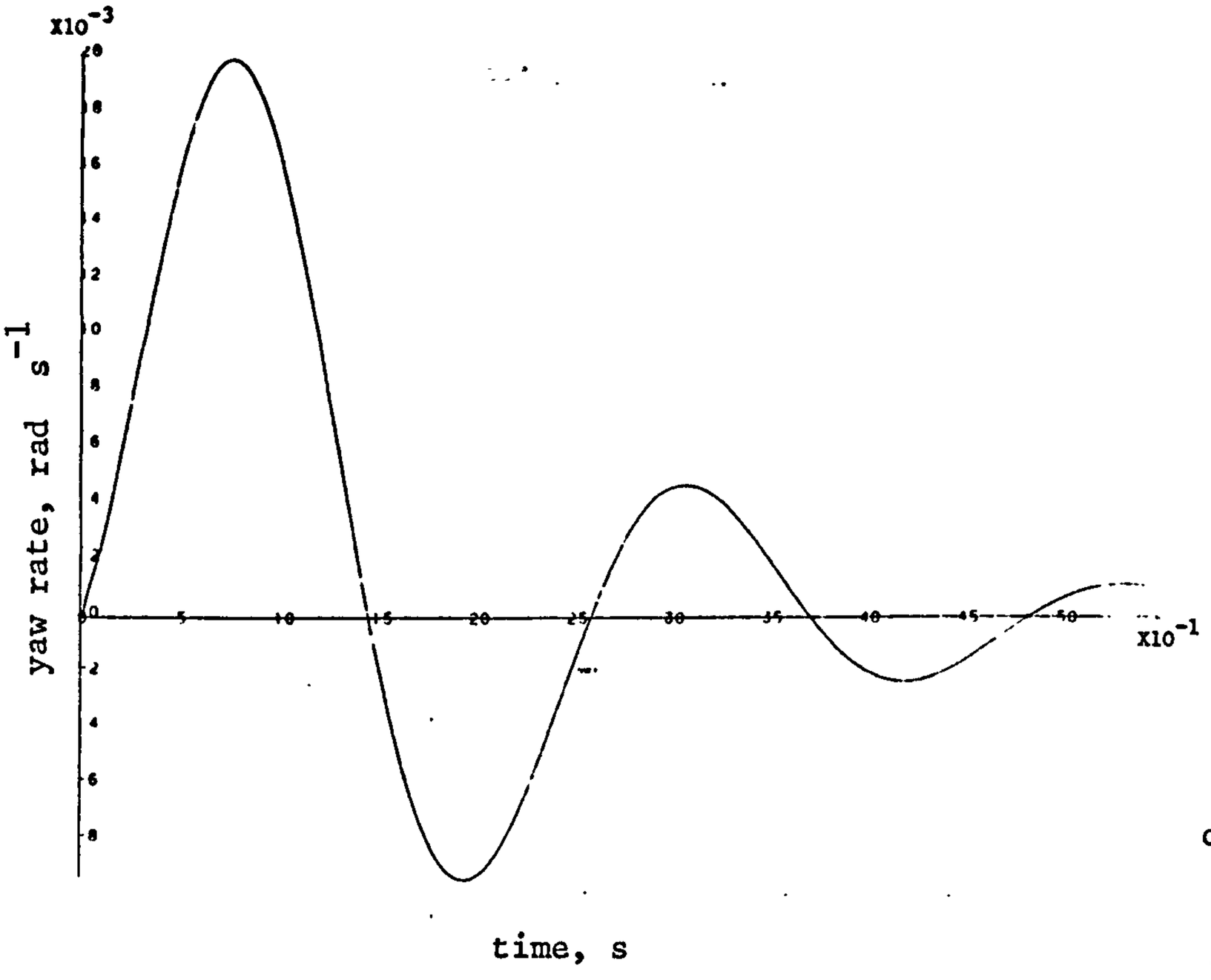
a)

time, s

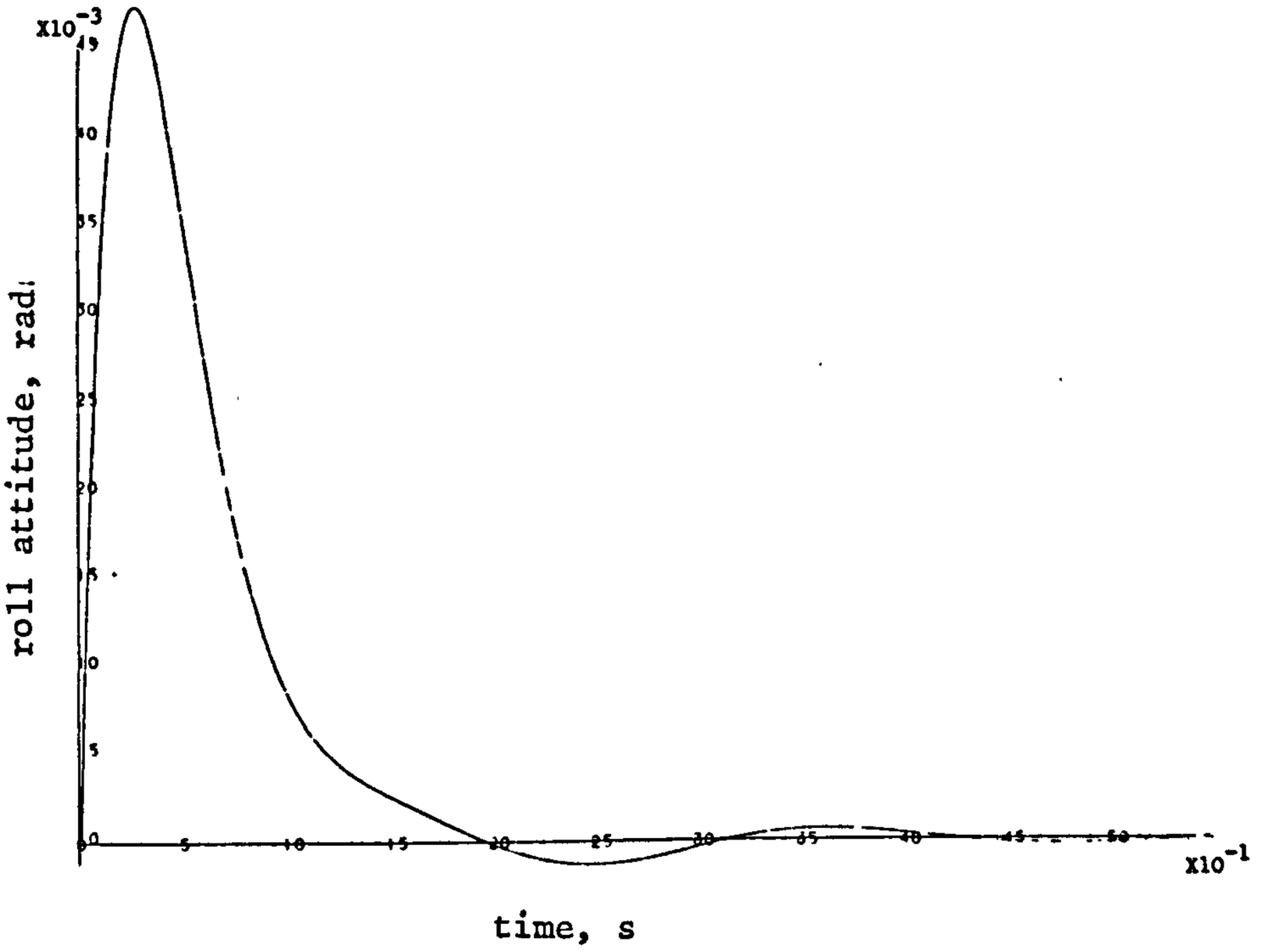


b)

time, s

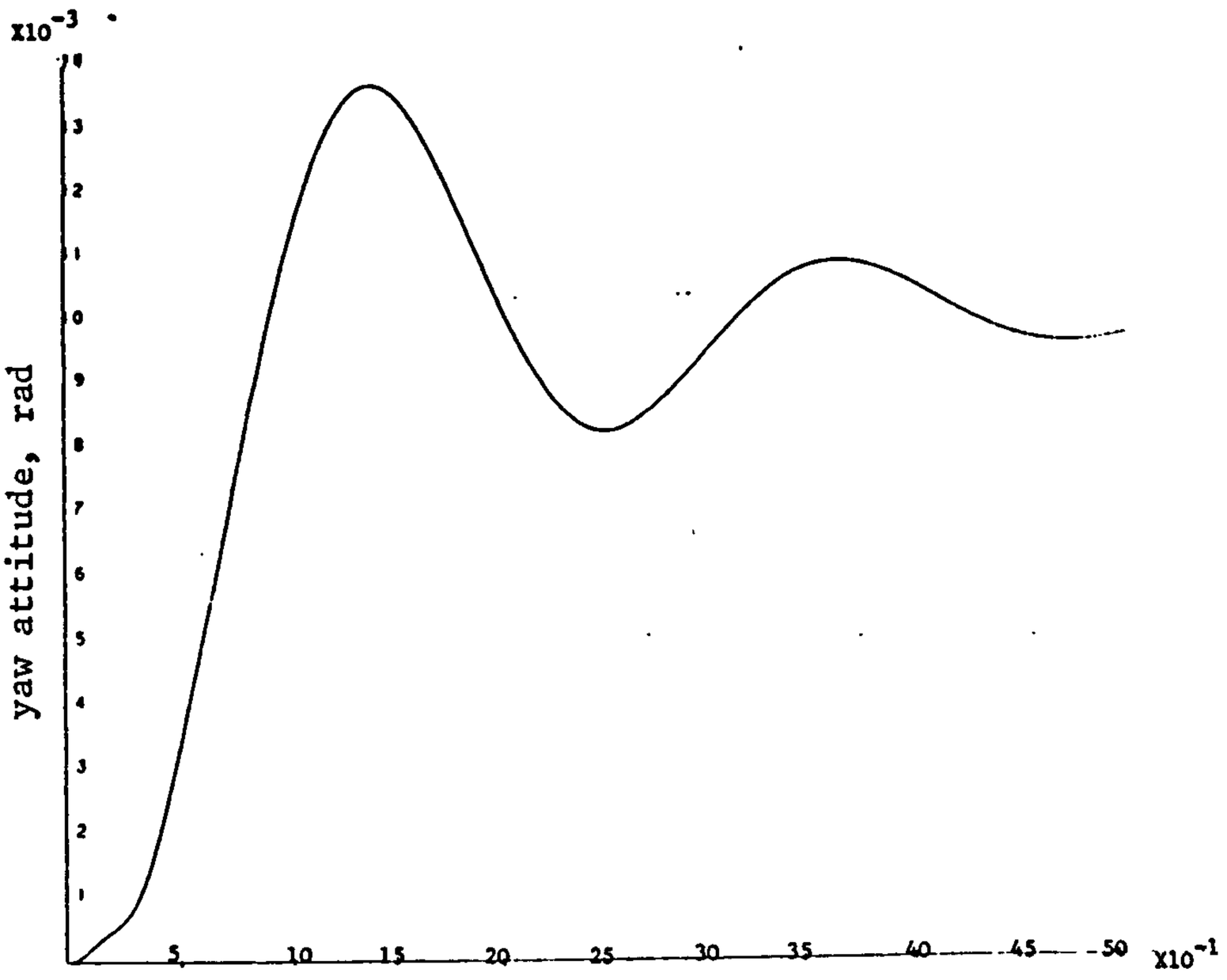

x10^-3
yaw rate, rad s^-1
0
5
10
15
20
25
30
35
40
45
50
x10^-1
time, s

c)

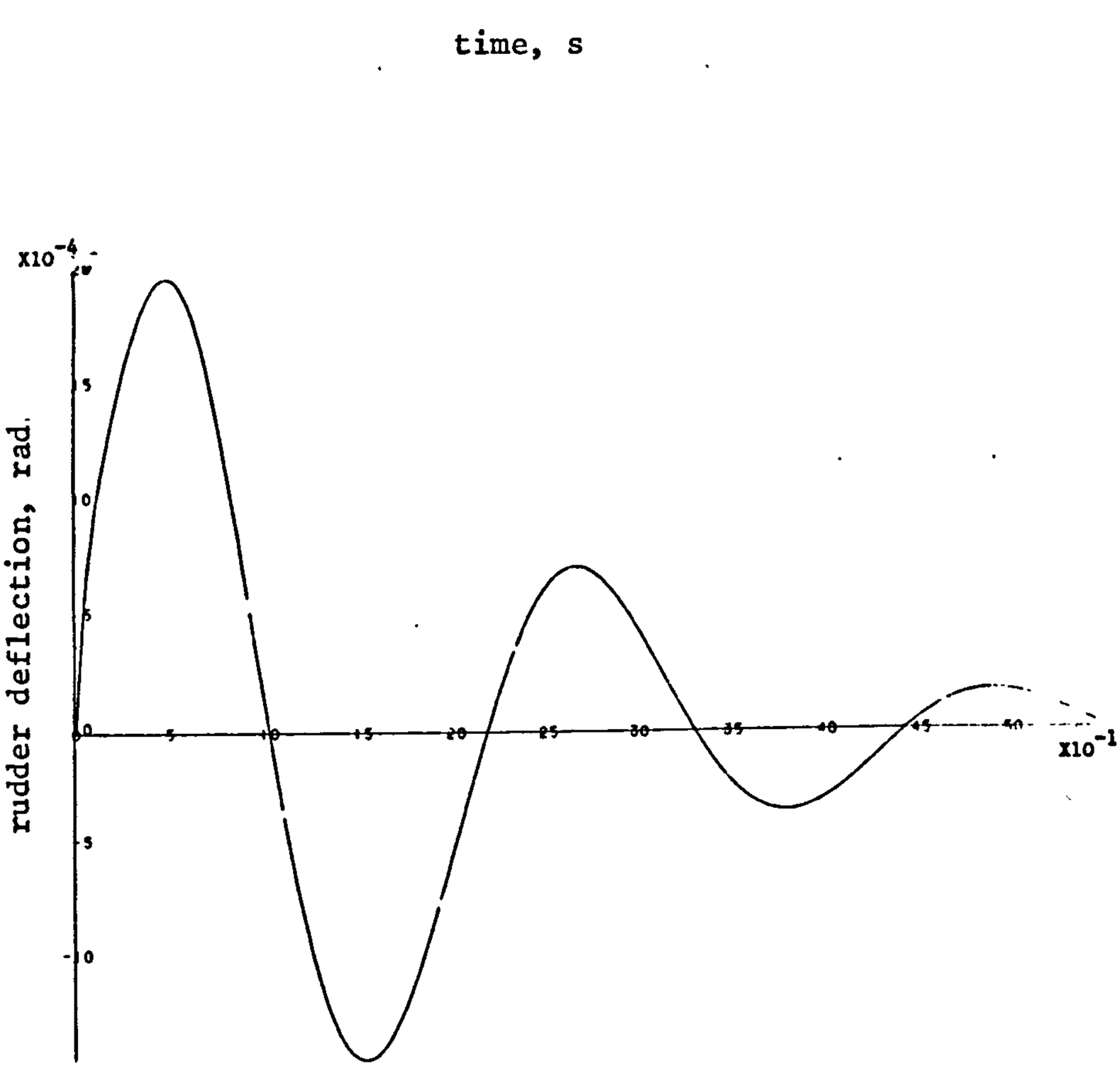

x10^-3
roll attitude, rad
0
5
10
15
20
25
30
35
40
45
50
x10^-1
time, s

d)

Fig 6.10 cont.

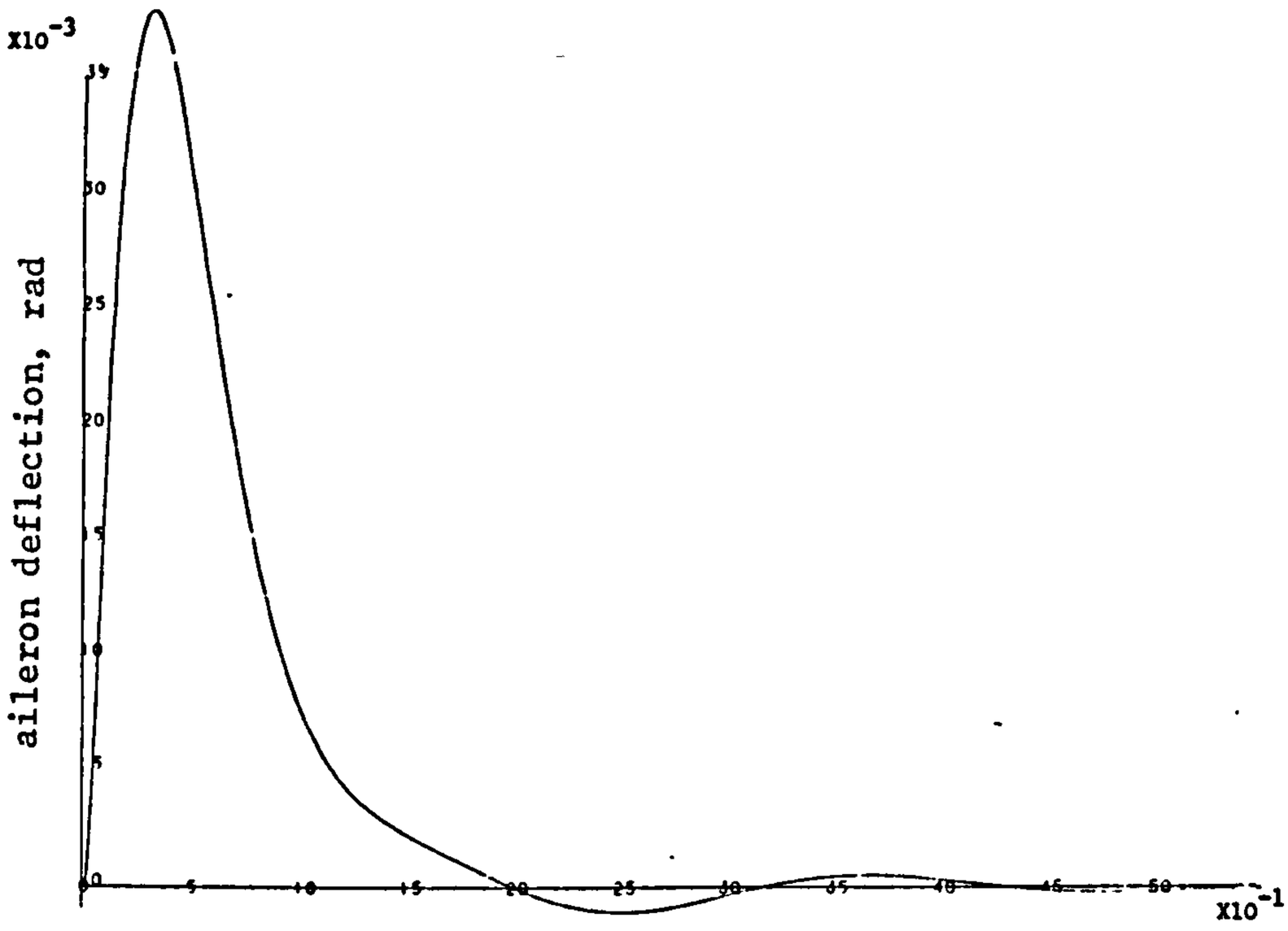


e)



f)

Fig 6.10 cont.



g)

attempt has been made to control the rolling motions and additionally, in our example, the aileron channel is left open-loop. One fairly clear point is that the actuators are normally reasonably well decoupled at least as far as the rudder/aileron are concerned. This is largely due to the particular values of aerodynamic derivatives for the Machan.

In order to test the likely performance of this reduced order controller it was implemented on the non-linear simulation discussed previously. To provide roll channel control a reasonably tight loop was provided from roll angle to aileron. Recall that the spiral mode for the Machan is open-loop unstable hence stabilisation should be possible if the mode is decoupled and controlled by aileron only. Figs. 6.10 a) to f) show the lateral state responses obtained for initial conditions of 0.5 rad s^{-1} in p (roll rate) and 0.1 ms^{-1} in side velocity, v, with 33 ms^{-1} airspeed. Note that the Dutch roll mode shows up primarily on the v and r states and has close-to the damping and period required. Note also that reasonable decoupling of the roll and yaw channels has been achieved with a rapid first order roll response and little evidence of Dutch roll coupling. The actuator responses also demonstrate the relatively high degree of decoupling achievable in this example. Since no feedback of yaw angle has been provided this exhibits a steady state offset but this is small (< 0.01 rad).

It would thus appear that for this example the application of intuitive arguments has led to a reduced order controller with apparently desirable performance. It is only necessary to measure v, r, τ and φ to provide an acceptable SAS system. This is clearly desirable but may not, in general, provide such a useful design technique. A compromise is, in fact, being provided between classical single-loop control and state feedback which, in this case, is valid. The robustness of this scheme is, of course, not guaranteed but it may be speculated that the control would be sensitive to changes in the aerodynamic derivatives directly concerned in equation 6.59 with $n_p = n_\zeta = 0$ and hence may be less sensitive to parameter values than a full state controller which considers all the aerodynamic

derivative terms.

6.8 Summary

In this Chapter the application of optimal control and state variable feedback to the aircraft problem has been demonstrated. In this context the derived controller should be capable of providing SAS and autopilot functions, i.e. should provide both demand-following and regulatory action. The ability of state feedback and optimal control to provide arbitrary pole placement and eigenvector assignment has been clearly demonstrated and a number of design techniques discussed in some detail. The ability to provide demand-following has also been indicated. Whilst not wishing to denigrate the power of such techniques a number of shortcomings have been indicated. These are principally :

i) The inability to guarantee robustness in the presence of time-varying systems.

ii) The relatively complex controller structure.

iii) The limited applicability of most design techniques when faced with, for example, non-minimum phase systems.

iv) The lack of guaranteed stability margins, again due to parameter changes in the controlled system.

The above aspects lead us to question the applicability of these design techniques to aircraft flight control particularly when coupled with the likelihood of unmeasurable state variables. The power of such schemes, however, must not be underestimated and in Chapter 8 the robustness problems are considered in greater detail.

CHAPTER 7

Variable Structure Systems

7.1 Introduction

Over the past 30 years a vast body of control systems theory has been developed around essentially linear analysis. The motivation behind using such techniques is principally to provide analytically tractable designs which retain some measure of elegance and simplicity. With such simple design tools engineers can provide control systems which fulfil specific requirements in terms of stability, speed of response, steady-state error, etc., i.e. the classical measures of system performance. In order to apply such linear design tools to a given problem one must, necessarily, have available a suitable linear model of the particular system under consideration. Such models can normally be formulated by tests on the system or by straightforward physical analysis. To a first order approximation the resulting model is then linear, or at least linearisable. This latter modelling technique is exemplified by the discussion of Chapter 2 where the physical system model of an aircraft was developed from first principles, together with a discussion of suitable linearising approximations and the derivation of a linear state space model. We are now open to employ any linear control design technique we have available (examples are classical s-domain/w-domain analysis as in Chapter 5 and state feedback as in Chapter 6) to provide closed-loop controllers capable of satisfying, in some sense, a set of performance objectives. The hope is then that by employing this controller with the real system (which is likely to be highly non-linear) a similar performance can be produced. In many cases these linear controllers achieve adequate control of the physical system and have been shown to provide good stability margins. A further aspect of the linear model is that it may be time-varying, i.e. some parameters of the physical system may change as a result of

steady-state operating point changes, etc. How the linear controller now performs is then open to some speculation. Much work has been done in the general areas of system modelling and identification (5) in addition to the robustness and sensitivity of given controller/system structures. Chapter 8 discusses some of these aspects with regard to the aircraft example. It is however generally true that a controller designed around a nominal linearised model at a particular operating point will not provide the specified performance when faced with the physical non-linear time-varying system. Schemes have been proposed which are adaptive and change the controller parameters as a result of changes in the system to be controlled (14). The stability aspects of such systems are, however, a problem. For the modern avionic system environment, where a high degree of confidence in the control-loop is required, there is still little confidence in the value of such adaptive schemes.

The inability of simple linear control theory to provide parameter insensitive control has led to the examination of other possible techniques. Notable amongst these are the non-linear or switched mode control designs which are receiving attention in the literature at present (60-66,125-130). Such schemes are characterised by a control structure which may be considered as a switched 'gain' element in the loop. This switching gives rise to discontinuities in the control law, these being determined by a consideration of a 'switching function' which is normally chosen as a function of system states and/or the derivatives of these states. The simplest form of such control can be conceived of as relay switched or 'bang-bang' type control which has some highly desirable features in some applications and is characterised by a switching control action. A considerable amount of theory is currently being developed to provide analytical design tools for such systems (67-73). Of particular interest is the so-called 'sliding mode' behaviour in which the closed-loop response becomes insensitive to changes in certain plant parameters. Changes in the nominal operating point, for example, thus have no effect on ultimate system response, a

highly desirable feature. The stability of such schemes must, of course, be examined carefully.

To assess these techniques and provide an insight into the design aspects, the following discussion considers a number of design schemes principally with application to the Machan aircraft example. Initially the ZOC and VICTOR (18,74,75) schemes will be investigated along with simple switched mode control and some recent sliding mode theory. Firstly, however, consider the problem itself by examining the aircraft system and likely performance requirements.

7.2 Linear Aircraft Flight Control and its Compromises

In Chapter 5 the use of classical design tools for the development of linear closed-loop flight control systems was examined. It will be recalled that these heuristic designs consisted of the application of frequency domain techniques, principally the s-domain diagram, to the analysis of a number of single-loop controllers providing acceptable responses in pitch, yaw, roll, etc. The performance of the resulting controller being assessed with regard to stability, steady-state error, damping of oscillatory modes, etc. This type of essentially linear modelling and design is typical of conventional flight control system synthesis. It was also pointed out that as a secondary consideration the disturbance rejection of the system was important. The aircraft in flight being subjected to external disturbances from, for example, atmospheric turbulence and also to internal influences such as noisy measurement devices. To minimise the response of the aircraft to such effects requires that a relatively low bandwidth be set on the controlled system's response. This is undesirable since we degrade the responsiveness of the aircraft control system itself. Such conflicting objectives lead to inevitable engineering compromises which must provide adequate responsiveness whilst maintaining good disturbance rejection.

An additional problem in the aircraft system is that,

as mentioned previously, the parameters associated with the aircraft, principally the aerodynamic derivatives, change as a result of changing flight regimes. To maintain control in such systems as these gain-scheduling techniques are invariably used in which the controller gains are linked to, for example, airspeed and thus change in some well defined manner with these variables. It is the designers' task to consider a number of different flight conditions and determine the gain values which provide a sensibly constant response from the airframe. The gains may be changed continuously, according to some law, or in discrete steps. Such scheduled systems provide adequate perfomance but may be somewhat conservative and not allow the maximum to be achieved from a given aircraft configuration.

To provide adequate system performance designers have used the techniques of linear system analysis to develop compensating networks which may be cascaded with the controller, these aiming to provide a trade off between the gain and phase characteristics of the closed-loop system. Such techniques can often provide a 'better' control structure and improve the ultimate performance achievable. The gain-scheduling problem still remains however and the need for such systems was discussed in Chapter 5.

The need for parameter insensitive control in this application is thus clear. If, along with this, improved disturbance rejection is provided then so much the better. The gain-scheduling type techniques may already be indicating how such control may be achieved. The change in controller structure as a result of scheduling may be a desirable attribute of any parameter insensitive control. How then are we to extend this apparently advantageous property to a generalised design technique and how might such systems perform? Initially, let us review two techniques which, although heuristic, would appear to show some promise of insensitivity, namely the ZOC and VICTOR control schemes.

7.3 ZOC and VICTOR

7.3.1 ZOC (Zero Overshoot Control) (75)

Whilst heuristic by nature this control policy is based on the intuitive observation that for systems with lightly damped oscillatory responses the likely occurence of an overshoot in the response is predictable by evaluating the rate of change of the system error signal. The error signal in any closed-loop linear system will tend to either zero or some steady-state offset. For oscillatory systems the error will exhibit oscillations about this steady value and the rate at which the error is approaching equilibrium will indicate the likely occurence of a transient overshoot. Hence, by monitoring the error rate it should be possible to predict overshoot and modify the system structure suitably to avoid this occurence. Note here the implication that it is required to evaluate the error signal derivative.

For a particular structure of system a mechanisation of the above scheme can be obtained in a reasonably straightforward manner. Consider the archetypal system shown in Fig. 7.1. The measured variable, q, is contaminated by sensor noise, n, and is also affected by external disturbances on the system. To remove the sensor noise a first order low-pass filter is included in the forward path. A simple proportional controller, gain K_c, is also included and a first order actuator then delivers the control action to the system.

The response of the system due to external disturbances improves markedly as the gain, K_c, is increased. The gain is however also constrained by system performance requirements such as stability, damping of dominant modes, etc. It is therefore desirable to provide some form of compensation to improve the system response. One standard technique is to provide phase advance compensation by replacing the noise filter by a generalised lead-lag network of the form

$$\frac{(1 + s\,T_1)}{(1 + s\,T_2)} \qquad T_1 > T_2 \qquad - 7.1$$

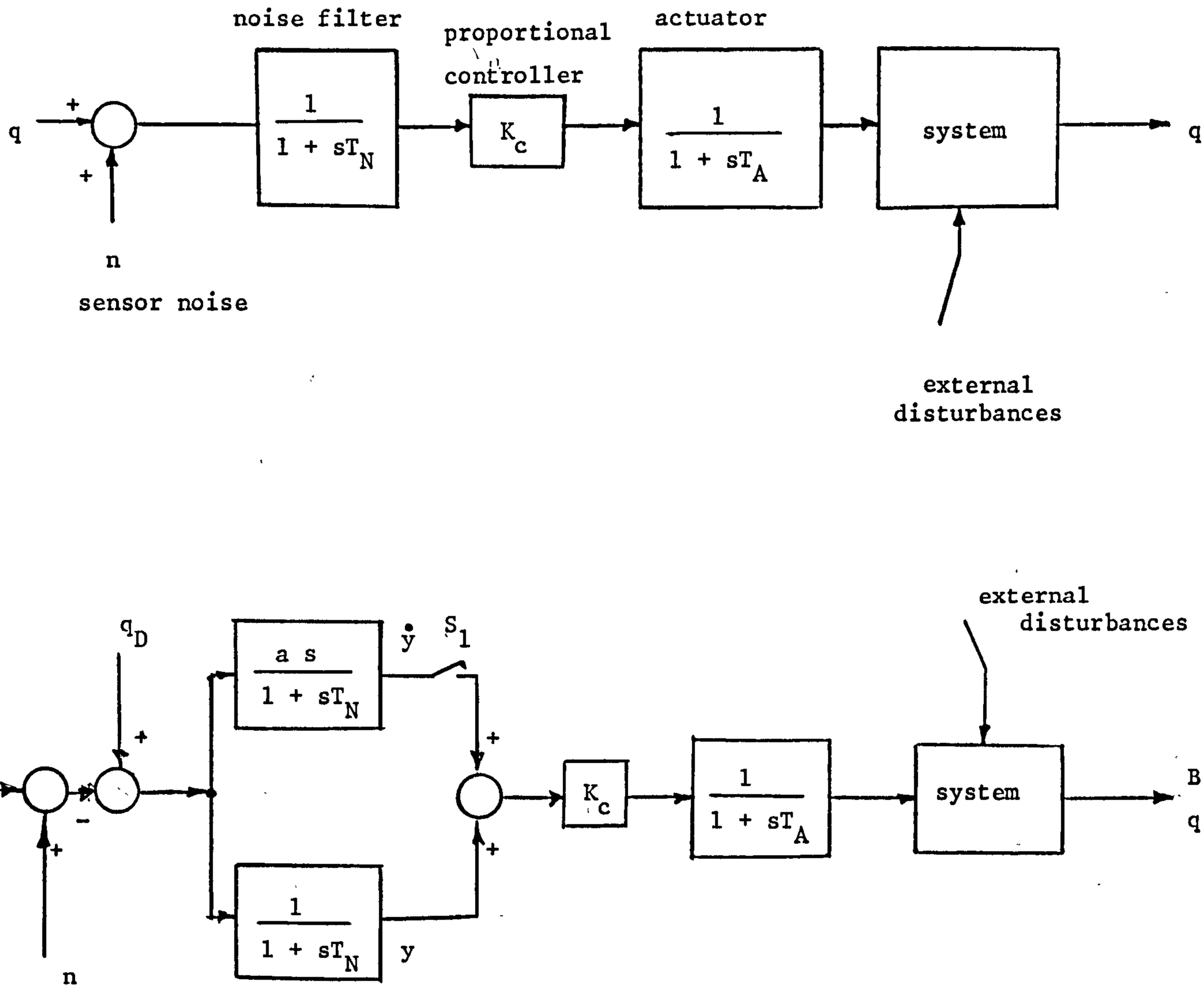


Fig 7.1 Archetypal System Structure

Fig 7.2 Archetypal ZOC Scheme

This has the advantage that the actuator mode, which may introduce undesirable phase lag at low frequencies, may be removed and replaced by a faster mode with consequent improvement in response and stability. The disadvantage of such a solution is that the actuator activity due to measurement noise is increased and hence overall measurement noise rejection is degraded. To solve this problem it can arranged that the phase advance is introduced only when control action is required and removed otherwise. To do this a solution such as that in Fig 7.2 may be adopted.

When the switch S_1 is open the system retains the form of Fig. 7.1. When S_1 is closed, however, the phase advance is introduced with consequent improvement in system response, some freedom being afforded by the choice of the 'a' parameter. The problem of identifying when S_1 should change state still, of course, remains. To solve this a second observation is used in that when control system response is required, the system 'error' signal is normally decaying towards its equilibrium, as noted above. Remembering that ultimately it is desired to reduce overshoots S_1 would thus be closed when the onset of an overshoot is indicated by too rapid a decrease in the 'error' signal, y, this action modifying the system structure. This should also improve the noise rejection of the system since sensor noise will normally give rise to rapid increases in the 'error' signal, y. The state of S_1 can thus be determined from a control law of the form

$$\text{close } S_1 \text{ if } (\text{sign}(k_1 y + \dot{y}) \neq \text{sign } (y)) \qquad - 7.2$$

where k_1 is a design constant determined heuristically. Note that the value of $\dot{y}$, the derivative of error signal, is required. Inspection of Fig. 7.2 should reveal that this is available at the input to the switch S_1.

The performance of this system is relatively easily demonstrated by initially neglecting the system dynamic and investigating the closed-loop form of Fig. 7.2 i.e. with points A and B joined. For the purposes of this exercise an arbitrary choice of T_N = 0.1 s, T_A = 10 s and K_c = 700

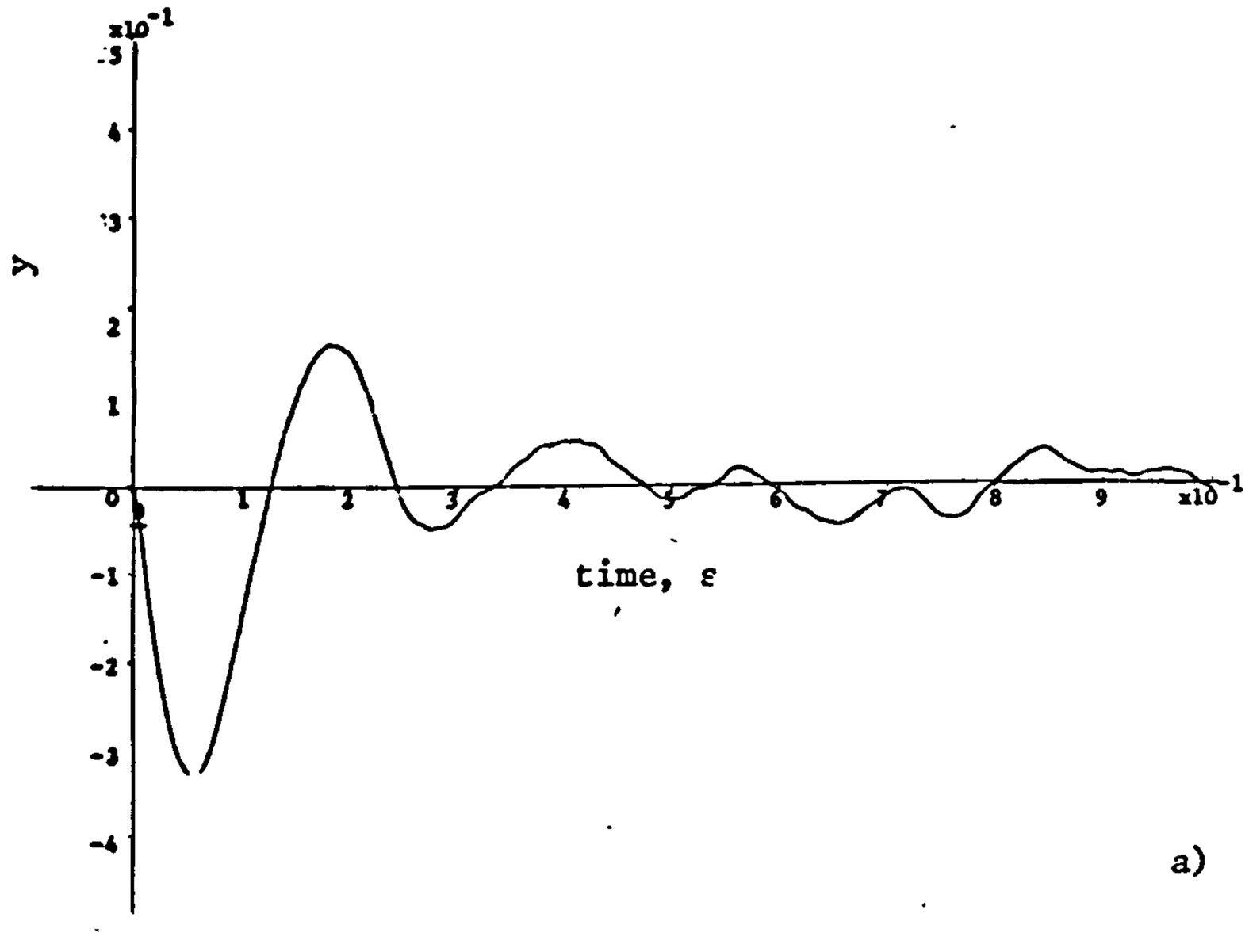
gives, for the unswitched system, a second order underdamped response with closed-loop poles at approximately $-5.0 \pm 26\,j$ ($\zeta = 0.19$, $\omega_n = 26$ rad s^{-1}). Additive sensor noise, n, was also provided as zero mean Gaussian white noise with 0.01 rms amplitude. The k_1 and 'a' parameters were chosen as k_1=0.75, a=10., which provided reasonable overshoot control in the 'y' response. Figs. 7.3 a) and b) show the y and q responses due to an initial condition of 1.0 in q for the unswitched system. Note that the y response shows large overshoots about the steady state as would be expected. Noise rejection is also fairly poor.

For the switched system, Figs. 7.4 a) to c) show the system response for the same initial conditions in q. The reduced overshoot in the y response is clear although the output noise performance is arguably worse than in the unswitched case. Note however that the overall actuator rate demands are reduced. A better choice of k_1 and 'a' may improve this still further. Fig. 7.4 c) shows the S_1 activity for the responses (S_1 being closed is indicated by a '1' on the plot).

It would therefore appear that employing this type of 'piecewise linear' controller affords some measure of improvement in general system performance in the sense of decreased settling time, better steady state accuracy since higher gains may be used, improved noise immunity, etc. There does remain the question of system stability, however. Since the inclusion of ZOC affects only the zeros of the open-loop system and phase advance compensation will improve stability margins, it may be speculated that the ZOC policy will improve system stability. It was recently demonstrated in a paper by McLean (76) that this is indeed the case and application of Popov's stability criterion shows that ZOC may be asymptotically stable, this depending upon the particular system it is used with. Some care may however be necessary when applying this control policy.

The second policy, namely VICTOR, will now be examined.

Fig 7.3 State Responses for Unswitched ZOC



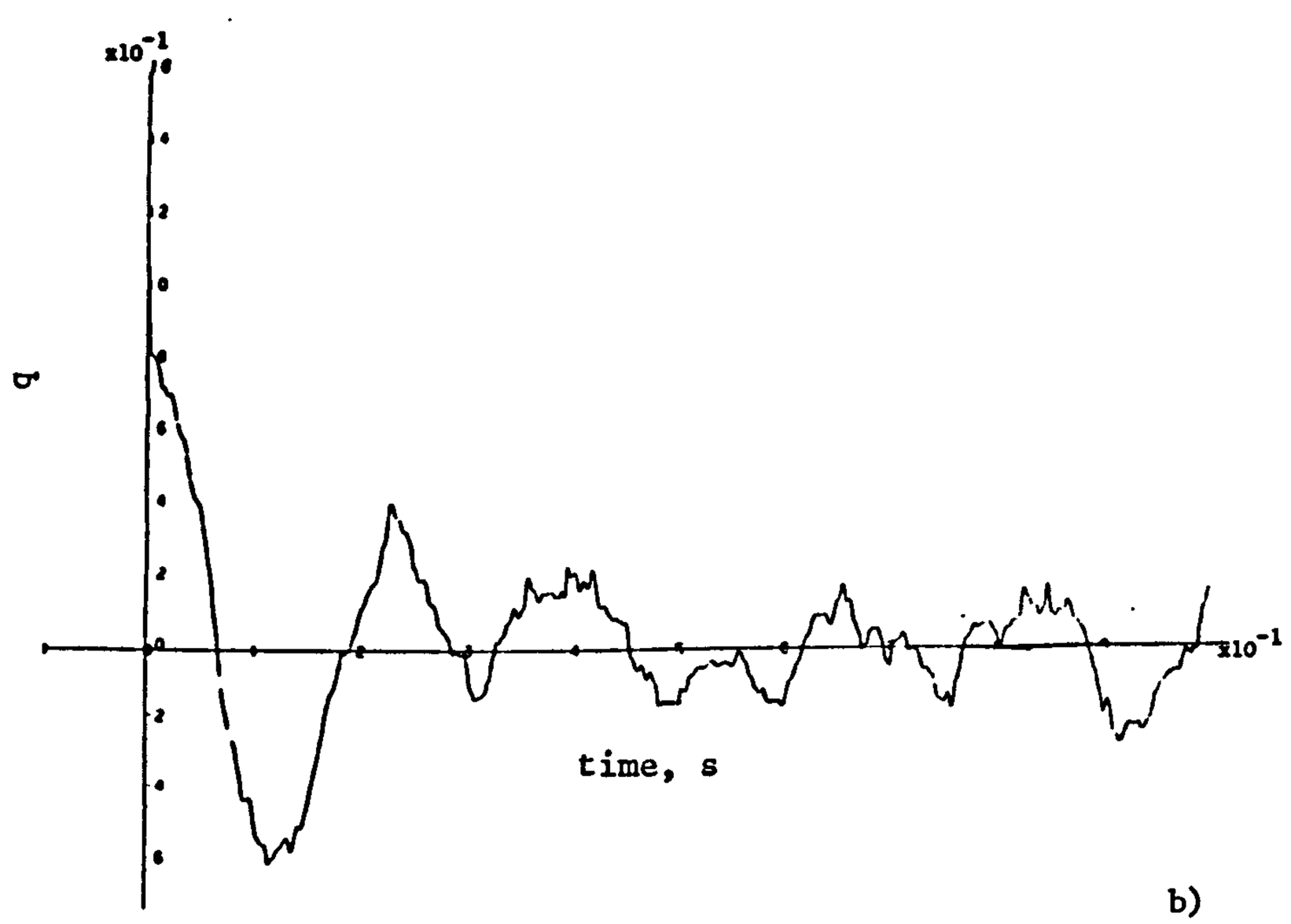
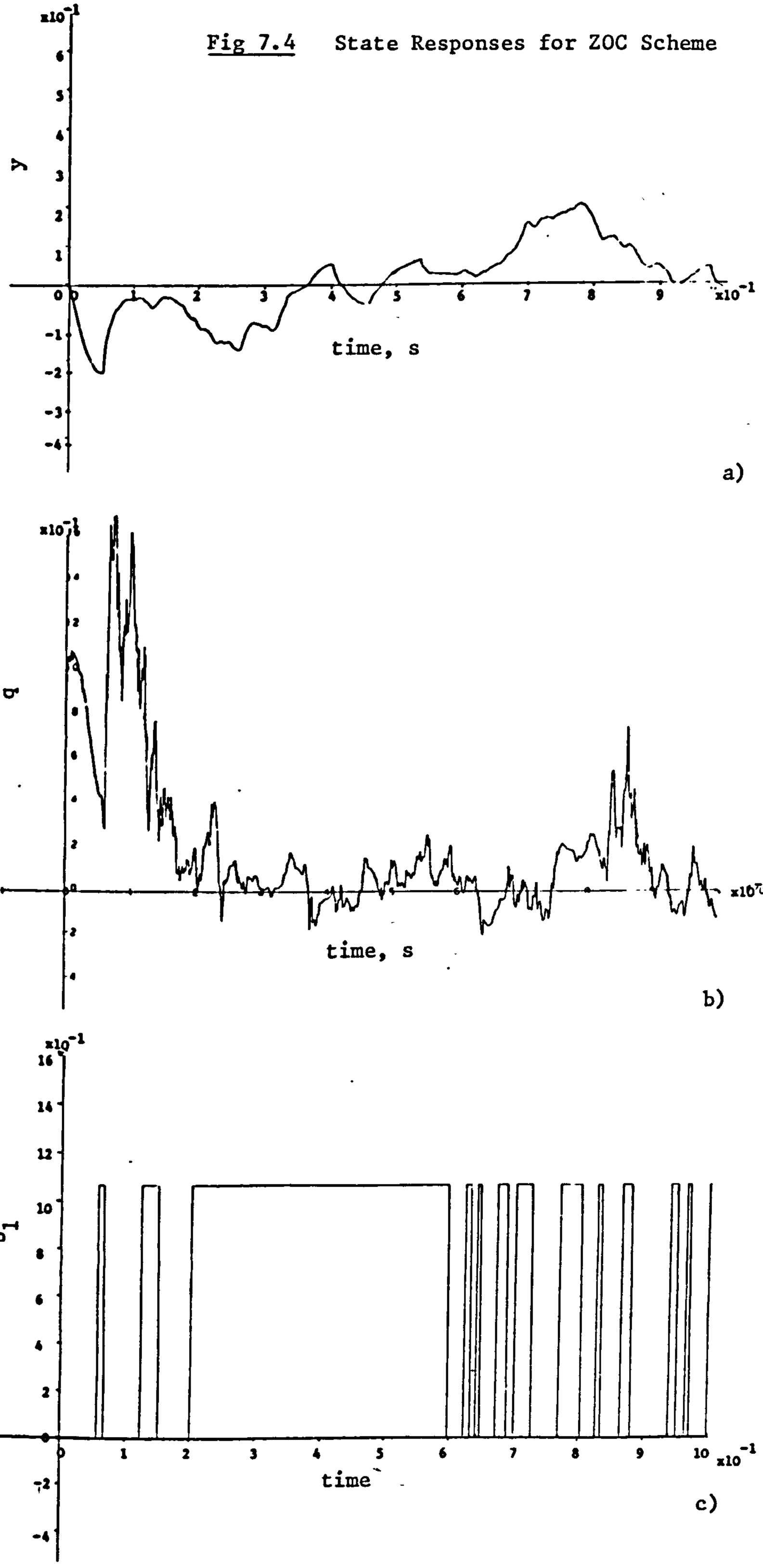


Fig 7.4 State Responses for ZOC Scheme



7.3.2 VICTOR

VICTOR is an acronym for Variable Integral Control To Optimise Response and again is a largely heuristic technique. The scheme itself was devised by Gill (18,74) for application to flight control systems.

In some senses this control policy may be likened to adaptive control since some of the controller parameters are varied as a result of an on-line measurement of the system performance. As its name suggests the controller structure is nominally of the Proportional + Integral type. The amount of integral action introduced may however be varied as a direct result of a measurement of system performance. A P+I control structure is chosen since this, in general, is capable of providing improvements in system response in terms of steady-state error, rise-time, etc. Making the integral action variable on-line does allow 'optimisation' of the system response subject to the choice of the variational law. There are, clearly, a number of possible controller configurations which may provide the desired structure, equally, the choice of performance measure is open to the designer. For the purpose of the current discussion one possible configuration due to McLean and Gill is examined which is based on the application to an aircraft pitch rate autopilot.

Recall from Chapter 5 that (classically) for pitch rate control a P+I controller is introduced and the controller parameters chosen to provide acceptable performance. Gain-scheduling was also shown to be required in order to preserve the system response and stability as a result of varying flight conditions. The system would appear to be a prime candidate for VICTOR. Consider then a controller of the form

$$G_c(s) = \left\{ 1 + \frac{g}{s + (\sigma g - \mu)} \right\} \qquad - 7.3$$

where σ and μ are constants which allow some flexibility in the design and g is a gain whose value depends upon the, as yet undefined, performance measure. The choice of σ and

μ will, for a given g, affect the effective integral action time and g will affect both the action time and proportional gain. The gain, g, is normally constrained to lie within some specified region (e.g. $0 < g < 1.0$).

To define a performance measure it may be wished to consider arguments similar to those of the ZOC scheme. Principally, the size and rate of change of closed-loop error signal can give an indication of system performance. Large values of error are undesirable and may be reduced by increasing the system gain. Additionally, the rate of change of error signal should not be excessive since limited actuator rates are normally required. In McLean's example (76) a performance measure is chosen such that

$$g = E - N \qquad - 7.4$$

with

$$E = \max \{ \gamma |y_1'|, \ \delta |\dot{y}_1'| \} \qquad - 7.5$$

$$N = \frac{\beta \lambda_c}{\lambda_c + s} \quad | \dot{y}_1' | \qquad - 7.6$$

where γ, β, δ are constants which require to be chosen with regard to the likely system constraints and λ_c is the inverse actuator time constant. y_1' and $\dot{y}_1'$ are the error and error derivative signals respectively. The latter signal is derived by applying the error signal, y_1, to a pseudo differentiator of the form :

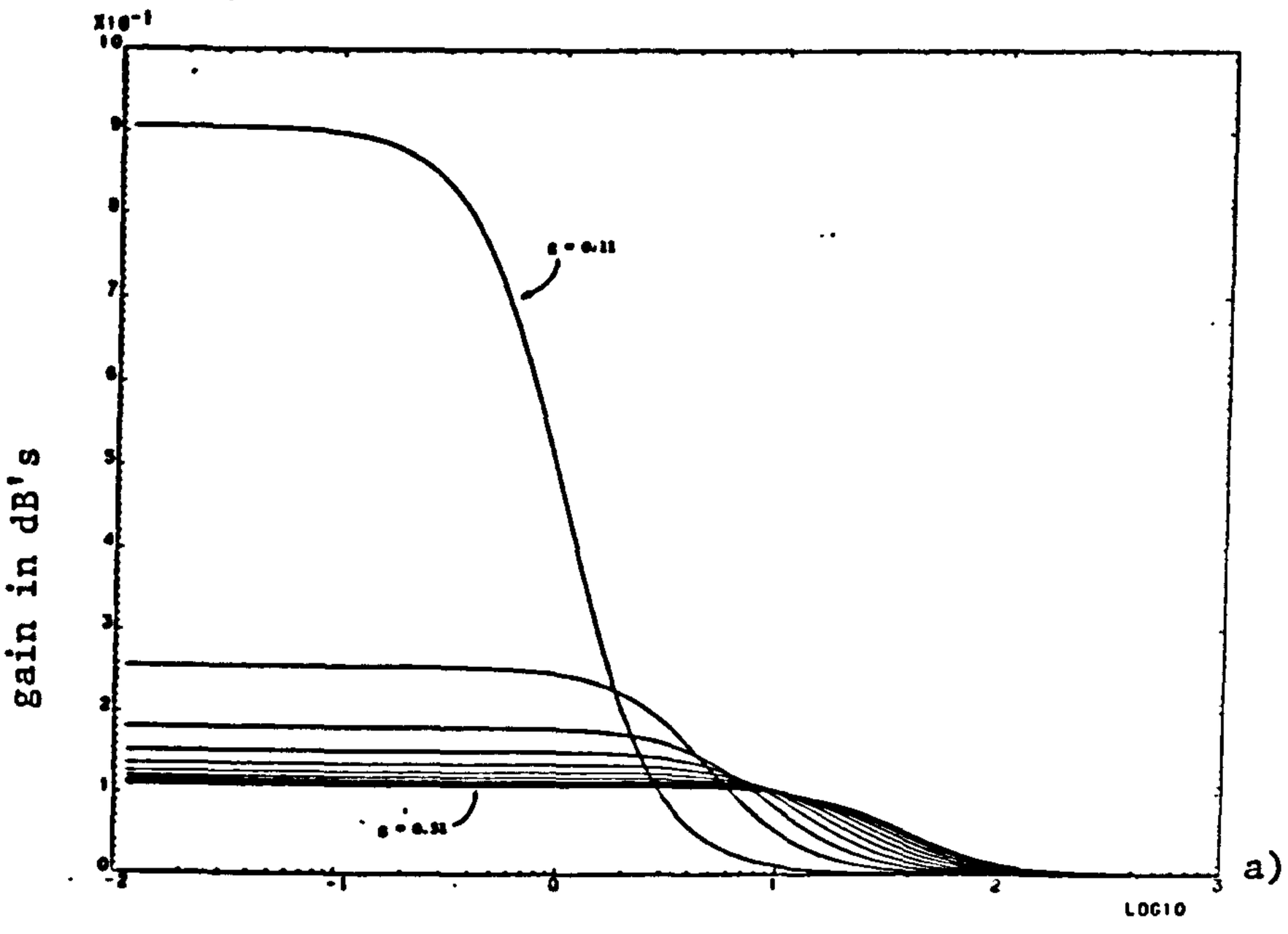
$$G_a(s) = \frac{s}{1 + s\tau} \qquad - 7.7$$

and y_1 is arranged to pass through a first order filter of time constant τ to preserve the magnitude and phase relationships.

The controller can be considered to have the form of a generalised lead-lag element since rearranging equation 7.3 gives

$$G_c(s) = \left\{ \frac{s + (\sigma g - \mu) + g}{s + (\sigma g - \mu)} \right\} \qquad - 7.8$$

Fig 7.5 VICTOR Bode Plots for Varying g (σ = 100, μ = 10)



Frequency, rad s^{-1}

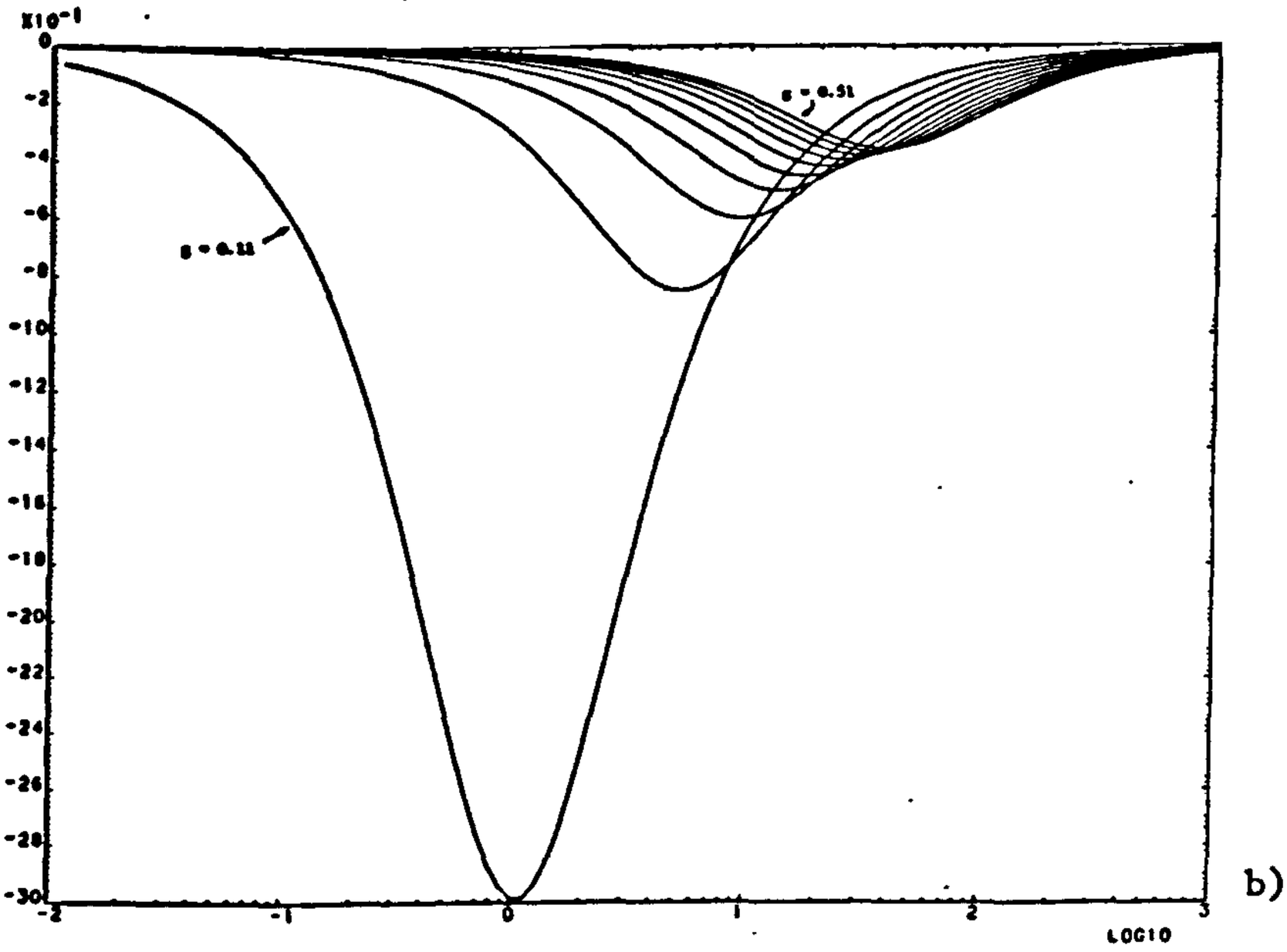
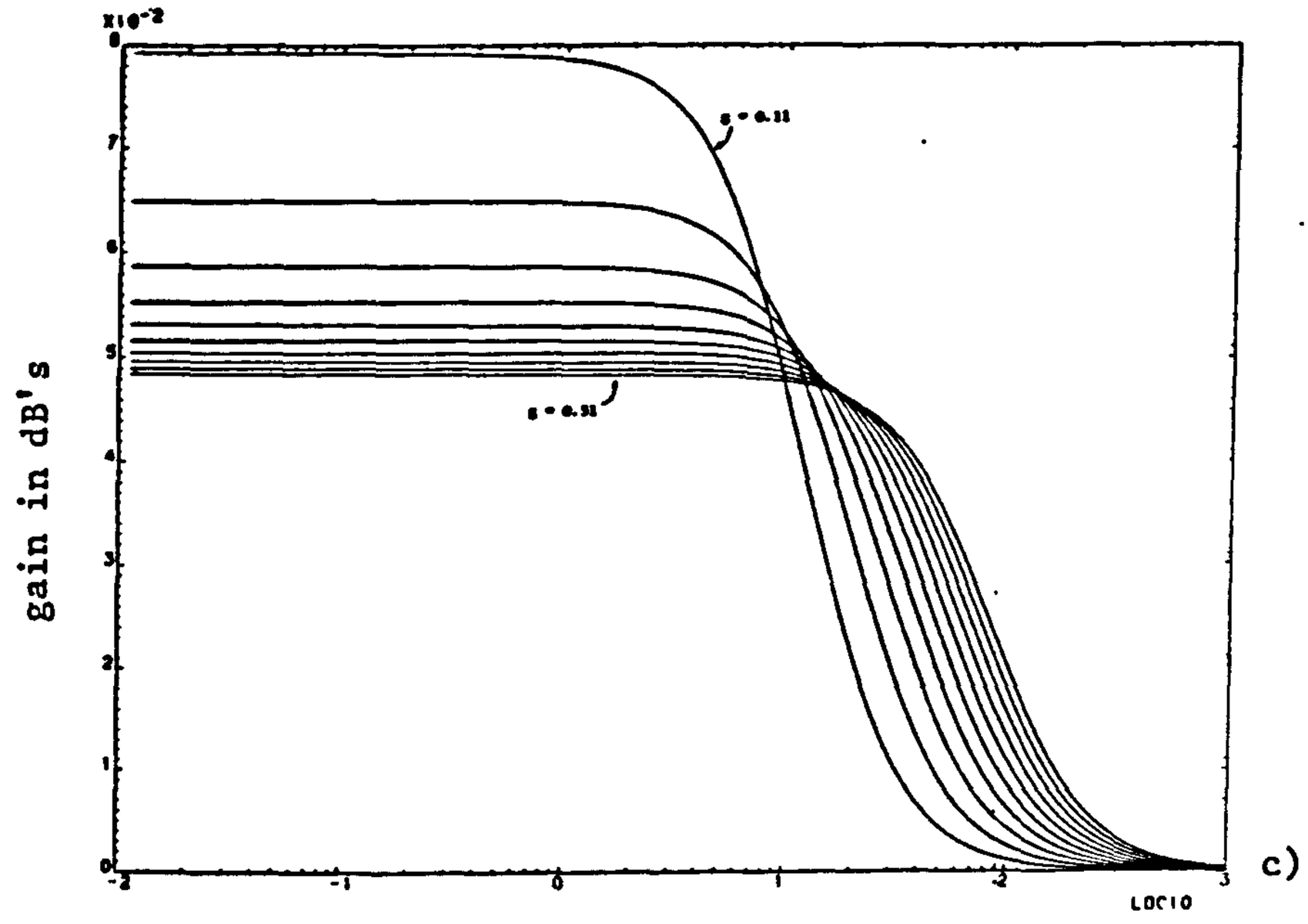


Fig 7.5 cont. (σ = 200, μ = 10)



The relative pole and zero position may be changed by changing g whilst the absolute pole/zero position is determined by σ, μ and g. Note that to guarantee that the pole has a negative real part we must have that

$$(\sigma g - \mu) \geqslant 0$$

$$\text{i.e.} \qquad \sigma \geqslant \mu/g \qquad - 7.9$$

A representative family of curves of amplitude and phase vs. frequency for equation 7.8 are shown in Figs 7.5 a) to d). Figs. 7.5 a) and b) have $\sigma = 100, \mu = 10$ and g varies from 0.11 to 0.51. Figs. 7.5 c) and d) are with $\sigma = 200, \mu = 10$ and the same range in g. Note from these that the low frequency gain is close to unity but decreases as g increases. The phase plots show that the maximum phase lag is introduced at low values of g and this lag reduces and the peak moves towards higher frequencies for increasing g. It might be concluded from this that the controller tends towards an ideal P+I structure for low g whilst higher g's produce almost unity transmittance.

The performance criteria of equations 7.4, 7.5 and 7.6 imply that g decreases as a result of large $\dot{y}_1$ values and increases due to large y_1 values, depending upon the choice of γ, β and δ. The implication here is that excessive $\dot{y}_1$ demands introduce more integral action whilst excessive y_1 values introduce pure gain. This would seem to be intuitively reasonable since excessive control rates will be reduced by increased integral action, excessive error deviations being reduced by increased loop gain.

From the above discussion it should be clear that this essentially heuristic technique requires the designer to select a large number of controller parameters. These must all be chosen with regard to the specific system application and the likely values of actuator rates, etc. Additionally, the stability of this technique is difficult to assess. In Gill's example it is necessary to include both the ZOC and VICTOR schemes within the design of a particular pitch rate SAS system. McLean (76) goes on to demonstrate that the system then can be shown to be globally stable. The parameter sensitivity of this type of controller structure

is also questionable and scheduling of some of the controller parameters, principally σ and μ , may be necessary as the flight conditions vary.

7.3.3 Summary

Whilst based on intuitive arguments it is felt that these systems, ZOC and VICTOR, cannot provide a general solution to the problems of parameter insensitive control. In the next sections the so-called sliding mode controllers are investigated with a view to assessing their performance in the aircraft environment.

7.4 Sliding Mode Theory

The theory of sliding mode behaviour in linear dynamical systems is a natural extension of the intuitively appealing concept of variable structure systems. In such systems the aim is to provide desirable closed-loop performance by changing the controller structure in response to a set of switching criteria based on measures of the system states. This idea is by no means new and typical of such schemes would be relay switched systems using 'bang-bang' control, the switching here being controlled by, for example, the sign of the closed-loop error signal. These switching schemes may combine desirable features from each of the system structures in addition to possessing properties which neither structure alone can exhibit. Typically, for example, an asymptotically stable system can be formed from two non-asymptotically stable structures. (The ZOC scheme described above can be considered as a switched mode type of controller, VICTOR is more akin to adaptive control). Characteristic of these switching systems is the fact that the overall system trajectory in the state space is composed of the trajectories of each structure, the trajectory changing at a switching surface. In this way the overall system trajectory can be made to converge on an equilibrium point, notionally the origin in the state-space, and hence achieve asymptotic stability. An

extension of this theory is to drive the system in such a way as to force the states to follow a trajectory in the state-space which is not inherently part of any of the switched structures.

This is the so-called sliding mode behaviour since, on reaching a defined switching surface, the state trajectory will 'slide' along this surface towards the equilibria of the system. An important aspect of such behaviour is that the motion in the state-space is invariant once the sliding surface is reached. This fundamental observation is important since it implies that the system response when sliding becomes invariant to changes in the underlying system. Hence, time varying parameters in the system will not affect the system's response. Much work has been undertaken in the general areas of variable structure systems and the sliding mode over the past 40 years. The sliding mode technique itself seems to have gained acceptance in control system design as a result of the work of Utkin (70,71,116) and Itkis (67) and the paper of reference (70) gives a good introduction to the topic. In recent years a significant body of research has led to the development of a generalised theory and design technique for sliding mode controllers (62,66,68,69,77-82,120). Additionally, these techniques have been applied to a number of problems, mostly in robotics (62-65).

The parameter insensitivity of the sliding mode regime would appear to be very desirable in the aircraft problem. It has already been shown that with many of the linear control designs (Chapters 5 and 6) applied to aircraft flight control it is difficult to provide parameter insensitive control without resorting to gain-scheduling techniques, for example. The possible benefits of employing non-linear sliding mode control should therefore be investigated. The ZOC and VICTOR techniques have already been discussed and in the following section the sliding mode design techniques will be investigated further.

7.4.1 Introduction

The design of a particular sliding mode controller may be considered in three stages. Firstly a desired sliding surface in state space must be chosen with due regard to the required system performance. Secondly, we must ensure that there exist sliding modes at every point along the surface by choice of the discontinuous control law, this is often termed the existence problem. Thirdly, we must ensure that the surface is reached along any trajectory of the non-sliding system, this being termed the reachability problem.

To illustrate the essentials of the design process consider the linear time-invariant system defined by :

$$\dot{\underline{x}} = A\,\underline{x} + B\,\underline{u} \qquad \underline{x} \in R^n \;;\; \underline{u} \in R^m$$
$$\underline{y} = C\,\underline{x} \qquad \underline{y} \in R^m \qquad - 7.10$$

The objective is to provide a feedback control law of the form

$$\underline{u} = - K^T\,\underline{x} \qquad - 7.11$$

$$\text{with} \qquad K^T = (\underline{k}^T + \Delta\,\underline{k}^T)$$

i.e. the control matrix K^T, is the sum of the fixed and switched matrices $\underline{k}^T$ and $\Delta\underline{k}^T$. The switching is governed by a switching function defined by

$$S = H\,\underline{x} \qquad S \in R^m \qquad - 7.12$$

and the condition S=0 is maintained along the switching surface. To do this we must ensure that the state trajectories on either side of the surface are directed towards the surface. The resulting motion is therefore confined to an infinitesimally small region on either side of the surface and the state 'slides' towards the equilibrium at $\underline{x} = 0$. To establish this we must have

$$\lim_{s_i \to 0^+} s_i < 0 \qquad \text{and} \qquad \lim_{s_i \to 0^-} s_i > 0$$

$$i = 1,2,\ldots,m \qquad - 7.13$$

or eqiuvahently $$s_i \dot{s}_i \leqslant 0 \qquad - 7.13\ a$$

The switched part of the control is arranged such that switching occurs if S deviates from 0 and the sense is such that the ensuing motion is directed towards the switching surface, hence satisfying 7.13.

The behaviour of the system in the sliding mode may be found by noting that during sliding

$$S = H\ \underline{x} = 0 \text{ and } \dot{S} = H\ \dot{\underline{x}} = 0 \qquad - 7.14$$

substituting for $\underline{x}$ from 7.10 gives

$$\dot{S} = 0 = H(A\ \underline{x} + B\ \underline{u})$$

$$\text{i.e. } \underline{u} = \underline{u}_{eq} = -(H\ B)^{-1}\ H\ A\ \underline{x} \qquad - 7.15$$

where $\underline{u}_{eq}$ is defined as the equivalent control which governs the response during sliding. Substituting back into 7.10 then gives

$$\begin{aligned} \dot{\underline{x}} &= A\ \underline{x} - B(H\ B)^{-1}\ H\ A\ \underline{x} \\ &= (A - B(H\ B)^{-1}\ H\ A)\ \underline{x} \\ &= A_{eq}\ \underline{x} \end{aligned} \qquad - 7.16$$

where now the A_{eq} matrix is the equivalent system matrix during sliding. Note that equation 7.12 expresses m of the states in terms of the remaining (n-m) and hence a reduction in system order is possible, i.e. from n to (n-m). Note also that r(HB)=m so that $(HB)^{-1}$ exists. When on the switching plane the control is not switched and hence it can be deduced from equations 7.11 and 7.15 that

$$K^T = (H\ B)^{-1}\ H\ A \qquad - 7.17$$

Consider the case when, without loss of generality, the switching equation is chosen to be the output equation of the system, equation 7.10. Hence H = C and the equivalent control becomes

$$\underline{u}_{eq} = -\ (C\ B)^{-1}\ C\ A\ \underline{x} \qquad - 7.18$$

with $$A_{eq} = (I_n - B(C\ B)^{-1}\ C)A \qquad - 7.19$$

where r(CB) = m to ensure non-singular control. Now, during sliding the behaviour of the system states is governed by

$$\dot{\underline{x}} = A_{eq}\ \underline{x} \qquad - 7.20$$

with the equivalent control as in equation 7.18. The n eigenvalues, λ_i, and associated eigenvectors $\underline{e}_i^{o}$ of the A_{eq} matrix will determine the responses during sliding. Note, however, that only (n-m) of these will be not necessarily zero valued since, during sliding, m of the states can be expressed in terms of the remaining (n-m) via.

$$C\ \underline{x} = 0$$

We also know that

$$y = C\ \underline{x} = 0 \qquad - 7.21$$

Any of the eigenvectors of A_{eq} which satisfy

$$C\ \underline{e}_i^{o} = 0 \qquad - 7.22$$

must lie within the null space of C and hence $\underline{e}_i^{o}$ must be a zero direction of the system S(A,B,C) and the associated eigenvalue must correspond to an invariant zero of S(A,B,C) (116). For the case of (CB) full rank, then, for a square system, there must be (n-m) system zeros. Hence (n-m) of the (not necessarily) zero eigenvalues of the A_{eq} must correspond to the invariant zeros of S(A,B,C) with associated zero directions as in equation 7.22. The zero eigenvectors lie within the null space of the output map C (117-119). This result was established recently by Zinober, Billings, et al (83,84) and is proposed as a method of evaluating system zeros. It can further be demonstrated that the action of the equivalent control, equation 7.18, is such as to assign some of the closed-loop pole positions to coincide with the system zeros and hence render the closed-loop system unobservable. This is easily shown by considering the standard observability criteria for the equivalent system, viz :

$$O(C,A_{eq}) = [C,\ C\ A_{eq},\ C\ A_{eq}^{2},.....,\ C\ A_{eq}^{n-1}]^{T}$$

Note, however, that from equation 7.19

$$C\ A_{eq} = C(I_n - B\ (C\ B)^{-1}\ C)A = 0$$

and hence by implication

$$C\ A_{eq}^{i} = 0 \qquad i=1,2,...,n-1$$

and thus

$$\text{rank}[O(C,A_{eq})] = \text{rank}\ (C) = m < n$$

It is worth comparing this control action with the eigenvalue/eigenvector placement techniques discussed in Chapter 6. Note here that it is again desired to provide specific closed-loop pole positions which, under high gain feedback, correspond to the system zero locations. The remaining poles then constitute the output response. A recent paper by Young, et al, (85) has demonstated the fundamental property that 'cheap' control, high gain feedback, etc. policies all tend to reduce the system order at high gains and may thus be analysed by singular perturbation theory. All of these high gain techniques possess desirable measures of robustness and disturbance rejection (86,87). With sliding mode control the objective is clearly similar in as much as the aim is to drive (n-m) eigenvalues to the invariant system zeros and hence the output response will be governed by the remaining m closed-loop eigenvalues.

Thus far the behaviour of the system when on the sliding surface has been considered via the so-called equivalent control method (67,71,115). The equation of the ideal sliding mode is obtained from a consideration of the time derivatives of a switching function, equations 7.13, 7.15 and 7.16. The existence problem can be considered as the ability to maintain the system arbitrarily close to the switching surfaces. This is achieved by ensuring that the direction of the velocity vector is towards the sliding surfaces, equation 7.13, and conditions to guarantee this

can be obtained via stability theory (67-69). Utkin (70) demonstrates that the existence problem can be solved given a certain set of conditions on the (m x m) matrix HB, derived via Lyapunov theory. Recall, however, that the H matrix determines the characteristics of the sliding mode and cannot, in general, be chosen arbitrarily to ensure existence. These two problems, i.e. existence and desirable sliding mode characteristics, can be decoupled through the results of Utkin (70). This decomposition of the problem relies on the following theorem due to Utkin :

Theorem 7.1 :- The equation of the sliding mode is invariant with respect to non-singular transformations :

$$S^* = H_s(x,t)\underline{s} \quad ; \; \underline{u}^* = H_u(x,t)\underline{u}$$

$$\det\{H_s\} \neq 0 \quad ; \det\{H_u\} \neq 0 \qquad - 7.23$$

The implication here is that the sliding mode is governed by equation 7.16 if the components of the control vector undergo discontinuities on new surfaces $s_i^* = 0$ or the components of the new control vector, u^*, undergo discontinuities on the chosen sliding surfaces $s_i = 0$. Also the equation of motion projected onto subspaces

$$(s_1^*, \; s_2^*, \; . \; . \; ., \; s_m^*)$$

in the first case and on

$$(s_1, \; s_2, \; . \; . \; ., \; s_m)$$

in the second depend upon H_s, H_u respectively.

In general therfore, for any switching function defined by the H matrix of equation 7.14, H_s or H_u may be chosen such that the matrix (H'B) is diagonal (or diagonally dominant) whence the problem may be decomposed to m scalar cases. Existence conditions of equation 7.13 may then be applied to each of the m switching hyperplanes.

Allied to the existence problem is the reachability problem, the two being very similar in that the first refers to asymptotic stability in the small, the second to asymptotic stability in the large. Utkin (70) has shown that for multivariable time-invariant VSS control with

systems of the form of equations 7.10, then the control is a piecewise linear function of some, k, of the states, namely

$$\underline{u} = - \Psi \underline{x}^k - \delta$$

$$(\underline{x}^k)^T = (x_1, x_2, \ldots, x_n)$$

$$\Psi_{ij} = \begin{cases} \alpha_{ij} & \text{if } s_i x_j > 0 \ , \ i=1,2,\ldots m \\ \beta_{ij} & \text{if } s_i x_j < 0 \ , \ j=1,2,\ldots k \end{cases}$$

α_{ij}, β_{ij} are constants

and $\delta^T = (\delta_1, \delta_2, \ldots, \delta_m) \ , \quad \delta_i = \delta_{oi} \text{ sign}(s_i)$ - 7.24

with δ_{oi}, small positive scalars. For the existence of m sliding hyperplanes with asymptotically stable sliding modes it is a necessary and sufficient condition that the system of equation 7.10 and 7.24, with $\Psi_{ij} = \Omega_{ij}$, where Ω_{ij} has some constant value between α_{ij} and β_{ij} , has (n-m) eigenvalues with negative real parts.

The implication here is that the system with partial state feedback retains (n-m) eigenvalues with negative real parts over a bounded range of feedback gains.

The reachability of the sliding hyperplanes can also be guaranteed by suitable choice of the control parameters of equation 7.24 subject to conditions depending on the H_s and H_u matrices (71). The design process then consists of chosing a linear control law of the form $u = K \underline{x}^k$ which places (n-m) of the eigenvalues at desired locations and then choosing the parameters of equation 7.24 to ensure reachability.

In general the invariancy of the sliding mode depends not only on the switching matrix, H, but also upon the system and disturbances and the choice of H can be optimised to provide maximum insensitivity to the likely disturbances and system parameter variations. For a class of system governed by equations of the form

$$\dot{\underline{x}} = A \underline{x} + \underline{h}(\underline{x},t) + B \underline{u} \qquad - 7.25$$

where $\underline{h}(\underline{x},t) \in R^n$ is a function of the states, disturbances and time varying parameters, it has been shown (72) that the sliding mode, $S = 0$, is insensitive to $\underline{h}(\underline{x},t)$ if

$$\text{rank } (B,\underline{h}) = \text{rank } (B) \qquad - 7.26$$

Note that systems in canonical form satisfy the above relationship.

In the general multivariable VSS problem it is therefore clear that the design process is quite complex and is best undertaken by computer aided design packages. The design may, to some extent, be iterative in that desirable invariancy may require an iterative choice of H. The results of a multivariable VSS controller applied to the lateral motion model of the unmanned aircraft previously discussed is investigated in section 7.6. Initially, it is instructive to investigate some simplistic SISO systems and the performance of VSS sliding control applied to these.

7.5 SISO VSS

Consider the third order system defined by the following state equation

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [20 \quad 9 \quad 1]\, \underline{x} \qquad - 7.27$$

which is in phase canonical form. It is required to produce a VSS design for this system. The system has eigenvalues of -1., -2. and -3. and two real invariant zeros at -4. and -5. (84). By choosing the output equation to be the switching function, i.e.

$$H = [20 \quad 9 \quad 1] \qquad - 7.28$$

we note that r(HB)=1 (i.e. full rank).

Evaluating the A_{eq} matrix for this system from equation

7.19 we have that

$$A_{eq} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -20 & -9 \end{bmatrix} \qquad - 7.29$$

also the equivalent control, u_{eq}, becomes :

$$u_{eq} = [-6 \quad 9 \quad 3] \qquad - 7.30$$

The eigenvalues of the A_{eq} matrix are 0., -4. and -5. with corresponding eigenvectors of

$$\underline{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \underline{e}_2 = \begin{bmatrix} 0.061 \\ -0.242 \\ 0.968 \end{bmatrix} \qquad \underline{e}_3 = \begin{bmatrix} 0.039 \\ -0.196 \\ 0.979 \end{bmatrix}$$

Note that, $C\underline{e}_1 \neq 0$, $C\underline{e}_2 = 0$, $C\underline{e}_3 = 0$ and hence the zeros of the system are at -4. and -5. and these have corresponding eigenvectors lying in the null space of C. A choice of linear gain vector of

$$K = [-6 \quad 9 \quad 3] \qquad - 7.31$$

will therefore provide the desired sliding mode with the null space response determined by the second order system given by :

$$\dot{\underline{x}} = A_{eq}\, \underline{x}$$

$$\text{with} \qquad S = C\, \underline{x} = 0$$

For SISO systems in phase variable controllable canonical form the existence condition will be fulfilled for an arbitrary choice of C provided that r(CB) = 1. The reachability conditions do, however, have to be considered with some care. Consider control of the form of equation 7.11 with :

$$u = -(\underline{k}^T + \Delta \underline{k}^T)\, \underline{x}$$

then a choice of $\underline{k}^T$ as 7.31 guarantees the sliding mode by placing (n-m) = 2 of the closed-loop sytem eigenvalues at the invariant sytem zeros and hence in the null space of C. The remaining eigenvalue, λ_n, governs the range space response and can be selected to guarantee reachability by suitable choice of $\Delta \underline{k}^T$, the switched gain vector. White

(88,89) has considered reachability in systems of this type and shown that for the case of full state switching, i.e.

$$\Delta k_i \neq 0 \qquad i=1,2,\ldots,n$$

then for reachability

$$\begin{aligned} \Delta k_i &> \lambda_n c_i \quad \text{for } s\, x_i > 0 \\ \Delta k_i &< \lambda_n c_i \quad \text{for } s\, x_i < 0 \end{aligned} \qquad i=1,2,..,n \qquad - 7.32$$

for the case of $\lambda_n \geqslant 0$, where λ_n is the range space eigenvalue. Note that in this example $\lambda_n = 0$. A choice of

$$\begin{aligned} \Delta k_i &> 0 \quad \text{for } s\, x_i > 0 \\ \Delta k_i &< 0 \quad \text{for } s\, x_i < 0 \end{aligned} \qquad i=1,2,\ldots n \qquad - 7.33$$

will therefore guarantee that the sliding mode is reachable.

The state trajectories for the above system are shown in Figs. 7.6 a) to c) for an initial condition of 1.0 in x_1, the Δk's being chosen as 10.0, in accordance with equation 7.33. The sliding mode is reached after approximately 0.7 secs. and the x_1 state then displays the dynamics of the null space since, for this example, in sliding :

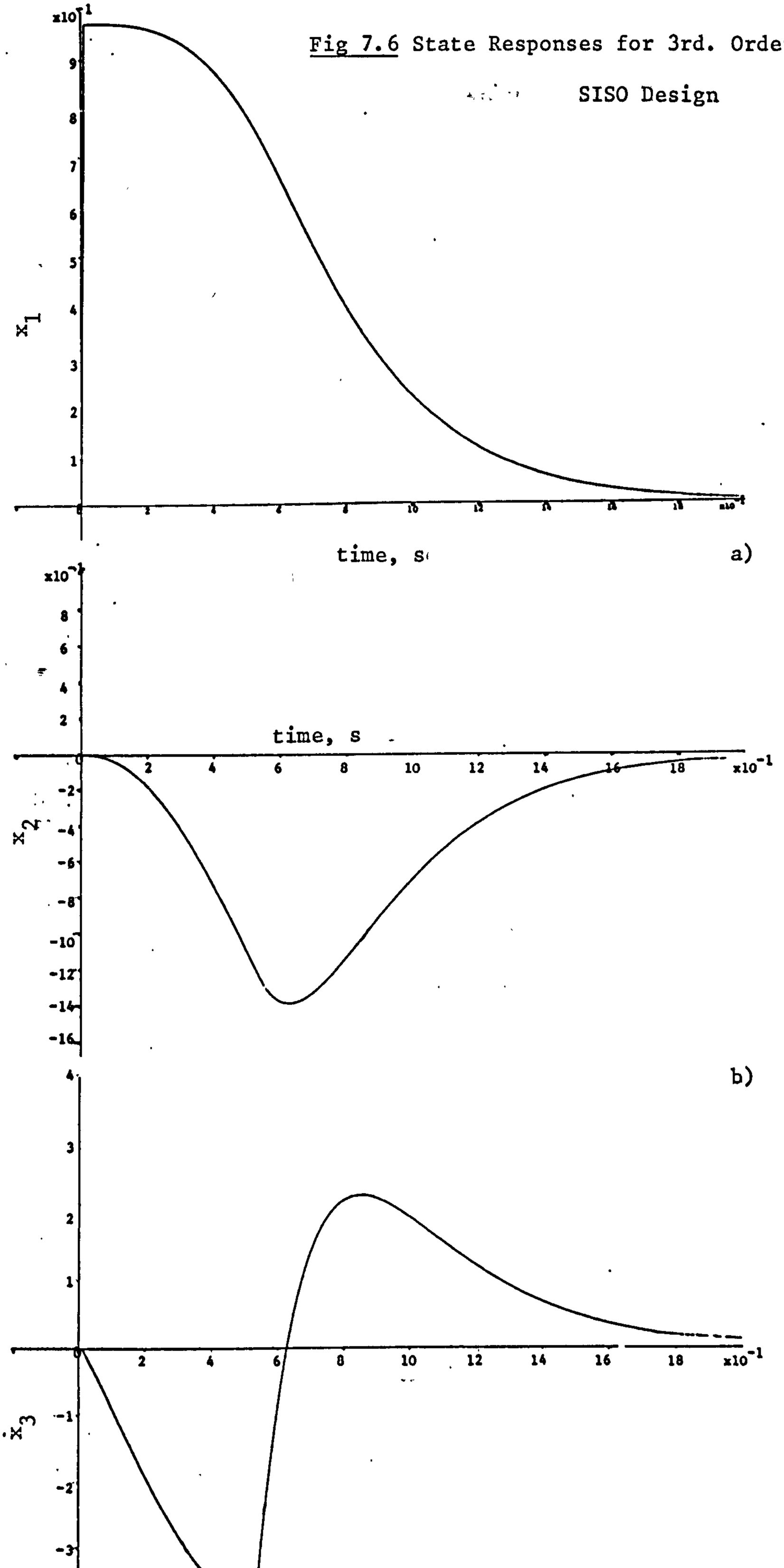
$$C\,\underline{x} = 20\,x_1 + 9\,x_2 + x_3 = 0$$

$$\text{i.e.} \qquad \ddot{x}_1 + 9\,\dot{x}_1 + 20\,x_1 = 0$$

7.6 Multivariable VSS

The simplistic ideas of the previous section for SISO systems may be extended to the general multivariable case by using the results of section 7.4.1 and more recent work by Ryan et al and Zinober et al (61,69,77,79). This latter work has led to the development of a CAD package (78,80) which has been used to provide VSS control designs for the examples which follow. In the following attention is turned to the lateral motion model of the Machan aircraft used previously and an investigation of the performance and robustness aspects of VSS applied to this example is made.

Fig 7.6 State Responses for 3rd. Order VSS SISO Design



Consider then the lateral motion model of the aircraft as given in equations 6.23 (section 6.5.1) with the inclusion of two first order actuators. A VSS sliding mode control design for this two input six state system is required. The design technique initially generates a suitable hyperplane switching matrix, H, as in equation 7.12. The choice of this matrix determines the system behaviour during sliding and the sliding mode eigenvalues and eigenvectors are given by the eigenstructure of the A_{eq} matrix of equation 7.16 as discussed previously. For the purposes of the design a similarity transform is initially introduced such that the system of equations 7.10 becomes

$$\left.\begin{aligned} \dot{\underline{z}} &= T\,A\,T^{-1}\,\underline{z} + T\,B\,\underline{u} \\ \underline{y} &= C\,T^{-1}\,\underline{z} \end{aligned}\right\} - 7.34$$

where $T\,B = \begin{bmatrix} 0 \\ B_2 \end{bmatrix}$; $T\,A\,T^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$;

$$C\,T^{-1} = [C_{11} \mid C_{12}]$$

thus

$$\begin{bmatrix} \dot{\underline{z}}_1 \\ \dot{\underline{z}}_2 \end{bmatrix} = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \end{bmatrix} \underline{u} \qquad - 7.35$$

$$\underline{y} = [C_{11} \mid C_{12}] \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \end{bmatrix} \qquad - 7.36$$

where $\underline{z}_1$ and $\underline{z}_2$ are ((n-m) x 1) and (m x 1) partitioned vectors of the $\underline{z}$ state vector. In this form the equivalent control, $\underline{u}_{eq}$ is given by equation 7.15, assuming that the switching matrix, H, is given by the output matrix, (CT^{-1}), as :

$$\underline{u}_{eq} = -\,(C_{12}B_2)^{-1}[(C_{11}A_{11} + C_{12}A_{21})\underline{z}_1 + (C_{11}A_{12} + C_{12}A_{22})\underline{z}_2]$$

The system behaviour during sliding is now determined by the first set of equations in 7.35 i.e.

$$\dot{\underline{z}}_1 = A_{11}\,\underline{z}_1 + A_{12}\,\underline{z}_2 \qquad - 7.37$$

and recall that during sliding

$$\dot{Y} = 0 = [C_{11} \quad C_{12}] \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \end{bmatrix} = C_{11}\, \underline{z}_1 + C_{12}\, \underline{z}_2 = 0 \qquad - 7.38$$

Substituting for $\underline{z}_2$ in 7.37 from 7.38 then gives the equivalent sliding mode sub-system as

$$\dot{\underline{z}}_1 = (A_{11} - A_{12}\, C_{12}^{-1}\, C_{11})\, \underline{z}_1 \qquad - 7.39$$

The results of Utkin and Young may then be used, principally, if C_{12} is chosen as I_m, the (m x m) unit matrix, then the eigenstructure of the ((n-m) x (n-m)) matrix

$$(A_{11} - A_{12}\, C_{11})$$

determines the system performance during sliding, provided that the (A_{11},A_{12}) pair is controllable. It can further be shown (63) that (A_{11},A_{12}) is controllable if (A,B) is controllable. Hence, the sliding modes can be chosen arbitrarily by suitable choice of C_{11}. Note also that with $C_{12} = I_m$ then

$$C = [C_{11} \quad I_m]\, T$$

hence $$s_i(x) = c_i\, x = 0 \qquad i=1,2,\ldots,m$$

The performance of the system away from the sliding mode is governed by the remaining sub-system of equation 7.35. Reachability can be guaranteed by ensuring that this sub-system has trajectories in the state-space which cross the switching hyperplane at some point. By choosing this m th. order sub-system to have eigenvalues with negative real parts, reachability is guaranteed.

Proceeding in this manner for this example, the (n-m)=4 eigenvalues and eigenvectors for the sliding part of the controller must initially be assigned via equation 7.39. This may be achieved by employing an eigenvector assignment technique for the equivalent system similar to that used in section 6.5.1. Using the same arguments as in section 6.5.1 with the same eigenvalues/eigenvectors an equivalent system having eigenvalues of :

$$-4.0,\ -0.05,\ \ -0.63 \pm\ \ 2.42j$$

is obtained with corresponding normalised eigenvectors :

$$\underline{e}_1 = \begin{bmatrix} 0.0068 \\ 0.9552 \\ -0.0666 \\ -0.2388 \\ -0.0233 \\ -0.1597 \end{bmatrix} \quad ; \quad \underline{e}_2 = \begin{bmatrix} 0.0000 \\ -0.0477 \\ 0.2877 \\ 0.9549 \\ -0.0179 \\ 0.0518 \end{bmatrix} \quad ; \quad \underline{e}_{3,4} = \begin{bmatrix} 0.997 \ \pm 0.0 \ j \\ -0.0151 \pm 0.0074j \\ 0.007 \ \pm 0.0711j \\ -0.0013 \pm 0.0066j \\ 0.0202 \pm 0.0017j \\ 0.0009 \pm 0.006 \ j \end{bmatrix}$$

Comparison of this eigenstructure with that obtained in section 6.5.1 shows that whilst the real eigenvectors are almost identical the complex pairs are somewhat different. This is largely due to the slight differences in the assignment algorithms used. Taking the above to be an acceptable design, however, gives a switching matrix, C, of

$$C = \begin{bmatrix} -0.0196 & 0.0297 & -0.0125 & 0.0276 & 1. & 0. \\ 0.0022 & 0.1561 & -0.1023 & -0.0156 & 0. & 1. \end{bmatrix}$$

On comparing this with equation 6.30 the similarity can be seen; the major difference being in element (1,3). Equation 6.30 would also be a, possibly better, candidate for the switching hyperplane.

The design then proceeds by selecting the remaining two range space eigenvalues. These should be made fast and stable since they will determine how quickly the system reaches the switching hyperplane. Remember, however, that there will be limitations on the physical actuator rates and amplitudes which restrict this choice. For this design a choice of -10.0 and -15.0 was made in order to remain within the limitations of the actuator bandwidths of 10 and 20 rad s^{-1}. The controller structure has been shown to be equivalent (69) to a summation of a linear state feedback and a scaled 'unit-vector' (non-linear) which is equivalent to the relay type structure used previously, equation 7.11 for example. The control thus becomes

$$\underline{u} = L\,\underline{x} + \frac{N\,\underline{x}}{||\ M\,\underline{x}\ ||} \qquad - 7.40$$

where $\underline{x}$ is the system state vector and the L, M and N matrices are derived as follows (69).

Consider a second transformation applied to the system of equations 7.35 and 7.36 such that :

$$\underline{w} = T_2\,\underline{z} \qquad - 7.41$$

with T_2 non-singular, orthogonal and specifically choose

$$T_2 = \begin{bmatrix} I_{n-m} & 0 \\ F & I_m \end{bmatrix} \quad ; \text{ where } F = C_{12}^{-1}\,C_{11} \qquad - 7.42$$

Partitioning $\underline{w}'=[\underline{w}_1'\ \ \underline{w}_2']$ with $\underline{w}_1 \in R^{n-m}$ and $\underline{w}_2 \in R^m$ it then follows that :

$$\underline{w}_1 = \underline{z}_1 \quad ; \quad \underline{w}_2 = F\,\underline{z}_1 + \underline{z}_2 \qquad - 7.43$$

The conditions on $\underline{w}_2$ corresponding to S=0, i.e. sliding, can be derived from equations 7.38 and 7.43 as

$$C_{11}\,\underline{w}_1 + C_{12}(\underline{w}_2 - F\,\underline{w}_1) = 0$$

$$\text{hence} \quad C_{12}\,\underline{w}_2 = 0$$

from which it follows that the conditions S=0 and $\underline{w}_2$=0 are equivalent in the sense that the points in the state space at which S=0 are precisely the points at which $\underline{w}_2$=0. Combining equations 7.35 and 7.36 with 7.43 then gives the transformed state equations as :

$$\dot{\underline{w}}_1 = \Sigma\,\underline{w}_1 + A_{12}\,\underline{w}_2 \qquad - 7.44$$

$$\dot{\underline{w}}_2 = \Theta\,\underline{w}_1 + \Psi\,\underline{w}_2 + B_2\,\underline{u} \qquad - 7.45$$

where $\Sigma = A_{11} - A_{12}\,F$; $\Theta = F\Sigma - A_{22}\,F + A_{21}$;

$$\Psi = F\,A_{12} + A_{22} \qquad - 7.46$$

To attain the sliding mode it is necessary to force $\dot{\underline{w}}_2$ and $\underline{w}_2$ to become identically zero. To do this the linear

part of the control may be chosen as :

$$u^L(w) = - B_2^{-1} \{ \Theta \underline{w}_1 + (\Psi - \Phi_*) \underline{w}_2 \} \qquad - 7.47$$

with Φ_* as any (mxm) matrix with left half-plane eigenvalues. In particular, given a spectrum

$$\{ \mu_i : Re(\mu_i) < 0; \ i=1,...,m \}$$

Φ_* may be chosen as $\Phi_* = diag\{ \mu_i \ ; \ i=1,...,m \}$. Transforming back into the original ($\underline{x}$) state space gives :

$$L = - B_2^{-1} [\Theta \quad \Psi - \Phi_*] T_2 \ T \qquad - 7.48$$

This linear control law, $\underline{u}^L$, serves only to drive the state component $\underline{w}_2$ to zero asymptotically; to attain the null space of C in finite time, the non-linear control component, $\underline{u}^N$, is required. This non-linear control must be discontinuous whenever $\underline{w}_2=0$ and zero elsewhere. Letting P_2 denote the positive definite unique solution of the Lyapunov equation :

$$P_2 \Phi_* + \Phi_*' P_2 + I_m = 0 \qquad - 7.49$$

then $P_2 \underline{w}_2=0$ iff $\underline{w}_2=0$ and the non-linear control component may be taken as :

$$\underline{u}^N = \frac{- \rho}{|| P_2 \underline{w}_2 ||} B_2^{-1} P_2 \underline{w}_2 \quad (\underline{w}_2 \neq 0) \qquad - 7.50$$

with ρ a scalar design parameter selected by the designer. When $\underline{w}_2=0$, $\underline{u}^N$ may be arbitrarily defined as any function satisfying $||\underline{u}^N|| \leqslant \rho$. Transforming back into the original ($\underline{x}$) state space then gives the N and M matrices of equation 7.40 as :

$$N = - B_2^{-1} [0 \quad P_2] T_2 \ T \qquad - 7.51$$

$$M = [0 \quad P_2] T_2 \ T \qquad - 7.52$$

The control law is thus formulated as a summation of $\underline{u}^L$ and $\underline{u}^N$ as in equation 7.40.

For this example the following are obtained :

$$L = \begin{bmatrix} 0.015 & -0.011 & -0.0202 & -0.0111 & -0.267 & 0.0426 \\ 0.0032 & -0.0215 & 0.0444 & 0.0315 & -0.0959 & -0.0529 \end{bmatrix}$$

$$M = \begin{bmatrix} 0.0001 & 0.0078 & -0.0051 & -0.0008 & 0. & 0.05 \\ -0.0007 & 0.001 & -0.0008 & 0.0009 & 0.0333 & 0. \end{bmatrix}$$

$$N = \begin{bmatrix} 0. & 0. & 0. & 0. & -0.0017 & 0. \\ 0. & -0.0008 & 0.0005 & 0.0001 & 0. & -0.005 \end{bmatrix}$$

In order to evaluate the performance of the VSS controller derived above the non-linear control law of equation 7.40 was initially implemented with a linear simulation of the lateral motion model of the aircraft as in equations 6.23.

The state responses for initial conditions of 0.1 ms^{-1} in v and 0.5 rad s^{-1} in p are shown in Figs. 7.7 a) to f). From these the following can be noted:

i) The Dutch roll mode appears dominantly in the v and r responses. The mode has approximately the desired period of 2.3 s and damping of 0.25. The control action for this mode appears dominantly on the rudder, as desired.

ii) The spiral mode appears on the roll state, φ , and is of the desired time constant of approximately 20 s. The mode also appears to some extent in r. The aileron response shows that this surface is largely responsible for providing control of this mode.

iii) The roll subsidence mode is associated dominantly with the roll rate, p, and has the desired fast, first order, response with approximately 0.25 s time constant.

iv) The actuator demands are acceptable although very slight 'chatter' is evident due to the discontinuous nature of the control. The sliding regime is reached almost immediately, as desired.

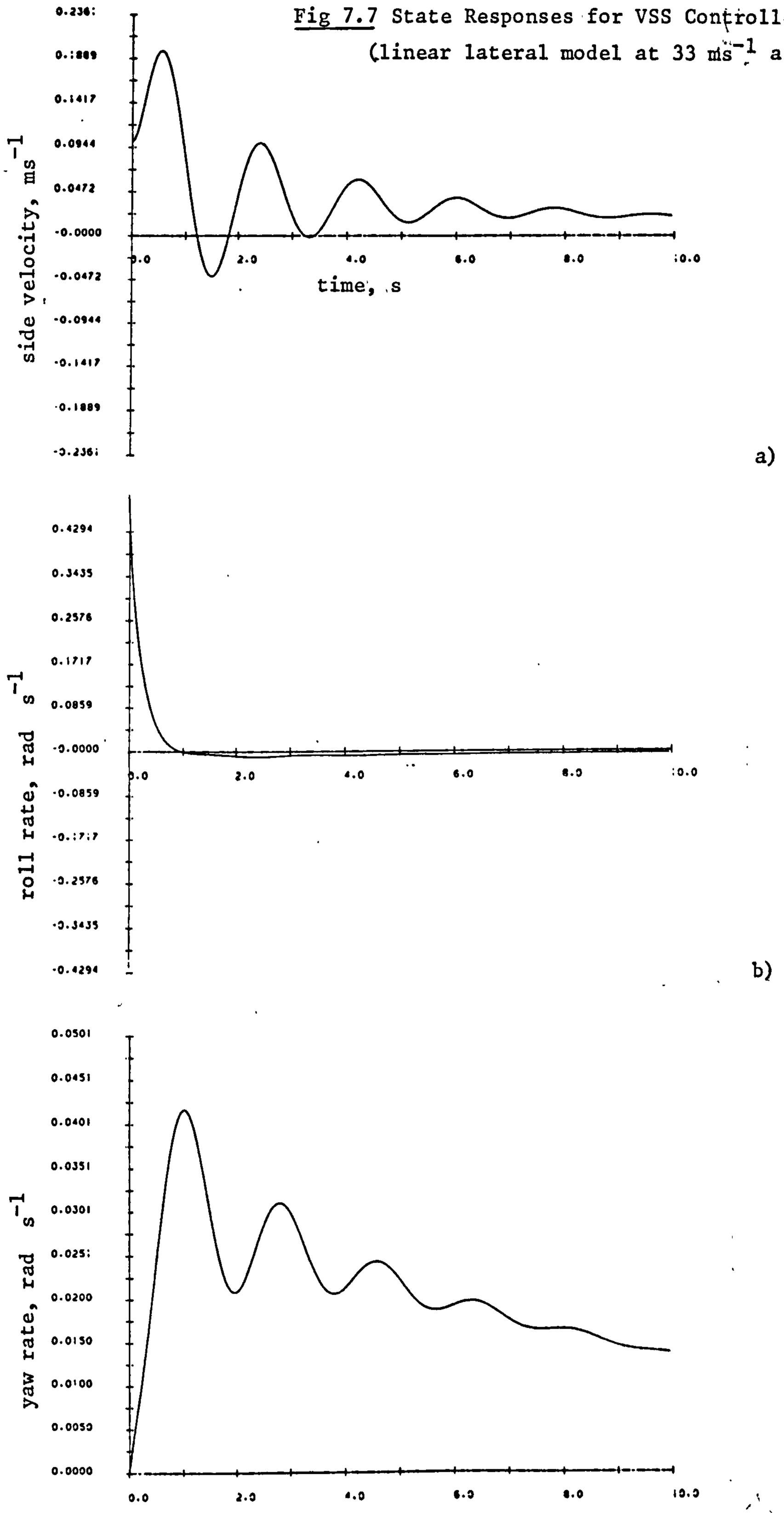
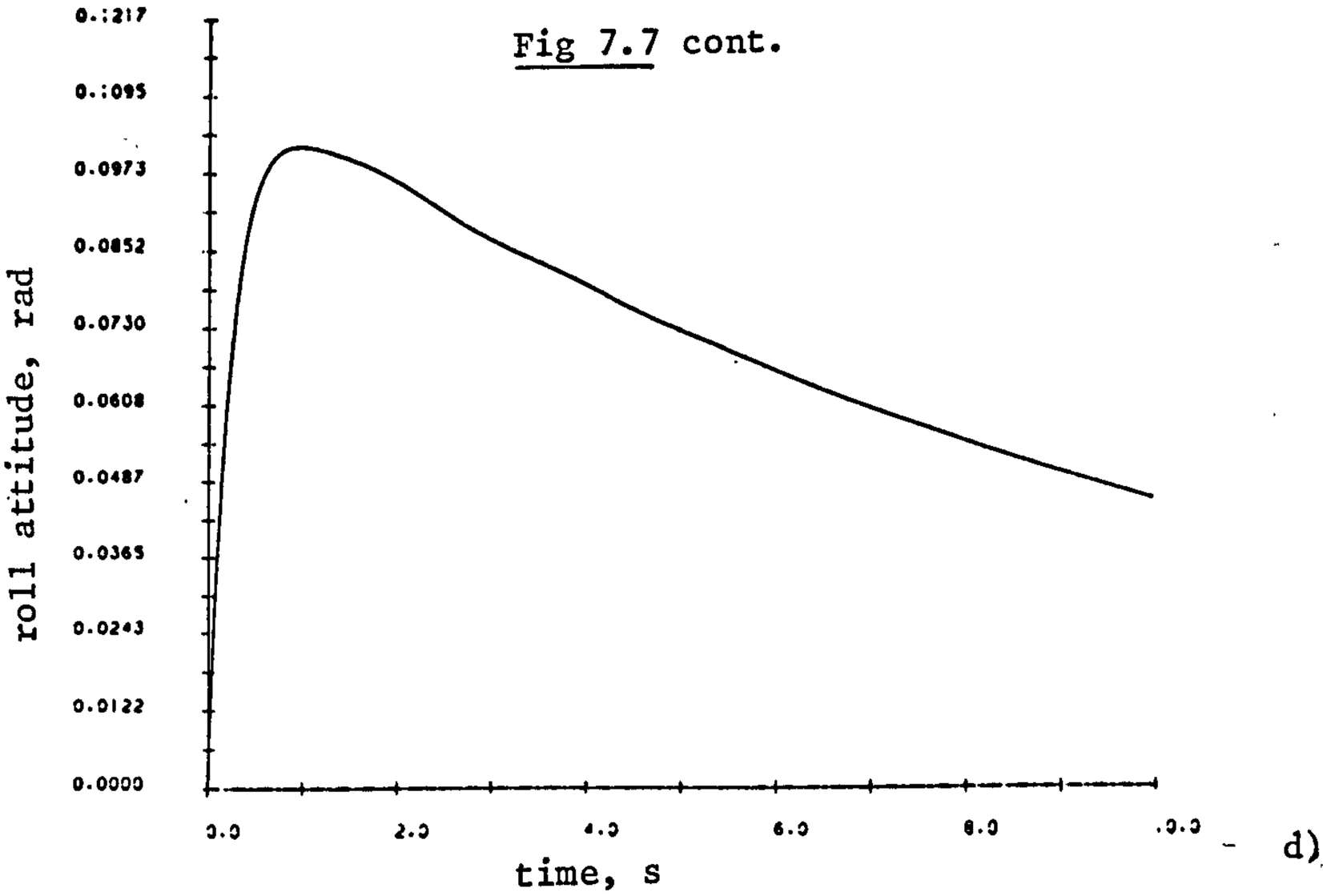
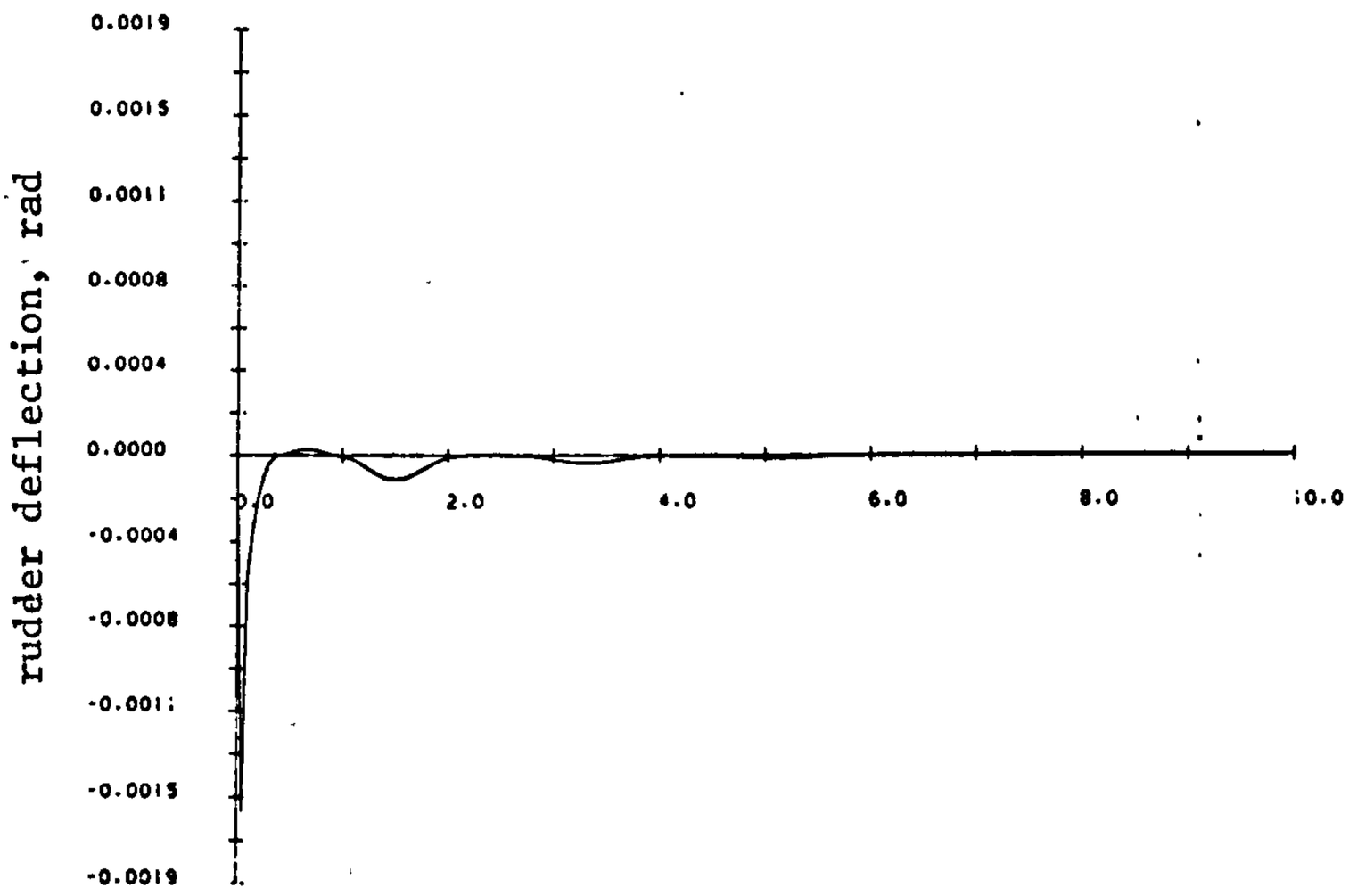


Fig 7.7 State Responses for VSS Controller
(linear lateral model at 33 ms^{-1} airspeed)

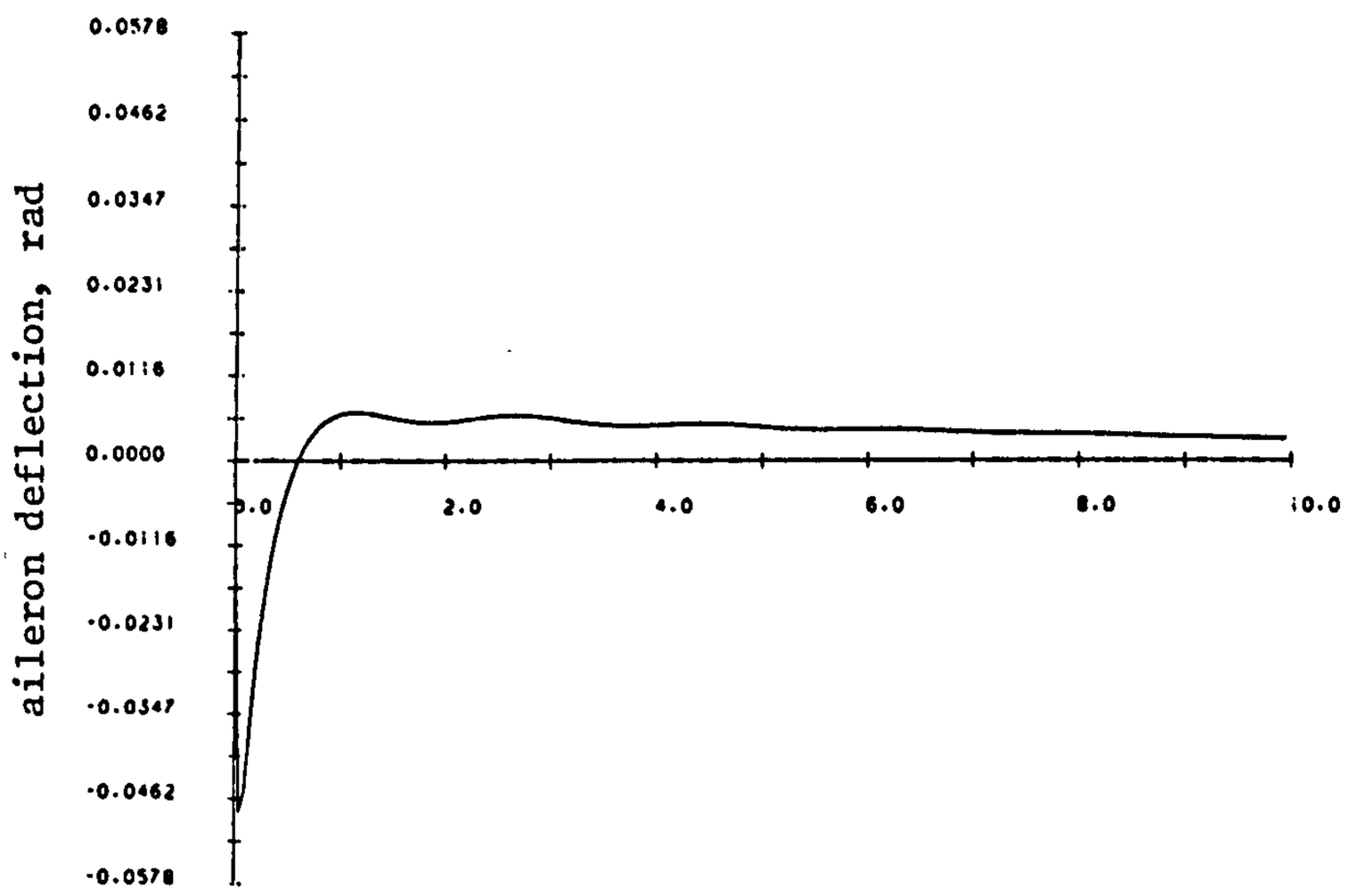
Fig 7.7 cont.



d)



e)



f)

The VSS controller clearly provides acceptable control when employed with this simple linear simulation and yields modal assignment and actuator decoupling similar to that obtained via the optimal control scheme of section 6.5.1. To assess the performance of the VSS controller when faced with a non-linear system it was implemented with the non-linear aircraft simulation developed in Chapter 2. Figs. 7.8 a) to h) show the resulting state responses of the lateral motion states with initial conditions of 0.1 ms^{-1} in v and 0.5 rad s^{-1} in p. Also shown, for comparison are the results of the linear simulation (dotted lines) subjected to the same initial conditions and drawn to the same time scale. A comparison of the VSS controller's performance with the linear and non-linear system models reveals that the responses are broadly similar, the major differences being in the amplitude and period of the Dutch roll mode (v and r states). Overall, however, the responses are acceptable and provide adequate control for both the linear and non-linear systems. Note that the actuator demands (Figs. 7.8 g) and h)) show more evidence of 'chatter' in the non-linear simulation.

To provide a feel for the tolerance of the VSS controller to changes in system parameters (i.e. how parameter insensitive it is) the design corresponding to the nominal 33 ms^{-1} airspeed was applied to the non-linear simulation at 45 ms^{-1} airspeed. Note that this will radically alter the 'A' matrix for the system (a discussion of this is included in Chapter 8) and hence some variation in system performance is to be expected.

Figs. 7.9 a) to h) show the resulting system responses for both the 33 ms^{-1} and 45 ms^{-1} airspeed cases (45 ms^{-1} shown dotted). The controller clearly does not provide identical performance for these two cases, the differences principally being a slight increase in Dutch roll frequency and amplitude, damping remaining approximately constant, and a decrease in the spiral mode time constant, as evidenced by the roll angle, φ, response. Note however, that the actuator motions remain very similar. One of the main consequences of increasing the aircraft's airspeed is to

Fig 7.8 State Responses for VSS Controller
(lateral model at 33 ms^{-1} airspeed,
linear vs. non-linear)

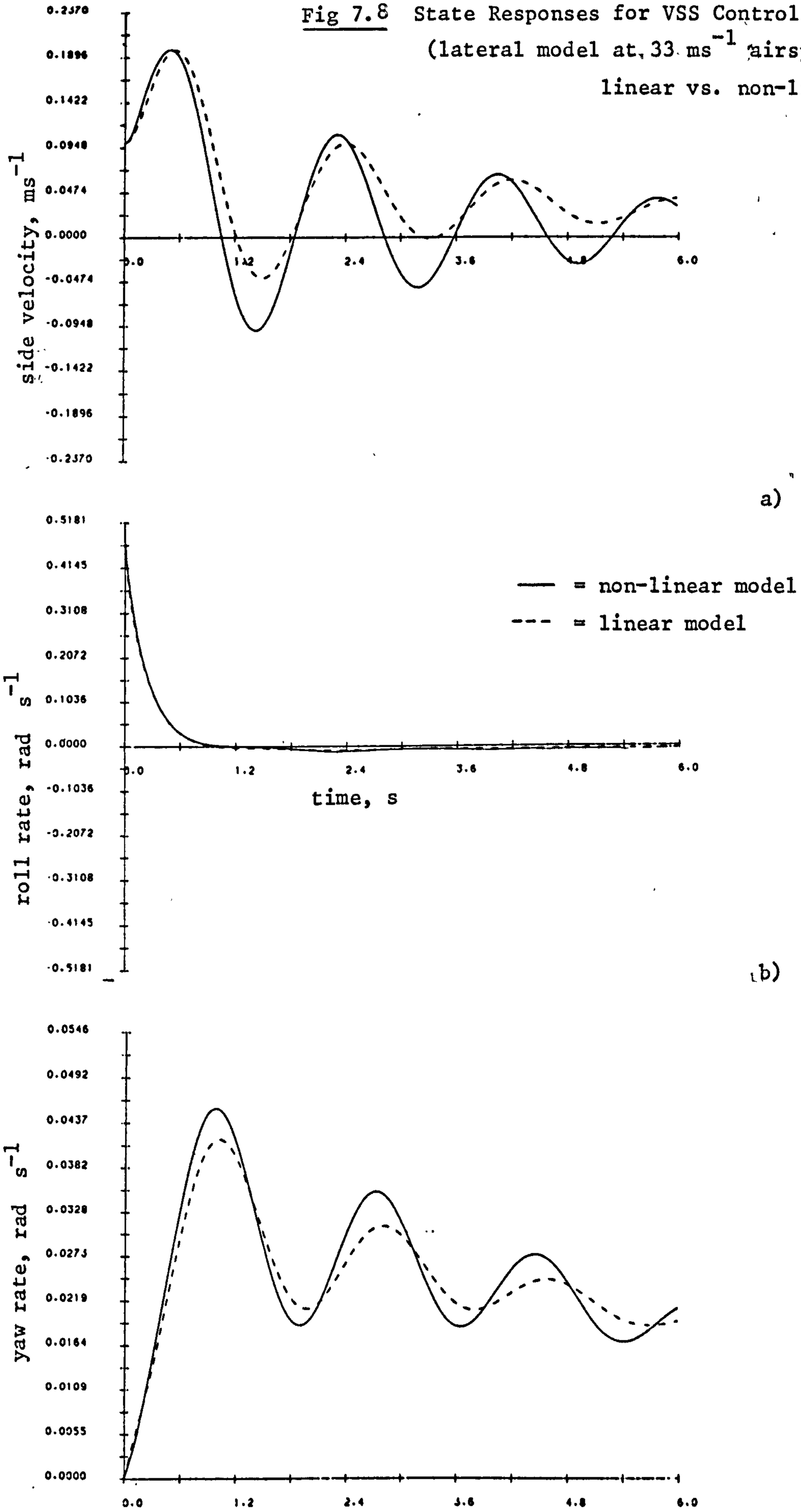
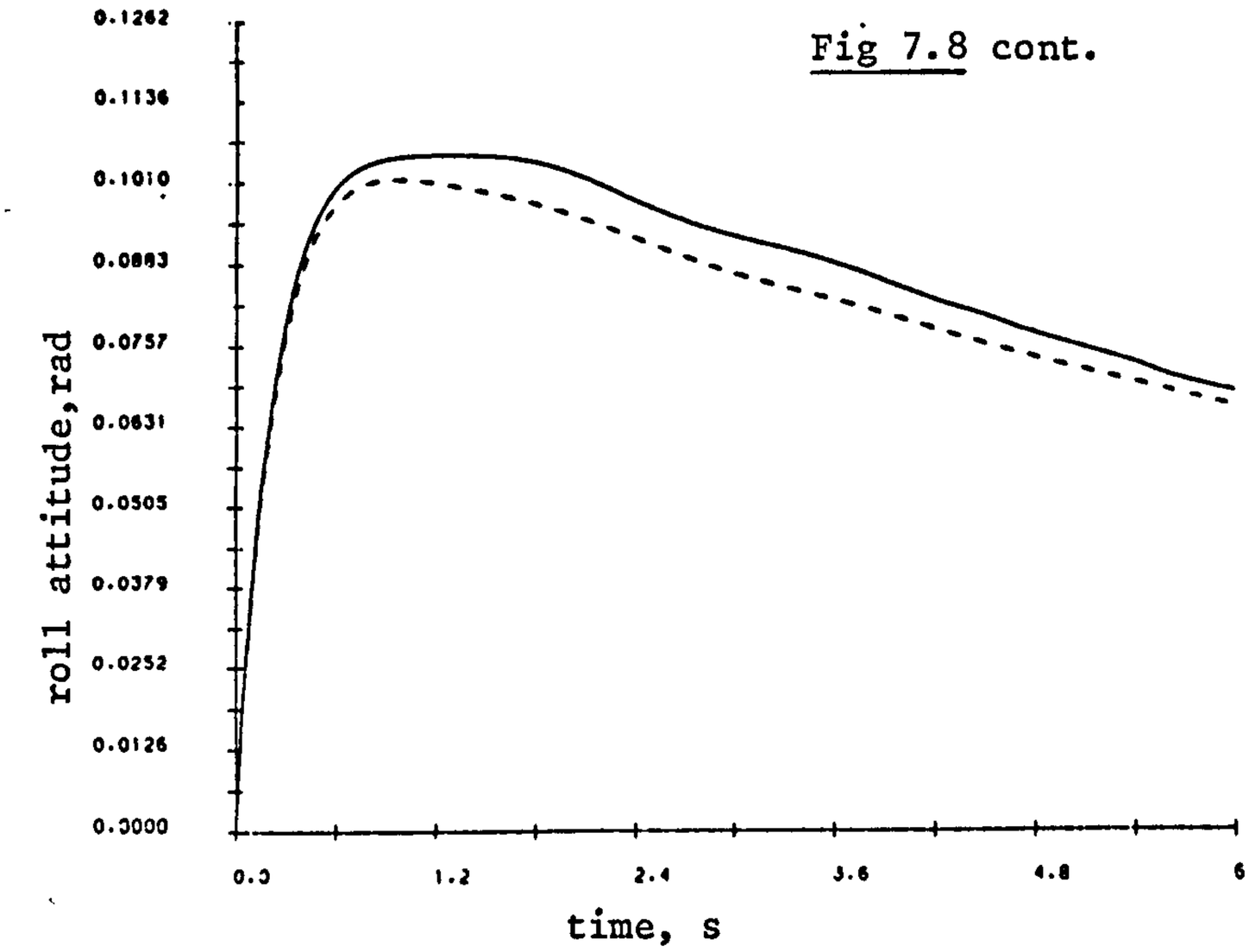
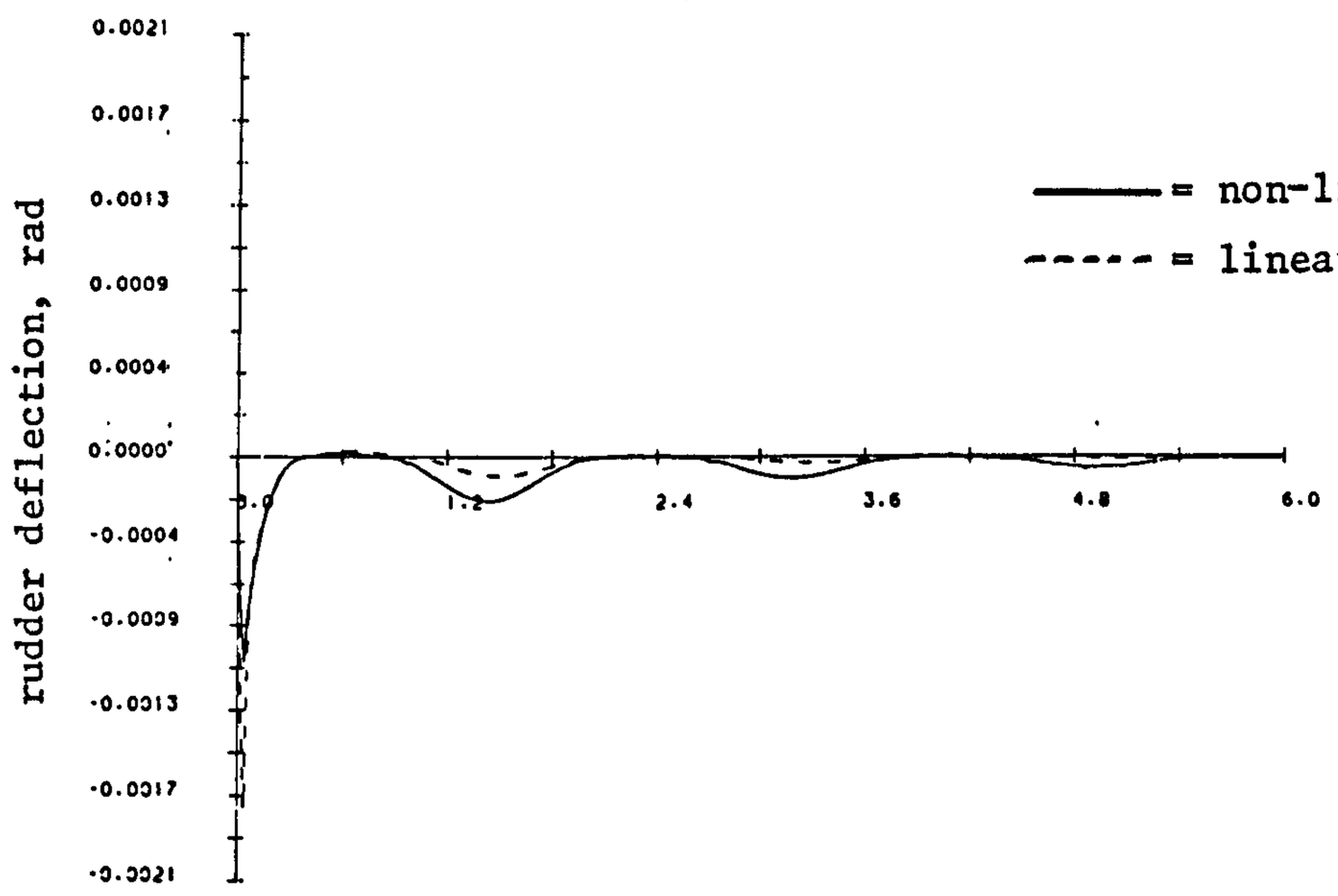


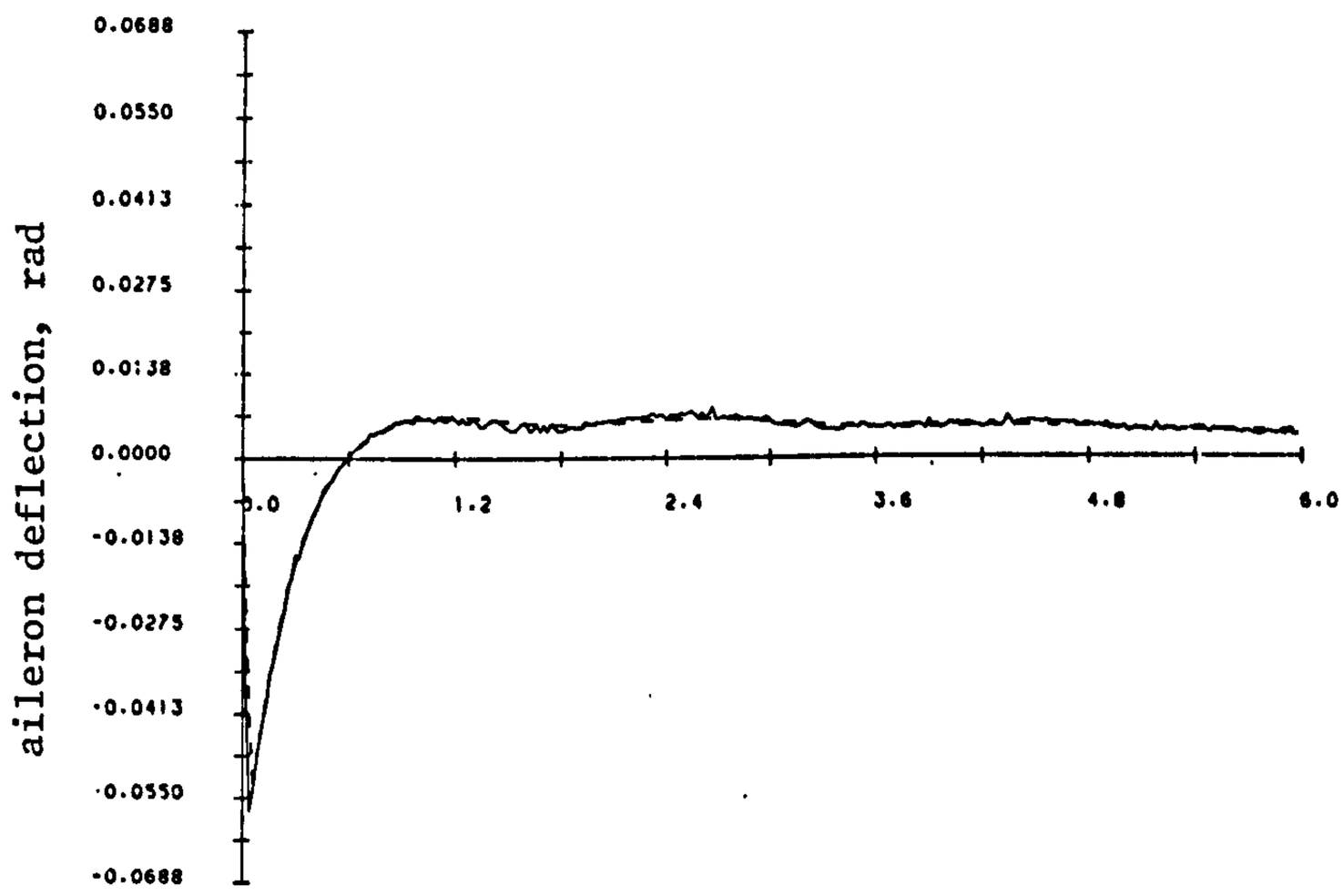
Fig 7.8 cont.



d)



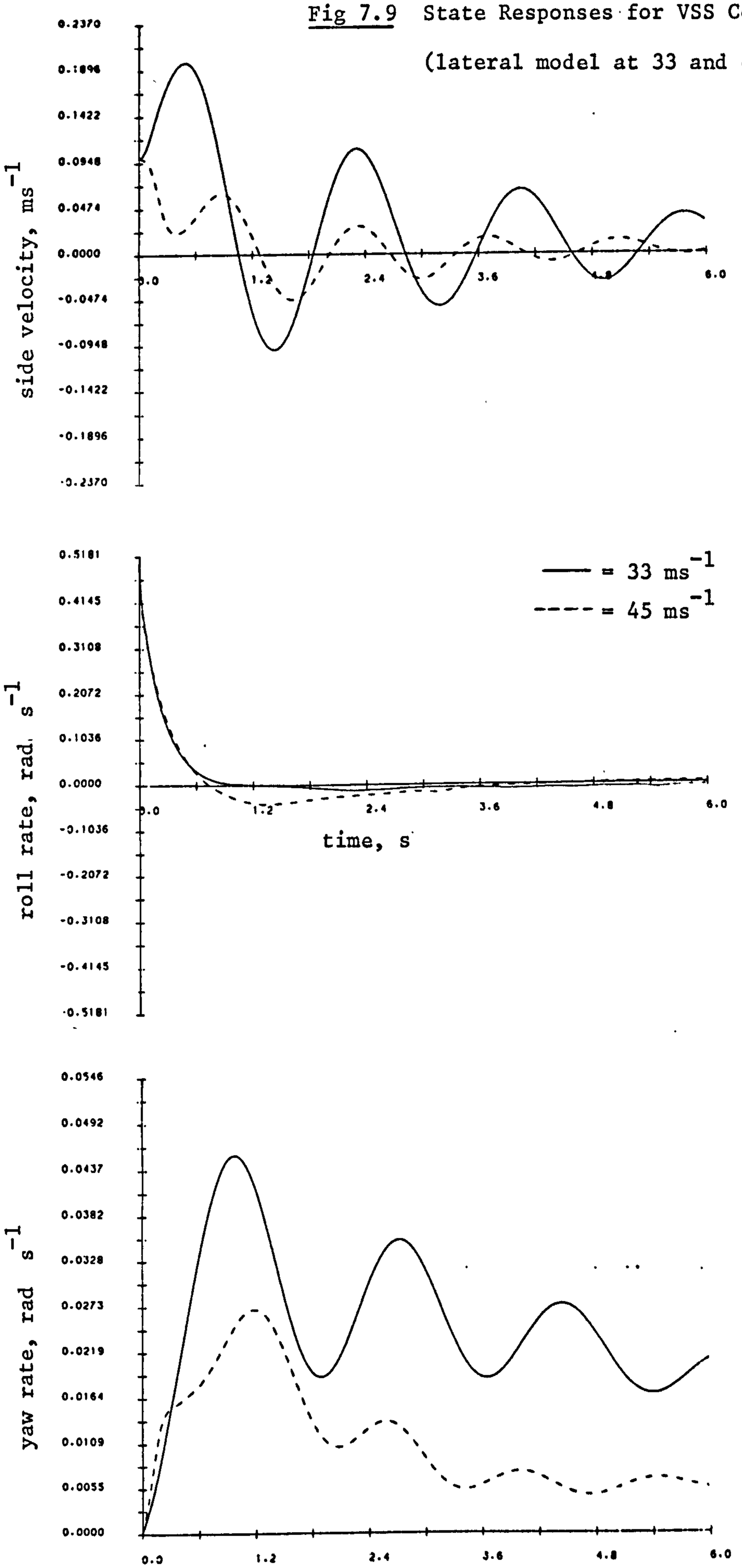
e)

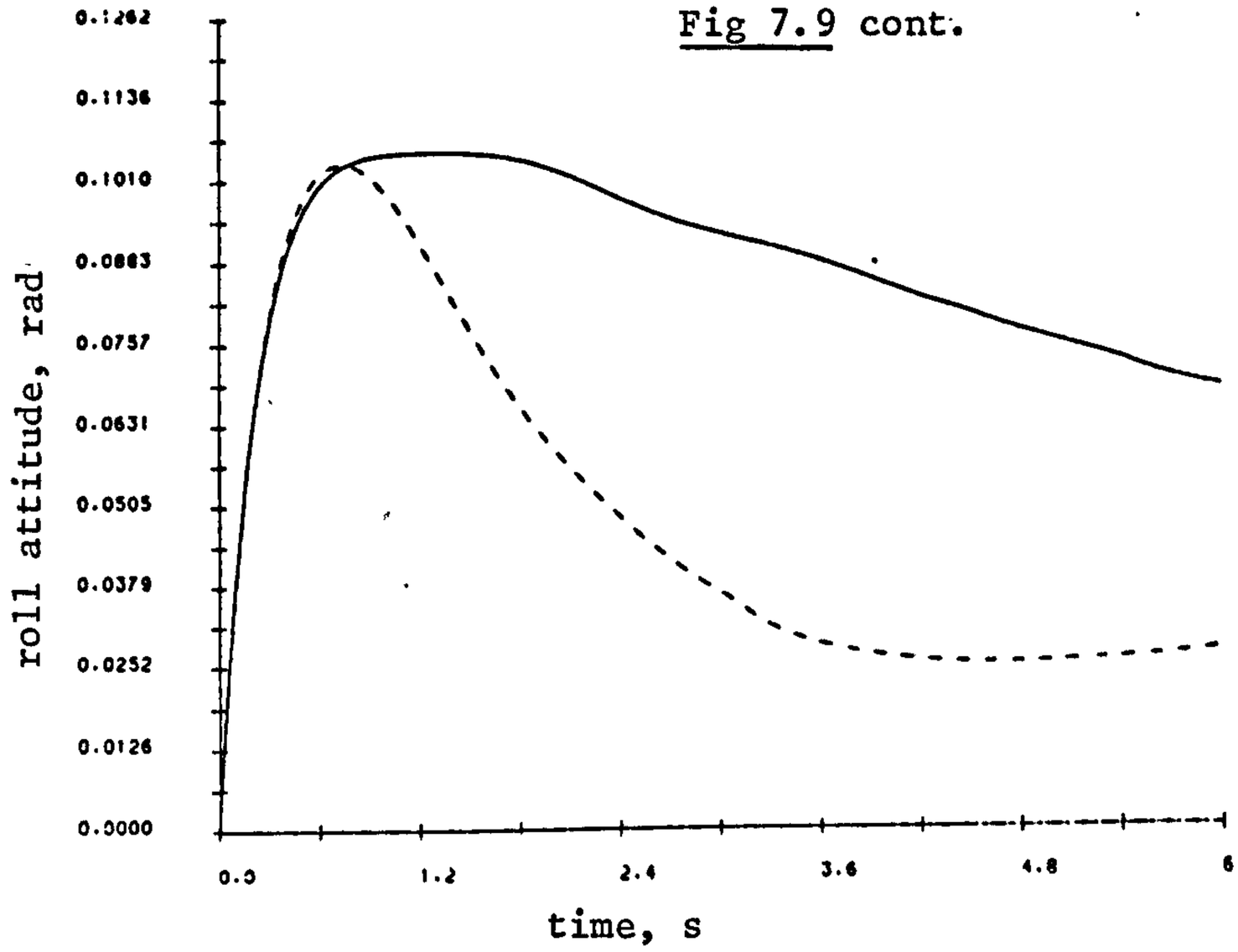


f)

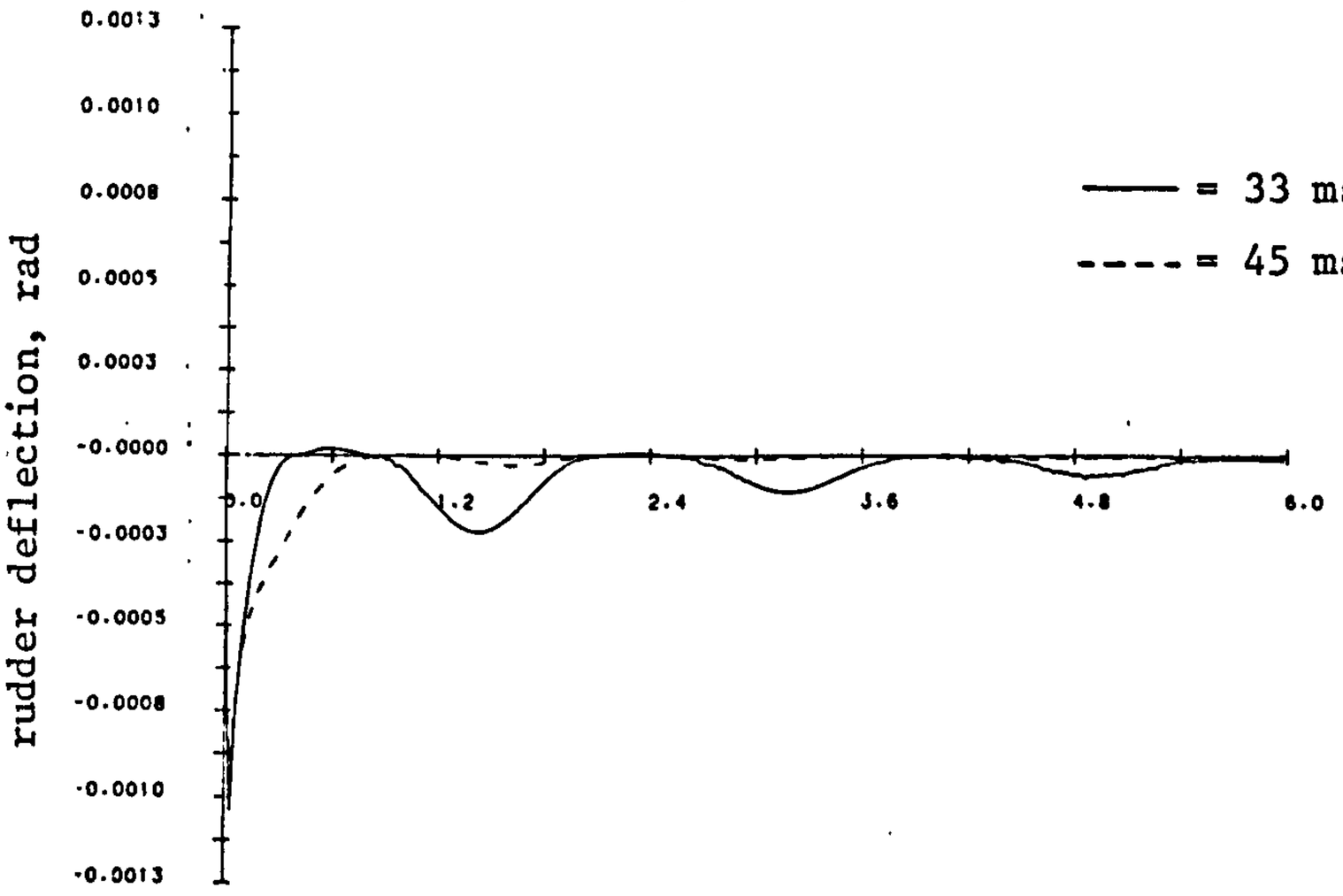
Fig 7.9 State Responses for VSS Controller

(lateral model at 33 and 45 ms^{-1} airspeeds)

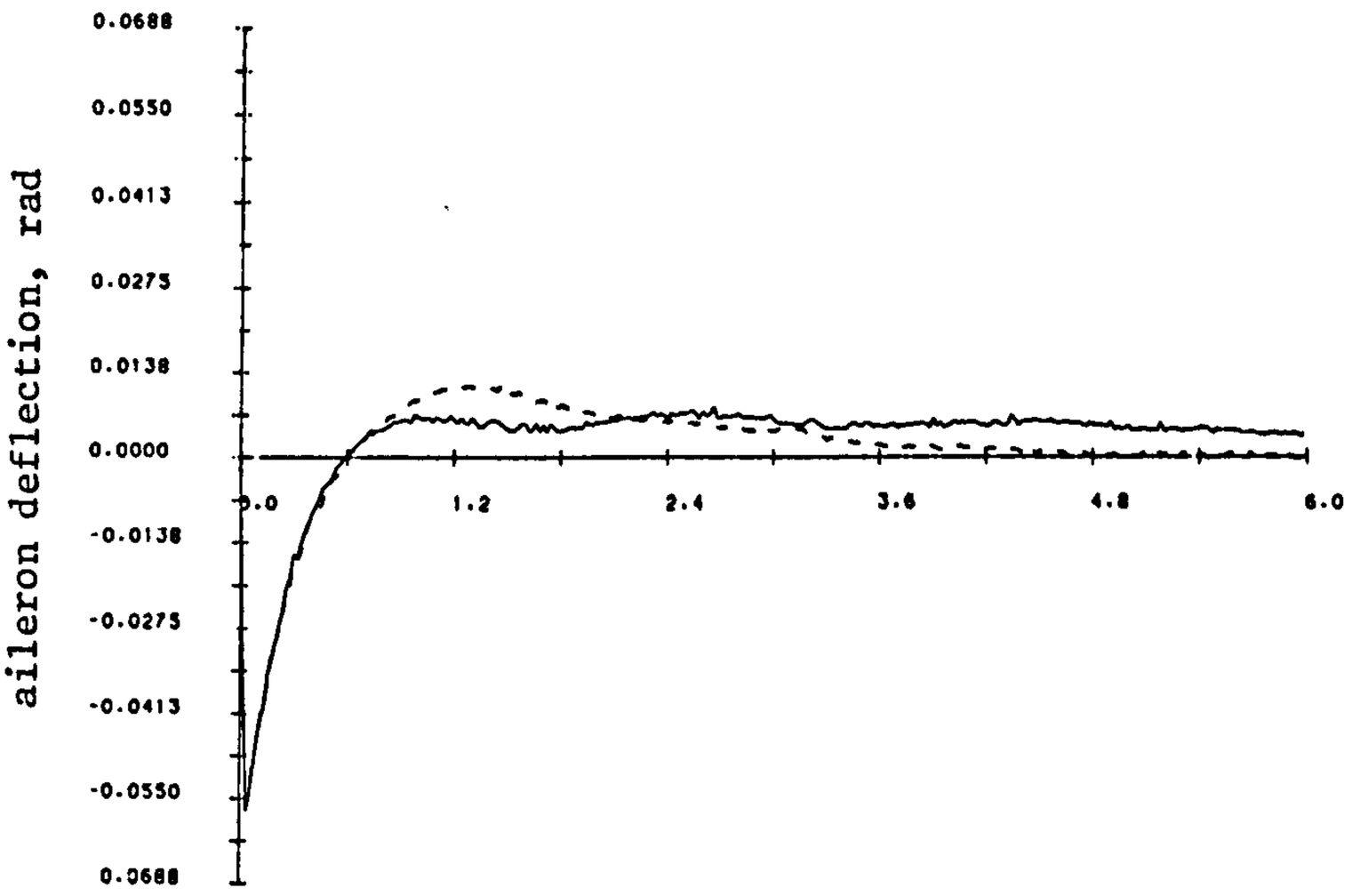



roll attitude, rad
0.1262
0.1136
0.1010
0.0883
0.0757
0.0631
0.0505
0.0379
0.0252
0.0126
0.0000
0.0
1.2
2.4
3.6
4.8
6.0
time, s

d)


rudder deflection, rad
0.0013
0.0010
0.0008
0.0005
0.0003
-0.0000
-0.0003
-0.0005
-0.0008
-0.0010
-0.0013
0.0
1.2
2.4
3.6
4.8
6.0
—— = 33 ms^{-1}
---- = 45 ms^{-1}

e)


aileron deflection, rad
0.0688
0.0550
0.0413
0.0275
0.0138
0.0000
-0.0138
-0.0275
-0.0413
-0.0550
-0.0688
0.0
1.2
2.4
3.6
4.8
6.0

f)

increase the effectiveness of the control surfaces in addition to changing the modal structure. This can be considered as effectively changing the gain in the actuator channels. Our VSS controller is only guaranteed insensitive to parameter changes which lie within the null space of the B matrix, since it is these which are contained in the sliding mode design. Changes in the effective actuator gains will hence affect the system response since these will lie within the range space of the B matrix. It is therefore none too suprising that the responses of Figs. 7.9 show such a degree of disparity, bearing in mind the fact that the control surface deflections are broadly similar. Given the fact that the spiral mode is now somewhat faster than desired, the overall response at 45 ms^{-1} is acceptable and clearly stable with the desired Dutch roll decoupling into the v and r states. Control surface demands are small in both cases.

It is unrealistic to consider the robustness properties of this control policy in isolation and Chapter 8 investigates the robustness achieved by VSS with that of the optimal control policies of Chapter 6.

In the above design recall that we employed full state feedback with 'switching' based on attaining two six-dimensional hyperplanes. This leads to a rather complex controller structure, even when equation 7.40 is employed. In addition it is necessary to have available all states for this implementation. Section 6.7 discussed and demonstrated the use of a reduced state model for the v and r Dutch roll subsystem with simple proportional gain employed in the roll channel. This decomposition was shown to provide acceptable performance when implemented via an optimal control policy. A reduced order (SISO) Dutch roll VSS controller as applied to the aircraft problem will now be examined.

7.7 Reduced Order Dutch Roll VSS Control

Dutch roll is largely constituted by the v and r states of the lateral motions. In section 6.7 a reduced model for these two states was derived with the inclusion of the

control surface, the rudder, actuator. This gives a third order single input sub-system of the form of equations 6.59, i.e.

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} -0.277 & -32.98 & -5.432 \\ 0.365 & -0.319 & -9.49 \\ 0. & 0. & -5. \end{bmatrix} \begin{bmatrix} v \\ r \\ \tau \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} \tau_d \quad - 7.53$$

Deriving the state 'A' and 'B' matrices from the above equation and applying the VSS design technique to this simple system a VSS controller may be derived as was done previously for the complete lateral system. For the null space dynamics we again choose the Dutch roll mode to have two complex eigenvalues of $-0.63 \pm 2.42j$,the eigenvectors being fixed in this case. This second order null space dynamic will then govern the behaviour of the system during sliding. After suitable transformation and partitioning, a consideration of the eigenstructure of the (2 x 2) equivalent system matrix, equation 7.39, gives the switching hyperplane matrix, C, as

$$C = [-0.002 \quad -0.0059 \quad 0.1]$$

in order to achieve the desired sliding mode. As before, the range space dynamics (first order in this case) are chosen to be fast and stable to ensure reachability and with due regard to the actuator bandwidth constraints. A choice of -10.0 for this range space eigenvalue should maintain resonable actuator demands for, in this case, the rudder. Employing the control design as in equation 7.40, i.e. a scaled unit-vector type of control, the following L, M and N matrices are obtained :

$$L = [0.0213 \quad -0.0083 \quad -0.5664]$$

$$M = [0.0001 \quad 0.0003 \quad -0.005]$$

$$N = [-0.001 \quad -0.0029 \quad 0.05]$$

The performance of such a controller was initially investigated by simulation with the simple third order model

of equation 7.53. The controller should provide the desired eigenvalues for the complex Dutch roll pair and ensure rapid reaching of the sliding hyperplane. Figs 7.10 a) to e) show the responses of the nominal third order simulation (solid lines) with the non-linear controller for an initial condition of 0.1 ms^{-1} in v. Note that the Dutch roll mode is evident on both the v and r responses in addition to the actuator response. Note, however, that there is some 'chatter' on the actuator demand signal (Fig 7.10 d)) due to the discontinuous nature of the control. Note also that the switching function (7.10 e)) is close to zero, clear evidence that the sliding mode is achieved and maintained. Also shown in Figs. 7.10 are the system responses due to a representitive $\pm$ 20% variation in the system 'A' matrix used in the simulation. The Dutch roll period is clearly changed by this action but damping remains similar in both cases. The actuator demands remain small, however.

To assess how this reduced order controller performs when used with the complete lateral motion model the controller derived above was employed with the linearised simulation of the lateral dynamics used previously for 33 ms^{-1} airspeed. The results of this simulation run are shown in Figs. 7.11 a) to f) (dotted response). For the purposes of this simulation the roll loop was closed from roll angle to aileron with a simple proportional gain. The roll loop is therefore a linear control loop and the roll rate, p, roll angle, ϕ, and aileron, ξ, responses reflect this fact. The non-linear controller is implemented in the yaw channel and uses the rudder as the control surface. The v and r responses demonstrate an acceptable Dutch roll performance with approximately the desired 2.3 sec. period, although the damping may be considered to be slightly more than desired. Also shown in Figs 7.11 a) to f) are the corresponding state responses when an identical control strategy was employed with the non-linear lateral motion model. The proportional roll loop clearly remains stable although with a changed dynamic. The Dutch roll mode is reasonably close to the linear simulation although the control activity is clearly increased. The control action is, however, roughly similar

Fig 7.10 Reduced Order VSS Controller Responses

(nominal 33 ms^{-1} dutch roll dynamic,

± 20 % deviation in A matrix)

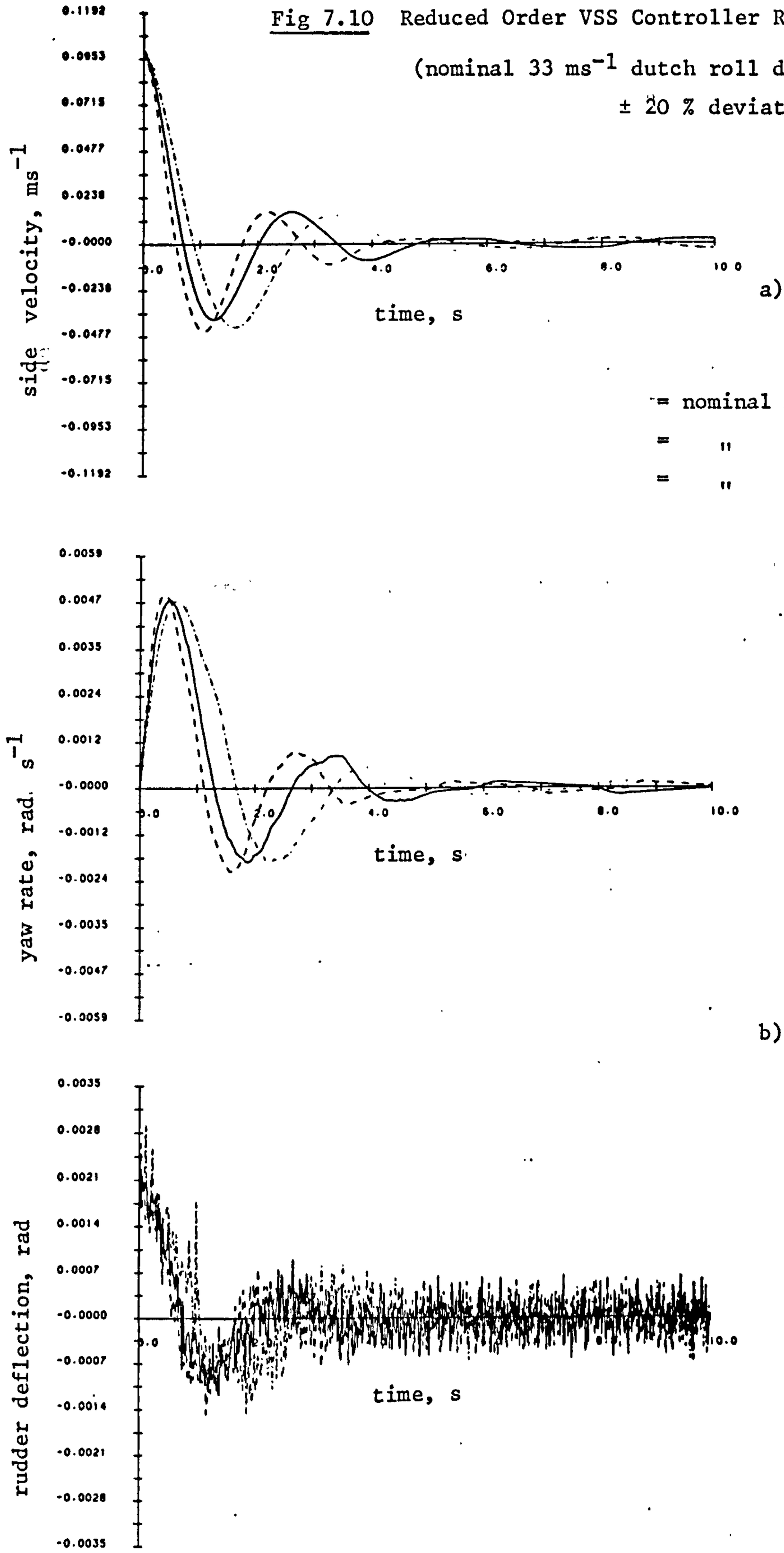
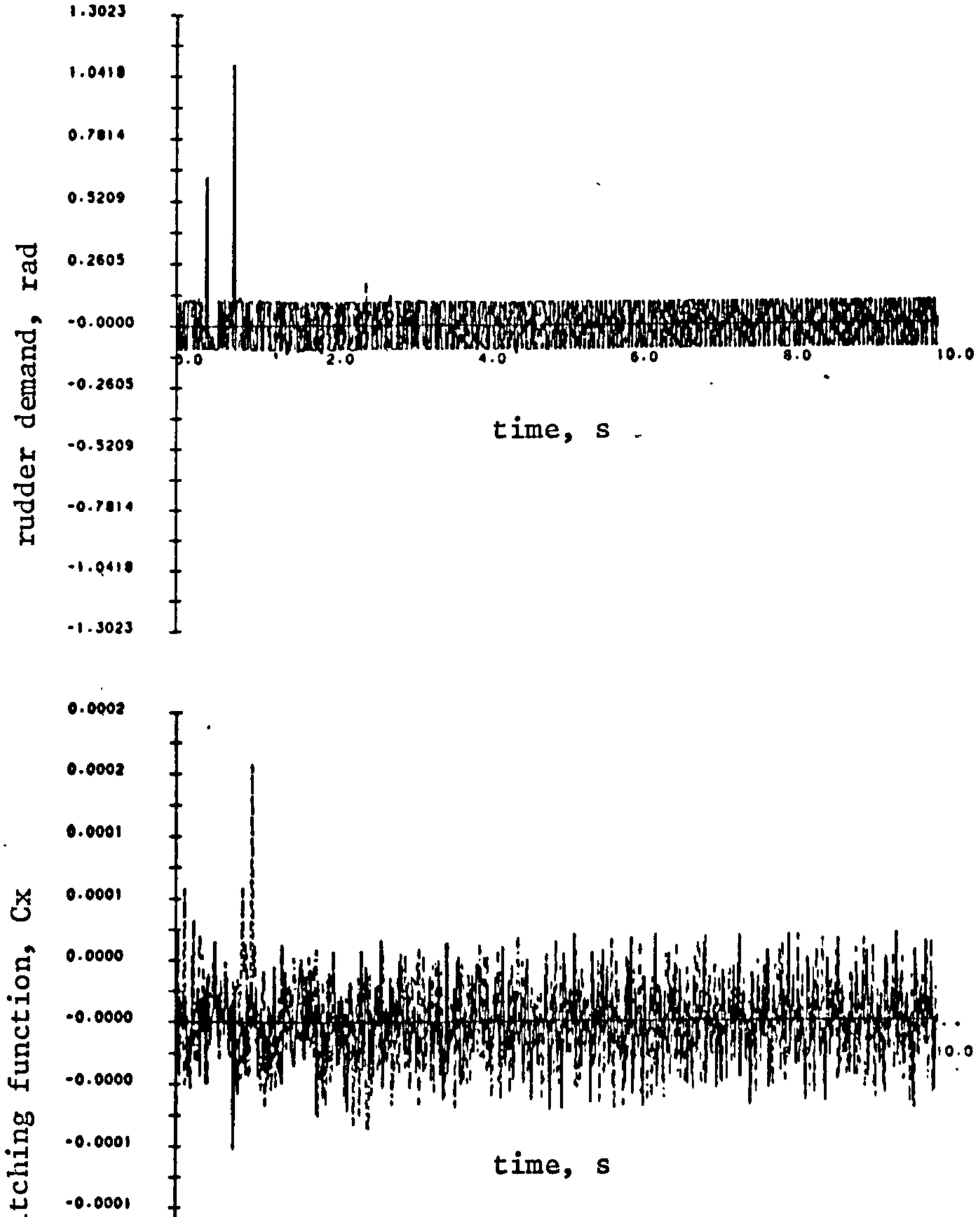
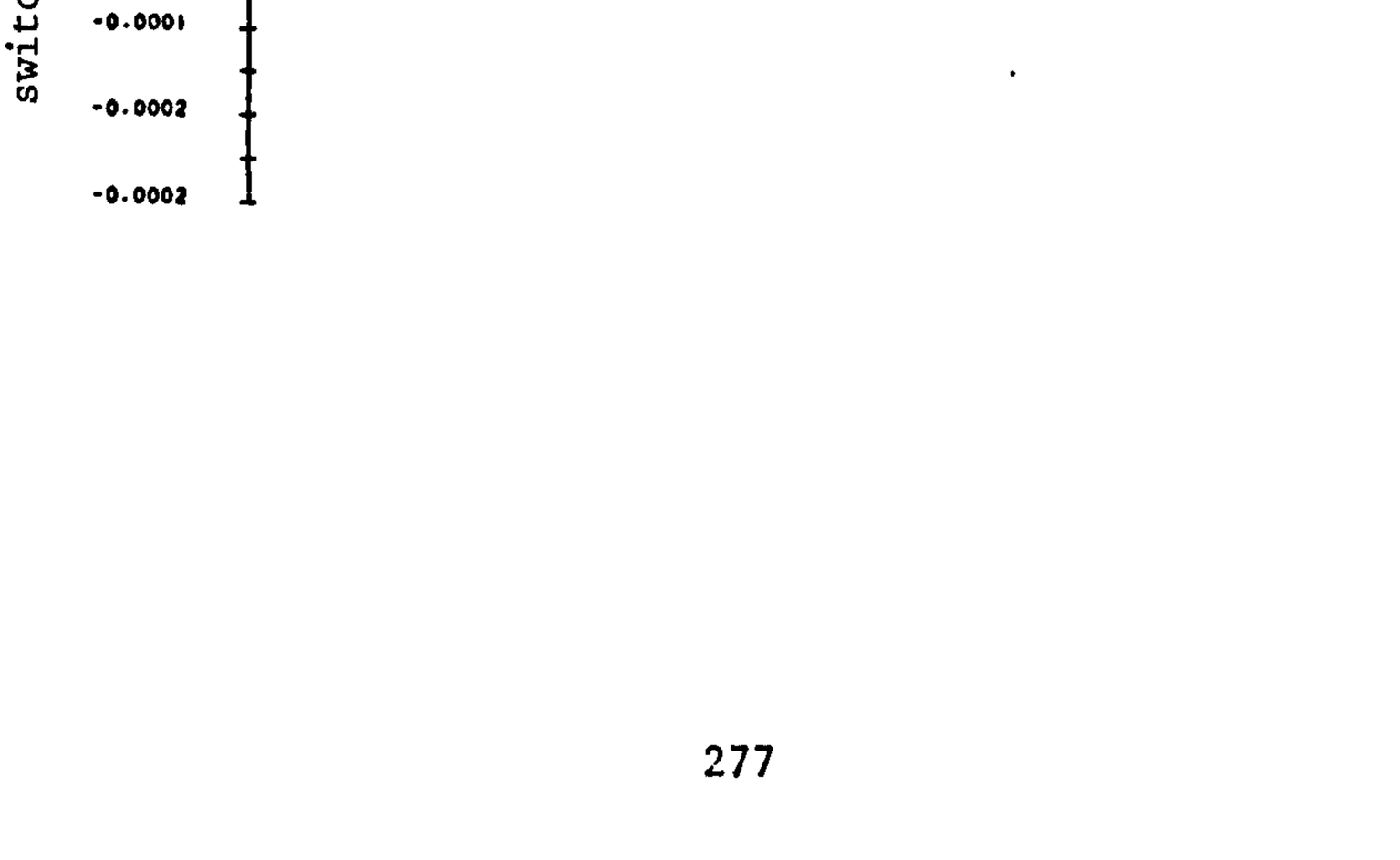


Fig 7.10 cont.



d)



e)

Fig 7.11 Reduced Order Dutch Roll State Responses for VSS Control (lateral model at 33 ms^{-1} airspeed, linear vs. non-linear)

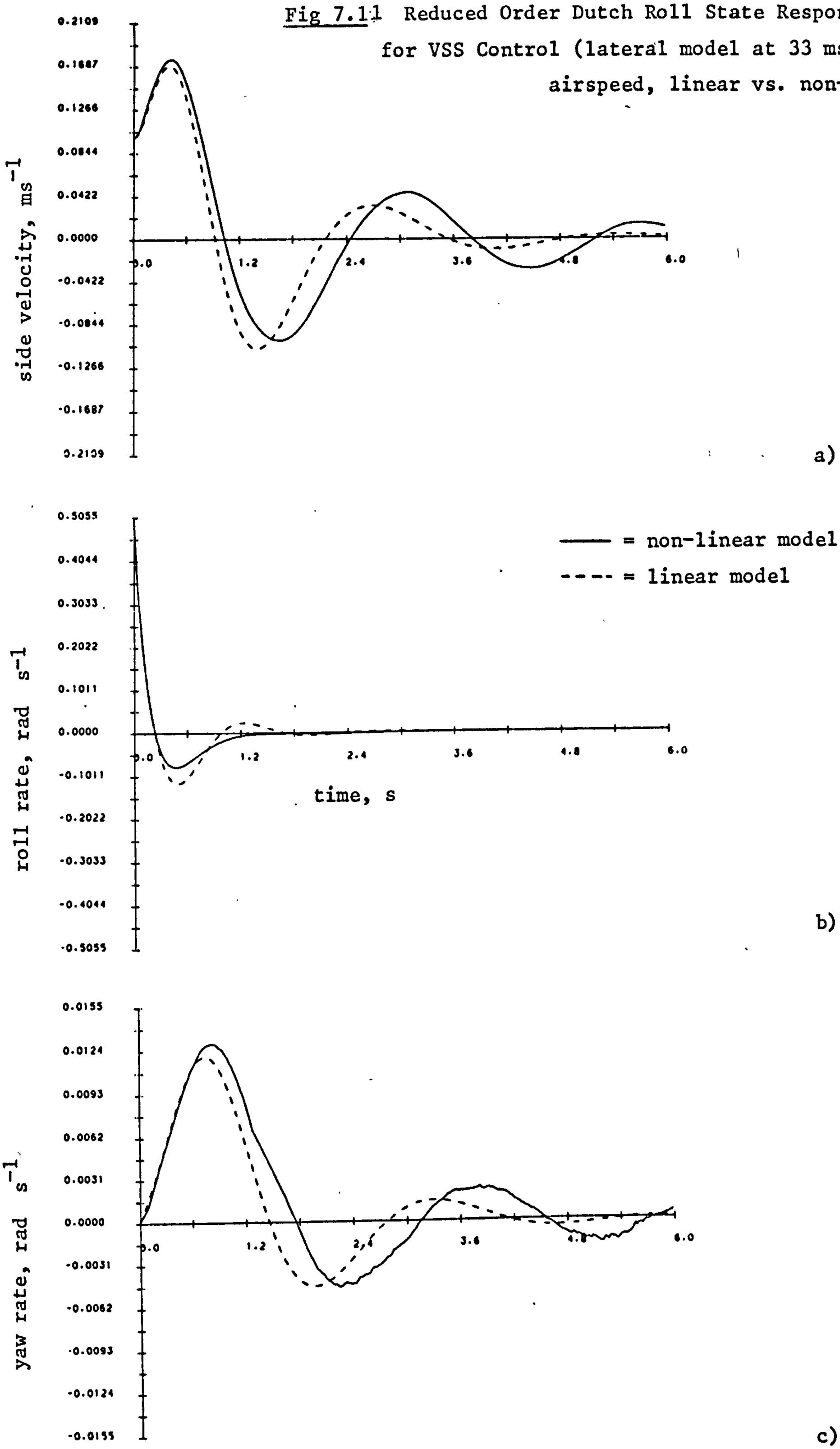
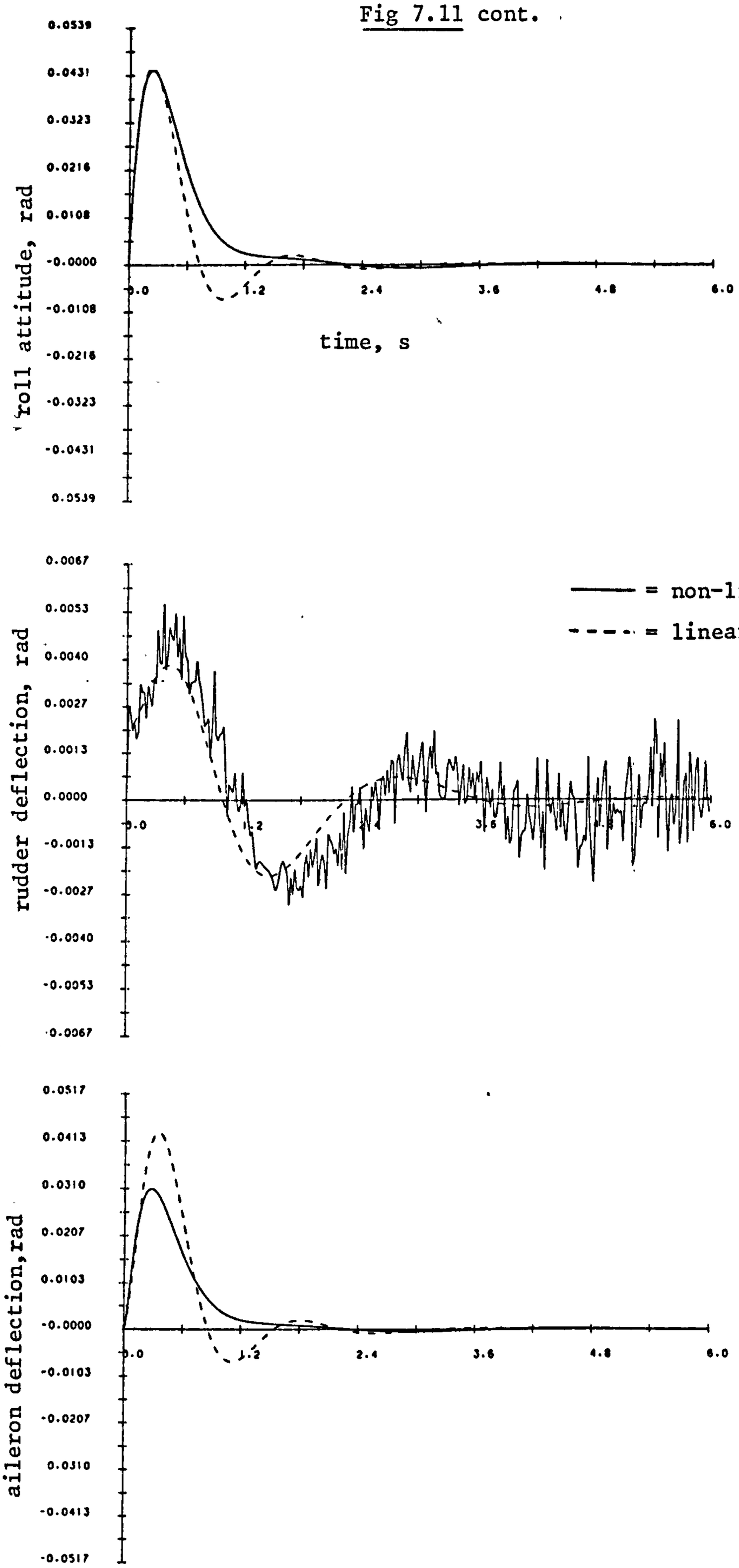


Fig 7.11 cont.


0.0539
0.0431
0.0323
0.0216
0.0108
-0.0000
-0.0108
-0.0216
-0.0323
-0.0431
-0.0539
roll attitude, rad
0.0
1.2
2.4
3.6
4.8
6.0
time, s
d)
0.0067
0.0053
0.0040
0.0027
0.0013
0.0000
-0.0013
-0.0027
-0.0040
-0.0053
-0.0067
rudder deflection, rad
——— = non-linear model
- - - - = linear model
e)
0.0517
0.0413
0.0310
0.0207
0.0103
-0.0000
-0.0103
-0.0207
0.0310
-0.0413
-0.0517
aileron deflection,rad
f)

in both cases. The relatively large demands placed upon the actuator, the rudder, by the discontinuous control may be considered a disadvantage of this scheme although much of the 'chatter' could be filtered out prior to the actuator. The larger degree of 'chatter' in the non-linear simulation is probably due to the influences of the 'unmodelled' system states.

The reduced order controller would thus appear to provide reasonable control of the Dutch roll mode when employed with the full non-linear system. It would, however, appear that the demands placed on the rudder are quite severe. It is also important that the cross-coupling inherent in the actual system between roll and yaw cannot force the v,r subsystem away from the sliding mode. These aspects require further investigation in order to ensure the stability and robustness of this scheme.

7.8 Summary

In this Chapter a variety of non-linear control schemes have been investigated and the 'performance' of each assessed. Of the schemes examined the VSS sliding mode theory would appear to be the most promising when compared with the heuristic approach of ZOC and VICTOR. This type of control scheme has a considerable mathematical foundation, as has been shown, and has very desirable robustness properties. The investigations of the VSS control applied to our particular application, the Machan aircraft, have demonstrated that the achievable performance is also acceptable in addition to the provision of a degree of parameter insensitivity.

These initial studies now require extending to provide a quantitative assessment of the improvements in parameter insensitivity achieved. In the following chapter the robustness aspects of both the VSS and optimal control schemes are briefly considered with a view to providing a critical comparison of these techniques.

CHAPTER 8

Aircraft Sensitivity Analysis

8.1 Introduction

To facilitate the analysis and design of control systems for physical systems we must, in general, derive a linear mathematical model of the particular system. This model will inevitably be an abstract simplification of the real system which will in practice probably be highly non-linear or physically highly complex. The motivation behind even attempting such a gross simplification is to enable us to apply relatively powerful mathematical techniques to the control design and synthesise suitable controller structures. We must, inevitably, lose some of the inherent properties of the physical system when such models are derived. For example, the non-linear nature of the system is always lost when we derive a linear or linearised system description. The resulting linear model will no longer be applicable to the physical system if, say, the operating point of the system changes. It is therefore inevitable that a control system designed for one particular system model will not perform as desired when implemented on the physical system which is highly unlikely to exactly match the simplified model for which the controller was designed. Such a statement does, naturally, raise the question as to why control systems are designed on the basis of linear models ? One is led to suppose that the answer to this question is "we have to start somewhere", which probably satisfies the pragmatists or engineers faced with practical problems. The purist would probably point to the fact that the controller designed for the linear system will provide 'acceptable' control over a bounded range of plant variations. Outside this range the performance will fail to be acceptable in some sense, e.g. poor stability, poor time response, etc. This latter approach has led to the development of a large body of theory which attempts to answer the question 'how well does a given controller strategy perform when faced with the real time-varying or

non-linear system' ? This property of control systems is generally termed sensitivity or robustness (i.e. the controller can be considered insensitive to a bounded set of changes in the system model).

One underlying concept of control system theory is that of feedback. Feedback is a very useful tool since not only does it allow a desired input-output relationship to be realised but it also reduces the effects of uncertainties on the system performance. Employing feedback thus reduces the overall effects of changes in the system parameters and allows a given performance to be maintained. This is a very important property and provides the motivation behind much control system theory in general. The provision of feedback control therefore improves the system performance and reduces the degree to which this performance changes as a result of system uncertainties.

Returning to our specific problem, that of the aircraft system, this is a highly 'uncertain' system. Primarily, this uncertainty is due to the non-linear nature of the aircraft equations of motion in addition to the external influences on the airframe, e.g. measurement noise, wind gusts, turbulence, all give rise to plant uncertainty.

In Chapter 2 a linearised model of the aircraft was derived making certain gross simplifications regarding the aircraft operating point and considering only small perturbations about this nominal operation. The linear model thus derived will therefore only be applicable at, or close to, this operating point. As an aircraft normally passes through a range of such operating points in any given manoeuvre, the model will be invalid over the majority of the aircraft operations.

In Chapters 5, 6 and 7 a number of different control strategies have been developed. Some pains have been taken to illustrate how these controllers perform with the non-linear system using a non-linear simulation. To provide a somewhat more rigorous analysis of the likely robustness properties of these controllers, the following discussion examines the degree to which each improves the

eigenvalue/eigenvector sensitivity and hence provides insensitive modal assignment. Perhaps initially we should differentiate between the terms robustness and sensitivity.

Whilst both of these terms have been used to refer to the ability of a system to tolerate uncertainties in modelling or non-linearities there is a definite difference between these measures. The following definitions due to Frank (90,91) serves to clarify this difference :

i) Parameter Sensitivity : A system function (or system) is said to be sensitive to parameter variations if $\underline{S}$ is unlike $\underline{0}$. In the special case $\underline{S} = \underline{0}$ the function (or system) is said to be zero sensitive. If a measure of $\underline{S}$ is small, the system function (or system) is termed low sensitive or insensitive.

ii) Robustness : A system is called robust if the system property of interest remains in a bounded region in the face of a class of finite bounded perturbations.

From the above it is clear that sensitivity refers principally to the local properties of the system around which the sensitivity measure $\underline{S}$ is evaluated at the nominal operating point of the system. This measure is of importance but provides little idea of the global properties of the system as a result of parameter variations which may, of course, change the nominal operating point. Robustness is a more general measure in as much as we define a particular system property to lie within a given region as opposed to at a fixed point. The parameter chosen may, for example, be a measure of relative stability or a measure of admissible control demands. Equally, the classes of perturbations may be quite broad and cover changes in system structure in additon to variations in plant parameters. Clearly then, one would in general, and certainly in the aircraft problem, be more interested in the robustness of the system as opposed to its sensitivity, although the two go hand in hand, to some extent.

A robust controller can be defined as one which maintains a given measure of system performance within prescribed limits in the face of a bounded set of parameter

variations. Much work has been conducted into providing specified robustness properties from conventional control design methodologies, for example LQP, etc. (90-108). These state or output feedback structures inevitably provide this robustness by sacrificing performance (90). The technique of VSS control changes the feedback structure dynamically and this may provide more desirable robustness properties (60,62,77,81).

The mathematical complexity which accompanies the robustness/sensitivity problem is somewhat intractable in all but the simplest of cases. For the present discussion a somewhat more pragmatic viewpoint is adopted and an examination made of the degree of 'robustness' achieved by the controller designs already considered. (A considerable body of literature is available on this topic and a short list of applicable work is included in the references (90-108,121,122)). Previous Chapters have developed both LQP and VSS controllers for the lateral motion sub-system of the aircraft model. It is these which will now be examined with a view to assessing the robustness achieved in both cases.

8.2 Robustness of Lateral SAS Controllers

Before moving on to examine the robustness aspects of the lateral SAS controllers a measure of a particular system property which it is required to be robust in some sense must be defined. There is clearly a considerable choice available here. It might, for example, be required that the time response of the system remain within certain bounds or that the demands placed on actuators be within prescribed limits. Obviously, these bounds will depend to a large extent upon the particular application and will probably be derived from a consideration of the performance requirements placed upon the real system. In the aircraft problem it is normal to specify, for example, that the Dutch roll transient response be bounded so that over the flight envelope this mode retains the specified degree of damping, etc. An alternative way of looking at this is to require that the system poles which constitute the Dutch roll (v and

r) remain within some bounded region on the s-plane as a result of changing flight conditions. This places a requirement primarily on maintaining the eigenvalues at prescribed locations. Allied to this must be a need to maintain prescribed eigenvector directions such that the 'designed for' degree of motal decoupling is achieved. It will be recalled that in both the LQP and VSS designs both eigenvalue and eigenvector assignment was provided based on a nominal lateral system model. A reasonable measure of robustness for these systems may thus be the degree to which the eigenstructure of the closed-loop sytem is maintained with typical variations in aircraft parameters. If such a measure is to be used limits for the variations which will inevitably occur in say the closed-loop eigenvalues must be initially defined.

For the purposes of the present discussion a fairly pragmatic view is adopted and an investigation made of the eigenvalue and eigenvector variations brought about by a typical manoeuvre in the non-linear aircraft simulation using the LQP and VSS designs. Before moving on to these aspects it is instructive to examine how the open-loop aircraft eigenvalues vary over a typical manoeuvre and for various airspeeds.

8.2.1 Open-Loop Aircraft Eigenvalue Sensitivity

From a practical viewpoint it would appear to be quite difficult to determine the aircraft eigenstructure over a typical manoeuvre. It would , for example, be difficult to carry out such a procedure 'on-line' in a real aircraft and, indeed, the results would reflect the closed-loop structure since, presumably, some type of SAS system would be operative. The only practical method is to examine the aircraft dynamic properties using simulation techniques as has already been demonstrated. A fully non-linear simulation would embody the aircraft structure and hence behave as the aircraft in flight. The real advantage here is that one has immediate access to all of the system states and variables. Even with this access it is difficult to determine any results regarding actual system modes and

distributions of these. Remember that in Chapter 2 two alternative techniques were reviewed for deriving linearised models for the aircraft. In each case a state space description of the aircraft was produced either via. purely intuitive arguments, given the physics of the aircraft, or by deriving the first order approximations to the non-linear equations. From such a state space model it is a relatively simple matter to compute the system eigenvalues and eigenvectors. Of the two techniques reviewed the linearisation using first order approximations to the individual elements of the system 'A' and 'B' matrices appears to be the most promising. This is largely due to the fact that the functional dependence of each of these elements on the set of state variables $\underline{x}$ is retained. In this way a locally linearised model results but this linearisation may be done at each point in the aircraft's flight envelope. If this is the case a set of A and B matrices for the system may be determined, each of which is applicable for a given set of the states, $\underline{x}$. The eigenstructure of the system represented by these A's and B's may then be found.

The technique adopted in the present studies has been as follows :

i) Choose a particular flight configuration and manoeuvre schedule for the aircraft.
ii) Using the non-linear simulation evaluate the elements of the system A and B matrices via. the results of Chapter 2 for a number of discrete time points.
iii) Evaluate, at each time point, the eigenstructure of the A matrix.
iv) Plot the eigenvalues graphically.

Using this technique gives a clear indication of how the eigenvalues of the system vary over the particular manoeuvre chosen in i).

To demonstate this technique a typical aircraft manoeuvre demand is considered. The discussion is limited to the lateral motion sub-system of the aircraft for which both VSS and LQP controllers have already been derived. In

the following results the aircraft simulation is employed with closed-loop feedback via. one of the above control schemes. The open-loop A matrices are derived from a straightforward application of the first variational expressions of Chapter 2. The eigenvalues and eigenvectors then follow. The closed-loop eigenvalues and eigenvectors may be derived by evaluating $\det\{(sI - (A_{1a} + B_{1a}K_{1a}))\}$ where the open-loop A and B are evaluated using the above techniques and K_{1a} is the nominal state feedback gain matrix. The locally linearised open-loop A's and B's are clearly valid whether the aircraft is employed in an open- or closed-loop configuration.

The particular manoeuvre chosen is an initial condition of 1 ms^{-1} in side velocity and 0.5 rad s^{-1} in roll rate. This will provide relatively large perturbations in all the lateral states with realtively large changes in the lateral A and B matrices. It will also maintain consistency with the time responses obtained previously. This manoeuvre was carried out at a range of airspeeds, namely 30, 40 and 50 ms^{-1}. The longitudinal motions have not been considered here but these were held approximately constant over the lateral manoeuvre. Note, however, that some lateral/longitudinal coupling is inevitable. A reference to Table 3, Chapter 2, should make this clear since the majority of the aerodynamic derivatives change with airspeed, equally, some pitch coupling into lateral motions is inherent in the structure of the actual aircraft (a non zero pitch angle is normal during static flight). In evaluating these linearised equations, and hence the eigenvalues, cross-coupling effects are included implicitly.

To demonstrate the typical magnitude ranges of the elements of the open-loop linearised lateral state A and B matrices a number of samples are shown in Table 8.1. These were derived from the non-linear simulation using the locally linearised equations of Table 2.2 (Chapter 2) and are evaluated for various values of yaw and roll attitudes. These should be compared with the results of section 2.3.2 where the longitudinal cross-coupling has been ignored. The results of Table 8.1 were obtained for 33 ms-1 forward

Table 8.1

Lateral A and B matrix variations with roll and yaw attitude : 33 ms^{-1} airspeed

φ = 0.0

$$A = \begin{bmatrix} -0.254 & 7.429\text{x}10^{-3} & -30.339 & 9.805 & 0. \\ -9.35\text{x}10^{-2} & -7.862 & 3.82 & 0. & 0. \\ 0.3366 & 6.273\text{x}10^{-2} & -0.3075 & 0. & 0. \\ 0. & 1. & -3.247\text{x}10^{-2} & -3.08\text{x}10^{-3} & 0. \\ 0. & 0. & 1. & 9.499\text{x}10^{-2} & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} -4.62 & 0. \\ 0. & -24.359 \\ -8.009 & 0. \\ 0. & 0. \\ 0. & 0. \end{bmatrix}$$

φ = 0.15 rad

$$A = \begin{bmatrix} -0.2562 & -8.741\text{x}10^{-3} & -29.788 & 9.7932 & 0. \\ -0.111 & -7.79 & 2.994 & 0. & 0. \\ 0.333 & -5.04\text{x}10^{-2} & -0.2799 & 0. & 0. \\ 0. & 1. & -5.844\text{x}10^{-2} & 4.23\text{x}10^{-3} & 0. \\ 0. & 0. & 1.0017 & -7.25\text{x}10^{-2} & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} -4.54 & 0. \\ 0. & -27.95 \\ -7.87 & 0. \\ 0. & 0. \\ 0. & 0. \end{bmatrix}$$

$\varphi = 0.25$ rad

$$A = \begin{bmatrix} -0.2667 & -1.775\text{x}10^{-3} & -29.08 & 9.8 & 0. \\ -0.1222 & -7.765 & 3.94 & 0. & 0. \\ 0.332 & -0.215 & -0.288 & 0. & 0. \\ 0. & 1. & -2.496\text{x}10^{-2} & 7.9\text{x}10^{-3} & 0. \\ 0. & 0. & 1. & -0.317 & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} -4.5 & 0. \\ 0. & -23.7599 \\ -7.812 & 0. \\ 0. & 0. \\ 0. & 0. \end{bmatrix}$$

$\psi = 0.0$

$$A = \begin{bmatrix} -0.2564 & 3.29\text{x}10^{-3} & -30.39 & 9.8 & 0. \\ -9.437\text{x}10^{-2} & -7.8735 & 3.358 & 0. & 0. \\ 0.3371 & 0.1143 & -0.3 & 0. & 0. \\ 0. & 1. & -4.023\text{x}10^{-2} & -6.77\text{x}10^{-3} & 0. \\ 0. & 0. & 1. & 0.16844 & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} -4.634 & 0. \\ 0. & -24.43 \\ -8.03 & 0. \\ 0. & 0. \\ 0. & 0. \end{bmatrix}$$

$\psi = 0.15$ rad

$$A = \begin{bmatrix} -0.254 & -8.324\text{x}10^{-3} & -30.088 & 9.6855 & 0. \\ -9.638\text{x}10^{-2} & -7.79 & 3.45 & 0. & 0. \\ 0.334 & -0.209 & -0.28 & 0. & 0. \\ 0. & 1. & -4.33\text{x}10^{-2} & 1.33\text{x}10^{-2} & 0. \\ 0. & 0. & 0.98921 & -0.306 & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} -4.542 & 0. \\ 0. & -23.944 \\ -7.874 & 0. \\ 0. & 0. \\ 0. & 0. \end{bmatrix}$$

$\psi = 0.25$ rad

$$A = \begin{bmatrix} -0.2527 & -4.884\text{x}10^{-4} & -30.003 & 9.492 & 0. \\ -0.1 & -7.77 & 3.94 & 0. & 0. \\ 0.333 & -0.183 & -0.29 & 0. & 0. \\ 0. & 1. & -1.14\text{x}10^{-2} & 3.08\text{x}10^{-3} & 0. \\ 0. & 0. & 0.9677 & -0.2614 & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} -4.513 & 0. \\ 0. & -23.794 \\ -7.824 & 0. \\ 0. & 0. \\ 0. & 0. \end{bmatrix}$$

speed, u. Inevitably, side velocity, v, and vertical velocity, w, components will be produced in the non-linear model and these will be reflected in the total airspeed, V_T. Hence the differences between the linearised state matrices and those derived in section 2.3.2 can be noted.

Figs. 8.1 a) to f) show the variation in the lateral motion open-loop eigenvalues over a ten second simulation with the initial conditions as above, for 30, 40 and 50 ms^{-1} airspeeds. These plots were obtained using the technique outlined above. The nominal lateral open-loop eigenvalues for 33 ms^{-1} 'stick-fixed' flight, ignoring longitudinal/lateral cross-coupling, are

$$-0.49983 \pm 3.5j \; ; \; -8.5579 \; ; \; 0.0 \; ; \; 0.119$$

It should be expected that the linearised lateral A matrix had eigenvalues close to these but some variation is to be expected as the airspeed varies in addition to the effects of pitch cross-coupling. This latter effect is most strongly evident on the real parts of the Dutch roll eigenvalues and the spiral mode by the large initial transient swings in these , Figs 8.1 a) and e). Consider the effects on each of the modes seperately :

a) The Dutch roll mode has a complex eigenvalue pair. Over the three airspeeds these eigenvalues vary dominantly in their imaginary parts, Fig. 8.1 b), the real parts remaining relatively constant, 8.1 a). After ten seconds of simulation these Dutch roll modes are at $-0.36 \pm 3.64j$ ($\zeta = 0.098$, $\omega_n = 0.58Hz$) and $-0.41 \pm 4.84j$ ($\zeta = 0.085$, $\omega_n = 0.77Hz$) for the 30 and 50 ms^{-1} airspeeds respectively. The damping of these modes decreases by 13.3% and the frequency increases by 32.8% over this 66% change in airspeed. Poorer damping being a consequence of the higher airspeeds.

b) The roll subsidence mode exhibits fairly large variations over the airspeeds chosen. Slight evidence of pitch cross-coupling is present initially. After ten seconds this mode has values of -8.9 and -11.6 for

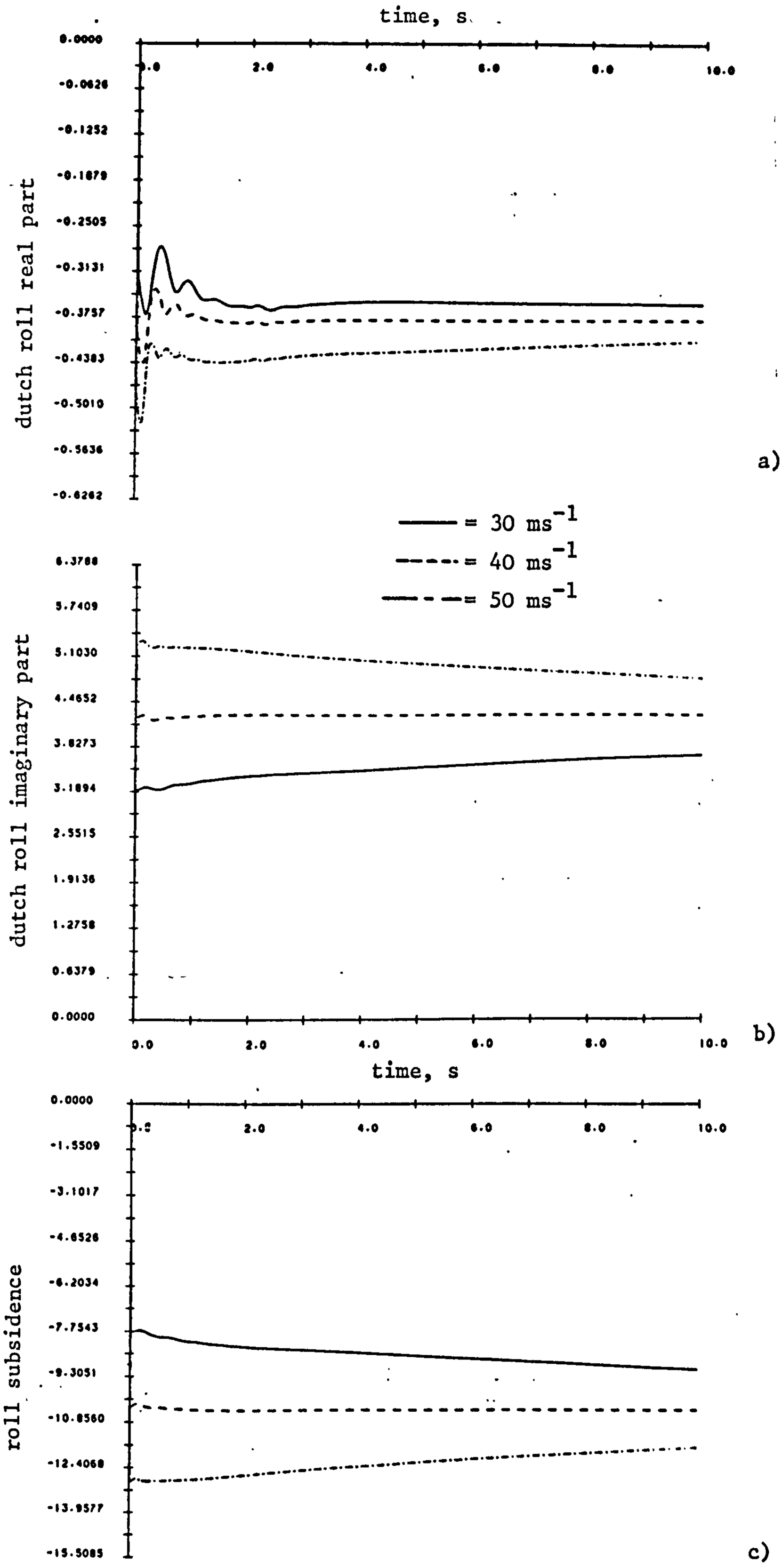
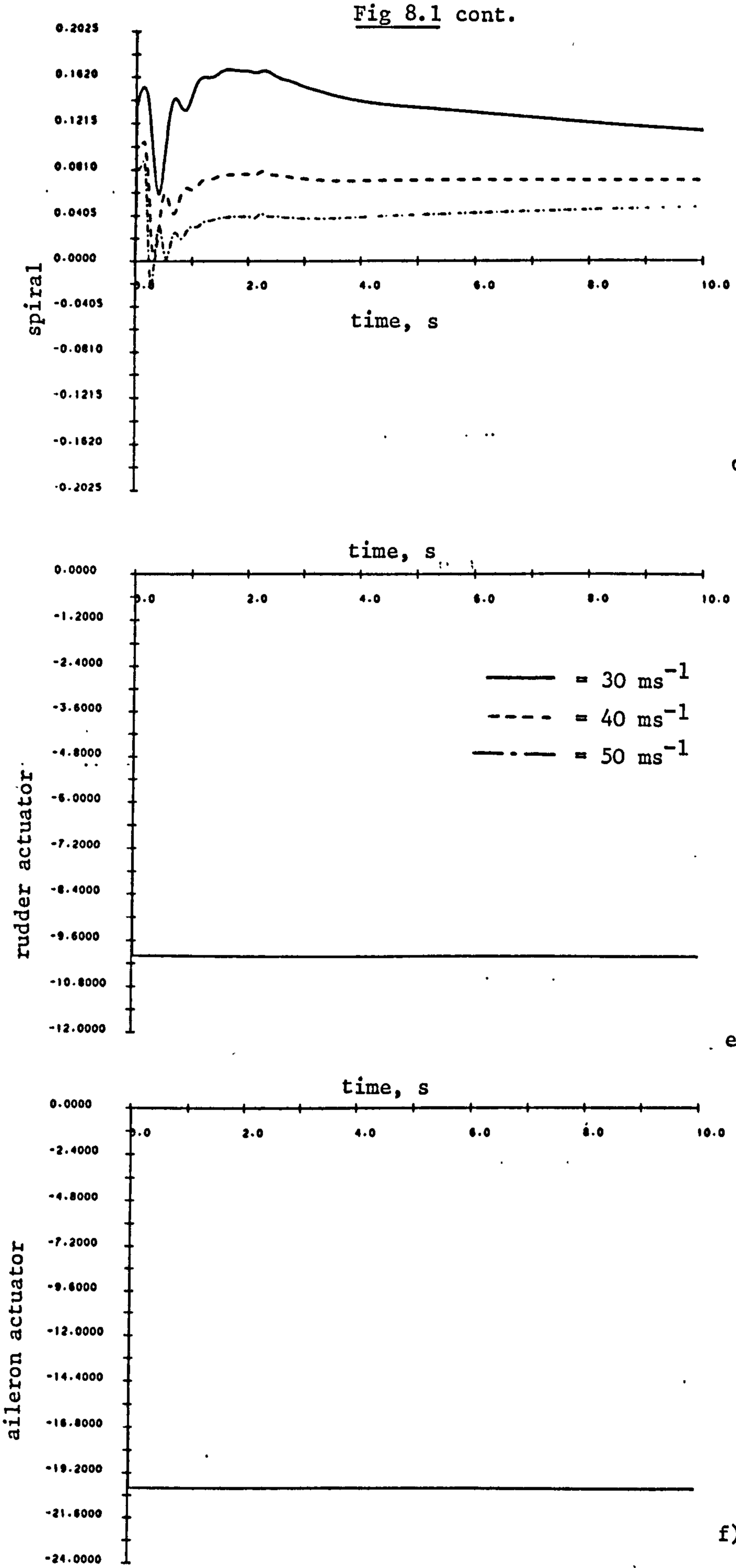


Fig 8.1 Open Loop Lateral Eigenvalue Sensitivity (LQP control)


0.2025
0.1620
0.1215
0.0810
0.0405
0.0000
-0.0405
-0.0810
-0.1215
-0.1620
-0.2025
spiral
0.0
2.0
4.0
6.0
8.0
10.0
time, s
d)
0.0000
-1.2000
-2.4000
-3.6000
-4.8000
-6.0000
-7.2000
-8.4000
-9.6000
-10.8000
-12.0000
rudder actuator
time, s
e)
0.0000
-2.4000
-4.8000
-7.2000
-9.6000
-12.0000
-14.4000
-16.8000
-19.2000
-21.6000
-24.0000
aileron actuator
time, s
f)
= 30 ms-1
= 40 ms-1
= 50 ms-1

the 30 and 50 ms^{-1} airspeeds respectively. This indicates a 30.3% increase in response time for this mode. Note, however, that the higher airspeeds tend to increase the 'stability' of this mode.

c) The spiral mode exhibits quite marked cross-coupling from pitch in its initial transient response. It also is sensitive to changes in the airspeed, but higher airspeeds tend to improve the stability of this unstable mode. Again, after ten seconds the spiral mode has values of 0.069 and 0.043 for the 40 and 50 ms^{-1} airspeeds respectively. This representing a reduction of 60.2% in this eigenvalue.

The remaining three eigenvalues are invariant to airspeed, changes. These are the two actuator modes at -10.0 and -20.0, Figs. 8.1 e) and f), which are simply modelled as first order invariant eigenvalues and the yaw integration which remains a simple integration over all flight conditions.

The open-loop eigenvalues thus exhibit quite marked variations in value over the particular manoeuvre chosen and vary with varying airspeed, the largest variation in mode being from the spiral divergence. After the ten seconds of simulation most of the eigenvalues appear to have settled to steady state values although evidence of the long term spiral decay shows through in some modes.

In order to decrease the succeptability of the eigenvalues of this system to changes in airspeed and to provide a given modal distribution, some form of closed-loop control would normally be employed. It would also be expected that this action would reduce the quite clear longitudinal/lateral cross-coupling. Because the system parameters change so radically with airspeed it is unrealistic to expect a fixed control structure to provide completely insensitive eigenvalue assignment, some variation will thus still be evident. To provide some idea of the insensitivity properties of fixed closed-loop state feedback the following section examines how well the previously designed optimal control state feedback law preserves the

system eigenstructure with changing airspeed.

8.2.2 Closed-Loop Aircraft Eigenvalue Sensitivity

Both LQP and VSS controllers have previously been derived for the lateral motion model of the Machan aircraft. State feedback should reduce the sensitivity of the system to changes in aircraft parameters as a result of changing flight configuration in addition to providing a prescribed eigenstructure. In the open-loop case it has already been demonstrated that the aircraft parameters vary quite widely with airspeed and one is now prompted to examine the degree to which fixed state feedback can minimise the resulting changes in the closed-loop eigenstructure. Recall that it is possible using the non-linear simulation to evaluate the locally linearised 'A' and 'B' state matrices at a number of time points over a prescribed aircraft manoeuvre and hence derive the open-loop eigenstructure. For the closed-loop state feedback case it is simply required to evaluate the eigenstructure of the equivalent closed-loop 'A' matrix i.e. $(A_{la} + B_{la} K_{la})$ with K_{la} being the nominal state feedback gain matrix. It is clearly possible to do this and, following the procedure adopted in section 8.2.1, produce the eigenvalue variations over the manoeuvre in graphical form. It is not possible to use this approach for the VSS controller since the effective state feedback gains change as a result of variations in the states and hence an equivalent K_{la} matrix is not available. For the moment, the discussion is confined to the LQP state feedback control law and its likely performance.

The results of Figs. 8.1 are repeated, for the closed-loop case, in Figs. 8.2 a) to f). Note that the state feedback gains are those derived via. the eigenvalue/eigenvector assignment technique of section 6.5. For this design recall that the closed-loop poles were disposed at :

$$-0.81 \pm 2.72j \; ; \; -4.39 \; ; \; -0.1$$

for the nominal 33 ms^{-1} 'stick fixed' linear model.

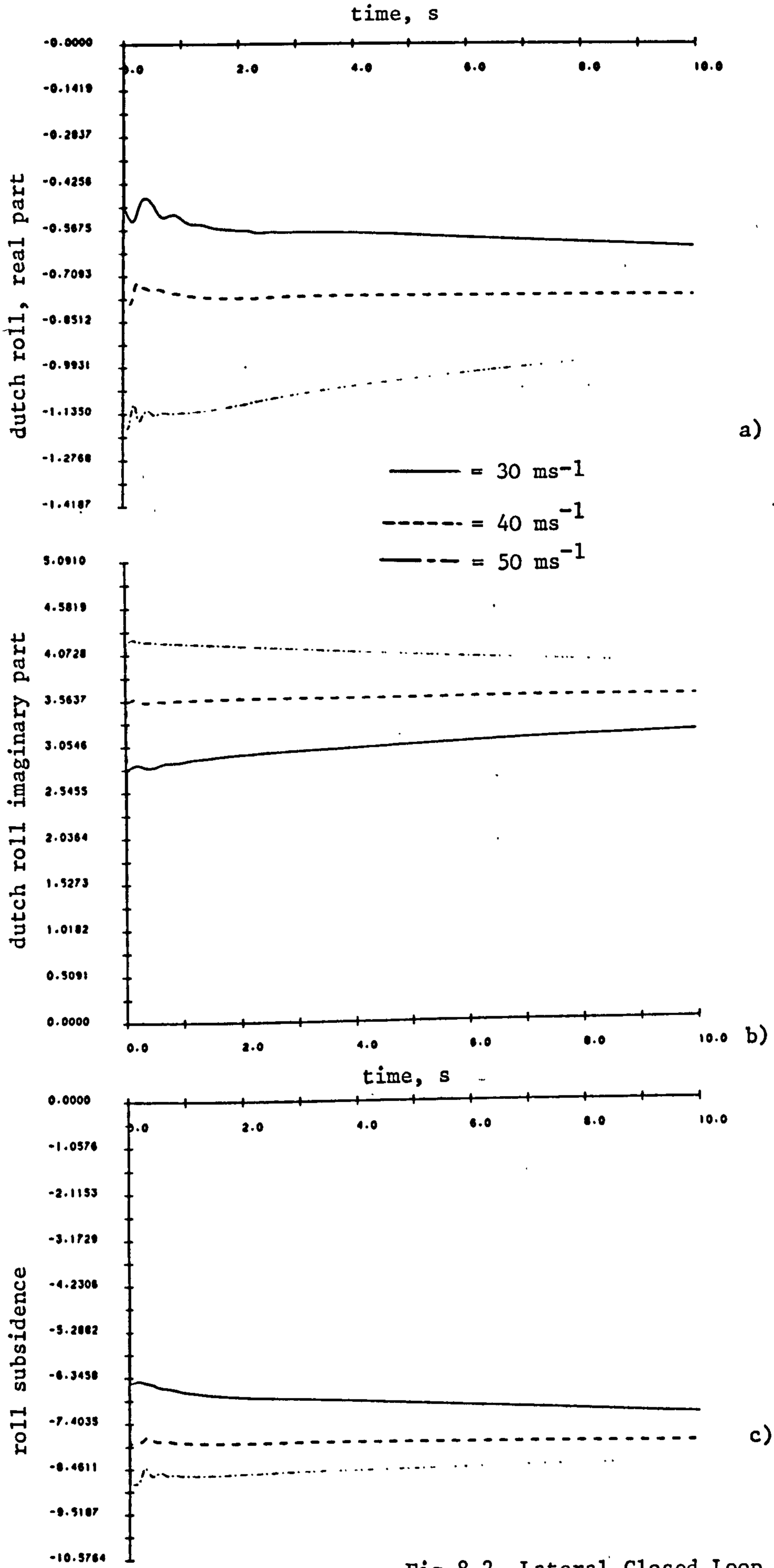
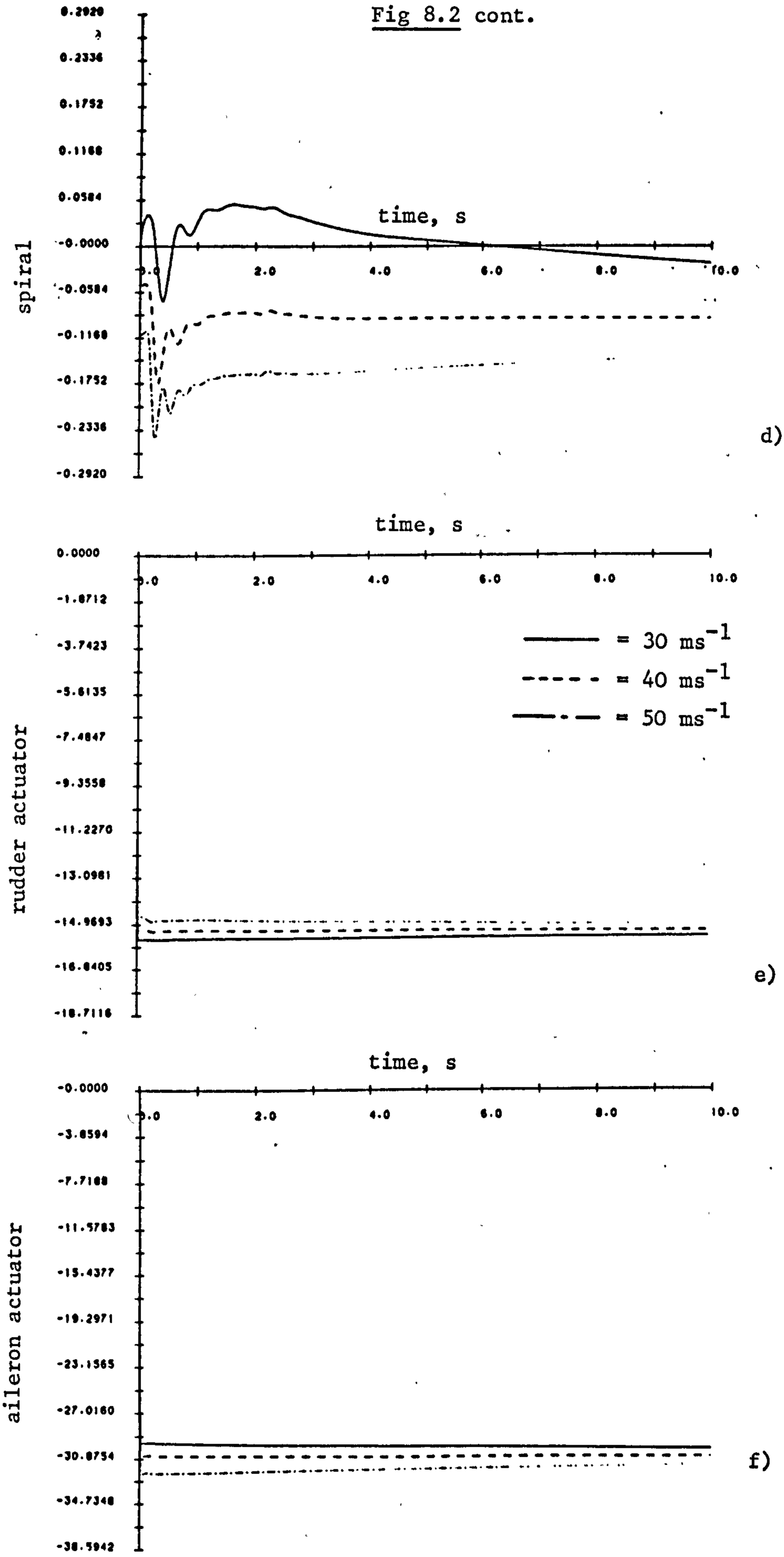


Fig 8.2 Lateral Closed Loop Eigenvalue Sensitivity (LQP control)

Fig 8.2 cont.


spiral
0.2920
0.2336
0.1752
0.1168
0.0584
-0.0000
-0.0584
-0.1168
-0.1752
-0.2336
-0.2920
time, s
0.0
2.0
4.0
6.0
8.0
10.0
d)
rudder actuator
0.0000
-1.8712
-3.7423
-5.6135
-7.4847
-9.3558
-11.2270
-13.0981
-14.9693
-16.8405
-18.7116
time, s
0.0
2.0
4.0
6.0
8.0
10.0
e)
aileron actuator
-0.0000
-3.8594
-7.7188
-11.5783
-15.4377
-19.2971
-23.1565
-27.0160
-30.8754
-34.7348
-38.5942
time, s
0.0
2.0
4.0
6.0
8.0
10.0
f)
——— = 30 ms^{-1}
- - - - = 40 ms^{-1}
——·—— = 50 ms^{-1}

Considering each mode separately, as before, it should be noted that :

a) The Dutch roll mode, Figs. 8.2 a) and b), again varies over the three airspeeds but the variations in the imaginary part are reduced with respect to Fig. 8.1 b). The degree of pitch cross-coupling is also reduced, as evidenced by the smaller initial transients in the real part of this mode, Fig. 8.2 a). The mode has values of $-0.64 \pm 3.2j$ ($\zeta=0.2$, $\omega_n=0.52$Hz) and $-0.94 \pm 4.0j$ ($\zeta=0.23$, $\omega_n=0.65$Hz) after ten seconds of simulation for the 30 and 50 ms^{-1} airspeeds respectively. This represents a 15% increase in damping and 25% increase in frequency for this mode. Note that in both cases the eigenvalues are fairly close to, but somewhat above, the achievable $-0.81 \pm 2.42j$ ($\zeta=0.32$, $\omega_n=0.41$Hz) eigenvalue.

b) The roll subsidence mode's, Fig. 8.2 c), variation with airspeed is now reduced although slightly more pitch cross-coupling is evident at higher airspeeds. After ten seconds of simulation the mode has values of -7.14 and -8.25 for the 33 and 50 ms^{-1} airspeeds respectively, this representing an increase of 15.5%. The mode is, however, somewhat faster than desired.

c) The spiral mode, Fig. 8.2 d), is somewhat improved over the open-loop case since, for the most part, it is rendered stable. In the 30 ms^{-1} airspeed case, however, note that the mote is initially slightly unstable although after six seconds it becomes stable but is relatively slow, even after ten seconds of simulation. Pitch cross-coupling is still clearly evident in this mode (c.f. Fig. 8.1 d)). After ten seconds of simulation the mode has values of -0.1 and -0.146 for the 40 and 50 ms^{-1} airspeeds, this representing an increase of 46% in response speed. This mode is clearly rather poorly behaved and could present something of a problem to control at lower airspeeds. At higher airspeeds, however, the mode has approximately the desired closed-loop value.

The final modes are the two actuators and the yaw integration. The two actuator modes are shown in Figs. 8.2 e) and f) and are generally rather faster than desired and show very little variation over the three airspeeds. The rudder actuator is nominally around -15.0 s^{-1} whilst the aileron actuator is around -31.0 s^{-1}, these being somewhat faster than the desired -13.0 and -21.0 s^{-1}. The yaw integration is left open-loop and hence remains an integration over all flight conditions.

The actual time histories of the lateral states and controls for the non-linear model are shown in Fig. 8.3 a) to i) for the particular manoeuvre and for the three airspeeds chosen. These support the general conclusions drawn above principally the increase in Dutch roll frequency and damping at higher airspeeds and the relatively small variations in the roll subsidence mode (p response). The behaviour of the spiral mode (φ response) also demonstrates that this is a relatively slow mode at low arispeeds whilst its stability and response speed is improved for higher airspeeds.

Also of interest is the degree to which the eigenvectors of the closed-loop system vary over this typical manoeuvre. The state feedback structure used should provide prescribed eigenvector assignment and this should remain insensitive to changes in airspeed, etc. The eigenvectors of interest are those associated with the v, p, r and φ states since it is these which were considered in the LQP eigenvector assignment. For the 30 ms^{-1} airspeed condition, which was chosen as the nominal case, the closed-loop eigenvectors are given in Table 8.2 for the manoeuvre considered above at 2.0, 5.0 and 7.0 seconds into the manoeuvre. These eigenvectors can be compared with those obtained for the linear design in section 6.5.1. Overall, the eigenvectors provide close to the desired modal decoupling although the coupling from the imaginary part of the Dutch roll into roll rate is larger than desired. The time histories of Fig. 8.3 also indicate that the prescribed modal coupling has been closely achieved. Over the manoeuvre chosen the eigenstructure is preserved quite well,

Fig 8.3 Lateral Manoeuvre Time Histories

(LQP controller)

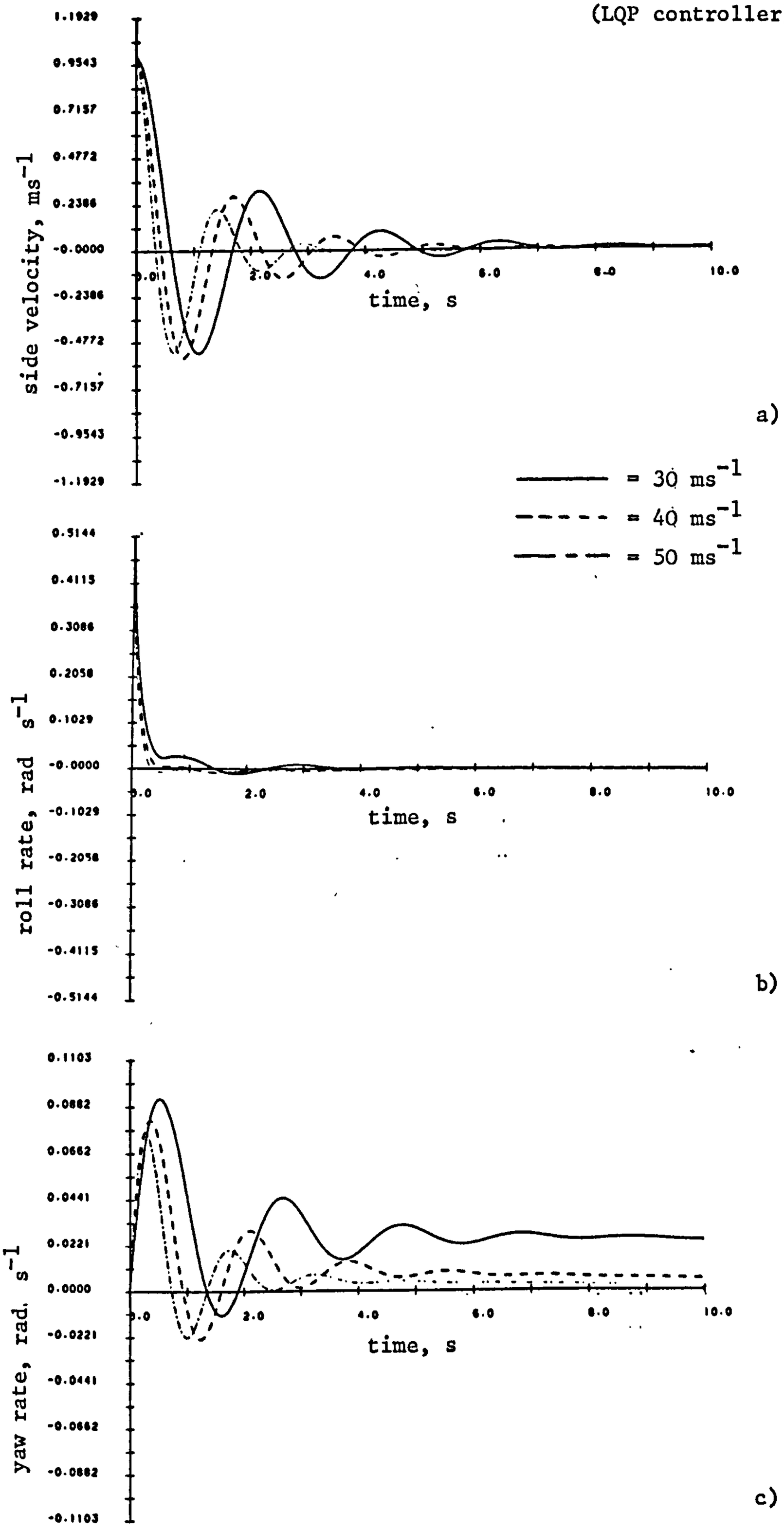
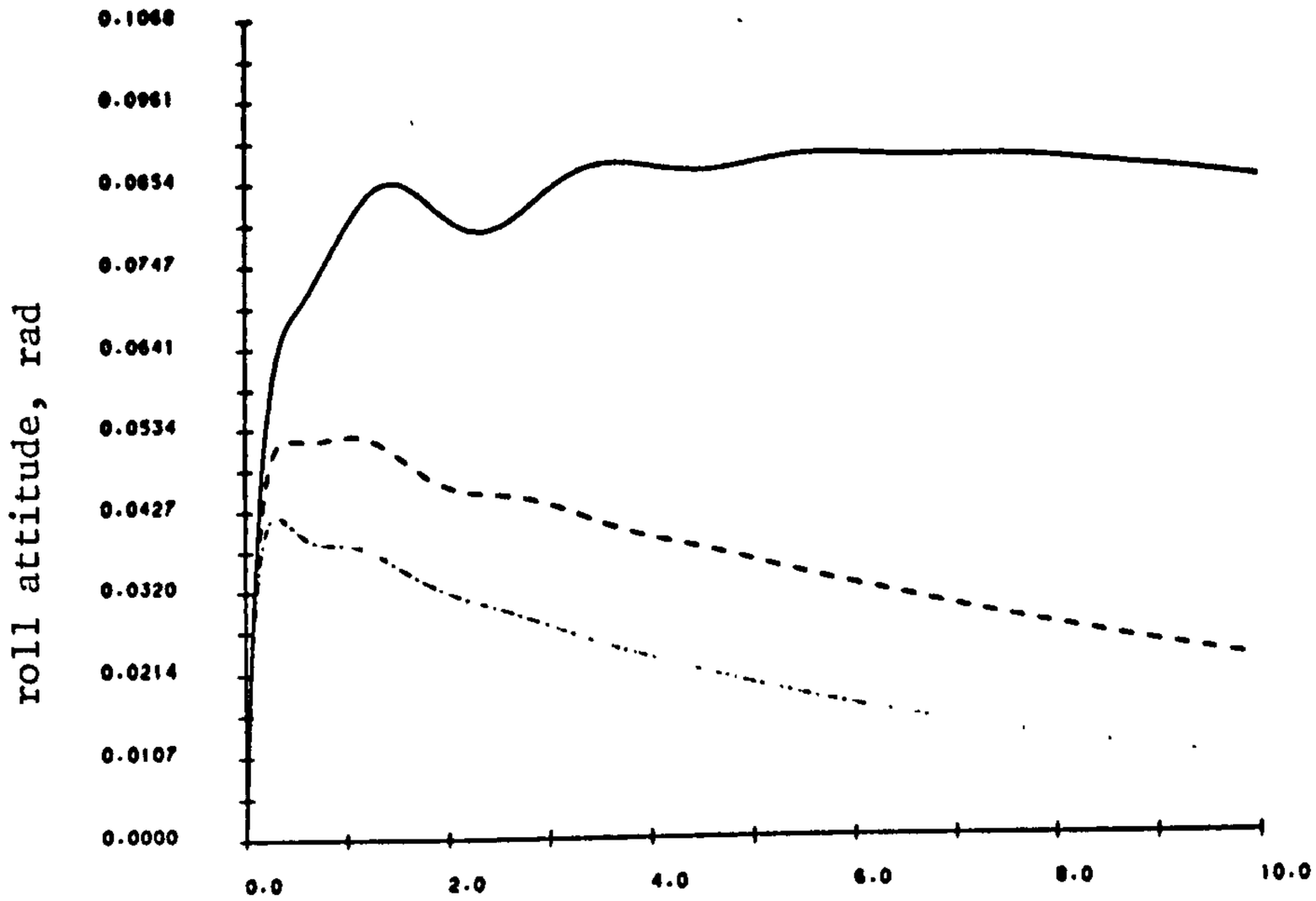
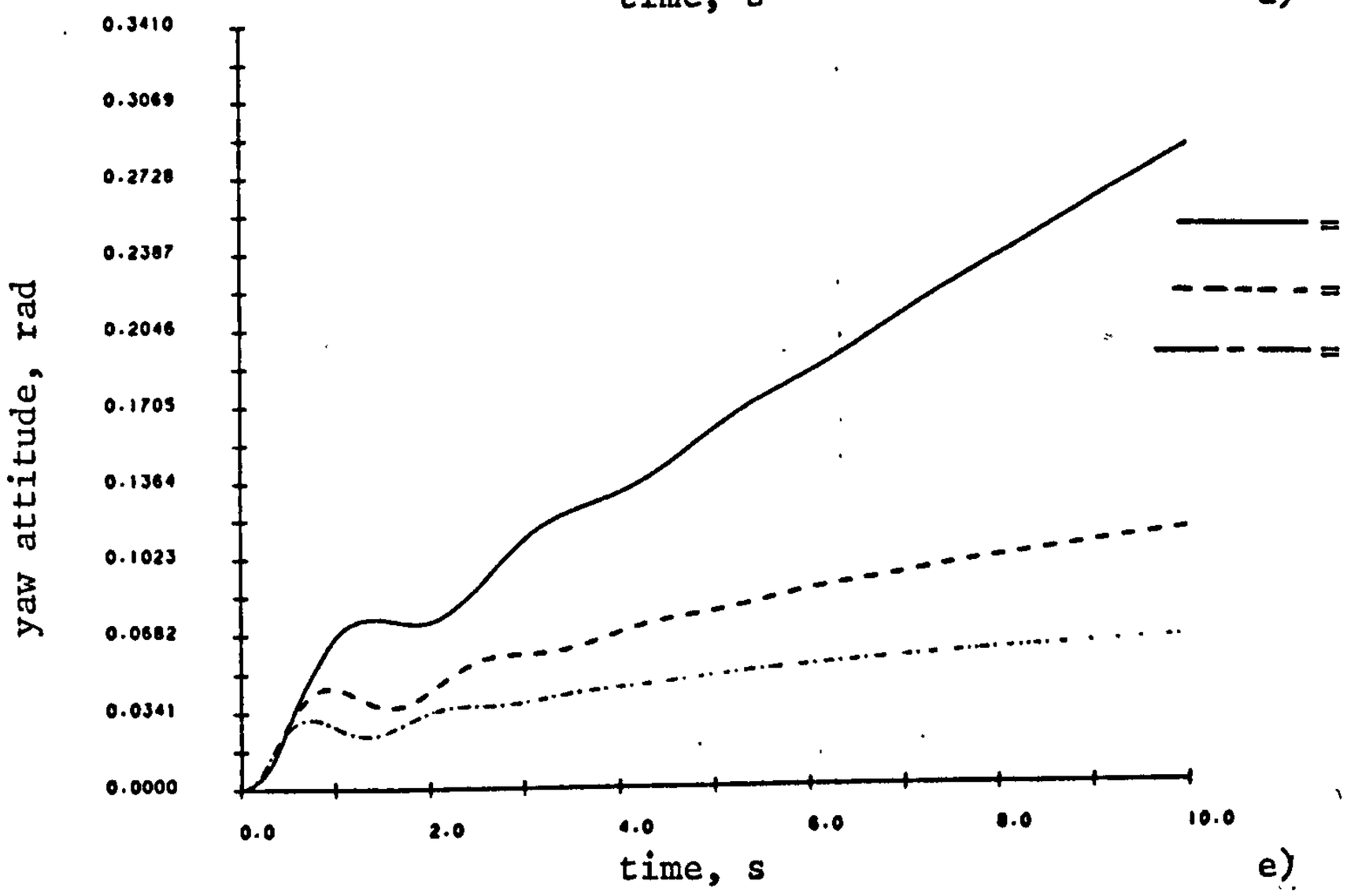


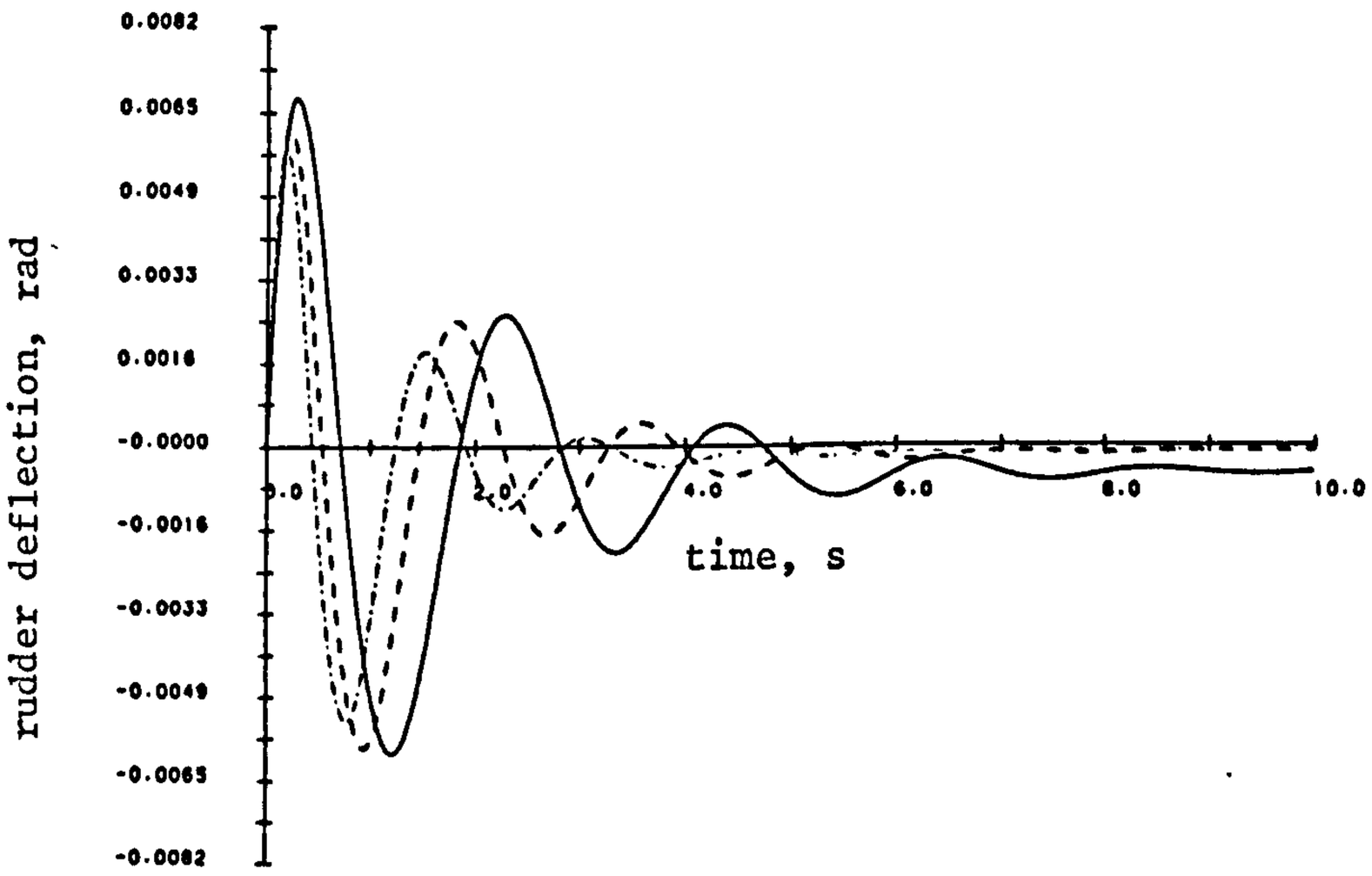
Fig 8.3 cont.



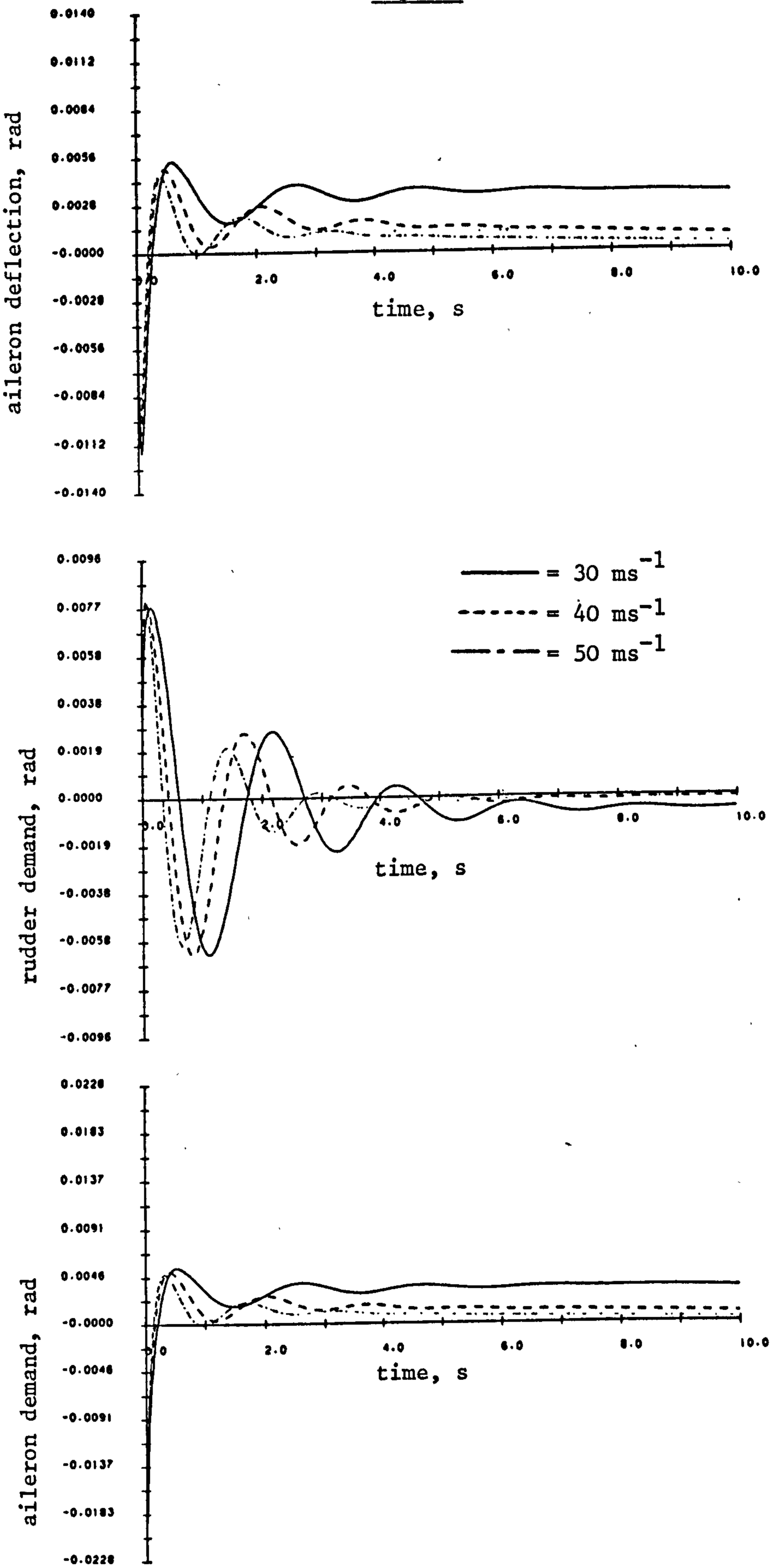
d)



e)



f)


aileron deflection, rad
0.0140
0.0112
0.0084
0.0056
0.0028
-0.0000
-0.0028
-0.0056
-0.0084
-0.0112
-0.0140
0
2.0
4.0
6.0
8.0
10.0
time, s
g)
rudder demand, rad
0.0096
0.0077
0.0058
0.0038
0.0019
0.0000
-0.0019
-0.0038
-0.0058
-0.0077
-0.0096
= 30 ms-1
= 40 ms-1
= 50 ms-1
time, s
h)
aileron demand, rad
0.0228
0.0183
0.0137
0.0091
0.0046
-0.0000
-0.0046
-0.0091
-0.0137
-0.0183
-0.0228
time, s
i)

the φ eigenvector, governing the spiral mode distribution, probably undergoing the largest variations in its coupling into the remaining states.

The eigenstructure also varies with the three airspeeds considered and Table 8.3 details the degree of misalignment of the eigenvectors from the 30 ms^{-1} airspeed case for the 40 and 50 ms^{-1} airspeed cases. The misalignment angles were evaluated by treating the 30 ms^{-1} eigenvectors as the basis set and using the relationship :

$$\cos\theta = \frac{| e_i, w_i |}{|| e_i ||}$$

with $\underline{e}_i$ the base eigenvector and $\underline{w}_i$ the eigenvector of interest. Note from Table 8.3 that all of the misalignment angles are small indicating that the eigenstructure is preserved quite well. The worst offender in this respect being the spiral mode and the imaginary part of the Dutch roll mode. This result is supported by the time histories of Fig. 8.3 which show little variation in modal coupling with airspeed and attitude variations.

It is evident from the preceeding discussion that the eigenstructure robustness properties of the system have been improved by the inclusion of the fixed optimal state feedback law. Both the eigenvalues and eigenvectors of the system change by a smaller amount in the closed-loop case as compared with the open-loop case and the system's time response is maintained tolerably well over the three airspeeds considered. Far better results can be achieved by employing a gain-scheduling philosophy which changed the state feedback gains with changing airspeed, for example, but this is arguably more difficult than using simple classical gain-scheduling schemes.

8.2.3 Robustness of VSS Controller

As mentioned above it is impossible to determine quantitatively the eigenstructure of the VSS lateral motion controller using the approach outlined above. This is due to the discontinuous nature of the control and the

Table 8.2

Lateral Eigenvectors for 33 ms^{-1} airspeed

Time = 2.0 secs

p e_1 = [0.0218 0.9865 -0.0358 -0.1448 -0.0231 -0.0594]

v{Re e_2 = [0.9946 -0.0252 0.0047 -0.0098 0.0102 0.0006]

r{Im e_3 = [0.0 -0.3514 -0.9284 0.1091 -0.0419 -0.0953]

ϕ e_4 = [0.1866 0.042 0.2872 0.9379 -0.0051 0.0363]

Time = 5.0 secs

p e_1 = [0.016 0.9871 -0.0346 -0.1418 -0.0237 -0.0596]

v{Re e_2 = [0.995 -0.0214 0.0055 -0.0068 0.0101 0.0006]

r{Im e_3 = [0.0 -0.2639 -0.9585 0.0898 -0.0454 0.0924]

ϕ e_4 = [0.1163 0.0035 0.2819 0.9515 -0.0071 0.0284]

Time = 7.0 secs

p e_1 = [0.006 0.9874 -0.036 -0.1396 -0.0244 -0.069]

v{Re e_2 = [0.9951 -0.0207 0.0059 -0.006 0.01 0.0005]

r{Im e_3 = [0.0 -0.2464 -0.9636 0.085 -0.0457 0.0917]

ϕ e_4 = [-0.0904 0.0 -0.2777 -0.9556 0.0 0.0155]

TABLE 8.3

Lateral Eigenvector Misalignment Angles (degrees)

TIME secs.	2.0				5.0				7.0			
Airspeed (m/s)	e_1	e_2	e_3	e_4	e_1	e_2	e_3	e_4	e_1	e_2	e_3	e_4
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40	6.8	5.5	10.1	12.4	6.0	5.4	0.0	9.4	5.4	5.4	13.4	10.1
50	15.5	13.6	18.1	20.2	16.3	14.1	13.0	15.8	16.9	5.3	12.5	14.1

consequent lack of a nominal set of state feedback gains. Perhaps the most instructive assessment of robustness for the VSS controller is by way of a comparison with the LQP design already considered. Again, however, a problem is encountered here in as much as the insensitivity properties of VSS are only guaranteed when sliding, the initial reaching dynamics will, inevitably, also be manifested in the time responses. It would therefore not be expected that the system time responses with VSS and LQP should be identical, nevertheless it may be instructive to consider such a comparison.

Consider the lateral manoeuvre chosen above at the three airspeeds 30, 40 and 50 ms^{-1} with the non-linear simulation. A comparison of the state responses after a ten second simulation, for the LQP and VSS controllers are given in Figs. 8.4, 8.5 and 8.6 for the 30, 40 and 50 ms^{-1} airspeeds respectively. Considering first the 30 ms^{-1} case, Fig. 8.4, note that the responses are broadly similar. The Dutch roll mode is however somewhat less well damped with the VSS controller in addition to having a slightly shorter period. Recall that the LQP design provided close to the desired response in Dutch roll and hence it may concluded that the VSS controller provides a slightly poorer Dutch roll performance in terms of damping and frequency. The roll response for the VSS controller is very similar to that achieved by the LQP design, note however that some Dutch roll coupling into roll is apparent. The VSS controller also appears to demand more of the aileron than the LQP design whilst requiring less of the rudder. These trends in the VSS control are also reflected in the responses at 40 ms^{-1} and 50 ms^{-1}, Figs. 8.5 and 8.6, with reduced Dutch roll damping and period and increased Dutch roll coupling into roll. Over the three airspeeds then the VSS controller's response varies quite widely. This is demonstrated in Fig. 8.7 which shows the state responses of the system under VSS control over the three airspeeds. A straightforward comparison of Figs. 8.7 and 8.3, giving the corresponding state responses of the LQP controller, perhaps indicates that the LQP design may have the better robustness properties in terms of preserved Dutch roll, spiral and roll

Fig 8.4 State Responses for Non-linear Lateral Model

(33 ms^{-1} airspeed, VSS vs. LQP control)

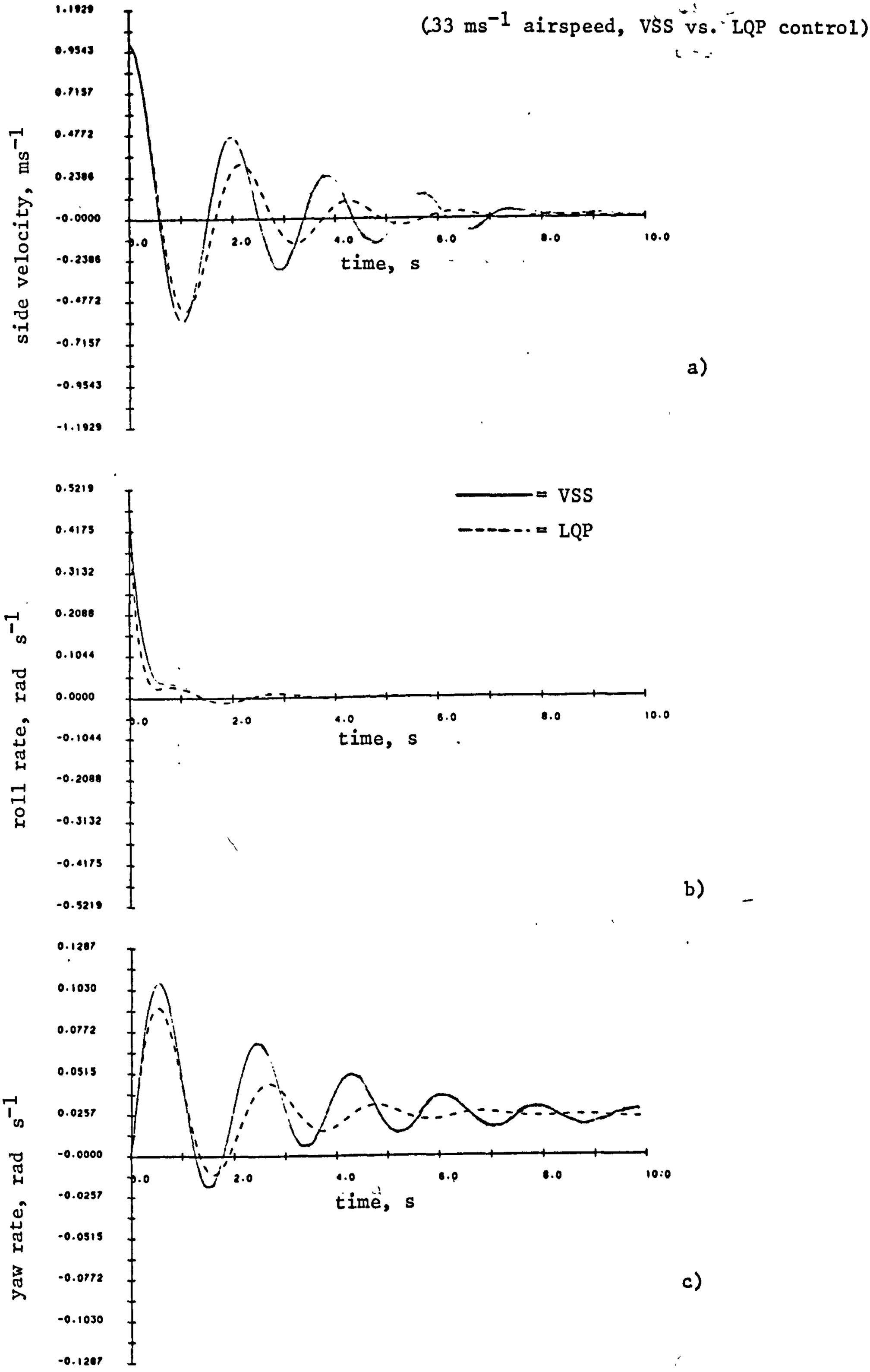


Fig 8.4 cont.

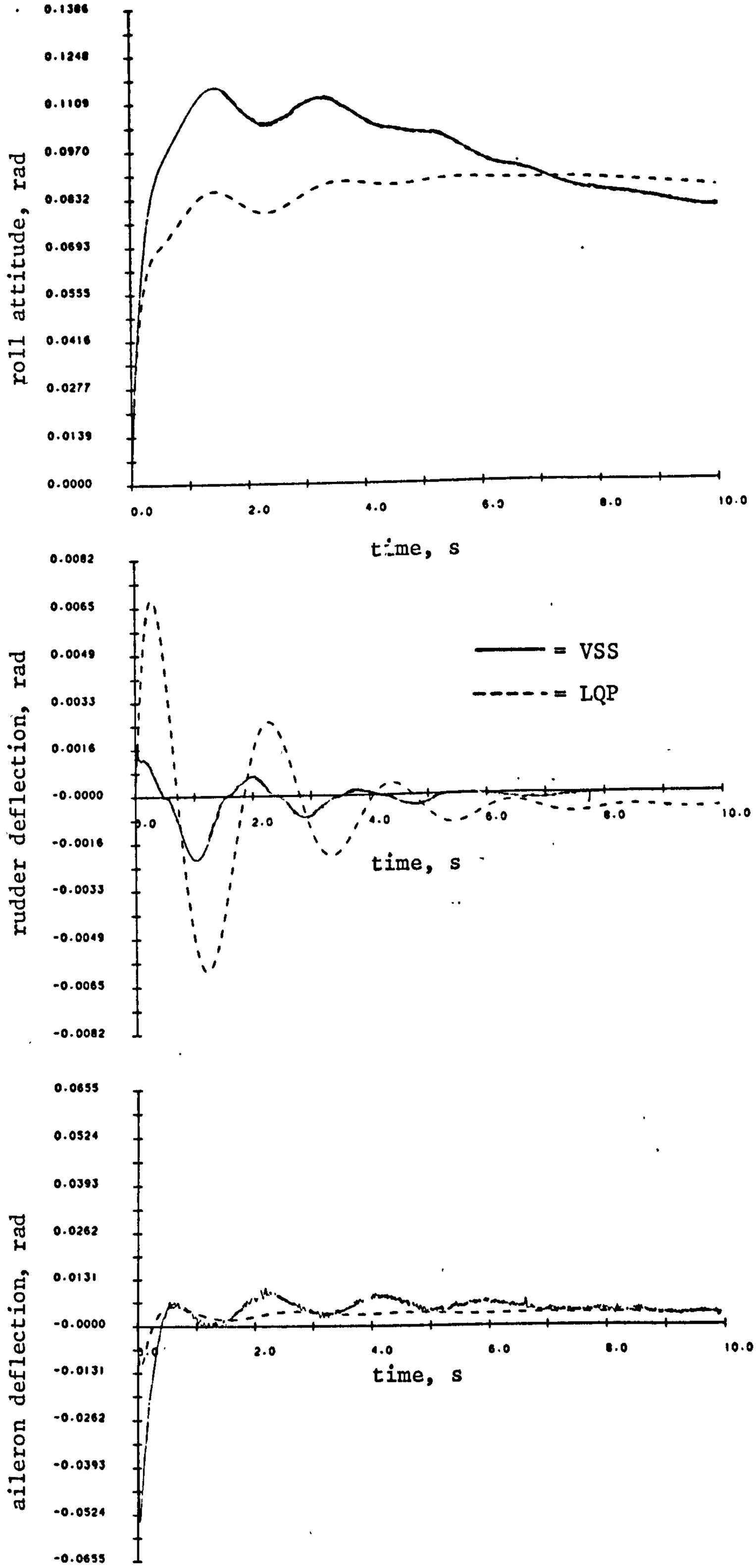
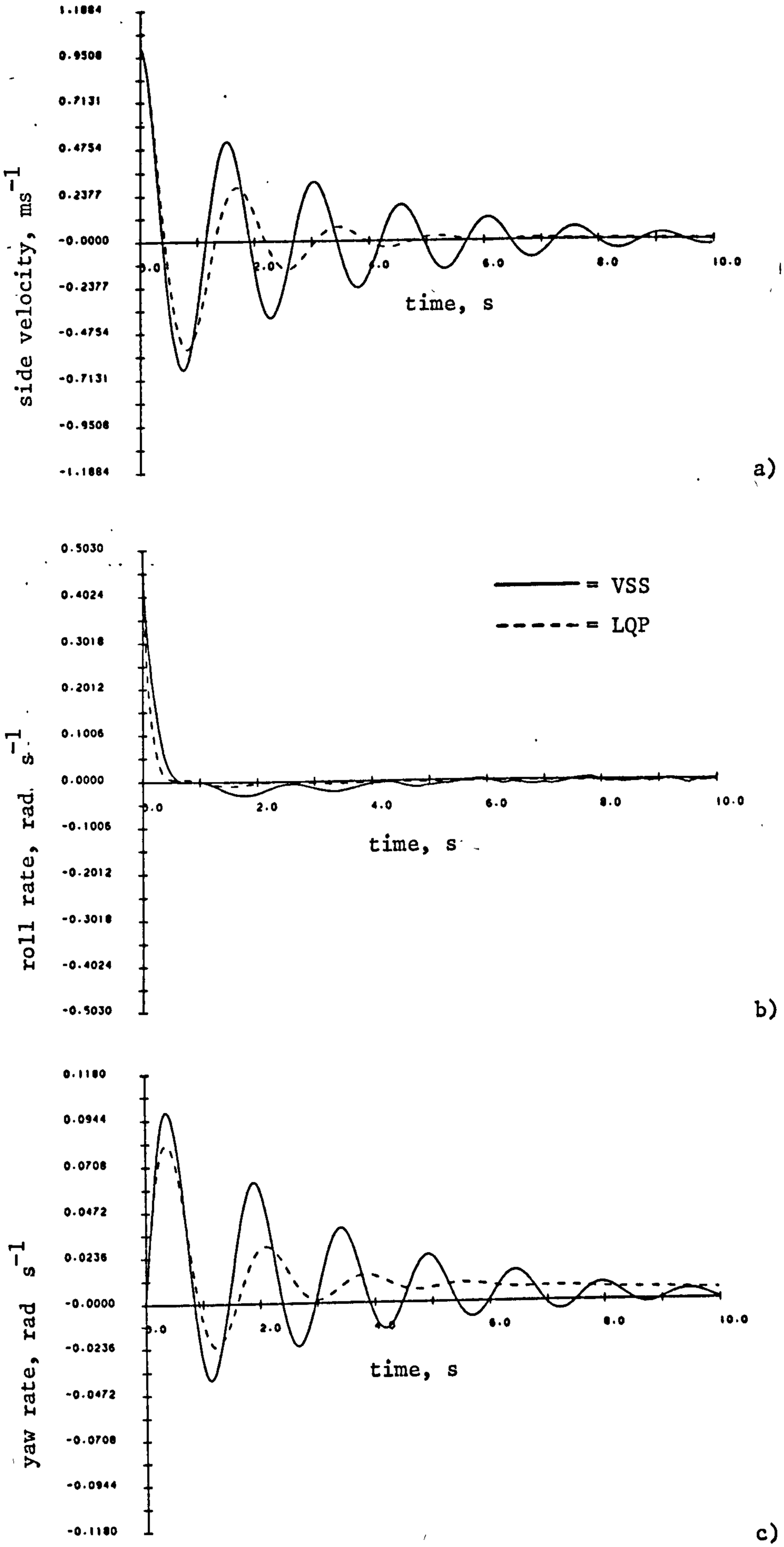

roll attitude, rad
0.1386
0.1248
0.1109
0.0970
0.0832
0.0693
0.0555
0.0416
0.0277
0.0139
0.0000
0.0
2.0
4.0
6.0
8.0
10.0
time, s
d)
rudder deflection, rad
0.0082
0.0065
0.0049
0.0033
0.0016
-0.0000
-0.0016
-0.0033
-0.0049
-0.0065
-0.0082
= VSS
= LQP
time, s
e)
aileron deflection, rad
0.0655
0.0524
0.0393
0.0262
0.0131
-0.0000
-0.0131
-0.0262
-0.0393
-0.0524
-0.0655
time, s
f)

Fig 8.5 As Fig 8.4 for 40 ms^{-1} airspeed



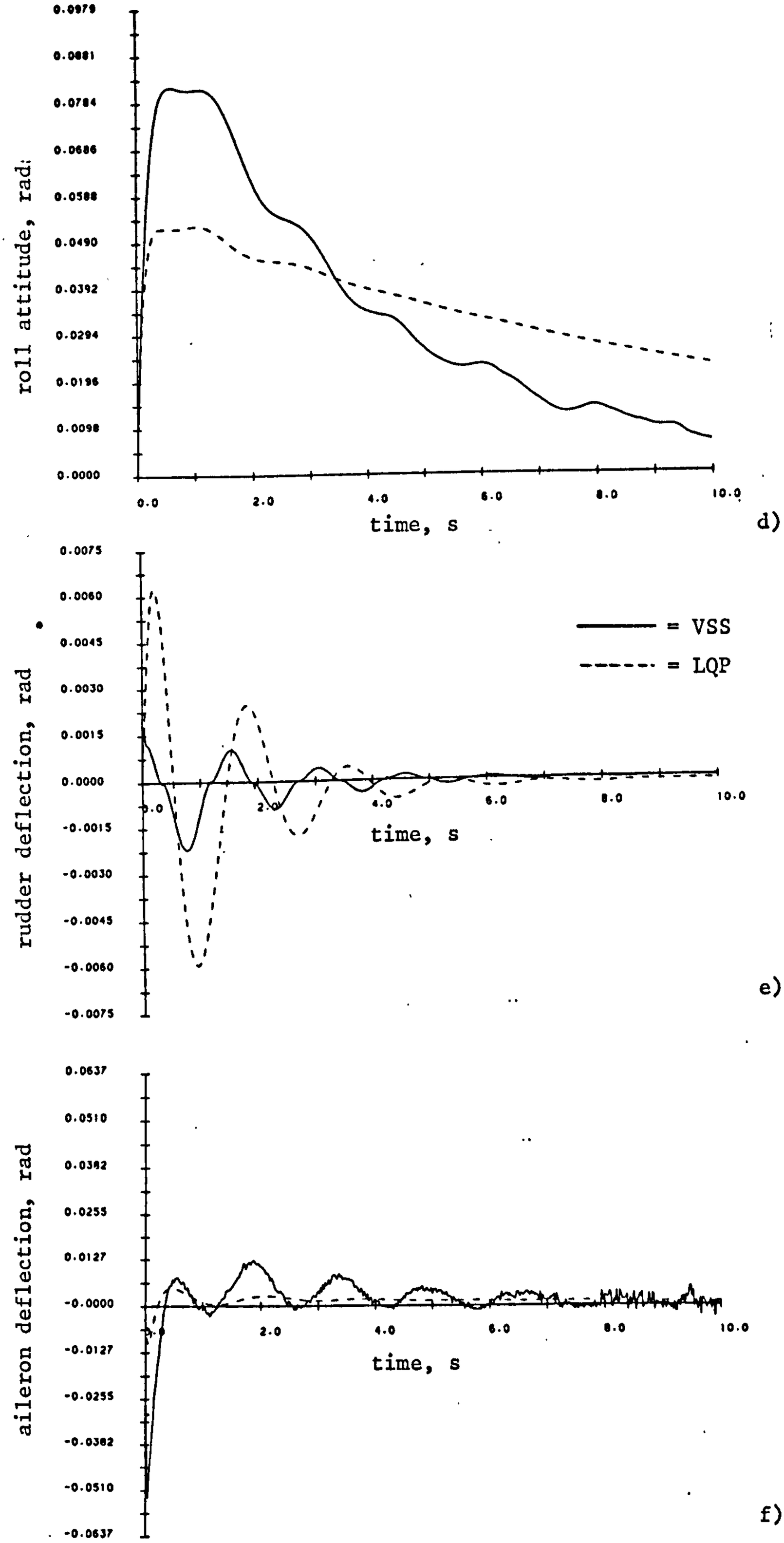
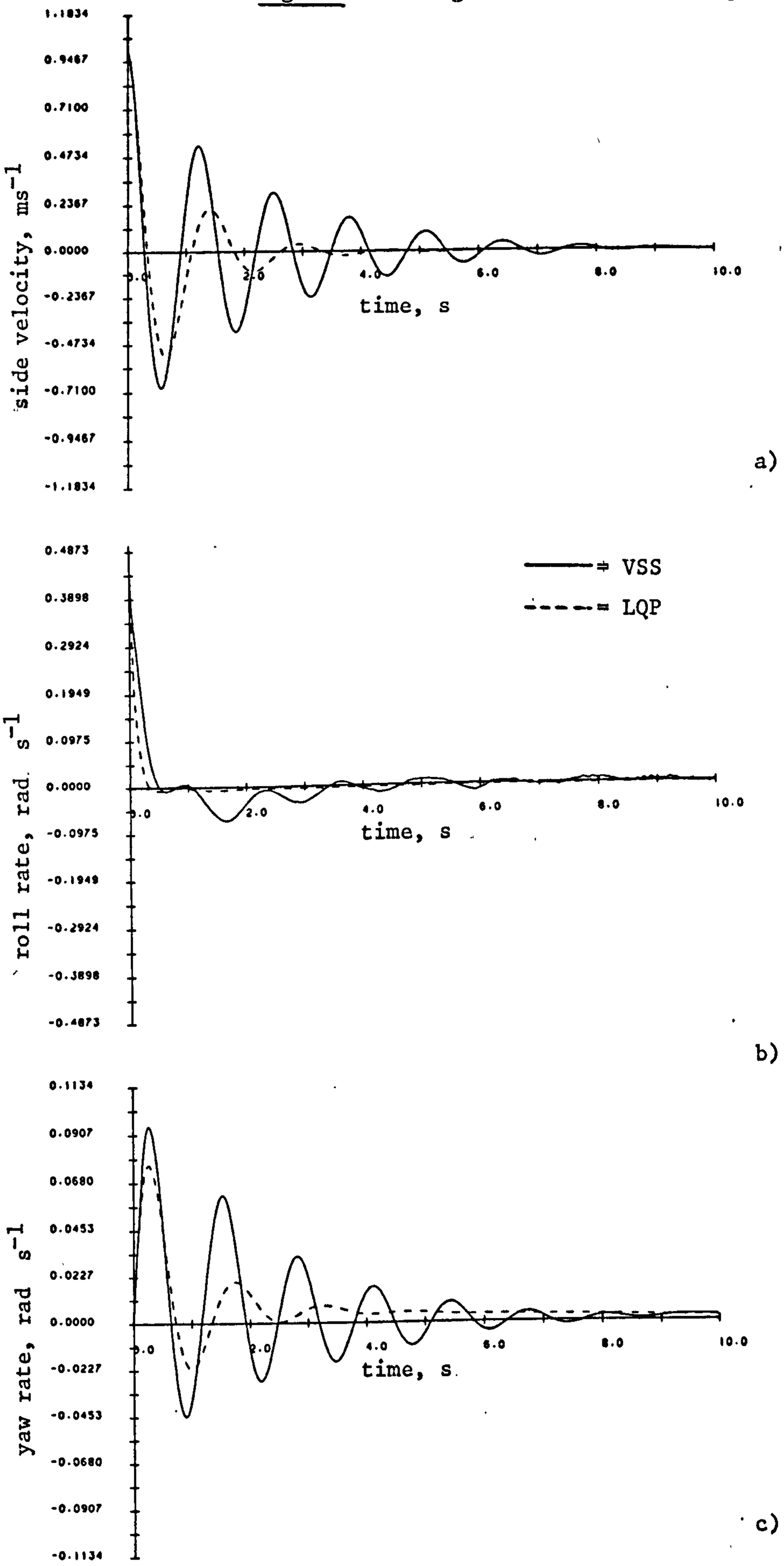
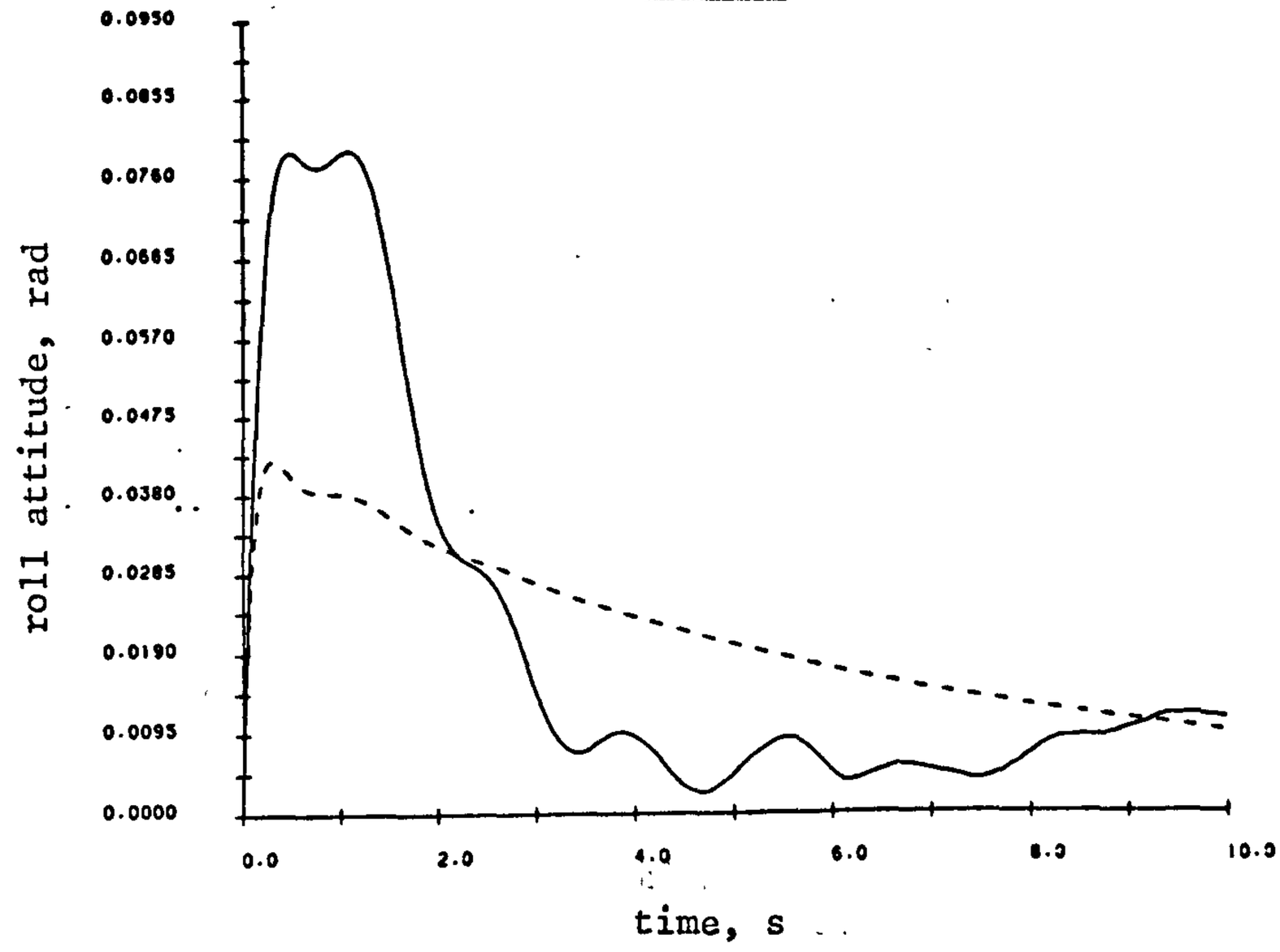
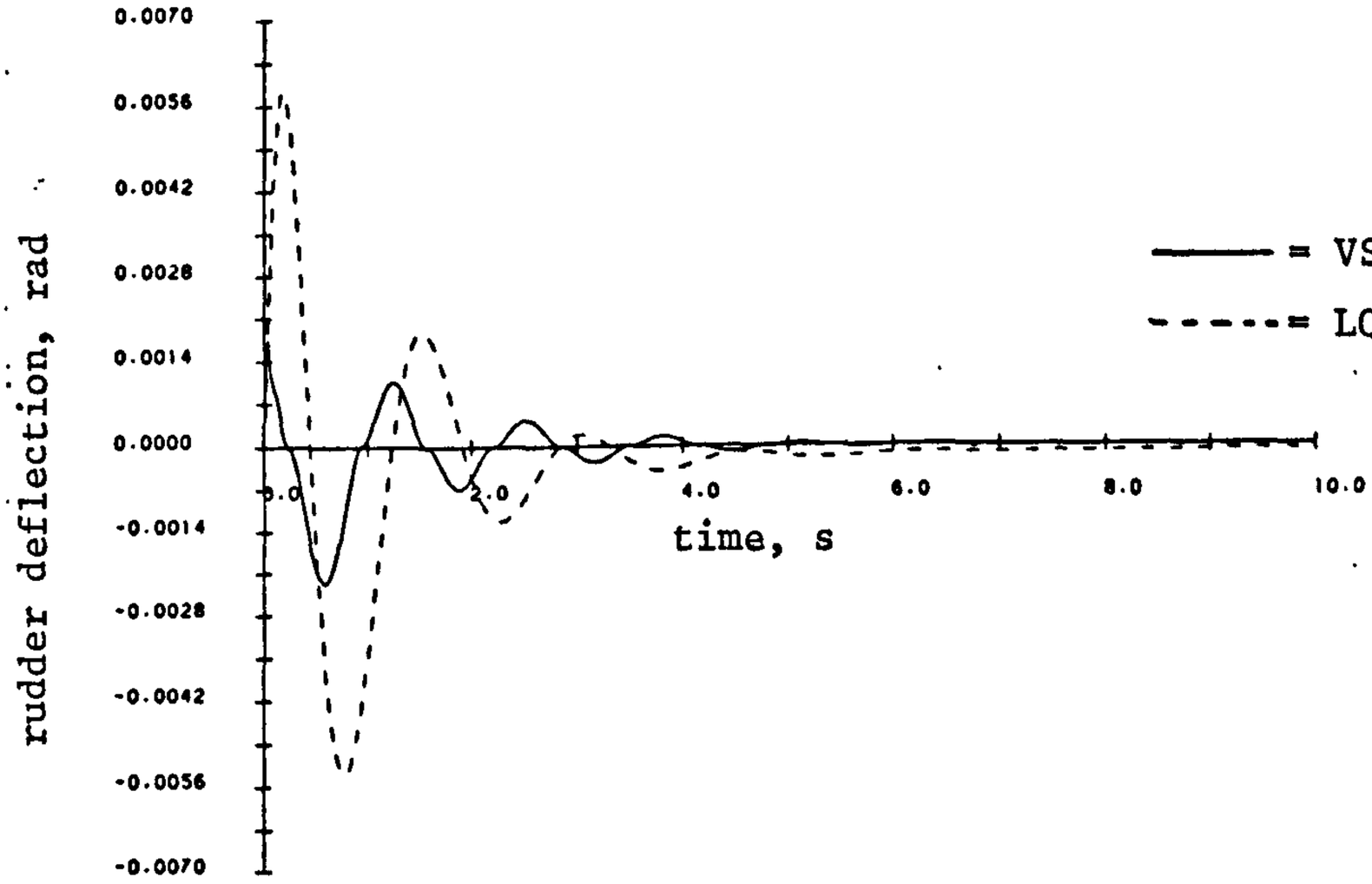

roll attitude, rad
0.0979
0.0881
0.0784
0.0686
0.0588
0.0490
0.0392
0.0294
0.0196
0.0098
0.0000
0.0
2.0
4.0
6.0
8.0
10.0
time, s
d)
rudder deflection, rad
0.0075
0.0060
0.0045
0.0030
0.0015
0.0000
-0.0015
-0.0030
-0.0045
-0.0060
-0.0075
0.0
2.0
4.0
6.0
8.0
10.0
time, s
e)
aileron deflection, rad
0.0637
0.0510
0.0382
0.0255
0.0127
-0.0000
-0.0127
-0.0255
-0.0382
-0.0510
-0.0637
2.0
4.0
6.0
8.0
10.0
time, s
f)
= VSS
= LQP

Fig 8.6· As Fig 8.4 for 50 ms^{-1} airspeed

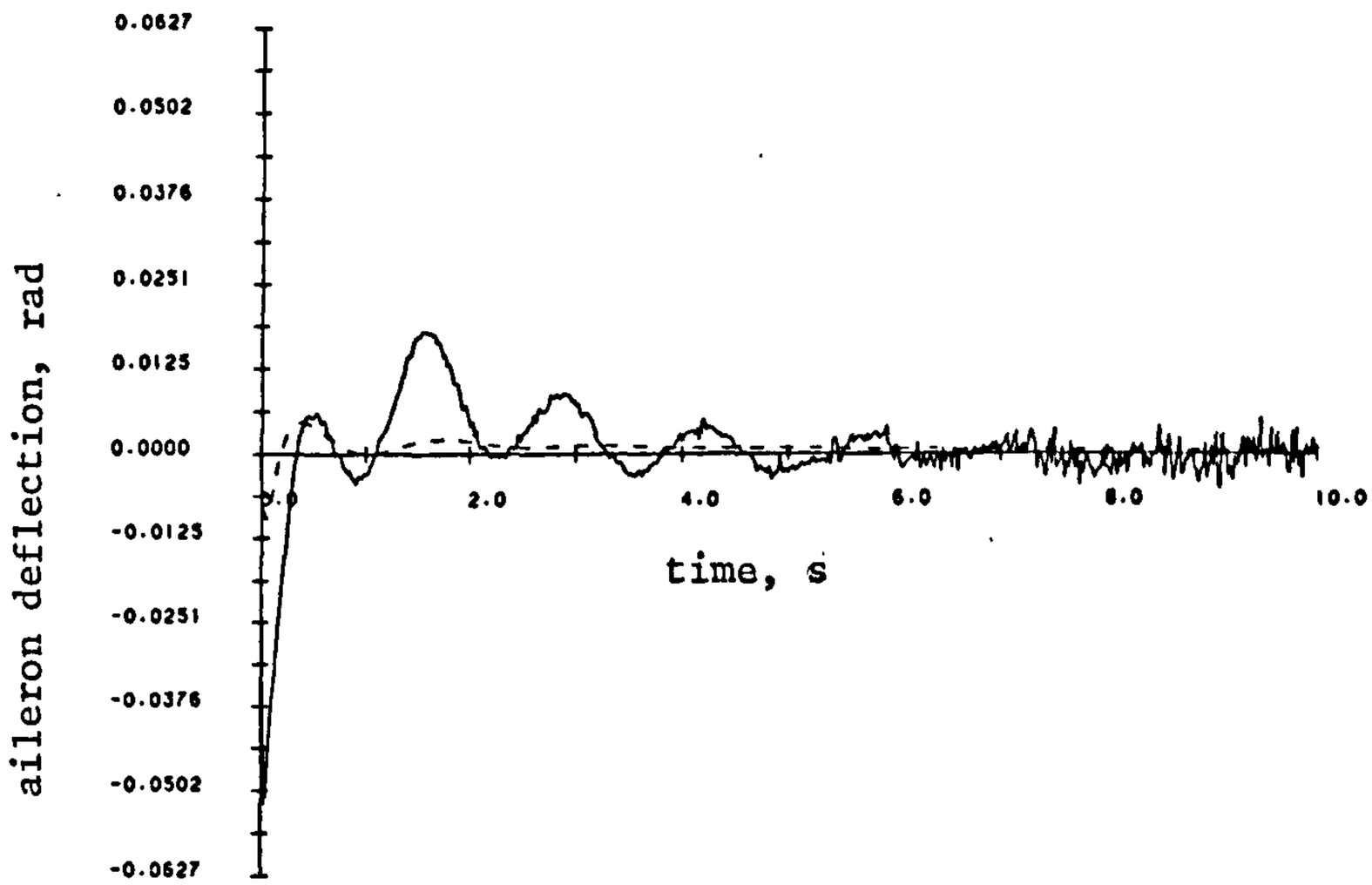



roll attitude, rad
0.0950
0.0855
0.0760
0.0665
0.0570
0.0475
0.0380
0.0285
0.0190
0.0095
0.0000
0.0
2.0
4.0
6.0
8.0
10.0
time, s

d)


rudder deflection, rad
0.0070
0.0056
0.0042
0.0028
0.0014
0.0000
-0.0014
-0.0028
-0.0042
-0.0056
-0.0070
0.0
2.0
4.0
6.0
8.0
10.0
time, s
= VSS
= LQP

e)


aileron deflection, rad
0.0627
0.0502
0.0376
0.0251
0.0125
0.0000
-0.0125
-0.0251
-0.0376
-0.0502
-0.0627
0.0
2.0
4.0
6.0
8.0
10.0
time, s

f)

Fig 8.7 Time Histories for Non-linear Lateral Model (VSS controller)

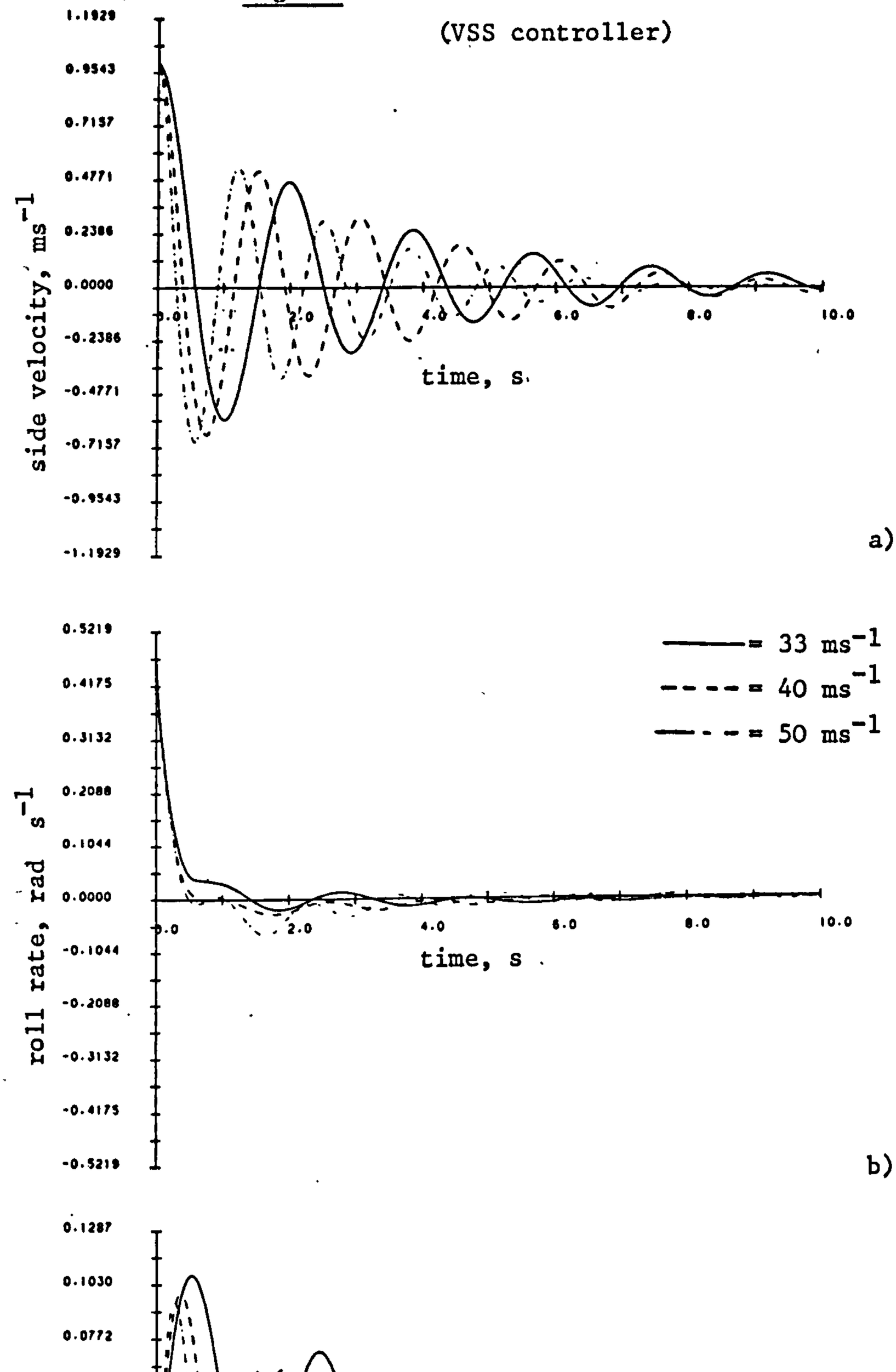
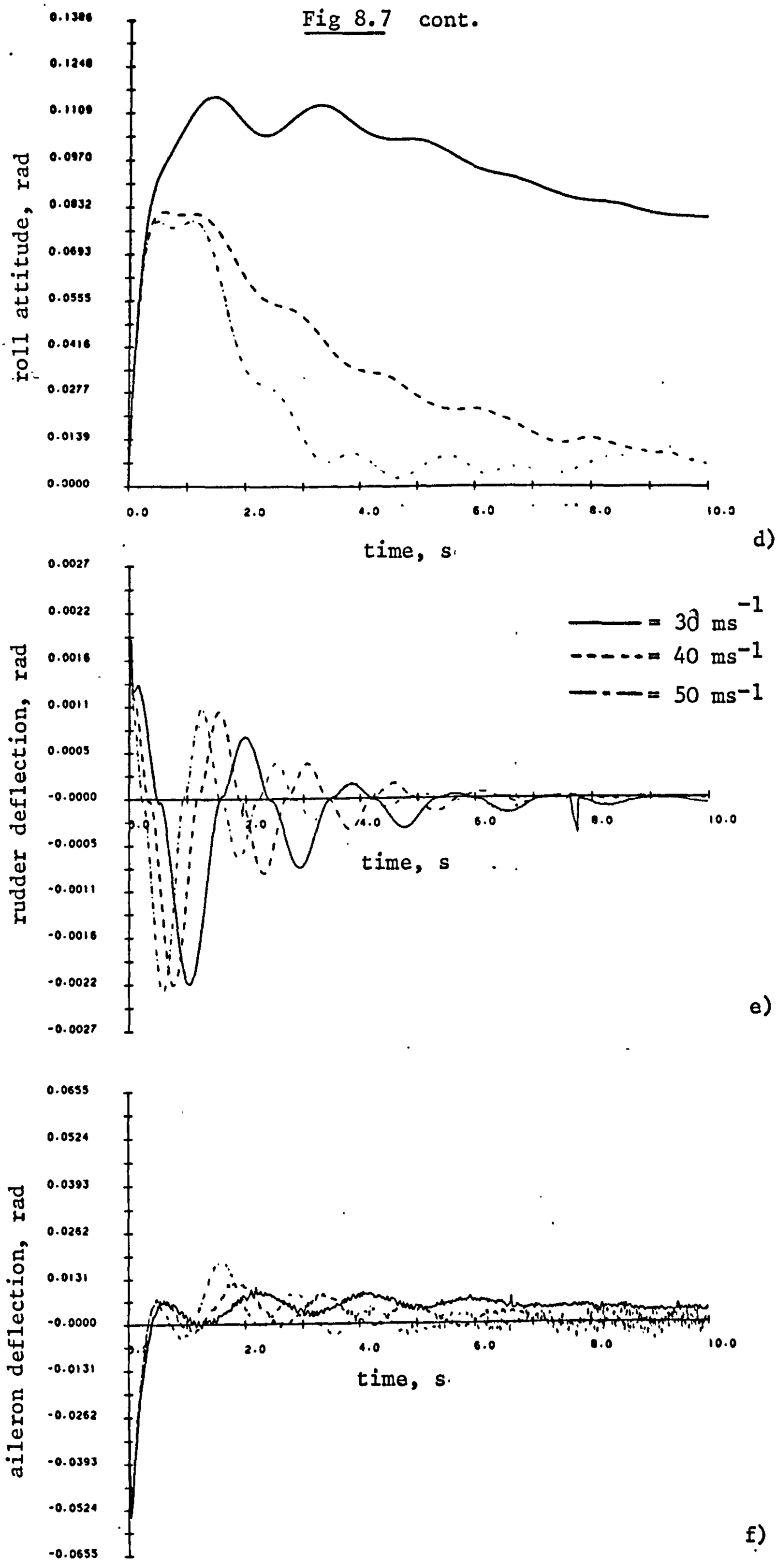


Fig 8.7 cont.


roll attitude, rad
0.1386
0.1248
0.1109
0.0970
0.0832
0.0693
0.0555
0.0416
0.0277
0.0139
0.0000
0.0
2.0
4.0
6.0
8.0
10.0
time, s
d)
rudder deflection, rad
0.0027
0.0022
0.0016
0.0011
0.0005
-0.0000
-0.0005
-0.0011
-0.0016
-0.0022
-0.0027
= 30 ms-1
= 40 ms-1
= 50 ms-1
time, s
e)
aileron deflection, rad
0.0655
0.0524
0.0393
0.0262
0.0131
-0.0000
-0.0131
-0.0262
-0.0393
-0.0524
-0.0655
time, s
f)

subsidence performance. It is, however, difficult to provide a quantitative assessment of the inherent robustness properties of the two designs without recourse to a more rigorous mathematical analysis. It must also be remembered that the VSS controller is, to some extent, at an early stage of development and a more robust design may be had by a more careful investigation of the likely changes in the system 'A' and 'B' matrices with airspeed and perhaps a better choice of null space dynamics. The VSS technique does, however, provide a stable and acceptable performance with relatively small actuator demands.

8.3 Summary

In this Chapter the robustness aspects of the two closed-loop state feedback schemes applied to the lateral motion sub-system of the aircraft have been investigated. In the above discussion a quantitative assessment has been made of the degree to which a linear state feedback law maintains the eigenstructure of the closed-loop system as the system parameters vary about some nominal operating point. This was made possible by employing the results of Chapter 2 to provide a locally linearised model of the aircraft at a number of time points over a typical lateral manoeuvre. The non-linear controller has also been assessed although in this case some difficulty was encountered principally due to the discontinuous nature of the control. A purely qualitative assessment revealed that the non-linear controller provided somewhat poorer performance than the LQP design. Since these result are to some extent preliminary it is now necessary to consider a more thorough analysis of the VSS design particularly bearing in mind the likely parameter variations in the actual system. The LQP design has, however, been shown to be of a robust nature and capable of providing desirable improvements in system performance.

CHAPTER 9

Conclusions and Discussions

The main aim of this project has been to provide a critical assessment of flight control system design using 'modern' control techniques based on a comparison with conventional classical design strategies. At a very early stage it was clear that to provide such an assessment would require that the contol systems be employed in a realistic aircraft environment. Here two alternative approaches were possible.

Firstly, the control laws could be designed around a 'linear' model of an aircraft and their performance assessed with a view to later implementation on the real aircraft. A number of disadvantages are clear principally due to the inherent non-linear nature of the aircraft system which would not be reflected in the linear model and hence little feel could be gained, at the design stage, for the 'sensitivity' of the control law to these non-linearities. Allied to this is the fact that the aircraft is subject to relatively large changes in its parameters as a result of changes in flight configuration which again are not manifest in the linear aircraft model, this being applicable at only one specific flight condition. In addition, the role of the pilot in any flight control scheme requires careful consideration. The "pilot-in-the-loop" aspects of the proposed control scheme are important and any assessment of flight control laws should include the pilot as an active element. It is often the case that the pilot's initial reactions to a flight control scheme are obtained 'on-line' ,so to speak, during flight trials. This can be a costly and dangerous method of assessing the flight controller's performance.

The second approach is based upon the use of real time and non-real time simulation facilities at an early stage in the design work. A number of advantages are to be accrued here since the control scheme can be provided with a

realistic non-linear environment without recourse to costly flight trials. Additionally, the "pilot-in-the-loop" aspects can be more thoroughly explored through real time simulation without the flight controller actually being flown. The relatively high capital outlay for such simulation facilities is something of a disadvantage but the rewards are much higher since the simulator can be reconfigured for a large number of individual airframes. Of prime importance for such simulation work, however, is an accurate mathematical description of the aircraft encompassing all of the inherent non-linearities and parameter changes.

It was with the development of the second approach in mind that the project moved initially towards the provision of a small scale real time simulation upon which to base all subsequent assessment work. The motivation here was provided by some work by Marconi Avionics who had available a small remotely piloted vehicle (RPV), the Machan, which was flown as a 'test bed' for flight control system design. An accurate mathematical model of this aircraft was derived and is detailed in Chapter 1. The aerodynamic data for this vehicle was derived from wind tunnel tests on the actual airframe. As mentioned above the real time aspects of such a simulation facility are important if the pilot's input to the control is to be included. From the outset then it was decided to attempt, as far as possible, to provide a real time response from the simulation.

The aircraft industry has also appreciated the relatively large gains to be made in providing cockpit-like environments on the ground using simulations, so much so that a great deal of time and money have been invested in the development of such facilites. The trend today is towards using very sophisticated digital computing power to provide powerful simulators. For the purposes of this project, which had a somewhat lower budget, a number of different microcomputers were investigated with particular regard to their ability to provide a close-to real time simulation. As the development of this facility proceeded it became apparent that some form of display system and

provision for pilot inputs would be required in addition to flight control system implementation. Chapter 3 considers these points in some detail. To provide all of these facilities on a relatively low power, but inexpensive, machine ultimately proved a difficult proposition without severely compromising the real time aspects. A far better solution would be to distribute the computational workload to several small machines with a single dedicated computing engine forming the nucleus and solving the non-linear vehicle equations of motion. Peripheral computers could then be dedicated to tasks such as flight control, display management, etc. This should allow a far better real time facility to be realised and work on a system such as this is currently being undertaken. This work will eventually lead to a relatively inexpensive but powerful flight simulation facility upon which a variety of avionics work can be undertaken.

Having completed an evaluation study into the real time simulation aspects, the problem of addressing the control system design still remained. There was little alternative at this stage but to attempt an essentially 'off-line' examination of the control system design methodologies. To this end the non-linear vehicle equations of motion for the Machan were used in a FORTRAN implementation of the simulation and this is used almost exclusively in the latter parts of the work (Chapters 5-8). Since the majority of control system designs rely upon essentially linear systems theory it was vital to provide a linearised model for the Machan aircraft. A number of approaches are possible here and Chapter 2 considers two of these.

The first approach is a so called stability axis derivation of a linearised state space system description. This technique relies upon specialising the trim conditions of the aircraft to 'straight-and-level' flight and applying intuitive arguments and an 'a-priori' knowledge of some of the aerodynamic derivative terms. Whilst this technique provides a relatively good idea of the small perturbation behaviour of the aircraft large perturbation effects are, for the most part, ignored. The result is a linear system

description which can, and indeed is, used as a basis for the design of stability augmentation systems which aim to provide acceptable stability margins from the basic airframe. An alternative technique, if somewhat more involved, is to consider a first order linearisation of the vehicle non-linear equations. Here the functional dependence of each of the elements of the system 'A' matrix on the set of system state variables is retained. In this case the resulting system description is 'locally linear' i.e. linearisation is updated at each point in the flight envelope and the resulting linear model is directly applicable at that point. This technique is thus a means of providing a more 'global' system description than the very specialised classical small perturbation analyses. The results produced by both of these techniques are, however, very similar and may explain why controllers designed using the small perturbation models provide acceptable performance. The small perturbation model does, however, retain some functional dependence on, for example, airpeed since the majority of the aerodynamic derivatives change with airspeed, air density, incidence, etc. and these are retained in this system description.

It would be difficult to provide a meaningful examination of modern control policies without some form of 'yardstick' to compare them against. The purpose of the work of Chapter 5 is to provide this 'yardstick' by examining the classical approach to flight control systems design. For many years such techniques have provided acceptable designs for flight controllers and engineers in the avionics industry have developed more than a passing familiarity with classical performance measures. Modern aircraft demand more from the controller in order to realise improvements in handling qualities. The work embodied in Chapter 5 provides a baseline stability augmentation system for the Machan RPV. This is an original design exercise based principally upon the application of root locus techniques to the linear small perturbation state-space model developed in Chapter 2. The Machan is a fairly unique control problem since long term attitude stability is vital for any remotely piloted vehicle. Performance requirements

for any aircraft are briefly reviewed in Chapter 5 but it is normal to require that the characteristic modes of the aircraft have some specified degree of damping and/or a specifed rise time or natural frequency. Given that the aerodynamic derivatives vary with varying airspeed, etc. it is important that this modal disposition is maintained and that a reasonable stability margin can be guaranteed. The root locus provides a very visible tool in this application since the poles and zeros of a system may be maintained within given regions of the s-plane as a result of changes in flight variables. Gain-scheduling is often employed to maintain the stability and responsiveness of the aircraft. The closure of a number of loops around the Machan and the design of suitable compensation elements within these loops results in an acceptable performance from the vehicle. The aircraft response has been shown, however, to be sensitive to changes in airspeed although adequate stability margins are achieved. The design process in the classical problem is thus relatively straightforward, if somewhat heuristic. Firstly, a number of loops must be identified (measured variable - control surface deflection) around which to provide a controller structure. The loop is then analysed, its transfer function determined, and a suitable controller developed which provides prescribed modal properties and adequate stability margins. In developing any modern control schemes it is important to ensure that these not only provide the specified performance but also that the design methodology is not too far removed from the classical techniques. This will ensure the easy acceptance of these new schemes in the avionics industry which is, understandably, somewhat resistant to change.

The identification of applicable modern control techniques which, in some degree, fulfilled the criteria above proved difficult. In Chapters 6 and 7 two state feedback schemes have been examined in some detail. Chapter 6 covers optimal control (LQP) whilst Chapter 7 is concerned with non-linear (variable structure, VSS) control. Optimal control has previously been examined as a means of designing flight control systems. Much of this work, however,

concentrates on the application to linear models. In this project these control schemes have been applied not only to the linear model but also to the non-linear simulation developed in the early work. A major problem with optimal control is in the choice of the Q and R weighting matrices for a particular problem and how the elements of these matrices relate to the properties of the final design. In Chapter 6 a technique has been developed which obviates the need to provide Q and R explicitly and requires the designer to specify only the desired eigenstructure of the closed-loop system. Further, the ability to only specify partially the eigenvectors of the system is of considerable importance. This allows some measure of flexibility in the design process but required the development of a least squares fitting algorithm to determine the 'best fit' to these partially specified eigenvectors. This is very similar to the classical design techniques where the designer specifies a given set of modes (eigenvalues) and implicitly imposes a given modal (eigevector) decomposition by suitable choice of the control loops around various sub-systems of the aircraft. For the aircraft problem this particular design technique would seem highly appropriate since the choice of the Q and R matrices is implicitly contained within the algorithm. The choice of modal coupling is easily determined from intuitive arguments regarding the airframe and, by careful choice of the state structure, actuator limitations may also be included.

The application of this technique to the Machan has been shown to provide very acceptable performance when used with the non-linear simulation at various airspeeds. A major drawback is that a set output structure is imposed by the design. The technique may, however, be extended to the output assignment problem and again it has been shown that, given certain restrictions on the system structure, the output assignment problem can be solved if limitations are imposed by the available measurement set. Output assignment is, however, undesirable in the aircraft problem since defined state coupling is required.

A possible difficulty with any state feedback scheme is

the need to measure all system states. This may not be easy on aircraft, particularly RPV's, where a limited instrumentation pack is carried. Observer techniques may be used but another possibility is to employ a state feedback design based on a reduced measurement set. This is worth considering with aircraft since the state equations are readily partitionable into sub-systems each associated with one or more of the characteristic airframe modes. By way of an example a reduced order Dutch roll controller is considered for a third order model of Machan using eigenstructure assignment techniques. It has been shown that this type of control is feasable by applying it to the non-linear simulation with classically designed loops closed in the other major aircraft sub-systems. This type of 'hybrid' configuration is seen as a very powerful method of combining both the modern and classical design approaches.

A more speculative design technique is examined in Chapter 7. This is basically an extension of some already existing work on non-linear control structures applied to aircraft flight control (ZOC and VICTOR). These control philosophies promise much in terms of better performance in the presence of plant uncertainties. Relay switched or 'bang-bang' control systems are by no means new but the sliding mode theory developed in recent years is only just begining to provide a more rigorous theoretical background for non-linear control design in general. In Chapter 7 the theory of the sliding mode is reviewed. A design technique has been described for multivariable VSS design. The aim of this technique is to force the system to follow a trajectory in state-space which the designer may specify. Along this trajectory the closed-loop response becomes insensitive to bounded changes in the underlying system. The technique is in many respects similar to the optimal control design since both exploit the high gain (asymptotic) properties of the system. This being the case the technique is the first real attempt to combine the elegance of the eigenstructure assignment methods within a novel non-linear controller structure. The significance of the eigenstructure in each case is slightly different although the ultimate response achieved is similar. There is a very strong parallel

between the sliding mode and the optimal control policies. Both exploit the asymptotic properties of the system in order to provide improvements in parameter insensitivity and prescribed modal decomposition. This would also seem to be a possible area in which multivariable root locus techniques could be usefully employed by providing a visual indication of the asymptotic properties of the locus.

The sliding mode design has been applied to the non-linear Machan simulation for a variety of airspeeds. The results obtained are similar in many respects to the optimal control designs considered in Chapter 6. A major criticism of any switched control design is the relatively high actuator activity which accompanies the discontinuous control applied to these devices. The results of Chapter 7 do not show this 'chatter' particularly well. This is primarily due to the filtering effects of the actuator, modelled as first order lags. The design allows for the magnitude of the discontinuities to be reduced somewhat although this affects the degree to which the sliding mode is maintained. Ideally very large discontinuities in the control are required to maintain the sliding mode but this would be very undesirable from the actuators' viewpoint. It is possible in the design to avoid the discontinuous control action although this implies that the sliding regime is only reached asymptotically, in infinite time. The results of Chapter 7 also attempt to compare the VSS responses when employed with an approximate linear model of the Machan dynamics with the fully non-linear simulation. This is perhaps a little unrepresentitive since both the initial reaching behaviour of the two systems and the subsequent sliding behaviour will be different. Similar time responses are, however, obtained. The reduced order Dutch roll control problem is again addressed, this time with a VSS controller. The results of this comparison show that a reduced order VSS controller is possible but actuator activity is very large. There is also the possibility that the sub-system may be forced away from the sliding regime by coupling from other sub-systems. Further work is clearly required on this aspect of VSS control. The development of

the VSS design technique and its application to the flight control problem is felt to provide a major contribution to this area of work.

An underlying theme of much of the work on flight control system design is the need to provide parameter insensitive control. This is vital in aircraft since adequate stability margins must be maintained in the presence of large parameter variations due to changes in flight condition. This general problem is one 0of robustness. This being a large scale measure of parameter insensitivity. Chapter 8 considers the problem of assessing the robustness achieved from both the LQP and VSS controllers. Whilst much theoretical work has been undertaken on parameter sensitivity and robustness measures this has largely been confined to linear systems analysis with a limited variation in the underlying system. The reasons for this are simply to provide analytically tractable solutions to the problem. For the aircraft example these techniques were considered inapplicable since the system is highly non-linear and all the parameters vary over large ranges. The robustness assessment is thus based on a fairly pragmatic 'brute-force' approach which uses the locally linearised aircraft dynamics of Chapter 2 and determines quantitatively the variations in the eigenstructure of the system due to changes in airspeed, etc. It proved difficult to display this information in a meaningful manner and a graphic approach was adopted. The use of this approach has demonstrated that the LQP design provided a robust eigenstructure over the relatively small range of maneouvres investigated. As a result of this work it has become apparent that there is no simple way of providing robustness measures for VSS control schemes. The sliding mode approach is, however, based on a strong mathematical structure whereas the more heuristic ZOC and VICTOR schemes provide no way of assessing robustness properties. Initial examination of these results indicates that the LQP design is more robust than the VSS design although refinement of the VSS design may provide improvements in this respect.

The results of this comparative study of modern control designs applied to the Machan can broadly be summarised as follows :

The classical design techniques are well understood in industry. An extension of these basic methodologies into optimal control has been shown to be possible using the eigenstructure assignment techniques of Chapter 6. In addition, these optimal control strategies have been shown to have desirable robustness properties by employing them with a non-linear simulation of the aircraft and providing a quantitative assessment of the degree to which the eigenstructure is maintained. The application of LQP designs to reduced order sub-systems of the aircraft has also been shown to provide a 'half-way-house' form of control by combining both classical and optimal control strategies in the one design. In contrast, the VSS controller provides a very similar route into the design and again exploits the asymptotic eigenstructure properties of the system. In this respect the designs are very similar. Some penalty is associated with VSS since the control is discontinuous by nature and this may place heavy demands on actuators. There would appear to be little to be gained from VSS in the nature of improved robustness although this may be an unfair criticism since the design is somewhat provisional. VSS may have other disadvantages particularly in terms of its implementation. Using sampled data flight control systems may give rise to unexpected complications in any switched control structure. Additionally, certain classes of non-linearity (e.g. hysteresis, dead-band) may generate unpredictable behaviour (e.g. limit cycle oscillations). On balance then the optimal control strategies would seem to be the most appropriate alternative. Further work into the VSS control, in particular the aspects identified above, may, however, alter this balance toward VSS.

The present project work is likely to be extended in a number of ways. Firstly the real time simulation will be further advanced to provide a distributed system. This will allow some further work to be undertaken into the

development of the flight control hardware. In the initial stages of this work the classically designed compensators will be implemented. Later the work will be extended to encompass the state variable type structures and sliding mode control. Further work is also required on the VSS designs. In particular it would be desirable to be able to specify prescribed degrees of robustness due to known changes in some or all of the elements of the system matrices. The ability to investigate the effects of introducing simple non-linearities into the system is also desirable. The provision of realistic small scale simulation facilities in addition to the development of modern control schemes, which are based on classical design methodologies, will be of major value to the avionics industry.

References

1) E. L. Hughton and N. B. Carruthers ; "Aerodynamics for Engineering Students", 3rd. edition, Edward Arnold.

2) A. L. Greensite ; "Analysis and Design of Space Vehicle Flight Control Systems", Vol II of 2, Spartan.

3) J. H. Blakelock ; "Automatic Control of Aircraft and Missiles", John Wiley, 1965.

4) D. McRuer, I. Ashkenas and D. Graham ; "Aircraft Dynamics and Automatic Control", Princeton University Press, 1973.

5) H. Nicholson (editor) ; "Modelling of Dynamical Systems Vols. I + II", Peter Peregrinus, 1980, IEE Control Series 12.

6) J. Aplin and P. Lamb ; "The Machan Simulation", Marconi Avionics, FARL Report No. 262/1332/TN/S.

7) D. C. Price and R. B. Smith ; "Joint CACD/FARL Proposal No. 262/0210 'Proposal for Flight Control Systems Research'", Marconi Avionics, Proposal No. 262/0210.

8) D. Howe ; "Machan - Wind Tunnel Tests" ; Cranfield College of Aeronautics Memo 8002, (Commercial in Confidence).

9) B. Etkin ; "Dynamics of Flight", J. Wiley, 1959.

10) W. J. Duncan ; "Control and Stability of Aircraft", Cambridge University Press, 1959.

11) W. A. Johnson and D. H. Weir ; "Pilot's Response to Stability Augmentation System Failures and Implications for Design", Journal of Aircraft, Nov-Dec 1969.

12) I. A. M. Hall ; "Study of the Human Pilot as a Servo-Element", Journal of Royal Aero. Soc., 67, June 1963.

13) I. L. Ashkenas and D. T. McRuer ; "A Theory of Handling Qualities Derived from Pilot - Vehicle System Considerations", Aerospace Engineering, 21, 2, Feb. 1962.

14) J. S. Tyler Jnr. ; "The Characteristics of Model-Following Systems Synthesised by Optimal Control", IEEE Trans. on Automatic Control, Oct. 1964.

15) C. A. Markland ; "Optimal Model-Following Control System Synthesis Techniques", Proc. IEE, 117, 3, March 1970, 623-627.

16) E. Kreindler and D. Rothchild ; "Model-Following in Linear-Quadratic Optimization", AIAA J., 14, 835-842, July 1976.

17) S. J. Asseo ; "Application of Optimal Control to Perfect Model-Following", AIAA Journal of Aircraft, Vol. 7, No. 4, July 1970, 308-313.

18) F. R. Gill and D. McLean ; "Computer Study of a Self-Adaptive Longitudinal Control System Employing Adjustments to the Command Input Filter", RAE Technical Report, TR79093, 1979.

19) J. R. Broussard and P. W. Berry ; "Relationship Between Implicit Model Following and Eigenvalue Eigenvector Placement", IEEE Trans. Auto. Cont., AC-25, 3, June 1980.

20) Y. T. Chan ; "Perfect Model-Following with a Real Model", Proc. Joint Automatic Contr. Conf., Columbus, Ohio, 287-293, 1973.

21) G. K. L. Kriechbaum and R. W. Stineman ; "Design of Desirable Airplane Handling Qualities via Optimal Control", AIAA Paper No. 71-956, Aug 1971.

22) C. A. Harvey and R. E. Pope ; "Design Techniques for Multivariable Flight Control Systems", In 'Control and Dynamic Systems', 18, Academic Press, 1982.

23) M. G. Safonov and M. Athans ; "Gain and Phase Margins for Multiloop LQG Regulators", IEEE Trans. Auto. Cont., AC-22, 173-176, 1977.

24) J. C. Doyle ; "Guaranteed Margins for LQG Regulators", IEEE Trans. Auto. Cont., AC-24, 607-611, 1979.

25) P. M. DeRusso, R. J. Roy and C. M. Close ; "State Variables for Engineers", J. Wiley, 1965.

26) S. S. L. Chang ; "Synthesis of Optimum Control Systems", McGraw Hill, 1961.

27) B. D. O. Anderson and J. B. Moore ; "Linear Optimal Control", Prentice Hall, 1971.

28) J. S. Tyler and F. B. Tuteur ; "The use of a Quadratic Performance Index to Design Multivariable Control Systems", IEEE Trans. Auto. Control, AC-11, 84-92, 1966.

29) C. H. Houpis and C. T. Constantinides ; "Correlation Between Conventional Control Figures of Merit and the Q Matrix of the Quadratic Cost Function", Proc. of 14th Mid West Symposium on Circuit Theory, 11.3.1-11.3.10, May 1971.

30) K. Ogata ; "State Space Analysis of Control Systems" Prentice Hall, 1976.

31) A. G. J. McFarlane ; "An Eigenvector Solution of the Optimal Linear Regulator Problem", J. Electronics and Control, 14, 6, June 1963.

32) M. Athans and P. L. Falb ; "Optimal Control, An Introduction to the Theory and Its Applications", McGraw-Hill, 1966.

33) J. J. D'Azzo and C. H. Houpis ; "Linear Control Systems Analysis and Design", McGraw Hill, 1975.

34) C. A. Harvey and G. Stein ; "Quadratic Weights for Asymptotic Regulator Properties", IEEE Trans. on Auto. Cont., AC-23, 378-387, 1978.

35) G. A Stein ; "Generalised Quadratic Weights for Asymptotic Regulator Properties", IEEE Trans. on Auto. Cont., AC-24, 559-566, 1979.

36) M. J. Grimble ; "Design of Optimal Output Regulators Using Multivariable Root Loci", IEE Proc. D, 128, 2, 41-49, March 1981.

37) T. E. Bullock and J. M. Elder ; "Quadratic Performance Index Generation for Optimal Regulator Design", Proc. IEEE Conf. on Decision and Contol, W8-3, 123-124, Dec. 1971.

38) D. Graupe ; "Derivation of Weighting Matrices Toward Satisfying Eigenvalue Requirements", Int. J. Control, 16, 881-888, 1972.

39) S. Shinathkumar and R. P. Rhoten ; "Eigenvalue/Eigenvector Assignment for Multivariable Systems", Electron. Lett., 11, 124-125, 1975.

40) S. L. Shad, D. G. Fisher and D. E. Seborg ; "Eigenvalue/Eigenvector Assignment for Multivariable Systems and Further Results for Output Feedback Control", Electron. Lett., 11, 388-389, 1975.

41) W. M. Whonham ; "On Pole Assignment in Multi-Input, Controllable Linear System", IEEE Trans. Auto. Control, AC-12, 660-665, Dec. 1967.

42) M. C. Maki and J. Van de Vegte ; "Optimization of Multiple-Input Systems with Assigned Poles", IEEE Trans. Auto. Cont., AC-19, 130-133, 1974.

43) F. Fallside ; "Control System Design by Pole-Zero Assignment", Academic Press, 1677.

44) G. Klein and C. B. Moore ; "Eigenvalue-Generalised Eigenvector Assignment with state Feedback", IEEE Trans. Auto. Contr., AC-22, 883-885, Oct. 1977.

45) B. C. Moore ; "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed-Loop Eigenvalue Assignment", IEEE Trans. on Auto. Cont., AC-21, 685-692, 1976.

46) B. Porter and J. J. D'Azzo ; "Comments on "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed-Loop Eigenvalue Assignment"", IEEE Trans. Auto. Control, AC-22, 888-889, 1977.

47) G. Stein and A. H. Henke ; "A Design Procedure and Handling Quality Criteria for Lateral-Directional Flight Control Systems", AFFDL-TR-70-152, May 1971.

48) H. Kwakernaak and R. Sivan ; "Linear Optimal Control Systems", Wiley-Interscience, New York, 1972.

49) M. J. Grimble ; "Optimal Return Difference Matrix", Proc. IEE, 124, 2, 167-168, 1977.

50) H. H. Rosenbrock ; "State Space and Multivariable Theory", Nelson, 1970.

51) D. H. Owens ; "Invariant Zeros of Multivariable Systems: A Geometric Analysis", Int. J. Control, 26, 537-578, 1977.

52) U. Shaked and N. Karcanias ; "The use of Zeros and Zero Directions in Model Reduction", Int. J. Cont., 23, Jan 1976.

53) D. H. Owens ; "Feedback and Multivariable Control", Peter Peregrinus, 1978.

54) D. H. Owens ; "First- and Second-Order-Like Structures in Linear Multivariable Control Systems Design", Proc. IEE, 122, 9, 935-941, 1975.

55) R. J. Patton and M. J. Grimble ; "An Investigation into the use of Kalman Filtering in Dynamic Ship Positioning", Research Report No. EEE/14, Sheffield City Polytechnic, 1978.

56) A. Zaki ; "Application of Modern Control Techniques in the Design of Aircraft Autopilots", Proc. Int. AMSE Conf. on Modelling and Simulation, Paris, July 1982.

57) H. N. Tobie, et al ; "A New Longitudinal Handling Qualities Criterion", National Aerospace Electronics Conference, Dayton, Ohio, May 1966.

58) M. J. Grimble ; "Design of Optimal Stochastic Regulating Systems Including Integral Action", Proc. IEE, 126, 9, Sept. 1979.

59) M. J. Grimble ; "The Design of s-Domain Optimal Controllers with Integral Action for Output Feedback Control Systems", Research Report No. EEE/24, Sheffield City Polytechnic, Dec. 1978.

60) G. Lietmann ; "On the Efficacy of Non-Linear Control in Uncertain Linear Systems", J. Dyn. Sys. Meas. Control, Trans. ASME, G, 101, 2, 1980.

61) E. P. Ryan ; "A Variable Structure Approach to Feedback Regulation of Uncertain Dynamical Systems", Int. J. Control, 38, 6, 1121-1134, 1983.

62) E. P. Ryan, G. Leitmann and M. Corless ; "Practical Stablizability of Uncertain Dynamical Systems: Application to Robot Tracking", submitted for publication, 1984.

63) A. Y. Sivaramakrishnan, M. V. Hariharan and M. C. Srisailam ; "Design of Variable-Structure Load-Frequency Controller Using Pole Assignment Techniques", Int. J. Control, 40, 3, 487-498, 1984.

64) J. J. Slotine and S. S. Sastry ; "Tracking Control of Non-Linear Systems using Sliding Surfaces, with Application to Robot Manipulators", Int. J. Control, 38, 2, 465-492, 1983.

65) K-K. D. Young and H. G. Kwatny ; "Variable Structure Servomechanism Design and Applications to Overspeed Protection Control", Automatica, 18, 4, 385-400, 1982.

66) A. S. I. Zinober ; "Controller Design Using the Theory of Variable Structure Systems", Self Tuning and Adaptive Control, (S. A. Billings and C. Harris eds.), Chapter 9.

67) U. Itkis ; "Control Systems of Variable Structure", J. Wiley, New York, 1976.

68) A. G. Filippov ; "Application of the Theory of Differential Equations with Discontinuous Right-Hand Sides to Non-linear Problems in Automatic Control", Proc. 1st IFAC Congress, Moscow, 1960, 2, 923-927.

69) E. P. Ryan and M. Corless ; "Ultimate Boundedness and Asymptotic Stability of a Class of Uncertain Dynamical Systems via Continuous and Discontinuous Feedback Control", IMA J. Math. Control & Information, Submitted for publication.

70) V. I. Utkin ; "Variable Structure Systems with Sliding Modes", IEEE Trans. Auto. Control, AC-22, 2, 212-222, 1977.

71) V. I. Utkin ; "Sliding Modes and Their Application in Variable Structure Systems", MIR, Moscow, 1978.

72) B. Drazenovic ; "The Invariancy Conditions in Variable Structure Systems", Automatica, 5, 287-295, 1969.

73) O. M. E. El-Ghezawi, A. S. I. Zinober and S. A. Billings ; "Analysis and Design of Variable-Structure Systems Using a Geometric Approach", Int. J. Control, 38, 3, 657-671, 1983.

74) F. R. Gill ; " Design Studies for Non-linear Control Laws for Automatic Approach and Landing, 1: Airspeed Hold", RAE Technical Report, TR80096, 1980.

75) F. R. Gill ; "Non-linear Pitch Rate to Elevator Control Laws for a Combat Aircraft", RAE Technical Report, TR79075, 1979.

76) D. McLean ; "Globally Stable Nonlinear Flight Control System" ; IEE Proc. D, Vol.130, No. 3, May 1983.

77) C. M. Dorling and A. S. I. Zinober ; "Hyperplane Design in Model Following Variable Structure Control Systems", Submitted to IFAC 1985 Symposium, University of York, July 1985.

78) C. M. Dorling and A. S. I. Zinober ; "Computer Aided Design of Robust Multivariable Variable Structure Control Systems", Submitted to 3rd. IFAC/IFIP Symposium on CAD in Engineering Systems, Copenhagen, July/Aug. 1985.

79) C. M. Dorling and A. S. I. Zinober ; "The Design and Implementation of Multivariable Variable Structure Control Systems", Submitted to IEE Control '85 Conference, Cambridge, July 1985.

80) C. M. Dorling and A. S. I. Zinober ; "Computer Aided Design of Variable Structure Control Systems", IMC Workshop on Computer Aided Control System Design, Brighton, 1984, 67-71.

81) T. A. Bezvodinskaya and E. F. Sabaev ; "Stability Conditions in the Large for Variable Structure Systems", Autom. and Remote Control, 10, 1596-1599, 1974.

82) G. I. Lozgachev ; "The Constuction of a Liapunov Function for Variable Structure Systems", Autom. and Remote Control, 8, 161-162, 1972.

83) O. M. E. El-Ghezawi, A. S. I. Zinober, D. H. Owens and S. A. Billings ; "Computation of the Zeros and Zero Directions of Linear Multivariable Systems", Int. J. Control, 36, 5, 833-843, 1982.

84) O. M. E. El-Ghezawi, S. A. Billings and A. S. I. Zinober ; "Variable Structure Systems and System Zeros", IEE Proc. D, Vol. 130, No. 1, January 1983.

85) K-K. D. Young, P. V. Kokotovic and V. I. Utkin ; "A Singular Perturbation Analysis of High-Gain Feedback Systems", IEEE Trans. Auto. Cont., AC-22, 931-937, Dec. 1977.

86) P. V. Kokotovic and P. Sannuti ; "Singular Perturbation Method for Reducing the Model Order in Optimal Control Design", IEEE Trans. Auto. Control, AC-13, 377-384, Aug. 1968.

87) U. Shaked ; "Design Techniques for High Feedback Gain Stability", Int. J. Control, 24, 1, 137-144, 1976.

88) B. A. White and P. M. Silson ; "Reachability in Variable-Structure Control Systems", IEE Proc. D, 131, 3, 85-91, May 1984.

89) B. A. White ; "Reduced-Order Switching Functions in Variable-Structure Control Systems", IEE Proc. D, 130, 2, 33-40, March 1983.

90) P. M. Frank ; "Robustness and Sensitivity: a Comparison of the Two Methods", ACI Conference, Copenhagen, 1983.

91) P. M. Frank ; "Introduction to System Sensitivity Theory", Academic Press, New York, 1978.

92) T. R. Crossley and B. Porter ; "Eigenvalue and Eigenvector Sensitivities in Linear Systems Theory", Int. J. Control, 10, 2, 163-170, 1969.

93) B. Porter ; "Eigenvalue Sensitivity of Modal Control Systems to Loop-Gain Variations", Int. J. Control, 10, 2, 159-162, 1969.

94) H. Qui and V. G. Gourishankar ; "Design of Optimal Feedback Controllers for Minimum Eigenvalue Sensitivity", Optimal Control Applications and Methods, 5, 309-317, 1984.

95) B. S. Morgan ; "Sensitivity Analysis and Synthesis of Multivariable Systems", IEEE Trans. Auto. Cont., AC-11, 506-512, 1966.

96) V. Gourishankar and K. Ramar ; "Pole Assignment with Minimum Eigenvalue Senstitivity to Plant Parameter Variations", Int. J. Control, 23, 493-504, 1976.

97) S. L. Shah, D. G. Fisher and D. E. Seborg ; "Eigenvalue Invariance to System Parameter Variations by Eigenvector Assignment", Int. J. Control, 26, 871-881, 1977.

97) S. L. Shah, D. G. Fisher and D. E. Seborg ; "Eigenvalue/Eigenvector Assignment for Multivariable Systems and Further Results for Output Feedback Control", Electron. Lettors., 11, 388-389, 1975.

98) S. Srinathkumar and R. P. Rhoten ; "Eigenvalue and Eigenvector Sensitivities in Linear System Theory", Int. J. Control, 10, 163-170, 1969.

99) C. M. Dorling and A. S. I. Zinober ; "A Comparative Study of the Sensitivity of Observers", ACI 83, 6.32-6.37, 1983.

100) P. Gawthrop ; "Robustness of Self-Tuning Controllers", IEE Colloquium on Robustness and Sensitivity in Feedback Systems, Digest No. 1982/64, 1982.

101) M. G. Safonov ; "Stability and Robustness of Multivariable Feedback Systems", MIT Press, 1980.

102) Y. Kamiya ; "Servo-system Design Technique Utilising Sensitivity as a Design Parameter" ; IEE Proc., Pt. D, 130, 5, Sept. 1983.

103) P. M. Frank, F. Heger and J. Radischat ; "Design of Robust Control Systems by Pole Assignment", Control and Computer, 10, 2, 1982.

104) M. A. Laughton ; "Sensitivity in Dynamical System Analysis", Int. J. Electron. & Control, 17, Nov 1964.

105) E. Soroka and U. Shaked ; "On the Robustness of LQ Regulators", IEEE Trans. Auto. Cont., AC-29, 7, July 1984.

106) M. J. Grimble ; "Robustness of Combined State and State-Estimate Feedback Control Schemes", IEEE Trans. Auto. Cont., AC-29, 7, July 1984.

107) R. Ortega ; "Assessment of Stability and Robustness for Adaptive Controllers", IEEE Trans. Auto. Cont., AC-28, 12, 1106-1109, Dec. 1983.

108) K. J. Astrom ; "Robustness of a Design Based on Assignment of Poles and Zeros", IEEE Trans. Auto. Cont., AC-25, June 1980.

109) R. E. Kalman ; "When is a Linear System Optimal ?", J. Basic Engineering, 86, 51-60, 1964.

110) J. M. Horowitz and U. Shaked ; "Superiority of Transfer Function Over State-Variable Methods in Linear Time-Invariant Feedback System Design", IEEE Trans. Auto. Control, AC-20, 84-97, 1975.

111) M. A. Woodhead and B. Porter ; "Optimal Modal Control", Measurement and Control, 6, 301-303, 1973.

112) W. Kassim ; "An Efficient Algorithm for Solving Algebraic Discrete & Continuous Ricatti Equations", ACI Conference, Copenhagen, 1983.

113) J. C. Juang and T. T. Lee ; "On Optimal Pole Allocation and Weighting Matrix Selection", ACI Conference, Copenhagen, 1983.

114) O. A. Solheim ; "Design of Optimal Control Systems with Prescribed Eigenvalues", Int. J. Control, 15, 143-160, 1972.

115) V. I. Utkin ; "Equations of the Slipping Regime in Discontinuous Systems", Automation and Remote Control, 12, 1897-1907, 1972.

116) N. Karcanias and B. Kouvaritakis ; "The Output Zeroing Problem and its Relationship to the Invariant Zero Structure: a Matrix Pencil Approach", University of Cambridge, Department of Engineering Report, CUED/F-CAMS/TR168, 1978.

117) H. Kwakernaak, "Asymptotic Root Loci for Multivariable Linear Optimal Regulators", IEEE Trans Auto. Cont., AC-21, 378-381, June 1976.

118) M. A. Johnson and M. J. Grimble ; "Asymptotic Behaviour of the Closed Loop Poles of Linear Optimal Multivariable Systems", Trans. ASME, Journal of Dyn. Sys., Meas. & Cont., 105, Sept. 1983.

119) M. A. Johnson and M. J. Grimble ; "Optimal Asymptotic Root Loci and a Linear System Canonical Form", Research Report No. EEE/68, Sheffeield City Polytechnic, May 1980.

120) N. E. Gough and Z. M. Ismail ; "Computer-Aided Design of Variable Structure Control Systems", Control and Computers, 10, 3, 71-75, 1982.

121) B. Porter ; "Eigenvalue Sensitivity of Modal Control Systems to Loop-Gain Variations", Int. J. Control, Vol. 10, No. 2, 1969, 159-162.

122) R. T. N. Chen and D. W. C. Shen ; "Sensitivity Analysis and Design of Multivariable Regulators Using Quadratic Performance Criterion", Proc. JACC, 229-238, 1969.

123) E. G. Rynaski, P. A. Reynolds and W. H. Shed ; "Design of Linear Flight Control Systems Using Optimal Control Theory", Tech. Doc. Report ASD-TDR-63-376, April 1964.

124) J. W. Gaul, R. P. Kaiser, G. T. Onega and F. T. DeCanio ; "Application of Optimal Control Theory to VTOL Flight Control System Design", Tech. Rep. AFFDL-TR-67-102, Sept. 1967.

125) G. Ambrosino, G. Celentano and F. Garofalo ; "Variable Structure Model Reference Adaptive Control Systems", Int. J. Control, 39, 6, 1339-1349, 1984.

126) A. Balestrino, G. DeMaria and L. Sciavicco ; "Hyperstable Adaptive Model-Following Control of Linear Plants", System & Control Letters, 1, 4, 232-236, 1982.

127) A. Balestrino, G. DeMaria and A. S. I. Zinober ; "Non-linear Adaptive Model-Following Control", Automatica, 20, 5, 559-568, 1984.

128) K-K. D. Young ; "Design of Variable Structure Model-Following Control Systems", IEEE Trans. Auto. Control, AC-23, 6, 1079-1085, 1978.

129) A. S. I. Zinober, O. M. E. El-Ghezawi and S. A. Billings ; "Multivariable Variable-Structure Adaptive Model-Following Control Systems", Proc. IEE, 129, D, 1, 6-12, 1982.

130) E. A. Barabashin and E. I. Gerashchenko ; "On Introduction to Sliding Modes in Control Systems", Differential Equations, 1, 1, 1965.

131) Y. D. Landau ; "Adaptive Control - The Model Reference Approach", Marcel Decker, New York.

132) E. G. Rynaski and R. F. Whitbeck ; "The Theory and Application of Linear Optimal Control", Tech. Report AFFDL-TR-65-28, January 1966.

133) K. J. B. Hosking ; "Dynamic Programming and Synthesis of Linear Optimal Control Systems", Proc. IEE, Vol. 113, No. 6, June 1966.

134) Y. Bar-ness ; "Optimal Closed-Loop Pole Assignment", Int. J. Control, 27, 421-430, 1978.

135) M. Healy ; "Optimal Control Theories in the Design of Aircraft Stability Augmentation and Control Systems", Measurement and Control, 7, Oct. 1974.

136) B. Porter and T. Crossley ; "Modal Control Theory and Applications", Taylor and Francis, 1972.

137) G. Lee ; "Pole Placement with Feedback gain Constraints", in Proc. 1975 Conf. Decision and Control, 188-190, 1975.

138) H. Kimura ; "Pole Assignment by gain Output Feedback", IEEE Trans. Auto Control, AC-20, 4, 509-516, Aug. 1975.

139) F. O'Hara ; "Stability Augmentation in Aircraft Design", Aeronautical Journal of the Royal Aeronautical Society, Vol. 75, April 1971.

140) F. O'Hara ; "Handling Criteria", Journal of the Royal Aeronautical Society, April 1967.

141) J. P. Sutherland ; "Fly-by-wire Flight Control Systems" AGARD Conference Proceeding No. 58, 1970

142) S. A. Sjoberg ; "Flying Qualities Associated with Several Types of Flight Control Systems", Proceedings of Seventh Anglo American Aero Conference, 1959.

143) N. H. Hughes ; "Some Studies into Improvements in Automatic Throttle Control", AGARD Conference Proceedings, No. 27, 1967.

144) H. M. Davis and R. L. Swain ; "Control of Flexible Aircraft Dynamic Response", AGARD Conference Proceedings, No. 17, 1966.

145) R. B. Johanns and P. M. Burris ; "Flight Controls Damp Big Aircraft Bending", Control Engineering, Sept. 1967.

146) R. K. Grief, E. B. Fry, R. M. Gerdes and T. D. Gossett ; "VTOL Control System Studies on a Six-degree-of-freedom Simulator", Proceedings of 5th Congress of ICAS, 1966.

147) J. D. Simon and S. K. Mitter ; "A Theory of Modal Control", Information and Control, No. 13, 265-278, 1968.

148) R. E. W. Marshall, K. E. Snelling and J. M. Corney ; "The Jaguar Fly-by-Wire Demonstator Integrated Flight Control System", 1981 Proc. Advanced Flight Controls Symposium, USAF Academy.

149) P. W. Giles ; "Power Spectral Density Models of Wind Gusts", University of Cambridge, Control Engineering Technical Note CN/70/4, August 1970.

150) S. Gutman and Z. Palmor ; "Properties of Min-Max Controllers in Uncertain Dynamical Systems", SIAM J. Control Optimization, 20, 6, 850-861, 1982.

151) J. Roskam ; "A Simplified Method to Identify and Cure Roll Coupling", J. Aeronaut. Sci., 29, 5, 1962

152) D. T. McRuer ; "A Feedback Theory Analysis of Airframe Cross-Coupling Dynamics", J. Aeronaut. Sci., 29, 525-533, 1962.

153) B. Porter ; "Synthesis of Dynamic Systems", Nelson, 1969.

154) A. G. J. MacFarlane and N. Karcanias ; "Poles and Zeros of Linear Multivariable Systems: A Survey of Algebraic, Geometric and Complex Variable Theory", Int. J. Control, 24, 33-74, 1976.

155) M. Athans ; "The Role and Use of the Stochastic Linear-Quadratic Gaussian Problem in Control System Design", IEEE Trans. Auto. Cont., AC-16, 529-551, Dec. 1971.

156) B. P. Molinari ; "The Stable Regulator Problem and its Inverse", IEEE Trans. Auto. Cont., AC-18, 454-459, Oct. 1973.

157) M. J. Grimble ; "Optimal Control of Linear Systems with Cross-Product Weighting", Proc. IEE, 126, 1, 95-103, 1979.

158) I. L. Ashkenas and D. T. McRuer ; "Optimization of the Flight-Control Airframe System", Journal of the Aero/Space Sciences, 197-218, March 1960.

159) M. M. Fahmy and J. O'Reilly ; "Eigenstructure Assignment in Linear Multivariable Systems - A Parametric Solution", IEEE Trans. Auto. Cont., AC-28, 10, 990-994, Oct. 1983.

160) P. P. Aslin ; "Aspects of Multivariable Root-Loci", MEng. Dissertation, University of Sheffield, 1981.

161) Hewlett Packard HP 9800 Computer Series, "BASIC Utilities Library I"

162) (ibid) "BASIC Programming Techiniques"

163) (ibid) "GPIO Interface Installation"

164) (ibid) "BASIC Interfacing Techniques"

165) R.S. Data sheet No. R/4507.

166) NAG FORTRAN mini Manual, Mark 9, Numerical Algorithms Group Ltd., Oxford, 1981.

167) E. H. Pallett ; "Automatic Flight Control", 2nd. edition, Granada Publishing, 1983.

168) P. Dyer, A. R. M. Norton and D. Rutherford ; "The Application of Dynamic Programming to the Design of Invariant Autostabilisers", Journal of the Royal Aeronautical Society, Vol. 70, April 1966, 469-476.

169) A. C. Robinson ; "Survey of Dynamic Analysis Methods for Flight Control Design", Journal of Aircraft, 6, 2, March-April 1969.

Appendix 1 :- Pascal Program Version 1 Listing

```
 1
 2   (* Segment contains 4 procedures used in MAINSIM *)
 3   (* Now also contains main program  *)
 4
 5
 6   VAR tend : real ; (* length of simulation (secs) *)
 7       tstep : real ; (* time increment per iteration (secs) *)
 8       tprint : real ; (* time between data outputs *)
 9       T : real ; (* incremental time *)
10       wspeed : real ; (* wind speed *)
11       wdirndeg : real ; (* wind direction (degrees) *)
12       wdirnrad : real ; (* wind direction (rads) *)
13       wgust : real ; (* vertical gust (positive upwards) *)
14       CG : real ; (* position of centre of gravity *)
15       mass : real ; (* aircraft mass *)
16       IX : real ; (* moment of inertia, X-axis *)
17       IY : real ; (* moment of inertia, Y-axis *)
18       IZ : real ; (* moment of inertia, Z-axis *)
19       UG : real ; (* X-speed *)
20       VG : real ; (* Y-speed *)
21       WG : real ; (* Z-speed, vertical downwards *)
22       P : real ; (* roll rate *)
23       Q : real ; (* pitch rate *)
24       R : real ; (* yaw rate *)
25       PDEG : real ; (* roll rate deg/sec *)
26       QDEG : real ; (* pitch rate deg/sec *)
27       RDEG : real ; (* yaw rate deg/sec *)
28       phi : real ; (* roll angle *)
29       theta : real ; (* pitch angle *)
30       psi : real ; (* yaw angle *)
31       phideg : real ; (* poll angle degrees *)
32       thetadeg : real ; (* pitch angle degrees *)
33       psideg : real ; (* yaw angle degrees *)
34       XN : real ; (* X- north position metres *)
35       YE : real ; (* Y- east position metres *)
36       height : real ; (* height metres *)
37       eta : real ; (* elevator angle rads *)
38       zeta : real ; (* aileron angle rads *)
39       tau : real ; (* rudder angle rads *)
40       etadeg : real ; (* elevator angle degrees *)
41       zetadeg : real ; (* aileron angle degrees *)
42       taudeg : real ; (* rudder angle degrees *)
43       UC : real ; (* X-airspeed in body axis m/sec *)
44       UB : real ; (* X-inertia speed in body axis m/sec *)
45       VC : real ; (* Y-airspeed in body axis m/sec *)
46       VB : real ; (* Y-inertia speed in body axis m/sec *)
47       WC : real ; (* Z-airspeed in body axis m/sec *)
48       WB : real ; (* Z-inertia speed in body axis m/sec *)
49       VT : real ; (* total speed m/sec *)
50       UWC2 : real ; (* UC*UC + UW*UW *)
51       UWC : real ; (* sqrt (UWC2) *)
52       UVWC2 : real ; (* UC*UC + VC*VC + WC*WC *)
53       UVWC : real ; (* sqrt (UVWC2) *)
54       alpha : real ; (* incidence angle rads *)
55       beta : real ; (* side slip angle rads *)
56       UA : real ; (* X-component of airspeed m/sec *)
57       VA : real ; (* Y-component of airspeed m/sec *)
58       WA : real ; (* Z-component of airspeed m/sec *)
59       X : real ; (* X-force newtons *)
60       Y :real ; (* Y-force newtons *)
61       Z : real ; (* Z-force newtons *)
62       L : real ; (* rolling moment newton metres *)
63       M : real ; (* pitching moment newton metres *)
64       N : real ; (* yawing moment newton metres *)
65
```

```
66
67
68        UBDOT : real ; (* X-acceleration m/sec/sec *)
69        VBDOT : real ; (* Y-acceleration m/sec/sec *)
70        WBDOT : real ; (* Z-acceleration m/sec/sec *)
71        PDOT : real ; (* roll acceleration rad/sec/sec *)
72        QDOT : real ; (* pitch acceleration rad/sec/sec *)
73        RDOT : real ; (* yaw acceleration rad/sec/sec *)
74        rho : real ; (* air density Kg/metre**3 *)
75        A1 : real ; (* lift curve slope rad**-1 *)
76        CL0 :real ; (* lift coefficient (alpha=0) *)
77        CLETA : real ; (* lift coefficient/elevator angle rad**-1 *)
78        surface : real ; (* wing area m**2 *)
79        LT : real ; (* tail moment arm metres *)
80        ZQ : real ; (* lift force/pitch rate sec/rad *)
81        CL : real ; (* lift coefficient *)
82        LW : real ; (* lift from wing *)
83        LQ : real ; (* lift from pitch rate *)
84        CD0 : real ; (* parasite drag coefficient *)
85        CD : real ; (* drag coefficient *)
86        K : real ; (* induced drag coefficient *)
87        dragforce : real ; (* drag force newtons *)
88        YV : real ; (* sideforce/sideslip derivative *)
89        YR : real ; (* sideforce/yaw rate derivative *)
90        YTAU : real ; (* sideforce/rudder angle coefficient *)
91        YA : real ; (* aerodynamic side force newtons *)
92        B : real ; (* wing span *)
93        LA : real ; (* aerodynamic rolling moment Nm *)
94        A1T : real ; (* lift curve slope of tail due to alpha /rad *)
95        A2ETA : real ; (* lift curve slope of tail due to roll rate /rad *)
96        CMWBD : real ; (* zero pitching moment of wing+body+duct *)
97        CMT : real ; (* pitching moment from tail *)
98        CMETA : real ; (* slope of tail pitching moment *)
99        CM0 : real ; (* pitching moment aft for alpha=0 *)
100       CML : real ; (* pitching moment derivative wrt CL *)
101       CBAR : real ; (* wing chord metres *)
102       LV : real ; (* rolling moment derivative due to sideslip *)
103       LS : real ; (* rolling moment derivative due to aileron *)
104       LP : real ; (* rolling moment derivative due to roll rate *)
105       LR : real ; (* rolling moment derivative due to yaw rate *)
106       MA : real ; (* aerodynamic pitch moment Nm *)
107       taillift : real ; (* tail lift newtons *)
108       ltotal : real ; (* wing+tail lift newtons *)
109       CMW : real ; (* pitching moment coefficient of wing *)
110       CLT : real ; (* lift coefficient of tail *)
111       NV : real ; (* yaw moment coef/sideslip velocity sec/m *)
112       NR : real ; (* yaw moment coef/yaw rate sec/rad *)
113       NR0 : real ; (* yaw moment coef at zero lift *)
114       NRL : real ; (* yaw moment derivative wrt CL *)
115       NTAU : real ; (* yaw moment coef/rudder angle rad**-1 *)
116       NA : real ; (* aerodynamic yawing moment Nm *)
117       U2 : real ; (* flow through duct m/sec *)
118       AD : real ; (* duct area m**2 *)
119       throttle : real ; (* throttle setting *)
120       pmax : real ; (* maximum engine power watts *)
121       etap : real ; (* propeller efficiency *)
122       KE : real ; (* engine rise rate *)
123       PP : real ; (* propeller pitch *)
124       XE : real ; (* engine thrust Newtons *)
125       LE : real ; (* engine torque Nm *)
126       pact : real ; (* actual engine power watts *)
127       pnom : real ; (* nominal engine power watts *)
128       U3 : real ; (* propeller wake speed m/sec *)
129       rps : real ; (* angular engine speed revs/sec *)
130       rpm : real ; (* angular engine speed revs/min *)
131
```

```
132
133
134         gravity : real ; (* gravitational acceleration m/sec/sec *)
135         UGDOT : real ; (* X-acceleration *)
136         VGDOT : real ; (* Y-acceleration *)
137         WGDOT : real ; (* Z-acceleration *)
138         PHIDOT : real ; (* roll rate *)
139         THETADOT : real ; (* pitch angle rate *)
140         PSIDOT : real ; (* yaw angle rate *)
141         pi : real ; (* mathematical constant - PI *)
142
143         iter : integer ;
144         stall : integer ; (* stall indicator *)
145         breakpoints : integer ; (* number of control input breakpoints *)
146
147         transform : array [1..3,1..3] of real ;
148         transpose : array [1..3,1..3] of real ;
149         time : array [0..100] of real ;
150         throtleset : array [0..100] of real ;
151         etaset : array [0..100] of real ;
152         etadegset : array [0..100] of real ;
153         zetaset : array [0..100] of real ;
154         zetadegset : array [0..100] of real ;
155         tauset : array [0..100] of real ;
156         taudegset : array [0..100] of real ;
157
158
159   PROCEDURE INDATA ;
160
161   VAR I : integer ;
162       f : text    ;
163
164   BEGIN
165       reset (f,'VAR1.DAT') ;
166       readln (f,tend) ;
167       readln (f,tstep) ;
168       readln (f,tprint) ;
169       readln (f,CG) ;
170       readln (f,wspeed) ;
171       readln (f,wdirndeg) ;
172       readln (f,UG) ;
173       readln (f,VG) ;
174       readln (f,WG) ;
175       readln (f,phideg) ;
176       readln (f,thetadeg) ;
177       readln (f,psideg) ;
178       readln (f,height) ;
179
180       I := 0 ;
181       reset (f,'VAR2.DAT') ;
182       REPEAT
183          readln (f,time [I]) ;
184          readln (f,throtleset [I]) ;
185          readln (f,etadegset [I]) ;
186          readln (f,zetadegset [I]) ;
187          readln (f,taudegset [I]) ;
188          I := I+1 ;
189       UNTIL EOF(f) ;
190
191       breakpoints:=I ;
192       time [breakpoints]:= tend ;
193
194       pi:=3.14159265 ;
195
196       reset (f,'VAR3.DAT') ;
197
```

```
198
199
200       readln (f,mass) ;
201       readln (f,IX) ;
202       readln (f,IY) ;
203       readln (f,IZ) ;
204       readln (f,A1) ;
205       readln (f,CL0) ;
206       readln (f,CLETA) ;
207       readln (f,ZQ) ;
208       readln (f,surface) ;
209       readln (f,LT) ;
210       readln (f,CBAR) ;
211       readln (f,CD0) ;
212       readln (f,K) ;
213       readln (f,YV) ;
214       readln (f,YR) ;
215       readln (f,YTAU) ;
216       readln (f,LV) ;
217       readln (f,LS) ;
218       readln (f,LP) ;
219       readln (f,LR) ;
220       readln (f,B) ;
221       readln (f,A1T) ;
222       readln (f,A2ETA) ;
223       readln (f,CM0) ;
224       readln (f,CML) ;
225       readln (f,CMWBD) ;
226       readln (f,CMETA) ;
227       readln (f,NV) ;
228       readln (f,NR0) ;
229       readln (f,NRL) ;
230       readln (f,NTAU) ;
231       readln (f,AD) ;
232       readln (f,pmax) ;
233       readln (f,ETAP) ;
234       readln (f,KE) ;
235       readln (f,PP) ;
236
237       wdirnrad:=pi*wdirndeg/180.0 ;
238       phi:=pi*phideg/180.0 ;
239       theta:=pi*thetadeg/180.0 ;
240       psi:=pi*psideg/180.0 ;
241
242       FOR I:=0 TO breakpoints DO
243         BEGIN
244           etaset[I]:=etadegset[I]*pi/180.0 ;
245           zetaset[I]:=zetadegset[I]*pi/180.0 ;
246           tauset[I]:=taudegset[I]*pi/180.0 ;
247           etadeg:=etadegset[0] ;
248           zetadeg:=zetadegset[0] ;
249           taudeg:=taudegset[0]
250         END
251       END;
252
253     PROCEDURE INIT ;
254       BEGIN
255         T:=0 ;
256         iter:=0 ;
257         P:=0.0 ;
258         Q:=0.0 ;
259         R:=0.0 ;
260         wgust:=0.0 ;
261         rho:=1.18 ;
262         gravity:=9.81 ;
263
```

```
264
265
266         pi:=3.14159 ;
267         stall:=0 ;
268
269        `XN:=0.0 ;
270         YE:=0.0 ;
271         XE:=0.0 ;
272         U2:=0.0 ;
273         U3:=0.0 ;
274
275         UB:=transpose[1,1]*UG+transpose[1,2]*VG+transpose[1,3]*WG ;
276         VB:=transpose[2,1]*UG+transpose[2,2]*VG+transpose[2,3]*WG ;
277         WB:=transpose[3,1]*UG+transpose[3,2]*VG+transpose[3,3]*WG ;
278
279      END; (* of INIT *)
280
281
282    PROCEDURE OUTDATA (option : integer) ;
283
284    VAR I : integer ;
285
286    BEGIN
287       IF option=1 THEN
288          BEGIN
289          writeln ('*******************************') ;
290          writeln ;
291          writeln ('*                             *') ;
292          writeln ;
293          writeln ('* MACHAN SIMULATION PROGRAM *') ;
294          writeln ;
295          writeln ('*                             *') ;
296          writeln ;
297          writeln ('*******************************') ;
298          writeln ;
299          writeln ;
300          writeln ('     TEND = ',tend:10:5) ;
301          writeln ;
302          writeln ('     TSTEP = ',tstep:10:5) ;
303          writeln ;
304          writeln ('     TPRINT = ',tprint:5:2) ;
305          writeln ;
306          writeln ;
307          writeln ('     CG = ',CG:4:2) ;
308          writeln ;
309          writeln ('     ETA = ',etadeg:4:2,' (DEG) ') ;
310          writeln ;
311          writeln ('     ZETA = ',zetadeg:4:2,' (DEG) ') ;
312          writeln ;
313          writeln ('     TAU = ',taudeg:4:2,' (DEG) ') ;
314          writeln ;
315          writeln ;
316          writeln ('     PHI = ',phideg:4:2,' (DEG) ') ;
317          writeln ;
318          writeln ('     THETA = ',thetadeg:4:2,' (DEG) ') ;
319          writeln ;
320          writeln ('     PSI = ',psideg:4:2,' (DEG) ') ;
321          writeln ;
322          writeln ;
323          writeln
324       END
325    ELSE IF (option=2) OR (option=3) THEN
326             BEGIN
327               IF option=2 THEN
328                 BEGIN
329
```

```
330
331
332                writeln ;
333                writeln ;
334                write (' TIME   YE  SPEED   XN SPEED') ;
335                writeln ('  HEADING    ROLL RATE   ROLL')
336              END (* of inner if *)
337
338          ELSE
339             BEGIN
340               writeln ;
341               writeln (' ',T:8:4,'      ',height:8:4,'     ',XN:8:4) ;
342               write (VT:8:4,'     ',phideg:8:4,'     ') ;
343               writeln (thetadeg:8:4,'     ',psideg:8:4) ;
344               writeln
345             END (* of ELSE *)
346       END
347    END ; (* of procedure *)
348
349    PROCEDURE AIR ;
350
351    VAR UW,VW,WW,UVWA2 : real ;
352        oppalpha,adjalpha,tanalpha : real ;
353        oppbeta,adjbeta,tanbeta : real ;
354     BEGIN
355        UW:=wspeed*COS(wdirnrad) ;
356        VW:=wspeed*SIN(wdirnrad) ;
357        WW:=wgust ;
358
359        UA:=UG+UW ;
360        VA:=VG+VW ;
361        WA:=WG+WW ;
362
363        UC:=transpose[1,1]*UA+transpose[1,2]*VA+transpose[1,3]*WA ;
364        VC:=transpose[2,1]*UA+transpose[2,2]*VA+transpose[2,3]*WA ;
365        WC:=transpose[3,1]*UA+transpose[3,2]*VA+transpose[3,3]*WA ;
366
367        UVWA2:=UA*UA+VA*VA+WA*WA ;
368        VT:=SQRT(UVWA2) ;
369        UWC2:=UC*UC+WC*WC ;
370        UWC:=SQRT(UWC2) ;
371        UVWC2:=UWC2+VC*VC ;
372        UVWC:=SQRT(UVWC2) ;
373        oppalpha:=(WC/SQRT(UWC2)) ;
374        adjalpha:=1-oppalpha*oppalpha ;
375        adjalpha:=SQRT(adjalpha) ;
376        tanalpha:=oppalpha/adjalpha ;
377        alpha:=ARCTAN(tanalpha) ;
378        IF tanalpha<0.0 THEN alpha:=alpha-pi ;
379        oppbeta:=VC/VT ;
380        adjbeta:=1-oppbeta*oppbeta ;
381        adjbeta:=SQRT(adjbeta) ;
382        tanbeta:=oppbeta/adjbeta ;
383        beta:=ARCTAN(tanbeta) ;
384        IF tanbeta<0.0 THEN beta:=beta-pi
385    END; (* of procedure AIR *)
386
387    PROCEDURE CONTRO ;
388
389    VAR I,tmatch : integer ;
390
391    BEGIN
392       I:=0 ;
393       tmatch:=0 ;
394
395
```

```
396
397
398       REPEAT
399         IF (T>=time[I]) AND (T<time[I+1]) THEN
400            BEGIN
401              throttle:=throtleset[I] ;
402              eta:=etaset[I] ;
403              zeta:=zetaset[I] ;
404              tau:=tauset[I] ;
405              etadeg:=etadegset[I] ;
406              zetadeg:=zetadegset[I] ;
407              taudeg:=taudegset[I] ;
408
409              tmatch:=1
410            END
411          ELSE I:=I+1 ;
412          UNTIL (tmatch=1) OR (I>breakpoints-1)
413
414     END ; (* of procedure CONTRO *)
415
416
417     PROCEDURE ACCEL ;
418
419     BEGIN
420        UBDOT:=X/mass-WB*Q+VB*R ;
421        VBDOT:=Y/mass-UB*R+WB*P ;
422        WBDOT:=Z/mass-VB*P+UB*Q ;
423
424        PDOT:=(L+(IY-IZ)*R*Q)/IX ;
425        QDOT:=(M+(IZ-IX)*P*R)/IY ;
426        RDOT:=(N+(IX-IY)*P*Q)/IZ
427
428     END ; (* of procedure ACCEL *)
429
430
431
432
433     PROCEDURE LIFT ;
434
435     BEGIN
436        stall:=0 ;
437        CL:=CL0+A1*alpha ;
438        writeln (CL) ;
439        writeln (alpha) ;
440
441        IF CL>=1.2 THEN
442           BEGIN
443             CL:=0 ;
444             stall:=1
445           END ;
446        LW:=0.5*rho*UWC2*surface*CL ;
447        LQ:=(-1.0)*rho*surface*UWC*LT*ZQ*Q
448
449     END ; (* of procedure LIFT *)
450
451     PROCEDURE DRAG ;
452
453     BEGIN
454       CD:=CD0+K*CL*CL ;
455       dragforce:=0.5*rho*UWC2*surface*CD
456     END ; (* of procedure DRAG *)
457
458     PROCEDURE SIDE ;
459
460     BEGIN
461
```

```
462
463
464       YA:=rho*surface*UVWC*(YV*VC+0.5*B*YR*R+UVWC*YTAU*tau)
465     END ; (* of procedure SIDE *)
466
467
468     PROCEDURE ROLL ;
469
470     BEGIN
471       LA:=0.5*rho*UVWC*B*surface*
472            (0.5*B*(LP*P+LR*R*CL)+LV*VC+UVWC*LS*zeta)
473     END ; (* of procedure ROLL *)
474
475     PROCEDURE PITCH ;
476
477     BEGIN
478       CMW:=CM0+CML*CL ;
479       IF stall=1 THEN CMW:=-0.3 ;
480       CMT:=CMETA*eta ;
481       taillift:=(CMT-CMWBD)*0.5*rho*UWC2*surface*CBAR/LT+LQ ;
482       MA:=0.5*rho*UWC2*(CMW+CMT)*CBAR*surface
483     END ; (* of procedure PITCH *)
484
485     PROCEDURE YAW ;
486
487     BEGIN
488       NR:=NR0+NRL*CL*CL ;
489       NA:=0.5*rho*surface*UVWC*B*
490            (NV*VC+0.5*B*NR*R+UVWC*NTAU*tau)
491     END ; (* of procedure YAW *)
492
493     PROCEDURE THRUST ;
494
495     VAR TDOT,expression : real ;
496
497     BEGIN
498       pnom:=pmax*throttle ;
499       pact:=pnom*ETAP ;
500       TDOT:=(pact-XE*U2)/KE ;
501       XE:=XE+TDOT*tstep ;
502       expression:=(2.0*XE)/(rho*AD)+UC*UC ;
503       U3:=SQRT(expression) ;
504       U2:=XE/(rho*AD*(U3-UC)) ;
505
506        writeln (expression:8:4,' ',UC:8:4,' ',U2:8:4,U3:8:4) ;
507       rps:=U2/PP ;
508       rpm:=rps*60.0 ;
509       IF rps=0.0 THEN LE:=0.0
510           ELSE LE:=pnom/(2.0*pi*rps) ;
511     LE:=0.0
512     END ; (* of procedure THRUST *)
513
514     PROCEDURE FORSUM ;
515
516     BEGIN
517       ltotal:=LW+taillift ;
518       X:=XE-dragforce*COS(alpha)+ltotal*SIN(alpha)-
519              mass*gravity*SIN(theta) ;
520       Y:=YA+mass*gravity*COS(theta)*SIN(phi) ;
521       Z:=(-1.0)*ltotal*COS(alpha)-dragforce*SIN(alpha) ;
522       Z:=Z+mass*gravity*COS(theta)*COS(phi) ;
523
524       L:=LE+LA ;
525       M:=MA+LW*(CG-0.25)*CBAR-(LT+(0.25-CG)*CBAR)*LQ ;
526       N:=NA
527
```

```
528
529
530   END ; (* of procedure FORSUM *)
531
532   PROCEDURE RATES ;
533
534   BEGIN
535
536     P:=P+PDOT*tstep ;
537     Q:=Q+QDOT*tstep ;
538     R:=R+RDOT*tstep ;
539
540     PDEG:=P*180.0/pi ;
541     QDEG:=Q*180.0/pi ;
542     RDEG:=R*180.0/pi
543
544   END ; (* of procedure RATES *)
545
546   PROCEDURE ATTITUDE ;
547
548   FUNCTION TAN (angle : real) : real ;
549      BEGIN
550         TAN:=SIN(angle)/COS(angle)
551       END ;
552
553   BEGIN
554     PHIDOT:=P+SIN(phi)*TAN(theta)*Q+COS(phi)*TAN(theta)*R ;
555     THETADOT:=COS(phi)*Q-SIN(phi)*R ;
556     PSIDOT:=SIN(phi)*Q/COS(theta)+R*COS(phi)/COS(theta) ;
557
558     phi:=phi+PHIDOT*tstep ;
559     theta:=theta+THETADOT*tstep ;
560     psi:=psi+PSIDOT*tstep ;
561     phideg:=phi*180.0/pi ;
562     thetadeg:=theta*180.0/pi ;
563     psideg:=psi*180.0/pi
564
565   END ; (* of procedure ATTITUDE *)
566
567   PROCEDURE TRANSF ;
568
569   VAR I,J : integer ;
570
571   BEGIN
572
573     transform[1,1]:=COS(theta)*COS(psi) ;
574     transform[1,2]:=SIN(phi)*SIN(theta)*COS(psi)-COS(phi)*SIN(psi) ;
575     transform[1,3]:=COS(phi)*SIN(theta)*COS(psi)+SIN(phi)*SIN(psi) ;
576     transform[2,1]:=COS(theta)*SIN(psi) ;
577     transform[2,2]:=SIN(phi)*SIN(theta)*SIN(psi)+COS(phi)*COS(psi) ;
578     transform[2,3]:=COS(phi)*SIN(theta)*SIN(psi)-SIN(phi)*COS(psi) ;
579     transform[3,1]:=(-1.0)*SIN(theta) ;
580     transform[3,2]:=SIN(phi)*COS(theta) ;
581     transform[3,3]:=COS(phi)*COS(theta) ;
582
583     FOR I:=1 TO 3 DO
584       BEGIN
585         FOR J:=1 TO 3 DO
586            transpose[I,J]:=transform[J,I]
587       END
588
589   END ; (* of procedure TRANSF *)
590
591   PROCEDURE RESOLVE ;
592
593
```

```
594
595
596   BEGIN
597     UB:=UB+UBDOT*tstep ;
598     VB:=VB+VBDOT*tstep ;
599     WB:=WB+WBDOT*tstep ;
600
601     UG:=transform[1,1]*UB+transform[1,2]*VB+transform[1,3]*WB ;
602     VG:=transform[2,1]*UB+transform[2,2]*VB+transform[2,3]*WB ;
603     WG:=transform[3,1]*UB+transform[3,2]*VB+transform[3,3]*WB
604
605   END ; (* of procedure RESOLVE *)
606
607   PROCEDURE POSN ;
608
609   BEGIN
610     XN:=XN+UG*tstep ;
611     YE:=YE+VG*tstep ;
612     height:=height-WG*tstep ;
613     writeln (XN:8:4) ;
614     writeln (YE:8:4) ;
615     writeln (height:8:4) ;
616     writeln (UG:8:4,' ',VG:8:4,' ',WG:8:4)
617
618   END ; (* of procedure POSN *)
619
620   PROCEDURE FORCES ;
621
622   BEGIN
623     LIFT  ;
624     DRAG  ;
625     SIDE  ;
626     ROLL  ;
627     PITCH ;
628     YAW   ;
629     THRUST ;
630     FORSUM
631   END ; (* of procedure FORCES *)
632
633   PROCEDURE INTEGRATE ;
634
635   BEGIN
636     RATES   ;
637     ATTITUDE ;
638     TRANSF ;
639     RESOLVE ;
640     POSN
641   END ; (* of procedure INTEGRATE *)
642
643
644   PROCEDURE MAINSIM ;
645
646   VAR I : integer ;
647       printinc : real ;
648
649   BEGIN
650     CONTRO ;
651     OUTDATA (1) ;
652     REPEAT
653       IF iter<>0 THEN
654         BEGIN
655           AIR ;
656           CONTRO ;
657           FORCES ;
658           ACCEL ;
659
```

```
660
661
662             INTEGRATE
663           END
664         ELSE
665           OUTDATA (2) ;
666         iter:=iter+1 ;
667         writeln (iter) ;
668         T:=iter*tstep ;
669         writeln (T) ;
670         printinc:=printinc+tstep ;
671         writeln (printinc) ;
672         writeln (stall) ;
673         IF printinc>= tprint THEN
674           BEGIN
675             OUTDATA (3) ;
676             printinc:=printinc-tprint
677           END ;
678       UNTIL (T>=tend) OR (stall=1)
679
680   END; (* of MAINSIM *)
681
682   BEGIN
683     INDATA ;
684     TRANSF ;
685     INIT   ;
686     AIR    ;
687       MAINSIM
688   END.
```

Appendix 2 :- BASIC Program Version 4 Listing

```
10    OPTION BASE 0
20    PRINTER IS 1
30    MASS STORAGE IS ":INTERNAL"
40    !
50  COM /Aero1/ REAL Mass,Ix,Iy,Iz,A1,CL0,Cleta,Zq,Surface
60  COM /Aero2/ REAL Lt,Coar,Cd0,K,Yv,Yr,Ytau,Lv,Ls,Lp,Lr
70  COM /Aero3/ REAL B,A1t,A2eta,Cm0,Cml,Cmmod,Cmeta
80  COM /Aero4/ REAL Nv,Nr0,Nr1,Ntau,Ac,Pmax,Etao,Ke,Pp
90  !
100 COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Hspeed,Hdrndeg
110 COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
120 !
130 COM /Arrays/ REAL Time(10),Thro_ttleset(10),Eta_degset(10)
140 COM /Arrays1/ REAL Zet_adegset(10),Tau_degset(10)
150 COM /Trans/ REAL T11,T12,T13,T21,T22,T23,T31,T32,T33
160 !
170 COM /Misc1/ REAL Hdrnrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
180 COM /Misc2/ REAL Eta_set(10),Zeta_set(10),Tau_set(10),INTEGER Breakpoints
190 COM /Misc3/ REAL T,Hgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
200 COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
210 COM /Misc5/ REAL Uwc2,Uwc,Uvwc2,Uvwc,Alpha,Beta,Ua,Va,Wa
220 COM /Misc6/ REAL X,Y,Z,L,M,N,Uboot,Vbdot,Wbdot,Pbot,Qdot,Rdot
230 COM /Misc7/ REAL Rho,Cl,Lw,Lq,Cd,Dragforce,Ya,La,Cat
240 COM /Misc8/ REAL Ma,Taillift,Ltotal,Cmw,Clt,Nr,Na,U2,Throttle
250 COM /Misc9/ REAL Xe,Le,Pact,Phom,U3,Ros,Rom,Gravity,Ugdot,Vgdot,Wgdot
260 COM /Misc10/ REAL Phidot,Thetadot,Psidot,INTEGER Iter,Stall
261 COM /Misc11/ Sintheta,Costheta,Sinpsi,Cospsi,Sinphi,Cosphi
270   !
280   COM /Lim/ Limits$[24],INTEGER Limits(0:3)
290   COM /Mess1/ Maxmin$[16],REAL Xyspec(0:3)
300   COM /Mess2/ REAL Tics(0:1),INTEGER Tics1(0:1)
310   !
320 Rerun:  LOADSUB ALL FROM "PREAMBLE"
330   CALL Preamble
340   DELSUB Preamble
850 LOADSUB ALL FROM "INITIALISE"
860 LOADSUB ALL FROM "MACH1"
870 CALL Incata
880 CALL Transf
890 CALL Init
900 CALL Air
910 DELSUB Incata,Init
920 LOADSUB ALL FROM "MACH2"
930 CALL Mainsim
940 !
950 DELSUB Air TO END
960 GOTO Rerun
970 !
990       END
```

```
10    SUB Preamble
20    !
30    COM /Lim/ Limits$[24],INTEGER Limits(0:3)
40    COM /Mess1/ Maxmin$[16],REAL Xyspec(0:3)
50    COM /Mess2/ REAL Tics(0:1),INTEGER Tics1(0:1)
60    INTEGER Rst
70    !
80    ON ERROR GOTO Reset
90    CREATE BDAT "STORE",10
100   OFF ERROR
110   ASSIGN @Write_data TO "STORE:INTERNAL";FORMAT OFF
120   OUTPUT @Write_data:0,1.1,0.0,1.1
130   OUTPUT @Write_data;133.0,0,100,5.,-1..50.,-30...1.1..9,4
140   ASSIGN @Write_data TO *
150   Rst=1
160 Reset:   OFF ERROR
170   !
180   ON ERROR GOTO Reset1
190   CREATE BDAT "XPAR",3
200   OFF ERROR
210   ASSIGN @Write_data TO "XPAR:INTERNAL";FORMAT OFF
220   OUTPUT @Write_data:1;2;"  TIME S  HEIGHT M";
230   ASSIGN @Write_data TO *
240   Rst=1
250 Reset1:   OFF ERROR
260   !
270 On_keys:   OUTPUT 2;CHR$(255)&"K";
280            PRINT TABXY(1,1)
290            PRINT USING """Select option from keys"""
300            PRINT USING """Note: Define plot channels and set up"""
310            PRINT USING """      graphics before running programme"",/"
320            PRINT USING """Default is latest defined values"""
330   !
340   ON KEY 0 LABEL " GSETUP " GOTO Setup
350   ON KEY 1 LABEL "  DEFCH " GOTO Defch
360   ON KEY 2 LABEL "  RUN   " GOTO Run
370   ON KEY 5 LABEL "  END   " GOTO End
380   !
390 Wait:   DISP "Choose option"
400         GOTO Wait
410         !
420   !
430 Defch:   OFF KEY
440          LOADSUB ALL FROM "XYCHAN"
450          CALL Setxy
460          DELSUB Setxy
470 Setup:   LOADSUB ALL FROM "AUTOPLT3"
480          CALL Gset
490          Rst=1
500          DELSUB Gset TO END
510          GOTO On_keys
520 Run:     OFF KEY
530          IF Rst THEN
540             LOADSUB ALL FROM "AUTOPLT3"
550             CALL Initg
560             DELSUB Gset TO END
570             Rst=0
580          END IF
581          SUBEXIT
582 End:     STOP
```

```
10  SUB Indata
20  !
30  COM /Aero1/ REAL Mass,Ix,Iy,Iz,A1,Cl0,Cleta,Zq,Surface
40  COM /Aero2/ REAL Lt,Cbar,Cd0,K,Yv,Yr,Ytau,Lv,Ls,Lp,Lr
50  COM /Aero3/ REAL B,A1t,A2eta,Cm0,Cma1,Cmabd,Cmeta
60  COM /Aero4/ REAL Nv,Nr0,Nr1,Ntau,Ad,Pmax,Etap,Ke,Pp
70  !
80  COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Wspeed,Wdrndeg
90  COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
100 !
110 COM /Arrays/ REAL Time(*),Thro_ttleset(*),Eta_degset(*)
120 COM /Arrays1/ REAL Zet_adegset(*),Tau_degset(*)
130 !
140 COM /Misc1/ REAL Wdrnrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
150 COM /Misc2/ REAL Eta_set(*),Zeta_set(*),Tau_set(*),INTEGER Breakpoints
160 !
170 INTEGER I
180 ASSIGN @Input_data1 TO "VAR2:INTERNAL";FORMAT OFF
190 ASSIGN @Input_data2 TO "VAR3:INTERNAL";FORMAT OFF
200 ASSIGN @Input_data TO "VAR1:INTERNAL";FORMAT OFF
210 !
220 ENTER @Input_data;Tend,Tstep,Tprint,Cg,Wspeed,Wdrndeg
230 ENTER @Input_data;Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
240 !
250 ENTER @Input_data2;Mass,Ix,Iy,Iz,A1,Cl0,Cleta,Zq,Surface
260 ENTER @Input_data2;Lt,Cbar,Cd0,K,Yv,Yr,Ytau,Lv,Ls,Lp,Lr
270 ENTER @Input_data2;B,A1t,A2eta,Cm0,Cma1,Cmabd,Cmeta
280 ENTER @Input_data2;Nv,Nr0,Nr1,Ntau,Ad,Pmax,Etap,Ke,Pp
290 !
300 I=0
310 !
320 ON END @Input_data1 GOTO Lab1
330 Repeat: ENTER @Input_data1;Time(I),Thro_ttleset(I)
340         ENTER @Input_data1;Eta_degset(I),Zet_adegset(I),Tau_degset(I)
350 I=I+1
360 GOTO Repeat
370 !
380 Lab1: OFF END @Input_data1
390       Breakpoints=I
400       Time(Breakpoints)=Tend
410       !
420       Wdrnrad=PI*Wdrndeg/180.0
430       Phi=PI*Phideg/180.0
440       Theta=PI*The_tadeg/180.0
450       Psi=PI*Psideg/180.0
460       !
470       ASSIGN @Input_data TO *
480       ASSIGN @Input_data1 TO *
490       ASSIGN @Input_data2 TO *
500       !
510       FOR I=0 TO Breakpoints
520          Eta_set(I)=Eta_degset(I)*PI/180.0
530          Zeta_set(I)=Zet_adegset(I)*PI/180.0
540          Tau_set(I)=Tau_degset(I)*PI/180.0
550          Etadeg=Eta_degset(0)
560          Zetadeg=Zet_adegset(0)
570          Taudeg=Tau_degset(0)
580       NEXT I
590 SUBEND
600 !
```

```
610 SUB Init
620 !
630 COM /Misc3/ REAL T,Wgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
640  COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
650  COM /Misc7/ REAL Rho,Cl,Lw,Lq,Cd,Dragforce,Ya,La,Cmt
651  COM /Misc9/ REAL Xe,Le,Pact,Pmax,U3,Rps,Rpm,Gravity,Ugdot,Vgdot,Wgdot
660  COM /Misc10/ REAL Phidot,Thetadot,Psidot,INTEGER Iter,Stall
670  !
680  COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
690  !
700  COM /Trans/ REAL T11,T12,T13,T21,T22,T23,T31,T32,T33
710  !
711  Stall=0
720  T=0.
730  Iter=0
740  P=0.
750  Q=0.
760  R=0.
770  Wgust=0.
780  Rho=1.18
790  Gravity=9.81
800  !
810  Ub=T11*Ug+T21*Vg+T31*Wg
820  Vb=T12*Ug+T22*Vg+T32*Wg
830  Wb=T13*Ug+T23*Vg+T33*Wg
840  !
850  SUBEND
860  !
```

```
10   SUB Outdata(INTEGER Option)
20   !
30   COM /Misc1/ REAL Radmrad,Phu,Theta,Psi,Etadeg,Zetadeg,Taudeg
40   COM /Misc3/ REAL T,Wgust,P,C,R,Pdeg,Qdeg,Rdeg,Xn,Ye
50   COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
60   !
70   COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Wspeed,Radmdeg
80   COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
90   !
100  INTEGER I
110  !
120  IF Option=1 THEN
130     PRINT USING Fmt1
140     PRINT USING Fmt2
150     PRINT USING Fmt3
160     PRINT USING Fmt2
170     PRINT USING Fmt1
180     PRINT USING Fmt4
190     PRINT USING "5X,""TEND = "",M4D.4D,/";Tend
200     PRINT USING "5X,""TSTEP = "",M4D.4D,/";Tstep
210     PRINT USING "5X,""TPRINT = "",M3D.2D";Tprint
220     PRINT USING Fmt4
230     PRINT USING "5X,""CG = "",M2D.4D,/";Cg
240     PRINT USING "5X,""ETA = "",M2D.4D,"" (DEG) "",/";Etadeg
250     PRINT USING "5X,""ZETA = "",M2D.4D,"" (DEG) "",/";Zetadeg
260     PRINT USING "5X,""TAU = "",M2D.4D,"" (DEG) "",/";Taudeg
270     PRINT USING Fmt4
280     PRINT USING "5X,""PHI = "",M2D.4D,"" (DEG) "",/";Phideg
290     PRINT USING "5X,""THETA = "",M2D.4D,"" (DEG) "",/";The_tadeg
300     PRINT USING "5X,""PSI = "",M2D.4D,"" (DEG) "",/";Psideg
310     PRINT USING Fmt4
320  ELSE
330     IF (Option=2) OR (Option=3) THEN
340        IF Option=2 THEN
350           PRINT USING Fmt4
370        ELSE
390           PRINT USING Fmt4
400           PRINT USING Fmt5;Ub,Vb,Wb,Vt,Uc,Vc,Wc
410           PRINT USING Fmt5a;Pdeg,Qdeg,Rdeg,Phideg,The_tadeg,Psideg
411           PRINT USING Fmt5b;Etadeg,Zetadeg,Taudeg,Height,Xn,Ye,Xe
420        END IF
430     END IF
440  END IF
450  !
460  Fmt1:  IMAGE "******************************"
470  Fmt2:  IMAGE "*                            *"
480  Fmt3:  IMAGE "* MACHAN SIMULATION PROGRAM *"
490  Fmt4:  IMAGE /
500  Fmt5:  IMAGE 1X,"TIME",3X,"YE",2X,"SPEED",3X,"XN SPEED"
510  Fmt5a:  IMAGE 2X,"HEADING",4X,"ROLL RATE",3X,"ROLL",/
520  Fmt6:  IMAGE 7(1X,M3D.5D),L
530  Fmt6a:  IMAGE 3(1X,M2D.6D),3(1X,M3D.5D),L
531  Fmt6b:  IMAGE 3(1X,M3D.5D),1X,M4D.4D,2(1X,M3D.6D),1X,M5D.3D
540  !
541  PRINTER IS 1
550  SUBEND
560  !
570  SUB Forces
571  !
580  COM /Aero1/ REAL Mass,Ix,Iy,Iz,A1,C10,Cleta,Zq,Surface
```

```
590  COM /Aero2/ REAL Lt,Cbar,Cd0,K,Yv,Yr,Ytau,Lv,Ls,Lp,Lr
600  COM /Aero3/ REAL B,A1t,A2eta,Cm0,Cmal,Cmaod,Cmeta
610  COM /Aero4/ REAL Nv,Nr0,Nr1,Ntau,Ad,Pmax,Etap,Ke,Pp
620  !
630  COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Wspeed,Wdrndeg
640  COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
650  !
660  COM /Trans/ REAL T11,T12,T13,T21,T22,T23,T31,T32,T33
670  !
680  COM /Misc1/ REAL Wdrnrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
690  COM /Misc2/ REAL Eta_set(*),Zeta_set(*),Tau_set(*),INTEGER Breakpoints
700  COM /Misc3/ REAL T,Wgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
710  COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
720  COM /Misc5/ REAL Uac2,Uac,Uwac2,Uwac,Alpha,Beta,Ua,Va,Wa
730  COM /Misc6/ REAL X,Y,Z,L,M,N,Ubdot,Vbdot,Wbdot,Pdot,Qdot,Rdot
740  COM /Misc7/ REAL Rho,Cl,Lw,Lq,Cd,Dragforce,Ya,La,Cmt
750  COM /Misc8/ REAL Ma,Taillift,Ltotal,Cmw,Clt,Nr,Na,U2,Throttle
760  COM /Misc9/ REAL Xe,Le,Pact,Phom,U3,Rps,Rpm,Gravity,Ugdot,Vgdot,Wgdot
770  COM /Misc10/ REAL Phidot,Thetadot,Psidot,INTEGER Iter,Stall
780  COM /Misc11/ REAL Sintheta,Costheta,Sinpsi,Cospsi,Sinphi,Cosphi
790  !
800  REAL Tdot,Expression,Const,Const1,Const2,Const3
810  REAL Cosalpha,Sinalpha,Tantheta
820  !
830  Cosalpha=COS(Alpha)
840  Sinalpha=SIN(Alpha)
850  Tantheta=TAN(Theta)
860  !
870  Const=Rho*Surface
880  Const1=Const*Cbar
890  Const2=Mass*Gravity
900  Const3=180/PI
910  !
920 Lift:  Stall=0
930        Cl=Cl0+A1*Alpha
940        IF Cl>=1.2 THEN
950           Cl=0.
960           Stall=1
970        END IF
980  !
990        Lw=.5*Uac2*Const*Cl
1000       Lq=(-1.0)*Const*Uac*Lt*Zq*Q
1010 !
1020 Drag: Cd=Cd0+K*Cl*Cl
1030       Dragforce=.5*Uac2*Const*Cd
1040 !
1050 Side: Ya=Const*Uwac*(Yv*Vc+.5*B*Yr*R+Uwac*Ytau*Tau)
1060 !
1070 Roll: La=.5*Uwac*B*Const*(.5*B*(Lp*P+Lr*R*Cl)+Lv*Vc+Uwac*Ls*Zeta)
1080 !
1090 Pitch: Cmw=Cm0+Cmal*Cl
1100        IF Stall=1 THEN
1110           Cmw=-.3
1120        END IF
1130        Cmt=Cmeta*Eta
1140        Taillift=(Cmt-Cmaod)*.5*Uac2*Const1/Lt+Lq
1150        Ma=.5*Uac2*(Cmt+Cmw)*Const1
1160 !
1170 Yaw: Nr=Nr0+Nr1*Cl*Cl
1180      Na=.5*Const*Uwac*B*(Nv*Vc+.5*B*Nr*R+Uwac*Ntau*Tau)
```

```
1190 !
1200 Thrust: Pnom=Pmax*Throttle
1210         Pact=Pnom*Etap
1220         Tdot=(Pact-Xe*U2)/Ke
1230         Xe=Xe+Tdot*Tstep
1240         Expression=(2.0*Xe)/(Rho*Ad)+Uc*Uc
1250         U3=SQR(Expression)
1260         U2=Xe/(Rho*Ad*(U3-Uc))
1270         Rps=U2/Pp
1280         Rpm=Rps*60.0
1290         IF Rps=0. THEN
1300            Le=0.
1310         ELSE
1320            Le=Pnom/(2.0*PI*Rps)
1330            Le=0.
1340         END IF
1350 !
1360 Forsum: Ltotal=Lw+Taillift
1370         X=Xe-Dragforce*Cosalpha+Ltotal*Sinalpha-Const2*Sintheta
1380         Y=Ya+Const2*Costheta*Sinphi
1390         Z=(-1.0)*Ltotal*Cosalpha-Dragforce*Sinalpha
1400         Z=Z+Const2*Costheta*Cosphi
1410 !
1420         L=Le+La
1430         M=Ma+Lw*(Cg-.25)*Cbar-(Lt+(.25-Cg)*Cbar)*Lo
1440         N=Na
1450 !
1460 Accel: Ubdot=X/Mass-Wb*Q+Vb*R
1470        Vbdot=Y/Mass-Ub*R+Wb*P
1480         Wbdot=Z/Mass-Vb*P+Ub*Q
1490 !
1500         Pdot=(L+(Iy-Iz)*R*Q)/Ix
1510         Qdot=(M+(Iz-Ix)*P*R)/Iy
1520         Rdot=(N+(Ix-Iy)*P*Q)/Iz
1530 !
1540 Rates: P=P+Pdot*Tstep
1550        Q=Q+Qdot*Tstep
1560        R=R+Rdot*Tstep
1570 !
1580        Pdeg=P*Const3
1590        Qdeg=Q*Const3
1600        Rdeg=R*Const3
1610 !
1620 Attitude: Phidot=P+Sinphi*Tantheta*Q+Cosphi*Tantheta*R
1630           Thetadot=Cosphi*Q-Sinphi*R
1640           Psidot=Sinphi*Q/Costheta+R*Cosphi/Costheta
1650 !
1660           Phi=Phi+Phidot*Tstep
1670           Theta=Theta+Thetadot*Tstep
1680           Psi=Psi+Psidot*Tstep
1690           Phideg=Phi*Const3
1700           The_tadeg=Theta*Const3
1710           Psideg=Psi*Const3
1720 !
1730 CALL Transf
1740 !
1750 Resolve: Ub=Ub+Ubdot*Tstep
1760          Vb=Vb+Vbdot*Tstep
1770          Wb=Wb+Wbdot*Tstep
1780 !
```

```
1790          Ug=T11*Ub+T12*Vb+T13*Wb
1800          Vg=T21*Ub+T22*Vb+T23*Wb
1810          Wg=T31*Ub+T32*Vb+T33*Wb
1820 !
1830 Posn: Xn=Xn+Ug*Tstep
1840       Ye=Ye+Vg*Tstep
1850       Height=Height-Wg*Tstep
1860 !
1870 SUBEND
1880 !
1890 SUB Contro
1891 !
1892 COM /Arrays/ REAL Time(*),Thro_ttleset(*),Eta_degset(*)
1893 COM /Arrays1/ REAL Zet_adegset(*),Tau_degset(*)
1900 !
1910 COM /Misc1/ REAL Wdrnrad,Phu,Theta,Psi,Etadeg,Zetadeg,Taudeg
1911 COM /Misc2/ REAL Eta_set(*),Zeta_set(*),Tau_set(*),INTEGER Breakpoints
1920 COM /Misc3/ REAL T,Wgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
1930 COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
1940 COM /Misc8/ REAL Ma,Tailift,Ltotal,Cmw,Clt,Mr,Na,U2,Throttle
1950 !
1960 INTEGER I,Tmatch
1961 !
1962 I=0
1963 Tmatch=0
1964 !
1965 Test1: IF (Tmatch=1) OR (I>Breakpoints-1) THEN
1966          SUBEXIT
1967       ELSE
1968          IF (T>=Time(I)) AND (T<Time(I+1)) THEN
1969             Throttle=Thro_ttleset(I)
1970             Eta=Eta_set(I)
1971             Zeta=Zeta_set(I)
1972             Tau=Tau_set(I)
1973             Etadeg=Eta_degset(I)
1974             Zetadeg=Zet_adegset(I)
1975             Taudeg=Tau_degset(I)
1976             Tmatch=1
1977          ELSE
1978             I=I+1
1979          END IF
1980       END IF
1981    GOTO Test1
1983 !
1984 SUBEND
```

```
10   SUB Air
20   !
30   COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Wspeed,Wdirdeg
40   COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
50   !
60   COM /Trans/ REAL T11,T12,T13,T21,T22,T23,T31,T32,T33
70   !
80   COM /Misc1/ REAL Wdirrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
90   COM /Misc3/ REAL T,Wgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
100  COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
110  COM /Misc5/ REAL Uwc2,Uwc,Uvwc2,Uvwc,Alpha,Beta,Ua,Va,Wa
120  !
130  REAL Uw,Vw,Ww,Uvwa2
140  REAL Oppalpha,Adjalpha,Tanalpha
150  REAL Oppbeta,Adjbeta,Tanbeta
160  !
170  Uw=Wspeed*COS(Wdirrad)
180  Vw=Wspeed*SIN(Wdirrad)
190  Ww=Wgust
200  !
210  Ua=Ug+Uw
220  Va=Vg+Vw
230  Wa=Wg+Ww
240  !
250  Uc=T11*Ua+T21*Va+T31*Wa
260  Vc=T12*Ua+T22*Va+T32*Wa
270  Wc=T13*Ua+T23*Va+T33*Wa
280  !
290  Uvwa2=Ua*Ua+Va*Va+Wa*Wa
300  Vt=SQR(Uvwa2)
310  Uwc2=Uc*Uc+Wc*Wc
320  Uwc=SQR(Uwc2)
330  Uvwc2=Uwc2+Vc*Vc
340  Uvwc=SQR(Uvwc2)
350  Oppalpha=(Wc/SQR(Uwc2))
360  Adjalpha=1-Oppalpha*Oppalpha
370  Adjalpha=SQR(Adjalpha)
380  Tanalpha=Oppalpha/Adjalpha
390  Alpha=ATN(Tanalpha)
400  Oppbeta=Vc/Vt
410  Adjbeta=1-Oppbeta*Oppbeta
420  Adjbeta=SQR(Adjbeta)
430  Tanbeta=Oppbeta/Adjbeta
440  Beta=ATN(Tanbeta)
450  !
460  SUBEND
470  !
480  SUB Transf
490  !
500  COM /Misc1/ REAL Wdirrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
510  COM /Trans/ REAL T11,T12,T13,T21,T22,T23,T31,T32,T33
511  COM /Misc11/ REAL Sintheta,Costheta,Sinpsi,Cospsi,Sinphi,Cosphi
520  !
530  INTEGER I,J
532  !
533  Sintheta=SIN(Theta)
534  Costheta=COS(Theta)
535  Sinpsi=SIN(Psi)
536  Cospsi=COS(Psi)
537  Sinphi=SIN(Phi)
```

```
530  Cosphi=COS(Phi)
540  !
550  T11=Costheta*Cospsi
560  T12=Sinphi*Sintheta*Cospsi-Cosphi*Sinpsi
570  T13=Cosphi*Sintheta*Cospsi+Sinphi*Sinpsi
580  T21=Costheta*Sinpsi
590  T22=Sinphi*Sintheta*Sinpsi+Cosphi*Cospsi
600  T23=Cosphi*Sintheta*Sinpsi-Sinphi*Cospsi
610  T31=-Sintheta
620  T32=Sinphi*Costheta
630  T33=Cosphi*Costheta
640  !
710  SUBEND
720  !
730  SUB Mainsim
740  !
750  COM /Aero1/ REAL Mass,Ix,Iy,Iz,A1,CL0,Cleta,Zq,Surface
760  COM /Aero2/ REAL Lt,Cbar,CdO,K,Yv,Yr,Ytau,Lv,Ls,Lp,Lr
770  COM /Aero4/ REAL Nv,NrO,Nr1,Ntau,Ad,Pmax,Etap,Ke,Pp
780  COM /Aero3/ REAL B,A1t,A2eta,CmO,Cml,Cmabd,Cmeta
790  !
800  COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Wspeed,Wdirndeg
810  COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
820  !
830  COM /Arrays/ REAL Time(*),Thro_ttleset(*),Eta_degset(*)
840  COM /Arrays1/ REAL Zet_adegset(*),Tau_degset(*)
850  COM /Trans/ REAL T11,T12,T13,T21,T22,T23,T31,T32,T33
860  !
870  COM /Misc1/ REAL Wdirnrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
880  COM /Misc2/ REAL Eta_set(*),Zeta_set(*),Tau_set(*),INTEGER Breakpoints
890  COM /Misc3/ REAL T,Wgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
900  COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
910  COM /Misc5/ REAL Uwc2,Uwc,Uwwc2,Uwwc,Alpha,Beta,Ua,Va,Wa
920  COM /Misc6/ REAL X,Y,Z,L,M,N,Ubdot,Vbdot,Wbdot,Pdot,Qdot,Rdot
930  COM /Misc7/ REAL Rho,Cl,Lw,Lq,Cd,Dragforce,Ya,La,Cmt
940  COM /Misc8/ REAL Ma,Taillift,Ltotal,Cmw,Clt,Nr,Na,U2,Throttle
950  COM /Misc9/ REAL Xe,Le,Pact,Pnom,U3,Rps,Rom,Gravity,Ugdot,Vgdot,Wgdot
960  COM /Misc10/ REAL Phidot,Thetadot,Psidot,INTEGER Iter,Stall
961  !
963  ALLOCATE REAL Xyco_ord(1:10)
970  !
980  INTEGER I,Xchan,Ychan
990  REAL Printinc
991  !
992  ASSIGN @Read_data TO "XPAR:INTERNAL";FORMAT OFF
993  ENTER @Read_data;Xchan,Ychan
994  ASSIGN @Read_data TO *
995  !
997  CALL Contro
998  CALL Outdata(1)
999  !
1000 Repeat: IF (T>=Tend) OR (Stall=1) THEN
1010           SUBEXIT
1020         ELSE
1030          IF Iter<>0 THEN
1040            CALL Air
1050            CALL Contro
1060            CALL Forces
1090          ELSE
1100            CALL Outdata(2)
```

```
1101          END IF
1110            Iter=Iter+1
1120            T=Iter*Tstep
1130            Printinc=Printinc+Tstep
1132            Xyco_ord(1)=T
1133            Xyco_ord(2)=Height
1134            Xyco_ord(3)=Xn
1135            Xyco_ord(4)=Qdeg
1136            Xyco_ord(5)=The_tadeg
1137            Xyco_ord(6)=Psdeg
1138            Xyco_ord(7)=Phideg
1139            Xyco_ord(8)=Etadeg
1140            Xyco_ord(9)=Zetadeg
1141            Xyco_ord(10)=Vt
1142            IF Iter=1 THEN
1143               PEN -1
1144            ELSE
1145               PEN 1
1146            END IF
1147            DRAW Xyco_ord(Xchan),Xyco_ord(Ychan)
1148              IF Printinc>=Tprint THEN
1150                CALL Outdata(3)
1160                Printinc=Printinc-Tprint
1170              END IF
1190          END IF
1200 GOTO Repeat
1210 !
1220 SUBEND
```

```
10    SUB Setxy
11    DIM Xy$[2]
20    OFF KEY
30    ALLOCATE Xyident$[18].INTEGER Xychan(0:1)
31    Xy$="XY"
40    !
50    OUTPUT 2:CHR$(255)&"K";
60    !
70    FOR I=0 TO 1
80      ON KEY 0 LABEL "  TIME  " GOTO Time
90      ON KEY 1 LABEL " HEIGHT " GOTO Height
100     ON KEY 2 LABEL " H DIST " GOTO Xdist
110     ON KEY 3 LABEL " P RATE " GOTO Pitchr
120     ON KEY 4 LABEL "P ANGLE " GOTO Pitcha
130     ON KEY 5 LABEL " R RATE " GOTO Rollr
140     ON KEY 6 LABEL "R ANGLE " GOTO Rolla
150     ON KEY 7 LABEL "ELV ANG " GOTO Elv
160     ON KEY 8 LABEL "AIL ANG " GOTO Ail
170     ON KEY 9 LABEL " SPEED  " GOTO Speed
180     !
190 Wait1:    DISP USING """Select "",A,"" channel""";Xy$[I+1]
200           GOTO Wait1
210     !
220 Time:     Xychan(I)=1
230           Xyident$[1+9*I]="  TIME S "
240           GOTO Fin
250 Height:   Xychan(I)=2
260           Xyident$[1+9*I]=" HEIGHT M"
270           GOTO Fin
280 Xdist:    Xychan(I)=3
290           Xyident$[1+9*I]="HOR DIS M"
300           GOTO Fin
310 Pitchr:   Xychan(I)=4
320           Xyident$[1+I*9]="PCHRT D/S"
330           GOTO Fin
340 Pitcha:   Xychan(I)=5
350           Xyident$[1+I*9]="P ANG DEG"
360           GOTO Fin
370 Rollr:    Xychan(I)=6
380           Xyident$[1+I*9]="ROLRT D/S"
390           GOTO Fin
400 Rolla:    Xychan(I)=7
410           Xyident$[1+I*9]="R ANG DEG"
420           GOTO Fin
430 Elv:      Xychan(I)=8
440           Xyident$[1+I*9]="ELVANG DG"
450           GOTO Fin
460 Ail:      Xychan(I)=9
470           Xyident$[1+I*9]="AILANG DG"
480           GOTO Fin
490 Speed:    Xychan(I)=10
500           Xyident$[1+I*9]="SPD M/S  "
510 !
520 Fin: NEXT I
530     OFF KEY
540 !
550 ASSIGN @Save_data TO "XPAR:INTERNAL";FORMAT OFF
560 OUTPUT @Save_data;Xychan(*),Xyident$
570 ASSIGN @Save_data TO *
580 SUBEND
```

```
10    SUB Gset
20    !
50 COM /Lim/ Limits$[24],INTEGER Limits(0:3)
70 COM /Mess1/ Maxmin$[16],REAL Xyspec(0:3)
80 COM /Mess2/ REAL Tics(0:1),INTEGER Tics1(0:1)
90 DIM On_off$[6],Axislab$[18]
100 INTEGER Ind1,Ind2,Ind3,Ind4,Dummy
110 INTEGER Ind5,Ind6,Ind7,Ind8,Xmitic,Ymitic
120 REAL Xmax,Xmin,Ymax,Ymin,Xmjtic,Ymjtic
130!
140 !
142         ASSIGN @Read_data TO "STORE:INTERNAL";FORMAT OFF
143         ENTER @Read_data;Ind1,Ind2,Ind3,Ind4,Ind5,Ind6,Ind7
144         ENTER @Read_data;Limits(*),Xyspec(*),Tics(*),Tics1(*)
145         ASSIGN @Read_data TO *
146 !
147 On_off$="OFFON "
150 Limit$="   TOPBOTTOM LEFT RIGHT"
160 Maxmin$="XMAXXMINYMAXYMIN"
170!
180!
190 Retype: OUTPUT 2;CHR$(255)&"K";
200 Retype1: PRINT TABXY(21,1),"AUTO PLOT"
210 PRINT TAB(21),"*********"
220 PRINT USING "/"
230 PRINT USING """SOFT CLIP"",2X,3A,1X,#";On_off$[1+3*Ind1]
240 PRINT USING """TOP = "",M3D,1X,""BOTTOM = "",M3D";Limits(0),Limits(1)
250 PRINT USING "14X,""LEFT = "",M3D,2X,""RIGHT = "",M3D,/";Limits(2),Limits(3)
260 PRINT USING """X AXIS XMAX = "",M4D.4D,1X,""XMIN = "",M4D.4D";Xyspec(0),Xyspec(1)
270 PRINT USING "2X,""TICS "",3A,1X,#";On_off$[1+3*Ind2]
280 PRINT USING """ DIST. BETWEEN TIC MARKS = "",4D.5D";Tics(0)
290 PRINT USING "18X,""NO. OF MINOR TICS = "",4D,/";Tics1(0)
300 PRINT USING """Y AXIS YMAX = "",M4D.4D,1X,""YMIN = "",M4D.4D";Xyspec(2),Xyspec(3)
310 PRINT USING "2X,""TICS "",3A,1X,#";On_off$[1+3*Ind3]
320 PRINT USING """ DIST. BETWEEN TIC MARKS = "",4D.5D";Tics(1)
330 PRINT USING "18X,""NO. OF MINOR TICS = "",4D,/";Tics1(1)
340 PRINT USING "2X,""GRID "",3A,20X,""FRAME "",3A,/";On_off$[1+3*Ind4],On_off$[1+3*Ind5]
350 PRINT USING "2X,""PEN TYPE (1 thru 10) "",2D,5X,#";Ind6
360 PRINT USING """LABLES "",3A";On_off$[1+3*Ind7]
370 !
380 OFF KEY
390 !
400 ON KEY 0 LABEL "  CLIP  " GOTO Clip
410 ON KEY 1 LABEL "  DEFX  " GOTO Defx
420 ON KEY 2 LABEL "  DEFY  " GOTO Defy
430 ON KEY 3 LABEL "  GRID  " GOTO Grid
440 ON KEY 4 LABEL " FRAME  " GOTO Frame
450 ON KEY 5 LABEL "   PEN  " GOTO Pen
460 ON KEY 6 LABEL " LABEL  " GOTO Label
470 ON KEY 9 LABEL "  EXIT  " GOTO Exit
480 !
490 Wait1: DISP "Select Key"
500        GOTO Wait1
510 !
520 Clip: !
530 !
540       OFF KEY
550       OUTPUT 2;CHR$(255)&"K";
560       PRINT TABXY(1,2)
570       PRINT USING """SOFT CLIP IS "",3A,"" TOGGLE ?"",/";On_off$[1+3*Ind1]
```

```
580       !
590       CAL. Yes_no(Ind1,Ans$)
600       IF NOT (Ind1) THEN Retype
610       PRINT "DEFINE LIMITS ? "
620       Ind8=1
630       CALL Yes_no(Ind8,Ans$)
640       IF Ind8 THEN Retype
650 Rpt2:   ON KEY 0 LABEL "   TOP   " GOTO Top
660         ON KEY 1 LABEL " BOTTOM " GOTO Btm
670         ON KEY 2 LABEL "  LEFT   " GOTO Left
680         ON KEY 3 LABEL " RIGHT   " GOTO Right
690         ON KEY 9 LABEL "  EXIT   " GOTO Retype
700 Wait2: DISP "Select Key"
710        GOTO Wait2
720 !
730 Top: CALL Prnt_mess(0)
740      GOTO Rpt2
750 Btm: CALL Prnt_mess(1)
760      GOTO Rpt2
770 Left: CALL Prnt_mess(2)
780       GOTO Rpt2
790 Right: CALL Prnt_mess(3)
800        GOTO Rpt2
810 !
820 Grid: Ind4=NOT (Ind4)
830       GOTO Retype1
840 !
850 Frame: Ind5=NOT (Ind5)
860        GOTO Retype1
870 !
880 Label: Ind7=NOT (Ind7)
890        GOTO Retype1
900 !
910 Defx: !
920       OFF KEY
930       OUTPUT 2;CHR$(255)&"K";
940       PRINT TABXY(1,2)
950 !
960 Rpt3:  ON KEY 0 LABEL "  XMAX   " GOTO Xmax
970        ON KEY 1 LABEL "  XMIN   " GOTO Xmin
980        ON KEY 2 LABEL "  TICS   " GOTO Tics
990        ON KEY 3 LABEL " XMAJTIC" GOTO Xmaj
1000       ON KEY 4 LABEL " XMINTIC" GOTO Xmintic
1010       ON KEY 9 LABEL "  EXIT   " GOTO Retype
1020!
1030 Wait3:DISP "Select key"
1040       GOTO Wait3
1050 Xmax:CALL Prnt_mess1(0)
1060      GOTO Rpt3
1070 Xmin:CALL Prnt_mess1(1)
1080      GOTO Rpt3
1090 Tics:Ind2=NOT (Ind2)
1100      PRINT USING """TICS ARE "",3A":On_off$[1+3*Ind2]
1110      GOTO Rpt3
1120 Xmaj:CALL Prnt_mess2(0)
1130      GOTO Rpt3
1140 Xmintic:CALL Prnt_mess3(0)
1150         GOTO Rpt3
1160!
1170 Defy:!
```

```
1180       OFF KEY
1190       OUTPUT 2:CHR$(255)&"K";
1200       PRINT TABXY(1,2)
1210 !
1220 Rpt4: ON KEY 0 LABEL "  YMAX  " GOTO Ymax
1230       ON KEY 1 LABEL "  YMIN  " GOTO Ymin
1240       ON KEY 2 LABEL "  TICS  " GOTO Ticsy
1250       ON KEY 3 LABEL " YMAJTIC" GOTO Ymaj
1260       ON KEY 4 LABEL " YMINTIC" GOTO Ymintic
1270       ON KEY 9 LABEL "  EXIT  " GOTO Retype
1280 !
1290 Wait4: DISP "Select key"
1300        GOTO Wait4
1310 !
1320 Ymax: CALL Prmt_mess1(2)
1330       GOTO Rpt4
1340 Ymin: CALL Prmt_mess1(3)
1350       GOTO Rpt4
1360 Ticsy: Ind3=NCT (Ind3)
1370        PRINT USING """TICS ARE "".3A";On_off$[1+3*Ind3]
1380        GOTO Rpt4
1390 Ymaj: CALL Prmt_mess2(1)
1400       GOTO Rpt4
1410 Ymintic: CALL Prmt_mess3(1)
1420          GOTO Rpt4
1430 !
1440 Pen: !
1450      OFF KEY
1460      OUTPUT 2;CHR$(255)&"K";
1470      PRINT TABXY(1,2)
1480 !
1490 PRINT USING """PRESENT PEN IS "",ZD,"" NEW PEN IS "",#";Ind6
1500 Rpt5: INPUT "1 Thru 10",Ind6
1510       IF Ind6>10 OR Ind6<1 THEN
1520          BEEP 976.56,.3
1530          DISP "Rubbish"
1540          WAIT 1
1550          GOTO Rpt5
1560       END IF
1570 PRINT USING "ZD";Ind6
1580 GOTO Retype
1590 !
1600 Exit:OFF KEY
1610      OUTPUT 2;CHR$(255)&"K";
1625      PURGE "STORE:INTERNAL"
1630      CREATE BDAT "STORE",10
1631      !
1640      ASSIGN @Write_data TO "STORE:INTERNAL";FORMAT OFF
1650      OUTPUT @Write_data;Ind1,Ind2,Ind3,Ind4,Ind5,Ind6,Ind7
1660      OUTPUT @Write_data;Limits(*),Xyspec(*),Tics(*),Tics1(*)
1670      ASSIGN @Write_data TO *
1671      CALL Initg
1672      WAIT 2
1674      DISP "O.K. ?"
1675      WAIT 2
1676      Dummy=1
1677      CALL Yes_no(Dummy,Ans$)
1678      GRAPHICS OFF
1680      IF Dummy THEN GOTO Retype
1684 SUBEND
```

```
1690!
1700 SUB Yes_no(INTEGER Ind,Ans$)
1710!
1720 Rpt1:INPUT "(Y/N)",Ans$
1730      IF Ans$="Y" THEN
1740         Ind=NOT (Ind)
1750         GOTO End
1760      ELSE
1770         IF Ans$<>"N" THEN
1780            BEEP 976.56,.3
1790            DISP "Rubbish"
1800            WAIT 1
1810            GOTO Rpt1
1820         END IF
1830      END IF
1840 End: SUBEND
1850!
1860 SUB Prnt_mess(INTEGER I)
1870 !
1880 COM /Lim/ Limit$[24],INTEGER Limits(*)
1890 PRINT USING "6A,"" LIMIT IS "",M3D,"" NEW VALUE IS "",#";Limit$[1+6*I],Limits(I)
1900 INPUT Limits(I)
1910 PRINT USING "Y3D";Limits(I)
1920 SUBEND
1930 !
1940 SUB Prnt_mess1(INTEGER I)
1950 !
1960 COM /Mess1/ Maxmin$[16],REAL Xyspec(*)
1970 PRINT USING "1X,4A,"" IS "",M4D.4D,"" NEW VALUE IS "",#";Maxmin$[1+4*I],Xyspec(I)
1980 INPUT Xyspec(I)
1990 PRINT USING "M4D.4D";Xyspec(I)
2000 SUBEND
2010 !
2020 SUB Prnt_mess2(INTEGER I)
2030 !
2040 COM /Mess2/ REAL Tics(*),INTEGER Tics1(*)
2050 PRINT USING """TIC INTERVAL IS "",4D.5D,"" NEW VALUE "",#";Tics(I)
2060 INPUT "WARNING: Must be >0 !",Tics(I)
2070 PRINT USING "4D.5D";Tics(I)
2080 SUBEND
2090 !
2100 SUB Prnt_mess3(INTEGER I)
2110 !
2120 COM /Mess2/ REAL Tics(*),INTEGER Tics1(*)
2130 PRINT USING """NO. OF MINOR TICS IS "",4D,"" NEW VALUE "",#";Tics1(I)
2140 INPUT "Must be integer >=0 ",Tics1(I)
2150 PRINT USING "4D";Tics1(I)
2160 SUBEND
2170 !
2180 SUB Initg
2230  !
2231  ALLOCATE Xyident$[18],INTEGER Xychan(0:1)
2240  INTEGER Ind1,Ind2,Ind3,Ind4,Ind5,Ind6,Ind7
2250  INTEGER Tics1(0:1),Limits(0:3),Mntic(0:1)
2260  INTEGER I,J,K,L,M,Mucotx,Mucnty
2270  REAL Xyspec(0:3),Tics(0:1),X,Y
2280  !
2290  ASSIGN @Read_data TO "STORE:INTERVAL";FORMAT OFF
2300  ENTER @Read_data;Ind1,Ind2,Ind3,Ind4,Ind5,Ind6,Ind7
2310  ENTER @Read_data;Limits(*),Xyspec(*),Tics(*),Tics1(*)
```

TEXT BOUND INTO

THE SPINE

```
ASSIGN @Read_data TO *
!
ASSIGN @Read_data TO "XPAR:INTERNAL";FORMAT OFF
ENTER @Read_data;Xychan(*),Xyident$
ASSIGN @Read_data TO *
!
GINIT
GRAPHICS ON
CSIZE 4,.75
!
IF NOT (Ind1) THEN
  CLIP OFF
  VIEWPORT 0,133.44816054,0,100
ELSE
  WINDOW 0,133,0,100
  Midptx=(Limits(3)+Limits(2))/2+Limits(2)
  Midpty=(Limits(0)+Limits(1))/2+Limits(1)
  CSIZE 6,.33
  MOVE Midptx-10,Limits(1)-7
  LABEL USING "9A,#";Xyident$[1]
  MOVE Limits(2)-1,Midpty-9
  LDIR (PI/2)
  LABEL USING "9A,#";Xyident$[10]
  LDIR 0
  CSIZE 4,.75
  CLIP Limits(2),Limits(3),Limits(1),Limits(0)
  VIEWPORT Limits(2),Limits(3),Limits(1),Limits(0)
END IF
!
WINDOW Xyspec(1),Xyspec(0),Xyspec(3),Xyspec(2)
!
IF NOT (Ind2) THEN
  X=0.
ELSE
  X=Tics(0)
END IF
IF NOT (Ind3) THEN
  Y=0.
ELSE
  Y=Tics(1)
END IF
Mntic(0)=Tics1(0)+1
Mntic(1)=Tics1(1)+1
!
AXES X,Y,0.0,Mntic(0),Mntic(1),4
IF Ind4 THEN
  LINE TYPE 4
  GRID X*Mntic(0),Y*Mntic(1),0.0,1,1
  LINE TYPE 1
END IF
IF Ind5 THEN FRAME
IF NOT (Ind7) THEN
  GOTO Exit
ELSE
  FOR K=0 TO 3
    IF K=0 OR K=1 THEN
      M=0
    ELSE
      M=1
    END IF
  I=INT(ABS(Xyspec(K))/Tics(M)/Mntic(M))
```

```
2760    FOR J=1 TO I
2770      IF K=0 OR K=2 THEN
2780        L=J
2790      ELSE
2800        L=-J
2810      END IF
2820      IF M THEN
2830        MOVE Tics(M-1)/2,L*Tics(M)*Mntic(M)+Tics(M)/2
2840      ELSE
2841        LORG 3
2850        MOVE L*Tics(M)*Mntic(M)-Tics(M)/2,0
2860      END IF
2870      LABEL USING "K,#";L*Tics(M)*Mntic(M)
2880    NEXT J
2890  NEXT K
2900  END IF
2910  !
2920 Exit: SUBEND
```

Appendix 3

Joystick Interface

Circuit Description

The basic requirement for this unit is to provide an interface between the General Purpose Interface Bus (GPIB) of the H.P. 9826 computer and a number of analogue voltages derived from the user controls. These controls comprise :

i) A 2 axis joystick control fitted with 2 10kΩ linear potentiometers (RS type 162-732), this giving pilot elevator and aileron commands.

ii) Four 100k linear slider potentiometers giving trim settings for the elevator, aileron and rudder in addition to a throttle control.

An A/D converter is thus required, capable of multiplexing six analogue inputs and producing an eight bit data word for each of the six channels in response to an address supplied by the computer. An eight channel multiplexed A/D converter thus forms the basis of the circuit. The device chosen was an RS type 303-545 (165) but only six of its eight possible channels are used in this application.

The circuit diagram is shown in Fig A.1 and roughly follows the RS data sheet for the A/D converter running in the bipolar (offset binary) mode.

The six analogue inputs are derived from the joystick and trim pots which, with the exception of the throttle control, VR1, are strapped between the +5V and -5V power rails. 2kΩ gain trimmer pots, T_1 - T_6, are included in each of the analogue input channels and the two unused inputs, 6 and 7, are taken to analogue ground. With this arrangement channel 0 becomes the joystick aileron demand, 1 the joystick elevator demand, 2, 3 and 4 are the elevator, aileron and rudder trims respectively and 5 is the throttle demand.

Channels 0, 1, 2, 3 and 4 thus give binary output words between 0 and 255, although the joystick channels, 0 and 1, have a relatively limited travel of + 27 binary about zero (binary 128). Channel 5 binary output is limited to positive excursions only i.e. binary 128 - 255. A summing amplifier arrangement (IC1 etc.) is used to provide for offset adjustment via. trimmer T_7 and the reference voltage, V_{REF}, for IC2 is derived from the -10V stabilised supply. The converter also requires a 1.6 MHz clock and this is derived from a simple oscillator comprising 1/4 IC3, R_1 and C_1. Analogue and digital grounds are commoned at pins 11 and 14 of IC2.

The A/D is interrogated by providing a three bit channel address on A_{0-2} which may be latched into IC2's address latch by taking the ALE input low. The current value of the selected channels eight bit output is then placed on the data bus provided that the chip select, $\overline{CS}$, is low, enabling the tri-state output buffers. Any channel may be interogated in this way irrespective of the conversion process since an 8 x 8 bit memory is used to store the A/D converter's outputs as the analogue inputs are continuously scanned. Data may thus be read from the bus at any time during the multiplexer's scan. A status pin is also provided which indicates the currently selected analogue input and may be read by the host computer although this is not used in this application.

The supplies for the circuit are provided by a mains transformer, rectifier and smoother. Voltage stabiliser IC's are then used in each supply rail, thse being at + 10V and +5V as shown in Fig. A.1. The unit is housed in a small case with the connections to the host computer brought out via. a 25 way D-type connector. The pin designations for this connector are given in Table A.1. Fig. A.1 also shows the connections to the GPIB which are required. Note that pull up and pull down terminating resistors are required on the GPIB output lines, DO_{0-15}, and buffered TTL input drivers are also required, DI_{0-15}.

The HP 9826 basically interrogates the unit by sending a 3 bit word on A_{0-2} specifying the analogue channel currently being accessed. With ALE held high and $\overline{CS}$ low IC2 then responds by placing the appropriate channel's current

output on the data bus. The BASIC program which reads the analogue inputs (see Appendix 4) is sufficiently slow to allow IC2 to place valid data on the data bus, after a change of address on A_{0-2}, before this data is read by the computer. The $\overline{CS}$ and ALE lines are thus tied low and high respectively to permanently enable IC2.

Table A.1

D-type Connector Pin Designations

D-type pin no.	Function
1	Data Bit 0 (lsb)
2	" " 1
3	" " 2
4	" " 3
5	" " 4
6	" " 5
7	" " 6
8	" " 7 (msb)
9	$\overline{CS}$
14	Gnd (digital)
15	Status
16	A_2
17	A_1
18	A_0
19	ALE

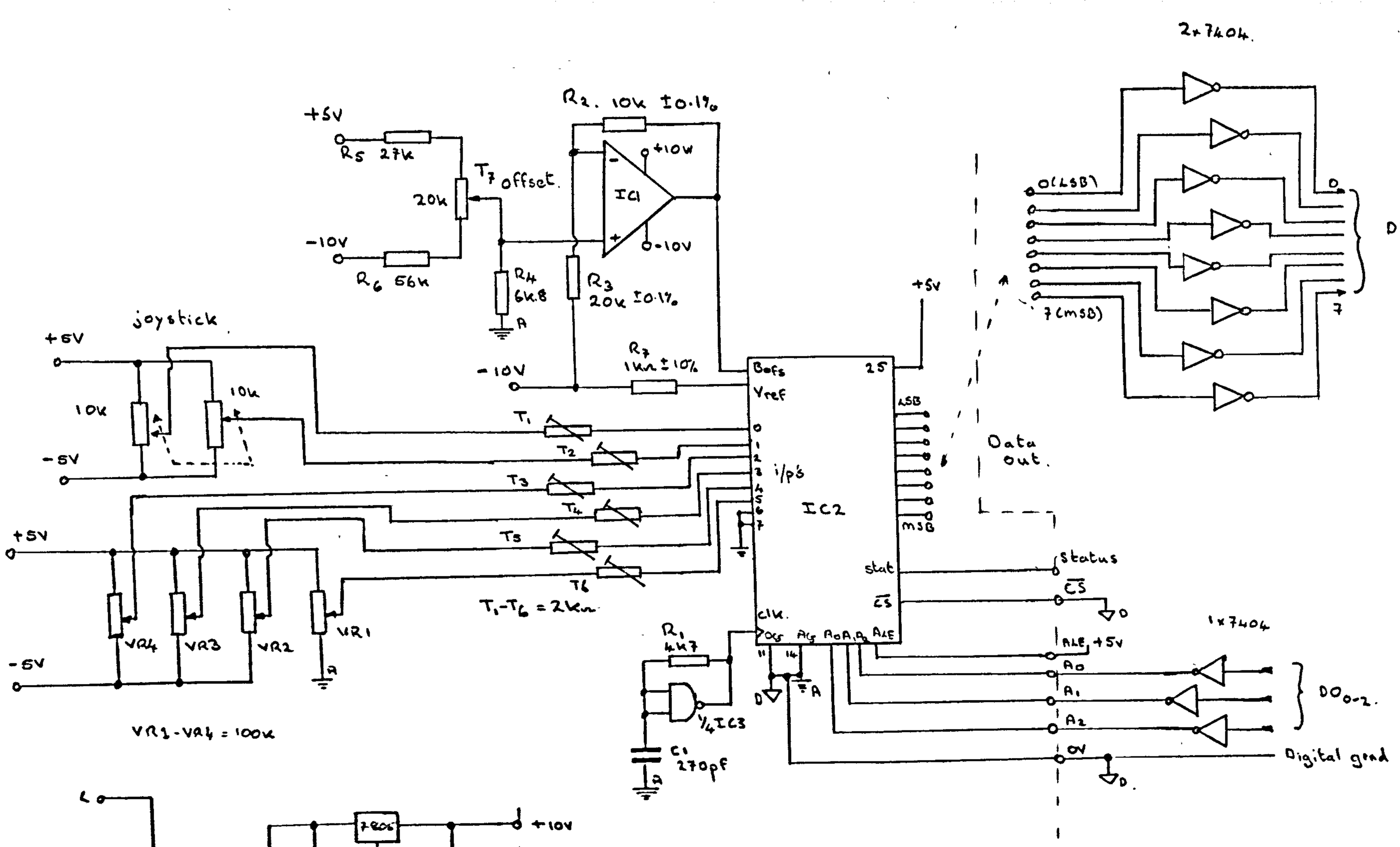


Fig. A.1 Joystick Interface Unit

IC1 = 741
IC2 = 7581
IC3 = 7400

Appendix 4 :- Contro Routine Version 5 Listing

```
1890 SUB Contro
1900 !
1910 COM /Misc1/ REAL wdirnrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
1920 COM /Misc3/ REAL T,Wgust,P,C,R,Poeg,Qdeg,Rdeg,Xn,Ye
1930 COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
1940 COM /Misc8/ REAL Ma,Taillift,Ltotal,Cmw,Clt,Nr,Na,U2,Throttle
1950 !
1960 INTEGER I
1970 !
1980 ALLOCATE INTEGER Para(6)
1990 !
2000 ASSIGN @Gpio_path TO 12;FORMAT OFF
2010 !
2020 FOR I=0 TO 5
2030     OUTPUT @Gpio_path;I
2040     ENTER @Gpio_path USING "#,W";Para(I)
2050 NEXT I
2060 !
2070 Etadeg=((Para(1)-128.)*20./27.)+((Para(2)-128.)*20./128.)
2080 Zetadeg=((Para(0)-128.)*20./27.)+((Para(3)-128.)*20./128.)
2090 Taudeg=(Para(4)-128.)*20./128.
2100 Throttle=(Para(5)-127.)/128.
2110 !
2120 Eta=Etadeg*PI/180.
2130 Zeta=Zetadeg*PI/180.
2140 Tau=Taudeg*PI/180.
2150 !
2160 ASSIGN @Gpio_path TO *
2170 !
2180 SUBEND
```

Appendix 5 :- Head-up Display Routines Version 6 Listing

```
10    OPTION BASE 0
20    PRINTER IS 1
30    MASS STORAGE IS ":INTERNAL"
40    !
50 COM /Aero1/ REAL Mass,Ix,Iy,Iz,A1,C10,Cleta,Zq,Surface
60 COM /Aero2/ REAL Lt,Cbar,Cd0,K,Yv,Yr,Ytau,Lv,Ls,Lp,Lr
70 COM /Aero3/ REAL B,A1t,A2eta,Cm0,Cma1,Cmabd,Cmeta
80 COM /Aero4/ REAL Nv,Nr0,Nr1,Ntau,Ad,Pmax,Etap,Ke,Pp
90 !
100 COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Wspeed,Wdrndeg
110 COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
120 !
150 COM /Arrays2/ REAL Trnsform(1:3,1:3),Trnspose(1:3,1:3)
160 !
170 COM /Misc1/ REAL Wdrnrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
190 COM /Misc3/ REAL T,Wgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
200 COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
210 COM /Misc5/ REAL Uwc2,Uwc,Uvwc2,Uvwc,Alpha,Beta,Ua,Va,Wa
220 COM /Misc6/ REAL X,Y,Z,L,M,N,Ubdot,Vbdot,Wbdot,Pdot,Qdot,Rdot
230 COM /Misc7/ REAL Rho,Cl,Lw,Lq,Cd,Dragforce,Ya,La,Cmt
240 COM /Misc8/ REAL Ma,Taillift,Ltotal,Cmw,Clt,Nr,Na,U2,Throttle
250 COM /Misc9/ REAL Xe,Le,Pact,Pnom,U3,Rps,Rpm,Gravity,Ugdot,Vgdot,Wgdot
260 COM /Misc10/ REAL Phidot,Thetadot,Psidot,INTEGER Iter,Stall
270   !
280   COM /Old/ REAL Temp(0:9),New(0:9),INTEGER Int(0:3)
310   !
320 Rerun:  LOADSUB ALL FROM "HEADUP"
330   CALL Headup
340   DELSUB Headup
850 LOADSUB ALL FROM "INITIALISE"
860 LOADSUB ALL FROM "MACH1"
870 CALL Indata
880 CALL Transf
890 CALL Init
900 CALL Air
910 DELSUB Indata,Init
920 LOADSUB ALL FROM "MACH2"
921 LOADSUB ALL FROM "UPDATE"
930 CALL Mainsim
940 !
950 DELSUB Air TO END
970 !
990       END
```

```
730  SUB Mainsim
740  !
750  COM /Aero1/ REAL Mass,Ix,Iy,Iz,A1,Cl0,Cleta,Zq,Surface
760 COM /Aero2/ REAL Lt,Cbar,Cd0,K,Yv,Yr,Ytau,Lv,Ls,Lp,Lr
770  COM /Aero4/ REAL Nv,Nr0,Nr1,Ntau,Ad,Pmax,Etap,Ke,Pp
780  COM /Aero3/ REAL B,A1t,A2eta,Cm0,Cm1,Cmabd,Cmeta
790  !
800  COM /Init1/ REAL Tend,Tstep,Tprint,Cg,Wspeed,Wdrndeg
810  COM /Init2/ REAL Ug,Vg,Wg,Phideg,The_tadeg,Psideg,Height
820  !
850  COM /Arrays2/ REAL Trnsform(*),Trnspose(*)
860  !
870  COM /Misc1/ REAL Wdrnrad,Phi,Theta,Psi,Etadeg,Zetadeg,Taudeg
890  COM /Misc3/ REAL T,Wgust,P,Q,R,Pdeg,Qdeg,Rdeg,Xn,Ye
900  COM /Misc4/ REAL Eta,Zeta,Tau,Uc,Ub,Vc,Vb,Wc,Wb,Vt
910  COM /Misc5/ REAL Uwc2,Uwc,Uvwc2,Uvwc,Alpha,Beta,Ua,Va,Wa
920  COM /Misc6/ REAL X,Y,Z,L,M,N,Ubdot,Vbdot,Wbdot,Pdot,Qdot,Rdot
930  COM /Misc7/ REAL Rho,Cl,Lw,Lq,Cd,Dragforce,Ya,La,Cmt
940  COM /Misc8/ REAL Ma,Taillift,Ltotal,Cmw,Clt,Nr,Na,U2,Throttle
950  COM /Misc9/ REAL Xe,Le,Pact,Pnom,U3,Rps,Rpm,Gravity,Ugdot,Vgdot,Wgdot
960  COM /Misc10/ REAL Phidot,Thetadot,Psidot,INTEGER Iter,Stall
961  COM /Old/ REAL Temp(*),New(*),INTEGER Int(*)
963  !
970  !
980  INTEGER Flag
990  REAL Printinc
991  !
995  !
997  CALL Contro
998  CALL Outdata(1)
999  Flag=1
1001 !
1002 Repeat: IF (T>=Tend) OR (Stall=1) THEN
1010           SUBEXIT
1020         ELSE
1030          IF Iter<>0 THEN
1040           CALL Air
1050           CALL Contro
1060           CALL Forces
1070           CALL Accel
1080           CALL Integrate
1090         ELSE
1100           CALL Outdata(2)
1101         END IF
1110           Iter=Iter+1
1120           T=Iter*Tstep
1130           Printinc=Printinc+Tstep
1131             IF Printinc>=Tprint THEN
1132           New(0)=Wspeed
1133           New(1)=Vt
1134           New(2)=Etadeg
1135           New(3)=Zetadeg
1136           New(4)=Taudeg
1137           New(5)=-The_tadeg
1138           New(6)=Psideg
1139           New(7)=Phideg
1140           New(8)=Height
1141           New(9)=Psideg
1142           CALL Update
1143           CALL Update1(Flag)
```

```
1144            CALL Update2
1145            CALL Reset
1146            CALL Fuse
1147            RAD
1150               CALL Outdata(3)
1160               Printinc=Printinc-Tprint
1170             END IF
1190         END IF
1200 GOTO Repeat
1210 !
1220 SUBEND
```

```
10    SUB Headup
20    !
30    ALLOCATE Label$[64],Hdng$[4]
40    Label$="W. SPEEDA. SPEEDELEVATOR AILERON RUDDER  PITCH    YAW     ROLL  "
41    Hdng$="NESW"
50    !
51    INTEGER Flag,Yord,I,M,Top,Btm
52    INTEGER Theta,K,Step,Step1
53    REAL Yorg
61    DEG
70    GINIT
80    GRAPHICS ON
90    !
100   LINE TYPE 1
110   !
120   VIEWPORT 0,130,16.67,83.33
130   WINDOW -71.6,71.6,-200,200
140   !
150   AXES 0,10,-55,-400,1,5
160   VIEWPORT 19,112,5,100
170   AXES 10,0,-100,-210,5,0
180   VIEWPORT 0,130,0,100
190   WINDOW -71.6,71.6,-300,300
200   !
210   CSIZE 4,.3
220   LORG 2
230   !
240   FOR I=-2 TO 2
250     MOVE -63,I*100
260     LABEL USING "M3D";I*100
270   NEXT I
280   LORG 5
290   FOR I=-1 TO 1
300     MOVE 50*I-1,-227
310     LABEL USING "M2D";I*50
320   NEXT I
330   CSIZE 4,.4
340   LORG 1
350   FOR I=0 TO 7
360     IF I>4 THEN
370       Yord=265
380       M=I-5
390     ELSE
400       Yord=-267
410       M=I
420     END IF
430     MOVE 1.1*((23+18*M)-65),Yord
440     LABEL USING "8A";Label$[1+8*I]
450   NEXT I
460   LORG 1
470   CSIZE 4,.4
480   FOR I=0 TO 7
490     IF I>4 THEN
500       Btm=90
510       Top=94
520       M=I-5
530     ELSE
540       Btm=1
550       Top=5
560       M=I
```

```
570       END IF
580       VIEWPORT 18*M+23,18*M+35,Btm,Top
590       FRAME
600    NEXT I
690    !
710    VIEWPORT 100,130,60,100
720    SHOW -1.5,1.5,-3.5,3.5
730    LORG 5
740    CSIZE 3,.4
750    !
760    FOR Yorg=2 TO -1.5 STEP -3.5
770       MOVE 0,Yorg+1
780       FOR Theta=0 TO 360 STEP 5
790          DRAW SIN(Theta),Yorg+COS(Theta)
800       NEXT Theta
810       IF Yorg=2 THEN
820          Step=72
830          End=324
840          Step1=36
850       ELSE
860          Step1=90
870          Step=90
880          End=270
890       END IF
900       FOR Theta=0 TO 360 STEP Step/2
910          MOVE .85*SIN(Theta),Yorg+.85*COS(Theta)
920          DRAW SIN(Theta),Yorg+COS(Theta)
930       NEXT Theta
940       FOR Theta=0 TO End STEP Step1
950          MOVE 1.2*SIN(Theta),Yorg+1.2*COS(Theta)
960          IF Yorg=2 THEN
970             LABEL K
980             K=K+1
990          ELSE
1000            LABEL USING "A";Hdng$[1+Theta/90]
1001         END IF
1010      NEXT Theta
1020   NEXT Yorg
1030   !
1040   CSIZE 4,.35
1050   MOVE -.025,.3
1060   LABEL "HEIGHT M*10"
1070   MOVE 0,-3.2
1080   LABEL "HEADING"
1081   !
1082   !End of setup procedure
1083   !
1084   SUBEND
```

```
1420  SUB Update
1430  COM /Old/ REAL Temp(*),New(*),INTEGER Int(*)
1431  ALLOCATE INTEGER Int1(0:3)
1440  INTEGER I,J
1441  REAL Y1,Y2
1450  DEG
1460  LINE TYPE 1
1470  LORG 1
1480  CSIZE 4,.4
1490  VIEWPORT 15,115,18,83
1500  WINDOW -55.,55.,-192,204
1510  !
1520  Int1(0)=35*COS(New(7))
1530  Int1(1)=35*SIN(New(7))*RATIO*4.1899
1540  Int1(2)=-5*SIN(New(7))
1550  Int1(3)=5*COS(New(7))*RATIO*4.1899
1551  !
1560  FOR J=-1 TO 1 STEP 2
1570     PEN J
1580     IF J=-1 THEN
1590        Y1=Temp(5)
1591        Y2=Temp(6)
1592     ELSE
1593        Y1=New(5)
1594        Y2=New(6)
1595        Int(0)=Int1(0)
1596        Int(1)=Int1(1)
1597        Int(2)=Int1(2)
1598        Int(3)=Int1(3)
1599     END IF
1600     FOR I=1 TO -1 STEP -2
1601        MOVE 0,Y1
1602        DRAW I*Int(0),Int(1)*I+Y1
1603     NEXT I
1604     MOVE 0,Y1
1605     DRAW Int(2),Y1+Int(3)
1606     LINE TYPE 9
1607     PEN J
1608     FOR I=1 TO -1 STEP -2
1609        MOVE Y2,-182
1610        IDRAW 0,10*I
1611        MOVE -50,Y1
1612        IDRAW 2*I,0
1613     NEXT I
1614     LINE TYPE 1
1616  NEXT J
1617  SUBEND
1625  !
1665  SUB Reset
1666  !
1668  COM /Old/ REAL Temp(*),New(*),INTEGER Int(*)
1678  INTEGER I
1688  FOR I=0 TO 7
1698     Temp(I)=New(I)
1708  NEXT I
1718  SUBEND
1728  !
1729  SUB Update1(INTEGER Flag)
1730  COM /Old/ REAL Temp(*),New(*),INTEGER Int(*)
1740  !
```

```
1750  INTEGER I,Btm,M,Subpen
1760  !
1770  DEG
1780  LINE TYPE 1
1790  LORG 1
1800  CSIZE 4,.4
1810  VIEWPORT 0,130,0,100
1820  WINDOW 0,130,0,100
1830  !
1840  FOR I=0 TO 7
1850     IF ABS(Temp(I)-New(I))>=.1 OR (Flag=1) THEN
1860        IF I>4 THEN
1870           Btm=90
1880           M=I-5
1890        ELSE
1900           Btm=1
1910           M=I
1920        END IF
1930        MOVE 18*M+23,Btm
1940        FOR Subpen=-1 TO 1 STEP 2
1950           PEN Subpen
1960           IF Subpen=-1 THEN
1970              LABEL USING "M4D.D":Temp(I)
1980           ELSE
1990              MOVE 18*M+23,Btm
2000              LABEL USING "M4D.D";New(I)
2010           END IF
2020        NEXT Subpen
2030     END IF
2040  NEXT I
2050  !
2060  VIEWPORT 15,115,18,83,
2070  WINDOW -55,55,-192,204
2071  Flag=0
2080  !
2090  SUBEND
2100  !
2110  SUB Update2
2120  !
2130  COM /Old/ REAL Temp(*),New(*),INTEGER Int(*)
2131  INTEGER L,J
2132  REAL Lngth,Angle,Angle1,Angle2
2140  !
2150  DEG
2160  LINE TYPE 1
2170  VIEWPORT 100,130,60,100
2180  SHOW -1.5,1.5,-3.5,3.5
2190  !
2200  Angle=3.6*New(8)
2210  FOR J=-1 TO 1 STEP 2
2220     PEN J
2230     IF J=-1 THEN
2240        Angle1=Temp(8)
2250        Angle2=Temp(9)
2260     ELSE
2270        Angle1=Angle
2280        Angle2=New(9)
2290     END IF
2300     Lngth=.7
2310     FOR L=0 TO 2
```

```
2320        MOVE 0,.2
2330        DRAW Lngth*SIN(Angle1/10^L),.2+Lngth*COS(Angle1/10^L)
2340        Lngth=Lngth-.2
2350     NEXT L
2360     MOVE 0,-1.5
2370     DRAW .7*SIN(Angle2),-1.5+.7*COS(Angle2)
2380  NEXT J
2390  Temp(9)=New(9)
2400  Temp(8)=Angle
2410  !
2420  VIEWPORT 15,115,18,83
2430  WINDOW -55,55,-192,204
2440  !
2450  SUBEND
2460  !
2470  SUB Fuse
2480  INTEGER Theta,Phi,I
2481  REAL Xcord,Ycord
2482  !
2499  LINE TYPE 1
2500  PEN 1
2508  !
2518  VIEWPORT 45,85,40,60
2528  SHOW -2,2,-1,1
2538  !
2548  DEG
2558     MOVE .052,0
2568     LORG 5
2578     CSIZE 10,1
2588     LABEL "0"
2618  !
2628     MOVE .32,0
2638     DRAW 1.5,0
2648     MOVE -.32,0
2658     DRAW -1.5,0
2668     MOVE 0,.32
2678     DRAW 0,.82
2688     MOVE -.22,.67
2698     DRAW .22,.67
2699     MOVE .232,-.232
2700     DRAW .424,-.424
2704     MOVE -.232,-.232
2705     DRAW -.424,-.424
2710  !
2711  VIEWPORT 15,115,18,83
2712  WINDOW -55,55,-192,204
2716  !
2717  SUBEND
```

Appendix 6 :- FORTRAN Program Version 7 Listing

```
C     HACHAN AIRCRAFT SIMULATION
C
C
      IMPLICIT REAL (I-N)
      EXTERNAL FCN,OUTPUT
C
C
      INTEGER IFAIL,IRELAB,N,RESCNT,I,J
      LOGICAL STALL
C
C
C
      DIMENSION X(16),XDOT(16),W(16,7)
C
C
      COMMON /AIRC/ WSPEED,WDIRN,WGUST
      COMMON /PITCHC/ CM0,CML,CMETA,CMQDD
      COMMON /YAWC/ NV,NTAU
      COMMON /LIFTC/ CL0,A1,Z0
      COMMON /DRAGC/ CD0,K
      COMMON /THRC/ PMAX,ETAP,KE,AD,PP
      COMMON /SIDEC/YV,YR,YTAU
      COMMON /ROLLC/ LP,LR,LV,LS
      COMMON /FORCC/ MASS,GRAV,CG
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,L1,B,CBAR,PI
      COMMON /FLAGS/ STALL
C
C
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /PITCHV/ TAILLF,MA
      COMMON /YAWV/ NR,NA
      COMMON /LIFTV/ CL,LW,LQ
      COMMON /DRAGV/ CD,DRAG
      COMMON /THRV/ LE,PACT,U2
      COMMON /SIDEV/ YA
      COMMON /ROLLV/ LA
      COMMON /FORCV/ XFORCE,YFORCE,ZFORCE,LMOMT,MMOMT,NMOMT
      COMMON /CONTRV/ TAU,ETA,ZETA,TH
      COMMON /TRANS/ TFORM(3,3),TPOSE(3,3)
      COMMON /CONTR/ KF1(4,14),KF2(4,11),REF(14)
      COMMON /PILMAT/ PILOTH(1)
      COMMON /STATIM/ TSTALL
      COMMON /RESUL/ RESULT(16,600),RESCNT
      COMMON /GRAVV/ UG,WG,VG
      COMMON /DEMS/ TIM(20),ETADEM(20),ZETDEM(20),TAUDEM(20),THDEM(20)
     *,NDEMPT
C
C
C     MAIN PROGRAM START
C
      CALL INITAL (X,XDOT)
C
      WRITE (5,110)
110   FORMAT (1X,'ENTER LENGTH OF SIMULATION')
      READ (5,*) TEND
C
40    WRITE (5,120)
120   FORMAT (1X,'ENTER TIME BET-EEN OUTPUT SAMPLES')
      READ (5,*) TSTEP
C
C     CHECK FOR TOO MANY SAMPLES
C
      IF (TEND/TSTEP.GT.599) GOTO 20
C
C
      WRITE (5,130)
130   FORMAT(1X,'Enter name of result file')
      READ (5,135) FNAM
135   FORMAT (A5)
C
C     SET UP PARAMETERS FOR D02BBF
      IFAIL=0
      IRELAB=0
      TOL=0.000001
      N=16
      T=0.
C
50    CALL D02BBF (T,TEND,N,X,TOL,IRELAB,FCN,OUTPUT,W,IFAIL)
      IF (STALL) WRITE (5,200) TSTALL
      IF (IFAIL.EQ.0) GOTO 10
      WRITE (5,100) IFAIL
C
10    CONTINUE
      GOTO 30
C
20    WRITE (5,150)
150   FORMAT (1X,'TOO MANY INTERMEDIATE STEPS')
      GOTO 40
C
30    CONTINUE
      OPEN (UNIT=1,FILE=FNAM)
      WRITE (1,*) TEND,TSTEP,9,1,RESCNT
      WRITE (1,*) ((RESULT(I,J),I=1,16),J=1,RESCNT)
      CLOSE (UNIT=1)
C
100   FORMAT (1X,'D02BBF FAILED WITH IFAIL = ',I2)
200   FORMAT (1X,'AIRCRAFT STALLED AT TIME = ',F10.5)
      END
C
C
C
C
C     SUBPROGRAM FCN
C     USED BY D02BBF TO CALCULATE STATES DIFFS FROM
C     STATES IN X().
C
      SUBROUTINE FCN (T,X,XDOT)
C
      IMPLICIT REAL (I-N)
C
      LOGICAL STALL
C
      DIMENSION X(16),XDOT(16)
C
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,L1,B,CBAR,PI
      COMMON /FLAGS/ STALL
      COMMON /STATIM/ TSTALL
C
      CALL CONTRL (X,T)
      CALL TRANSF (X(7),X(8),X(9))
```

```
      CALL AIR (X)
      CALL LIFT (X)
      CALL DRAG (X)
      CALL SIDE (X)
      CALL ROLL (X)
      CALL PITCH (X)
      CALL YAW (X)
      CALL THRUST (X)
      CALL FORSUM (X)
C
      CALL DIFFS (X,XDOT)
C
      IF ((STALL.EQ..FALSE.).OR.(TSTALL.GT.0.0)) GOTO 10
      TSTALL=T
C
10    RETURN
      END
C
C
C
C
C     SUBPROGRAM AIR
C     CALCULATES AIRSPEEDS, X,Y,Z AXIS SPEEDS, X,Y,Z BODY AXIS SPEEDS
C
C
      SUBROUTINE AIR (X)
C
      IMPLICIT REAL (I-H)
C
      DIMENSION X(16)
C
C
      COMMON /AIRC/ WSPEED,WDIRH,WGUST
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /TRANS/ TFORM(3,3),TPOSE(3,3)
      COMMON /GRAVV/ UG,VG,WG
C
C
      UW=WSPEED*COS(WDIRH)
      VW=WSPEED*SIN(WDIRH)
      WW=WGUST
C
C     CALCULATE AIRSPEEDS
C
C
      UA=UG+UW
      VA=VG+VW
      WA=WG+WW
C
C     CALCULATE BODY AXIS SPEEDS
C
      CALL PULL (TPOSE,UA,VA,WA,UC,VC,WC)
C
      UVWA2=UA**2+VA**2+WA**2
      VT=SQRT(UVWA2)
C
      UWC2=UC**2+WC**2
      UWC=SQRT(UWC2)
C
      UVWC2=UC**2+VC**2+WC**2
      UVWC=SQRT(UVWC2)
C
C     CALCULATE INCIDENCE AND SIDESLIP ANGLES
C
      ALPHA=ASIN(WC/UWC)
30    BETA=ASIN(VC/VT)
C
      RETURN
      END
C
C
C
C
C     SUBPROGRAM LIFT
C     CLACULATES LIFT FROM WINGS AND PITCH RATE
C
      SUBROUTINE LIFT (X)
C
      IMPLICIT REAL (I-H)
C
      LOGICAL STALL
C
      DIMENSION X(16)
C
      COMMON /LIFTC/ CL0,A1,Z0
      COMMON /LIFTV/ CL,LW,LQ
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,D,CBAR,PI
      COMMON /FLAGS/ STALL
C
C
      CL=CL0+A1*ALPHA
      IF (CL.LE.1.2) GOTO 10
      STALL=.TRUE.
      CL=0
10    LW=0.5*RHO*UWC2*SURFAC*CL
      LQ=-RHO*SURFAC*UWC*LT*Z0*X(2)
C
      RETURN
      END
C
C
C
C
C     SUBPROGRAM DRAG
C     CALCULATES DRAG FORCE
C
      SUBROUTINE DRAG (X)
C
      IMPLICIT REAL (I-H)
C
      DIMENSION X(16)
C
      COMMON /DRAGC/ CD0,K
      COMMON /DRAGV/ CD,DRAG
      COMMON /LIFTV/ CL,LW,LQ
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /MISC/ TSTEP,TEND, IX,IY,IZ,RHO,SURFAC,LT,D,CBAR,PI
C
C
```

```
      CD=CD0+K*CL**2
      DRAG=0.5*RHO*UWC2*SURFAC*CD
C
      RETURN
      END
C
C
C
C
C     SUBPROGRAM SIDE
C     CALCULATES SIDE FORCES
C
      SUBROUTINE SIDE (X)
C
      IMPLICIT REAL (I-N)
C
      DIMENSION X(16)
C
      COMMON /SIDEC/ YV,YR,YTAU
      COMMON /SIDEV/ YA
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
C
C
      YA=RHO*SURFAC*UVWC*(YV*VC+0.5*B*YR*X(3)+UVWC*YTAU*X(16))
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE ROLL
C
      SUBROUTINE ROLL (X)
C
      IMPLICIT REAL (I-N)
C
      DIMENSION X(16)
C
      COMMON /ROLLC/ LP,LR,LV,LS
      COMMON /ROLLV/ LA
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
      COMMON /LIFTV/ CL,LW,LQ
C
      LA=0.5*RHO*UVWC*B*SURFAC*(0.5*B*(LP*X(1)+LR*CL*X(3))+LV*VC+UVWC*LS
     1*X(15))
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE PITCH
C
      SUBROUTINE PITCH (X)
C
      IMPLICIT REAL (I-N)
C
      LOGICAL STALL
C
      DIMENSION X(16)
C
      COMMON /PITCHC/ CM0,CML,CMETA,CMWDD
      COMMON /PITCHV/ TAILLF,MA
      COMMON /LIFTV/ CL,LW,LQ
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /FLAGS/ STALL
C
C
      CMW=CM0+CML*CL
      IF (STALL) CMW=-0.3
      CMT=CMETA*X(11)
      TAILLF=(CMT-CMWDD)*0.5*RHO*UWC2*SURFAC*CBAR/LT+LQ
      MA=0.5*RHO*UWC2*SURFAC*CBAR*(CMW+CMT)
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE YAW
C
      SUBROUTINE YAW (X)
C
      IMPLICIT REAL (I-N)
C
      DIMENSION X(16)
C
      COMMON /YAWC/ NV,NTAU
      COMMON /YAWV/ NR,NA
      COMMON /LIFTV/ CL,LW,LQ
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
C
      NR=-0.046-0.015*CL**2
      NA=0.5*RHO*SURFAC*UVWC*B*(NV*VC+0.5*B*NR*X(3)+UVWC*NTAU*X(16))
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE THRUST
C
      SUBROUTINE THRUST (X)
C
      IMPLICIT REAL (I-N)
C
      DIMENSION X(16)
C
      COMMON /THRC/ PMAX,ETAP,KE,AD,PP
      COMMON /THRV/ LE,PACT,U2
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
```

```
      COMMON /CONTRV/ TAU,ETA,ZETA,TH
C
      PNOM=PMAX*TH
      PACT=PNOM*ETAP
      U3=SQRT((2*X(13)+RHO*AD*UC**2)/(RHO*AD))
      U2=X(13)/(RHO*AD*(U3-UC))
      RPS=U2/PP
      RPM=RPS*60
      LE=PNOM/(2*PI*RPS)
      LE=0.0
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE FORSUM
C     CALCULATES TOTAL FORCES ACTING ON PLANE
C
      SUBROUTINE FORSUM (X)
C
      IMPLICIT REAL (I-N)
C
      DIMENSION X(16)
C
      COMMON /FORCC/ MASS,GRAV,CG
      COMMON /FORCV/ XFORCE,YFORCE,ZFORCE,LMOMT,MMOMT,NMOMT
      COMMON /LIFTV/ CL,LW,LQ
      COMMON /PITCHV/ TAILLF,MA
      COMMON /SIDEV/ YA
      COMMON /YAWV/ NR,NA
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /DRAGV/ CD,DRAG
      COMMON /THRV/ LE,PACT,U2
      COMMON /ROLLV/ LA
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
C
      LTOTAL=LW+TAILLF
C
      XFORCE=X(13)-DRAG*COS(ALPHA)+LTOTAL*SIN(ALPHA)-MASS*GRAV*SIN(X(8))
      YFORCE=YA+MASS*GRAV*COS(X(8))*SIN(X(7))
      ZFORCE=-LTOTAL*COS(ALPHA)-DRAG*SIN(ALPHA)+MASS*GRAV*COS(X(8))
     1*COS(X(7))
C
      LMOMT=LE+LA
      MMOMT=MA+LW*(CG-0.25)*CBAR-(LT+(0.25-CG)*CBAR)*LQ
      NMOMT=NA
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE DIFFS
C     CALCULATES XDOTS FROM STATES X
C
      SUBROUTINE DIFFS (X,XDOT)
C
      IMPLICIT REAL (I-N)
C
      DIMENSION X(16),XDOT(16)
C
      COMMON /FORCV/ XFORCE,YFORCE,ZFORCE,LMOMT,MMOMT,NMOMT
      COMMON /FORCC/ MASS,GRAV,CG
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
      COMMON /TRANS/ TFORM(3,3),TPOSE(3,3)
      COMMON /THRV/ LE,PACT,U2
      COMMON /THRC/ PMAX,ETAP,KE,AD,PP
      COMMON /CONTRV/ TAU,ETA,ZETA,TH
      COMMON /GRAVV/ UG,WG,VG
C
      XDOT(1)=(LMOMT+(IY-IZ)*X(3)*X(2))/IX
      XDOT(2)=(MMOMT+(IZ-IX)*X(1)*X(3))/IY
      XDOT(3)=(NMOMT+(IX-IY)*X(1)*X(2))/IZ
C
      XDOT(4)=XFORCE/MASS-X(6)*X(2)+X(5)*X(3)
      XDOT(5)=YFORCE/MASS-X(4)*X(3)+X(6)*X(1)
      XDOT(6)=ZFORCE/MASS-X(5)*X(1)+X(4)*X(2)
C
      XDOT(7)=X(1)+SIN(X(7))*TAN(X(8))*X(2)+COS(X(7))*TAN(X(8))*X(3)
      XDOT(8)=COS(X(7))*X(2)-SIN(X(7))*X(3)
      XDOT(9)=SIN(X(7))*X(2)/COS(X(8))+COS(X(7))*X(3)/COS(X(8))
C
      CALL MULL (TFORM,X(4),X(5),X(6),UG,VG,WG)
      XDOT(10)=UG
      XDOT(11)=VG
      XDOT(12)=-WG
C
      XDOT(13)=(PACT-X(13)*U2)/KE
C
C     CONTROL DYNAMICS
C
      XDOT(14)=20.0*(ETA-X(14))
      XDOT(15)=20.0*(ZETA-X(15))
      XDOT(16)=10.0*(TAU-X(16))
C
C     ADD RATE LIMITS
C
      IF (ABS(XDOT(14)).GT.50.) XDOT(14)=SIGN(50.,XDOT(14))
      IF (ABS(XDOT(15)).GT.50.) XDOT(15)=SIGN(50.,XDOT(15))
      IF (ABS(XDOT(16)).GT.50.) XDOT(16)=SIGN(50.,XDOT(16))
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE TRANSF
C     CALCULATES TRANSFORM AND TRANSPOSE MATRIX
C
      SUBROUTINE TRANSF (PHI,THETA,PSI)
C
      INTEGER I,J
C
      COMMON /TRANS/ TFORM(3,3),TPOSE(3,3)
C
      TFORM(1,1)=COS(THETA)*COS(PSI)
```

```
      TFORM(1,2)=SIN(PHI)*SIN(THETA)*COS(PSI)-COS(PHI)*SIN(PSI)
      TFORM(1,3)=COS(PHI)*SIN(THETA)*COS(PSI)+SIN(PHI)*SIN(PSI)
      TFORM(2,1)=COS(THETA)*SIN(PSI)
      TFORM(2,2)=SIN(PHI)*SIN(THETA)*SIN(PSI)+COS(PHI)*COS(PSI)
      TFORM(2,3)=COS(PHI)*SIN(THETA)*SIN(PSI)-SIN(PHI)*COS(PSI)
      TFORM(3,1)=-SIN(THETA)
      TFORM(3,2)=SIN(PHI)*COS(THETA)
      TFORM(3,3)=COS(PHI)*COS(THETA)
C
      DO 10 I=1,3
      DO 10 J=1,3
      TPOSE(I,J)=TFORM(J,I)
10    CONTINUE
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE MULL
C     CALCULATE MATRIX TRANS MULTIPLICATION
C
      SUBROUTINE MULL (TRAN,X1,X2,X3,Y1,Y2,Y3)
C
      DIMENSION TRAN(3,3)
C
      Y1=TRAN(1,1)*X1+TRAN(1,2)*X2+TRAN(1,3)*X3
      Y2=TRAN(2,1)*X1+TRAN(2,2)*X2+TRAN(2,3)*X3
      Y3=TRAN(3,1)*X1+TRAN(3,2)*X2+TRAN(3,3)*X3
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE INITIAL
C
      SUBROUTINE INITAL (X,XDOT)
C
      IMPLICIT REAL (I-H)
C
      INTEGER RESCNT,I
      LOGICAL STALL
C
      DIMENSION X(16),XDOT(16)
C
      COMMON /AIRC/ WSPEED,WDIRN,WGUST
      COMMON /PITCHC/ CMO,CML,CMETA,CMWBD
      COMMON /YAWC/ NV,NTAU
      COMMON /LIFTC/ CLO,A1,ZO
      COMMON /DRAGC/ CDO,K
      COMMON /THRC/ PMAX,ETAP,KE,AD,PP
      COMMON /SIDEC/YV,YR,YTAU
      COMMON /ROLLC/ LP,LR,LV,LS
      COMMON /FORCC/ MASS,GRAV,CG
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
      COMMON /FLAGS/ STALL
C
C
      COMMON /CONTRV/ TAU,ETA,ZETA,TH
      COMMON /CONTH/ KF1(4,14),KF2(4,14),KEF(14)
      COMMON /PILMAT/ PILOTU(4)
      COMMON /RESUL/ RESULT(16,600),RESCNT
      COMMON /TRANS/ TFORM(3,3),TPOSE(3,3)
      COMMON /GRAVV/ UG,WG,VG
      COMMON /DEMS/ TIN(20),ETADEM(20),ZETDEM(20),TALDEM(20),THDEM(20)
     *,NBRKPT
C
C     READ IN ALL CONSTANTS
C
      OPEN (UNIT=1,FILE='CONT')
C
      READ (1,*) UG
      READ (1,*) VG
      READ (1,*) WG
      READ(1,*) WSPEED
      READ(1,*) WDIRN
      READ(1,*) WGUST
      READ(1,*) CMO
      READ(1,*) CML
      READ(1,*) CMETA
      READ(1,*) CMWBD
      READ(1,*) NV
      READ(1,*) NTAU
      READ(1,*) CLO
      READ(1,*) A1
      READ(1,*) ZO
      READ(1,*) CDO
      READ(1,*) K
      READ(1,*) PMAX
      READ(1,*) ETAP
      READ(1,*) KE
      READ(1,*) AD
      READ(1,*) PP
      READ(1,*) YV
      READ(1,*) YR
      READ(1,*) YTAU
      READ(1,*) LP
      READ(1,*) LR
      READ(1,*) LV
      READ(1,*) LS
      READ(1,*) MASS
      READ(1,*) GRAV
      READ(1,*) CG
      READ(1,*) IX
      READ(1,*) IY
      READ(1,*) IZ
      READ(1,*) RHO
      READ(1,*) SURFAC
      READ(1,*) LT
      READ(1,*) B
      READ(1,*) CBAR
      READ(1,*) PI
C
      CLOSE(UNIT=1)
C
C     SET UP FEEDBACK MATRICES
C
```

```
      CALL SETFB
C
CC    SET UP INITIAL VALUES OF STATES
C
      OPEN (UNIT=1,FILE='INSTAT')
      READ (1,*) (X(I),I=1,16)
      CLOSE (UNIT=1)
C
      OPEN (UNIT=1,FILE='DEMAND')
      READ (1,*) NBRKPT
      DO 100 I=1,NBRKPT
100   READ (1,*) TIM(I),THDEM(I),ETADEM(I),ZETDEM(I),TAUDEM(I)
      CLOSE (UNIT=1)
C
CC    SET FLAGS
C
      STALL=.FALSE.
C
CC    SET RESULT OUTPUT COUNTER
C
      RESCNT=0
C
      ETA=0.0
      ZETA=0.0
      TAU=0.0
      TH=0.0
      CALL TRANSF(X(7),X(8),X(9))
      CALL FULL(TPOSE,UG,VG,WG,X(1),X(5),X(6))
C
      PILOTU(1)=ETADEM(1)
      PILOTU(2)=THDEM(1)
      PILOTU(4)=ZETDEM(1)
      PILOTU(3)=TAUDEM(1)
      RETURN
      END
C
C
C
C
CC    SUBROUTINE CONTROL
C
      SUBROUTINE CONTRL (X,T)
C
      IMPLICIT REAL (I-N)
C
      INTEGER I,TMATCH
C
      DIMENSION X(16),XCON(14),U(4),F1(4),F2(4)
C
      COMMON /CONTRV/ TAU,ETA,ZETA,TH
      COMMON /CONTM/ KF1(4,14),KF2(4,14),REF(14)
      COMMON /PILMAT/ PILOTU(4)
      COMMON /DEMS/ TIM(20),ETADEM(20),ZETDEM(20),TAUDEM(20),THDEM(20)
     *,NBRKPT
C
CC    SET UP REQUIRED STATES
C
      XCON(1)=X(4)
      XCON(2)=X(6)
      XCON(3)=X(2)
      XCON(4)=X(8)
      XCON(5)=X(12)
      XCON(6)=X(13)
      XCON(7)=X(14)
      XCON(8)=X(5)
      XCON(9)=X(1)
      XCON(10)=X(3)
      XCON(11)=X(7)
      XCON(12)=X(9)
      XCON(13)=X(15)
      XCON(14)=X(16)
C
      TMATCH=0
      I=1
C
40    IF ((IMATCH.EQ.1).OR.(I.GT.(NBRKPT-1))) GO TO 50
      IF ((T.GE.TIM(I)).AND.(T.LT.TIM(I+1))) GO TO 60
      GO TO 70
60    PILOTU(1)=ETADEM(I)
      PILOTU(2)=THDEM(I)
      PILOTU(4)=ZETDEM(I)
      PILOTU(3)=TAUDEM(I)
      TMATCH=1
      GO TO 80
70    I=I+1
80    GO TO 40
C
C
50    CALL MMULL(XCON,KF1,F1)
      CALL MMULL(REF,KF2,F2)
      DO 10 I=1,4
      U(I)=PILOTU(I)+F1(I)-F2(I)
10    CONTINUE
C
      ETA=U(1)
      TH=U(2)
      ZETA=U(4)
      TAU=U(3)
C
CC    SET HARD LIMITS ON CONTROL SURFACES AND THROTTLE
C
      IF (TH.GT.1.0) TH=1.0
      IF (TH.LT.0.0) TH=0.0
C
CC    ADD HARD LIMITS TO ACTUAL CONTROL SURFACE OUTPUTS
C
      IF (ABS(X(14)).GT.0.4363) X(14)=SIGN(0.4363,X(14))
      IF (ABS(X(15)).GT.0.4363) X(15)=SIGN(0.4363,X(15))
      IF (ABS(X(16)).GT.0.5236) X(16)=SIGN(0.5236,X(16))
C
20    RETURN
      END
C
C
C
C
CC    SUBROUTINE MMULL
CC    MULTIPLIES 4*14 AND 1*14 TO PRODUCE 1*4 MATRIX
C
      SUBROUTINE MMULL (X,Y,U)
```

```
C
      INTEGER I,J
C
      DIMENSION X(14),Y(1,14),U(1)
C
      DO 10 I=1,1
      U(I)=0
10    CONTINUE
C
      DO 20 I=1,4
      DO 20 J=1,14
      U(I)=U(I)+Y(I,J)*X(J)
20    CONTINUE
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE SET FEEDBACK
C     SET UP STATE FEEDBACK MATRICES
C
      SUBROUTINE SETFB
C
      IMPLICIT REAL (I-N)
C
      INTEGER I
C
      COMMON /CONTM/ KF1(4,14),KF2(4,14),REF(14)
      COMMON /PILMAT/ PILOTU(1)
C
      OPEN (UNIT=1,FILE='FEEDBK')
C
      READ(1,*) (KF1(1,I),I=1,14)
      READ(1,*) (KF1(2,I),I=1,14)
      READ(1,*) (KF1(3,I),I=1,14)
      READ(1,*) (KF1(4,I),I=1,14)
C
      READ(1,*) (KF2(1,I),I=1,14)
      READ(1,*) (KF2(2,I),I=1,14)
      READ(1,*) (KF2(3,I),I=1,14)
      READ(1,*) (KF2(4,I),I=1,14)
C
      READ(1,*) (REF(I),I=1,11)
C
      CLOSE (UNIT=1)
C
      DO 10 I=1,4
      PILOTU(I)=0.0
10    CONTINUE
C
      RETURN
      END
C
C
C
C
C     SUBROUTINE OUTPUT
C     PRESENTATION OF RESULTANT STATES AFTER SUSSESIVE INTEVALS OF TSTEP
C
      SUBROUTINE OUTPUT (TIME,X)
C
      IMPLICIT REAL (I-N)
C
      INTEGER I,RESCNT
C
      DIMENSION X(16)
C
      COMMON /MISC/ TSTEP,TEND,IX,IY,IZ,RHO,SURFAC,LT,B,CBAR,PI
      COMMON /RESUL/ RESULT(16,600),RESCNT
      COMMON /AIRV/ UC,VC,WC,UWC2,UWC,UVWC2,UVWC,ALPHA,BETA,VT
      COMMON /CONTRV/ TAU,ETA,ZETA,TH
C
      RESCNT=RESCNT+1
      TIME=TIME+TSTEP
C
      DO 10 I=1,16
10    RESULT(I,RESCNT)=X(I)
C
      RETURN
      END
C
```

Appendix 7

This appendix provides a listing of the LQP optimal control CAD package which was developed in connection with the current work. The package is intended to allow interactive design of control systems employing LQP techniques and has been used extensively to provide the controller designs presented in Chapter 6. The development of the optimal control gain matrix relies on a well known eigenvector solution of the infinite time algebraic Ricatti equation (31). The package is thus only applicable to systems with 'infinite time' end points when the steady state Ricatti formulation is required. The system is also required to be controllable and minimum phase.

The package is currently under revision and hence it is not intended to examine its structure in great detail. A purely functional description is thus provided in the following discussion.

The package has been structured around a previously developed suite of routines for the evaluation and display of multivariable root loci (160) and system time responses. The majority of root locus routines have been removed however due to core limitations on the DEC-10 machine on which the program was developed. The program is structured 'top down' and is essentially menu driven with sub menus presented to the user at each stage of decision making. The initial main menu provides the user with the following options :

INP - inputs system A, B and C matrices either from the keyboard or disc file (from previous store).

OUT - exits to system

CHE - displays current system data

MOD - allows existing data to be modified

COM - allows a forward path compensator, defined in state form, to be introduced

STO - stores current data on disc file (retrievable via INP)

OPT - enters optimal control design routines

Typing any of the above allows the appropriate option to be executed. Note that in all cases the system data is required in state space form and the system must be strictly proper. The compensator may, however, be proper. Each option prompts the user to input either data or a response before returning to the main menu. The OPT option is somewhat different, however, since it allows the user to design the optimal control gains and investigate the time responses of the optimally controlled system. On entry to OPT the user is required to input the Q and R matrices of the quadratic performance criteria which may be chosen as C^TC and I respectively. A sub-menu is then presented which allows the following options :

1 - modify either of Q or R
2 - display any of Q, R, system matrices or optimal control gains, if already evaluated
3 - evaluate and display the two optimal gain matrices
4 - evaluate and display optimally controlled system time responses
5 - store or retrieve data to disc file
6 - return to main menu

Note that both the state optimal control gain and the state reference matrix (see Chapter 6, section 6.6) are evaluated in option 3. Option 4 again provides a sub menu to allow the user to specify the time steps, type of input, device for display, etc. This option also allows the user to introduce a random disturbance into the system, defined by a coupling matrix. The type of disturbance introduced is somewhat specific to the aircraft problem since it is based on the well known Dryden (149) model of atmospheric turbulence. For deterministic systems the user would not use this facility.

A normal itterative procedure may be adopted for the design, the user first specifying the Q and R and, after evaluating the optimal control gains, checking that the time

response is satisfactory. If it is not Q and/or R must be modified and the time response checked again. This itterative choice of Q and R is undesirable and it would be more usefull to employ, for example, and eigenvalue/eigenvector assignment technique, such as reviewed in Chapter 6, to assist in the choice of Q and R and to provide prescribed time responses. It is intended to provide this facility in addition to allowing multivariable root loci to be plotted for the optimally controlled system and its asymptotic properties investigated.

The following pages provide a listing of the package written in FORTRAN V7. Note that extensive use is made of both NAG and GINO routines for matrix handling and graphics control. Some elementary matrix handling, e.g. addition, multiplication, transposition, etc., is provided by a smaller suite of routines. A routine, PHIDEL, is used to evaluate the system time response and this relies on the discrete time solution of the state equations.

```
C This package is intended to allow LQP controllers to be designed
C and their performance assessed.
c The package is menu driven and allows the user to modify controller
C parameters interactively.  Output is to a T4010 compatable terminal.
C
C     SET UP COMMON STORAGE BLOCKS
C
C23456789
      COMMON /SYST/A(10,10),B(10,10),C(10,10),N,M
      COMMON /COMP/ACOMP(10,10),BCOMP(10,10),CCOMP(10,10),
     1DCOMP(10,10),NC,MC
      COMMON /SYSTR/ASTR(10,10),BSTR(10,10),CSTR(10,10),NSTR,MSTR
      COMMON /OPTSYS/ A1(10,10),B1(10,10),C1(10,10)
      COMMON /QRVF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
      COMMON /VECTS/ REF(10),XO(10)
      COMMON /INOUT/ ISTAT(10),IOPT(10)
      WRITE (5,10)
10    FORMAT (2X,'MULTIVARIABLE DESIGN PACKAGE',/,1X,'**********',
     1'**********************',//)
      IARS=0
      ISTS=0
      IPLT=0
       INS=0
      KSCL=1
      ICOM=0
      NORRDS=0
      NOCRDS=0
      CALL GINO
C
C     CHOOSE OPTIONS
C
15    WRITE (5,20)
20    FORMAT (1X,'TYPE OPTION, ??? FOR OPTION LISTING',/)
      READ (5,30) OPT
30    FORMAT (A3)
      IF (OPT.EQ.'???') CALL OPTLST
      IF (OPT.EQ.'INP') CALL INPUT(NA,MA,INS,IARS,ISTS,ICOM)
      IF (OPT.EQ.'OUT') GO TO 100
      IF (OPT.EQ.'CHE') CALL CHECK(ICOM)
      IF (OPT.EQ.'MOD') CALL MODIFY(IARS,ICOM,INS,NA,MA)
      IF (OPT.EQ.'STO') CALL STORE1(ICOM)
      IF (OPT.EQ.'OPT') CALL OPTIM(ICOM,NORRDS,NOCRDS)
      IF (OPT.EQ.'COM') CALL CMPSTR(ICOM,INS,NA,MA,IARS)
      GO TO 15
100   CONTINUE
      CALL DEVEND
      CALL FINISH
      END
C
      SUBROUTINE OPTLST
CC
      WRITE (5,10)
10    FORMAT (2X,'OPTIONS AVAILABLE',/,1X,'*****************',//,1X,
     1'INP- INPUTS SYSTEM A,B AND C MATRICES',/,1X,
     3'OUT- EXITS ROUTINE',/,1X,
     5'CHE- DISPLAYS EXISTING DATA',/,1X,
     6'MOD- ALLOWS EXISTING DATA TO BE MODIFIED',/,1X,
     8'COM- DEFINES FORWARD PATH COMPENSATOR',/,1X,
     9'STO- ALLOWS CURRENT SYSTEM DATA TO BE WRITTEN TO DISC',/,1X,
     2'OPT- EVALUATES OPTIMAL CONTROL GAINS & PLOTS RESPONSES',/)
      RETURN
      END
C
C  input system data
C
      SUBROUTINE INPUT(NA,MA,INS,IARS,ISTS,ICOM)
      COMMON /COMP/ACOMP(10,10),BCOMP(10,10),CCOMP(10,10),
     1DCOMP(10,10),NC,MC
      COMMON /SYST/A(10,10),B(10,10),C(10,10),N,M
      IF (ICOM.EQ.0) GO TO 5
      CALL DELETE (ICOM,INS,NA,MA,IARS)
5     WRITE (5,600)
600   FORMAT (1X,'IS THE SYSTEM DATA TO BE RETRIEVED FROM DISK ? (Y/N)')
      READ (5,700) REP
700   FORMAT (A1)
      IF (REP.EQ.'N') GO TO 10
      IF (REP.NE.'Y') GO TO 5
      WRITE (5,800)
800   FORMAT (1X,'ENTER A 5 CHARACTER FILENAME')
      READ (5,90) FNAM
90    FORMAT (5A)
      OPEN (UNIT=1,DEVICE='DSK',ACCESS='SEQIN',MODE='ASCII',FILE=FNAM)
      READ (1,50) N,M
      DO 300 I=1,N
      DO 300 J=1,N
      CALL MAP(N,IR,IC,I,J)
300   READ (1,110) A(IR,IC)
      DO 400 I=1,N
      DO 400 J=1,M
      CALL MAP(N,IR,IC,I,J)
400   READ (1,110) B(IR,IC)
      DO 500 I=1,M
      DO 500 J=1,N
      CALL MAP(M,IR,IC,I,J)
500   READ (1,110) C(IR,IC)
      CLOSE (UNIT=1,DEVICE='DSK',FILE=FNAM)
110   FORMAT (F12.6)
      WRITE (5,120)
120   FORMAT (1X,'SYSTEM RETRIEVED FROM DISK')
      NA=N
      MA=M
      GO TO 1
10    WRITE (5,20)
20    FORMAT (1X,'TYPE THE ORDER OF THE STATE A MATRIX, I2 FORMAT',/)
      READ (5,30) N
30    FORMAT (I2)
      NA=N
      WRITE (5,40)
40    FORMAT (1X,'TYPE THE NUMBER OF INPUTS, (= NO. OF OUTPUTS),',/,
     11X,'I2 FORMAT',/)
      READ (5,50) M
50    FORMAT (I2)
      MA=M
C
C     READ THE A,B,C MATRICES
C
      WRITE (5,60)
60    FORMAT (1X,'TYPE IN THE ELEMENTS OF THE A MATRIX UNDER MREAD',/)
      CALL MREAD(A,N,N,5)
      WRITE (5,70)
```

```
 70     FORMAT (1X,'TYPE IN THE ELEMENTS OF THE B MATRIX UNDER MREAD',/)
        CALL MREAD(B,N,M,5)
        WRITE (5,80)
 80     FORMAT (1X,'TYPE IN THE ELEMENTS OF THE C MATRIX UNDER MREAD',/)
        CALL MREAD(C,M,N,5)
 1      INS=1
        IARS=0
        ISIS=0
        ICOM=0
        RETURN
        END
C
c write out system etc. matrices
C
        SUBROUTINE CHECK(ICOM)
        COMMON /SYST/A(10,10),B(10,10),C(10,10),N,M
        COMMON /COMP/ACOMP(10,10),BCOMP(10,10),CCOMP(10,10),
       1DCOMP(10,10),NC,MC
        COMMON /SYSTR/ASTR(10,10),BSTR(10,10),CSTR(10,10),NSTR,MSTR
        IF (ICOM.EQ.0) GO TO 100
 5      WRITE (5,10)
 10     FORMAT (1X,'*OPTIONS*',/,1X,'TYPE ONE OF THE FOLLOWING',/,
       11X,'1- DISPLAY COMPENSATOR ONLY',/,1X,'2- DISPLAY SYSTEM ',
       2'TO BE CONTROLLED',/,1X,'3- DISPLAY AUGMENTED SYSTEM',/,1X,
       3'4- RETURN')
        READ (5,20) KANS
 20     FORMAT (I1)
        IF (KANS.GE.1.AND.KANS.LE.4) GO TO 30
        WRITE (5,40)
 40     FORMAT (1X,'UNRECOGNISED CHARACTER, TRY AGAIN')
        GO TO 5
 30     IF (KANS.EQ.1) GO TO 50
        IF (KANS.EQ.2) GO TO 80
        IF (KANS.EQ.3) GO TO 100
        RETURN
 50     WRITE (5,60) NC,NC
 60     FORMAT (1X,'COMPENSATOR STATE MATRIX DESCRIPTION',/,1X,
       1'AC MATRIX ORDER= ',I2,/,1X,'COMPENSATOR DEFINED AS',1X,
       212,1X,'INPUT')
        IF (ICOM.EQ.0) GO TO 70
        CALL MWRITE(ACOMP,NC,NC,20H  COMSTR AC MATRIX  ,5)
        CALL MWRITE(BCOMP,NC,NC,20H  COMSTR BC MATRIX  ,5)
        CALL HOLD
        CALL MWRITE(CCOMP,NC,NC,20H  COMSTR CC MATRIX  ,5)
 70     CALL MWRITE(DCOMP,NC,NC,20H  COMSTR DC MATRIX  ,5)
        GO TO 5
 80     WRITE (5,90) NSTR,MSTR
 90     FORMAT (1X,'SYSTEM TO BE CONTROLLED, STATE MATRIX ',
       1'DESCRIPTION',/,1X,'A MATRIX ORDER = ',I2,/,1X,'SYSTEM ',
       2'DEFINED AS',1X,I2,1X,'INPUT')
        CALL MWRITE(ASTR,NSTR,NSTR,20H   SYSTEM A MATRIX  ,5)
        CALL HOLD
        CALL MWRITE(BSTR,NSTR,MSTR,20H   SYSTEM B MATRIX  ,5)
        CALL MWRITE(CSTR,NSTR,NSTR,20H   SYSTEM C MATRIX  ,5)
        GO TO 5
 100    WRITE (5,110) N,M
 110    FORMAT (1X,'SYSTEM STATE MATRIX DESCRIPTION',/,1X,
       1'A MATRIX ORDER = ',I2,/,1X,'SYSTEM DEFINED AS',1X,I2,1X,
       2'INPUT')
        CALL MWRITE(A,N,N,20H   SYSTEM A MATRIX  ,5)
        CALL HOLD
        CALL MWRITE(B,N,M,20H   SYSTEM B MATRIX  ,5)
        CALL MWRITE(C,M,N,20H   SYSTEM C MATRIX  ,5)
        IF (ICOM.NE.0) GO TO 5
        RETURN
        END
C
C    PAUSE ROUTINE
C
        SUBROUTINE HOLD
        WRITE (5,10)
 10     FORMAT (1X,'TYPE RETURN TO CONTINUE')
        READ (5,20) LOOSE
 20     FORMAT (A1)
        RETURN
        END
C
C     COMPENSATOR DEFINITION
C
        SUBROUTINE CMPSTR(ICOM,INS,NA,MA,IARS)
        DIMENSION BCCOMP(10,10),BDCOMP(10,10)
        COMMON /SYST/A(10,10),B(10,10),C(10,10),N,M
        COMMON /COMP/ACOMP(10,10),BCOMP(10,10),CCOMP(10,10),
       1DCOMP(10,10),NC,MC
        COMMON /SYSTR/ASTR(10,10),BSTR(10,10),CSTR(10,10),NSTR,MSTR
        IF (INS.NE.0) GO TO 5
        WRITE (5,6)
 6      FORMAT (1X,'NO INPUT DATA!')
        RETURN
 5      IF (ICOM.NE.0) GO TO 20
        WRITE (5,10)
 10     FORMAT (1X,'THIS OPTION ALLOWS THE INTRODUCTION OF A ',
       1'FORWARD PATH',/,1X,'COMPENSATOR INTO THE SYSTEM',/,1X
       2,'COMPENSATOR MUST BE DEFINED IN STATE MATRIX FORM I.E.',
       3/,1X,'S(AC,BC,CC,DC)',/,1X,'ANY MATRIX MAY BE CHOSEN AS ',
       4'NULL',/,1X,'NOTE: MAXIMUM STATE ORDER IS 10',/)
        GO TO 30
 20     WRITE (2,40)
 40     FORMAT (1X,'COMPENSATOR PREVIOUSLY DEFINED',/,1X,'TYPE '
       1,'D TO DELETE EXISTING COMPENSATOR, I TO REDEFINE',/,1X,
       2'COMPENSATOR, E TO RETURN TO MASTER')
        GO TO 70
 30     WRITE (5,50)
 50     FORMAT (1X,'TYPE I TO DEFINE COMPENSATOR, E TO RETURN TO'
       1,' MASTER')
 70     READ (5,60) RES
 60     FORMAT (A1)
        IF (RES.EQ.'I'.OR.RES.EQ.'D'.OR.RES.EC.'E') GO TO 80
        WRITE (5,90)
 90     FORMAT (1X,'UNRECOGNISED CHARACTER, TRY AGAIN')
        GO TO 70
 80     IF (RES.EQ.'I') GO TO 100
        IF (RES.EQ.'D'.AND.ICOM.NE.0) GO TO 110
        RETURN
        ENTRY DELETE
 110    CALL MNULL(A,N,N)
        CALL MNULL(B,N,M)
        CALL MNULL(C,M,N)
        N=NSTR
        M=MSTR
```

```
      CALL MEQUIV(ASTR,A,N,N)
      CALL MEQUIV(BSTR,B,N,M)
      CALL MEQUIV(CSTR,C,M,N)
      NA=N
      MA=M
      CALL MNULL(ACOMP,NC,NC)
      CALL MNULL(BCOMP,NC,MC)
      CALL MNULL(CCOMP,MC,NC)
      CALL MNULL(DCOMP,MC,MC)
      ICOM=0
      RETURN
100   NC=P
      WRITE (5,120) NC
120   FORMAT (1X,'*COMPENSATOR DEFINITION*',/,1X,'COMPENSATOR',
     1' IS ASSUMED SQUARE AND OF INPUT/OUTPUT',/,1X,'DIMENSION',
     21X,I2)
125   WRITE (5,130)
130   FORMAT (1X,'TYPE THE ORDER OF THE COMPENSATOR  ''A'' ',/,
     11X,'MATRIX (AC), I2 FORMAT')
      READ (5,140) NC
140   FORMAT (I2)
      IF (NC.GE.0) GO TO 520
      WRITE (5,530)
530   FORMAT (1X,'RUBBISH')
      GO TO 125
520   IF (NC.EQ.0) GO TO 290
      IF (ICOM.NE.0) GO TO 500
      ISTAT=NC+N
      GO TO 510
500   ISTAT=NC+NSTR
510   IF (ISTAT.LT.10) GO TO 180
      WRITE (5,135)
135   FORMAT (1X,'MAX. STATE DIMENSION EXCEEDED')
      GO TO 125
180   WRITE (5,200)
200   FORMAT (1X,'INPUT THE ELEMENTS OF THE AC MATRIX,'/,1X,
     1'UNDER MREAD')
      CALL MREAD(ACOMP,NC,NC,5)
230   WRITE (5,250)
250   FORMAT (1X,'INPUT THE ELEMENTS OF THE BC MATRIX,',/,1X,
     1'UNDER MREAD')
      CALL MREAD(BCOMP,NC,MC,5)
280   WRITE (5,300)
300   FORMAT (1X,'INPUT THE ELEMENTS OF THE CC MATRIX,',/,1X,
     1'UNDER MREAD')
      CALL MREAD(CCOMP,MC,NC,5)
290   WRITE (5,310)
310   FORMAT (1X,'COMPENSATOR ''D'' MATRIX, TYPE 0 TO SET NULL',
     1' MATRIX',/,1X,'1 TO INPUT ELEMENTS')
      READ (5,160) KST
160   FORMAT (I1)
      IF (KST.EQ.0) GO TO 320
      IF (KST.EQ.1) GO TO 330
      WRITE (5,90)
      GO TO 290
320   CALL MNULL(DCOMP,MC,MC)
      GO TO 340
330   WRITE (5,350)
350   FORMAT (1X,'INPUT THE ELEMENTS OF THE DC MATRIX,',/,1X,
     1'UNDER MREAD')
      CALL MREAD(DCOMP,MC,MC,5)
340   IF (ICOM.EQ.1) GO TO 360
      CALL MEQUIV(A,ASTR,N,N)
      CALL MEQUIV(B,BSTR,N,M)
      CALL MEQUIV(C,CSTR,M,N)
      NSTR=N
      MSTR=M
      ENTRY NEWMAT
360   IF (NC.EQ.0) GO TO 600
      N=NSTR+NC
      CALL MMULT(BSTR,CCOMP,BCCOMP,NSTR,N,NC)
      CALL MMULT(BSTR,DCOMP,BDCOMP,NSTR,N,M)
C
C     SET UP A NEW MATRIX
C
      LOOP=1
      I1=0
      ITOT=N*NSTR
      DO 1100 I=1,ITOT
      IF (LOOP.LE.N) GO TO 1030
      LOOP=2
      GO TO 1040
1030  LOOP=LOOP+1
1040  J=INT((I-1)/10.0)+1
      IROW=I-(J-1)*10
      IF (LOOP.GT.NSTR+1) GO TO 1010
      I1=I1+1
      J1=INT((I1-1)/10.0)+1
      IROW1=I1-(J1-1)*10
      A(IROW,J)=ASTR(IROW1,J1)
      GO TO 1100
1010  A(IROW,J)=0.0
1100  CONTINUE
      LOOP=1
      I1=0
      I2=0
      ITOT=ITOT+1
      ITOTA=N**2
      DO 1200 I=ITOT,ITOTA
      IF (LOOP.LE.N) GO TO 1050
      LOOP=2
      GO TO 1060
1050  LOOP=LOOP+1
1060  J=INT((I-1)/10.0)+1
      IROW=I-(J-1)*10
      IF (LOOP.GT.NSTR+1) GO TO 1070
      I1=I1+1
      J1=INT((I1-1)/10.0)+1
      IROW1=I1-(J1-1)*10
      A(IROW,J)=BCCOMP(IROW1,J1)
      GO TO 1200
1070  I2=I2+1
      J2=INT((I2-1)/10.0)+1
      IROW2=I2-(J2-1)*10
      A(IROW,J)=ACOMP(IROW2,J2)
1200  CONTINUE
C
C  SET UP NEW B MATRIX
C
      LOOP=1
```

```
      I1=0
      I2=0
      ITOT=M*N
      DO 2100 I=1,ITOT
      IF (LOOP.LE.N) GO TO 2030
      LOOP=2
      GO TO 2040
2030  LOOP=LOOP+1
2040  J=INT((I-1)/10.0)+1
      IROW=I-(J-1)*10
      IF (LOOP.GT.NSTR+1) GO TO 2010
      I1=I1+1
      J1=INT((I1-1)/10.0)+1
      IROW1=I1-(J1-1)*10
      B(IROW,J)=BDCOMP(IROW1,J1)
      GO TO 2100
2010  I2=I2+1
      J2=INT((I2-1)/10.0)+1
      IROW2=I2-(J2-1)*10
      B(IROW,J)=BCOMP(IROW2,J2)
2100  CONTINUE
C
C     SET UP NEW C MATRIX
C
      ITOTC=M*NSTR+1
      ITOTN=M*NC+ITOTC-1
      DO 3100 LVAR=ITOTC,ITOTN
      J1=INT((LVAR-1)/10.0)+1
      IROW=LVAR-(J1-1)*10
      C(IROW,J1)=0.0
3100  CONTINUE
      NA=N
      MA=M
      ICON=1
      GO TO 610
600   CALL PMULT(BSTR,DCOMP,B,NSTR,M,N)
      CALL PEQUIV(ASTR,A,NSTR,NSTR)
      CALL PEQUIV(CSTR,C,M,NSTR)
      ICON=1
610   IF (IARS.GT.1) IARS=1
      RETURN
      END
C
C     MODIFY ROUTINES
C
      SUBROUTINE MODIFY(IARS,ICON,INS,NA,MA)
      COMMON /SYST/A(10,10),B(10,10),C(10,10),N,M
      COMMON /COMP/ACOMP(10,10),BCOMP(10,10),CCOMP(10,10),
     1DCOMP(10,10),NC,MC
      COMMON /SYSTR/ASTR(10,10),BSTR(10,10),CSTR(10,10),NSTR,MSTR
      IF (ICON.NE.0) GO TO 10
      CALL PEQUIV(A,ASTR,N,N)
      CALL PEQUIV(B,BSTR,N,M)
      CALL PEQUIV(C,CSTR,M,N)
      NSTR=N
      MSTR=M
      CALL MODSYS
      CALL PEQUIV(ASTR,A,N,N)
      CALL PEQUIV(BSTR,B,N,N)
      CALL PEQUIV(CSTR,C,M,N)
      GO TO 40
10    WRITE (5,20)
20    FORMAT (1X,'DO YOU WISH TO MODIFY SYSTEM OR COMPENSATOR',
     1' MATRICES?',/,1X,'TYPE SYS TO MODIFY SYSTEM, COM TO ',
     2'MODIFY COMPENSATOR,',/,1X,'RET TO RETURN TO MASTER')
      READ (5,30) REP
30    FORMAT (A3)
      IF (REP.NE.'SYS'.AND.REP.NE.'COM'.AND.REP.NE.'RET')
     1GO TO 10
      IF (REP.EQ.'SYS') CALL MODSYS
      IF (REP.EQ.'COM') CALL MODCOM
      CALL NEWMAT(ICOM,INS,NA,MA,IARS)
40    IF (IARS.GT.1) IARS=1
      RETURN
      END
C
C     SYSTEM MODIFY
C
      SUBROUTINE MODSYS
      COMMON /SYSTR/ASTR(10,10),BSTR(10,10),CSTR(10,10),NSTR,MSTR
      WRITE (5,10)
10    FORMAT (1X,'THIS OPTION ALLOWS INDIVIDUAL ELEMENTS OF THE',/,1X
     1,'A,B OR C MATRICES TO BE MODIFIED.',/,1X,'TO MODIFY SYSTEM ',
     2'ORDER OR NUMBER OF INPUTS USE INP OPTION',/)
15    WRITE (5,20)
20    FORMAT (1X,'TYPE THE SYSTEM MATRIX TO BE MODIFIED, A,B OR C'
     1,/,1X,'TYPE E TO RETURN TO MASTER',/)
      READ (5,30) Z
30    FORMAT (A1)
      IF (Z.NE.'A'.AND.Z.NE.'B'.AND.Z.NE.'C'.AND.Z.NE.'E') GO TO 15
      IF (Z.EQ.'E') GO TO 70
35    WRITE (5,80)
80    FORMAT (1X,'TYPE ROW/COLUMN INDEX OF ELEMENT TO BE MODIFIED,'
     1,/,' SEPERATED BY A SPACE, I2 FORMAT',/)
      READ (5,90) IR,IC
90    FORMAT (I2,1X,I2)
      WRITE (5,100)
100   FORMAT (1X,'NEW VALUE ?',/)
      READ (5,110) VNEW
110   FORMAT (F10.4)
      IF (Z.EQ.'A') GO TO 40
      IF (Z.EQ.'B') GO TO 50
      IF (Z.EQ.'C') GO TO 60
40    CALL MAP(NSTR,IRA,ICA,IR,IC)
      WRITE (5,160) IR,IC,ASTR(IRA,ICA),VNEW
      ASTR(IRA,ICA)=VNEW
      GO TO 120
50    CALL MAP(NSTR,IRB,ICB,IR,IC)
      WRITE (5,160) IR,IC,BSTR(IRB,ICB),VNEW
      BSTR(IRB,ICB)=VNEW
      GO TO 120
60    CALL MAP(MSTR,IRC,ICC,IR,IC)
      WRITE (5,160) IR,IC,CSTR(IRC,ICC),VNEW
      CSTR(IRC,ICC)=VNEW
120   WRITE (5,130)
130   FORMAT (1X,'TYPE F TO MODIFY ANOTHER ELEMENT, M TO MODIFY AN',/,
     11X,'ELEMENT OF ANOTHER MATRIX, E TO RETURN TO MASTER',/)
      READ (5,140) X
140   FORMAT (A1)
      IF (X.NE.'F'.AND.X.NE.'M'.AND.X.NE.'E') GO TO 120
```

```
      IF (X.EQ.'F') GO TO 35
      IF (X.EQ.'H') GO TO 15
70    CONTINUE
160   FORMAT (1X,'ELEMENT ',I2,1X,I2,'    OLD VALUE = ',F10.4,/,15X,'NEW
     1 VALUE = ',F10.4)
      RETURN
      END
C
C     COMPENSATOR MODIFY
C
      SUBROUTINE MODCOM
      COMMON /COMP/ACOMP(10,10),BCOMP(10,10),CCOMP(10,10),
     1DCOMP(10,10),NC,MC
      WRITE (5,10)
10    FORMAT (1X,'THIS OPTION ALLOWS INDIVIDUAL ELEMENTS OF THE',/,1X
     1,'AC,BC,CC OR DC MATRICES TO BE MODIFIED.',/,1X,'TO MODIFY ',
     2'COMPENSATOR ORDER USE COM OPTION')
15    WRITE (5,20)
20    FORMAT (1X,'TYPE THE COMPENSATOR MATRIX TO BE MODIFIED, '
     1,'AC,BC,CC OR DC',/,1X,'TYPE EX TO RETURN TO MASTER',/)
      READ (5,30) Z
30    FORMAT (A2)
      IF (Z.NE.'AC'.AND.Z.NE.'BC'.AND.Z.NE.'CC'.AND.Z.NE.'DC'.AND.Z.NE.
     1'EX') GO TO 15
      IF (Z.EQ.'EX') GO TO 70
35    WRITE (5,80)
80    FORMAT (1X,'TYPE ROW/COLUMN INDEX OF ELEMENT TO BE MODIFIED,'
     1,/,'SEPERATED BY A SPACE, I2 FORMAT',/)
      READ (5,90) IR,IC
90    FORMAT (I2,1X,I2)
      WRITE (5,100)
100   FORMAT (1X,'NEW VALUE ?',/)
      READ (5,110) VNEW
110   FORMAT (F10.4)
      IF (Z.EQ.'AC') GO TO 40
      IF (Z.EQ.'BC') GO TO 50
      IF (Z.EQ.'CC') GO TO 60
      IF (Z.EQ.'DC') GO TO 150
40    CALL MAP(NC,IRAC,ICAC,IR,IC)
      WRITE (5,160) IR,IC,ACOMP(IRAC,ICAC),VNEW
      ACOMP(IRAC,ICAC)=VNEW
      GO TO 120
50    CALL MAP(NC,IRBC,ICBC,IR,IC)
      WRITE (5,160) IR,IC,BCOMP(IRBC,ICBC),VNEW
      BCOMP(IRBC,ICBC)=VNEW
      GO TO 120
60    CALL MAP(NC,IRCC,ICCC,IR,IC)
      WRITE (5,160) IR,IC,CCOMP(IRCC,ICCC),VNEW
      CCOMP(IRCC,ICCC)=VNEW
      GO TO 120
150   CALL MAP(NC,IRDC,ICDC,IR,IC)
      WRITE (5,160) IR,IC,DCOMP(IRDC,ICDC),VNEW
      DCOMP(IRDC,ICDC)=VNEW
120   WRITE (5,130)
130   FORMAT (1X,'TYPE F TO MODIFY ANOTHER ELEMENT, H TO MODIFY AN',/,
     11X,'ELEMENT OF ANOTHER MATRIX, E TO RETURN TO MASTER',/)
      READ (5,140) X
140   FORMAT (A1)
      IF (X.NE.'F'.AND.X.NE.'H'.AND.X.NE.'E') GO TO 120
      IF (X.EQ.'F') GO TO 35
      IF (X.EQ.'H') GO TO 15
70    CONTINUE
160   FORMAT (1X,'ELEMENT ',I2,1X,I2,'    OLD VALUE = ',F10.4,15X,'NEW
     1 VALUE = ',F10.4)
      RETURN
      END
C
C utility routine for matrix row/column indices
C
      SUBROUTINE MAP(IB,IRA,ICA,IR,IC)
      IRA=(IC-1)*IB+IR
      IF (IRA.LT.10) GO TO 10
      ICA=INT(IRA/10.)+1
      IRA=IRA-(ICA-1)*10
      GO TO 20
10    ICA=1
20    CONTINUE
      RETURN
      END
C
C plotting routines
C
      SUBROUTINE PLTXES(YMAX1,YMIN1,INT1,T,ICX,ICY,YSCL,TSCL)
C     PLOT AXES & DATA POINTS
C
5     WRITE (5,10)
10    FORMAT (1X,'Output to T4010 or Calcomp ?, (T/C)')
      READ (5,15) REP
15    FORMAT (A1)
      IF (REP.EQ.'T') GO TO 20
      IF (REP.EQ.'C') GO TO 30
      GO TO 5
20    CALL TKINIT
      GO TO 40
30    CALL PLTINT
40    CALL PICCLE
      YSCL=(ABS(YMAX1)+ABS(YMIN1))/550.0
      TSCL=(INT1*T)/860.0
      ICY=100+INT(ABS(YMIN1)/YSCL)
      ICX=138
      ISX=50
      ISY=50
      IEX=998
      IEY=710
      ITICX=50
      ITICY=50
      CALL AXIS(ISX,ISY,ICX,ICY,IEX,IEY,ITICX,ITICY)
C
C     WRITE SCALING ONTO AXES
C          SCALE X AXIS
C
      DO 470 I=4,16,4
      ITICA=50*(I-1)+ICX
      ICYA=ICY-20
      TICA=I*TSCL*50.0
      CALL TPLOT(0,ITICA,ICYA)
      CALL CHAFLO(TICA,10)
470   CONTINUE
C
C     SCALE Y AXIS
```

```
C
      IMAX=INT((60.0+(YMAX1/YSCL))/50.0)+1
      IMIN=INT((50.0+((-YMIN1)/YSCL))/50.0)
      DO 480 I=1,IMAX,2
      ITIC=50*(I-1)+ICY
      TIC=YSCL*50.0*(I-1)
      CALL TPLOT(0,0,ITIC)
      CALL CHAFLO(TIC,10)
480   CONTINUE
      IF (IMIN.LE.1) GO TO 490
      DO 490 I=2,IMIN,2
      ITIC=ICY-50*I
      TIC=-YSCL*50.0*I
      CALL TPLOT(0,0,ITIC)
      CALL CHAFLO(TIC,10)
490   CONTINUE
      RETURN
      END
C     DEFINE STEP SIZE
C
      SUBROUTINE STEP1(T,INT1,NFL1)
      WRITE (5,90)
90    FORMAT (1X,'THE MAXIMUM NUMBER OF TIME STEPS IS 300',/,1X,
     1'TYPE DESIRED NUMBER OF STEPS, I3 FORMAT')
      READ (5,100) INT1
100   FORMAT (I3)
      WRITE (5,110)
110   FORMAT (1X,'TYPE THE STEP SIZE')
      READ (5,120) T
120   FORMAT (F15.8)
      NFL1=1
      RETURN
      END
C
C     SCALING ROUTINES
C
      SUBROUTINE SCALEX(XMAX,XMIN,XSTP,ICX,ICY,KX)
      IF (ICY.LT.20) GO TO 10
      ICYA=ICY-20
      GO TO 20
10    ICYA=ICY+10
20    IF (KX.EQ.1) GO TO 100
      GO TO 150
100   IMAX=INT(XMAX/XSTP/50.0)
      IMIN=INT(XMIN/XSTP/50.0)+2
      ICXA=5-INT(XMIN/XSTP)
      ISGN=1
      GO TO 170
150   IF (KX.EQ.2) GO TO 160
      GO TO 220
160   IMAX=INT((-XMIN)/XSTP/50.0)
      IMIN=INT((-XMAX)/XSTP/50.0)+2
      ICXA=1018-INT(XMAX/XSTP)
      ISGN=-1
170   DO 200 J=IMIN,IMAX,4
      ITICA=50*(J-ISGN)*ISGN+ICXA
      TICA=J*XSTP*50.0*ISGN
      CALL TPLOT(0,ITICA,ICYA)
      CALL CHAFLO(TICA,10)
200   CONTINUE
      GO TO 400
220   IMAX=INT((XMAX/XSTP)/50.0)
      IMIN=-(1+INT((XMIN/XSTP)/50.0))
      IF (XMIN.EQ.0.0) IMIN=0
      IF (IMAX.LT.4) GO TO 300
      DO 300 I=4,IMAX,4
      ITICA=(I-1)*50+ICX
      IF (ITICA) 320,340,330
320   ITICA=5
      GO TO 340
330   IF (883-ITICA) 350,340,340
350   ITICA=883
340   TICA=I*50.0*XSTP
      CALL TPLOT(0,ITICA,ICYA)
      CALL CHAFLO(TICA,10)
300   CONTINUE
      IF (IMIN.LT.4) GO TO 400
      DO 400 I=4,IMIN,4
      ITICA=ICX-50*(I+1)
      IF (ITICA.GT.5) GO TO 430
      ITICA=5
430   TICA=-I*50.0*XSTP
      CALL TPLOT(0,ITICA,ICYA)
      CALL CHAFLO(TICA,10)
400   CONTINUE
      RETURN
      END
C
      SUBROUTINE SCALEY(YMAX,YMIN,YSTP,ICX,ICY,KY)
      IF (ICX.GT.883) GO TO 20
      ICXA=ICX+5
      GO TO 10
20    ICXA=ICX-140
10    IF (KY.EQ.1) GO TO 100
      GO TO 150
100   IMAX=INT(YMAX/YSTP/50.0)
      IMIN=INT(YMIN/YSTP/50.0)+1
      ICYA=5-INT(YMIN/YSTP)
      ISGN=1
      GO TO 170
150   IF (KY.EQ.2) GO TO 160
      GO TO 220
160   IMAX=INT((-YMIN)/YSTP/50.0)
      IMIN=INT((-YMAX)/YSTP/50.0)+1
      ICYA=755-INT(YMAX/YSTP)
      ISGN=-1
170   DO 200 J=IMIN,IMAX,2
      ITICB=50*J*ISGN+ICYA
      TICB=J*YSTP*50.0*ISGN
      CALL TPLOT(0,ICXA,ITICB)
      CALL CHAFLO(TICB,10)
200   CONTINUE
      GO TO 400
220   IMAX=INT((YMAX/YSTP)/50.0)
      IMIN=INT(((-YMIN)/YSTP)/50.0)
      IF (IMAX.LT.2) GO TO 300
      DO 300 I=2,IMAX,2
      ITICB=50*I+ICY
      TICB=I*50.0*YSTP
      CALL TPLOT(0,ICXA,ITICB)
```

```
      CALL CHAFLO(TICB,10)
300   CONTINUE
      IF (IMIN.LT.2) GO TO 100
      DO 400 I=2,IMIN,2
      ITICB=ICY-50*I
      TICB=-I*50.0*YSTP
      CALL TPLOT(0,ICXA,ITICB)
      CALL CHAFLO(TICB,10)
400   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE STORE1(ICOM)
      COMMON /SYST/A(10,10),B(10,10),C(10,10),N,M
      COMMON /SYSTR/ASTR(10,10),BSTR(10,10),CSTR(10,10),NSTR,MSTR
5     WRITE (5,10)
10    FORMAT (1X,'THIS OPTION ALLOWS CURRENT SYSTEM DATA TO BE',/,
     11X,'STORED ON DISK.',/,1X,'TYPE S TO STORE DATA, E TO RETURN')
      READ (5,20) REP
20    FORMAT (A1)
      IF (REP.EQ.'E') RETURN
      IF (REP.NE.'S') GO TO 5
      WRITE (5,30)
30    FORMAT (1X,'ENTER A 5 CHARACTER FILENAME')
      READ (5,40) FNAM
40    FORMAT (A5)
      OPEN (UNIT=1,DEVICE='DSK',ACCESS='SEQOUT',MODE='ASCII',FILE=FNAM)
      IF (ICOM.EQ.0) GO TO 100
      WRITE (1,50) NSTR,MSTR
      DO 200 I=1,NSTR
      DO 200 J=1,NSTR
      CALL MAP(NSTR,IR,IC,I,J)
200   WRITE (1,60) A(IR,IC)
      DO 300 I=1,NSTR
      DO 300 J=1,MSTR
      CALL MAP(NSTR,IR,IC,I,J)
300   WRITE (1,60) BSTR(IR,IC)
      DO 400 I=1,MSTR
      DO 400 J=1,NSTR
      CALL MAP(MSTR,IR,IC,I,J)
400   WRITE (1,60) CSTR(IR,IC)
      GO TO 110
100   WRITE (1,50) N,M
      DO 500 I=1,N
      DO 500 J=1,M
      CALL MAP(N,IR,IC,I,J)
500   WRITE (1,60) A(IR,IC)
      DO 600 I=1,N
      DO 600 J=1,M
      CALL MAP(N,IR,IC,I,J)
600   WRITE (1,60) B(IR,IC)
      DO 700 I=1,M
      DO 700 J=1,N
      CALL MAP(M,IR,IC,I,J)
700   WRITE (1,60) C(IR,IC)
110   CLOSE (UNIT=1,DEVICE='DSK',FILE=FNAM)
50    FORMAT (I2)
60    FORMAT (F12.6)
      RETURN
      END
C
C
```

```
C
C     * Evaluate time responses *
C
      SUBROUTINE EXEC1(IGAIN,INODO,N,M,NFL1,NFL2,NFL3,T,INT1,KYFLG,
     1LSTAT)
      COMMON /INOUT/ ISTAT(10),IOPT(10)
      COMMON /VECTS/ REF(10),XO(10)
      COMMON /OPTSYS/ A1(10,10),B1(10,10),C1(10,10)
      COMMON /ORKF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
      DIMENSION DUM1(10,10),CLMAT(10,10),AA(10,10),BB(10,10),CC(10,10)
     1,PHI(10,10),DELTA(10,10),XOLD(10),FSO1(10,300)
      DIMENSION XNEW(10),YSO(10,300),XSO(10,300),DUM2(10)
     1,DUM3(10),DUM4(10),GAMMA(10,10),FSO(10,300),OM(10),OM1(10)
      NSEQ=0
      IF (IGAIN.EQ.1.AND.INODO.EQ.0) GO TO 90
      WRITE (5,5)
5     FORMAT (1X,'Evaluate or re-evaluate the state & reference gain
     1 matrices',/,1X,'before this option!',/)
      RETURN
90    IF (NFL1.NE.0) GO TO 100
      WRITE (5,10)
10    FORMAT (1X,'Step size undefined, use option 1')
      RETURN
100   IF (NFL2.NE.0) GO TO 105
      WRITE (5,15)
15    FORMAT (1X,'Input states undefined, use option 2')
      RETURN
105   IF (NFL3.NE.0) GO TO 110
      WRITE (5,20)
20    FORMAT (1X,'Outputs for display undefined, use option 3')
      RETURN
110   WRITE (5,25)
25    FORMAT (1X,'Do you wish to specify a noise input ? (Y/N)')
      READ (5,30) ANS
30    FORMAT (A1)
      IF (ANS.EQ.'N') GO TO 300
      IF (ANS.NE.'Y') GO TO 110
      WRITE (5,35)
35    FORMAT (1X,'Input the number of noise sequences, I2 format')
      READ (5,40) NSEQ
40    FORMAT (I2)
      CALL SNOISE(GAMMA,FSO,NSEQ,N,INT1,T)
      GO TO 305
300   CALL FNULL(GAMMA,10,10)
      CALL FNULL(FSO1,10,300)
305   CALL FMULT(B1,GKF1,DUM1,N,N,N)
      CALL FADD(A1,DUM1,CLMAT,N,N)
      CALL FMULT(B1,GKF2,DUM1,N,N,N)
      CALL FNEG(DUM1,N,N)
      IF (NSEQ.EQ.0) GO TO 405
      DO 400 L=1,INT1
      DO 410 J=1,NSEQ
      CALL MAP(NSEQ,IR,IC,J,L)
410   OM(J)=FSO(IR,IC)
      CALL FMULT(GAMMA,OM,OM1,N,NSEQ,1)
      DO 420 J=1,N
      CALL MAP(N,IR,IC,J,L)
400   FSO1(IR,IC)=OM1(J)
405   CALL MWRITE(CLMAT,N,N,20HClosed loop A matrix,5)
      CALL MWRITE (XO,N,1,20H--------------------,5)
      CALL MWRITE (REF,N,1,20H--------------------,5)
      CALL HOLD
      CONVAC=1.0E-06
      CALL PHIDEL(CLMAT,DUM1,AA,BB,CC,PHI,DELTA,N,N,T,CONVAC)
      DO 115 I=1,N
      XOLD(I)=XO(I)
115   XSO(I,1)=XOLD(I)
      CALL MMULT(C1,XOLD,XNEW,N,N,1)
      DO 120 I=1,N
120   YSO(I,1)=XNEW(I)
      INT2=INT1-1
      CALL MMULT(DELTA,REF,DUM2,N,N,1)
      DO 130 L=2,INT1
      DO 420 I=1,N
      CALL MAP(N,IR,IC,I,L)
420   OM(I)=FSO1(IR,IC)
      CALL MMULT(PHI,XOLD,DUM3,N,N,1)
      CALL MADD(DUM3,DUM2,XNEW,N,1)
      CALL MADD(XNEW,OM,XNEW,N,1)
      CALL MMULT(C1,XNEW,DUM4,M,N,1)
      DO 135 K=1,M
      IF (DUM4(K).GT.1.0E04) DUM4(K)=1.0E04
      IF (DUM4(K).LT.-1.0E04) DUM4(K)=-1.0E04
      CALL MAP(M,KR,KC,K,L)
135   YSO(KR,KC)=DUM4(K)
      DO 140 K=1,N
      CALL MAP(N,KR,KC,K,L)
      XSO(KR,KC)=XNEW(K)
140   XOLD(K)=XNEW(K)
130   CONTINUE
      IF (LSTAT.EQ.1) GO TO 150
      IF (KYFLG.EQ.0) GO TO 210
      CALL SEQ(YSO,M,T,INT1,IOPT)
      GO TO 200
210   CALL CYCLE(YSO,M,T,INT1)
      GO TO 200
150   IF (KYFLG.EQ.0) GO TO 220
      CALL SEQ(XSO,N,T,INT1,ISTAT)
      GO TO 200
220   CALL CYCLE(XSO,N,T,INT1)
200   RETURN
      END
```

```
C
C
C These routines complement the MASTER programme and allow
C the user to evaluate the optimal control gain matrices
C for the system under investigation.  The time response
C of the optimally controlled system may also be evaluated
C and displayed.  These routines must be linked into the
C MASTER segment and are accessed via. this programme.
C
C
C23456789
      SUBROUTINE OPTIM(ICOM,NORRDS,NOQRDS)
      COMMON /OPTSYS/ A1(10,10),B1(10,10),C1(10,10)
      COMMON /QRKF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
      COMMON /VECTS/ REF(10),XO(10)
      COMMON /INOUT/ ISTAT(10),IOPT(10)
      COMMON /SYST/ A(10,10),B(10,10),C(10,10),N,M
      COMMON /SYSTR/ ASTR(10,10),BSTR(10,10),CSTR(10,10),NSTR,MSTR
C
      WRITE (5,10)
10    FORMAT (1X,'This option allows the user to interactively
     1 design a state variable',/,1X,'feedback control scheme
     2using a linear quadratic performance criteria.',/,1X,'The
     3 feedback gains are derived from an infinite time eigenvalue
     4',/,1X,'solution of the algebraic Ricatti equations.',/,1X,
     5'The user must initially define the Q and R matrices of
     6 appropriate orders.',/,1X,'Note:- any forward path compensators
     7 defined previously are ignored here.',/,1X,'A state reference
     8vector may also be defined for the transient response',/,1X,
     9'routines',/,1X,'CONTINUE (Y/N) ?',/)
      READ (5,15) ANS
15    FORMAT (A1)
      IF (ANS.EQ.'Y') GO TO 100
      RETURN
100   IF (ICOM.EQ.0) GO TO 110
      CALL MEQUIV(ASTR,A1,NSTR,NSTR)
      CALL MEQUIV(BSTR,B1,NSTR,MSTR)
      CALL MEQUIV(CSTR,C1,MSTR,NSTR)
      NSOPT=NSTR
      MSOPT=NSTR
      GO TO 115
110   CALL MEQUIV(A,A1,N,N)
      CALL MEQUIV(B,B1,N,M)
      CALL MEQUIV(C,C1,M,N)
      NSOPT=N
      MSOPT=M
      ENTRY RERDR(ICOM,NORRDS,NOQRDS)
115   WRITE (5,25) NSOPT
25    FORMAT (1X,'Input the R matrix ( ',I2,' square).  This matrix
     1 may be set to',/,1X,'the unit matrix by typing I, otherwise
     2 type return and input',/,1X,'the elements of R under MREAD',/)
      READ (5,15) ANS
      IF (ANS.NE.'I') GO TO 120
      DO 125 I=1,NSOPT
      DO 125 J=1,NSOPT
      EL=0.0
      IF (I.EQ.J) EL=1.0
      CALL MAP(MSOPT,IR,IC,I,J)
125   R(IR,IC)=EL
      GO TO 127
120   CALL MREAD(R,NSOPT,NSOPT,5)
127   IF (NORRDS.EQ.0) GO TO 129
      NORRDS=0
      RETURN
      ENTRY RERDQ(ICOM,NORRDS,NOQRDS)
129   WRITE (5,30) NSOPT
30    FORMAT (1X,'Input the Q matrix ( ',I2,' square).  This matrix
     1 may be set to',/,1X,'CTRANSPOSE*C by typing C, otherwise type
     2 return and input the elements',/,1X,'of Q under MREAD',/)
      READ (5,15) ANS
      IF (ANS.NE.'C') GO TO 130
      CALL MTRANS(CT,C1,MSOPT,NSOPT)
      CALL MMULT(CT,C1,Q,NSOPT,MSOPT,NSOPT)
      GO TO 137
130   CALL MREAD (Q,NSOPT,NSOPT,5)
137   IF (NQORDS.EQ.0) GO TO 139
      NOQRDS=0
      RETURN
139   IMODQ=1
      IGAIN=0
160   WRITE (5,35)
35    FORMAT (1X,'*OPTIONS*',//,1X,'Type one of the following',/,1X,
     1'1-  Modify or redefine Q and/or R',/,1X,'2-  Display any of
     2 Q, R, system or state & reference gain matrices',/,1X,'3-
     2Evaluate and display K1 and K2 matrices',/,1X,'4-
     3 Evaluate and plot time response for states or outputs (requires
     43)',/,1X,'5-  Store or retrieve data from disk',/,1X,'6-  Exit
     5to MASTER',/)
150   READ (5,40) JREP
40    FORMAT (I1)
      IF (JREP.LE.6.AND.JREP.GE.1) GO TO 140
      WRITE (5,45)
45    FORMAT (1X,'Unrecognised character, try again')
      GO TO 150
140   IF (JREP.EQ.1) CALL MODQR(IMODQ,NSOPT,MSOPT,NORRDS,NOQRDS
     1,ICOM)
      IF (JREP.EQ.2) CALL QCHECK(IGAIN,NSOPT,MSOPT)
      IF (JREP.EQ.3) CALL RICATI(IGAIN,IMODQ,NSOPT,MSOPT)
      IF (JREP.EQ.4) CALL QTRESP(IGAIN,NSOPT,MSOPT,IMODQ)
      IF (JREP.EQ.5) CALL QSTORE(IGAIN,IMODQ,NSOPT,MSOPT)
      IF (JREP.EQ.6) RETURN
      GO TO 160
      END
C
C Modify Q and/or R matrices
C
      SUBROUTINE MODQR(IMODQ,NSOPT,MSOPT,NORRDS,NOQRDS,ICOM)
      COMMON /QRKF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
C
      WRITE (5,10)
10    FORMAT (1X,'This option allows for individual elements of the
     1previously defined Q and R',/,1X,'matrices to be modified or for
     2 the C and R matrices to be redefined',/,1X,'Type M to modify Q
     3or R, I to redefine Q or R or E to Exit',/)
12    READ (5,15) ANS
C23456789
15    FORMAT (A1)
      IF (ANS.EQ.'M') GO TO 100
      IF (ANS.EQ.'I') GO TO 200
      IF (ANS.EQ.'E') GO TO 310
```

```
      WRITE (5,20)
20    FORMAT (1X,'Undefined character, try again')
      GO TO 12
100   WRITE (5,25)
25    FORMAT (1X,'Type either Q or R to modify elements of C or R')
      READ (5,15) ANS
      IF (ANS.NE.'Q'.AND.ANS.NE.'R') GO TO 100
105   WRITE (5,30)
30    FORMAT (1X,'Type row/column index of element to be modified,',/,
     11X,'Separated by a space, I2 format',/)
      READ (5,35) IR,IC
35    FORMAT (I2,1X,I2)
      WRITE (5,40)
40    FORMAT (1X,'New Value ?')
      READ (5,45) VNEW
45    FORMAT (F10.4)
      IF (ANS.EQ.'Q') GO TO 110
      IF (ANS.EQ.'R') GO TO 120
110   CALL MAP(NSOPT,IRQ,ICQ,IR,IC)
      WRITE (5,42) IR,IC,Q(IRQ,ICQ),VNEW
42    FORMAT (1X,'Element ',I2,1X,I2,'  Old value = ',F10.4,/,20X,
     1'New value = ',F10.4,/)
      Q(IRQ,ICQ)=VNEW
      GO TO 130
120   CALL MAP(NSOPT,IRR,ICR,IR,IC)
      WRITE (5,42) IR,IC,R(IRR,ICR),VNEW
      R(IRR,ICR)=VNEW
130   WRITE (5,50)
50    FORMAT (1X,'Type F to modify another element, E to modify an
     1 element of',/,1X,'another matrix, E to return',/)
      READ (5,15) ANS1
      IF (ANS1.NE.'F'.AND.ANS1.NE.'M'.AND.ANS1.NE.'E') GO TO 130
      IF (ANS1.EQ.'F') GO TO 105
      IF (ANS1.EQ.'M') GO TO 100
      GO TO 300
200   WRITE (5,60)
60    FORMAT (1X,'Redefine Q or R ?')
      READ (5,15) ANS
      IF (ANS.NE.'Q'.AND.ANS.NE.'R') GO TO 200
      IF (ANS.EQ.'J') GO TO 210
      NORRDS=1
      CALL RERDR(ICON,NORRDS,NOORDS)
      GO TO 300
210   NOCFDS=1
      CALL RERDC(ICON,NORRDS,NOORDS)
300   IMODG=1
310   CONTINUE
      RETURN
      END
C
C   * Check routines for optimal controller *
C
C23456789
      SUBROUTINE OCHECK(IGAIN,NSOPT,MSOPT)
      COMMON /ORKF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
      COMMON /OPTSYS/ A1(10,10),B1(10,10),C1(10,10)
5     WRITE (5,10)
10    FORMAT (1X,'* OPTIONS *',/,1X,'Type one of the following:-',
     1/,1X,'1- Display Q matrix',/,1X,'2- Display R matrix',/,1X,'
     23- Display current system matrices',/,1X,'4- Exit')
      IF (IGAIN.EQ.1) WRITE (5,15)
15    FORMAT (1X,'5- Display optimal gain matrices')
105   READ (5,20) JREP
20    FORMAT (I1)
      IF ((JREP.LT.1).OR.(IGAIN.EQ.1.AND.JREP.GT.5).OR.(IGAIN.EQ.0.AND.
     1JREP.GT.4)) GO TO 110
      GO TO 100
110   WRITE (5,25)
25    FORMAT (1X,'Unrecognised character, try again')
100   IF(JREP.EQ.1) CALL MWRITE(Q,NSOPT,NSOPT,20H Q WEIGHTING MATRIX ,5)
      IF(JREP.EQ.2) CALL MWRITE(R,MSOPT,MSOPT,20H R WEIGHTING MATRIX ,5)
      IF (JREP.EQ.3) GO TO 150
      IF (JREP.EQ.4) RETURN
      IF (JREP.EQ.5) GO TO 160
      GO TO 5
150   WRITE (5,30) NSOPT,NSOPT
30    FORMAT (1X,'System state matrix description',/,1X,' A matrix
     1 order = ',I2,/,1X,'System defined as ',I2,' input',/)
      CALL MWRITE (A1,NSOPT,NSOPT,20H   System A Matrix  ,5)
      CALL HOLD
      CALL MWRITE (B1,NSOPT,MSOPT,20H   System B Matrix  ,5)
      CALL MWRITE (C1,NSOPT,NSOPT,20H   System C Matrix  ,5)
      GO TO 5
160   CALL MWRITE (GKF1,MSOPT,NSOPT,20H State Gain Matrix  ,5)
      CALL HOLD
      CALL MWRITE (GKF2,MSOPT,NSOPT,20H Ref.  Gain Matrix  ,5)
      GO TO 5
      END
C
C   * Evaluate Ricatti matrix P *
C
      SUBROUTINE RICATI(IGAIN,IMODO,NSOPT,MSOPT)
      COMMON /ORKF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
      COMMON /OPTSYS/ A1(10,10),B1(10,10),C1(10,10)
      DIMENSION RI(10,10),W(10),BT(10,10),DUM(10,10),S(10,10),P(10,10)
     1,TEMP(10,10),AT(10,10),CC(10,10)
      DIMENSION AR(20,20),GR(20,10),GI(20,10),VR(20,20),VI(20,20)
      COMPLEX CA1(10,10),CA2(10,10),CA3(10,10),CA4(10,10)
      WRITE (5,10)
10    FORMAT (1X,'Routine evaluates and displays state and reference
     1gain matrices'/)
      IF (IMODO.EQ.0) GO TO 200
      IFAIL=0
      CALL F01AAF(R,MSOPT,MSOPT,RI,MSOPT,W,IFAIL)
      IF (IFAIL.NE.0) GO TO 300
      CALL MTRANS(BT,B1,NSOPT,MSOPT)
      CALL MWRITE(BT,MSOPT,NSOPT,20H....B Transpose.....,5)
      CALL HOLD
      CALL MMULT(RI,BT,DUM,MSOPT,MSOPT,NSOPT)
      CALL MWRITE(DUM,MSOPT,MSOPT,20H......RI * BT........,5)
      CALL HOLD
      CALL MMULT(B1,DUM,S,NSOPT,MSOPT,NSOPT)
      CALL MWRITE(S,NSOPT,MSOPT,20H....B * RI * BT......,5)
      CALL HOLD
      CALL RICATT(Q,S,A1,P,AR,GR,GI,VR,VI,CA1,CA2,CA3,CA4,MSOPT,NSOPT,
     12*MSOPT,$310)
      CALL MWRITE(P,NSOPT,NSOPT,20H   RICATTI MATRIX   ,5)
      CALL HOLD
      CALL MMULT(P,B1,TEMP,NSOPT,MSOPT,NSOPT)
      CALL MNEG(TEMP,NSOPT,NSOPT)
```

```
      CALL MMULT(TEMP,DUM,CC,NSOPT,NSOPT,NSOPT)
      CALL MTRANS(AT,A1,NSOPT,NSOPT)
      CALL MNULL(TEMP,10,10)
      CALL MADD(CC,AT,TEMP,NSOPT,NSOPT)
      CALL MWRITE(TEMP,NSOPT,NSOPT,20H    AT-P*B*RI*BT    ,5)
      CALL HOLD
      CALL F01AAF(TEMP,NSOPT,NSOPT,CC,NSOPT,W,IFAIL)
      IF (IFAIL.NE.0) GO TO 300
      CALL MNULL(S,10,10)
      CALL MMULT(DUM,CC,S,NSOPT,NSOPT,NSOPT)
      CALL MMULT(S,Q,GKF2,NSOPT,NSOPT,NSOPT)
      CALL MMULT(DUM,P,GKF1,NSOPT,NSOPT,NSOPT)
      CALL MNEG(GKF1,NSOPT,NSOPT)
      DO 100 I=1,NSOPT
      DO 100 J=1,NSOPT
      CALL MAP(NSOPT,IR,IC,I,J)
      IF (ABS(GKF1(IR,IC)).LT.1.0E-09) GKF1(IR,IC)=0.0
100   IF (ABS(GKF2(IR,IC)).LT.1.0E-09) GKF2(IR,IC)=0.0
200   CALL MWRITE(GKF1,NSOPT,NSOPT,20H OPTIMAL GAIN MATRIX,5)
      CALL HOLD
      CALL MWRITE(GKF2,NSOPT,NSOPT,20H OPTIMAL REF. MATRIX,5)
      IGAIN=1
      IMODC=0
      RETURN
300   WRITE (5,15) IFAIL
15    FORMAT (1X,'Error detected in NAG routine during Ricatti calculation
     1',/,1X,'Terminated with IFAIL = ',I5)
310   RETURN
      END
C
C  * Utility for Ricatti calculation *
C
      SUBROUTINE RICATT(Q,S,A,P,AR,GR,GI,VR,VI,A1,A2,A3,CX,N,M,NN,*)
      DIMENSION Q(N,N),S(N,N),A(N,N),P(N,N),RR(20),RI(20),WW(20)
      DIMENSION AR(NN,NN),GR(NN,N),GI(NN,N),VR(NN,NN),VI(NN,NN),IV1(20)
      COMPLEX A1(N,N),A2(N,N),A3(N,N),CX(N,N)
      NP1=N+1
      DO 1 I=1,N
      DO 1 J=1,N
      A3(I,J)=CMPLX(0.0,0.0)
      AR(I,J)=A(I,J)
      IF (I.EQ.J) A3(I,J)=CMPLX(1.0,0.0)
      JJ=J+N
      AR(I,JJ)=-S(I,J)
      II=I+N
      AR(II,J)=-Q(I,J)
1     AR(II,JJ)=-A(J,I)
      IFAIL=0
      CALL F02AGF(AR,NN,NN,RR,RI,VR,NN,VI,LN,IV1,IFAIL)
      IF (IERR.NE.0) GO TO 300
      WRITE (5,10) (RR(I),RI(I),I=1,NN)
10    FORMAT (10X,'THE EIGENVALUES OF THE STATE/COSTATE MATRIX ARE',//
     1,3X,'REAL PARTS',4X,'IMAGINARY PARTS',/,2X,'************',2X,
     2'****************',/,2(1X,F15.7,3X))
      IC=1
      DO 100 I=1,NN
      IF (RR(I).GT.0.0) GO TO 105
      DO 110 J=1,NN
      GI(J,IC)=VI(J,I)
110   GR(J,IC)=VR(J,I)
105   IF (RR(I).LT.0.0) IC=IC+1
100   CONTINUE
      IF (IC.EQ.NP1) GO TO 115
      WRITE (5,15)
15    FORMAT (1X,'SYSTEM IS ASYMPTOTICALLY UNSTABLE')
      RETURN
115   DO 120 I=1,N
      DO 120 J=1,N
      A1(I,J)=CMPLX(GR(I,J),GI(I,J))
      II=I+N
120   A2(I,J)=CMPLX(GR(II,J),GI(II,J))
      IFAIL=0
      CALL F04ADF(A1,N,A3,N,N,N,CX,N,WW,IFAIL)
      IF (IFAIL.NE.0) GO TO 300
      DO 130 I=1,N
      DO 130 J=1,N
      P(I,J)=0.0
      DO 130 II=1,N
      P(I,J)=P(I,J)+REAL(A2(I,II))*REAL(CX(II,J))
130   P(I,J)=P(I,J)-AIMAG(A2(I,II))*AIMAG(CX(II,J))
      DO 135 I=1,N
      DO 135 J=1,N
      S(I,J)=(P(I,J)+P(J,I))/2.
135   CONTINUE
      DO 136 I=1,N
      DO 136 J=1,N
      P(I,J)=S(I,J)
136   CONTINUE
      RETURN
300   WRITE (5,20) IFAIL
20    FORMAT (1X,'ERROR DETECTED IN NAG ROUTINE DURING RICATT',/,1X,
     1'TERMINATED WITH IFAIL = ',I5)
      RETURN 1
      END
C
C   * Time Response Routines *
C
      SUBROUTINE OTRESP(IGAIN,NSOPT,MSOPT,IMODO)
      COMMON /OPTSYS/ A1(10,10),B1(10,10),C1(10,10)
      COMMON /QRKF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
      COMMON /VECTS/ REF(10),XO(10)
      IF (IGAIN.EQ.1.AND.IMODO.EQ.0) GO TO 5
      WRITE (5,10)
10    FORMAT (1X,'Evaluate optimal gain matrices before this option')
      RETURN
5     WRITE (5,15)
15    FORMAT (1X,'This option allows the closed loop time response of
     1 the optimal controller',/,1X,'to be evaluated.  State initial
     2 conditions and reference vector may',/,1X,'be input.  Graphical
     3 output of state or output responses',/,1X,'is available',/)
      NFL1=0
      NFL2=0
      NFL3=0
80    WRITE (5,20)
20    FORMAT (1X,'* OPTIONS *',/,1X,'Type one of the following:-',
     1/,1X,'1-  Define time step size and number',/,1X,'2-  Specify
     2 reference vector or initial conditions',/,1X,'3-  Specify
     3 type of output required',/,1X,'4-  Evaluate and plot response
     4 (needs 1,2 & 3)',/,1X,'5-  Exit',/)
90    READ (5,25) JANS
```

```
25    FORMAT (I1)
      IF (JANS.LE.5.AND.JANS.GE.1) GO TO 100
      WRITE (5,30)
30    FORMAT (1X,'Unrecognised character, try again')
      GO TO 90
100   IF (JANS.EQ.1) CALL STEP1(T,INT1,NFL1)
      IF (JANS.EQ.2) CALL INPUT1(NSOPT,NSOPT,NFL2)
      IF (JANS.EQ.3) CALL OUTPT1(NSOPT,NSOPT,KYFLG,LSTAT,NFL3)
      IF (JANS.EQ.4) CALL EXEC1(IGAIN,IMODG,NSOPT,NSOPT,NFL1,NFL2,
     1NFL3,T,INT1,KYFLG,LSTAT)
      IF (JANS.EQ.5) RETURN
      GO TO 80
      END
C
C  * Define type of input required *
C
      SUBROUTINE INPUT1(N,M,NFL2)
      COMMON /VECTS/ REF(10),XO(10)
      WRITE (5,10) N
10    FORMAT (1X,'System currently defined as',1X,I2,1X,'state')
      WRITE (5,15)
15    FORMAT (1X,'Do you wish to define an initial condition state
     1 vector (Y/N) ?',/,1X,'N defaults to I.C. = 0')
90    READ (5,20) ANS
20    FORMAT (A1)
      IF (ANS.EQ.'Y') GO TO 100
      IF (ANS.EQ.'N') GO TO 110
      WRITE (5,25)
25    FORMAT (1X,'Unrecognised character, try again')
      GO TO 90
100   WRITE (5,30)
30    FORMAT (1X,'Input the initial state vector, under MREAD')
      CALL MREAD (XO,N,1,5)
      GO TO 120
110   DO 115 I=1,N
115   XO(I)=0.0
120   WRITE (5,35)
35    FORMAT (1X,'Do you wish to define a reference vector (Y/N) ?',
     1/,1X,'N defaults to REF = 0')
      READ (5,20) ANS
      IF (ANS.EQ.'Y') GO TO 200
      IF (ANS.EQ.'N') GO TO 210
      WRITE (5,25)
      GO TO 120
200   WRITE (5,40)
40    FORMAT (1X,'Input the reference vector, under MREAD')
      CALL MREAD (REF,N,1,5)
      GO TO 220
210   DO 215 I=1,N
215   REF(I)=0.0
220   NFL2=1
      RETURN
      END
C
C  * Define the outputs to be displayed *
C
      SUBROUTINE OUTPT1(N,M,KYFLG,LSTAT,NFL3)
      COMMON /INOUT/ ISTAT(10),IOPT(10)
5     WRITE (5,10)
10    FORMAT (1X,'Do you wish to display state or output responses
     1 (S/O) ?')
      READ (5,15) ANS
15    FORMAT (A1)
      IF (ANS.EQ.'S') GO TO 100
      IF (ANS.EQ.'O') GO TO 110
      WRITE (5,20)
20    FORMAT (1X,'Unrecognised character, try again')
      GO TO 5
100   LSTAT=1
      IF (N.LE.1) GO TO 135
      WRITE (5,25)
25    FORMAT (1X,'Responses may be displayed cyclically, with a pause
     1 between each display.',/,1X,'You may also define specific
     2 channels for display on the same diagram',/,1X,'Type C or D
     3respectively for these options')
90    READ (5,15) ANS
      IF (ANS.EQ.'C') GO TO 120
      IF (ANS.EQ.'D') GO TO 130
      WRITE (5,20)
      GO TO 90
120   KYFLG=0
      GO TO 300
130   WRITE (5,40) N
40    FORMAT (1X,'Input a ',I2,' vector defining channels to be
     1Displayed, under MREAD',/,1X,'A 1 in any position of this vector
     2 displays the corresponding',/,1X,'response.',/,1X,'A 0 supresses
     3 the display')
      CALL MREAD (ISTAT,N,1,5)
      KYFLG=1
      GO TO 300
135   KYFLG=0
      GO TO 300
110   LSTAT=0
      IF (M.LE.1) GO TO 235
      WRITE (5,25)
80    READ (5,15) ANS
      IF (ANS.EQ.'C') GO TO 220
      IF (ANS.EQ.'D') GO TO 230
      WRITE (5,20)
      GO TO 80
220   KYFLG=0
      GO TO 300
230   WRITE (5,40) M
      CALL MREAD (IOPT,M,1,5)
      KYFLG=1
      GO TO 300
235   KYFLG=0
300   NFL3=1
      RETURN
      END
C
C   * Utility routines for time response plotting *
C
      SUBROUTINE CYCLE(OPSO,N,T,INT1)
      DIMENSION OPSO(N,INT1)
      DO 100 J=1,N
      YMAX1=-100.0
      YMIN1=100.0
      YMAX=OPSO(J,1)
      YMIN=OPSO(J,1)
```

```
      YMAX1=AMAX1(YMAX1,YMAX)
      YMIN1=AMIN1(YMIN1,YMIN)
      DO 110 L=2,INT1
      YMAX1=AMAX1(YMAX1,OPSO(J,L))
110   YMIN1=AMIN1(YMIN1,OPSO(J,L))
      IF (YMAX1.EQ.0.0.AND.YMIN1.EQ.0.0) GO TO 115
      GO TO 120
115   YMAX1=1.0
      YMIN1=0.0
120   CALL PLTXES(YMAX1,YMIN1,INT1,T,ICX,ICY,YSCL,TSCL)
C
C   * PLOT DATA *
C
      CALL IPLOT(0,ICX,ICY)
      DO 130 I=1,INT1
      ITSC=ICX+INT((I-1)*T/TSCL)
      IYSC=INT(OPSO(J,I)/YSCL)+ICY
130   CALL IPLOT(1,ITSO,IYSO)
      CALL CHAINT(J,2)
      CALL IPLOT(0,0,750)
      CALL CHAMOD
      CALL HOLD
      CALL HOLD
100   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SEQ(OPSO,N,T,INT1,OPTV)
      DIMENSION OPSO(N,INT1),OPTV(N),ITSO(300)
      YMAX1=-100.0
      YMIN1=100.0
      NO=0
      DO 120 I=1,N
      IF (OPTV(I).NE.1) GO TO 105
      YMAX=OPSO(I,1)
      YMIN=OPSO(I,1)
      YMAX1=AMAX1(YMAX1,YMAX)
      YMIN1=AMIN1(YMIN1,YMIN)
      DO 110 L=2,INT1
      YMAX1=AMAX1(YMAX1,OPSO(I,L))
110   YMIN1=AMIN1(YMIN1,OPSO(I,L))
      GO TO 100
105   NO=NO+1
100   CONTINUE
      IF (YMAX1.EQ.0.0.AND.YMIN1.EQ.0.0.OR.NO.EQ.N) GO TO 115
      GO TO 120
115   YMAX1=1.0
      YMIN1=0.0
120   CALL PLTXES(YMAX1,YMIN1,INT1,T,ICX,ICY,YSCL,TSCL)
C
C   * PLOT DATA
C
      DO 130 I=1,INT1
130   ITSC(I)=ICX+INT((I-1)*T/TSCL)
      DO 110 I=1,N
      IF (OPTV(I).NE.1) GO TO 140
      CALL IPLOT(0,ICX,ICY)
      DO 150 L=1,INT1
      IYSC=INT(OPSO(I,L)/YSCL)+ICY
150   CALL IPLOT (1,ITSO(L),IYSO)
      CALL CHAINT (1,2)
140   CONTINUE
      CALL IPLOT(0,0,50)
      CALL CHAMOD
      CALL HOLD
      CALL HOLD
      RETURN
      END
C
C   * Store or retrieve data from disk *
C
      SUBROUTINE OSTORE(IGAIN,INODO,N,M)
      COMMON /OPTSYS/ A1(10,10),B1(10,10),C1(10,10)
      COMMON /GRKF/ Q(10,10),R(10,10),GKF1(10,10),GKF2(10,10)
5     WRITE (5,15)
15    FORMAT (1X,'This option allows system, weighting and gain matrices
     1 to be stored',/,1X,'or Retrieved from disk.  Type S to store, R
     2 to retrieve.',/,1X,'Note: to retrieve data it must previously
     3 have been stored using this option')
90    READ (5,20) ANS
20    FORMAT (A1)
      IF (ANS.EQ.'S') GO TO 100
      IF (ANS.EQ.'R') GO TO 200
      IF (ANS.EQ.'E') RETURN
      WRITE (5,25)
25    FORMAT (1X,'Unrecognised character, try again')
      GO TO 90
100   IF (IGAIN.NE.0) GO TO 105
      WRITE (5,10)
10    FORMAT (1X,'Gain matrices not defined, use option 3')
      RETURN
105   WRITE (5,30)
30    FORMAT (1X,'Enter a 5 character filename')
      READ (5,35) FNAM
35    FORMAT (A5)
      OPEN (UNIT=1,DEVICE='DSK',ACCESS='SEQOUT',MODE='ASCII',FILE=FNAM)
      WRITE (1,40) N,M
      DO 110 I=1,N
      DO 110 J=1,N
      CALL MAP(N,IR,IC,I,J)
      WRITE (1,50) A1(IR,IC)
110   WRITE (1,50) Q(IR,IC)
      DO 115 I=1,N
      DO 115 J=1,M
      CALL MAP(N,IR,IC,I,J)
115   WRITE (1,50) B1(IR,IC)
      DO 120 I=1,N
      DO 120 J=1,N
      CALL MAP(N,IR,IC,I,J)
120   WRITE (1,50) C1(IR,IC)
      DO 125 I=1,N
      DO 125 J=1,M
      CALL MAP(N,IR,IC,I,J)
125   WRITE (1,50) R(IR,IC)
      DO 135 I=1,M
      DO 135 J=1,N
      CALL MAP(N,IR,IC,I,J)
      WRITE (1,50) GKF1(IR,IC)
135   WRITE (1,50) GKF2(IR,IC)
```

```
      CLOSE (UNIT=1,DEVICE='DSK',FILE=FNAM)
      RETURN
200   WRITE (5,30)
      READ (5,35) FNAM
      OPEN (UNIT=1,DEVICE='DSK',ACCESS='SEQIN',MODE='ASCII',FILE=FNAM)
      READ (1,40) N,M
      DO 210 I=1,N
      DO 210 J=1,N
      CALL MAP(N,IR,IC,I,J)
      READ (1,50) A1(IR,IC)
210   READ (1,50) Q(IR,IC)
      DO 215 I=1,N
      DO 215 J=1,M
      CALL MAP(N,IR,IC,I,J)
215   READ (1,50) B1(IR,IC)
      DO 220 I=1,N
      DO 220 J=1,N
      CALL MAP(M,IR,IC,I,J)
220   READ (1,50) C1(IR,IC)
      DO 230 I=1,M
      DO 230 J=1,M
      CALL MAP(M,IR,IC,I,J)
230   READ (1,50) R(IR,IC)
      DO 235 I=1,N
      DO 235 J=1,N
      CALL MAP(N,IR,IC,I,J)
      READ (1,50) GKF1(IR,IC)
235   READ (1,50) GKF2(IR,IC)
      CLOSE (UNIT=1,DEVICE='DSK',FILE=FNAM)
      WRITE (5,60)
60    FORMAT (1X,'System retrieved from disk')
50    FORMAT (F12.6)
40    FORMAT (I2)
      IGAIN=1
      IMODC=0
      RETURN
      END
```

```
C
C PHIDEL ROUTINE FOR MASTER, ETC.
C
C23456789
      SUBROUTINE PHIDEL(A,B,AA,BB,CC,PHI,DELTA,N,M,T,CONVAC)
      DIMENSION A(N,N),B(N,M),AA(N,N),BB(N,N),CC(N,N),
     1PHI(N,N),DELTA(N,N)
      DO 10 I=1,N
      DO 10 J=1,N
      PHI(I,J)=0.0
      IF (I.EQ.J) PHI(I,J)=1.0
10    BB(I,J)=PHI(I,J)*T
      L=0
40    L=L+1
      LCNT=0
      IFAIL=0
      IF (L.EQ.1) GO TO 50
      CALL F01CKF(CC,AA,A,N,N,N,Z1,1,1,IFAIL)
      GO TO 60
50    CALL F01CMF(A,N,CC,N,N,N)
60    DO 30 I=1,N
      DO 30 J=1,N
      AA(I,J)=CC(I,J)*T/L
      IF (ABS(AA(I,J)).LE.CONVAC) LCNT=LCNT+1
      PHI(I,J)=PHI(I,J)+AA(I,J)
30    BB(I,J)=BB(I,J)+AA(I,J)*T/(L+1)
      IF (LCNT.LT.N**2) GO TO 40
      IFAIL=0
      CALL F01CKF(DELTA,BB,B,N,M,N,Z1,1,1,IFAIL)
      RETURN
      END
```

```
C
C     Prog. to evaluate dryden noise model (1 motion)
C     links to optimal control routines
C
C23456789
      SUBROUTINE SNOISE(GAMMA,FSQ,NSEQ,N,INT1,T)
      DIMENSION R(10,10),VADD(10,300),A(2,2),B(2,1),AA(2,2),BB(2,2),
     1CC(2,2),PHI(2,2),DELTA(2,2),XOLD(2),XNEW(2),XSQ(2,300),DUM2(2),
     2DUM3(2),GAMMA(N,NSEQ),FSQ(NSEQ,INT1)
      REAL LAMDA
25    FORMAT (F12.4)
      WRITE (5,30)
30    FORMAT (1X,'Input gust length, in meters')
      READ (5,25) LAMDA
      WRITE (5,35)
35    FORMAT (1X,'Input air speed, in meters per sec')
      READ (5,25) AIR
      X=LAMDA/AIR
      X1=AIR*SQRT(LAMDA/3.14159)
      RHO=1.
      CALL MNOISE(INT1,NSEQ,T,RHO,R,VADD)
      A(1,1)=0.0
      A(1,2)=1.0
      A(2,1)=-1.0/(X*X)
      A(2,2)=-2.0/X
      B(1,1)=X1*SQRT(3.)/X
      B(2,1)=X1*(1.0-2.0*SQRT(3.))/(X*X)
      CONVAC=1.0E-07
      CALL PHIDEL(A,B,AA,BB,CC,PHI,DELTA,2,1,T,CONVAC)
      DO 300 J=1,NSEQ
      DO 102 I=1,2
100   XOLD(I)=0.0
      DO 120 L=1,INT1
      U=VADD(J,L)
      CALL MSCMUL(DELTA,U,DUM3,2,1)
      CALL MMULT(PHI,XOLD,DUM2,2,2,1)
      CALL MADD(DUM2,DUM3,XNEW,2,1)
      DO 130 I=1,2
      XSQ(I,L)=XNEW(I)
130   XOLD(I)=XNEW(I)
120   CONTINUE
      DO 300 L=1,INT1
300   FSQ(J,L)=XSQ(1,L)/1000.
      WRITE (5,40)N,NSEQ
40    FORMAT (1X,'Input a ',I2,1X,I2,' matrix defining the noise
     1 coupling, under MREAD')
      CALL MREAD(GAMMA,N,NSEQ,5)
      RETURN
      END
C
C
      SUBROUTINE MNOISE(N,MM,M,RHO,R,VADD)
      DIMENSION R(10,10),VADD(10,300)
      DIMENSION VEC(10),VR(300),VRR(300)
      INTEGER TEMP,TEMP1
      KT=10
      IT=429
      DO 50 II=1,MM
      SC=RHO
      VEC(II)=IT+KT*II
      DO 60 I=1,M
      IF (I.EQ.1) IX=VEC(II)
      CALL RANDOM(YFL,I,IX,M,FM,TEMP,TEMP1)
      VR(I)=YFL
      IF (I.EQ.1) IX2=VEC(II)+10
      CALL RANDOM(YFLB,I,IX2,M,FM,TEMP,TEMP1)
      VRR(I)=YFLB
60    CONTINUE
      SD=SQRT(SC)
      CALL GAUSS(SD,0.0,VR,M,VRR)
      DO 62 I=1,M
      VADD(II,I)=VR(I)
62    CONTINUE
61    R(II,II)=SC/M
50    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE RANDOM(Z,I,IX,M,FM,X,A)
      INTEGER A,X
      IF (I.NE.1) GO TO 1
      M=2**20
      FM=M
      X=IX
      A=2**10+3
1     X=MOD(A*X,M)
      FX=X
      Z=ABS(FX/FM)
      RETURN
      END
C
C
      SUBROUTINE GAUSS(SDEV,TM,RA,N,RB)
      DIMENSION RA(N),RB(N)
      DO 1 I=1,N
      V1=(-2.0*ALOG(RA(I)))**0.5*COS(6.28318507*RB(I))
      RA(I)=V1
1     CONTINUE
      ZM=0.0
      DO 2 I=1,N
      ZM=ZM+RA(I)
2     CONTINUE
      ZMEAN=ZM/N
      DO 3 I=1,N
      RA(I)=RA(I)-(ZMEAN-TM)
3     CONTINUE
      DO 4 I=1,N
      RA(I)=RA(I)*SDEV
4     CONTINUE
      RETURN
      END
```