

AD-A124 871

HIGH-GAIN ERROR ACTUATED FLIGHT CONTROL SYSTEMS FOR
CONTINUOUS LINEAR MUL. (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI. T LEWIS

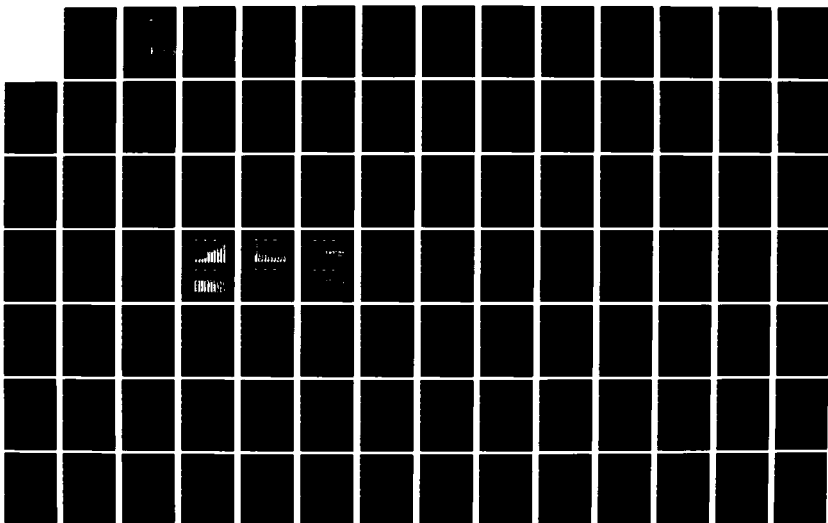
1/2

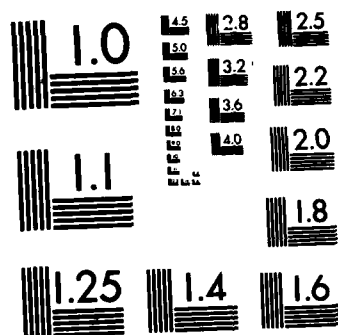
UNCLASSIFIED

DEC 82 AFIT/GAE/EE/82D-1

F/G 1/3

NL

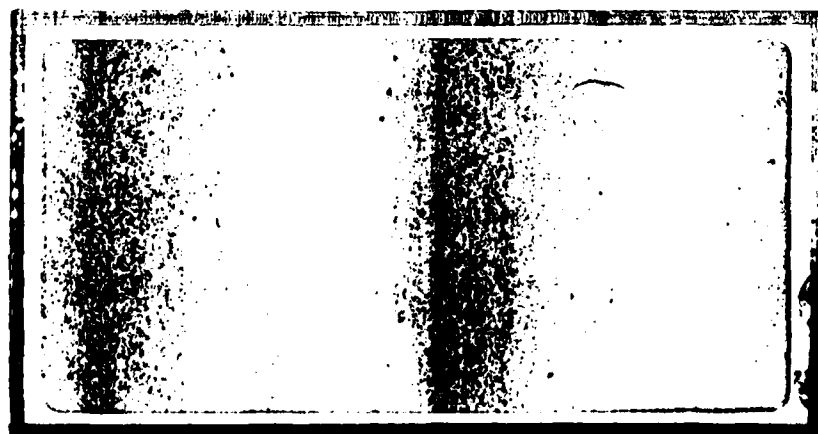




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A 124 871

DTIC FILE COPY



DTIC
ELECTE
FEB 24 1983

This document has been approved
for public release and sale; its
distribution is unlimited.

*Original contains color
plates: All DTIC reproductions
will be in black and
white*

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

83 02 024 037

AFIT/GAE/EE/82D-1

HIGH-GAIN ERROR ACTUATED FLIGHT
CONTROL SYSTEMS FOR CONTINUOUS LINEAR
MULTIVARIABLE PLANTS

THESIS

AFIT/GAE/EE/82D-1

Thomas Lewis
2Lt USAF

Approved for public release; distribution unlimited

"Original contains color
plates: All DTIC reproduct-
ions will be in black and
white"

HIGH-GAIN ERROR ACTUATED FLIGHT
CONTROL SYSTEMS FOR CONTINUOUS LINEAR
MULTIVARIABLE PLANTS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Thomas Lewis, B.S.A+AE.

2Lt

USAF

Graduate Aeronautical Engineering

December 1982

Approved for public release; distribution unlimited

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
Distribution/	
Availability Codes	
Dist	Avail and/or
A	Special



Acknowledgements

I would like to thank my thesis advisors, Dr. John D'Azzo and Dr. Constantine Houpis, for all their encouragement and guidance. I would also like to thank the Flight Dynamics Laboratory for sponsoring my thesis, especially the Control Techniques Group and Control Data Group for sharing their facilities with me.

Dedication

I dedicate this thesis to my grandfather William Lewis, a self taught engineer, who was an inspiration throughout my engineering education.

Contents

	Page
Preface	ii
Illustrations	v
Symbols	vii
Abstract	x
I. Introduction	1
Background	2
Problem Statement and Approach	5
Assumptions	6
Organization	7
II. Theoretical Development	8
III. Discussion of Results	22
Design Procedure	22
Robustness Tests	24
Actuator Dynamics	42
IV. Conclusions and Recommendations	56
Bibliography	58
Appendix A-1 Development of the Aircraft Equations of Motion .	60
Appendix A-2 Dimensionalization Computer Program	72
Appendix B-1 Computer Program to Calculate System Zeros	85
Appendix B-2 Listing of Program ZERO and Examples	90
Appendix C Main Computer Program	114
Vita	137

Illustrations

Figures	Page
2-1 - Tracking System	21
2-2 - Tracking System with Actuators	21
3-1 - Figures of Merit for Output Responses	28
3-2 - Comparison of Figures of Merit	
(a)- Peak Overshoot to a Step r Input	32
(b)- Steady-state Pointing Angle ψ	32
(c)- Absolute Time Integral ϕ	33
(d)- Maximum Control Surface Deflection Error	33
(e)- Velocity Vector Pointing Angle Range	34
(f)- Steady-state Velocity Vector Pointing Angle	34
3-3 - Low Q Yaw Pointing	35
3-4 - Mid Q Yaw Pointing	36
3-5 - High Q Yaw Pointing	37
3-6 - Design 2A Additional Maneuvers	39
3-7 - Mid Q Additional Maneuvers	40
3-8 - Low Q Yaw Pointing Without Actuators	45
3-9 - Mid Q Yaw Pointing Without Actuators	46
3-10 - High Q Yaw Pointing Without Actuators	47
3-11 - Design 2 Additional Maneuvers	48
3-12 - All Constants Changed	49
3-13 - One Constant Decreased	50
3-14 - One Constant Increased	51
3-15 - Two Constants Decreased	52
3-16 - Two Constants Increased	53
3-17 - One Constant Decreased, One Constant Increased	54
B-1 - Program ZERO Flow Chart	89
C-1 - Main Computer Program Flow Chart	115

Tables

3-1 - Input Commands	25
3-2 - Design 1A Figures of Merit	29
3-3 - Design 2A Figures of Merit	30
3-4 - Design 3A Figures of Merit	31

Illustrations

Tables	Page
3-5 - Mid Q Designs	41
A-1 - Flight Condition Data	63
A-2 - AFTI-16 Non-Dimensional Stability Axis Data	66
A-3 - Lateral-Directional Dimensionalization Factors	68
A-4 - AFTI-16 Fixed Aircraft Parameters	68
A-5 - AFTI-16 Plant Matrices	69
B-1 - Required Input for Program ZERO	88

Symbols

$\underline{A}$	- State Matrix
$\underline{A_c}$	- Actuator Contant Matrix
$\underline{A_{CL}}$	- Closed-loop State Matrix
$\underline{AA}$	- Generalized Eigenvalue Problem Matrix
b	- Wing Span
$\underline{B}$	- Input Matrix
$\underline{B_{CL}}$	- Closed-loop Input Matrix
$\underline{BB}$	- Generalized Eigenvalue Problem Matrix
$\underline{c_i}$	- i^{th} Row of Output Matrix
C	- Set of Complex Numbers
$\underline{C}$	- Output Matrix
$\underline{C_{CL}}$	- Closed-loop Output Matrix
C_l	- Roll Moment Coefficient
C_n	- Yaw Moment Coefficient
C_y	- Side Force Coefficient
$\underline{d}$	- Riccati Equation Vector Due to Tracking
dt	- Time Derivative
$\underline{e}$	- Error Vector
$\underline{F}$	- Measurement Output Matrix
g	- Acceleration Due to Gravity
	- Forward-loop Gain Parameter
$\underline{G}$	- Transfer Function
$\underline{G_A}$	- Transfer Function for Actuator Dynamics
$\underline{H}$	- Weighting Matrix
$\underline{I_i}$	- Identity Matrix of Rank i
$\underline{I_{xx}}$	- Moment of Inertia x body axis
$\underline{I_{xz}}$	- Cross Product of Inertia x and z body axis
$\underline{I_{zz}}$	- Moment of Inertia z body axis
J	- Performance Index
$\underline{K}$	- Controller Matrix
	- Feedback Gain Matrix
	- Random Matrix
l	- Number of Outputs
L	- Roll Moment
L'	- Roll Moment with Inertia Terms

Symbols

m	- Number of Inputs
$\underline{M}$	- Measurement Matrix
n	- Number of States
N	- Yaw Moment
N'	- Yaw Moment with Inertia Terms
p	- Roll Rate
$\underline{P}$	- Riccati Equation Solution Matrix
Q	- Dynamic Pressure
$\underline{Q}$	- Weighting Matrix
r	- Yaw Rate
R	- Set of Real Numbers
$\underline{R}$	- Weighting Matrix
s	- Laplace Transformation Parameter
S	- Wing Area
t	- Time
t_f	- Final Time
$\underline{u}$	- Input Vector
$\underline{u}_A$	- Input Vector with Actuator Dynamics
$\underline{u}^*$	- Optimal Input Vector
U	- Aircraft Velocity
v	- Perturbation Side Velocity
$\underline{v}$	- Command Vector
w	- Weight
$\underline{w}$	- Measurement Vector
$\underline{x}$	- State Vector
$\underline{y}$	- Output Vector
Y	- Side Force with v as a State
Y^*	- Side Force with β as a State
$\underline{z}$	- Time Integral of Error Vector
	- Generalized Eigenvector
z_1	- First Set of Slow Eigenvalues
z_2	- Second Set of Slow Eigenvalues
z_3	- Set of Fast Eigenvalues

Symbols

α_o	- Trim Angle of Attack
β	- Side-Slip Angle
γ	- Flight Path Angle
$\bar{\Gamma}$	- Asymptotic Transfer Function
$\bar{\Gamma}_s$	- Slow Asymptotic Transfer Function
$\bar{\Gamma}_f$	- Fast Asymptotic Transfer Function
ϕ	- Roll Angle
δ_a	- Aileron Deflection
δ_c	- Cannard Deflection
δ_{dt}	- Differential Tail Deflection
δ_f	- Flaperon Deflection
δ_r	- Rudder Deflection
ϵ	- Perturbation Parameter
	- " is a member of "
$\underline{\xi}$	- Eigenvector
θ_o	- Trim Flight Path Angle
λ	- Eigenvalue
ρ	- Air Density
σ	- Fast Pole
$\underline{\Sigma}$	- Matrix of Fast Poles
τ	- Time Dummy Argument
$\underline{\omega}$	- Portion of Null Space
ψ	- Pointing Angle
$\dot{}$	- Time Derivative of a Vector
N	- Null Space
$\det$	- Determinant of a Matrix
$ $	- Determinant of a Matrix
$[]^T$	- Transpose of a Matrix
$[]^{-1}$	- Inverse of a Matrix

Subscripts

b	- Body Axis
s	- Stability Axis
ss	- Steady-State

Abstract

The theory of high-gain error actuated feedback control, developed by Porter and Bradshaw, is applied to the design of various lateral-directional decoupling flight control systems for an advanced aircraft. The controllers developed in this report utilize output feedback with proportional plus integral control to produce desirable closed-loop responses with minimal interactions between outputs. Because of the structure of the system, measurement variables in addition to the outputs are necessary to apply this method. The report examines controller design robustness by varying the flight conditions or maneuver commands from the ones the controller is specifically designed for, and then judges system performance. The results show that the controller is robust with respect to varying flight conditions, but is not robust with respect to varying maneuver commands. This report also examines the effect of first-order actuator dynamics in the system model. Actuator dynamics are shown to significantly affect the control system response, indicating that a simplified model, without actuators, is not desirable in one's control design scheme. Also a computer program to determine transmission zeros and decoupling zeros is developed.

HIGH-GAIN ERROR ACTUATED FLIGHT
CONTROL SYSTEMS FOR CONTINUOUS LINEAR
MULTIVARIABLE PLANTS

I. Introduction

Advanced Fighter Technology Integration (AFTI) is the culmination of the most recent technology advances in flight control and avionics. The AFTI-16 aircraft, which incorporates this technology, is presently undergoing flight tests. A new digital flight control system, along with additional control surfaces, such as vertical canards, allow the aircraft to perform direct force, decoupled maneuvers.

This report contains an investigation of a new controller design method for the AFTI-16. Specifically, an analogue controller is designed to perform four lateral-directional decoupled maneuvers. These maneuvers are horizontal translation, wings level turn, constant altitude coordinated turn, and yaw pointing. The maneuvers can be described by three output variables: side-slip angle, β ; roll angle, ϕ ; and yaw rate, r . Initially all output variables are zero. For horizontal translation the side-slip angle, β , changes, while ϕ and r remain unchanged. In a wings level turn the yaw rate, r , changes, while β and ϕ remain unchanged. In a constant altitude coordinated turn the yaw rate, r , is proportional to the roll angle, ϕ ,

$$r = \frac{g}{U} \sin \phi, \quad (1-1)$$

while β remains zero. For yaw pointing the time rate of change of the side-slip angle, β , must be equal in magnitude and opposite in sign to

the yaw rate, r , while ϕ remains zero.

Background

Controllers have been designed for single input/single output (SISO) systems using conventional techniques for a long time. The advance of computational technology readily permits the design of controllers for multiple input/multiple output (MIMO) systems. A MIMO system design offers a greater amount of control and greater precision. Conventional control employs widely accepted design techniques, such as root-locus and frequency response analysis. Multivariable control design is still growing and a number of different theories and design methods have been developed. In this report the design method developed and proposed by Dr Brian Porter of the University of Salford, England is utilized and evaluated.

Other multivariable control methods presently available utilize optimal control, entire eigenstructure assignment, and frequency domain techniques.

In the optimal control design method (Ref.2) a control law is found by minimizing the value of the performance index or cost functional

$$\begin{aligned} J = & 1/2 \int_0^{t_f} \left[\left[\underline{y}(t) - \underline{v}(t) \right]^T \underline{Q}(t) \left[\underline{y}(t) - \underline{v}(t) \right] \right. \\ & \left. + \underline{u}^T(t) \underline{R}(t) \underline{u}(t) \right] dt \\ & + 1/2 \left[\underline{y}(t_f) - \underline{v}(t_f) \right]^T \underline{H} \left[\underline{y}(t_f) - \underline{v}(t_f) \right] \end{aligned} \quad (1-2)$$

for the system described by the state and output equations

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A} \underline{x}(t) + \underline{B} \underline{u}(t) \\ \underline{y}(t) &= \underline{C} \underline{x}(t) \end{aligned} \quad (1-3)$$

where $\underline{y}(t)$ is the output vector.

$\underline{v}(t)$ is the command vector, and $\underline{Q}(t)$, $\underline{R}(t)$, and $\underline{H}$ are symmetric positive definite quadratic weighting matrices. The optimal control law that minimizes J is

$$\underline{u}^*(t) = \underline{K}(t) \underline{x}(t) - \underline{R}^{-1}(t) \underline{B}^T \underline{d}(t) \quad (1-4)$$

The time varying feedback gain matrix is

$$\underline{K}(t) = - \underline{R}^{-1}(t) \underline{B}^T \underline{P}(t) \quad (1-5)$$

where $\underline{P}(t)$ is the solution of a nonlinear time varying matrix differential equation, commonly known as the Riccati Equation:

$$\begin{aligned} \dot{\underline{P}}(t) = & -\underline{A}^T \underline{P}(t) - \underline{P}(t) \underline{A} + \underline{P}(t) \underline{B} \underline{R}^{-1}(t) \underline{B}^T \underline{P}(t) \\ & - \underline{C}^T \underline{Q}(t) \underline{C} \end{aligned} \quad (1-6)$$

where $\underline{P}(t_f) = \underline{C}^T \underline{H} \underline{C}$.

The vector $\underline{d}(t)$ is the solution to the matrix differential equation

$$\begin{aligned} \dot{\underline{d}}(t) = & \underline{A}^T \underline{d}(t) + \underline{K}(t) \underline{B} \underline{R}^{-1}(t) \underline{B}^T \underline{d}(t) \\ & + \underline{C}^T \underline{Q}(t) \underline{C} \end{aligned} \quad (1-7)$$

where $\underline{d}(t_f) = -\underline{C}^T \underline{H} \underline{C} \underline{v}(t_f)$.

Due to the boundary conditions on the two matrix differential equations, they must be integrated backwards in time. This requires that command input information be known in the future. The computation required for the numerical solutions to these equations are extremely costly

and difficult to achieve. Also, selection of the positive definite weighting matrices, $\underline{Q}(t)$, $\underline{R}(t)$, and $\underline{H}$, must be made. This is usually accomplished by a trial and error iterative process, requiring repeated integrations of the matrix differential equations.

In the entire eigenstructure assignment design method (Ref 11), the designer selects the system closed-loop eigenvalues, λ_i , and corresponding eigenvectors, $\underline{\xi}_i$. Given the system described by eq. 1.3, a state feedback control law

$$\underline{u}(t) = \underline{K} \underline{x}(t) + \underline{v}(t) \quad (1-8)$$

is assumed. Substituting this control law into the state equations yields the closed-loop system

$$\dot{\underline{x}}(t) = [\underline{A} + \underline{B} \underline{K}] \underline{x}(t) + \underline{B} \underline{v}(t) \quad (1-9)$$

The feedback gain matrix, $\underline{K}$, is

$$\underline{K} = [\underline{\omega}_1, \underline{\omega}_2, \dots, \underline{\omega}_n] [\underline{\xi}_1, \underline{\xi}_2, \dots, \underline{\xi}_n]^{-1} \quad (1-10)$$

The feedback gain matrix of eq. 1-10 produces a closed-loop system with the selected eigenvalues and corresponding eigenvectors, provided all eigenvectors, $\underline{\xi}_i$, and vectors, $\underline{\omega}_i$, satisfy the relation

$$\begin{bmatrix} \underline{\xi}_i \\ \underline{\omega}_i \end{bmatrix} \in N[\underline{A} - \lambda_i \underline{I}_n, \underline{B}] \quad (1-11)$$

Extreme care must be taken when selecting the eigenvectors from the null space of $[\underline{A} - \lambda_i \underline{I}, \underline{B}]$, in order that the matrix $[\underline{\xi}_1, \underline{\xi}_2, \dots, \underline{\xi}_n]^{-1}$

exists. Also, the eigenvalues and eigenvectors must be chosen to minimize the amount of interaction between outputs. Selection of the eigenvalues, λ_1 , and eigenvectors, $\underline{\xi}_1$, is an art, requiring insight and a lot of trial and error iterations.

Porter's design method, described in Chapter II, often produces a successful controller design with less trial and error iterations than the two methods described above.

Problem Statement and Approach

It is desired to design and evaluate a flight controller, using Porter's design method (Ref 7), for an AFTI-16 aircraft which simulates maneuvers of horizontal translation, wings level turn, constant altitude coordinated turn, and yaw pointing. The evaluation consists of examining robustness and effect of first-order actuators in the simulation.

A robust controller design can perform satisfactorily at conditions other than the one it is specifically designed for. In this report robustness is examined by using a controller designed for a yaw pointing maneuver at a specific design flight condition, and using it without change in the simulation of a yaw pointing maneuver at other flight conditions. Also the same controller is used in the simulation of the three other maneuvers at the design flight condition. The performance of the three maneuvers is compared with the performance of controllers designed to perform the specific maneuver.

Three different controllers are designed to perform the yaw pointing maneuver. One at a low dynamic pressure, Q , design flight condition, another at a medium dynamic pressure design flight condition, and the

third at a high dynamic pressure design flight condition. Each of these controllers are then included in the simulation of a yaw pointing maneuver at nine other flight conditions, covering a spectrum of dynamic pressures.

A successful design is based on several criteria. In this report the following specifications are established: First, the closed-loop system must be stable. Second, all transient responses should be within 98% of their steady-state value in three seconds. Third, the steady-state response should achieve the commanded value with no observable error. Fourth, the control surfaces are not commanded beyond a pre-set limit.

Assumptions

The following assumptions are made to simplify the computations and analysis:

1. The earth is flat and non-rotating.
2. The atmosphere is at rest with respect to the earth.
3. The acceleration due to gravity is constant.
4. The aircraft is a rigid body.
5. The aircraft has a constant mass.
6. The aircraft performs maneuvers while flying straight with wings level, unless otherwise specified.
7. The motion of the aircraft about a straight and level trim condition can be adequately modeled by a linear dynamical system of equations.
8. The dynamics of the actuators can be adequately modeled by a first-order differential equation.
9. The side slip angle, β , can be measured.

Organization

First, the theory used in Porter's design is described, and the method for developing a controller is discussed. Also included in Chapter II are methods for including first-order actuator dynamics in the model. Chapter III presents the results of applying this method to an aircraft model. The effects of varying flight conditions and maneuver commands are described in detail. Also the effects of actuator dynamics in the simulation are discussed. Chapter IV discusses the results and provides recommendations for further study.

Included in the Appendices are discussion of the aircraft equations of motion, a computer program to calculate system zeros, and the computer program used to perform design iterations and simulations.

II. Theoretical Development

Porter's design methods (Ref 7) are both conceptually and computationally simple. These techniques utilize singular perturbation methods, and are applicable to the design of both analogue and digital controllers. As stated in the introduction these methods are applied to tracking systems incorporating high-gain error actuated analogue controllers.

High-gain frequently has the advantage of making the closed-loop system robust, or insensitive to plant parameter variations. Also, high-gain can produce a tight tracking loop which may be desirable from the viewpoint of disturbance rejection. Robustness is examined in this report, but disturbance rejection is not.

Error actuated control minimizes the difference between the commands and outputs. The closed-loop behavior becomes increasingly tight and non-interacting as the gain parameter is increased.

Singular perturbation methods are used to exhibit the asymptotic structure of the transfer function matrices. These methods examine the overall effect of letting certain parameters become smaller and smaller. By letting the perturbation parameter, ϵ , be the reciprocal of the forward path gain, g , the results of singular perturbation theory can be applied, since as $\epsilon \rightarrow 0$, $g = 1/\epsilon \rightarrow \infty$. With singular perturbation theory the asymptotic properties of the closed-loop transfer functions can be examined to determine the location of the closed-loop poles, as the gain becomes increasingly large. These asymptotic closed-loop poles determine the system response.

The tracking system consists of a linear multivariable plant governed on the continuous time set $T = (0, \infty)$ represented by state and output equations of the form

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{B} \underline{u}(t) \quad (2.1)$$

$$\underline{y}(t) = \underline{C} \underline{x}(t)$$

The state and output equations, eq. 2.1, expressed in expanded form are

$$\begin{bmatrix} \dot{\underline{x}}_1(t) \\ \dot{\underline{x}}_2(t) \end{bmatrix} = \begin{bmatrix} \underline{A}_{11} & \underline{A}_{12} \\ \underline{A}_{21} & \underline{A}_{22} \end{bmatrix} \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{B}_2 \end{bmatrix} \underline{u}(t) \quad (2.2)$$

$$\underline{y}(t) = [\underline{C}_1 \quad \underline{C}_2] \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} \quad (2.3)$$

Where $\underline{B}_2$ is a square non-singular matrix. Any system of the form of eq. 2.1 can, by proper state transformation, be put into the form of eqs. 2.2 and 2.3. In an improper system, where $\underline{C}_2 \underline{B}_2$ is rank deficient, the output measurements which track the command inputs are

$$\underline{w}(t) = [\underline{C}_1 + \underline{M} \underline{A}_{11}, \underline{C}_2 + \underline{M} \underline{A}_{12}] \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} \quad (2.4)$$

$$= [\underline{F}_1, \underline{F}_2] \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} \quad (2.5)$$

where $\underline{M}$ is an extra plant measurement matrix which must be designed so that $\underline{F}_2 \underline{B}_2$ has full rank. The trackers examined in this report are

improper, requiring a measurement matrix, $\underline{M}$. The state vector $\underline{x}(t) = (\underline{x}_1(t), \underline{x}_2(t))^T$ in these equations represent the kinematic variables used in the aircraft equations of motion

$$\underline{x}_1(t) = \phi(t) \quad (2.6)$$

$$\underline{x}_2(t) = (\beta(t), p(t), r(t))^T \quad (2.7)$$

The high-gain error actuated analogue controller is governed on τ by a proportional plus integral control law equation of the form

$$\underline{u}(t) = g(K_0 \underline{e}(t) + K_1 \underline{z}(t)) \quad (2.8)$$

where K_0 and K_1 are controller gain matrices, $\underline{e}(t)$ is the error vector

$$\underline{e}(t) = \underline{v}(t) - \underline{w}(t) \quad (2.9)$$

$\underline{v}(t)$ is the command vector, and $\underline{z}(t)$ is the time integral of $\underline{e}(t)$.

$$\dot{\underline{z}}(t) = \underline{e}(t) = [\underline{v}(t) - \underline{w}(t)] = [\underline{v}(t) - \underline{F}_1 \underline{x}_1(t) - \underline{F}_2 \underline{x}_2(t)] \quad (2.10)$$

A functional block diagram of the complete tracker is shown in Figure 2.1.

It is desired that the control input vector, $\underline{u}(t)$, cause the output vector, $\underline{y}(t)$, to track any constant command input vector, $\underline{v}(t)$, so that

$$\lim_{t \rightarrow \infty} (\underline{v}(t) - \underline{y}(t)) = 0 \quad (2.11)$$

Such tracking is possible using the controller form of eq. 2.8, due to the fact that the error vector, $\underline{e}(t)$, assumes the steady-state value (Ref 7).

$$\lim_{t \rightarrow \infty} \underline{e}(t) = \lim_{t \rightarrow \infty} (\underline{v}(t) - \underline{w}(t)) = 0 \quad (2.12)$$

The steady-state value of $\underline{w}(t)$ is

$$\lim_{t \rightarrow \infty} \underline{w}(t) = \lim_{t \rightarrow \infty} (\underline{F}_1 \underline{x}_1(t) + \underline{F}_2 \underline{x}_2(t)) \quad (2.13)$$

$$= \lim_{t \rightarrow \infty} ((\underline{C}_1 + \underline{M} \underline{A}_{11}) \underline{x}_1(t) + (\underline{C}_2 + \underline{M} \underline{A}_{12}) \underline{x}_2(t)) \quad (2.14)$$

$$= \lim_{t \rightarrow \infty} ((\underline{C}_1 \underline{x}_1(t) + \underline{C}_2 \underline{x}_2(t)) + \underline{M}(\underline{A}_{11} \underline{x}_1(t) + \underline{A}_{12} \underline{x}_2(t))) \quad (2.15)$$

$$= \lim_{t \rightarrow \infty} ((\underline{C}_1 \underline{x}_1(t) + \underline{C}_2 \underline{x}_2(t)) + \underline{M} \dot{\underline{x}}_1(t)) \quad (2.16)$$

For a stable system with a constant vector input, the states reach constant values, and therefore it is evident from the state equation, eq. 2.2, that

$$\lim_{t \rightarrow \infty} \dot{\underline{x}}_1(t) = \lim_{t \rightarrow \infty} (\underline{A}_{11} \underline{x}_1(t) + \underline{A}_{12} \underline{x}_2(t)) = 0 \quad (2.17)$$

Therefore eq. 2.16 yields

$$\lim_{t \rightarrow \infty} \underline{w}(t) = \lim_{t \rightarrow \infty} (\underline{C}_1 \underline{x}_1(t) + \underline{C}_2 \underline{x}_2(t)) \quad (2.18)$$

$$= \lim_{t \rightarrow \infty} (\underline{y}(t)) \quad (2.19)$$

Thus, in the steady-state the output, $\underline{y}(t)$ tracks the command input, $\underline{v}(t)$, for arbitrary initial conditions. Equations 2.12 through 2.19 show that tracking is possible for any measurement matrix, $\underline{M}$, provided that the condition that $\underline{F}_2 \underline{B}_2$ have full rank is satisfied.

In the above equations, $\underline{x}_1(t) \in \mathbb{R}^{n-l}$, $\underline{x}_2(t) \in \mathbb{R}^l$, $\underline{u}(t) \in \mathbb{R}^l$, $\underline{y}(t) \in \mathbb{R}^l$, $\underline{w}(t) \in \mathbb{R}^l$, $\underline{e}(t) \in \mathbb{R}^l$, $\underline{z}(t) \in \mathbb{R}^l$, $\underline{v}(t) \in \mathbb{R}^l$, $\underline{A}_{11} \in \mathbb{R}^{(n-l) \times (n-l)}$, $\underline{A}_{21} \in \mathbb{R}^{l \times (n-l)}$, $\underline{A}_{22} \in \mathbb{R}^{l \times l}$, $\underline{B}_2 \in \mathbb{R}^{l \times l}$, $\underline{C}_1 \in \mathbb{R}^{l \times (n-l)}$, $\underline{C}_2 \in \mathbb{R}^{l \times l}$, $\underline{F}_1 \in \mathbb{R}^{l \times (n-l)}$, $\underline{F}_2 \in \mathbb{R}^{l \times l}$, $\underline{K}_0 \in \mathbb{R}^{l \times l}$, $\underline{K}_1 \in \mathbb{R}^{l \times l}$, $\underline{M} \in \mathbb{R}^{l \times (n-l)}$.

The closed-loop equations are formed by combining eqs. 2.2, 2.3, 2.5, 2.8, and 2.10, to yield

$$\begin{bmatrix} \dot{\underline{z}}(t) \\ \dot{\underline{x}}_1(t) \\ \dot{\underline{x}}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & -\underline{F}_1 & -\underline{F}_2 \\ 0 & \underline{A}_{11} & \underline{A}_{12} \\ g\underline{B}_2 \underline{K}_1, & \underline{A}_{21} - g\underline{B}_2 \underline{K}_0 \underline{F}_1, & \underline{A}_{22} - g\underline{B}_2 \underline{K}_0 \underline{F}_2 \end{bmatrix} \begin{bmatrix} \underline{z}(t) \\ \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} + \begin{bmatrix} \underline{I}_l \\ 0 \\ g\underline{B}_2 \underline{K}_0 \end{bmatrix} \underline{v}(t) \quad (2.20)$$

$$\underline{y}(t) = \begin{bmatrix} 0, & \underline{C}_1, & \underline{C}_2 \end{bmatrix} \begin{bmatrix} \underline{z}(t) \\ \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} \quad (2.21)$$

The closed-loop transfer function matrix relating the plant output vector to the command input vector of the system governed by eqs 2.20 and 2.21 is

$$\underline{G}(\lambda) = (0, \underline{C}_1, \underline{C}_2) \begin{bmatrix} \lambda \underline{I}_\ell & , & \underline{F}_1 & , & \underline{F}_2 \\ 0 & , & \lambda \underline{I}_{n-\ell} - \underline{A}_{11} & , & -\underline{A}_{12} \\ -g\underline{B}_2 \underline{K}_1, & -\underline{A}_{21} + g\underline{B}_2 \underline{K}_0 \underline{F}_1, & -\lambda \underline{I}_\ell - \underline{A}_{22} + g\underline{B}_2 \underline{K}_0 \underline{F}_2 \end{bmatrix}^{-1} \begin{bmatrix} \underline{I}_\ell \\ 0 \\ g\underline{B}_2 \underline{K}_0 \end{bmatrix} \quad (2.22)$$

which is equivalent to

$$\underline{G}(\lambda) = \underline{C}_{CL} \begin{bmatrix} \lambda \underline{I}_{n+\ell} - \underline{A}_{CL} \end{bmatrix}^{-1} \underline{B}_{CL} \quad (2.23)$$

where

$$\underline{A}_{CL} = \begin{bmatrix} \underline{A}_1 & \underline{A}_2 \\ \underline{A}_3/\varepsilon & \underline{A}_4/\varepsilon \end{bmatrix} \quad (2.24)$$

$$\underline{B}_{CL} = \begin{bmatrix} \underline{B}_1 \\ \underline{B}_2/\varepsilon \end{bmatrix} \quad (2.25)$$

$$\underline{C}_{CL} = \begin{bmatrix} \underline{C}_1, \underline{C}_2 \end{bmatrix} \quad (2.26)$$

and

$$\underline{A}_1 = \begin{bmatrix} 0 & -\underline{F}_1 \\ 0 & \underline{A}_{11} \end{bmatrix} \quad (2.27)$$

$$\underline{A}_2 = \begin{bmatrix} -\underline{F}_2 \\ \underline{A}_{12} \end{bmatrix} \quad (2.28)$$

$$\underline{A}_3 = \begin{bmatrix} +\underline{B}_2 \underline{K}_1, \underline{A}_{21}\varepsilon - \underline{B}_2 \underline{K}_0 \underline{F}_1 \end{bmatrix} \quad (2.29)$$

$$\underline{A}_4 = \begin{bmatrix} \underline{A}_{22}\varepsilon - \underline{B}_2 \underline{K}_0 \underline{F}_2 \end{bmatrix} \quad (2.30)$$

$$\underline{B}_1 = \begin{bmatrix} \underline{I}_\ell \\ 0 \end{bmatrix} \quad (2.31)$$

$$\underline{B}_2 = \begin{bmatrix} \underline{B}_2 & \underline{K}_0 \end{bmatrix} \quad (2.32)$$

$$\underline{C}_1 = \begin{bmatrix} 0, & \underline{C}_1 \end{bmatrix} \quad (2.33)$$

$$\underline{C}_2 = \begin{bmatrix} \underline{C}_2 \end{bmatrix} \quad (2.34)$$

By letting the perturbation parameter $\epsilon = 1/g$ go to zero the transfer function matrix $\underline{G}(\lambda)$, approaches (Ref 7) the asymptotic form

$$\underline{\Gamma}(\lambda) = \underline{\tilde{\Gamma}}(\lambda) + \underline{\hat{\Gamma}}(\lambda) \quad (2.35)$$

where

$$\underline{\tilde{\Gamma}}(\lambda) = \underline{C}_0 \left[\lambda \underline{I}_n - \underline{A}_0 \right]^{-1} \underline{B}_0 \quad (2.36)$$

and

$$\underline{\hat{\Gamma}}(\lambda) = \underline{C}_2 \left[\lambda \underline{I}_\ell - g \underline{A}_4 \right]^{-1} g \underline{B}_2 \quad (2.37)$$

The matrices in eq. 2.36 are

$$\underline{A}_0 = \begin{bmatrix} \underline{A}_1 - \underline{A}_2 & \underline{A}_4^{-1} \underline{A}_3 \end{bmatrix} = \begin{bmatrix} -\underline{K}_0^{-1} \underline{K}_1 & 0 \\ \underline{A}_{12} \underline{F}_2^{-1} \underline{K}_0 \underline{K}_1 & \underline{A}_{11} - \underline{A}_{12} \underline{F}_2^{-1} \underline{F}_1 \end{bmatrix} \quad (2.38)$$

$$\underline{B}_0 = \begin{bmatrix} \underline{B}_1 - \underline{A}_2 \underline{A}_4^{-1} \underline{A}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \underline{A}_{12} \underline{F}_2^{-1} \end{bmatrix} \quad (2.39)$$

$$\underline{C}_0 = \begin{bmatrix} \underline{C}_1 - \underline{C}_2 \underline{A}_4^{-1} \underline{A}_3 \end{bmatrix} = \begin{bmatrix} \underline{C}_2 \underline{F}_2^{-1} \underline{K}_0^{-1} \underline{K}_1, & \underline{C}_1 - \underline{C}_2 \underline{F}_2^{-1} \underline{F}_1 \end{bmatrix} \quad (2.40)$$

The "slow" modes of the tracking system are defined as those in which the eigenvalues are not direct functions of g . These are the poles of $\underline{\tilde{\Gamma}}(\lambda)$ and are easily found from

$$\det \left[\lambda \underline{I}_n - \underline{A}_0 \right] = 0 \quad (2.41)$$

Since $\underline{A}_0$ is lower block triangular, eq. 2.38 yields two sets of slow poles described by

$$z_1 = \{ \lambda \in \mathbb{C} : | \lambda \underline{K}_0 + \underline{K}_1 | = 0 \} \quad (2.42)$$

and

$$z_2 = \{ \lambda \in \mathbb{C} : | \lambda \underline{I}_{n-l} - \underline{A}_{11} + \underline{A}_{12} \underline{F}_2^{-1} \underline{F}_1 | = 0 \} \quad (2.43)$$

Set z_2 contains the transmission zeros of the system.

The "fast" modes of the tracking system are those in which the eigenvalues are directly dependent on g . These eigenvalues are the poles of $\hat{\underline{\Gamma}}(\lambda)$ and are easily found from

$$\det [\lambda \underline{I}_\ell - g \underline{A}_4] = 0 \quad (2.44)$$

This equation yields the set of fast poles described by

$$z_3 = \{ \lambda \in \mathbb{C} : | \lambda \underline{I}_\ell + g \underline{F}_2 \underline{B}_2 \underline{K}_0 | = 0 \} \quad (2.45)$$

In order for tracking to occur, the closed-loop system must obviously be stable, requiring that all the eigenvalues in the sets z_1 , z_2 and z_3 be in the left half of the complex plane. Therefore, for sufficiently large gains, the controller matrices, $\underline{K}_0$ and $\underline{K}_1$, and measurement matrix, $\underline{M}$, must be chosen so that $z_1 \in \mathbb{C}^-$, $z_2 \in \mathbb{C}^-$, and $z_3 \in \mathbb{C}^-$.

To achieve a non-interacting system it is desired to have a diagonal asymptotic transfer function matrix $\underline{\Gamma}(\lambda)$. In order for the fast asymptotic transfer function, $\hat{\underline{\Gamma}}(\lambda)$, to be diagonal it is neces-

$$\underline{F}_2 \underline{B}_2 \underline{K}_0 = (\underline{C}_2 + \underline{M} \underline{A}_{12}) \underline{B}_2 \underline{K}_0 = \underline{\Sigma} = \text{diag } \sigma_1, \sigma_2, \dots, \sigma_\ell \quad (2.46)$$

where $\sigma_j \in \mathbb{R}^+$ ($j = 1, 2, \dots, \ell$). By selecting appropriate values for σ_j , the elements of $\underline{K}_0$ can then be determined

$$\underline{K}_0 = (\underline{F}_2 \underline{B}_2)^{-1} \underline{\Sigma} \quad (2.47)$$

Thus the fast asymptotic eigenvalues are

$$z_3 = \{-\sigma_1 g, -\sigma_2 g, \dots, -\sigma_{\ell} g\} \quad (2.48)$$

$\underline{K}_1$ may be chosen as a positive multiple of $\underline{K}_0$

$$\underline{K}_1 = b \underline{K}_0 \quad (2.49)$$

Solving eq. 2.40, the first set of slow asymptotic eigenvalues are

$$z_1 = \{-b\} \quad (2.50)$$

There are two methods, presented by Ridgley, Banda, and D'Azzo (Ref 9), for selecting a measurement matrix, $\underline{M}$, which diagonalizes the slow asymptotic transfer function, $\tilde{\underline{G}}(\lambda)$. The first method is used when $n > 2\ell$ and when

$$\det \underline{B}^* = \det \begin{bmatrix} \underline{C}_1^T & \underline{A}'^{d_1} \underline{B}' \\ \vdots & \\ \underline{C}_\ell^T & \underline{A}'^{d_\ell} \underline{B}' \end{bmatrix} \neq 0 \quad (2.51)$$

where $\underline{A}' = \underline{A}_{11}$, $\underline{B}' = \underline{A}_{12}$, $\underline{C}' = \underline{C}_1$, $\underline{C}_i^T$ is the i th row of $\underline{C}'$, and $d_i = \min (j: \underline{C}_i^T \underline{A}^{1j} \underline{B}' \neq 0, j = 1, 2, \dots, n-\ell-1)$ or $d_i = n-\ell-1$ if $\underline{C}_i^T \underline{A}^{1j} \underline{B}' = 0$ for all j .

The measurement matrix, $\underline{M}$, is chosen to satisfy

$$\underline{F}_2 = (\underline{C}_2 + \underline{M} \underline{A}_{12}) = \underline{B}^* \quad (2.52)$$

and

$$\underline{F}_1 = (\underline{C}_1 + \underline{M} \underline{A}_{11}) = - \left[\sum_{k=0}^{\delta} \underline{N}_k \underline{C}' \underline{A}'^k - \underline{A}^* \right] \quad (2.53)$$

where δ is the maximum value of d_i , $\underline{M}_k$ are suitably chosen diagonal matrices, and

$$\underline{A}^* = \begin{bmatrix} \underline{C}_1^T & \underline{A}'^{d_1} \\ \vdots & \vdots \\ \underline{C}_\ell^T & \underline{A}'^{d_\ell} \end{bmatrix} \underline{A}' \quad (2.54)$$

It is not always possible to find measurement matrix, $\underline{M}$, which satisfies both eqs. 2.52 and 2.53. In this case it is not possible to diagonalize the slow asymptotic transfer function matrix, $\underline{\hat{\Gamma}}(\lambda)$.

The second method is used when $n < 2\ell$ and $\det \underline{B}^* = 0$. The assignable elements of $\underline{F}_2 = (\underline{C}_2 + \underline{M} \underline{A}_{12})$, those containing elements of $\underline{M}$, are permitted non-zero values only if $\underline{B}^*$ has a non-zero element in the corresponding position. Also $\underline{C}_2 \underline{F}_2^{-1}$ should be chosen to be diagonal. Again it is not always possible to find a matrix $\underline{M}$ which diagonalizes the slow asymptotic transfer function, $\underline{\hat{\Gamma}}(\lambda)$.

When there is no measurement matrix, $\underline{M}$, which will decouple the system outputs and thus make the asymptotic transfer function, $\underline{\hat{\Gamma}}(\lambda)$, diagonal, then $\underline{M}$ must be iteratively selected, aiming at a closed-loop system that is stable and nearly non-interactive. This is a limitation of these two methods.

When it is desired to include first-order actuators into the model, two methods are available. One is to add an actuator transfer function to the forward loop path, as shown in Figure 2.2. The other is to augment the open-loop state equations.

The actuator transfer function, in the Laplace domain, for first-order actuators is

$$\underline{G}_A(s) = \underline{A}_C \left[sI + \underline{A}_C \right]^{-1} \quad (2.55)$$

where $\underline{A}_C$ is a diagonal matrix containing the l values of positive actuator constants. The resulting vector differential equation is

$$\dot{\underline{u}}_A(t) = -\underline{A}_C \underline{u}_A(t) + \underline{A}_C \underline{u}(t) \quad (2.56)$$

Adding the actuator transfer function, $\underline{G}_A$, to the forward path augments the closed-loop equations, eqs. 2.20 and 2.21. Combining eqs. 2.5, 2.8, 2.9, and 2.56 yields

$$\dot{\underline{u}}_A(t) = -\underline{A}_C \underline{u}_A(t) + \underline{A}_C \left[g(\underline{K}_0 \underline{e}(t) + \underline{K}_1 \underline{z}(t)) \right] \quad (2.57)$$

$$= -\underline{A}_C \underline{u}_A(t) + g\underline{A}_C \left[\underline{K}_0 (\underline{u}(t) - \underline{F}_1 \underline{x}_1(t) - \underline{F}_2 \underline{x}_2(t)) + \underline{K}_1 \underline{z}(t) \right] \quad (2.58)$$

The closed-loop state vector is augmented to include $\underline{u}_A$ as a state. The resulting closed-loop equations are

$$\begin{bmatrix} \dot{\underline{z}}(t) \\ \dot{\underline{x}}_1(t) \\ \dot{\underline{x}}_2(t) \\ \dot{\underline{u}}_A(t) \end{bmatrix} = \begin{bmatrix} 0 & , & -\underline{F}_1 & , & -\underline{F}_2 & , & 0 \\ 0 & , & \underline{A}_{11} & , & \underline{A}_{12} & , & 0 \\ 0 & , & \underline{A}_{21} & , & \underline{A}_{22} & , & \underline{B}_2 \\ g\underline{A}_C \underline{K}_1 & , & -g\underline{A}_C \underline{K}_0 \underline{F}_1 & , & -g\underline{A}_C \underline{K}_0 \underline{F}_2 & , & -\underline{A}_C \end{bmatrix} \begin{bmatrix} \underline{z}(t) \\ \underline{x}_1(t) \\ \underline{x}_2(t) \\ \underline{u}_A(t) \end{bmatrix} + \begin{bmatrix} \underline{I}_l \\ 0 \\ 0 \\ g\underline{A}_C \underline{K}_0 \end{bmatrix} \underline{u}(t) \quad (2.59)$$

$$\underline{y}(t) = \begin{bmatrix} 0, & \underline{c}_1, & \underline{c}_2, & 0 \end{bmatrix} \begin{bmatrix} \underline{z}(t) \\ \underline{x}_1(t) \\ \underline{x}_2(t) \\ \underline{u}_A(t) \end{bmatrix} \quad (2.60)$$

To add the first-order actuators by augmenting the open-loop state equations, the open-loop state vector is expanded to include $\underline{u}_A$. The resulting open-loop equations are

$$\begin{bmatrix} \dot{\underline{x}}_1'(t) \\ \dot{\underline{x}}_2'(t) \end{bmatrix} = \begin{bmatrix} \underline{A}_{11}' & \underline{A}_{12}' \\ \underline{A}_{21}' & \underline{A}_{22}' \end{bmatrix} \begin{bmatrix} \underline{x}_1'(t) \\ \underline{x}_2'(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{B}_2' \end{bmatrix} \bar{\underline{u}}(t) \quad (2.61)$$

$$\underline{y}(t) = \begin{bmatrix} \underline{C}_1' & \underline{C}_2' \end{bmatrix} \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} \quad (2.62)$$

where

$$\underline{x}_1' = \begin{bmatrix} \underline{x}_1(t) \\ \underline{x}_2(t) \end{bmatrix} \quad (2.63)$$

$$\underline{x}_2' = \begin{bmatrix} \underline{u}_A(t) \end{bmatrix} \quad (2.64)$$

$$\underline{A}_{11}' = \begin{bmatrix} \underline{A}_{11} & \underline{A}_{12} \\ \underline{A}_{21} & \underline{A}_{22} \end{bmatrix} \quad (2.65)$$

$$\underline{A}_{12}' = \begin{bmatrix} 0 \\ \underline{B}_2 \end{bmatrix} \quad (2.66)$$

$$\underline{A}_{21}' = \begin{bmatrix} 0 \end{bmatrix} \quad (2.67)$$

$$\underline{A}_{22}' = \begin{bmatrix} -\underline{A}_c \end{bmatrix} \quad (2.68)$$

$$\underline{B}_2' = \begin{bmatrix} \underline{A}_c \end{bmatrix} \quad (2.69)$$

$$\underline{C}_1' = \begin{bmatrix} \underline{C}_1' & \underline{C}_2' \end{bmatrix} \quad (2.70)$$

$$\underline{C}_2' = \begin{bmatrix} 0 \end{bmatrix} \quad (2.71)$$

Equations 2.62 and 2.63 are substituted for eqs. 2.2 and 2.3, then the theory outlined in this chapter is applied.

In this report the first-order actuators are added by augmenting the closed-loop equations.

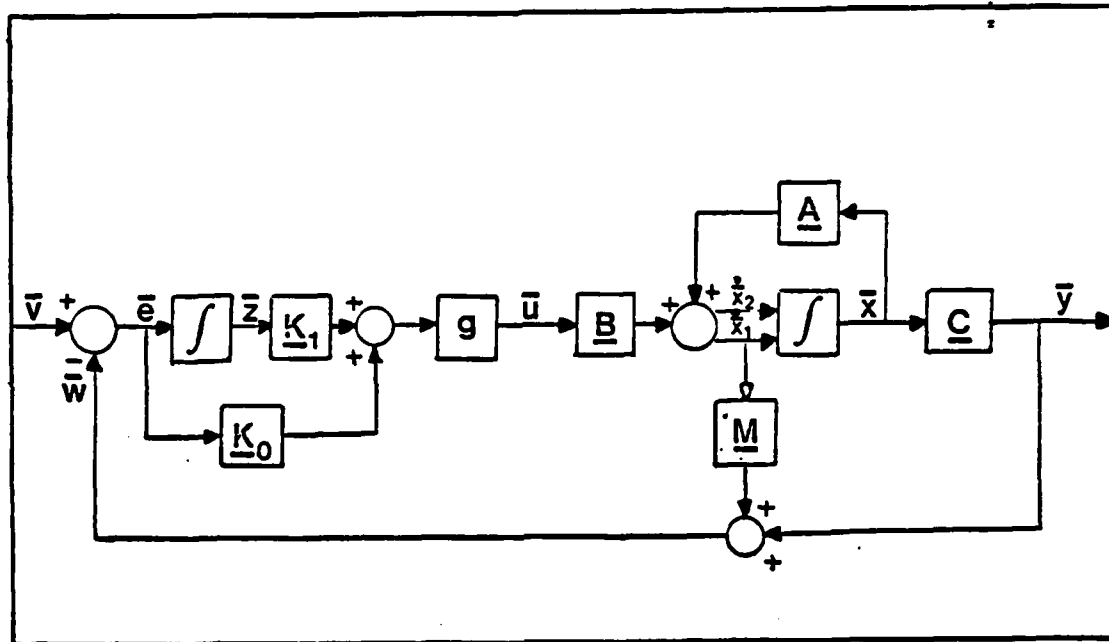


Figure 2-1 Tracking System

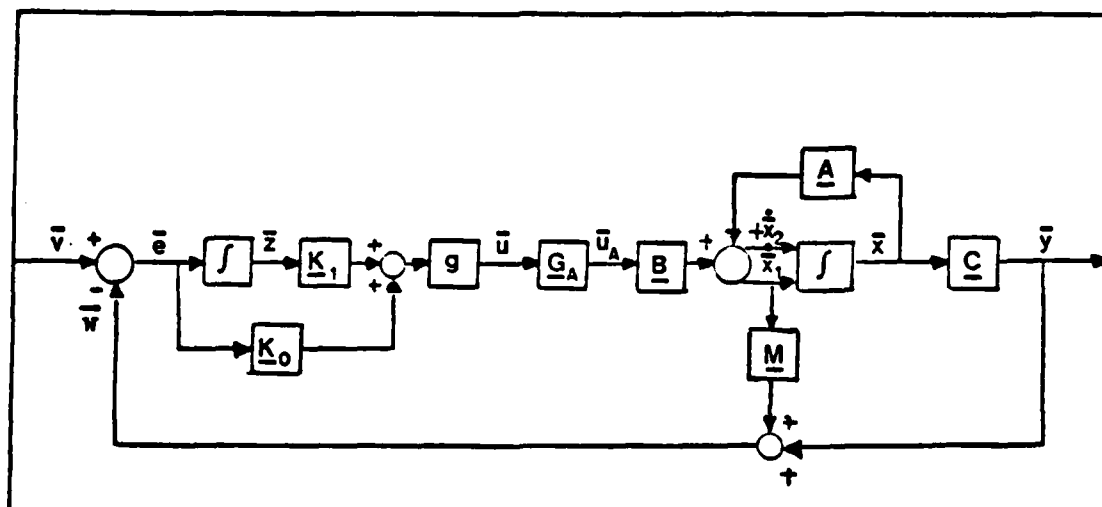


Figure 2-2 Tracking System with Actuators

III. Discussion of Results

Design Procedure

As a result of this thesis it is concluded that Porter's design method is easy to apply. Satisfactory controllers are found for a specific maneuver and flight condition without many design iterations. Also a designer's "feel" is quickly obtained, making controller designs easy to obtain. A feel is when the designer knows how different parameters affect the output response. The effect of the various parameters discovered during the design iterations for the controllers of this report are now discussed.

The measurement matrix, $\underline{M}$, is selected as a sparse matrix using the second method discussed in Chapter II (Ref 9). The reciprocal of the non-zero element of the $\underline{M}$ matrix becomes the transmission zero of the system. A transmission zero of -4 is selected for all of the controller designs in this report. This value is recommended by Lt Ridgely (Ref 10).

The value of the sigma matrix, $\underline{\Sigma}$, does affect the outputs in a predictable way. To keep the outputs decoupled, and the fast asymptotic transfer function, $\hat{\underline{F}}$, diagonal, $\underline{\Sigma}$ is selected as a diagonal matrix. Each diagonal element, σ_i , affects the corresponding output (the first diagonal element affects the first output, etc.). Increasing the value of the element causes the output to track the command input closer. A large diagonal sigma element is desirable when the specific output response is commanded to zero. An adverse effect of a large diagonal element is that the magnitude of the control surface responses are increased. Decreasing the diagonal element causes an

increase in the transient response of the output. Varying the diagonal elements in the sigma matrix is the main avenue used to achieve desired output and control surface time responses for the controller's designs contained in this report. An observation made during the design iterations is that an optimal sigma matrix found for a specific maneuver and specific flight condition is very close to the optimal sigma matrix for the same maneuver at a different design flight condition. This observation is shown by comparing the sigma matrices for the three yaw pointing designs, Tables 3-2 through 3-4.

The gain value, g , is used in the design iterations for the controllers in this report, to adjust the peak value and settling time of the response to a step input. The gain can be adjusted so the output response has no noticeable peak and an acceptable settling time. For all values of gain an overshoot is found to exist. Very high gain and very low gain cause a large overshoot. The gain for a minimum overshoot is a medium value. Settling time decreases with increasing gain value.

The value of the input commands can be used to adjust the control surface time responses. Changes to the values of the input commands do not adversely affect the output responses, in that they will track the commands. The control surface responses are, naturally, sensitive to the magnitude of the input command. The value of the input command can be reduced to keep the control surface deflections within limits. This is the last parameter to be adjusted in the design procedure. The initial values of input commands are obtained from AFTI wind tunnel data (Ref 12).

The control surface deflection limits for the designs of this report are:

$$\text{Rudder, } -35^\circ \leq \delta_r \leq 35^\circ$$

$$\text{Aileron, } -35^\circ \leq \delta_a \leq 35^\circ$$

$$\text{Cannard, } -30^\circ \leq \delta_c \leq 30^\circ$$

These limit values are selected by the designer and considered reasonable.

The interactive computer program, listed in Appendix C, make rapid design iterations possible.

Robustness Tests

As stated in the Introduction, the robustness tests are performed with the yaw pointing maneuver. The output matrix is

$$[C_1:C_2] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

The values of the input commands to produce a yaw pointing angle are shown in Table 3-1. The side-slip angle, β , is commanded as a ramp with the slope $-\beta_0$ until the value of the side-slip angle reaches minus three degrees. At this time, τ , the slope is changed to zero. The ramp input is used as a way of commanding the time rate of the side-slip angle, $\dot{\beta}$. The roll angle, ϕ , is commanded to zero. The yaw rate, r , is commanded as a pulse with a constant magnitude equal to β_0 . The value of β_0 is a function of the dynamic pressure, Q , which is determined by the flight condition. At time τ the value of yaw rate is commanded to zero.

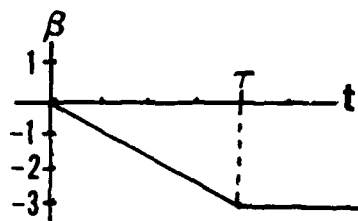
TABLE 3-1

INPUT COMMANDS

1) Side Slip Angle, β

$$\beta(t) = -\beta_0 t \quad 0 \leq t \leq \tau$$

$$= -3 \quad t > \tau$$

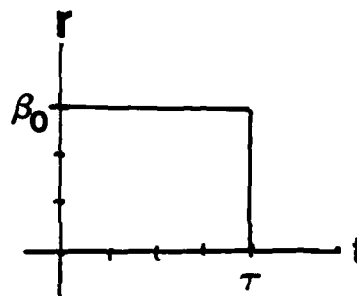
2) Roll Angle, ϕ

$$\phi(t) = 0 \quad t \geq 0$$

3) Yaw Rate, r

$$r(t) = \beta_0 \quad 0 \leq t \leq \tau$$

$$= 0 \quad t > \tau$$



Q CONDITION	SLOPE, β_0 (deg/sec)	TIME CONSTANT, τ (seconds)	$\int_0^{\tau} \beta(t) dt$ (deg-sec)	$\int_0^{\tau} r(t) dt$ (degrees)
Low	0.80	3.75	5.625	3.00
Low-Mid	1.00	3.00	4.500	3.00
Mid	3.00	1.00	1.500	3.00
High	4.00	0.75	1.125	3.00

The results of the yaw pointing robustness test for varying flight conditions are summarized in Tables 3-2 through 3-4 and Figures 3-2a through 3-2f.* In all cases, except one (Design 1A, $Q = 825$) the controller performs the desired yaw pointing maneuver. This is shown by the fact that the steady-state side-slip angle, β_{ss} , and the steady-state pointing angle, ψ_{ss} , are equal in magnitude and opposite in sign. The pointing angle, ψ , is the time integral of the yaw rate

$$\psi(t) = \int_0^t r(t) dt \quad (3.2)$$

The most noticeable difference, when the controller is used outside its design flight condition, is the response of the yaw rate, r . Outside the design flight condition the yaw rate has an initial overshoot and a few oscillations. This does not affect the performance of the maneuver, based on the smooth response of the side-slip angle and the pointing angle.

Figures 3-3 through 3-5 show the output and control surface time responses of all three controllers performing a yaw pointing maneuver at a specific flight condition. Figure 3-3 shows the responses of a low dynamic pressure maneuver ($Q = 109$). Figure 3-4 shows the responses of a medium dynamic pressure maneuver ($Q = 443$). Figure 3-5 shows the responses of a high dynamic pressure maneuver ($Q = 825$).

The controller designed at a low dynamic pressure, Design 1A, consistently overcommanded the control surfaces when performing the yaw pointing maneuver outside its design flight condition, as shown

*Figure 3-1 shows the basis for the data in Tables 3-2 through 3-4.

in Figure 3-2d. Controllers designed at medium dynamic pressure, Design 3A, only overcommanded the control surfaces at two of the ten flight conditions simulated, and the value of the deflection error is not great.

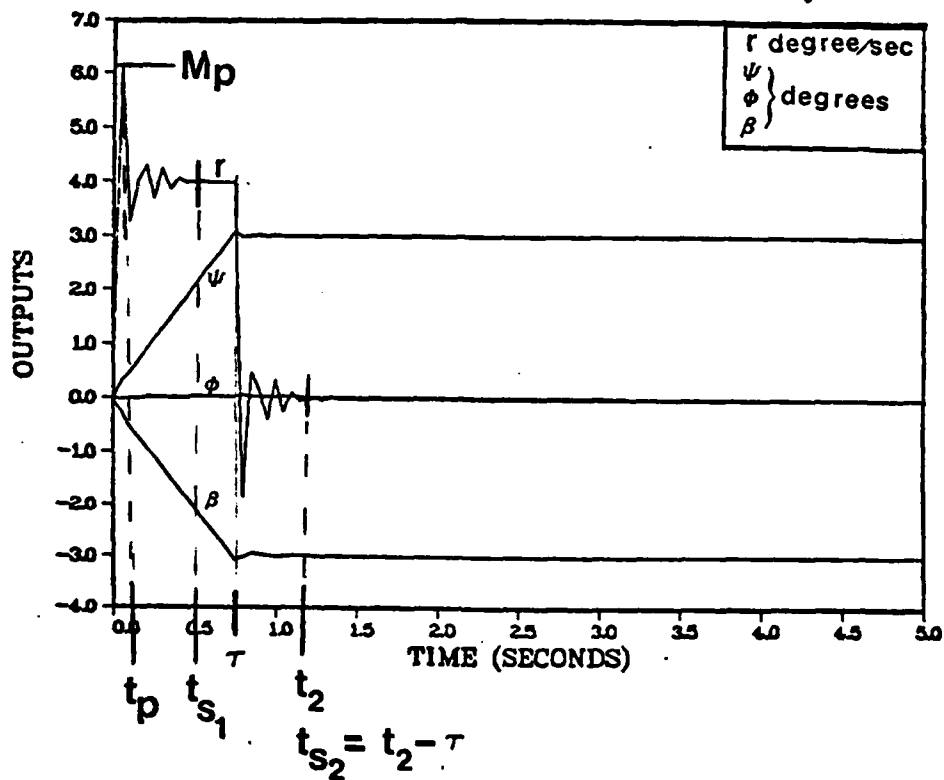
Figures 3-2e and 3-2f show the range of motion of the velocity vector pointing angle. The velocity vector is the sum of the sideslip angle and the pointing angle. Figure 3-2e shows the range of values that the velocity vector pointing angle moves. In the worst case (Design 1A, $Q = 825$) the total horizontal translation is not greater than 10 feet, during the time of the maneuver, τ . In this case the steady-state velocity vector pointing angle is 0.13 degrees, resulting in a steady-state horizontal velocity of 2.2 feet/second.

Figure 3-2c shows the time integral of the roll angle. This quantity is used as a judgement of performance, and a basis for comparison. In the worst case (Design 3A, $Q = 70$) the value is 0.176 degree-second. This is equivalent to a roll angle of 0.06 degrees for three seconds. The magnitudes of the roll angle time integral shows that in all cases the plane remained essentially wings level.

Based on all performance criteria shown in Tables 3-2 through 3-4, and Figures 3-2a through 3-2f, the controller designed for a medium dynamic pressure flight condition, Design 2A, is the more robust design.

Another robustness test is judging the performance of a controller designed to perform a specific maneuver at a specific flight condition performing other maneuvers at the same flight condition.

DESIGN 2A Y.P. AT HIGH Q



$$\%error = \frac{M_p - \beta_0}{\beta_0} 100$$

Figure 3-1 Figures of Merit for Output Responses

TABLE 3-2 Design 1A Figures of Merit

DESIGN 1A LOW Q Q = 109 lb/ft² h = 20,000 ft Mach = 0.4

$$\underline{M} = \begin{bmatrix} 0 \\ 0.25 \\ 0 \end{bmatrix} \quad \underline{\Sigma} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 30 \end{bmatrix} \quad \underline{A}_0 = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

$$\underline{K}_0 = \begin{bmatrix} 29.1768 & 1.6411 & -6.4843 \\ 5.4674 & -4.3414 & -1.6377 \\ 44.6171 & 0.5608 & 20.4908 \end{bmatrix} \quad \underline{K}_1 = \underline{K}_0 \quad g = 29.5$$

Q	β_{ss}	$\int_0^t r(\xi) d\xi$ ψ_{ss}	M_p	%error	t_p	t_{s_1}	t_{s_2}	$\int_0^t \phi(\xi) d\xi$	CONTROL SURFACE COMMENT
70	-2.999	2.993	1.0508	31.350	0.19	0.40	0.45	0.0414	exceeds limits $\delta_c > 38.0^\circ$
109	-2.999	3.012	0.8197	2.462	0.35	0.40	0.30	0.0242	within limits
163	-2.999	3.005	1.0114	28.453	0.10	0.45	0.40	0.0162	exceeds limits $\delta_c > 10.9^\circ$
223	-2.999	2.992	1.5377	53.770	0.05	0.40	0.45	0.0188	within limits
436	-2.999	3.018	3.7456	24.853	0.10	0.45	0.35	0.0166	exceeds limits $\delta_c > 28.9^\circ$
443	-2.999	3.037	3.6850	22.820	0.15	0.45	0.35	0.0147	exceeds limits $\delta_c > 43.1^\circ$
532	-2.999	3.007	4.2513	41.710	0.05	0.40	0.40	0.0153	exceeds limits $\delta_c > 28.1^\circ$
652	-2.999	2.9915	4.5565	51.883	0.05	0.40	0.40	0.0164	exceeds limits $\delta_c > 33.9^\circ$
825	-2.999	3.132	4.2150	5.375	0.60	0.75	0.10	0.0159	exceeds limits $\delta_r > 10.6^\circ$
1198	-2.999	2.993	6.3388	58.470	0.05	0.40	0.15	0.0171	exceeds limits $\delta_r > 6.5^\circ$

TABLE 3-3 Design 2A Figures of Merit
 DESIGN 2A MID Q Q = 443 lb/ft² h = 5,000 Mach = 0.6

$$\underline{N} = \begin{bmatrix} 0 \\ 0.25 \\ 0 \end{bmatrix} \quad \underline{\Sigma} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 30 \end{bmatrix} \quad \underline{A}_0 = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

$$\underline{\Sigma}_0 = \begin{bmatrix} 11.6944 & 0.1981 & -2.0615 \\ 2.7389 & -1.0522 & -0.1395 \\ 14.6517 & -0.2256 & 4.4020 \end{bmatrix} \quad \underline{K}_1 = 2\underline{\Sigma}_0 \quad s = 29.0$$

Q	β_{ss}	$\int_0^t r(\xi) d\xi$ ψ_{ss}	M _p	%error	t _p	t _{s1}	t _{s2}	$\int_0^t \phi(\xi) d\xi$	CONTR SURFACE COMME
70	-3.000	3.011	1.1771	47.1375	0.05	0.50	0.45	0.1069	within limits
109	-3.000	3.001	1.3366	67.0750	0.05	0.45	0.45	0.0644	within limits
163	-3.000	2.995	1.1883	48.5250	0.05	0.45	0.45	0.0459	within limits
223	-3.000	2.9821	1.4072	40.7200	0.10	0.45	0.40	0.0463	within limits
436	-3.000	3.053	3.0801	2.6700	0.40	0.30	0.30	0.0385	within limits
443	-3.000	3.043	3.0414	1.3800	0.40	0.20	0.35	0.0342	within limits
532	-3.000	3.020	3.6418	21.3933	0.15	0.40	0.40	0.0324	within limits
652	-3.000	3.011	4.1880	39.6000	0.10	0.50	0.35	0.0368	within limits
825	-3.000	2.993	6.0860	52.1500	0.05	0.40	0.40	0.0367	exceeds limits $\delta_c > 2.0$
1198	-3.000	2.981	5.4505	36.2625	0.05	0.45	0.40	0.0365	exceeds limits $\delta_c > 5.0$

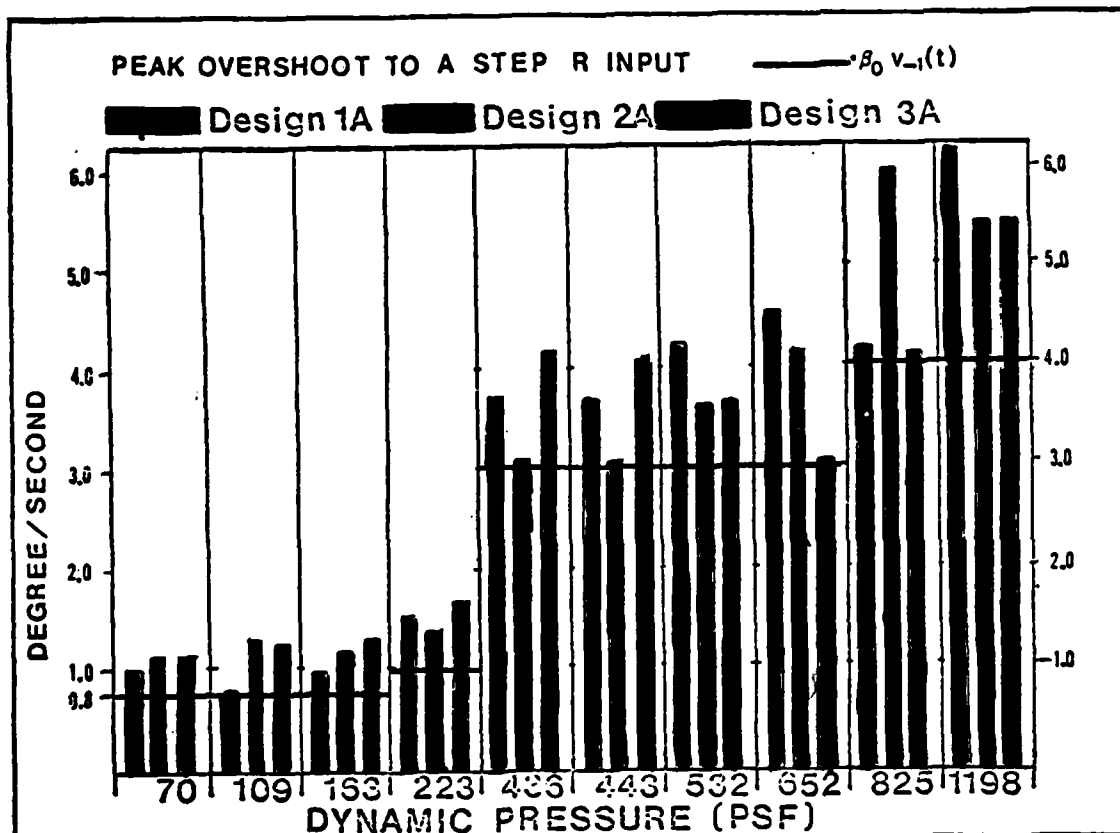
TABLE 3-4 Design 3A Figures of Merit

DESIGN 3A HIGH Q $Q = 825 \text{ lb/ft}^2$ $h = 10,000 \text{ ft}$ Mach = 0.9

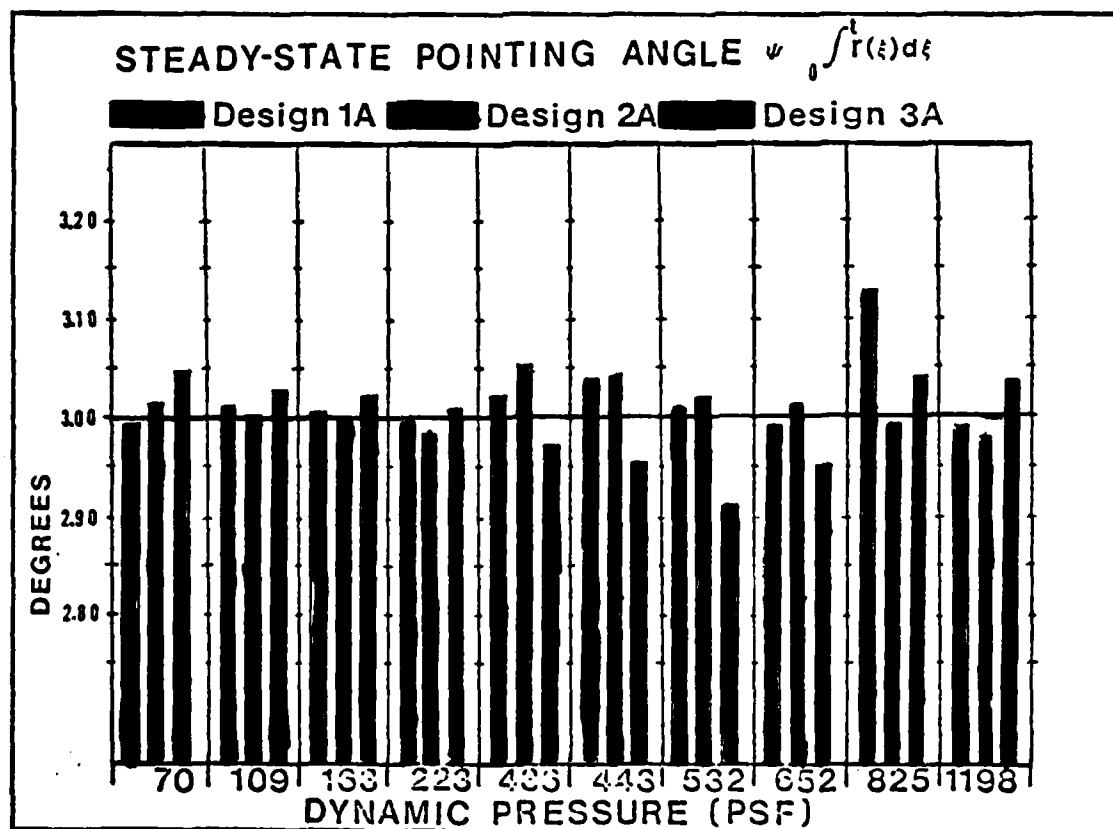
$$\underline{M} = \begin{bmatrix} 0 \\ 0.25 \\ 0 \end{bmatrix} \quad \underline{\Sigma} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 30 \end{bmatrix} \quad \underline{A}_0 = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

$$\underline{K}_0 = \begin{bmatrix} 22.5871 & 0.1857 & -1.6393 \\ 5.9893 & -0.7110 & -0.0607 \\ 22.3193 & -0.1024 & 1.9477 \end{bmatrix} \quad \underline{K}_1 = 2\underline{K}_0 \quad g = 29.5$$

Q	β_{ss}	$\int_0^t r(\xi) d\xi$ ψ_{ss}	M_p	%error	t_p	t_{s_1}	t_{s_2}	$\int_0^t \phi(\xi) d\xi$	CONTROL SURFACE COMMENT
70	-3.000	3.045	1.1625	45.312	0.10	0.80	0.60	0.1763	within limits
109	-3.000	3.025	1.2601	57.512	0.05	0.50	0.55	0.1119	within limits
163	-3.000	3.021	1.3857	73.212	0.05	0.50	0.45	0.0829	within limits
223	-3.000	3.009	1.6871	68.710	0.05	0.45	0.50	0.1361	within limits
436	-3.000	2.972	4.1703	39.010	0.10	0.50	0.60	0.0705	exceeds limits $\delta_c > 8.6^\circ$
443	-3.000	2.952	4.0944	36.480	0.10	0.40	0.65	0.0662	exceeds limits $\delta_c > 2.3^\circ$
532	-3.000	2.907	3.6494	21.646	0.15	0.95	0.60	0.0632	within limits
652	-3.000	2.951	3.0646	2.153	0.30	0.40	0.55	0.0652	within limits
825	-3.000	3.041	4.1046	2.615	0.35	0.45	0.60	0.0635	within limits
1198	-3.000	3.045	5.4566	36.415	0.10	0.40	0.55	0.0655	within limits

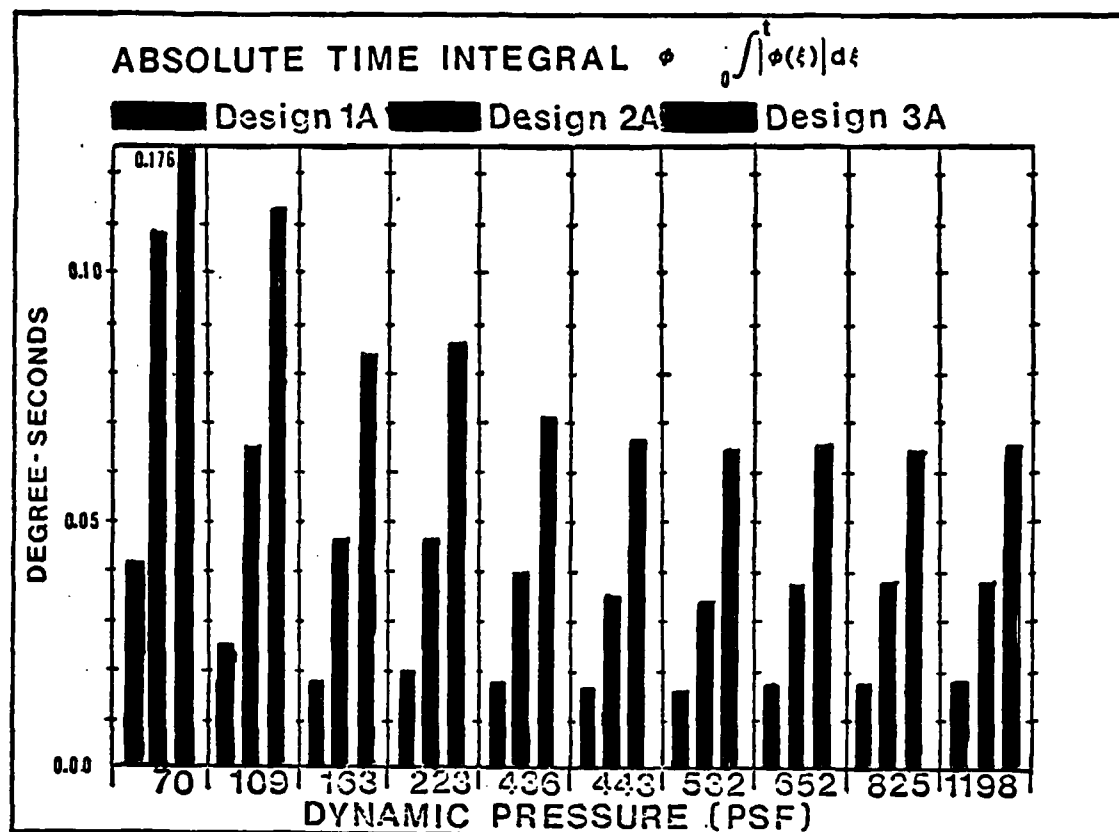


(a)

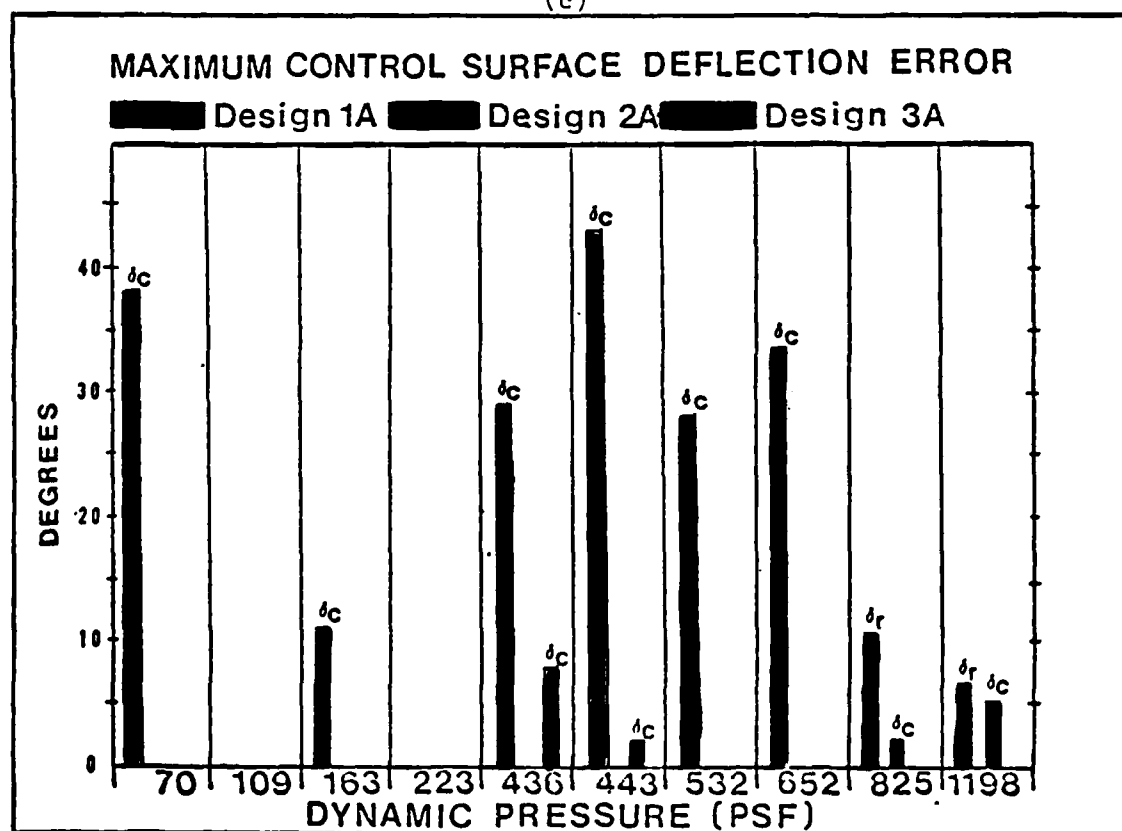


(b)

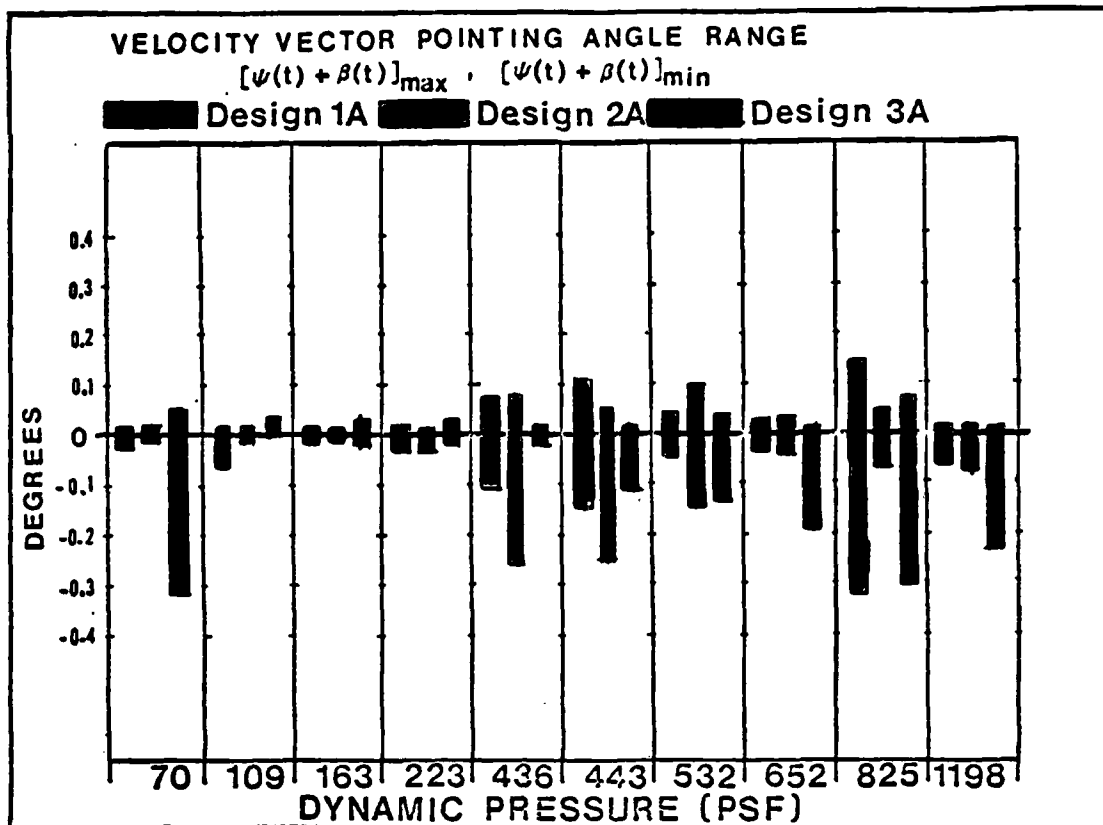
Figure 3-2 Comparison of Figures of Merit



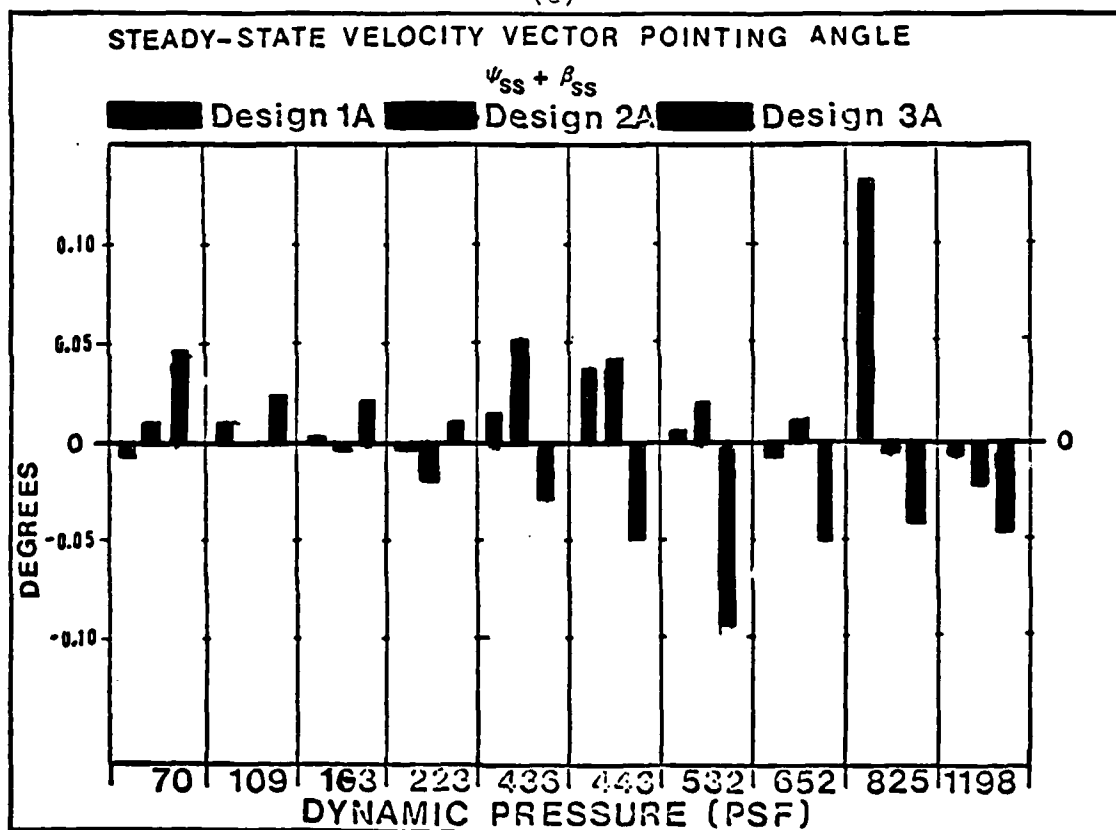
(c)



(d)



(e)



(f)

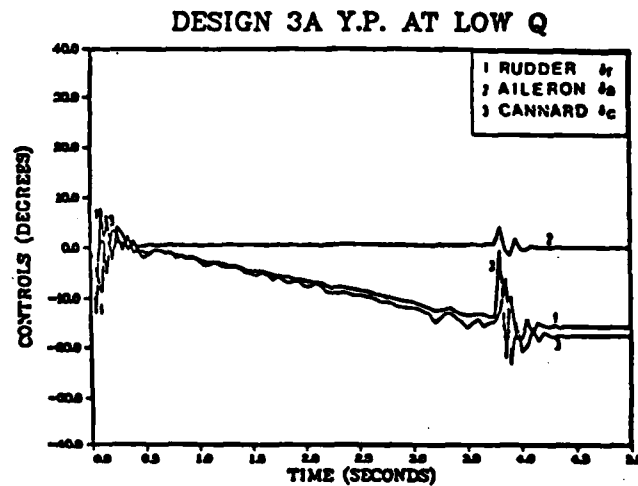
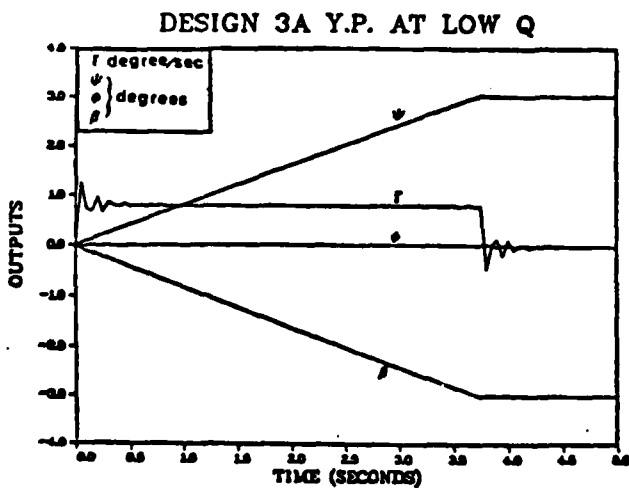
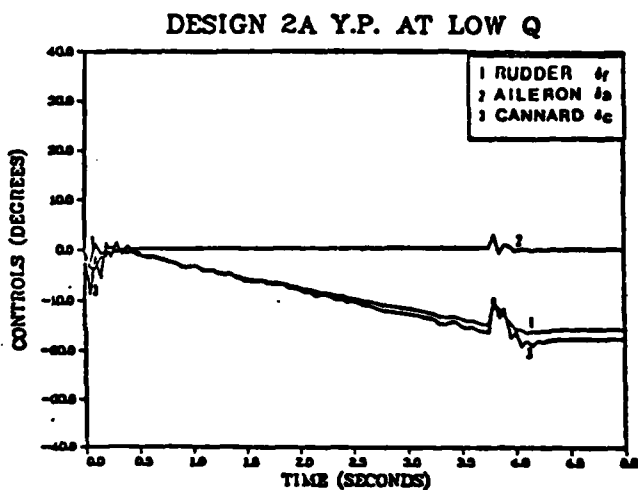
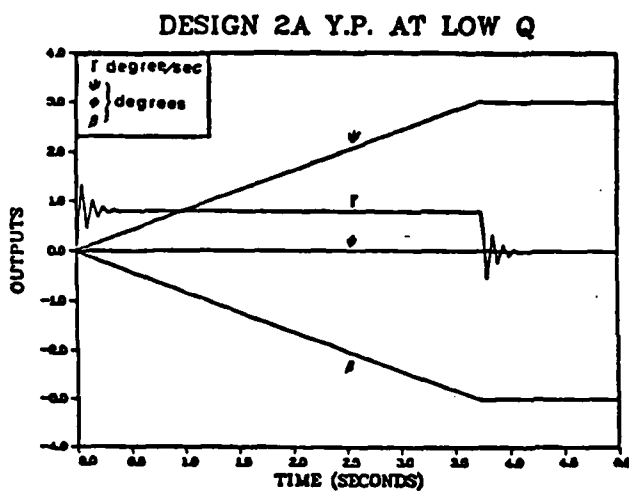
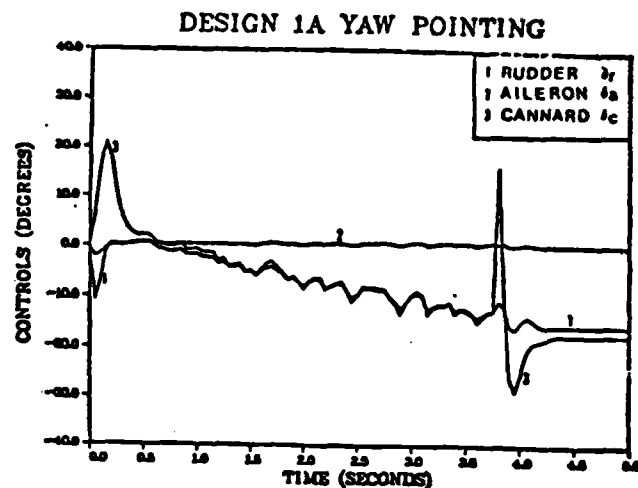
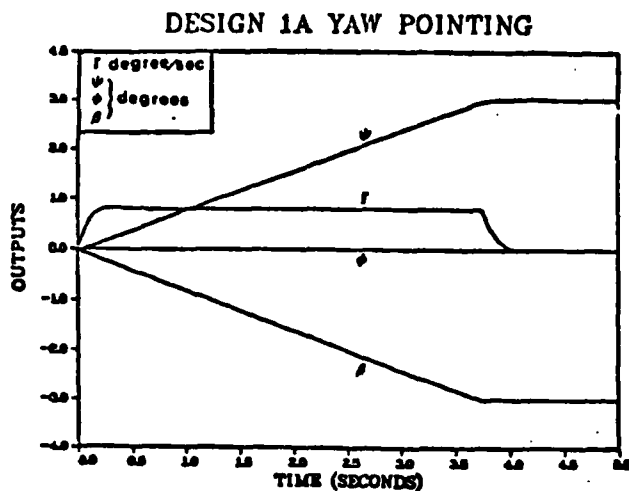


Figure 3-3 Low Q Yaw Pointing

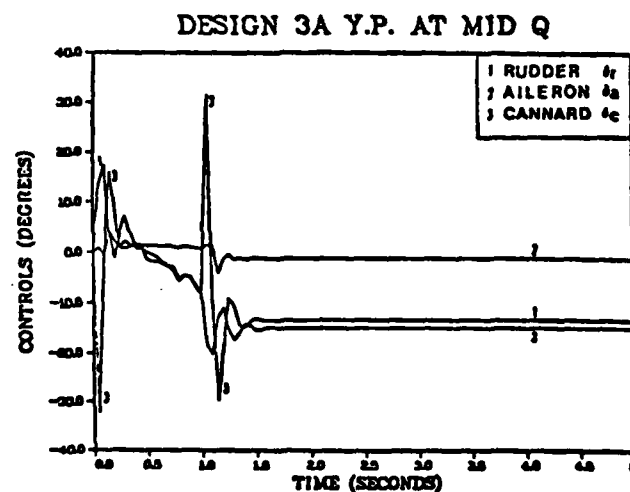
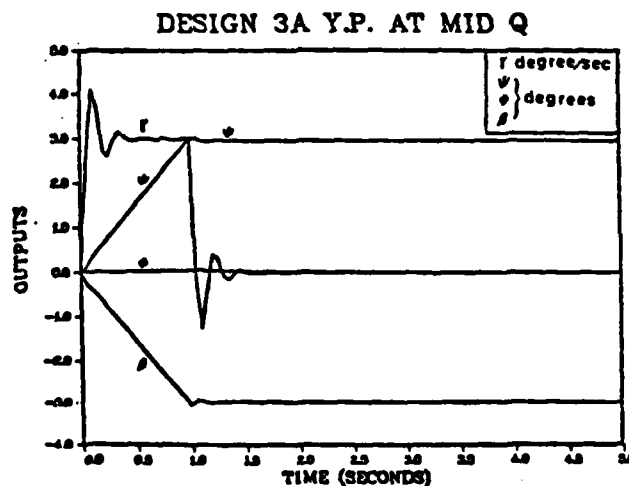
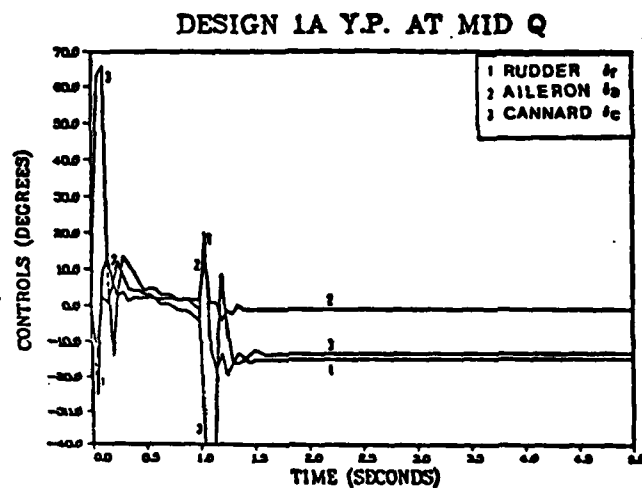
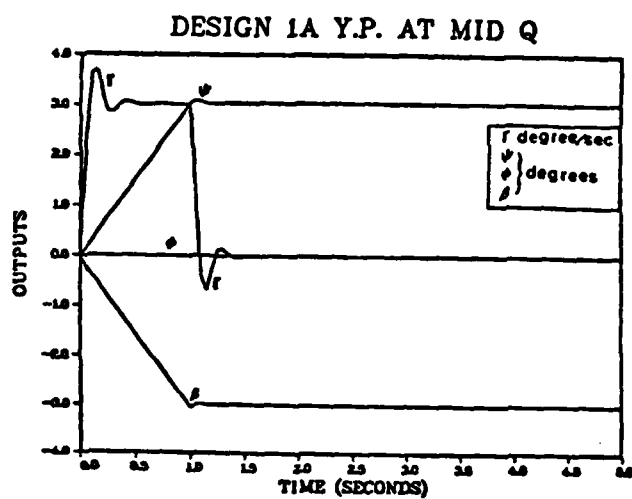
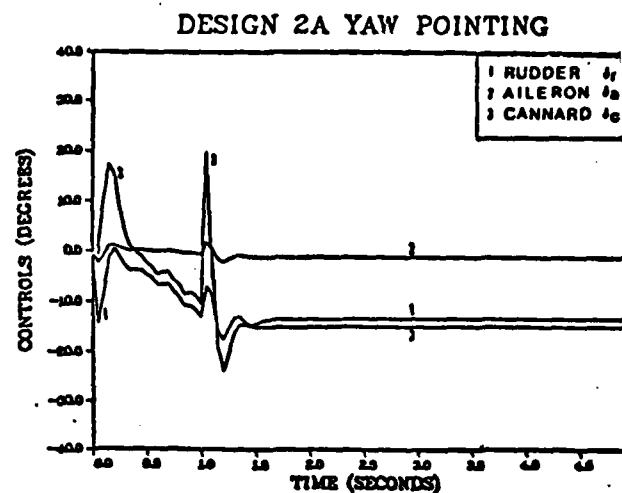
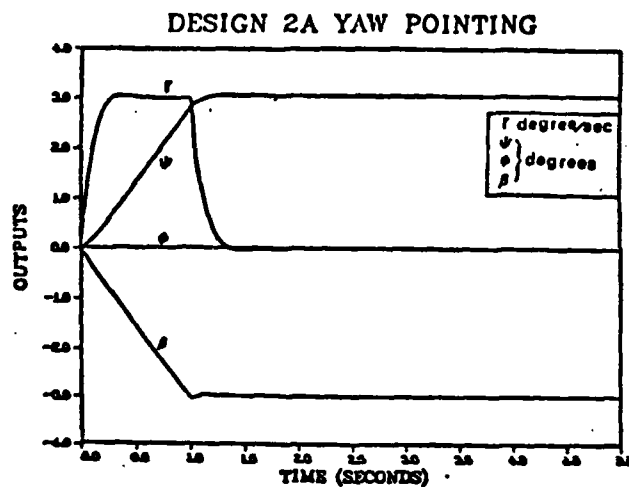


Figure 3-4 Mid Q Yaw Pointing

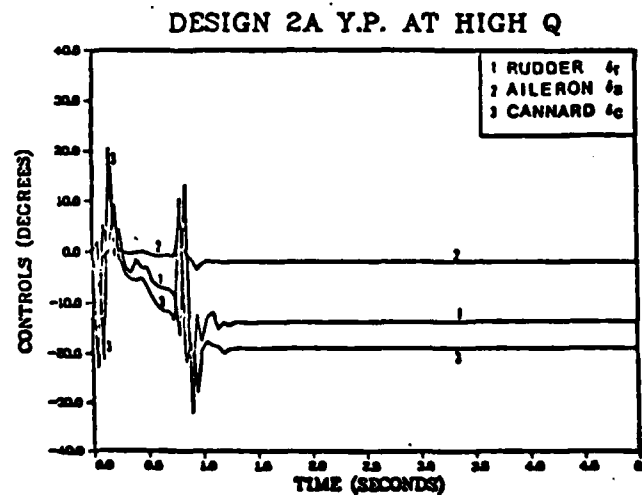
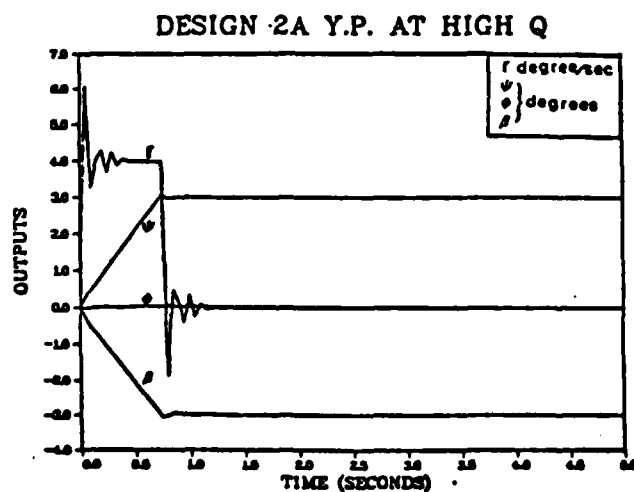
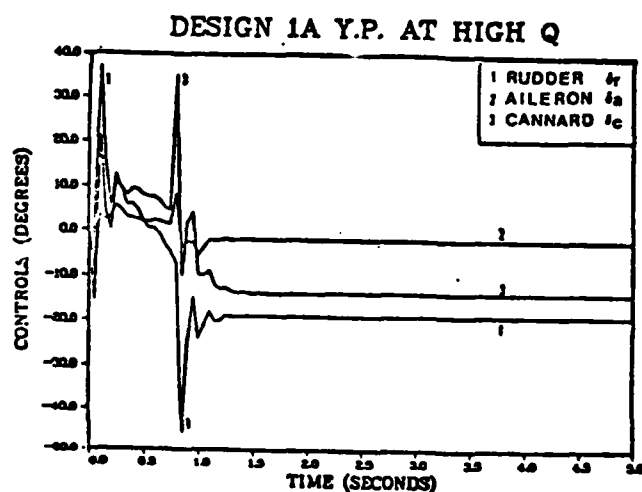
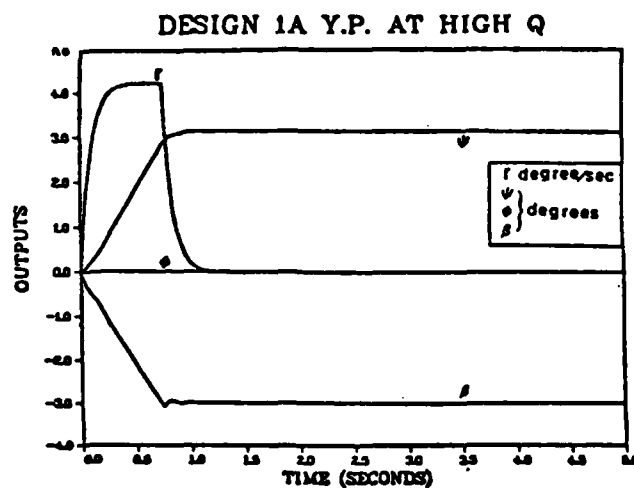
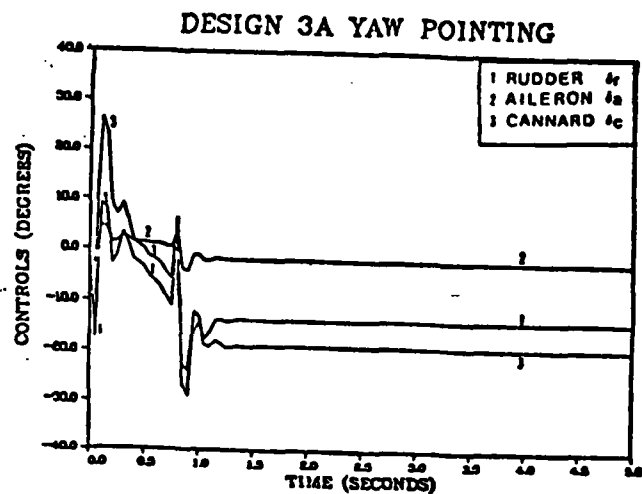
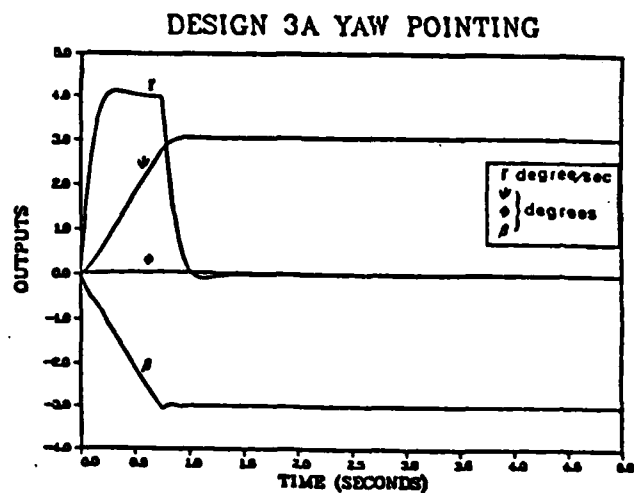


Figure 3-5 High Q Yaw Pointing

The controller design 2A is given commands to perform a wings level turn, a horizontal translation, and a constant altitude coordinated turn. The magnitudes of these commands are adjusted until the maneuver can be performed without commanding the control surfaces beyond their limits.

The input command for a wings level turn is a step input for yaw rate, r , while side-slip angle, β , and roll angle, ϕ , are commanded to zero. For horizontal translation the side-slip angle is commanded as a ramp, till a side-slip angle of four degrees is reached. The roll angle and yaw rate are commanded to zero. For a coordinated turn the relation between yaw rate and roll angle is

$$r = \frac{g}{U} \sin \phi \quad (3.3)$$

while side-slip angle is commanded to zero.

The output and control surface time responses of Design 2A performing the above three maneuvers are shown in Figure 3-6. These responses are compared with those of Figure 3-7 which show the responses of controllers designed to specifically perform the maneuvers at the design 2A flight condition ($Q = 443$). It is easily seen that Design 2A can not perform the maneuvers as well as the design for that specific maneuver. The magnitudes of the maneuvers of Design 2A are much less than the magnitudes of the Mid Q designs. For example the Mid Q design for horizontal translation can successfully command a side-slip angle of four degrees in 1.33 seconds, while the controller of Design 2A took 4.5 seconds. In this respect, the controller is not robust with respect to varying command inputs. The Mid Q

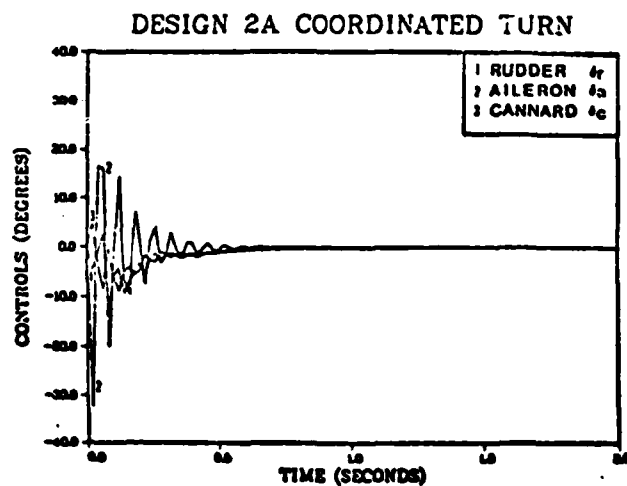
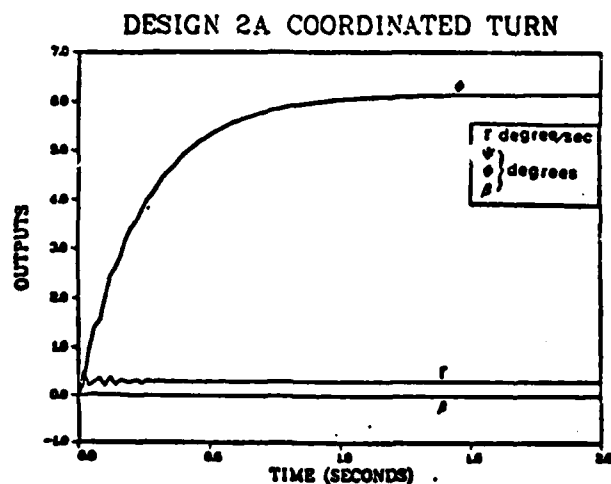
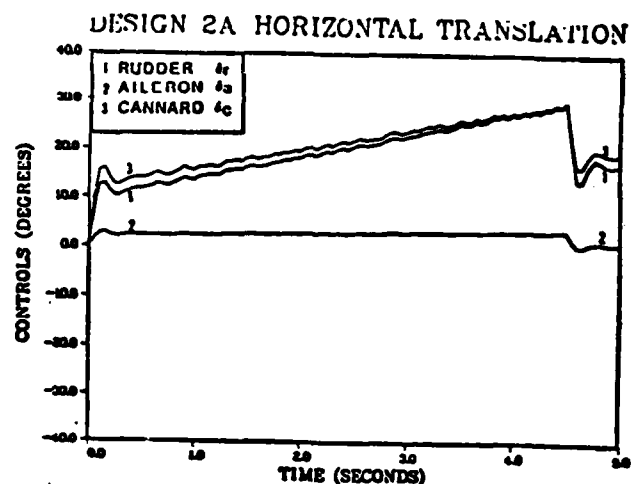
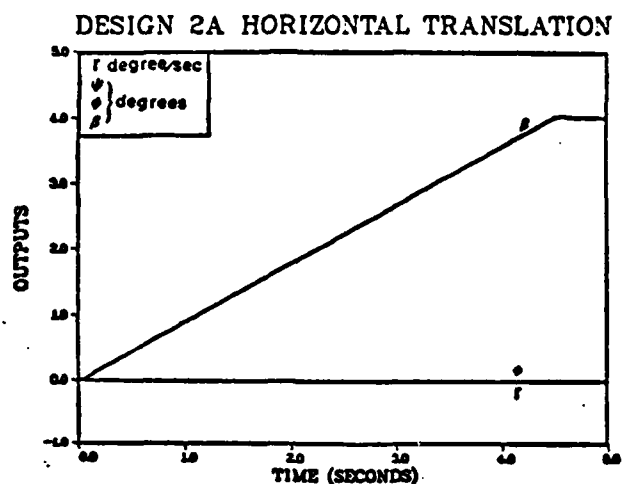
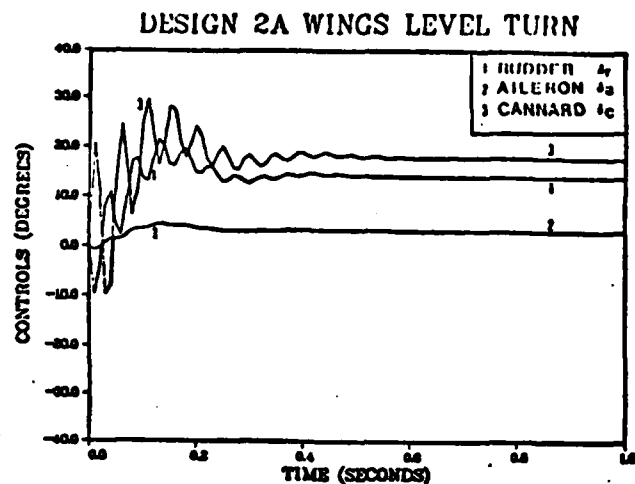
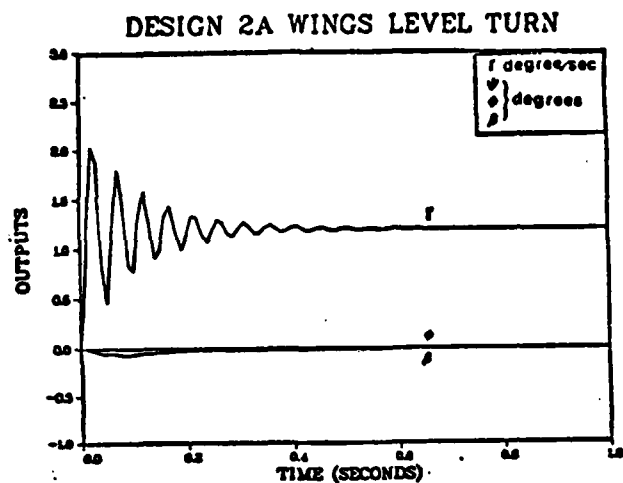


Figure 3-6 Design 2A Additional Maneuvers

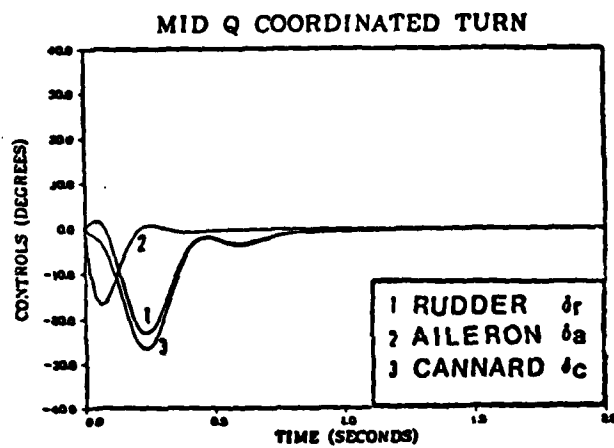
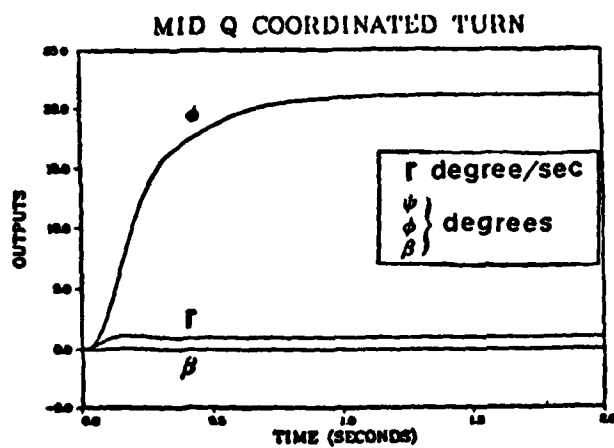
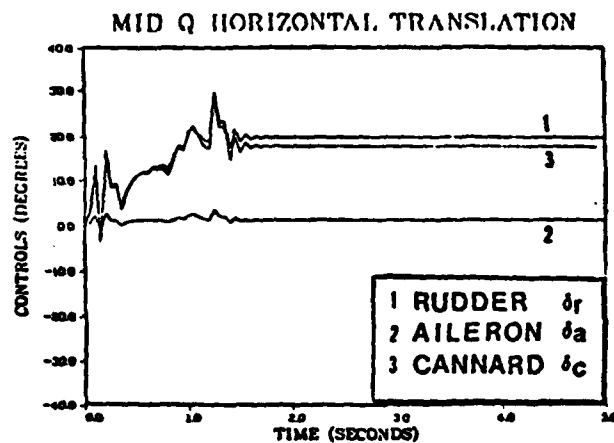
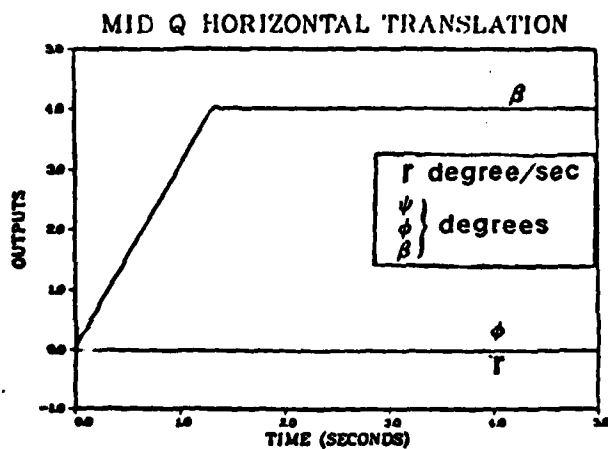
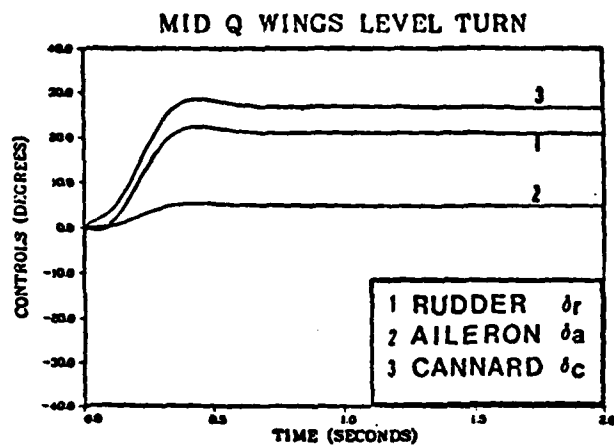
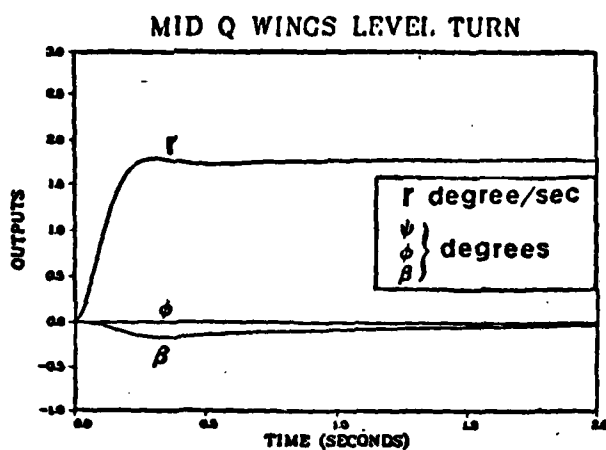


Figure 3-7 Mid Q Additional Maneuvers

Table 3-5 Mid Q designs

MID Q WINGS LEVEL TURN

$$\underline{M} = \begin{bmatrix} 0 \\ 0.25 \\ 0 \end{bmatrix} \quad \underline{\Sigma} = \begin{bmatrix} .5 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & .5 \end{bmatrix} \quad \underline{A}_c = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

$$\underline{K}_0 = \begin{bmatrix} 5.8472 & 0.1321 & -0.0344 \\ 1.3695 & -0.7015 & -0.0023 \\ 7.3259 & -0.1504 & 0.0734 \end{bmatrix} \quad \underline{K}_1 = \underline{K}_0 \quad g = 20$$

	Steady-state Yaw Rate	Peak Overshoot	Settling Time
Design 2A	1.20 deg/sec	2.0 deg/sec	0.5 sec
Mid Q	1.75 deg/sec	1.8 deg/sec	0.5 sec

MID Q HORIZONTAL TRANSLATION

$$\underline{M} = \begin{bmatrix} 0 \\ 0.25 \\ 0 \end{bmatrix} \quad \underline{\Sigma} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 75 \end{bmatrix} \quad \underline{A}_c = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

$$\underline{K}_0 = \begin{bmatrix} 58.4719 & 0.6604 & -5.1539 \\ 13.6946 & -3.5073 & -0.3487 \\ 73.2587 & -0.7521 & 11.0051 \end{bmatrix} \quad \underline{K}_1 = 2\underline{K}_0 \quad g = 40$$

	Steady-state Side-slip angle	Maneuver Time	Maneuver Rate
Design 2A	4.00 degrees	4.50 seconds	0.9 deg/sec
Mid Q	4.00 degrees	1.33 seconds	3.0 deg/sec

MID Q COORDINATED TURN

$$\underline{M} = \begin{bmatrix} 0 \\ 0.25 \\ 0 \end{bmatrix} \quad \underline{\Sigma} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \underline{A}_c = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

$$\underline{K}_0 = \begin{bmatrix} 11.6944 & 0.0132 & -0.0697 \\ 2.7389 & -0.0701 & -0.0046 \\ 14.6517 & -0.0150 & 0.1457 \end{bmatrix} \quad \underline{K}_1 = \underline{K}_0 \quad g = 20$$

	Steady-state Roll Angle	Steady-state Yaw Rate
Design 2A	6.15 degrees	0.30 deg/sec
Mid Q	20.93 degrees	1.00 deg/sec

controller designs and performance comparison summary are shown in Table 3-5.

Actuator Dynamics

As discussed in Chapter 2 there are two methods to add first-order actuator dynamics to the system. The method used in this report is to augment the closed-loop system matrix. This does not cause the size of the measurement matrix to increase. If the actuator dynamics are added by augmenting the open-loop system matrix, the size of the measurement matrix increases from a 3-by-1 matrix to a 3-by-4 matrix. When an attempt is made to add the actuator dynamics this way, the method used for selecting a measurement matrix that both stabilizes the system and diagonalizes the asymptotic transfer function fails. This leaves the selection of a measurement matrix to a trial and error iterative procedure.

The actuator dynamics are significant in the simulations. For comparison the yaw pointing robustness test is repeated for the same three controllers without actuator dynamics in the model. The results of these simulation tests are very misleading. Figures 3-8 through 3-10 show the output and control surface time responses of the three controller designs performing the yaw pointing maneuver at a specific flight condition. When these figures are compared with Figures 3-3 through 3-5 the significance of the actuator dynamics is clearly shown. Note that in all cases the outputs follow the commands exactly, with no overshoots or oscillations. Also note the smooth control surface response curves. The actuator dynamics in the model greatly affect the time responses and should not be ignored

when designing a controller. This same trend is observed when Figure 3-11 is compared with Figure 3-6.

Without actuator dynamics the output responses are much more insensitive to changes in the diagonal elements in the sigma matrix and gain values. This is another reason why the actuator dynamics should not be ignored when designing a controller.

Also studied is the effect of changing the first-order actuator dynamics time constant. The controller is designed with all actuator constants at 20. This design is used in the simulation of a yaw pointing maneuver with the actuator constants varying. The responses are shown in Figures 3-12 through 3-18. The three numbers on the title of the response correspond to the three actuator constants. The respective order is: rudder actuator constant, aileron actuator constant, and cannard actuator constant. For example, (20, 10, 40) represents a rudder constant of 20, a aileron constant of 10, and a cannard constant of 40.

Figure 3-12 shows the responses when all constants are decreased and increased by a factor of two. Figure 3-13 shows the responses when just one constant is decreased by a factor of two. Figure 3-14 shows the responses when just one constant is increased by a factor of two. Figure 3-15 shows the responses when two constants are decreased by a factor of two. Figure 3-16 shows the responses when two constants are increased by a factor of two. Figure 3-17 shows the responses when one constant is decreased and one constant is increased by a factor of two.

This actuator study shows:

1. The aileron has little effect in the yaw pointing maneuver
2. Decreasing the cannard constant increases settling time and increases cannard activity
3. Increasing the rudder constant causes increased rudder activity
4. Increasing both rudder and cannard constants cause the magnitude of surface deflection to decrease.

Overall this study amplifies the importance of actuator dynamics in the model, in that they significantly affect the output and control surface time responses. The actuator time constant is dependent on the physical system used in the aircraft. Varying the constants in the design can be advantageous in a preliminary design study and can be used as a basis for selecting the best actuator.

In this study there is no constraint placed on the rate of control surface deflection. As a result the control surface deflection rates were often unreasonably fast. Any further study should include this constraint.

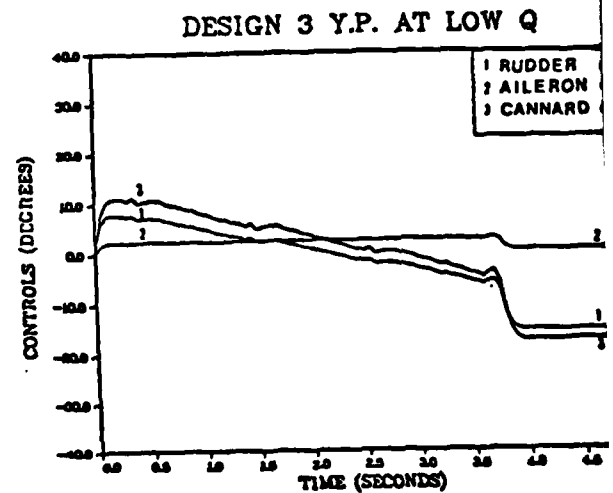
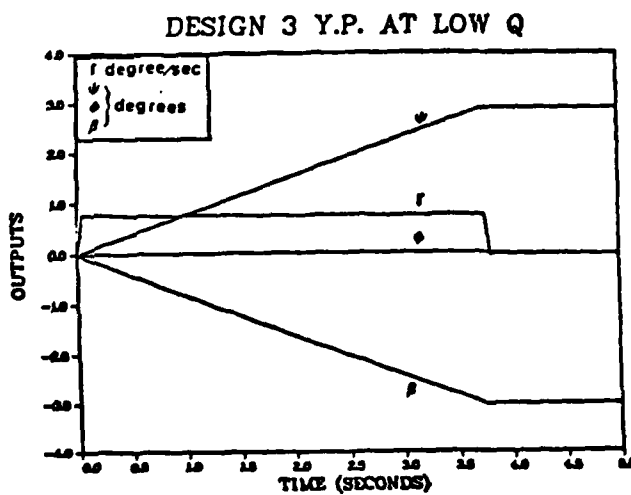
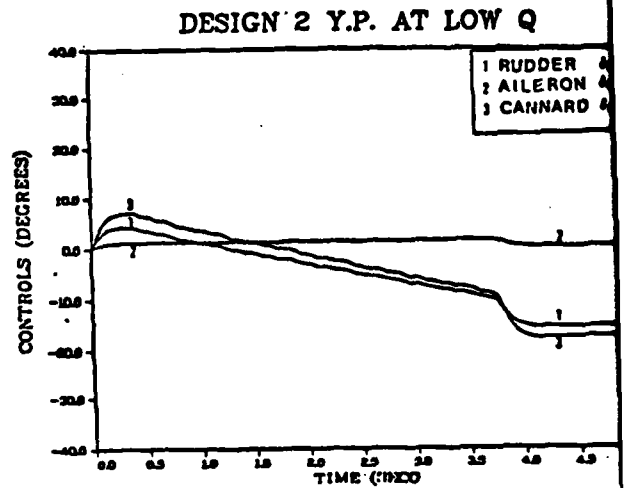
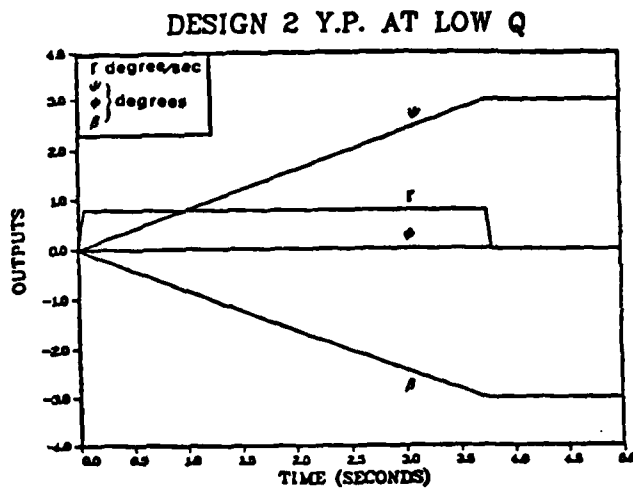
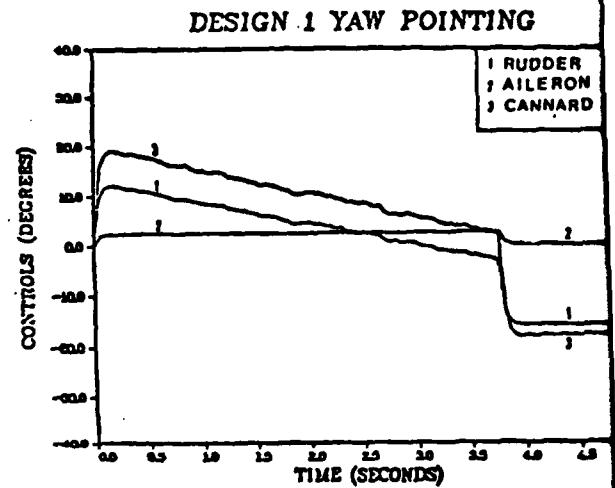
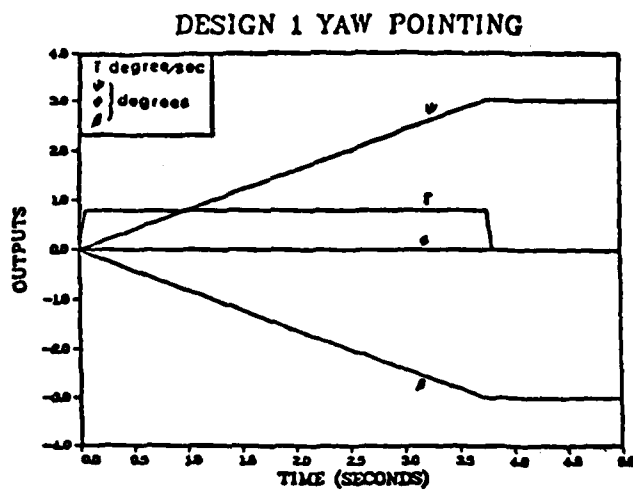


Figure 3-8 Low Q Yaw Pointing Without Actuators

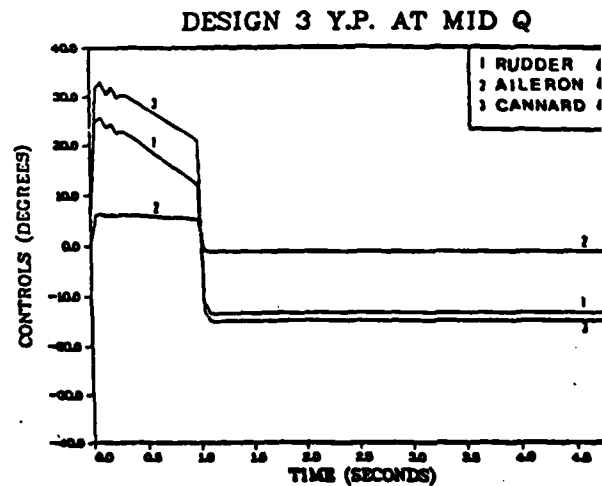
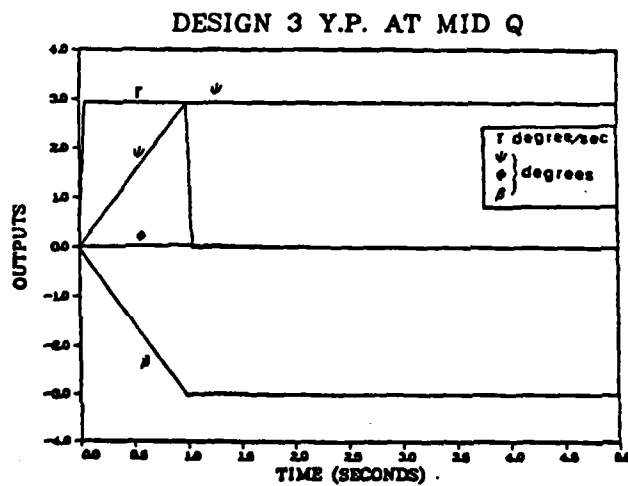
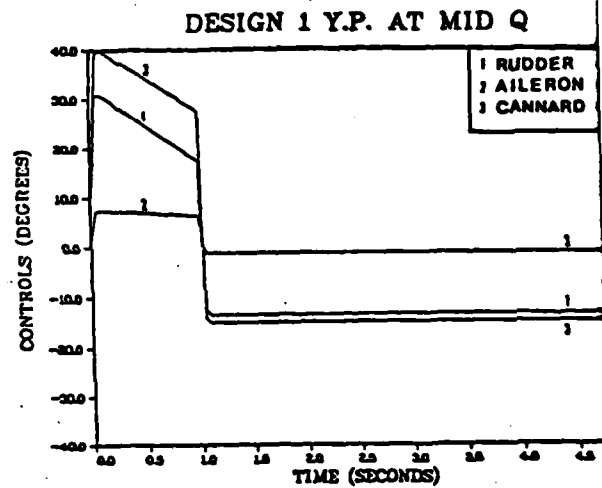
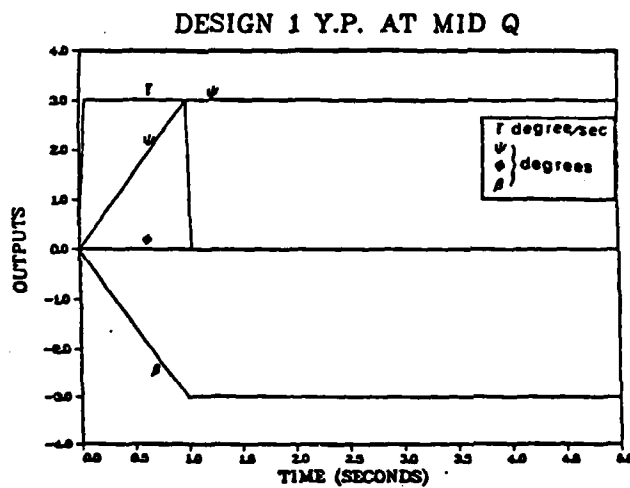
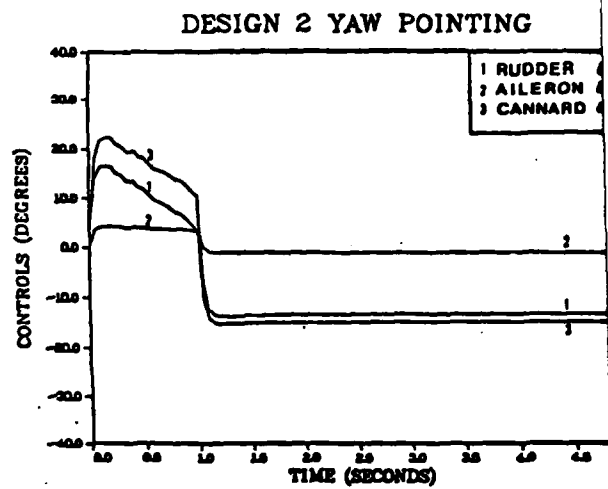
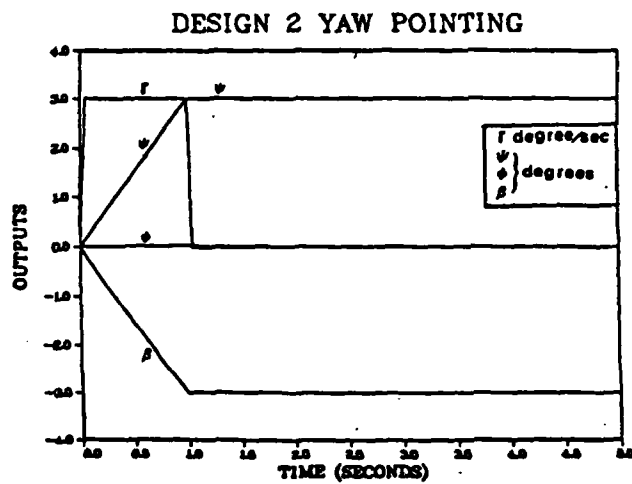


Figure 3-9 Mid Q Yaw Pointing Without Actuators

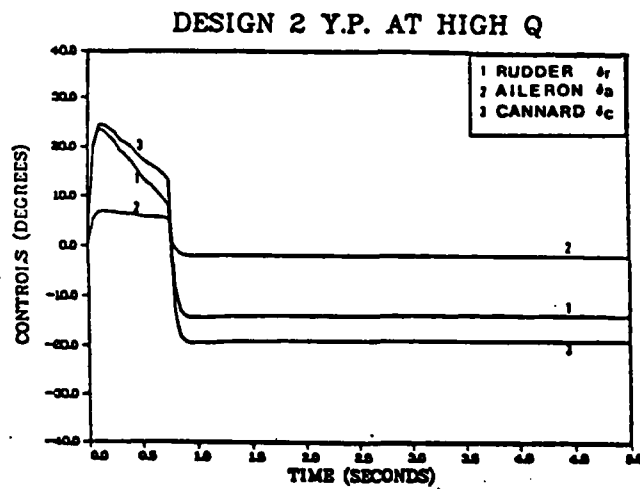
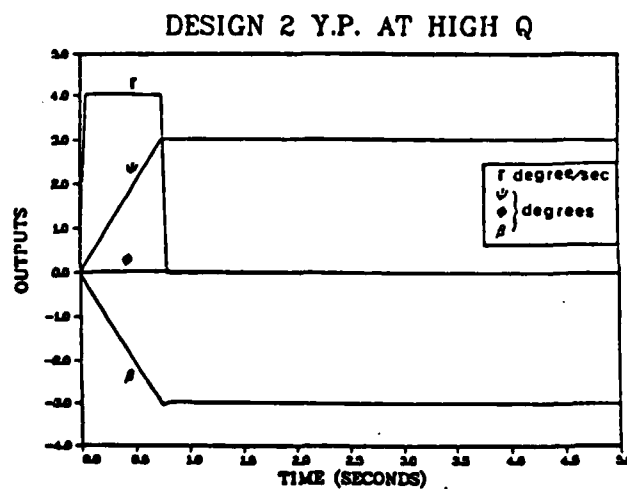
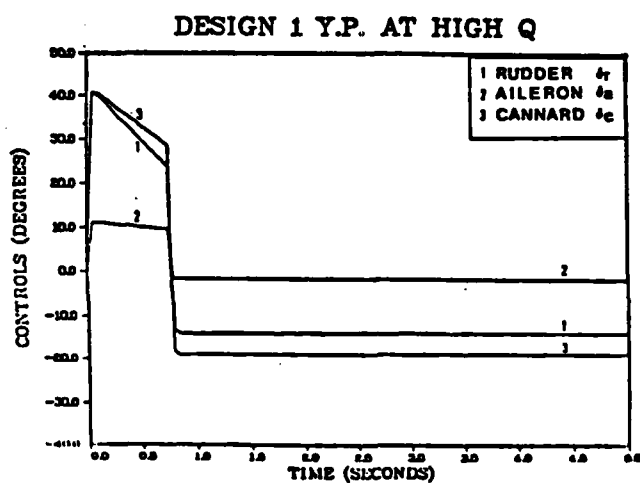
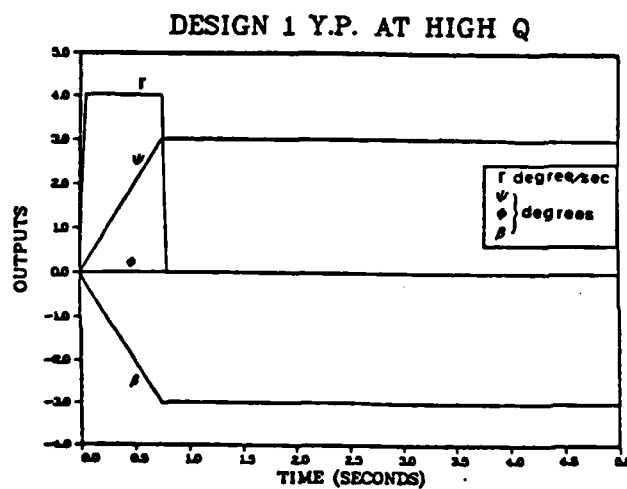
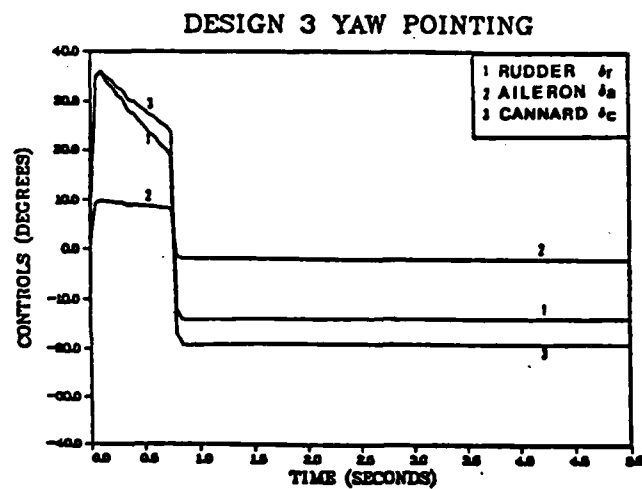
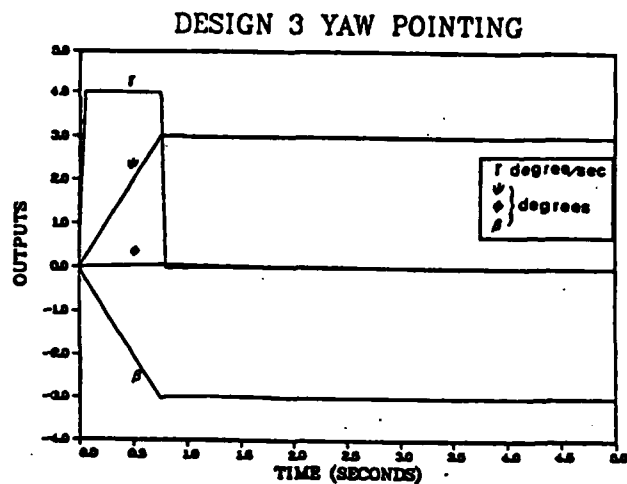


Figure 3-10 High Q Yaw Pointing Without Actuators

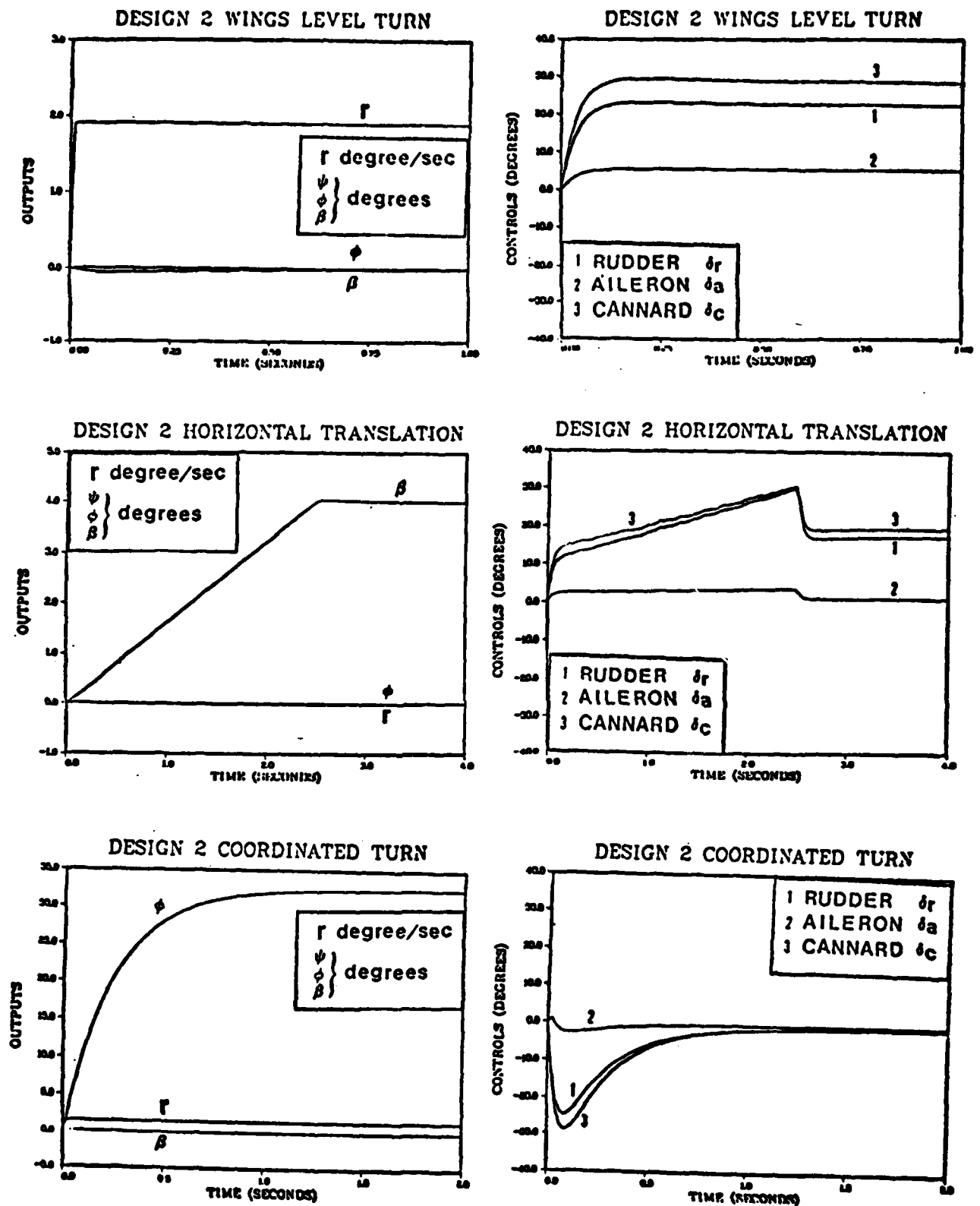


Figure 3-11 Design 2 Additional Maneuvers

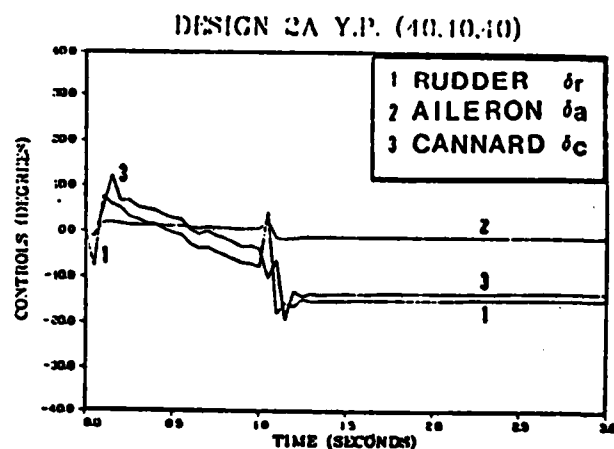
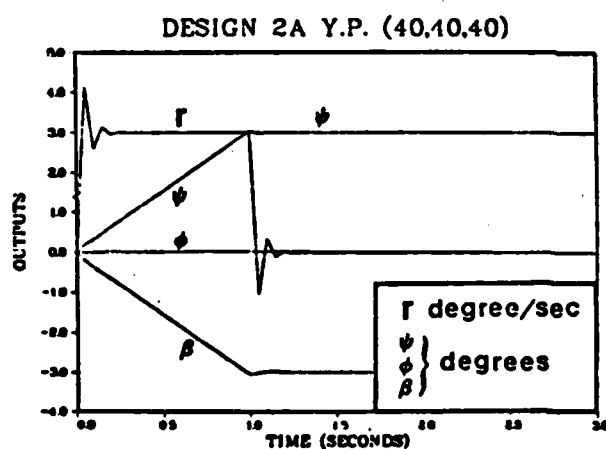
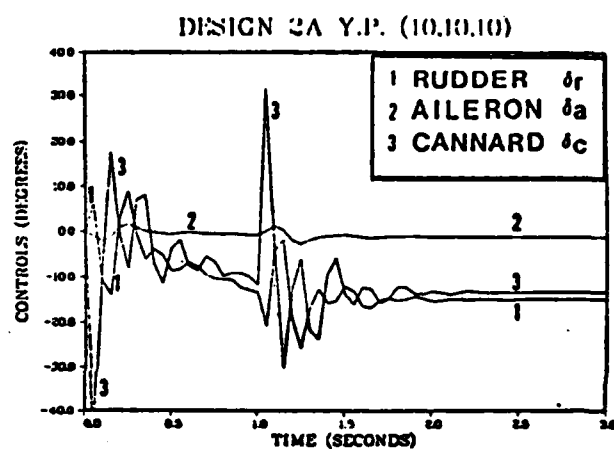
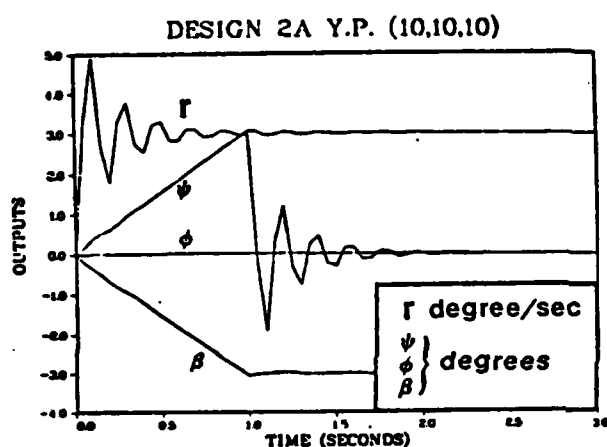
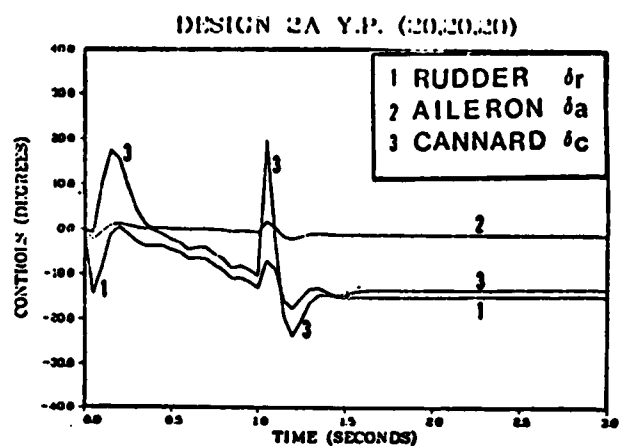
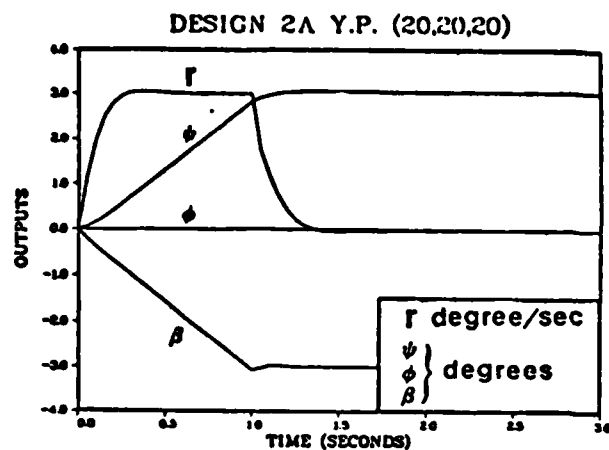


Figure 3-12 All Constants Changed

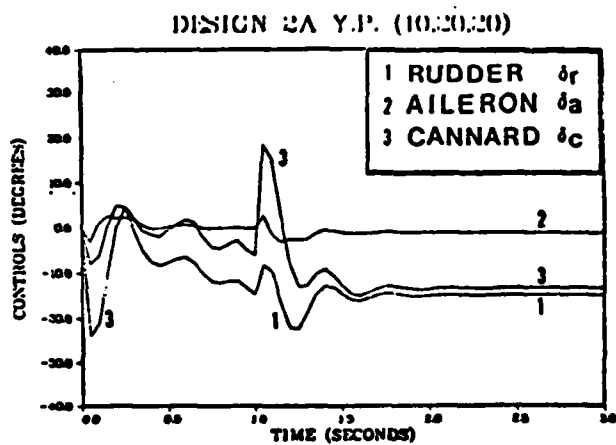
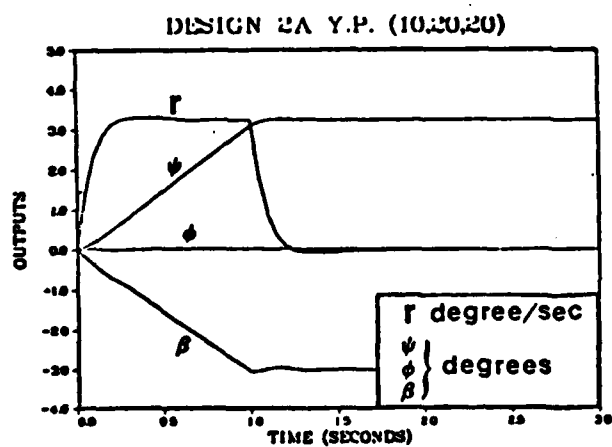
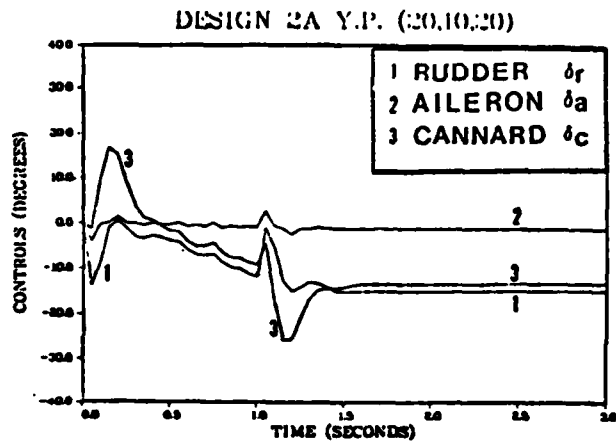
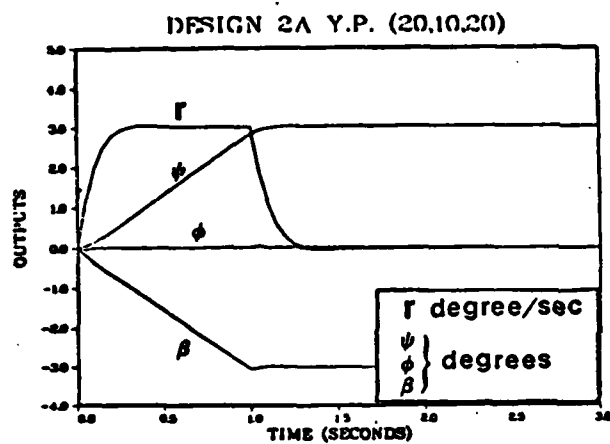
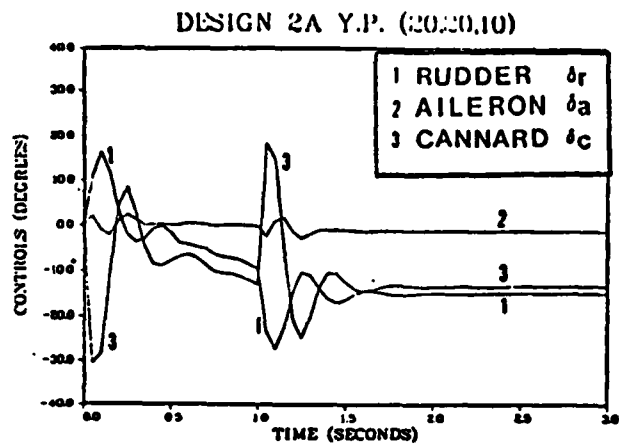
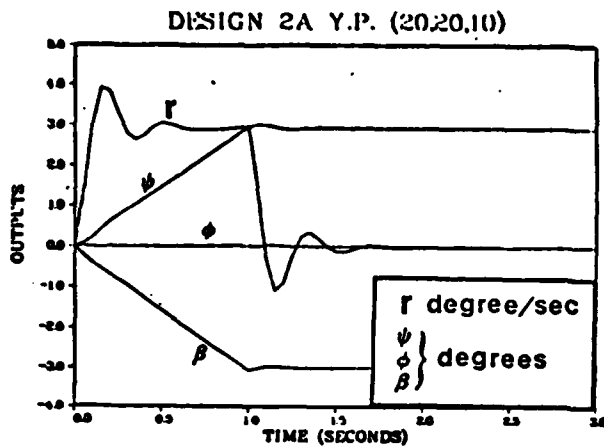


Figure 3-13 One Constant Decreased

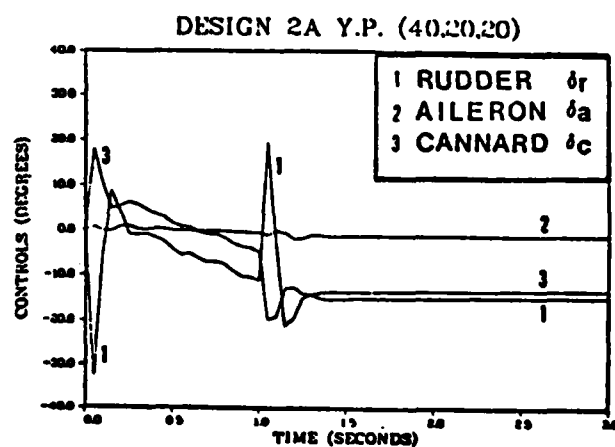
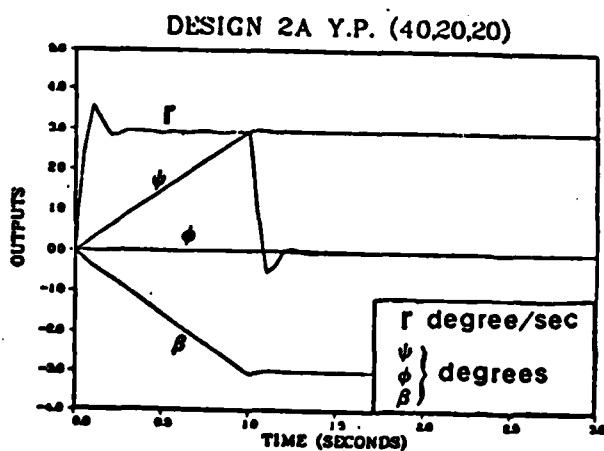
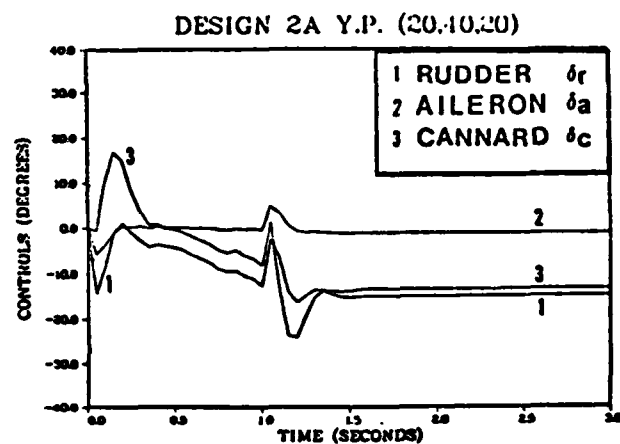
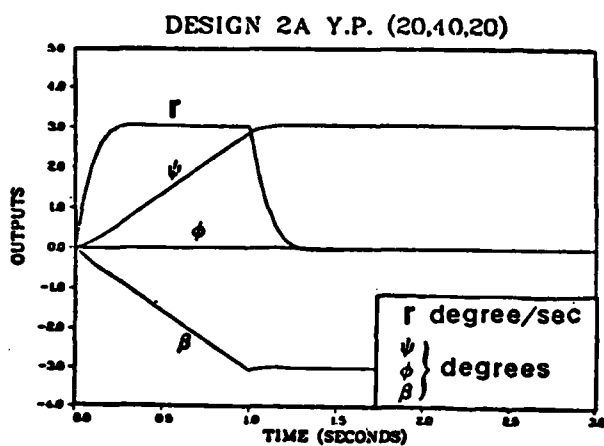
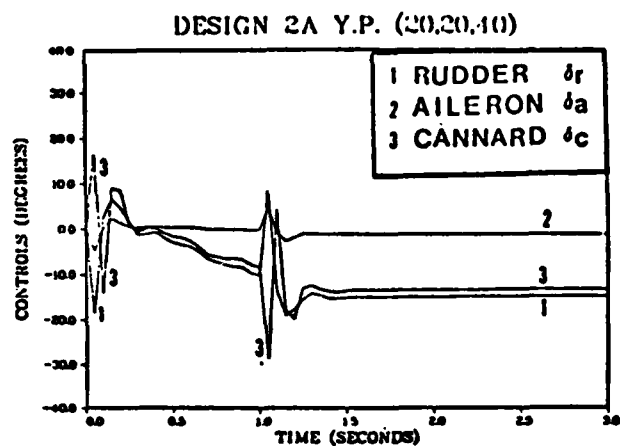
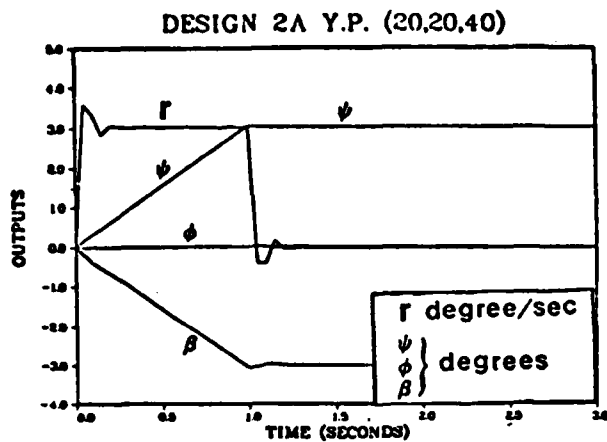


Figure 3-14 One Constant Increased

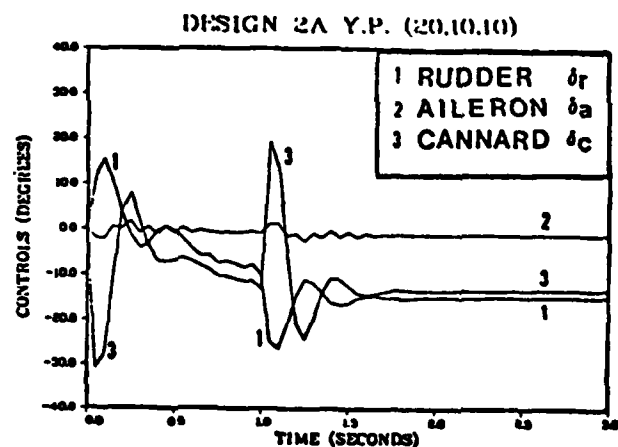
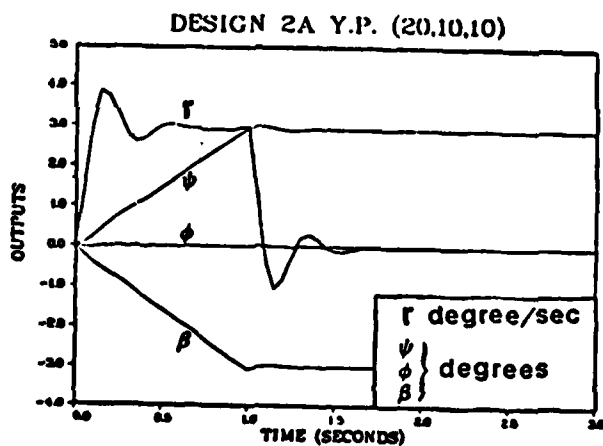
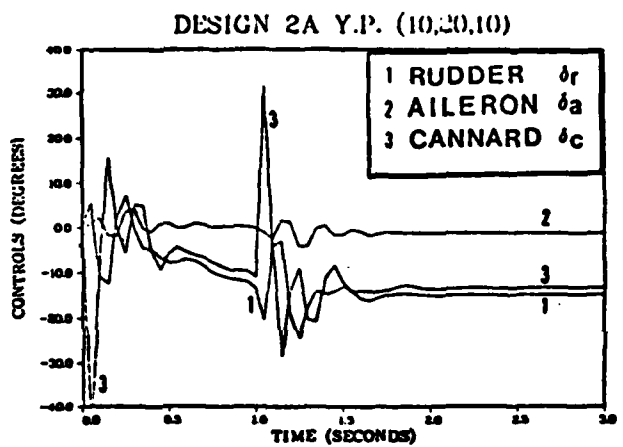
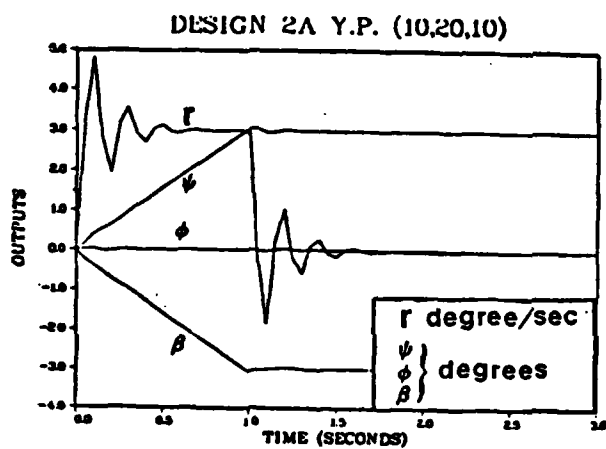
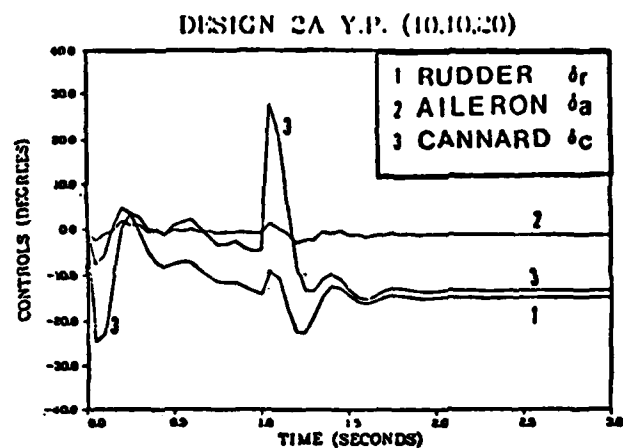
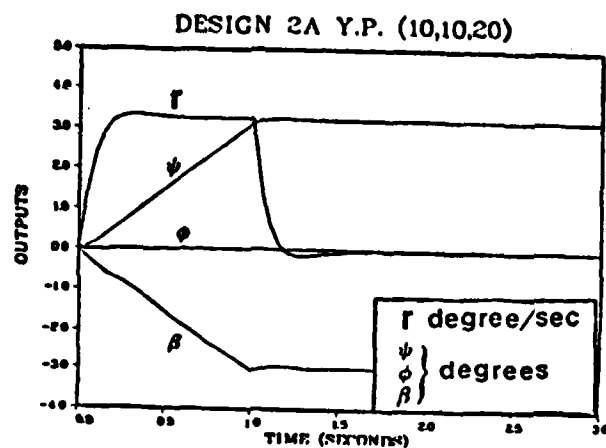


Figure 3-15 Two Constants Decreased

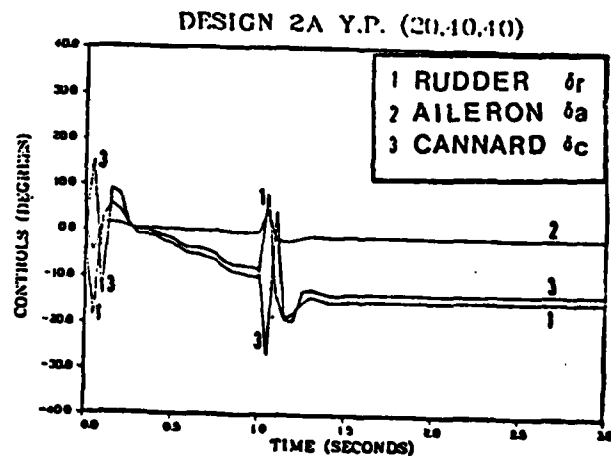
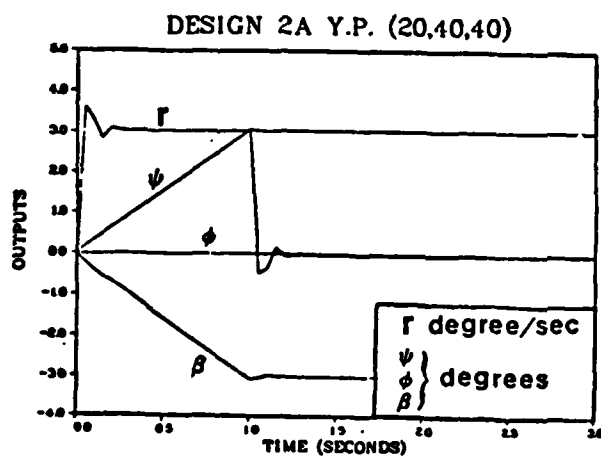
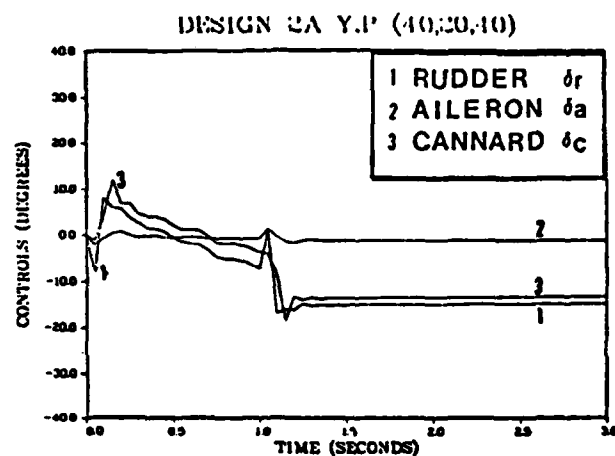
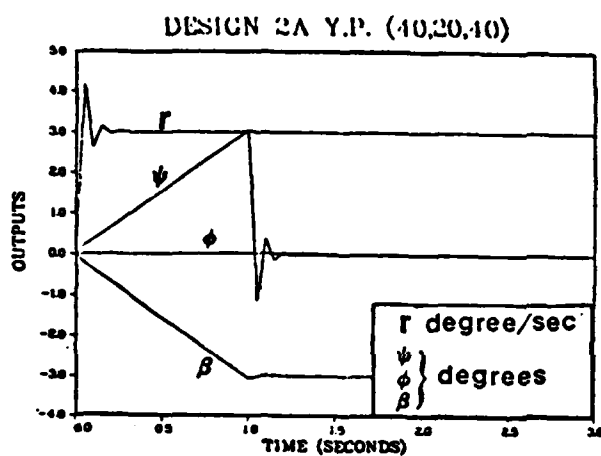
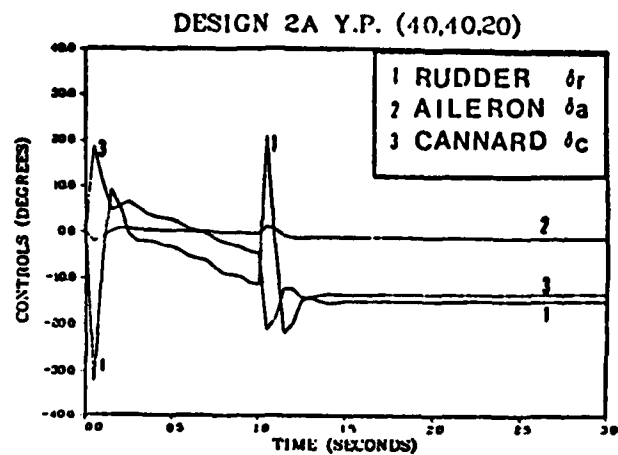
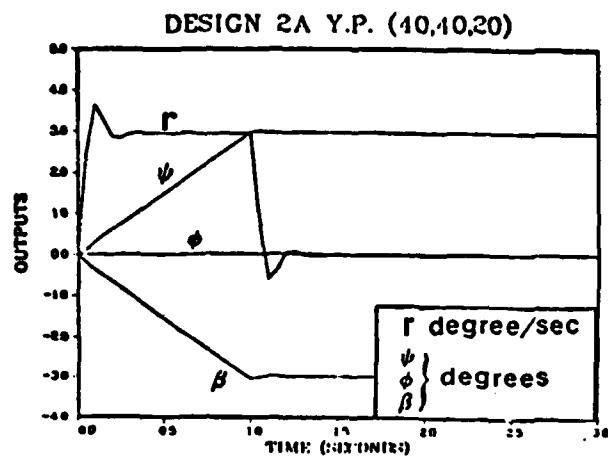


Figure 3-16 Two Constants Increased

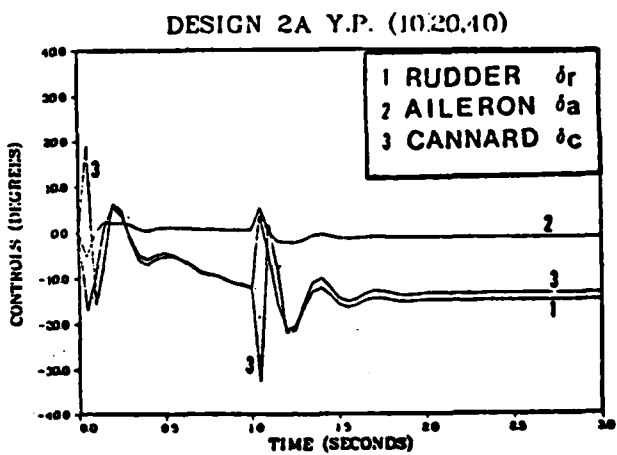
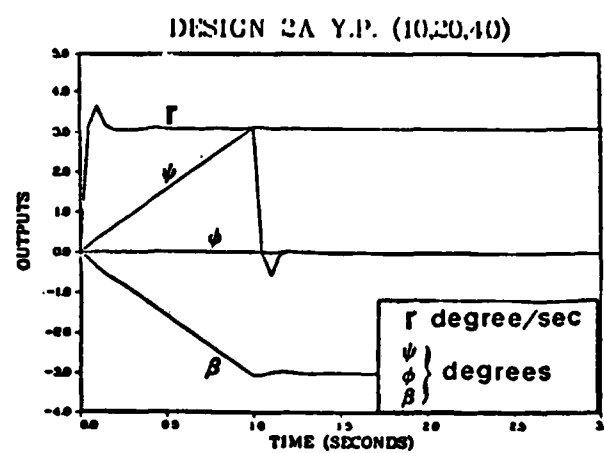
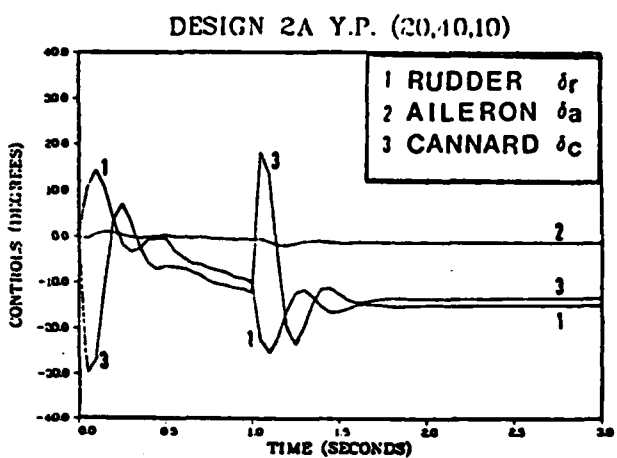
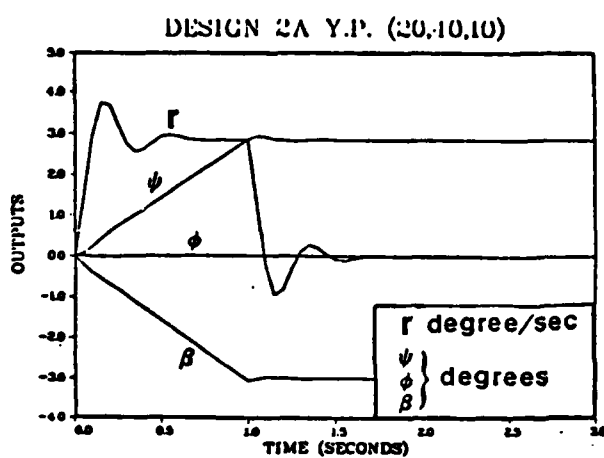
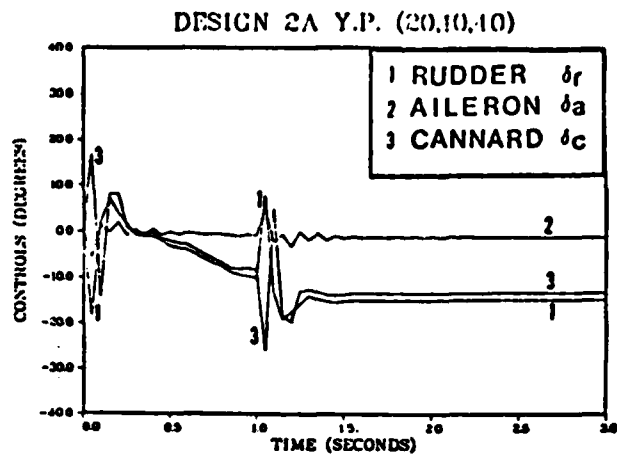
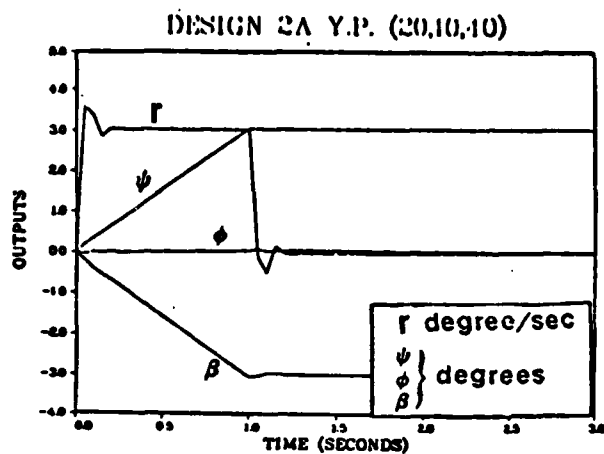
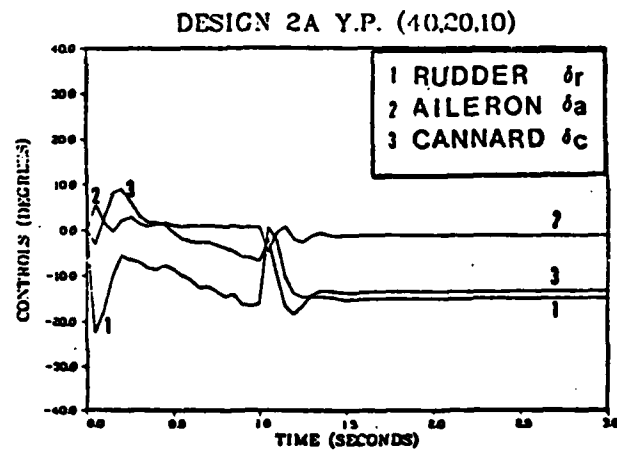
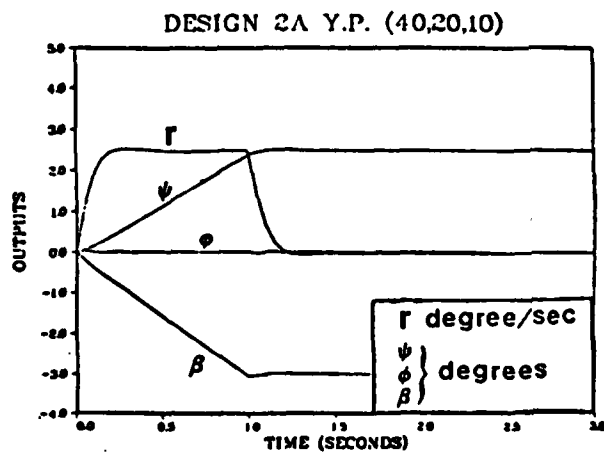
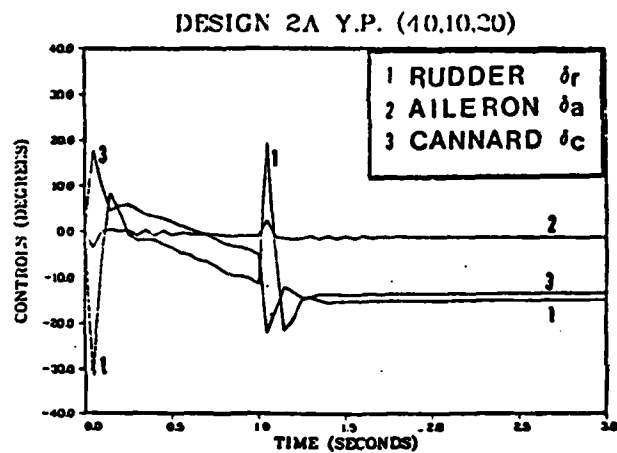
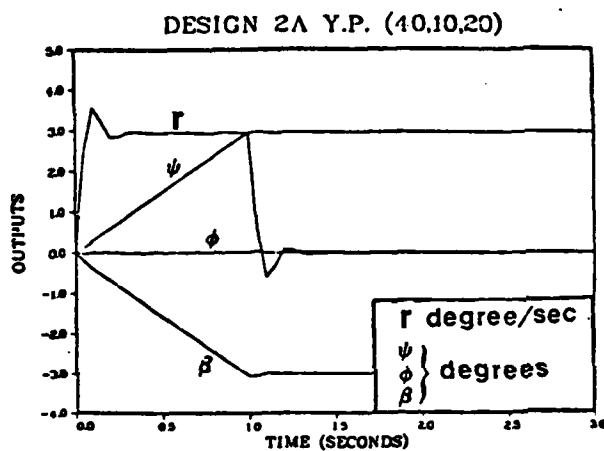
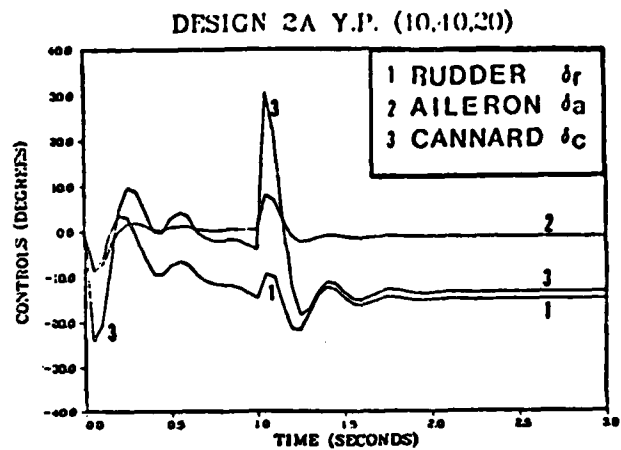
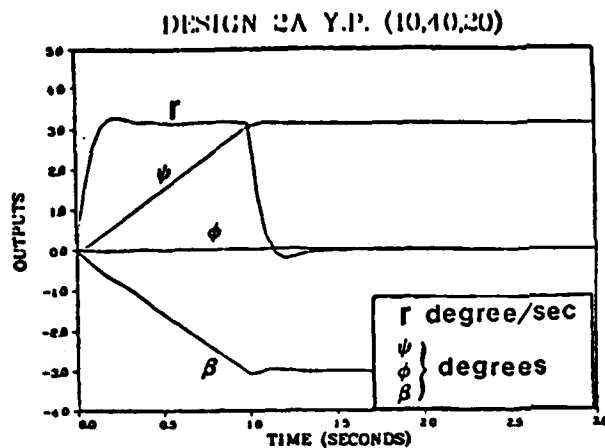


Figure 3-17 One Constant Decreased, One Constant Increased



IV. Conclusions and Recommendations

Conclusions

Major conclusions are based on the results of this report. They are first presented in Chapter III and are now summarized.

The design method of Dr Porter is easy to apply and does produce controller designs to perform desired maneuvers. This method is relatively new and promising. Research and applications of this method should continue.

The controllers are robust with respect to varying flight conditions. They are not robust with respect to varying maneuver commands. The controller designed at a medium dynamic pressure flight condition produced the more robust controller design.

Actuator dynamics significantly affect the output and control surface time responses. Simplified models without actuator dynamics included provide misleading results. Actuator dynamics should not be ignored in the design of flight controllers with this method.

Recommendations

This study followed the study performed by Lt Ridgely (Ref 10). His study examines the longitudinal modes of AFTI. This study examines the lateral-directional modes of AFTI. The next study should combine the two modes and examine the response of the total aircraft when performing various maneuvers.

Further robustness examination should be conducted to find if a controller can be designed that is both robust with respect to flight condition and maneuver commands. If so, what parameters affect robustness.

The computer program used for this study is limited when compared with the computer program developed for the design of digital controllers using Porter's method, MULTI (Ref 13). MULTI should be augmented to include the design of analogue controllers.

The precision of program ZERO should be increased from single precision to double precision to increase the program's accuracy. Also the subroutines in ZERO that check the rank of the system matrices should be changed so that complex system zeros can also be tested.

Bibliography

1. Fortmann, T. and Hitz, K. 1977. An Introduction to Linear Control Systems. New York: Marcel Dekker Inc.
2. Kirk, D. 1970. Optimal Control Theory. New Jersey: Prentice-Hall.
3. Laub, A.J. and Moore, B.C. 1977. Calculation of Transmission Zeros Using QZ Techniques. Proceedings of the 1977 IEEE Conference on Decision and Control.
4. MacFarlane, A.G.J. and Karcnias, N. 1976. Poles and Zeros of Linear Multivariable Systems. International Journal of Control. 24 no.1 :33-74.
5. McRuer, Ashkenas, and Graham. 1973. Aircraft Dynamics and Automatic Control. New Jersey: Princeton University Press.
6. Moler, C.B. and Stewart, G.W. 1973. An Algorithm for Generalized Matrix Eigenvalue Problems. SIAM Journal of Numerical Analysis. 24 no.10: 241-255.
7. Porter, B. 1981. Design of High-Performance Tracking Systems. AFWAL/FIGL. Wright-Patterson AFB. Ohio.
8. Porter, B. 1979. Computation of the Zeros of Linear Multivariable Systems. International Journal of System Science. 10 no.12: 1427-1432.
9. Ridgely, Banda, D'Azzo. 1981. Decoupling of High-Gain Multivariable Tracking Systems. AFWAL/FIGCA. Wright-Patterson AFB. Ohio.
10. Ridgely, D. 1981. Use of Entire Eigenstructure Assignment with High-Gain Error-Actuated Flight Control Systems. Masters Thesis. Air Force Institute of Technology. Wright-Patterson AFB. Ohio.

Bibliography

11. Silverthorn, J. and Reid, J.G. 1980. Computation of the Subspaces for Entire Eigensructure Assignment Via Singular Value Decomposition. Proceedings of the 19th IEEE onference on Decision and Control. 12 no.10: 1206-1207.
12. Wind Tunnel Data, AFTI/F-16. AFWAL/FII. Wright-Patterson AFB. Ohio
13. Porter, D. 1981. Design and Analysis of Multivariable Digital Controller for the A-7D Digital II Aircraft and the Development of an Interactive Computer Design Program. Masters Thesis. Air Force Institute of Technology. Wright-Patterson AFB. Ohio.

Appendix A-1. Development of the Aircraft Equations of Motion

As stated in the introduction, only the lateral-directional, decoupled maneuvers are analyzed in this report. Hence, only the lateral-directional set of equations are developed here. The airframe model is a conventional, small perturbation set of equations, which are linearized about straight and level flight ($\theta_0 = \alpha_0$). All four maneuvers analyzed are described by body axis variables, therefore, the equations of motion are body axis equations. The linearized equations in the Laplace domain are (Ref 5):

$$\begin{aligned} s v + U_0 r - w_0 p - g \cos \theta_0 \phi \\ = Y_v v + Y_p p + Y_r r + \Sigma Y_\delta \delta \end{aligned} \quad (A.1)$$

$$s p = L'_v v + L'_p p + L'_r r + \Sigma L'_\delta \delta \quad (A.2)$$

$$s r = N'_v v + N'_p p + N'_r r + \Sigma N'_\delta \delta \quad (A.3)$$

$$s \phi = p + r \tan \theta_0 \quad (A.4)$$

These equations are transformed to the time domain, and arranged to have the matrix form

$$\dot{\underline{x}} = \underline{A} \underline{x} + \underline{B} \underline{u} \quad (A.5)$$

where

$$\underline{x}^T = [\phi, \beta, p, r] \quad (A.6)$$

$$\underline{u}^T = [\delta_r, \delta_a, \delta_c] \quad (A.7)$$

$$\underline{A} = \begin{bmatrix} 0 & , & 0 & , & 1 & , & \tan \theta_o \\ g \cos \theta_o / u & , & Y_v & , & (Y_p + w_o) / u & , & (Y_r - u_o) / u \\ 0 & , & L'_\beta & , & L/p & , & L'_r \\ 0 & , & N'_\beta & , & N'_p & , & N'_r \end{bmatrix} \quad (A.8)$$

$$\underline{B} = \begin{bmatrix} 0 & , & 0 & , & 0 \\ Y^*_{\delta r} & , & Y^*_{\delta a} & , & Y^*_{\delta c} \\ L'_{\delta r} & , & L'_{\delta a} & , & L'_{\delta c} \\ N'_{\delta r} & , & N'_{\delta a} & , & N'_{\delta c} \end{bmatrix} \quad (A.9)$$

The AFIT-16 data (Ref 12) is not consistent with eqs. A.1 through A.9, since the stability coefficients are in non-dimensional form and are for the stability axis system. Also some of these stability coefficients have units of per radian, while others have units of per degree. The moments of inertia given are in dimensional, body axis form.

The steps used to transform the given data to data consistent with eqs. A.1 through A.9 are:

1. Transform all stability coefficients with units of per degree to per radian.
2. Transform the moments of inertias to the stability axis system.
3. Dimensionalize the stability coefficients.
4. Transform the L and N dimensional stability coefficients to prime form.

5. Transform the dimensional stability coefficients to the body axis sytem.

Ten different flight conditions are used in this study, requiring ten sets of equations of motion. The data for these flight conditions are listed in Table A-1. A computer program is written to perform all of the above transformations. Use of the computer program guarantees that the data transformations are applied consistently. An example and program listing are in Appendix A-2.

The stability coefficients, as originally given, in non-dimensional stability axis form are listed in Table A-2. The dimensional transformation of step 1 consists of multiplication by $180^\circ/\pi$. The transformation of step 2 makes the multiplication factors of steps 3 and 4 consistent, that is, inertias and stability coefficients are all in the stability axis system. The transformation from body axis to stability axis is accomplished by using the following equations:

$$\begin{aligned} (I_{xx})_s &= (I_{xx})_b \cos^2 \theta_o - 2(I_{xz})_b \cos \theta_o \sin \theta_o \\ &\quad + (I_{zz})_b \sin^2 \theta_o \end{aligned} \quad (A.10)$$

$$\begin{aligned} (I_{zz})_s &= (I_{zz})_b \cos^2 \theta_o + 2(I_{xz})_b \cos \theta_o \sin \theta_o \\ &\quad + (I_{xx})_b \sin^2 \theta_o \end{aligned} \quad (A.11)$$

$$\begin{aligned} (I_{xz})_s &= ((I_{xx})_b - (I_{zz})_b) \cos \theta_o \sin \theta_o \\ &\quad + (I_{xz})_b (\cos^2 \theta_o - \sin^2 \theta_o) \end{aligned} \quad (A.12)$$

TABLE A-1

Flight Condition Data

Dynamic Pressure, Q (lb/ft ²)	Altitude (feet)	Density, ρ (slug/ft ³)	Sonic Velocity (ft/sec)	Mach	Velocity, U (ft/sec)	Angle of Attack, α_o (degree)
70	30,000	0.000889	994.6	0.4	397.84	11.400
109	20,000	0.001266	1036.8	0.4	414.72	6.952
163	10,000	0.001755	1077.4	0.4	430.96	4.552
223	40,000	0.000585	968.1	0.9	871.29	3.692
436	20,000	0.001266	1036.8	0.8	829.44	2.285
443	5,000	0.002048	1097.1	0.6	658.26	1.824
532	30	0.002377	1116.4	0.6	669.84	1.547
652	10,000	0.001755	1077.4	0.8	861.92	1.733
825	10,000	0.001755	1077.4	0.9	969.66	1.462
1198	30	0.002377	1116.4	0.9	1004.76	1.247

The multiplication factors required to dimensionalize the stability coefficients, step 3, are shown in Table A-3. The transformation of step 4 is accomplished by using the equations

$$L'_i = \frac{L_i + ((I_{xz})_s / (I_{xx})_s) N_i}{(1 - ((I_{xz})_s^2 / (I_{xx})_s (I_{zz})_s))} \quad (A.13)$$

$$N'_i = \frac{N_i + ((I_{xz})_s / (I_{zz})_s) L_i}{(1 - ((I_{xz})_s^2 / (I_{xx})_s (I_{zz})_s))} \quad (A.14)$$

The fixed aircraft parameters required for steps 3 and 4 for the AFIT-16 are shown in Table A-4. The transformation from the stability axis system to the body axis system, step 5, is accomplished with the following set of equations:

$$(Y_{v,\delta})_b = (Y_{v,\delta})_s \quad (A.15)$$

$$(Y_p)_b = (Y_p)_s \cos \theta_o - (Y_r)_s \sin \theta_o \quad (A.16)$$

$$(Y_r)_b = (Y_r)_s \cos \theta_o + (Y_p)_s \sin \theta_o \quad (A.17)$$

$$(L'_{v,\delta})_b = (L'_{v,\delta})_s \cos \theta_o - (N'_{v,\delta})_s \sin \theta_o \quad (A.18)$$

$$(L'_p)_b = (L'_p)_s \cos^2 \theta_o - ((L'_r)_s + (N'_p)_s) \sin \theta_o \cos \theta_o + (N'_p)_s \sin^2 \theta_o \quad (A.19)$$

$$(L'_r)_b = (L'_r)_s \cos^2 \theta_o - ((N'_r)_s - (L'_p)_s) \sin \theta_o \cos \theta_o - (N'_p)_s \sin^2 \theta_o \quad (A.20)$$

$$(N'_{v,\delta})_b = (N'_{v,\delta})_s \cos \theta_o + (L'_{v,\delta})_s \sin \theta_o \quad (A.21)$$

$$\begin{aligned} (N'_p)_b &= (N'_p)_s \cos^2 \theta_o - ((N'_r)_s - (L'_p)_s) \sin \theta_o \cos \theta_o \\ &\quad - (L'_r)_s \sin^2 \theta_o \end{aligned} \quad (A.22)$$

$$\begin{aligned} (N'_r)_b &= (N'_r)_s \cos^2 \theta_o + ((L'_r)_s + (N'_p)_s) \sin \theta_o \cos \theta_o \\ &\quad + (L'_p)_s \sin^2 \theta_o \end{aligned} \quad (A.23)$$

The side-slip angle, β , is defined as

$$\beta = v/U \quad (A.24)$$

The dimensional β stability coefficients differ from the v stability coefficients by a factor of $1/g$. Note, for example, that $N_{\beta} \beta = N_r v$. The Y^* terms represent a multiplication by $1/U$, that is

$$Y^*_i = Y_i/U \quad (A.25)$$

The matrices A and B for all ten flight conditions are listed in Table A-5.

TABLE A-2

AFTI-16 Non-Dimensional Stability Axis Data

Lateral-Directional CN

Q	C_{n_β}	C_{n_p}	C_{n_r}	$C_{n_{\delta r}}$	$C_{n_{\delta f}}$	$C_{n_{\delta dt}}$	$C_{n_{\delta c}}$
(lb/ft ²) (deg ⁻¹)	(rad ⁻¹)	(rad ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)
70	.002681	-.038417	-.496924	-.001667	.000379	-.000979	.000934
109	.002121	-.020725	-.492522	-.001618	.000141	-.000986	.001071
163	.001784	-.013125	-.487107	-.001557	.000007	-.000955	.001118
223	.001800	-.010649	-.279610	-.001436	-.000084	-.001069	.001196
436	.001710	-.006850	-.346045	-.001333	-.000144	-.000978	.001215
443	.001775	-.005461	-.476796	-.001379	-.000137	-.000893	.001168
532	.001761	-.004666	-.473181	-.001342	-.000147	-.000862	.001169
652	.001677	-.005165	-.341586	-.001229	-.000149	-.000899	.001217
825	.001636	-.004298	-.273077	-.001137	-.000149	-.000885	.001250
1198	.001447	-.003620	-.260486	-.000947	-.000121	-.000741	.001228

Lateral-Directional CL

Q	C_{l_β}	C_{l_p}	C_{l_r}	$C_{l_{\delta r}}$	$C_{l_{\delta f}}$	$C_{l_{\delta dt}}$	$C_{l_{\delta c}}$
(lb/ft ²) (deg ⁻¹)	(rad ⁻¹)	(rad ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)
70	-.002583	-.223766	.164036	.000179	-.001508	-.001857	-.000014
109	-.002396	-.240803	.103189	.000323	-.001899	-.001649	.000072
163	-.002162	-.242331	.064776	.000387	-.002032	-.001673	.000127
223	-.002358	-.362413	.087114	.000331	-.002195	-.001836	.000226

TABLE A-2 (cont)

Q	C_{l_β}	C_{l_p}	C_{l_r}	$C_{l_{\delta r}}$	$C_{l_{\delta f}}$	$C_{l_{\delta dt}}$	$C_{l_{\delta c}}$
(lb/ft ²) (deg ⁻¹)	(deg ⁻¹)	(rad ⁻¹)	(rad ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)
436	-.001929	-.313529	.031455	.000347	-.001971	-.001810	.000205
443	-.001703	-.232873	.003219	.000392	-.002030	-.001699	.001409
532	-.001608	-.230739	-.004990	.000386	-.001954	-.001656	.000144
652	-.001738	-.306430	.011514	.000326	-.001738	-.001702	.000199
825	-.001742	-.342130	.006361	.000294	-.001523	-.001682	.000224
1198	-.001602	-.328623	-.002132	.000237	-.001184	-.001542	.000217

Lateral-Directional CY

Q	C_{Y_β}	C_{Y_p}	C_{Y_r}	$C_{Y_{\delta r}}$	$C_{Y_{\delta f}}$	$C_{Y_{\delta dt}}$	$C_{Y_{\delta c}}$
(lb/ft ²) (deg ⁻¹)	(deg ⁻¹)	(rad ⁻¹)	(rad ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)	(deg ⁻¹)
70	-.002015	.224381	.603623	.003064	.000703	.002916	.001006
109	-.021784	.189589	.557355	.003069	.000667	.002528	.001075
163	-.020995	.141151	.535556	.003066	.000571	.002132	.001233
223	-.022820	.110272	.556043	.002667	-.000138	.001998	.001350
436	-.021901	.070000	.547367	.002622	-.000113	.001618	.001509
443	-.021095	.058391	.537543	.002991	-.000080	.001364	.000146
532	-.020953	.051413	.537411	.002920	-.000094	.001280	.001462
652	-.021466	.056100	.543785	.002422	-.000112	.001463	.001622
825	-.021582	.045047	.548795	.002110	-.000077	.001492	.001785
1198	-.020906	.038223	.537804	.001745	-.000063	.001208	.001793

TABLE A-3

Lateral-Directional Dimensionalization Factors

$$\begin{array}{lll}
 Y_v = \frac{\rho S U}{2m} C_{Y\beta} & N_\beta = \frac{\rho S U^2 b}{2I_{zz}} C_{n\beta} & L_\beta = \frac{\rho S U^2 b}{2I_{xx}} C_{l\beta} \\
 Y_r = \frac{\rho S b}{4m} C_{Yr} & N_r = \frac{\rho S U b^2}{4I_{zz}} C_{nr} & L_r = \frac{\rho S U b^2}{4I_{xx}} C_{lr} \\
 Y_p = \frac{\rho S b}{4m} C_{Yp} & N_p = \frac{\rho S U b^2}{4I_{zz}} C_{np} & L_p = \frac{\rho S U b^2}{4I_{xx}} C_{lp} \\
 Y_\delta^* = \frac{\rho S U}{2m} C_{Y\delta} & N_\delta = \frac{\rho S U^2 b}{2I_{zz}} C_{n\delta} & L_\delta = \frac{\rho S U^2 b}{2I_{xx}} C_{l\delta}
 \end{array}$$

TABLE A-4

AFTI-16 Fixed Aircraft Parameters

weight, w	21018 pounds
span, b	30 feet
Area, S	300 (feet) ²
(I _{xx}) _b	10033.429 slug-ft ²
(I _{zz}) _b	61278.452 slug-ft ²
(I _{xz}) _b	282.132 slug-ft ²

TABLE A-5

AFTI-16 Plant Matrices

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}, \quad \bar{x}^T = [\phi, \beta, p, r] \quad \bar{u}^T = [\delta_r, \delta_a, \delta_c]$$

Design 1 $Q = 109$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .1219 \\ .0770 & -.1506 & .1210 & -.9926 \\ 0.0000 & -14.7042 & -.8989 & .4595 \\ 0.0000 & 1.5960 & .0007 & -.2750 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0212 & .0090 & .0074 \\ 2.8507 & -12.7814 & -.2979 \\ -1.4230 & -.4112 & .9809 \end{bmatrix}$$

Design 2 $Q = 443$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0318 \\ .0489 & -.3744 & .0318 & -.9995 \\ 0.0000 & -39.9264 & -2.1137 & .0797 \\ 0.0000 & 6.2376 & -.0063 & -.7074 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0531 & .0046 & .0250 \\ 9.7929 & -55.7388 & 2.6033 \\ -5.0544 & -1.8927 & 4.3880 \end{bmatrix}$$

Design 3 $Q = 825$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0255 \\ .0332 & -.4836 & .0255 & -.9997 \\ 0.0000 & -75.3125 & -3.9178 & .0382 \\ 0.0000 & 10.6995 & -.0294 & -.5117 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0473 & .0066 & .0399 \\ 13.4743 & -82.0763 & 8.3887 \\ -7.7774 & -3.2921 & 8.7542 \end{bmatrix}$$

 $Q = 70$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .2016 \\ .0793 & -.1025 & .1977 & -.9803 \\ 0.0000 & -11.0376 & -.6157 & .4996 \\ 0.0000 & 1.2029 & .0009 & -.1777 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0143 & .0067 & .0047 \\ 1.7994 & -7.0917 & -.7020 \\ -.9382 & -.1855 & .5372 \end{bmatrix}$$

TABLE A-5 (cont)

 $Q = 163$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0796 \\ .0744 & -.2091 & .0794 & -.9968 \\ 0.0000 & -19.1782 & -1.2617 & .4152 \\ 0.0000 & 2.1153 & -.0009 & -.3992 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0305 & .0110 & .0123 \\ 4.2081 & -20.3236 & .3607 \\ -2.0671 & -.6771 & 1.5439 \end{bmatrix}$$

 $Q = 223$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0645 \\ .0369 & -.1532 & .0644 & -.9979 \\ 0.0000 & -28.1034 & -1.2593 & .2752 \\ 0.0000 & 2.9444 & -.0149 & -.1532 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0179 & .0024 & .0091 \\ 4.7528 & -30.0084 & 1.7593 \\ -2.6169 & -1.1128 & 2.2662 \end{bmatrix}$$

 $Q = 436$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0399 \\ .0388 & -.3029 & .0399 & -.9992 \\ 0.0000 & -44.5171 & -2.2233 & .2199 \\ 0.0000 & 5.7767 & -.0167 & -.3982 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0363 & .0040 & .0209 \\ 8.8177 & -53.9258 & 3.6280 \\ -4.7193 & -2.0255 & 4.4971 \end{bmatrix}$$

 $Q = 532$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0270 \\ .0480 & -.4392 & .0270 & -.9996 \\ 0.0000 & -45.1440 & -2.4708 & -.0066 \\ 0.0000 & 7.4966 & -.0081 & -.8300 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0612 & .0047 & .0306 \\ 11.4009 & -64.6667 & 3.2286 \\ -5.9204 & -3.2106 & 5.2760 \end{bmatrix}$$

TABLE A-5 (cont)

 $Q = 652$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0303 \\ .0373 & -.4275 & .0302 & -.9995 \\ 0.0000 & -59.6596 & -3.1213 & .1119 \\ 0.0000 & 8.6325 & -.0212 & -.5683 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0482 & .0051 & .0323 \\ 11.9760 & -72.1512 & 5.6206 \\ -6.6298 & -.7405 & 6.7320 \end{bmatrix}$$

 $Q = 1198$

$$A = \begin{bmatrix} 0.0000 & 0.0000 & 1.0000 & .0218 \\ .0320 & -.6574 & .0218 & -.9998 \\ 0.0000 & -100.3136 & -5.2784 & -.0773 \\ 0.0000 & 13.7921 & -.0377 & -.6860 \end{bmatrix} \quad B = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ .0549 & .0075 & .0564 \\ 15.6158 & -96.4546 & 12.0815 \\ -9.4350 & -3.880 & 12.4986 \end{bmatrix}$$

Appendix A-2

Dimensionalization Computer Program

Example and Program Listing

The example shown is that of Design 2.

$$Q = 443 \text{ lb/ft}^2.$$

THIS PROGRAM CONVERTS NON-DIMENSIONAL-STABILITY AXIS-
DATA TO DIMENSIONAL-BODY AXIS-DATA FOR THE AFTI-16

SELECT MODE. LONGITUDNAL ONLY (1)
LATERAL ONLY (2)
BOTH (3) : 2

STANDARD AFTI AIRCRAFT?

YES=1. NO=0. : 1

FIXED AIRCRAFT PARAMETERS

WEIGHT (LBS) :21018.
SPAN (FT) :30.
AREA (FT **2):300.
MAC (FT) :11.32
I-X :10033.429
I-Y :53876.269
I-Z :61278.452
I-XZ :282.13217

INPUT VARIING PARMETERS

HEIGHT (FT) : 5000
MACH : 0.6
SONIC VELOCITY (FT/SEC) : 1097.1
DENSITY (SLUG/FT **3) : 0.002048
ANGLE OF ATTACK (DEG) : 1.824

MOMENT COEF.

CL : .1572
CM : .0026
CD : .0231

INPUT THE LATERAL STABILITY COEFICIENTS

CN , CL , CY
BETA (DEG) : .001775,-.001703,-.021095
P (RAD) : -.005461,-.232873,.058391
R (RAD) : -.476796,.003219,.537543
RUDDER (DEG) : -.001379,.000392,.002991
FLAPERON (DEG) : -.000137,-.002030,-.000080
DIFFAIL (DEG) : -.000893,-.001699,.001364
CANNARD (DEG) : .001168,.000146,.001409

INPUT ALERON COMBINATION TERMS

ALERON= A*FLAPERON + B*DIFF TAIL

ENTER A, B : 1,.25

HEIGHT=5000. (FT) MACH=.6

BODY AXIS DATA -LATERAL-

	N	L	Y	
BETA	6.23792738	-39.92845659	-.37444266	(VEL.)
P	-.00631482	-2.11380615	.00029122	
R	-.70740908	.07970246	.00380601	
RUDDER	-5.05463826	9.79342839	.05309116	
AILERONS	-1.89279772	-55.74144715	.00463283	
CANNARD	4.38821795	2.60330899	.02501018	

THE A-MATRIX IS :

-.37444266	.03182987	-.99948753	.04885258
-39.92845659	-2.11380615	.07970246	0.00000000
6.23792738	-.00631482	-.70740908	0.00000000
0.00000000	1.00000000	.03184556	0.00000000

THE STATE VECTOR TRANSPOSE IS :
(BETA , P , R , PHI)

THE B-MATRIX IS :

.05309116	.00463283	.02501018
9.79342839	-55.74144715	2.60330899
-5.05463826	-1.89279772	4.38821795
0.00000000	0.00000000	0.00000000

THE CONTROL VECTOR TRANSPOSE IS:
(RUDDER, AILERON, CANNARD)

NEW FLIGHT CONDITON?

YES=1. NO=0. : 0

STOP

024600 MAXIMUM EXECUTION FL.

0.103 CP SECONDS EXECUTION TIME.

COMMAND-

```

100=      PROGRAM DIMEN(INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT)
110=      REAL B1(8),B2(8),C1(9),C2(9),C3(9),D1(9),D2(9),D3(9)
120=      REAL E1(9),E2(9),E3(9),F1(9),F2(9),F3(9),G1(2),G2(2)
130=      REAL C4(9),C5(9),C6(9),D4(9),D5(9),D6(9),F4(9),F5(9),F6(9)
140=      REAL G4(9),G5(9),G6(9),AMTL(4,4),BMTL(4,3)
150=      REAL G3(2),AMAT(4,4),BMAT(4,3)
160=C
170=      PRINT*," "
180=      PRINT*,"      THIS PROGRAM CONVERTS NON-DIMENSIONAL-STABILITY AXIS-"
190=      PRINT*,"      DATA TO DIMENSIONAL-BODY AXIS-DATA FOR THE AFTI-16"
200= 500 PRINT*," "
210=      PRINT*,"      SELECT MODE.  LONGITUDNAL ONLY (1)"
220=      PRINT*,"      LATERAL ONLY (2)"
230=      PRINT*,"      BOTH (3)  :"
240=      READ*,I6
250=C
260=      THE FIXED AIRCRAFT PARAMETERS ARE CONTAINED IN THE
270=C      VECTOR B1.
280=C
290=      PRINT*,"  STANDARD AFTI AIRCRAFT?"
300=      PRINT*,"  YES=1.  NO=0.  :"
310=      READ*,I1
320=      CALL AFTI(B1,I1)
330=C
340=      PRINT*," "
350=      PRINT*,"  FIXED AIRCRAFT PARAMETERS"
360=      PRINT*," "
370=      PRINT*,"  WEIGHT (LBS)  :",B1(1)
380=      PRINT*,"  SPAN (FT)    :",B1(2)
390=      PRINT*,"  AREA (FT **2):",B1(3)
400=      PRINT*,"  MAC (FT)     :",B1(4)
410=      PRINT*,"  I-X          :",B1(5)
420=      PRINT*,"  I-Y          :",B1(6)
430=      PRINT*,"  I-Z          :",B1(7)
440=      PRINT*,"  I-XZ         :",B1(8)
450=      PRINT*," "
460=C
470=C
480=C      THE PARAMETERS THAT VARY WITH FLIGHT CONDITIONS ARE
490=C      CONTAINED IN THE VECTOR B2
500=C
510=      PRINT*," "
520=      PRINT*,"  INPUT VARIOUS PARAMETERS"
530=      PRINT*," "
540=      PRINT*,"  HEIGHT (FT)  :"
550=      READ*,B2(1)
560=      PRINT*,"  MACH  :"
570=      READ*,B2(2)
580=      PRINT*,"  SONIC VELOCITY (FT/SEC)  :"
590=      READ*,B2(3)
600=      PRINT*,"  DENSITY (SLUG/FT **3)  :"
610=      READ*,B2(4)
620=      PRINT*,"  ANGLE OF ATTACK (DEG)  :"
630=      READ*,B2(5)

```

```

640= PRINT*, " "
650= PRINT*, "    MOMENT COEF."
660= PRINT*, " "
670= PRINT*, "    CL : "
680= READ*, B2(6)
690= PRINT*, "    CM : "
700= READ*, B2(7)
710= PRINT*, "    CD : "
720= READ*, B2(8)
730=C
740=C    THE NON DIMENSIONAL-STABILITY AXIS-DATA ARE CONTAINED
750=C    IN THE VECTORS C1, C2, C3, C4, C5, C6.
760=C
770=C    LATERAL-DIRECTIONAL
780=C
790=C    C1 CONTAINS THE N-MOMENT COEFFICIENTS
800=C    C2 CONTAINS THE L-MOMENT COEFFICIENTS
810=C    C3 CONTAINS THE Y-FORCE COEFFICIENTS
820=C
830=C    LONGITUDINAL
840=C
850=C    C4 CONTAINS THE D-FORCE COEFFICIENTS
860=C    C5 CONTAINS THE M-MOMENT COEFFICIENTS
870=C    C6 CONTAINS THE L-FORCE COEFFICIENTS
880=C
890= IF(I6.EQ.1) GO TO 1200
900=C
910= PRINT*, " "
920= PRINT*, "    INPUT THE LATERAL STABILITY COEFFICIENTS"
930= PRINT*, " "
940= PRINT*, "                CN , CL , CY"
950= PRINT*, "    BETA (DEG) : "
960= READ*, C1(1), C2(1), C3(1)
970= PRINT*, "    P (RAD) : "
980= READ*, C1(2), C2(2), C3(2)
990= PRINT*, "    R (RAD) : "
1000= READ*, C1(3), C2(3), C3(3)
1010= PRINT*, "    RUDDER (DEG) : "
1020= READ*, C1(4), C2(4), C3(4)
1030= PRINT*, "    FLAPERON (DEG) : "
1040= READ*, C15, C25, C35
1050= PRINT*, "    DIFFAIL (DEG) : "
1060= READ*, C15A, C25A, C35A
1070= PRINT*, "    CANNARD (DEG) : "
1080= READ*, C1(6), C2(6), C3(6)
1090= PRINT*, "    INPUT ALERON COMBINATION TERMS"
1100= PRINT*, "    ALERON= A*FLAPERON + B*DIFF TAIL"
1110= PRINT*, "    ENTER A, B : "
1120= READ*, X7, X8
1130= C1(5)=(X7*C15)+(X8*C15A)
1140= C2(5)=(X7*C25)+(X8*C25A)
1150= C3(5)=(X7*C35)+(X8*C35A)
1160=C
1170=C
1180= IF(I6.EQ.2) GO TO 1300

```

```

1190=C
1200= 1200 PRINT*," "
1210= PRINT*," INPUT THE LONGITUDINAL STABILITY COEFICEINTS"
1220= PRINT*," "
1230= PRINT*," CD CM CL"
1240= PRINT*," "
1250= PRINT*," VELOCITY (FT/SEC) : "
1260= READ*,C4(1),C5(1),C6(1)
1270= PRINT*," ALPHA (DEG) : "
1280= READ*,C4(2),C5(2),C6(2)
1290= PRINT*," ELEVATOR (DEG) : "
1300= READ*,C4(3),C5(3),C6(3)
1310= PRINT*," T.E. FLAP (DEG) : "
1320= READ*,C4(4),C5(4),C6(4)
1330= PRINT*," L.E. FLAP (DEG) : "
1340= READ*,C4(5),C5(5),C6(5)
1350= PRINT*," Q (RAD) ** CM AND CL ONLY ** : "
1360= READ*,C5(6),C6(6)
1370= PRINT*," ALPHA DOT (RAD) ** CM AND CL ONLY ** : "
1380= READ*,C5(7),C6(7)
1390=C
1400=C CONVERT B2(5) ANGLE OF ATTACK FROM
1410=C DEGREES TO RADIANS.
1420=C
1430= 1300 PI=3.141592654
1440= DTR=180.0/PI
1450=C
1460= B2(5)=B2(5)/DTR
1470=C
1480= CA=COS(B2(5))
1490= SA=SIN(B2(5))
1500=C
1510=C CONVERT MOMENTS OF INERTIA FROM BODY AXIS TO
1520=C STABILITY AXIS TO MAKE FUTURE CALCULATIONS CONSISTANT.
1530=C
1540= B15=(B1(5)*(CA**2))-(2.0*B1(8)*CA*SA)+(B1(7)*(SA**2))
1550= B17=(B1(7)*(CA**2))+(2.0*B1(8)*CA*SA)+(B1(5)*(SA**2))
1560= B18=((B1(5)-B1(7))*CA*SA)+(B1(8)*((CA**2)-(SA**2)))
1570=C
1580= B1(5)=B15
1590= B1(7)=B17
1600= B1(8)=B18
1610=C
1620=C CONVERT B1(1) -WEIGHT(LBS)- TO
1630=C -MASS(LBS*(SEC**2)/FT)-
1640=C
1650= B1(1)=B1(1)/32.2
1660=C
1670=C CALCULATE VELOCITY -V- =B2(2)*B2(3)
1680=C B2(2) -MACH-
1690=C B2(3) -SONIC VELOCITY-
1700=C
1710= V=B2(2)*B2(3)
1720=C
1730= U0=V*CA

```

```

1740=      W0=V*SA
1750=C
1760=C      CONVERT STABILITY DERIVATIVES WITH (1/DEG) TO (1/RAD)
1770=C      SPECIFICALLY BETA, RUDDER, AILERONS, AND CANNARDS LATERAL
1780=C      ALPHA, ELEVATOR, AND FLAPS LONGITUDINAL
1790=C
1800=C
1810=      IF(I6.EQ.1) GO TO 1210
1820=C
1830=      C1(1)=C1(1)*DTR
1840=      C2(1)=C2(1)*DTR
1850=      C3(1)=C3(1)*DTR
1860=      C1(4)=C1(4)*DTR
1870=      C2(4)=C2(4)*DTR
1880=      C3(4)=C3(4)*DTR
1890=      C1(5)=C1(5)*DTR
1900=      C2(5)=C2(5)*DTR
1910=      C3(5)=C3(5)*DTR
1920=      C1(6)=C1(6)*DTR
1930=      C2(6)=C2(6)*DTR
1940=      C3(6)=C3(6)*DTR
1950=C
1960=      IF(I6.EQ.2) GO TO 1310
1970=C
1980= 1210 C4(2)=C4(2)*DTR
1990=      C5(2)=C5(2)*DTR
2000=      C6(2)=C6(2)*DTR
2010=      C4(3)=C4(3)*DTR
2020=      C5(3)=C5(3)*DTR
2030=      C6(3)=C6(3)*DTR
2040=      C4(4)=C4(4)*DTR
2050=      C5(4)=C5(4)*DTR
2060=      C6(4)=C6(4)*DTR
2070=      C4(5)=C4(5)*DTR
2080=      C5(5)=C5(5)*DTR
2090=      C6(5)=C6(5)*DTR
2100=C
2110=C
2120=C      G1 G2 G3 CONTAIN THE CONVERSION
2130=C      FACTORS TO CONVERT NON-DIMEN DATA
2140=C      TO DIMEN DATA IN THE STABILITY AXIS
2150=C
2160= 1310 IF(I6.EQ.1) GO TO 1220
2170=C
2180=      G1(1)=(B2(4)*B1(3)*(V**2)*B1(2))/(2.*B1(7))
2190=      G1(2)=(B2(4)*B1(3)*V*(B1(2)**2))/(4.*B1(7))
2200=      G2(1)=G1(1)*B1(7)/B1(5)
2210=      G2(2)=G1(2)*B1(7)/B1(5)
2220=      G3(1)=(B2(4)*B1(3)*V)/(2.*B1(1))
2230=      G3(2)=(B2(4)*B1(3)*B1(2))/(4.*B1(1))
2240=      CALL DIMN(C1,G1,D1)
2250=      CALL DIMN(C2,G2,D2)
2260=      CALL DIMN(C3,G3,D3)
2270=C
2280=C      THE VECTORS D1, D2, D3 CONTAIN THE

```

```

2290=C      DIMEN DATA IN THE STABILITY AXIS
2300=C
2310=      CALL PRM(D1,D2,E1,E2,B1)
2320=C
2330=C      THE VECTORS E1,E2 CONTAIN THE DIMEN
2340=C      DATA IN THE STABILITY N AXIS EXCEPT
2350=C      THE DATA IS PRIMED,( SEE MC RUER, AIRCRAFT
2360=C      DYNAMICS AND AUTOMATIC CONTROL, PG 257)
2370=C
2380=C      THE DIMEN DATA -BODY AXIS- ARE CONTAINED IN
2390=C      THE VECTORS F1,F2,F3
2400=C
2410=      F1(1)=(E1(1)*CA)+(E2(1)*SA)
2420=      F1(2)=(E1(2)*(CA**2))-((E1(3)-E2(2))*CA*SA)-(E2(3)*(SA**2))
2430=      F1(3)=(E1(3)*(CA**2))+((E2(3)+E1(2))*CA*SA)+(E2(2)*(SA**2))
2440=      DO 10 I=4,6
2450= 10 F1(I)=(E1(I)*CA)+(E2(I)*SA)
2460=C
2470=      F2(1)=(E2(1)*CA)-(E1(1)*SA)
2480=      F2(2)=(E2(2)*(CA**2))-((E2(3)+E1(2))*CA*SA)+(E1(3)*(SA**2))
2490=      F2(3)=(E2(3)*(CA**2))-((E1(3)-E2(2))*CA*SA)-(E1(2)*(SA**2))
2500=      DO 20 I=4,6
2510= 20 F2(I)=(E2(I)*CA)-(E1(I)*SA)
2520=C
2530=      F3(1)=D3(1)
2540=      F3(2)=(D3(2)*CA)-(D3(3)*SA)
2550=      F3(3)=(D3(3)*CA)+(D3(2)*SA)
2560=      DO 30 I=4,6
2570= 30 F3(I)=D3(I)
2580=C
2590=      PRINT*," "
2600=      PRINT*,"      HEIGHT=",B2(1)," (FT)      MACH=",B2(2)
2610=      PRINT*," "
2620=      PRINT*,"      BODY AXIS DATA -LATERAL-"
2630=      PRINT*," "
2640=      PRINT*,"                      N          L          Y"
2650=      PRINT*," "
2660=      PRINT 50,F1(1),F2(1),F3(1)
2670=      PRINT 51,F1(2),F2(2),F3(2)
2680=      PRINT 52,F1(3),F2(3),F3(3)
2690=      PRINT 53,F1(4),F2(4),F3(4)
2700=      PRINT 54,F1(5),F2(5),F3(5)
2710=      PRINT 55,F1(6),F2(6),F3(6)
2720= 50 FORMAT(5X," BETA ",5X,3(F16.9,3X)," (VEL.)",/)
2730= 51 FORMAT(5X," P ",5X,3(F16.9,3X),/)
2740= 52 FORMAT(5X," R ",5X,3(F16.9,3X),/)
2750= 53 FORMAT(5X,"RUDDER",5X,3(F16.9,3X),/)
2760= 54 FORMAT(5X,"ALERONS",5X,3(F16.9,3X),/)
2770= 55 FORMAT(5X,"CANNARD",5X,3(F16.9,3X))
2780=C
2790=C      PUT EQUATIONS OF MOTION IN MATRIX FORM.
2800=C      SEE MCRUER PAGE 259
2810=C
2820=      DO 60 I=1,4
2830=      DO 60 J=1,4

```

```

2840=      60 AMAT(I,J)=0.0
2850=      DO 70 I=1,4
2860=      DO 70 J=1,3
2870=      70 BMAT(I,J)=0.0
2880=C
2890=      UO=V*CA
2900=      WO=V*SA
2910=C
2920=      AMAT(1,1)=F3(1)
2930=      AMAT(1,2)=(F3(2)+WO)/V
2940=      AMAT(1,3)=(F3(3)-UO)/V
2950=      AMAT(1,4)=32.174*CA/V
2960=      AMAT(2,1)=F2(1)
2970=      AMAT(2,2)=F2(2)
2980=      AMAT(2,3)=F2(3)
2990=      AMAT(3,1)=F1(1)
3000=      AMAT(3,2)=F1(2)
3010=      AMAT(3,3)=F1(3)
3020=      AMAT(4,2)=1.0
3030=      AMAT(4,3)=SA/CA
3040=      BMAT(1,1)=F3(4)
3050=      BMAT(1,2)=F3(5)
3060=      BMAT(1,3)=F3(6)
3070=      BMAT(2,1)=F2(4)
3080=      BMAT(2,2)=F2(5)
3090=      BMAT(2,3)=F2(6)
3100=      BMAT(3,1)=F1(4)
3110=      BMAT(3,2)=F1(5)
3120=      BMAT(3,3)=F1(6)
3130=C
3140=      PRINT*," "
3150=      PRINT*," THE A-MATRIX IS :"
3160=      CALL PRINTM(4,4,AMAT)
3170=      PRINT*," "
3180=      PRINT*," THE STATE VECTOR TRANSPOSE IS :"
3190=      PRINT*," ( BETA , P , R , PHI )"
3200=      PRINT*," "
3210=      PRINT*," THE B-MATRIX IS :"
3220=      CALL PRINTM(4,3,BMAT)
3230=      PRINT*," "
3240=      PRINT*," THE CONTROL VECTOR TRANSPOSE IS:"
3250=      PRINT*," (RUDDER, ALERON, CANNARD)"
3260=C
3270=      IF(I6.EQ.2) GOTO 1320
3280=C
3290=C      THE VECTORS G4,G5,G6 CONTAIN THE FACTORS
3300=C      THAT CONVERT NON DIMENSIONAL DATA TO
3310=C      DIMENSIONAL DATA -STABILITY AXIS-
3320=C
3330= 1220 G4(1)=B2(4)*B1(3)*V/B1(1)
3340=      G4(2)=G4(1)/2.
3350=      G4(3)=B2(4)*B1(3)*(V**2)/(2.*B1(1))
3360=C
3370=      G5(1)=B2(4)*B1(3)*V*B1(4)/B1(6)
3380=      G5(2)=G5(1)/2.

```

```

3390=      G5(3)=B2(4)*B1(3)*(B1(4)**2)/(4.*B1(6))
3400=      G5(4)=G5(3)*V
3410=      G5(5)=B2(4)*B1(3)*(V**2)*B1(4)/(2.*B1(6))
3420=C
3430=      G6(1)=G4(1)
3440=      G6(2)=G4(2)
3450=      G6(3)=B2(4)*B1(3)*B1(4)/(4.*B1(1))
3460=      G6(4)=G6(3)*V
3470=      G6(5)=B2(4)*B1(3)*(V**2)/(2.*B1(1))
3480=C
3490=C
3500=C
3510=C
3520=C
3530=C
3540=C
3550=C
3560=C
3570=C
3580=C
3590=C
3600=C
3610=C
3620=      D4(1)=G4(1)*(-B2(8)-C4(1))
3630=      D4(2)=G4(2)*(B2(6)-C4(2))
3640=      D4(3)=0.0
3650=      D4(4)=0.0
3660=C
3670=      DO 700 I=5,7
3680= 700 D4(I)=G4(3)*(-C4(I-2))
3690=C
3700=      D5(1)=G5(1)*(B2(7)+C5(1))
3710=      D5(2)=G5(2)*C5(2)
3720=      D5(3)=G5(3)*C5(7)
3730=      D5(4)=G5(4)*C5(6)
3740=C
3750=      DO 710 I=5,7
3760= 710 D5(I)=G5(5)*C5(I-2)
3770=C
3780=      D6(1)=G6(1)*(-B2(6)-C6(1))
3790=      D6(2)=G6(2)*(-C6(2)-B2(8))
3800=      D6(3)=G6(3)*C6(7)
3810=      D6(4)=G6(4)*(-C6(6))
3820=C
3830=      DO 720 I=5,7
3840= 720 D6(I)=G6(5)*(-C6(I-2))
3850=C
3860=C
3870=C
3880=C
3890=      F4(1)=(D4(1)*(CA**2))-((D4(2)+D6(1))*SA*CA)+(D6(2)*(SA**2))
3900=      F4(2)=(D4(2)*(CA**2))+((D4(1)-D6(2))*SA*CA)-(D6(1)*(SA**2))
3910=      F4(3)=(D4(3)*(CA**2))-(D6(3)*SA*CA)
3920=C
3930=      DO 800 I=4,7

```

THE VECTORS D4,D5,D6 CONTAIN THE DIMENSIONAL
DATA -STABILITY AXIS-

** NOTE THE VECTOR SEQUENCE CHANGE **

1 -- VELOCITY --
2 -- W --
3 -- W DOT --
4 -- Q --
5 -- ELEVATOR --
6 -- T.E. FLAP -
7 -- L.E. FLAP --

```

3940= 800 F4(I)=(D4(I)*CA)-(D6(I)*SA)
3950=C
3960= F5(1)=(D5(1)*CA)-(D5(2)*SA)
3970= F5(2)=(D5(2)*CA)+(D5(1)*SA)
3980= F5(3)=D5(3)*CA
3990=C
4000= DO 810 I=4,7
4010= 810 F5(I)=D5(I)
4020=C
4030= F6(1)=(D6(1)*(CA**2))-((D6(2)-D4(1))*SA*CA)-(D4(2)*(SA**2))
4040= F6(2)=(D6(2)*(CA**2))+((D6(1)+D4(2))*SA*CA)+(D4(1)*(SA**2))
4050= F6(3)=(D6(3)*(CA**2))+((D4(3))*SA*CA)
4060=C
4070= DO 820 I=4,7
4080= 820 F6(I)=(D6(I)*CA)+(D4(I)*SA)
4090=C
4100= PRINT*," "
4110= PRINT*," HEIGHT :",B2(1)," MACH :",B2(2)
4120= PRINT*," "
4130= PRINT*," BODY AXIS DATA -LONGITUDINAL-"
4140= PRINT*," "
4150= PRINT*," X M Z"
4160= PRINT*," "
4170= PRINT 90,F4(1),F5(1),F6(1)
4180= PRINT 91,F4(2),F5(2),F6(2)
4190= PRINT 92,F4(3),F5(3),F6(3)
4200= PRINT 93,F4(4),F5(4),F6(4)
4210= PRINT 94,F4(5),F5(5),F6(5)
4220= PRINT 95,F4(6),F5(6),F6(6)
4230= PRINT 96,F4(7),F5(7),F6(7)
4240=C
4250= 90 FORMAT(5X,"VELOCITY",3X,3(F16.9,3X),/)
4260= 91 FORMAT(5X," W ",3X,3(F16.9,3X),/)
4270= 92 FORMAT(5X," W-DOT ",3X,3(F16.9,3X),/)
4280= 93 FORMAT(5X," Q ",3X,3(F16.9,3X),/)
4290= 94 FORMAT(5X,"ELEVATOR",3X,3(F16.9,3X),/)
4300= 95 FORMAT(5X,"T.E.FLAP",3X,3(F16.9,3X),/)
4310= 96 FORMAT(5X,"L.E.FLAP",3X,3(F16.9,3X),/)
4320=C
4330= Z3=1.0/(1.0-F6(3))
4340=C
4350= AMTL(1,1)=F4(1)+(F4(3)*F6(1)*F6(3))
4360= AMTL(1,2)=(F4(2)*U0)+(F4(3)*U0*F6(2)*Z3)
4370= AMTL(1,3)=(F4(4)-W0)+(F4(3)*(F6(4)+U0)*Z3)
4380= AMTL(1,4)=-32.174*CA*(1.0-(F4(3)*Z3))
4390=C
4400= AMTL(2,1)=F6(1)*Z3/U0
4410= AMTL(2,2)=F6(2)*Z3
4420= AMTL(2,3)=(F6(4)+U0)*Z3/U0
4430= AMTL(2,4)=-32.174*CA*Z3/U0
4440=C
4450= AMTL(3,1)=F5(1)+(F5(3)*F6(1)*Z3)
4460= AMTL(3,2)=(F5(2)*U0)+(F5(3)*U0*F6(2)*Z3)
4470= AMTL(3,3)=F5(4)+(F5(3)*(F6(4)+U0)*Z3)
4480= AMTL(3,4)=-32.174*CA*F5(3)*Z3
4490=C

```

```

4500=      AMTL(4,1)=0.0
4510=      AMTL(4,2)=0.0
4520=      AMTL(4,3)=1.0
4530=      AMTL(4,4)=0.0
4540=C
4550=      BMTL(1,1)=F4(5)+(F4(3)*F6(5)*Z3)
4560=      BMTL(1,2)=F4(6)+(F4(3)*F6(6)*Z3)
4570=      BMTL(1,3)=F4(7)+(F4(3)*F6(7)*Z3)
4580=C
4590=      BMTL(2,1)=F6(5)*Z3/U0
4600=      BMTL(2,2)=F6(6)*Z3/U0
4610=      BMTL(2,3)=F6(7)*Z3/U0
4620=C
4630=      BMTL(3,1)=F5(5)+(F5(3)*F6(5)*Z3)
4640=      BMTL(3,2)=F5(6)+(F5(3)*F6(6)*Z3)
4650=      BMTL(3,3)=F5(7)+(F5(3)*F6(7)*Z3)
4660=C
4670=      BMTL(4,1)=0.0
4680=      BMTL(4,2)=0.0
4690=      BMTL(4,3)=0.0
4700=C
4710=C
4720= 1320 PRINT*," "
4730=      PRINT*,"      NEW FLIGHT CONDITON?"
4740=      PRINT*,"      YES=1.  NO=0.  :"
4750=      READ*,I2
4760=      IF(I2.EQ.1) GO TO 500
4770=C
4780=C
4790=C
4800=      STOP
4810=      END
4820=C
4830=      SUBROUTINE DIMN(A1,G1,A2)
4840=      REAL A1(9),G1(2),A2(9)
4850=      A2(1)=A1(1)*G1(1)
4860=      DO 10 J=2,3
4870= 10  A2(J)=A1(J)*G1(2)
4880=      DO 20 K=4,6
4890= 20  A2(K)=A1(K)*G1(1)
4900=      RETURN
4910=      END
4920=C
4930=      SUBROUTINE PRM(A1,A2,E1,E2,B1)
4940=      REAL A1(9),A2(9),E1(9),E2(9),B1(8)
4950=      P1=1-((B1(8)**2)/(B1(7)*B1(5)))
4960=      P2=B1(8)/B1(5)
4970=      P3=B1(8)/B1(7)
4980=      DO 10 I=1,6
4990=      E1(I)=(A1(I)+(P3*A2(I)))/P1
5000= 10  E2(I)=(A2(I)+(P2*A1(I)))/P1
5010=      RETURN
5020=      END
5030=C
5040=      SUBROUTINE AFTI(B1,I1)

```

```

5050=      REAL B1(7)
5060=      IF(I1.EQ.1) GO TO 100
5070=C
5080=      PRINT*," "
5090=      PRINT*," INPUT*FIXED AIRCRAFT PARAMETERS : "
5100=      PRINT*," "
5110=      PRINT*," WEIGHT (LBS) : "
5120=      READ*,B1(1)
5130=      PRINT*," "
5140=      PRINT*," SPAN (FT) : "
5150=      READ*,B1(2)
5160=      PRINT*," "
5170=      PRINT*," AREA (FT **2) : "
5180=      READ*,B1(3)
5190=      PRINT*," "
5200=      PRINT*," MEAN AERODYNAMIC CHORD (FT) : "
5210=      READ*,B1(4)
5220=      PRINT*," "
5230=      PRINT*," MOMENTS OF INERTIA "
5240=      PRINT*," "
5250=      PRINT*," I-X : "
5260=      READ*,B1(5)
5270=      PRINT*," "
5280=      PRINT*," I-Y : "
5290=      READ*,B1(6)
5300=      PRINT*," "
5310=      PRINT*," I-Z : "
5320=      READ*,B1(7)
5330=      PRINT*," "
5340=      PRINT*," I-XZ : "
5350=      READ*,B1(8)
5360=      GO TO 200
5370=      100 B1(1)=21018.0
5380=      B1(2)=30.0
5390=      B1(3)=300.0
5400=      B1(4)=11.32
5410=      B1(5)=10033.429
5420=      B1(6)=53876.269
5430=      B1(7)=61278.452
5440=      B1(8)=282.13217
5450=      200 RETURN
5460=      END
5470=C
5480=      SUBROUTINE PRINTM(N,L,A)
5490=      REAL A(4,4)
5500=      PRINT*," "
5510=C
5520=      DO 100 I=1,N
5530=      PRINT 200,(A(I,J),J=1,L)
5540=      100 CONTINUE
5550=C
5560=      200 FORMAT(1X,8(F15.8,3X))
5570=      RETURN
5580=      END
5590=*EOR
5600=*EOF

```

AD-A124 871

HIGH-GAIN ERROR ACTUATED FLIGHT CONTROL SYSTEMS FOR
CONTINUOUS LINEAR MUL. (U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI. T LEWIS
DEC 82 AFIT/GRE/EE/82D-1 F/G 1/3

2/2

UNCLASSIFIED

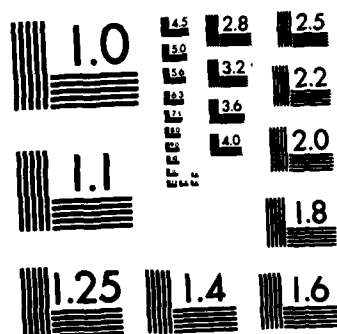
NL

END

FILED

11

DTIC



Appendix B-1

Computer Program to Calculate System Zeros

As part of this thesis effort, Program ZERO is developed to calculate the zeros of the linear matrix system

$$\begin{aligned}\dot{\underline{x}} &= \underline{A} \underline{x} + \underline{B} \underline{u} \\ \dot{\underline{y}} &= \underline{C} \underline{x} + \underline{D} \underline{u}\end{aligned}\tag{B.1}$$

ZERO calculates transmission and decoupling zeros by solving the appropriate generalized eigenvalue problem with the powerful QZ technique (Ref 6).

Three specific eigenvalue problems are used (Ref 8).

The generalized eigenvalue problem to calculate transmission zeros depends on the number of inputs, m , and the number of outputs, ℓ , of the system.

For $\ell \geq m$ the problem is

$$\begin{bmatrix} \underline{A} & \underline{B} & \underline{O} \\ \underline{C} & \underline{D} & \underline{O} \\ \underline{O} & \underline{O} & \underline{K} \end{bmatrix} \underline{z} = \lambda \begin{bmatrix} \underline{I}_n & \underline{O} & \underline{O} \\ \underline{O} & \underline{O} & \underline{O} \\ \underline{O} & \underline{O} & \underline{O} \end{bmatrix} \underline{z}\tag{B.2}$$

For $m \geq \ell$ the problem is

$$\begin{bmatrix} \underline{A} & \underline{B} & \underline{O} \\ \underline{C} & \underline{D} & \underline{O} \\ \underline{O} & \underline{I}_m & \underline{K} \end{bmatrix} \underline{z} = \lambda \begin{bmatrix} \underline{I}_n & \underline{O} & \underline{O} \\ \underline{O} & \underline{O} & \underline{O} \\ \underline{O} & \underline{O} & \underline{O} \end{bmatrix} \underline{z}\tag{B.3}$$

The generalized eigenvalue problem to calculate decoupling zeros is

$$\begin{bmatrix} \underline{A} & \underline{B} & \underline{O} \\ \underline{C} & \underline{D} & \underline{I}_x \\ \underline{O} & \underline{I}_m & \underline{K} \end{bmatrix} \underline{z} = \lambda \begin{bmatrix} \underline{I}_n & \underline{O} & \underline{O} \\ \underline{O} & \underline{O} & \underline{O} \\ \underline{O} & \underline{O} & \underline{O} \end{bmatrix} \underline{z} \quad (\text{B.4})$$

In the above generalized eigenvalue problems, $\underline{k}$ is a random $m - \text{by} - l$ real matrix.

The generalized eigenvalue problem can be described as

$$[\underline{AA}] \underline{z} = \lambda [\underline{BB}] \underline{z} \quad (\text{B.5})$$

The QZ algorithm has three steps. The first reduces the matrix $\underline{AA}$ to upper Hessenberg form and the matrix $\underline{BB}$ to upper triangular form. Next the Hessenberg matrix is further reduced to quasi-triangular form, while maintaining the triangular form of the other matrix. The final step reduces the quasi-triangular matrix further, such that the remaining 2 - by - 2 blocks correspond to pairs of complex eigenvalues. The QZ algorithm returns quantities whose ratio gives the generalized eigenvalues.

The computer program ZERO makes some decisions before the system zeros are finally classified. Refer to the flow chart shown in Figure B-1. For each path of the zero program, the appropriate generalized eigenvalue problem, G.E.P., is solved twice (runs 1 and 2) with only the $\underline{K}$ matrix changing. The outputs of runs 1 and 2 are compared. If the value of an eigenvalue is the same for both runs, then it is classified as a zero.

The zeros calculated by the QZ algorithm, set z_{TA} , may contain some decoupling zeros. Hence, set z_{TA} is compared with the decoupling zeros calculated by the QZ algorithm, set z_D . Those values in set z_{TA} that also exist in set z_D are eliminated from set z_{TA} . The remaining zeros in set z_{TA} form the set of transmission zeros, set z_T .

Experience shows that occasionally some values are incorrectly identified as system zeros. Hence, ZERO uses three different tests in order to filter out incorrect values from the sets z_T and z_D , and to further classify the real decoupling zeros.

Test A (Subroutine Syszero) uses singular value decomposition to check the rank of the system matrix

$$\underline{P}(\lambda) = \begin{bmatrix} \lambda \underline{I}_n - \underline{A} & -\underline{B} \\ \underline{C} & \underline{D} \end{bmatrix} \quad (\text{B.6})$$

An invariant zero causes $\underline{P}(\lambda)$ to lose rank. All transmission zeros are invariant zeros, along with some decoupling zeros. All of the real transmission zeros and real decoupling zeros are sent through Test A. Any transmission zero that does not cause $\underline{P}(\lambda)$ to lose rank is rejected as a zero. Any decoupling zero that does cause $\underline{P}(\lambda)$ to lose rank is classified as an invariant decoupling zero.

Test B (Subroutine RDZ) uses singular value decomposition to check the rank of the matrices $\begin{bmatrix} \lambda \underline{I}_n - \underline{A} & \underline{B} \end{bmatrix}$ and $\begin{bmatrix} \lambda \underline{I}_n - \underline{A} \\ \underline{C} \end{bmatrix}$. All of the real decoupling zeros are sent through Test B, in order to classify them as input, output, or input/output decoupling zeros. Input decoupling zeros cause the matrix $\begin{bmatrix} \lambda \underline{I}_n - \underline{A} & \underline{B} \end{bmatrix}$ to be rank deficient. Output decoupling zeros

cause the matrix $\begin{bmatrix} \lambda I_n - A \\ \underline{C} \end{bmatrix}$ to be rank deficient. Input/output decoupling zeros cause both matrices to be rank deficient. A real decoupling zero that causes neither matrix to lose rank is rejected as a zero.

Test C (Subroutine Complex) is for complex system zeros. All complex zeros must occur in conjugate pairs. Any complex zero without a conjugate is rejected.

ZERO is an interactive program. The required inputs are listed in Table B-1. Input data may be read from Tape3 = ZDATA. The order of the inputs for Tape3 = ZDATA is also listed in Table B-1. The ZDATA input format is (3I2, (n²+nm+2n²)F20.7). Two ASD libraries, EISPACK and IMSL, must be connected before execution of the program ZERO.

The current version of ZERO is not 100% accurate. Sometimes an incorrect value will pass through the filter tests. These values can be easily identified because they are very large values. In the third example the number -19,722,484.728 is incorrectly identified as a transmission zero.

TABLE B-1

Required Input Data for Program Zero

Name	Symbol	Dimension	Input Order	
			Interactive	ZDATA
Number of States	n		1	1
<u>A</u> Matrix	AMAT	n-by-n	2	4
Number of Inputs	m		3	2
<u>B</u> Matrix	BMAT	n-by-m	4	5
Number of Outputs			5	3
<u>C</u> Matrix	CMAT	l-by-n	6	6
<u>D</u> Matrix	DMAT	l-by-m	7	
<u>F</u> Matrix	FMAT	l-by-n		7

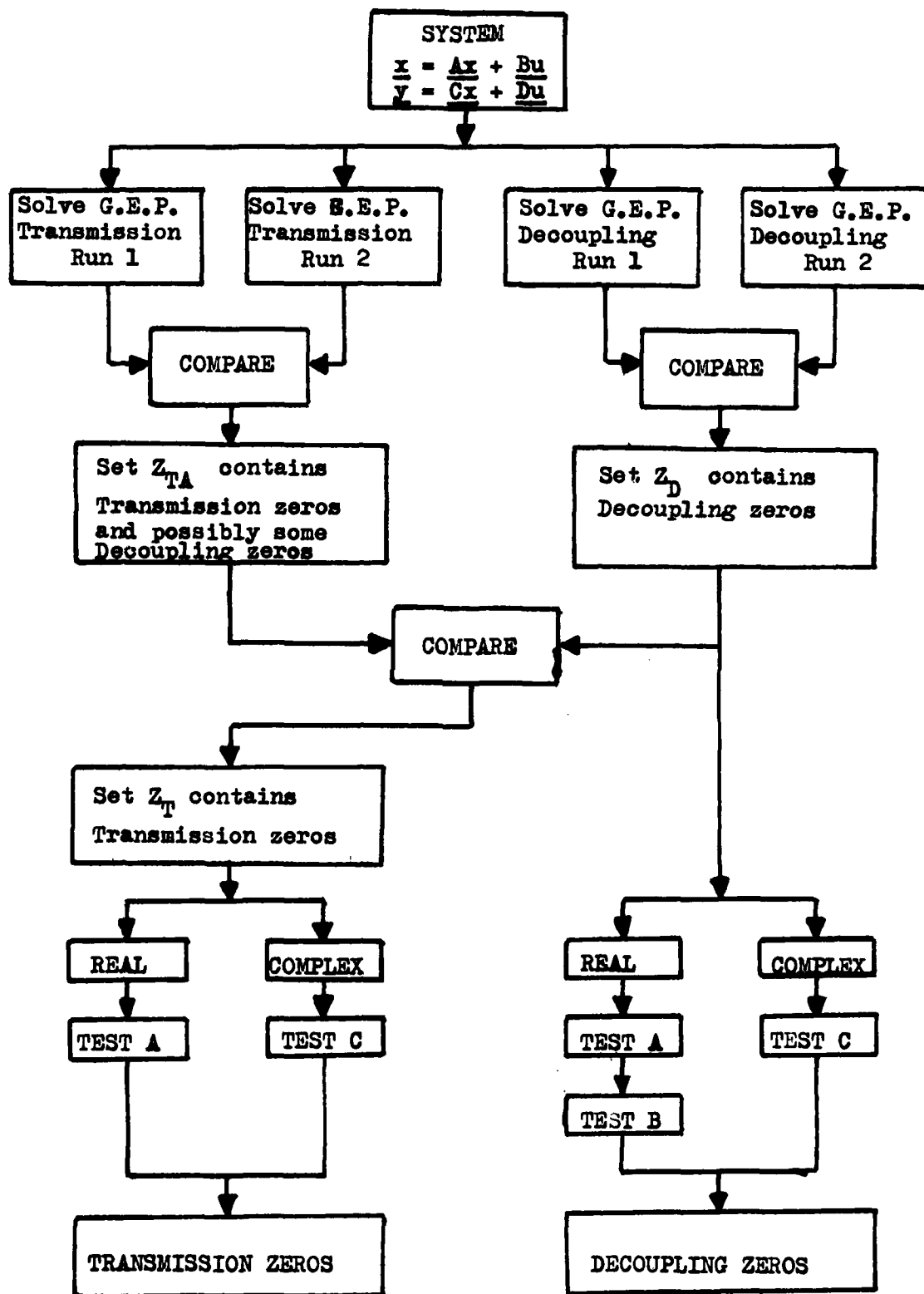


Figure B-1 Program ZERO Flow Chart

Appendix B-2

Listing of Program ZERO and Examples

Three examples are included in this appendix. The first two examples are from published papers (Ref 4:60) and (Ref 3:792). The third example is the system of design 2 used in this thesis.

THIS TEST PROGRAM CALCULATES ZEROS OF A LINEAR
MATRIX SYSTEM.

MAXIMUM # OF STATES = 8.
OF INPUTS = 4.
OF OUTPUTS = 4.

TO INCREASE MAXIMUM, SEE INSTRUCTIONS AT THE
BEGINNING OF THE PROGRAM LISTING.

FILE ZERO LIS, ID=LEWIS, SN=AFIT

ARE THE MATRICES STORED ON TAPE 3 = ZDATA ?
(1 = YES. 0 = NO) : 0

ENTER THE DIMENSION OF THE SYSTEM : 6

ENTER THE A MATRIX (6 BY 6) :

ROW 1 : 1,0,0,0,0,0
ROW 2 : 0,1,0,0,0,0
ROW 3 : 0,0,3,0,0,0
ROW 4 : 0,0,0,-4,0,0
ROW 5 : 0,0,0,0,-1,0
ROW 6 : 0,0,0,0,0,3

ENTER THE NUMBER OF INPUTS : 2

ENTER THE B MATRIX (6 BY 2)

ROW 1 : 0,-1
ROW 2 : -1,0
ROW 3 : 1,-1
ROW 4 : 0,0
ROW 5 : 0,1
ROW 6 : -1,-1

ENTER THE NUMBER OF OUTPUTS : 3

ENTER THE C MATRIX (3 BY 6)

ROW 1 : 1,0,0,1,0,0
ROW 2 : 0,1,0,1,0,1
ROW 3 : 0,0,1,0,0,1

IS THERE A D MATRIX ?
(YES=1 NO=0) : 0

THE D MATRIX IS A (3 BY 2) ZERO MATRIX

HE A MATRIX IS :

1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	3.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-4.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	3.0000

THE B MATRIX IS :

0.0000	-1.0000
-1.0000	0.0000
1.0000	-1.0000
0.0000	0.0000
0.0000	1.0000
-1.0000	-1.0000

THE C MATRIX IS :

1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
0.0000	1.0000	0.0000	1.0000	0.0000	1.0000
0.0000	0.0000	1.0000	0.0000	0.0000	1.0000

*** WARNING ERROR (IER = 33) FROM IMSL ROUTINE LSVDF
*** WARNING ERROR (IER = 33) FROM IMSL ROUTINE LSVDF
WARNING 33 MEANS A MATRIX DOES NOT
HAVE FULL RANK. THIS MESSAGE IS EXPECTED

THE TRANSMISSION ZEROS

REAL PART	IMMAGINARY PART
-----------	-----------------

2.0000000000	0.
--------------	----

THE DECOUPLING ZEROS

-1.0000	OUTPUT INVARIANT
-4.0000	INPUT

DO YOU WISH TO CHANGE MATRICES
AND CONTINUE?

```

STOP = 0
CHANGE A ONLY   = 1
CHANGE B ONLY   = 2
CHANGE C ONLY   = 3
CHANGE A,B,C,D = 4
CHANGE B,C ONLY = 5
CHANGE C,D ONLY = 6
: 4
ENTER THE DIMENSION OF THE SYSTEM : 6

ENTER THE A MATRIX (6 BY 6) :

ROW 1 : 0,1,0,0,0,0
ROW 2 : 0,0,1,0,0,0
ROW 3 : 0,0,0,0,0,0
ROW 4 : 0,0,0,0,1,0
ROW 5 : 0,0,0,0,0,1
ROW 6 : 0,0,0,0,0,0

ENTER THE NUMBER OF INPUTS : 2

ENTER THE B MATRIX (6 BY 2)

ROW 1 : 0,0
ROW 2 : 0,0
ROW 3 : 1,0
ROW 4 : 0,0
ROW 5 : 0,0
ROW 6 : 0,1

ENTER THE NUMBER OF OUTPUTS : 2

ENTER THE C MATRIX (2 BY 6)

ROW 1 : 1,1,0,0,0,0
ROW 2 : 0,0,0,-1,-1,0

'IS THERE A D MATRIX ?
(YES=1 NO=0) : 1

ENTER THE D MATRIX (2 BY 2)

ROW 1 : 1,0
ROW 2 : 1,0

```

THE A MATRIX IS :

0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

THE B MATRIX IS :

0.0000	0.0000
0.0000	0.0000
1.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	1.0000

THE C MATRIX IS :

1.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	-1.0000	-1.0000	0.0000

THE D MATRIX IS :

1.0000	0.0000
1.0000	0.0000

*** WARNING ERROR (IER = 33) FROM IMSL ROUTINE LSVDF
 *** WARNING ERROR (IER = 33) FROM IMSL ROUTINE LSVDF
 WARNING 33 MEANS A MATRIX DOES NOT
 HAVE FULL RANK. THIS MESSAGE IS EXPECTED

THE TRANSMISSION ZEROS

REAL PART	IMAGINARY PART
.34116390191	1.1615414000
.34116390191	-1.1615414000
-1.0000000000	0.
-.6823278033	0.

THERE ARE NO DECOUPLING ZEROS

THIS TEST PROGRAM CALCULATES ZEROS OF A LINEAR
MATRIX SYSTEM.

MAXIMUM # OF STATES = 8.

OF INPUTS = 4.

OF OUTPUTS = 4.

TO INCREASE MAXIMUM, SEE INSTRUCTIONS AT THE
BEGINNING OF THE PROGRAM LISTING.

FILE ZEROLIS, ID=LEWIS, SN=AFIT

ARE THE MATRICES STORED ON TAPE 3 = ZDATA ?

(1 = YES. 0 = NO) : 1

THE A MATRIX IS :

0.0000	0.0000	1.0000	.0318
.0489	-.3744	.0318	-.9995
0.0000	-39.9283	-2.1138	.0797
0.0000	6.2379	-.0063	-.7074

THE B MATRIX IS :

0.0000	0.0000	0.0000
.0531	.0046	.0250
9.7929	-55.7388	2.6033
-5.0544	-1.8927	4.3880

THE C MATRIX IS :

0.0000	1.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	1.0000

*** WARNING ERROR (IER = 33) FROM IMSL ROUTINE LSVDF
*** WARNING ERROR (IER = 33) FROM IMSL ROUTINE LSVDF
WARNING 33 MEANS A MATRIX DOES NOT
HAVE FULL RANK. THIS MESSAGE IS EXPECTED

THE TRANSMISSION ZEROS

REAL PART

IMAGINARY PART

-19722484.728 0.

THERE ARE NO DECOUPLING ZEROS

DO YOU WISH TO CHANGE MATRICES
AND CONTINUE?

STOP = 0
CHANGE A ONLY = 1
CHANGE B ONLY = 2
CHANGE C ONLY = 3
CHANGE A,B,C,D = 4
CHANGE B,C ONLY = 5
CHANGE C,D ONLY = 6
: 3
CHANGE C-MATRIX TO F-MATRIX ?
YES=1, NO=0 : 1

THE C MATRIX IS :

0.0000	1.0000	0.0000	0.0000
1.0000	0.0000	.2500	.0080
0.0000	0.0000	0.0000	1.0000

*** WARNING ERROR (IER = 33) FROM IMSL ROUTINE LSVDF
WARNING 33 MEANS A MATRIX DOES NOT
HAVE FULL RANK. THIS MESSAGE IS EXPECTED

THE TRANSMISSION ZEROS

REAL PART	IMMAGINARY PART
-----------	-----------------

-4.0000000000	0.
---------------	----

THERE ARE NO DECOUPLING ZEROS

DO YOU WISH TO CHANGE MATRICES
AND CONTINUE?

STOP = 0
CHANGE A ONLY = 1
CHANGE B ONLY = 2
CHANGE C ONLY = 3
CHANGE A,B,C,D = 4
CHANGE B,C ONLY = 5
CHANGE C,D ONLY = 6
: 0

```

100=    PROGRAM ZERO(INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT,
110= 1   ZDATA,TAPE3=ZDATA)
120=    REAL TRAR(16),TRAI(16),DECR(16),DECI(16),FMAT(8,8)
130=    REAL AMAT(8,8),BMAT(8,8),CMAT(8,8),DMAT(8,8)
140=    INTEGER IFLT(16),IFL(16),IO(16)
150=C
160=    LOGICAL MATV
170=C
180=C    THIS PROGRAM CAN BE SIMPLY ALTERED TO ACCOMMODATE
190=C    MORE STATES AND INPUTS/OUTPUTS.
200=C    THIS IS DONE BY CHANGING DIMENSION STATEMENTS
210=C    THE IMPORTANT DIMENSION NUMBERS ARE:
220=C        N - NUMBER OF STATES
230=C        M - NUMBER OF INPUTS
240=C        L - NUMBER OF OUTPUTS
250=C        NM - N + MAX(M,L)
260=C        NN - N + M + L
270=C
280=C    CHANGE ALL ARRAYS WITH (8,8) TO (N,N)
290=C    CHANGE ALL ARRAYS WITH (12,12) TO (NM,NM)
300=C    CHANGE ,12, IN IMSL FUNCTION LSVDF IN SUBROUTINE RDZ
310=C    AND SUBROUTINE SYSZERO TO ,NM.
320=C    CHANGE ALL VECTORS WITH (16) AND ARRAYS
330=C    WITH (16,16) TO (NN) OR (NN,NN)
340=C    ALSO CHANGE THE ,16, IN THE QZ SUBROUTINES
350=C    TO ,NN.
360=C
370=C    A SAMPLE COMMAND IN EDITOR IS : /16/= /NN/,A,U,V (VETO)
380=C    SAVE NEW FILE THEN RECOMPILE
390=C
400=C
410=    PRINT*," "
420=    PRINT*,"      THIS TEST PROGRAM CALCULATES ZEROS OF A LINEAR"
430=    PRINT*,"      MATRIX SYSTEM."
440=    PRINT*,"      MAXIMUM # OF STATES = 8."
450=    PRINT*,"      OF INPUTS = 4."
460=    PRINT*,"      OF OUTPUTS= 4."
470=    PRINT*,"      TO INCREASE MAXIMUM, SEE INSTRUCTIONS AT THE"
480=    PRINT*,"      BEGINNING OF THE PROGRAM LISTING."
490=    PRINT*,"      FILE ZEROLIS,ID=LEWIS,SN=AFIT"
500=    PRINT*," "
510=    PRINT*,"      ARE THE MATRICES STORED ON TAPE 3 = ZDATA ?"
520=    PRINT*,"      ( 1 = YES.   0 = NO ) : "
530=    READ*,N1
540=    IF(N1.EQ.1) GO TO 1100
550=C
560=    PRINT*," "
570=    MATCH=0
580= 5400 PRINT*,"      ENTER THE DIMENSION OF THE SYSTEM : "
590=    READ*,N
600=C
610=C    INPUT THE BLOCK A MATRIX
620=C
630=    PRINT*," "

```

```

640=      PRINT*,"      ENTER THE A MATRIX (",N," BY ",N,") : "
650=      CALL READM(N,N,AMAT)
660=      IF(MATCH.EQ.1)GO TO 5430
670=C
680=      PRINT*," "
690= 5410 PRINT*,"      ENTER THE NUMBER OF INPUTS : "
700=      READ*,M
710=C
720=C      INPUT THE BLOCK B MATRIX
730=C
740=      PRINT*," "
750=      PRINT*,"      ENTER THE B MATRIX (",N," BY ",M,")"
760=      CALL READM(N,M,BMAT)
770=      IF(MATCH.EQ.2)GO TO 5430
780=C
790=      PRINT*," "
800= 5420 IF(N1.EQ.1) GO TO 5421
810= 5424 PRINT*,"      ENTER THE NUMBER OF OUTPUTS : "
820=      READ*,L
830=      GO TO 5422
840= 5421 PRINT*,"      CHANGE C-MATRIX TO F-MATRIX ?"
850=      PRINT*,"      YES=1, NO=0 : "
860=      READ*,N2
870=      IF(N2.EQ.1) GO TO 5423
880=      GO TO 5424
890= 5422 CONTINUE
900=C
910=C      INPUT THE BLOCK C MATRIX
920=C
930=      PRINT*," "
940=      PRINT*,"      ENTER THE C MATRIX (",L," BY ",N,")"
950=      CALL READM(L,N,CMAT)
960=C
970=      IF(MATCH.EQ.4) GO TO 5435
980=      IF(MATCH.EQ.6) GO TO 5435
990=      IF(MATCH.NE.0) GO TO 1200
1000=C
1010= 5435 PRINT*," "
1020=      PRINT*,"      IS THERE A D MATRIX ?"
1030=      PRINT*,"      (YES=1 NO=0) : "
1040=      READ*,I9
1050=      IF(I9.EQ.1) GO TO 5440
1060=C
1070=      DO 5450 I=1,L
1080=      DO 5450 J=1,M
1090= 5450 DMAT(I,J)=0.0
1100=      PRINT*," "
1110=      PRINT*,"      THE D MATRIX IS A (",L," BY ",M,") ZERO MATRIX"
1120=      PRINT*," "
1130=      GO TO 5430
1140=C
1150= 5440 PRINT*," "
1160=      PRINT*,"      ENTER THE D MATRIX (",L," BY ",M,")"
1170=      CALL READM(L,M,DMAT)

```

```

1180=C
1190= 5430 GO TO 1200
1200=C
1210=C
1220=C      READ THE INPUT DATA FROM TAPE 3 = ZDATA
1230=C
1240=C
1250= 1100 CONTINUE
1260=C
1270=      READ(3,45) N,M,L
1280= 45 FORMAT(3I2)
1290=C
1300=      DO 50 I=1,N
1310=      DO 50 J=1,N
1320= 50 READ(3,55) AMAT(I,J)
1330=C
1340=      DO 60 I=1,N
1350=      DO 60 J=1,M
1360= 60 READ(3,55) BMAT(I,J)
1370=C
1380=      DO 70 I=1,L
1390=      DO 70 J=1,N
1400= 70 READ(3,55) CMAT(I,J)
1410=C
1420=      DO 80 I=1,L
1430=      DO 80 J=1,N
1440= 80 READ(3,55) FMAT(I,J)
1450=C
1460=      I9=0
1470=      DO 90 I=1,L
1480=      DO 90 J=1,M
1490= 90 DMAT(I,J)=0.0
1500=C
1510=C
1520= 55 FORMAT(F20.7)
1530=      GO TO 1200
1540=C
1550= 5423 CONTINUE
1560=      DO 85 I=1,L
1570=      DO 85 J=1,N
1580= 85 CMAT(I,J)=FMAT(I,J)
1590=C
1600= 1200 CONTINUE
1610=C
1620=C
1630=      IF(MATCH.EQ.2) GO TO 91
1640=      IF(MATCH.EQ.3) GO TO 92
1650=      IF(MATCH.EQ.5) GO TO 91
1660=      IF(MATCH.EQ.6) GO TO 92
1670=C
1680=      PRINT*," "
1690=      PRINT*,"      THE A MATRIX IS :"
1700=      PRINT*," "
1710=      CALL PRINTM(N,N,AMAT)

```

```

1720=C
1730=      IF(MATCH.EQ.1) GO TO 95
1740=C
1750=  91 PRINT*," "
1760=      PRINT*," "
1770=      PRINT*,"      THE B MATRIX IS :"
1780=      PRINT*," "
1790=      CALL PRINTM(N,M,BMAT)
1800=C
1810=      IF(MATCH.EQ.2) GO TO 95
1820=C
1830=  92 PRINT*," "
1840=      PRINT*," "
1850=      PRINT*,"      THE C MATRIX IS :"
1860=      PRINT*," "
1870=      CALL PRINTM(L,N,CMAT)
1880=      PRINT*," "
1890=      IF(MATCH.EQ.3) GO TO 95
1900=      IF(MATCH.EQ.5) GO TO 95
1910=C
1920=      IF(I9.EQ.1) GO TO 96
1930=      GO TO 95
1940=  96 PRINT*," "
1950=      PRINT*,"      THE D MATRIX IS :"
1960=      PRINT*," "
1970=      CALL PRINTM(L,M,DMAT)
1980=      PRINT*," "
1990=  95 CONTINUE
2000=C
2010=C
2020=C      CALCULATE ZEROS
2030=C
2040=C
2050=      NN=N+M+L
2060=      MATV=.FALSE.
2070=C
2080=C
2090=C      SUBROUTINE TZ CALCULATES THE TRANSMISSION ZEROS
2100=C
2110=C
2120=      CALL TZ(AMAT,BMAT,CMAT,DMAT,NN,N,M,L,MATV,
2130=      +TRAR,TRAI,IFLT)
2140=C
2150=C      DOUBLE CHECK REAL TRANSMISSION ZEROS BY TESTING
2160=C      THE RANK OF THE SYSTEM MATRIX. A TRANSMISSION
2170=C      ZERO CAUES THE SYSTEM MATRIX TO BE RANK DEFICIENT.
2180=C
2190=      EX3=1.0E-3
2200=C
2210=      DO 440 I=1,NN
2220=      IF(IFLT(I).EQ.0)GO TO 440
2230=      IF(TRAI(I).LT.EX3.AND.TRAI(I).GT.-EX3) GO TO 445
2240=      GO TO 440
2250=  445 Z=TRAR(I)

```

```

2260=      CALL SYSZERO(AMAT,BMAT,CMAT,DMAT,Z,N,M,L,IT,1)
2270=      IF(IT.EQ.0) IFLT(I)=0
2280=  440 CONTINUE
2290=C
2300=C      SUBROUTINE DZ CALCULATES THE DECOUPLING ZEROS
2310=C
2320=C
2330=      CALL DZ(AMAT,BMAT,CMAT,DMAT,NN,N,M,L,MATV,
2340=      +DECR,DECI,IFL)
2350=C
2360=C
2370=C      COMPARE THE TRANSMISSION ZEROS FOUND IN TZ
2380=C      WITH THE DECOUPLING ZEROS FOUND IN DZ.
2390=C      IF THE SAME VALUE APPEARS IT IS ONLY A
2400=C      DECOUPLING ZERO.
2410=C
2420=      EX=1.0E-5
2430=C
2440=      DO 200 I=1,NN
2450=C
2460=      IF(IFLT(I).EQ.0) GO TO 200
2470=      ATRAR=ABS(TRAR(I))
2480=      ATRAI=ABS(TRAI(I))
2490=C
2500=      DO 210 J=1,NN
2510=C
2520=      IF(IFL(J).EQ.0) GO TO 210
2530=      ADECR=ABS(DECR(J))
2540=      ADECI=ABS(DECI(J))
2550=C
2560=C      COMPARE REAL SYSTEM ZEROS
2570=C
2580=      IF(ATRAI.GT.EX3) GO TO 201
2590=      IF(ADECI.GT.EX3) GO TO 210
2600=      IF(ATRAR.LT.EX3) GO TO 202
2610=C
2620=      DT=(TRAR(I)-DECR(J))/TRAR(I)
2630=      IF(ABS(DT).LT.EX) IFLT(I)=0
2640=      GO TO 210
2650=C
2660=  202 IF(TRAR(I).LT.DECR(J)+EX.AND.TRAR(I).GT.DECR(J)-EX) IFLT(I)=0
2670=      GO TO 210
2680=C
2690=C      COMPARE IMMANGINARY SYSTEM ZEROS
2700=C
2710=  201 IF(ADECI.LT.EX3) GO TO 210
2720=      IF(ATRAR.LT.EX3) GO TO 203
2730=C
2740=      DTI=(TRAR(I)-DECR(J))/TRAR(I)
2750=      IF(ABS(DTI).LT.EX) GO TO 204
2760=      GO TO 210
2770=C
2780=  203 IF(TRAR(I).LT.DECR(J)+EX.AND.TRAR(I).GT.DECR(J)-EX) GO TO 204
2790=      GO TO 210

```

```

2800=C
310= 204 IF(ATRAI.LT.EX3) GO TO 205
2820= DTII=(TRAI(I)-DECI(J))/TRAI(I)
2830= IF(ABS(DTII).LT.EX) IFLT(I)=0
2840= GO TO 210
2850=C
2860= 205 IF(TRAI(I).LT.DECI(J)+EX.AND.TRAI(I).GT.DECI(J)-EX) IFLT(I)=0
2870=C
2880= 210 CONTINUE
2890= 200 CONTINUE
2900=C
2910=C CHECK CALCULATED COMPLEX ZEROS FOR COMPLEX
2920=C CONJUGATE PAIRS. COMPLEX ZEROS MUST OCCUR IN
2930=C CONJUGATE PAIRS.
2940=C
2950= CALL COMPLEX(NN,TRAI,IFLT)
2960= CALL COMPLEX(NN,DECI,IFL)
2970=C
2980= KTZ=0
2990= ITZ=1
3000= DO 300 I=1,NN
3010= IF(IFLT(I).EQ.1) GO TO 300
3020= KTZ=KTZ+1
3030= IF(KTZ.EQ.NN) GO TO 305
3040= GO TO 300
3050= 305 ITZ=0
3060= 300 CONTINUE
3070=C
3080= KDZ=0
3090= IDZ=1
3100= DO 310 J=1,NN
3110= IF(IFL(J).EQ.1) GO TO 310
3120= KDZ=KDZ+1
3130= IF(KDZ.EQ.NN) GO TO 315
3140= GO TO 310
3150= 315 IDZ=0
3160= 310 CONTINUE
3170=C
3180=C
3190=C TEST DECOUPLING ZEROS TO DETERMIN WETHER THEY
3200=C ARE INVARIANT, AND IF THEY ARE AN INPUT OR OUTPUT
3210=C ZERO.
3220=C
3230=C
3240= DO 500 I=1,NN
3250= IO(I)=0
3260= IF(IFL(I).EQ.0) GO TO 500
3270= IF(DECI(I).LT.EX3.AND.DECI(I).GT.-EX3) GO TO 550
3280= GO TO 500
3290= 550 Z=DECR(I)
3300= CALL RDZ(AMAT,BMAT,CMAT,Z,N,M,L,KIO)
3310= CALL SYSZERO(AMAT,BMAT,CMAT,DMAT,Z,N,M,L,ID,2)
3320= IF(ID.EQ.1) GO TO 510
3330= IF(KIO.EQ.1) IO(I)=1

```

```

3340=      IF(KIO.EQ.2) IO(I)=2
3350=      IF(KIO.EQ.3) IO(I)=3
3360=      GO TO 500
3370=  510 IF(KIO.EQ.1) IO(I)=4
3380=      IF(KIO.EQ.2) IO(I)=5
3390=      IF(KIO.EQ.3) IO(I)=6
3400=  500 CONTINUE
3410=C
3420=      IF(ITZ.EQ.0.AND.IDZ.EQ.0) GO TO 399
3430=      PRINT*,"      WARNING 33 MEANS A MATRIX DOES NOT"
3440=      PRINT*,"      HAVE FULL RANK.  THIS MESSAGE IS EXPECTED"
3450=      PRINT*," "
3460=      PRINT*," "
3470=  399 CONTINUE
3480=C
3490=C
3500=      IF(ITZ.EQ.0) GO TO 400
3510=      PRINT*," "
3520=      PRINT*,"      THE TRANSMISSION ZEROS"
3530=      PRINT*," "
3540=      PRINT 412
3550=      PRINT*," "
3560=C
3570=      DO 410 I=1,NN
3580=      IF(IFLT(I).EQ.0) GO TO 410
3590=      PRINT*," "
3600=      PRINT 411,(TRAR(I),TRAI(I))
3610=  410 CONTINUE
3620=C
3630=  415 IF(IDZ.EQ.0) GO TO 405
3640=      PRINT*," "
3650=      PRINT*,"      THE DECOUPLING ZEROS"
3660=C
3670=      DO 420 I=1,NN
3680=      IF(IFL(I).EQ.0) GO TO 420
3690=      IF(IO(I).EQ.0) GO TO 421
3700=      IF(IO(I).EQ.1) GO TO 422
3710=      IF(IO(I).EQ.2) GO TO 423
3720=      IF(IO(I).EQ.4) GO TO 424
3730=      IF(IO(I).EQ.5) GO TO 426
3740=      IF(IO(I).EQ.6) GO TO 427
3750=C
3760=      PRINT*," "
3770=      PRINT 413,DECR(I)
3780=      GO TO 420
3790=  427 PRINT*," "
3800=      PRINT 417,DECR(I)
3810=      GO TO 420
3820=  426 PRINT*," "
3830=      PRINT 418,DECR(I)
3840=      GO TO 420
3850=  424 PRINT*," "
3860=      PRINT 419,DECR(I)
3870=      GO TO 420

```

```

3880= 423 PRINT*," "
3890= PRINT 414,DECR(I)
3900= GO TO 420
3910= 422 PRINT*," "
3920= PRINT 416,DECR(I)
3930= GO TO 420
3940= 421 PRINT*," "
3950= PRINT 412
3960= PRINT 411,(DECR(I),DECI(I))
3970= 420 CONTINUE
3980=C
3990= 411 FORMAT(2G20.11)
4000= 412 FORMAT(8X,"REAL PART",8X,"IMMAGINARY PART")
4010= 413 FORMAT(5X,F10.4,5X," INPUT/OUTPUT")
4020= 414 FORMAT(5X,F10.4,5X," OUTPUT")
4030= 416 FORMAT(5X,F10.4,5X," INPUT")
4040= 417 FORMAT(5X,F10.4,5X," INPUT/OUTPUT INVARIANT")
4050= 418 FORMAT(5X,F10.4,5X," OUTPUT INVARIANT")
4060= 419 FORMAT(5X,F10.4,5X," INPUT INVARIANT")
4070=C
4080=C
4090=C
4100= GO TO 430
4110= 400 PRINT*," "
4120= PRINT*," THERE ARE NO TRANSMISSION ZEROS"
4130= PRINT*," "
4140= PRINT*," "
4150= PRINT*," "
4160= GO TO 415
4170= 405 PRINT*," "
4180= PRINT*," THERE ARE NO DECOUPLING ZEROS"
4190= PRINT*," "
4200= PRINT*," "
4210= PRINT*," "
4220= 430 CONTINUE
4230=C
4240=C
4250=C SELECT OPTION. CONTINUE OR RETURN
4260=C
4270= PRINT*," "
4280= PRINT*," DO YOU WISH TO CHANGE MATRICES"
4290= PRINT*," AND CONTINUE?"
4300= PRINT*," "
4310= PRINT*," STOP = 0"
4320= PRINT*," CHANGE A ONLY = 1"
4330= PRINT*," CHANGE B ONLY = 2"
4340= PRINT*," CHANGE C ONLY = 3"
4350= PRINT*," CHANGE A,B,C,D = 4"
4360= PRINT*," CHANGE B,C ONLY = 5"
4370= PRINT*," CHANGE C,D ONLY = 6"
4380= PRINT*," : "
4390= READ*,MATCH
4400= IF(MATCH.EQ.1) GO TO 5400
4410= IF(MATCH.EQ.2) GO TO 5410

```

```

4420=      IF(MATCH.EQ.3) GO TO 5420
4430=      IF(MATCH.EQ.4) GO TO 5400
4440=      IF(MATCH.EQ.5) GO TO 5410
4450=      IF(MATCH.EQ.6) GO TO 5420
4460=      STOP
4470=      END
4480=C
4490=C
4500=      SUBROUTINE READM(L,N,A)
4510=      REAL A(8,8)
4520=      PRINT*," "
4530=C
4540=      DO 100 I=1,L
4550=      PRINT*,"      ROW ",I," : "
4560=      READ*,(A(I,J),J=1,N)
4570= 100 CONTINUE
4580=C
4590=      RETURN
4600=      END
4610=C
4620=      SUBROUTINE PRINTM(L,N,A)
4630=      REAL A(8,8)
4640=      PRINT*," "
4650=C
4660=      DO 100 I=1,L
4670=      PRINT 200,(A(I,J),J=1,N)
4680= 100 CONTINUE
4690=C
4700= 200 FORMAT(1X,8(F12.4,3X))
4710=      RETURN
4720=      END
4730=C
4740=C
4750=      SUBROUTINE MSUB(A,B,C,L,N)
4760=      REAL A(8,8),B(8,8),C(8,8)
4770=C
4780=      DO 100 I=1,L
4790=      DO 100 J=1,N
4800= 100 C(I,J)=A(I,J)-B(I,J)
4810=C
4820=      RETURN
4830=      END
4840=C
4850=C
4860=      SUBROUTINE ABCD(AA,BB,A,B,C,D,NN,N,M,L)
4870=      REAL AA(16,16),BB(16,16),A(8,8),B(8,8),
4880=      +C(8,8),D(8,8)
4890=C
4900=C      THIS SUBROUTINE SETS UP THE MATRICES AA AND BB
4910=C      WHERE AA=( A,B ), AND BB= ( I,O )
4920=C              ( C,D ),          ( O,O )
4930=C
4940=      DO 200 I=1,NN
4950=      DO 200 J=1,NN

```

```

4960=      AA(I,J)=0.
4970= 200  BB(I,J)=0.
4980=C
4990=      DO 210 I=1,N
5000= 210  BB(I,I)=1.
5010=C
5020=      DO 220 I=1,N
5030=      DO 220 J=1,N
5040= 220  AA(I,J)=A(I,J)
5050=C
5060=      DO 230 I=1,N
5070=      DO 230 J=1,M
5080= 230  AA(I,N+J)=B(I,J)
5090=C
5100=      DO 240 I=1,L
5110=      DO 240 J=1,N
5120= 240  AA(N+I,J)=C(I,J)
5130=C
5140=      DO 250 I=1,L
5150=      DO 250 J=1,M
5160= 250  AA(N+I,N+J)=D(I,J)
5170=C
5180=      RETURN
5190=      END
5200=C
5210=C
5220=      SUBROUTINE QZALGM(N,A,B,ALFR,ALFI,BETA,
5230=      +MATZ,Z,IFAIL)
5240=      LOGICAL MATZ
5250=C
5260=      THIS SUBROUTINE CALLS THE QZ SUBROUTINES FROM THE
5270=      SLATEC (OR EISPACK) LIBRARY
5280=C
5290=      REAL A(16,16),B(16,16),ALFR(16),ALFI(16),BETA(16),Z(16,16)
5300=      ISAVE=IFAIL
5310=      IFAIL=1
5320=      ESP=-1.0E-6
5330=      CALL QZHES(16,N,A,B,MATZ,Z)
5340=      CALL QZIT(16,N,A,B,ESP,MATZ,Z,IFAIL)
5350=      CALL QZVAL(16,N,A,B,ALFR,ALFI,BETA,MATZ,Z)
5360=      IF(IFAIL.NE.0)RETURN
5370=      IF(MATZ)CALL QZVEC(16,N,A,B,ALFR,ALFI,BETA,Z)
5380=      RETURN
5390=      END
5400=C
5410=      SUBROUTINE CAZERO(AA,BB,NN,VV,MATV,ZERR,ZERI,
5420=      +TR,TI,LABR,LABI,IFL,JF,IX)
5430=C
5440=C
5450=      REAL AA(16,16),BB(16,16),ALFR(16),ALFI(16),BETA(16),
5460=      +VV(16,16),LABR(16),LABI(16),ZERR(16),ZERI(16),
5470=      +TR(16),TI(16)
5480=      INTEGER IFL(16)
5490=      LOGICAL MATV

```

```

5500=C
5510=C
5520=C
5530=C
5540=C
5550=C
5560=C
5570=C
5580=C
5590=C
5600=C
5610=C
5620=C
5630=C
5640=C
5650=C
5660=
5670=
5680=C
5690=
5700=C
5710=
5720=
5730=
5740=
5750=
5760=
5770=C
5780=C
5790=
5800=
5810=
5820=
5830=
5840=
5850=
5860=
5870=
5880=
5890=
5900=
5910=
5920=
5930=
5940=
5950=
5960=
5970=C
5980=C
5990=C
6000=
6010=
6020=
6030=

```

THIS SUBROUTINE TAKES DATA COMPUTED BY THE QZ ALGORITHM AND CALCULATES THE SYSTEM ZEROS. IT ALSO DECIDES IF THE NUMBER IS OR ISN'T A SYSTEM ZERO. THIS IS DONE BY USING THE QZ ALGORITHM TWICE AND COMPARING THE OUTPUTS OF EACH RUN. IF THE VALUE IS SAME FOR BOTH RUNS, IT IS A SYSTEM ZERO.

THE VALUE OF IX SIGNALS WHICH SYSTEM ZERO IS BEING CALCULATED.

TRANSMISSION ZEROS IX=1
DECOUPLING ZEROS IX=2

```

IF(IX.EQ.1) EX1=1.0E-8
IF(IX.EQ.2) EX1=1.0E-4
EX2=1.0E-8
IFAIL=1
CALL QZALGM(NN,AA,BB,ALFR,ALFI,BETA,
+MATV,VV,IFAIL)
IF(IFAIL.EQ.0) GOTO 150
PRINT*," RUN FAILED"
150 CONTINUE
DO 120 I=1,NN
LABR(I)=0.
IF(JF.EQ.1) TR(I)=0.
LABI(I)=0.
IF(JF.EQ.1) TI(I)=0.
BT=BETA(I)
IFL(I)=0
IF(ABS(BT).GE.EX2) GOTO 40
GO TO 120
40 IFL(I)=1
LABR(I)=ALFR(I)/BT
LABI(I)=ALFI(I)/BT
IF(JF.EQ.2) GOTO 120
TR(I)=LABR(I)
TI(I)=LABI(I)
120 CONTINUE
IF(JF.EQ.2) GOTO 50
GOTO 1000
50 CONTINUE
COMPARE CALCULATED ZEROS FROM RUNS 1 AND 2
DO 130 I=1,NN
IF(IFL(I).EQ.0) GOTO 130
ZERR(I)=LABR(I)
ZERI(I)=LABI(I)

```

```

6040=C
6050=      AZR=ABS(ZERR(I))
        30=      AZI=ABS(ZERI(I))
6070=C
6080=      K=0
6090=C
6100=      DO 42 J=1,NN
6110=C
6120=      ATR=ABS(TR(J))
6130=      ATI=ABS(TI(J))
6140=C
6150=C
6160=C      COMPARE REAL ZEROS
6170=C
6180=      IF(AZI.GT.EX2) GO TO 400
6190=      IF(ATI.GT.EX2) GO TO 43
6200=      IF(AZR.GT.EX2) GO TO 310
6210=      IF(ZERR(I).LT.TR(J)+EX1.AND.ZERR(I).GT.TR(J)-EX1) GO TO 42
6220=      GO TO 43
6230=C
6240= 310 DT=(ZERR(I)-TR(J))/ZERR(I)
6250=      IF(ABS(DT).LT.EX1) GO TO 42
6260=      GO TO 43
6270=C
6280=C      COMPARE IMMANGINARY ZEROS
6290=C
6300= 400 IF(ATI.LT.EX2) GO TO 43
6310=      IF(AZR.GT.EX2) GO TO 410
6320=      IF(ZERR(I).LT.TR(J)+EX1.AND.ZERR(I).GT.TR(J)-EX1) GO TO 420
6330=      GO TO 43
6340=C
6350= 410 DT=(ZERR(I)-TR(J))/ZERR(I)
6360=      IF(ABS(DT).LT.EX1) GO TO 420
6370=      GO TO 43
6380=C
6390= 420 IF(AZI.GT.EX2) GO TO 430
6400=      IF(ZERI(I).LT.TI(J)+EX1.AND.ZERI(I).GT.TI(J)-EX1) GO TO 42
6410=      GO TO 43
6420=C
6430= 430 DTI=(ZERI(I)-TI(J))/ZERI(I)
6440=      IF(ABS(DTI).LT.EX1) GO TO 42
6450=C
6460= 43 K=K+1
6470= 42 CONTINUE
6480=      IF(K.EQ.NN) IFL(I)=0
6490= 130 CONTINUE
6500=C
6510= 1000 CONTINUE
6520=      RETURN
6530=      END
6540=C
6550=C
6560=      SUBROUTINE TZ(A,B,C,D,NN,N,M,L,MATV,TRAR,TRAI,IFLT)
        70=C

```

```

6580=C
6590=      REAL AA(16,16),BB(16,16),A(8,8),B(8,8),C(8,8),D(8,8),
6600=      +TRAR(16),TRAI(16),TR(16),TI(16),LABR(16),LABI(16),VV(16,16)
6610=      INTEGER IFLT(16)
6620=C
6630=      LOGICAL MATV
6640=C
6650=      THIS SUBROUTINE AND THE SUBROUTINES CALLED BY IT
6660=C      COMPUTE THE TRANSMISSION ZEROS OF A SYSTEM
6670=C
6680=      DO 100 II=1,2
6690=      CALL ABCD(AA,BB,A,B,C,D,NN,N,M,L)
6700=      IF(L.LT.M) GO TO 200
6710=C
6720=      DO 110 I=1,L
6730= 110 AA(N+I,N+M+I)=1.
6740=C
6750=      GO TO 300
6760=C
6770= 200 CONTINUE
6780=C
6790=      DO 210 I=1,M
6800= 210 AA(N+L+I,N+I)=1.
6810=C
6820= 300 CONTINUE
6830=C
6840=      DO 220 I=1,M
6850=      DO 220 J=1,L
6860= 220 AA(N+L+I,N+M+J)=RANF(DUM)*10.
6870=C
6880=C
6890=      JF=II
6900=      CALL CAZERO(AA,BB,NN,VV,MATV,TRAR,TRAI,
6910=      +TR,TI,LABR,LABI,IFLT,JF,1)
6920= 100 CONTINUE
6930=C
6940=      RETURN
6950=      END
6960=C
6970=      SUBROUTINE DZ(A,B,C,D,NN,N,M,L,MATV,DECR,
6980=      +DECI,IFL)
6990=C
7000=C
7010=      INTEGER IFL(16)
7020=C
7030=      THIS SUBROUTINE AND THE SUBROUTINES CALLED BY
7040=C      IT COMPUTE THE DECOUPLING ZEROS OF A SYSTEM.
7050=C
7060=      REAL AA(16,16),BB(16,16),A(8,8),B(8,8),C(8,8)
7070=      +,D(8,8),TR(16),TI(16),VV(16,16),
7080=      2LABR(16),LABI(16),DECR(16),DECI(16)
7090=      LOGICAL MATV
7100=C
7110=      DO 100 II=1,2

```

```

7120=      CALL ABCD(AA,BB,A,B,C,D,NN,N,M,L)
7130=C
7140=      DO 110 I=1,L
7150= 110 AA(N+I,N+M+I)=1.
7160=C
7170=      DO 120 I=1,M
7180= 120 AA(N+L+I,N+I)=1.
7190=C
7200=      DO 130 I=1,M
7210=      DO 130 J=1,L
7220= 130 AA(N+L+I,N+M+J)=RANF(DUM)*10.
7230=C
7240=C
7250=      JF=II
7260=      CALL CAZERO(AA,BB,NN,VV,MATV,DECR,DECI,
7270=      +TR,TI,LABR,LABI,IFL,JF,2)
7280= 100 CONTINUE
7290=      RETURN
7300=      END
7310=C
7320=      SUBROUTINE RDZ(A,B,C,Z,N,M,L,KIO)
7330=      REAL A(8,8),B(8,8),C(8,8),S(8),SQ(8)
7340=      REAL U(12,12),UD(12,12),SI(8,8),SIA(8,8)
7350=      REAL DCIN(12,12),DCOT(12,12),WK(12),WKO(12)
7360=C
7370=C      THIS SUBROUTINE DETERMINES WEATHER A DECOUPLING ZERO
7380=C      IS AN INPUT,OUTPUT,OR INPUT/OUTPUT DECOUPLING
7390=C      ZERO. THIS IS DONE BY TAKING THE SINGULAR VALUE
7400=C      DECOMPOSITION OF THE MATRICES (SI-A,B) AND (SI-A)
7410=C      ( C )
7420=C      TO TEST FOR FULL RANK
7430=C
7440=      MO=N+M
7450=      LO=N+L
7460=C
7470=      EX1=1.0E-1
7480=      EX2=1.0E-5
7490=C
7500=      DO 190 I=1,N
7510=      DO 190 J=1,N
7520= 190 SI(I,J)=0.
7530=C
7540=      DO 200 I=1,N
7550= 200 SI(I,I)=Z
7560=C
7570=      CALL MSUB(SI,A,SIA,N,N)
7580=C
7590=      DO 210 I=1,N
7600=      DO 210 J=1,N
7610=      DCIN(I,J)=SIA(I,J)
7620= 210 DCOT(I,J)=SIA(I,J)
7630=C
7640=      DO 220 I=1,N
7650=      DO 220 J=1,M

```

```

7660= 220 DCIN(I,J+N)=B(I,J)
7670=C
7680= DO 230 I=1,L
7700= DO 230 J=1,N
7700= 230 DCOT(I+N,J)=C(I,J)
7710=C
7720= CALL LSVDF(DCIN,12,N,M0,U,12,0,S,WK,IER)
7730=C
7740= KIO=0
7750= KI=0
7760= DO 300 I=1,N
7770= IF(S(I).GT.EX1) GO TO 300
7780= IF(S(I).LT.-EX1) GO TO 300
7790= IF(S(I).LT.EX2.AND.S(I).GT.-EX2) GO TO 350
7800= GO TO 300
7810= 350 KI=1
7820= 300 CONTINUE
7830=C
7840= CALL LSVDF(DCOT,12,L0,N,U0,12,0,SO,WK0,IER0)
7850=C
7860= KOT=0
7870= DO 310 I=1,N
7880= IF(S0(I).GT.EX1) GO TO 310
7890= IF(S0(I).LT.-EX1) GO TO 310
7900= IF(S0(I).LT.EX2.AND.S0(I).GT.-EX2) GO TO 360
7910= GO TO 310
7920= 360 KOT=1
7930= 310 CONTINUE
7940=C
7950= IF(KI.EQ.1) KIO=1
7960= IF(KOT.EQ.1) KIO=2
7970= IF(KI.EQ.1.AND.KOT.EQ.1) KIO=3
7980= RETURN
7990= END
8000=C
8010=C
8020= SUBROUTINE SYSZERO(A,B,C,D,Z,N,M,L,IZ,IX)
8030= REAL A(8,8),B(8,8),C(8,8),D(8,8),S(8),WK(12)
8040= REAL SYSM(12,12),SI(8,8),SIA(8,8),U(12,12)
8050=C
8060=C THIS SUBROUTINE DETERMINS THE INVARIANT ZEROS
8070=C ALL TRANSMISSION ZEROS ARE INVARIANT
8080=C SOME DECOUPLING ZEROS ARE INVARIANT
8090=C THE SUBROUTINE TAKES THE SYSTEM MATRIX ( SI-A , -B)
8100=C ( C , D)
8110=C AND TESTS FOR FULL RANK BY SINGULAR VALUE DECOMPOSITION
8120=C AN INVARIANT ZERO CAUSES THE SYSTEM MATIX TO LOOSE RANK
8130=C
8140= IZ=0
8150=C
8160= M0=N+M
8170= L0=N+L
8180=C
8190= IF(IX.EQ.2) GO TO 20

```

```

8200=C
8210=      EX1=1.0E-1
8220=      EX2=1.0E-9
8230=      GO TO 25
8240=C
8250=      20 EX1=1.0E-1
8260=      EX2=1.0E-5
8270=      25 CONTINUE
8280=C
8290=      DO 100 I=1,L0
8300=      DO 100 J=1,M0
8310=      100 SYSM(I,J)=0.0
8320=C
8330=      DO 110 I=1,N
8340=      DO 110 J=1,M
8350=      110 SYSM(I,N+J)=-B(I,J)
8360=C
8370=      DO 120 I=1,L
8380=      DO 120 J=1,N
8390=      120 SYSM(N+I,J)=C(I,J)
8400=C
8410=      DO 130 I=1,L
8420=      DO 130 J=1,M
8430=      130 SYSM(N+I,N+J)=D(I,J)
8440=C
8450=C
8460=      DO 200 I=1,N
8470=      DO 200 J=1,N
8480=      200 SI(I,J)=0.0
8490=C
8500=      DO 210 I=1,N
8510=      210 SI(I,I)=Z
8520=C
8530=      CALL MSUB(SI,A,SIA,N,N)
8540=C
8550=      DO 220 I=1,N
8560=      DO 220 J=1,N
8570=      220 SYSM(I,J)=SIA(I,J)
8580=C
8590=      CALL LSVDF(SYSM,12,L0,M0,U,12,0,S,WK,IER)
8600=C
8610=      IF(L0.GT.M0) GO TO 320
8620=C
8630=      DO 300 I=1,L0
8640=      IF(S(I).GT.EX1) GO TO 300
8650=      IF(S(I).LT.-EX1) GO TO 300
8660=      IF(S(I).LT.EX2.AND.S(I).GT.-EX2) IZ=1
8670=      300 CONTINUE
8680=C
8690=      GO TO 330
8700=C
8710=      320 DO 310 I=1,M0
8720=      IF(S(I).GT.EX1) GO TO 310
8730=      IF(S(I).LT.-EX1) GO TO 310

```

```

8740=      IF(S(I).LT.EX2.AND.S(I).GT.-EX2) IZ=1
8750= 310 CONTINUE
8760=C
8770= 330 CONTINUE
8780=C
8790=      RETURN
8800=      END
8810=C
8820=C
8830=      SUBROUTINE COMPLEX(NN,ZERI,IFL)
8840=      REAL ZERI(16)
8850=      INTEGER IFL(16)
8860=C
8870=C      THIS SUBROUTINE TESTS FOR COMPLEX CONJIGATE PAIRS.
8880=C      A COMPLEX SYSTEM ZERO WILL ALWAYS OCCUR IN CONJIGATE PAIRS
8890=C      ANY COMPLEX ZERO CALCULATED WITHOUT A CONJIGATE PAIR WILL BE
8900=C      REJECTED AS A SYSTEM ZERO
8910=C
8920=      EX1=1.0E-6
8930=C
8940=      DO 100 I=1,NN
8950=      IF(IFL(I).EQ.0) GO TO 100
8960=      IF(ZERI(I).LT.EX1.AND.ZERI(I).GT.-EX1) GO TO 100
8970=C
8980=      DO 110 J=1,NN
8990=      IF(IFL(J).EQ.0) GO TO 110
9000=      IF(ZERI(J).LT.EX1.AND.ZERI(J).GT.-EX1) GO TO 110
9010=C
9020=      CCP=ZERI(I)
9030=      CCN=ZERI(J)
9040=      CCC=CCP+CCN
9050=      IF(CCC.LT.EX1.AND.CCC.GT.-EX1) GO TO 120
9060=      GO TO 130
9070= 120 IFL(I)=1
9080=      IFL(J)=1
9090=      GO TO 100
9100= 130 IFL(I)=0
9110=C
9120= 110 CONTINUE
9130= 100 CONTINUE
9140=C
9150=      RETURN
9160=      END
9170=*EOR
9180=*EOF

```

Appendix C

Main Computer Program

This appendix contains a listing of the computer program used to design controllers and simulate them. The program is adopted from the program used by Lt Brett Ridgely (Ref 10). A simplified flow chart is shown in Figure C-1. The program listed is for designs with actuator dynamics. For designs without actuator dynamics lines 2840 through 3450 are replaced by the lines 3300 through 4380 shown at the end of the main program listing.

The example is design 2A simulated at the design condition (5,000 ft, Mach 0.6) and also at the condition of design 3 (10,000 ft, Mach 0.9).

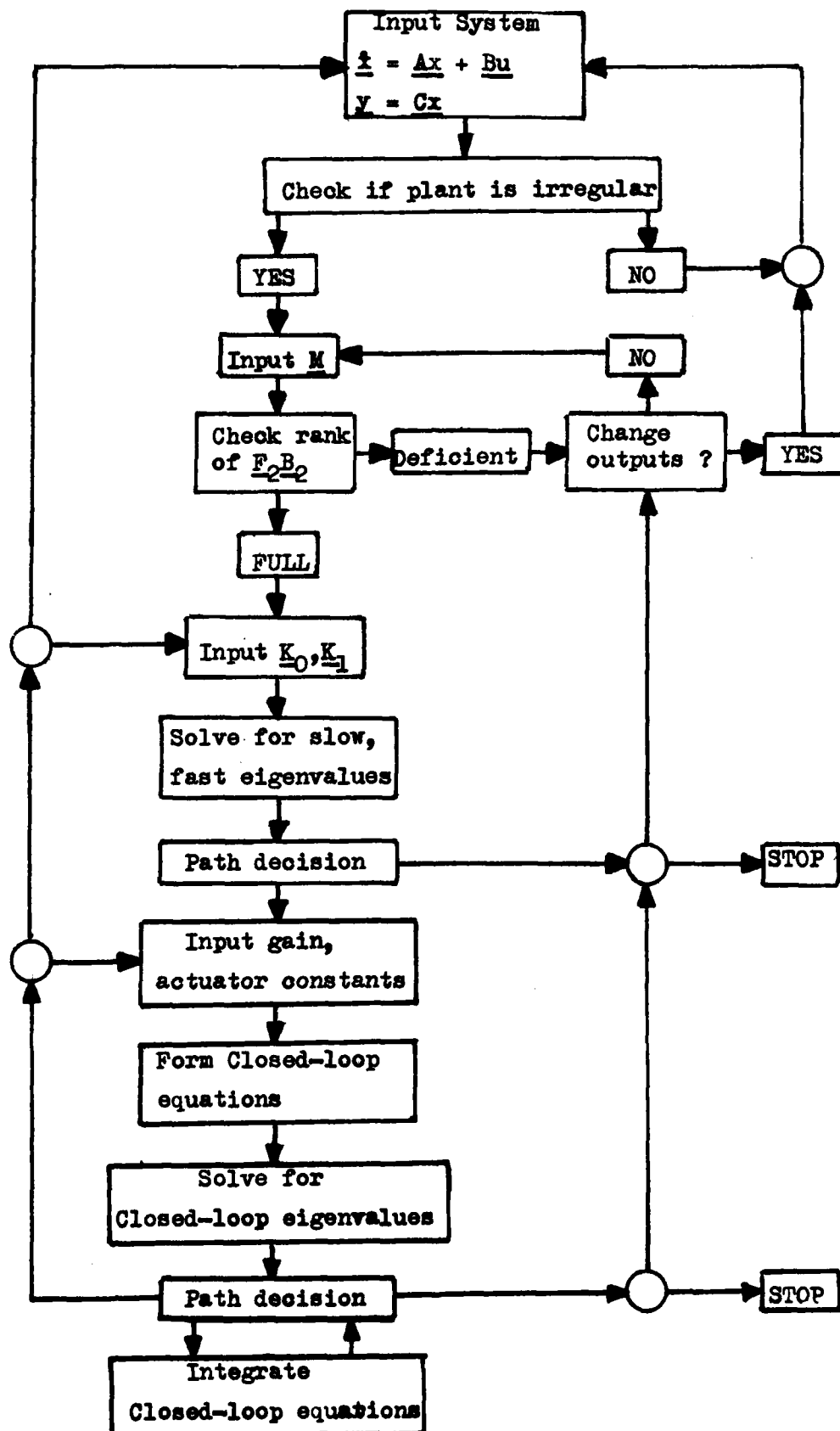


Figure C-1 Main Computer Program Flow Chart

WELCOME TO THE SINGULARLY PERTURBED ASYMPTOTIC METHOD
(SPAM FOR SHORT) FOR IRREGULAR PLANTS - VERSION 3.2
MAKE SURE BOTH THE IMSL AND CC6600 LIBRARIES ARE SET

DESIGN 1 (1) OTHER (2) :
450008 CM STORAGE USED
1.812 CP SECONDS COMPILATION TIME
NON-FATAL LOADER ERRORS -
NON-EXISTENT LIBRARY GIVEN - SYSIO 1

THE A11 MATRIX IS -

0.0000

THE A12 MATRIX IS -

0.0000 1.0000 .0318

THE A21 MATRIX IS -

.0489
0.0000
0.0000

THE A22 MATRIX IS -

-.3744 .0318 -.9995
-39.9283 -2.1138 .0797
6.2379 -.0063 -.7074

THE B1 MATRIX IS A 1 BY 3 ZERO MATRIX

THE B2 MATRIX IS -

.0531 .0046 .0250
9.7929 -55.7388 2.6033
-5.0544 -1.8927 4.3880

SYSTEM DIMENSION = 4
INPUTS/OUTPUTS = 3

ENTER C1 (3 BY 1)

ROW 1 : 0
ROW 2 : 1
ROW 3 : 0

ENTER C2 (3 BY 3)

ROW 1 : 1,0,0
ROW 2 : 0,0,0
ROW 3 : 0,0,1

THIS METHOD IS FOR IRREGULAR PLANTS, THAT IS, FOR
C2B2 NOT OF FULL RANK

*** TERMINAL ERROR (IER = 129) FROM IMSL ROUTINE LUDATF
*** TERMINAL ERROR (IER = 130) FROM IMSL ROUTINE LINV3F
THE PLANT IS IRREGULAR

ENTER THE M MATRIX (3 BY 1) :

ROW 1 : 0
ROW 2 : .25
ROW 3 : 0

THE MATRIX F1 IS -

0.0000
1.0000
0.0000

THE MATRIX F2 IS -

1.0000	0.0000	0.0000
0.0000	.2500	.0080
0.0000	0.0000	1.0000

THE MATRIX F2*B2 IS -

.0531	.0046	.0250
2.4080	-13.9498	.5857
-5.0544	-1.8927	4.3880

NOW VALUES OF SIGMA MUST BE CHOSEN. HOW DO YOU WANT
TO INPUT THESE (0=DIAGONAL, 1=FULL) : 0

ENTER THE 3 VALUES OF SIGMA (MUST BE POSITIVE) : 1,15,30

THE MATRIX K0 IS -

11.6944	.1981	-2.0615
2.7389	-1.0522	-.1395
14.6517	-.2256	4.4020

THE MATRIX K1 MUST NOW BE INPUT. HOW DO YOU WISH
TO DO THIS (0=SCRATCH, 1=MULTIPLE OF K0) : 1

WHAT MULTIPLE OF K0 IS K1 : 2

THE K1 MATRIX IS -

23.3888	.3962	-4.1231
5.4778	-2.1044	-.2789
29.3035	-.4513	8.8041

THE FIRST SET OF SLOW EIGENVALUES ARE :

LAMBDA(1) = (-2.,0.)

LAMBDA(2) = (-2.,0.)

LAMBDA(3) = (-2.,0.)

THE SECOND SET OF SLOW EIGENVALUES ARE :

LAMBDA(4) = (-4.,0.)

THE FAST EIGENVALUES ARE (MUST BE MULTIPLIED BY G) :

LAMBDA(5) = (-1.,0.)

LAMBDA(6) = (-30.,0.)

LAMBDA(7) = (-15.,0.)

FORM CLOSED LOOP MATRICES (1), CHANGE OUTPUTS/EIGENVALUES
(2), OR STOP (0) : 1

PRINT C-L MATRICES (1=YES) : 0

ENTER THE GAIN CONSTANT , G : 29

ENTER THE 3 ACTUATOR CONTANTS :

RUDDER CONSTANT : 20

AILERON CONSTANT : 20

CANNARD CONSTANT : 20

THE OVERALL CLOSED LOOP EIGENVALUES ARE :

EIGENVALUE (1) = (-9.348964874273,131.5128606026)
EIGENVALUE (2) = (-9.348964874273,-131.5128606026)
EIGENVALUE (3) = (-8.041546031385,92.59541875491)
EIGENVALUE (4) = (-8.041546031385,-92.59541875491)
EIGENVALUE (5) = (-9.135660521417,21.57045712481)
EIGENVALUE (6) = (-9.135660521417,-21.57045712481)
EIGENVALUE (7) = (-4.027317394378,0.)
EIGENVALUE (8) = (-2.104785028134,0.)
EIGENVALUE (9) = (-2.010762726775,0.)
EIGENVALUE (10) = (-2.000431356545,0.)

CHANGE OUTPUTS/EIGENVALUES (1), CHANGE ONLY GAIN (2),
CHANGE SIGMA (K0/K1) (3), INTEGRATE C-L EGNS (4),
CHANGE PLANT (5), STOP (0) : 4

***** ENTER INTEGRATING ROUTINE *****
FOR THIS SYSTEM THERE ARE 3 INPUTS AND 3 OUTPUTS
DO YOU WISH TO SEE THE INPUT HISTORIES OR OUTPUT
HISTORIES ? (1=INPUT,2=OUTPUT) : 2

ENTER MANEUVER (Y.P.=1,W.L.T.=2,H.T.=3) : 1

ENTER INPUT COMMAND

FINAL VALUE, RAMP TIME (UNIT STEP = 1.0)

1 : -3.1

2 : 0.0

ENTER STEP RESPONSE END TIME (INFINITY=0) : 0

3 : 3.0

ENTER STEP RESPONSE END TIME (INFINITY=0) : 1

ENTER FINAL TIME AND TIME STEP : 1.75,.05

T	OUTPUTS				
0.00	0.0000	0.0000	0.0000	0.0000	0.0000
.05	-.1936	.0014	1.1608	.0580	-.1356
.10	-.3831	.0028	1.9707	.1566	-.2266
.15	-.5369	.0044	2.4866	.2809	-.2560
.20	-.6728	.0062	2.7875	.4203	-.2525
.25	-.8110	.0079	2.9457	.5676	-.2434
.30	-.9563	.0095	3.0173	.7184	-.2379
.35	-1.1077	.0112	3.0411	.8705	-.2372
.40	-1.2591	.0129	3.0414	1.0226	-.2365
.45	-1.4072	.0144	3.0323	1.1742	-.2330
.50	-1.5562	.0159	3.0213	1.3252	-.2310
.55	-1.7034	.0174	3.0115	1.4758	-.2276
.60	-1.8586	.0188	3.0040	1.6260	-.2325
.65	-2.0098	.0201	2.9999	1.7759	-.2339
.70	-2.1550	.0212	2.9955	1.9257	-.2293
.75	-2.3016	.0223	2.9934	2.0754	-.2262
.80	-2.4463	.0233	2.9923	2.2250	-.2213
.85	-2.6002	.0242	2.9916	2.3746	-.2256
.90	-2.7491	.0251	2.9912	2.5242	-.2249
.95	-2.8976	.0259	2.9910	2.6737	-.2239
1.00	-3.0552	.0267	2.9909	2.8233	-.2319
1.05	-3.0155	.0261	1.7695	2.9117	-.1038
1.10	-2.9676	.0254	1.2151	2.9725	.0049
1.15	-2.9602	.0241	.7618	3.0106	.0504
1.20	-2.9779	.0229	.4358	3.0324	.0545
1.25	-2.9932	.0217	.2229	3.0435	.0503
1.30	-2.9964	.0205	.0954	3.0483	.0519
1.35	-2.9930	.0192	.0261	3.0496	.0566
1.40	-2.9903	.0178	-.0069	3.0492	.0589
1.45	-2.9907	.0166	-.0193	3.0483	.0576
1.50	-2.9925	.0153	-.0211	3.0472	.0547
1.55	-2.9940	.0142	-.0183	3.0463	.0523
1.60	-2.9946	.0131	-.0142	3.0456	.0510
1.65	-2.9949	.0120	-.0104	3.0451	.0502
1.70	-2.9952	.0110	-.0073	3.0447	.0495
1.75	-2.9957	.0101	-.0051	3.0444	.0488

BETA COMMAND =-1.633031846067

R COMMAND =2.823251138512

PHI =.02907321598383

BETA TRANS. =.01173669636142

R TRANS. =.3810131352851

HANGE OUTPUTS/EIGENVALUES (1), CHANGE ONLY GAIN (2),
CHANGE SIGMA (K0/K1) (3), INTEGRATE C-L EGNS (4),
CHANGE PLANT (5), STOP (0) : 5

NEW DESIGN (1) OR ROBUSTNESS TEST (2) ? : 2

ENTER THE DIMENSION OF THE SYSTEM : 4

ENTER THE NUMBER OF ACTIVE INPUTS (SIZE OF B2) : 3

ENTER A11 (1 BY 1)

ROW 1 : 0

ENTER A12 (1 BY 3)

ROW 1 : 0.1,.0255

ENTER A21 (3 BY 1)

ROW 1 : .0332

ROW 2 : 0

ROW 3 : 0

ENTER A22 (3 BY 3)

ROW 1 : -.4836,.0255,-.9997

ROW 2 : -75.3125,-3.9178,.0382

ROW 3 : 10.6995,-.0294,-.5117

B1 IS A ZERO MATRIX OF DIMENSION 1 BY 3

ENTER B2 (3 BY 3)

ROW 1 : .0473,.0066,.0399

ROW 2 : 13.4743,-82.0763,8.3887

ROW 3 : -7.7774,-3.2921,8.7542

*** TERMINAL ERROR (IER = 129) FROM IMSL ROUTINE LUDATF

*** TERMINAL ERROR (IER = 130) FROM IMSL ROUTINE LINV3F

THE PLANT IS IRREGULAR

THE MATRIX F1 IS -

0.0000

1.0000

0.0000

THE MATRIX F2 IS -

1.0000

0.0000

0.0000

0.0000

.2500

0.0000

0.0000

.0064

1.0000

HE MATRIX F2*B2 IS -

.0473	.0066	.0399
3.3190	-20.5401	2.1530
-7.7774	-3.2921	8.7542

THE MATRIX K0 IS -

11.6944	.1981	-2.0615
2.7389	-1.0522	-.1395
14.6517	-.2256	4.4020

THE K1 MATRIX IS -

23.3888	.3962	-4.1231
5.4778	-2.1044	-.2789
29.3035	-.4513	8.8041

THE FIRST SET OF SLOW EIGENVALUES ARE :

LAMBDA(1) = (-2.,0.)

LAMBDA(2) = (-2.,0.)

LAMBDA(3) = (-2.,0.)

THE SECOND SET OF SLOW EIGENVALUES ARE :

LAMBDA(4) = (-4.,0.)

THE FAST EIGENVALUES ARE (MUST BE MULTIPLIED BY G) :

LAMBDA(5) = (-1.11882838742,0.)

LAMBDA(6) = (-55.06024604491,0.)

LAMBDA(7) = (-21.78931709532,0.)

THE OVERALL CLOSED LOOP EIGENVALUES ARE :

EIGENVALUE (1) = (-9.511823000371,178.4317581339)
EIGENVALUE (2) = (-9.511823000371,-178.4317581339)
EIGENVALUE (3) = (-8.918649595212,111.9021828101)
EIGENVALUE (4) = (-8.918649595212,-111.9021828101)
EIGENVALUE (5) = (-8.963727600824,23.04292622563)
EIGENVALUE (6) = (-8.963727600824,-23.04292622563)
EIGENVALUE (7) = (-3.951230438002,0.)
EIGENVALUE (8) = (-2.121727763373,0.)
EIGENVALUE (9) = (-2.005870702296,.002953256932167)
EIGENVALUE (10) = (-2.005870702296,-.002953256932167)

**

NTER INTEGRATING ROUTINE *****
 FOR THIS SYSTEM THERE ARE 3 INPUTS AND 3 OUTPUTS
 DO YOU WISH TO SEE THE INPUT HISTORIES OR OUTPUT
 HISTORIES ? (1=INPUT,2=OUTPUT) : 2
 ENTER MANEUVER (Y.P.=1,W.L.T.=2,H.T.=3) : 1

ENTER INPUT COMMAND
 FINAL VALUE, RAMP TIME (UNIT STEP = 1,0)
 1 : -3,.75
 2 : 0,0
 ENTER STEP RESPONSE END TIME (INFINITY=0) : 0
 3 : 4,0
 ENTER STEP RESPONSE END TIME (INFINITY=0) : .75

ENTER FINAL TIME AND TIME STEP : 1.5,.05

T	OUTPUTS				
0.00	0.0000	0.0000	0.0000	0.0000	0.0000
.05	-.2520	-.0216	6.0860	.3043	.0523
.10	-.5291	-.0078	3.2783	.4682	-.0609
.15	-.7149	-.0059	4.0252	.6695	-.0454
.20	-.8935	.0076	4.2972	.8843	-.0091
.25	-1.0717	.0090	3.6887	1.0688	-.0029
.30	-1.2708	.0118	4.2317	1.2804	.0096
.35	-1.4809	.0139	3.8643	1.4736	-.0074
.40	-1.6837	.0157	4.0574	1.6764	-.0073
.45	-1.8823	.0184	3.9882	1.8759	-.0065
.50	-2.0740	.0201	3.9926	2.0755	.0015
.55	-2.2712	.0218	4.0049	2.2757	.0045
.60	-2.4731	.0234	3.9821	2.4748	.0017
.65	-2.6779	.0253	3.9960	2.6746	-.0033
.70	-2.8755	.0273	3.9971	2.8745	-.0010
.75	-3.0723	.0297	3.9862	3.0738	.0015
.80	-3.0254	.0543	-1.8588	2.9809	-.0446
.85	-2.9424	.0390	.4810	3.0049	.0625
.90	-2.9556	.0379	.1216	3.0110	.0554
.95	-2.9753	.0255	-.3879	2.9916	.0163
1.00	-2.9935	.0247	.3308	3.0081	.0147
1.05	-2.9927	.0231	-.2394	2.9962	.0034
1.10	-2.9859	.0216	.1051	3.0014	.0155
1.15	-2.9858	.0209	-.0525	2.9988	.0130
1.20	-2.9877	.0190	-.0132	2.9981	.0104
1.25	-2.9911	.0180	.0088	2.9986	.0075
1.30	-2.9922	.0166	.0263	2.9973	.0050
1.35	-2.9925	.0154	.0059	2.9976	.0050
1.40	-2.9929	.0141	-.0141	2.9968	.0039
1.45	-2.9936	.0128	-.0033	2.9967	.0031
1.50	-2.9944	.0118	-.0058	2.9964	.0019

BETA COMMAND =-1.261148520343
 R COMMAND =3.073795857869
 PHI =.03064685696701
 BETA TRANS. =.01610555728732
 R TRANS. =.3820396586508

```

100=      PROGRAM PORTER(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
110=      1 PLOT,TAPE4=PLOT)
120=C
130=      REAL A11(10,10),A12(10,10),A21(10,10),A22(10,10)
140=      REAL B1(10,10),B2(10,10),C1(10,10),C2(10,10)
150=      REAL F1(10,10),F2(10,10),M(10,10),UC(10)
160=      REAL K0(10,10),K1(10,10),X(10),Y(10),Z(10)
170=      REAL A12FF(10,10),V(10),C2B2(10,10)
180=      REAL WKAREA(10),MA11(10,10),MA12(10,10),F2B2(10,10)
190=      REAL SIGMA(10,10),FBINV(10,10),K00(10,10),K11(10,10)
200=      REAL BETA(10),SLOW2(10),AFIF(10,10),F2IF1(10,10),F2INV(10,10)
210=      COMPLEX ALFA(10),LAMBDA(10),LAM(10),W(10),W1(10),ZQ(10,10)
220=      REAL AY(10,10),BEE(10,10),CEE(10,10)
230=      REAL AC(10,10),ACK1(10,10),ACKF1(10,10),ACKF2(10,10),ACK0(10,10)
240=      REAL FB2(10,10),F2B2K0(10,10),FAST(10,10)
250=      REAL IDEN(10,10),F2S(10,10),AY1(10,10)
260=      REAL XP(10),MULT,ZDOT(10),U(10),ER(5),RR(2)
270=      REAL VC(10),VDT(10),SLOPE(10),VST(10)
280=      INTEGER KVR(10),KVST(10)
290=      DIMENSION WORK(500),IWORK(10)
300=      COMMON AY,BEE,V,NL,L
310=      EXTERNAL F
320=C
330=C
340=      DO 99 I=1,3
350=      99 PRINT*," "
360=C
370=      PRINT*,"WELCOME TO THE SINGULARLY PERTURBED ASYMPTOTIC METHOD"
380=      PRINT*,"(SPAM FOR SHORT) FOR IRREGULAR PLANTS - VERSION 3.2"
390=      PRINT*,"MAKE SURE BOTH THE IMSL AND CCG600 LIBRARIES ARE SET"
400=      PRINT*," "
410=C
420=      X10=0
430=      X11=1
440=      PRINT*,"      DESIGN 1 (1)  OTHER (2) : "
450=      READ*,XX
460=      IF(XX.EQ.1) GOTO 10
470=C
480=      9970 IF(X10.NE.5) GO TO 9971
490=      PRINT*," "
500=      PRINT*,"      NEW DESIGN (1) OR ROBUSTNESS TEST (2) ? : "
510=      READ*,X11
520=      9971 CONTINUE
530=      PRINT*," "
540=      PRINT*,"ENTER THE DIMENSION OF THE SYSTEM : "
550=      READ*,N
560=C
570=      IF (N.LE.7) GOTO 100
580=      PRINT*,"THE DIMENSIONS MUST BE CHANGED TO HANDLE THAT"
590=      GOTO 9999
600=C
610=      100 PRINT*,"ENTER THE NUMBER OF ACTIVE INPUTS (SIZE OF B2) : "
620=      READ*,L
630=      NM=N-L

```

```

640=C
650= PRINT*," ENTER A11 ( ",NM," BY ",NM," )"
660= CALL READM(NM,NM,A11)
670= PRINT*," "
680= PRINT*," ENTER A12 ( ",NM," BY ",L," )"
690= CALL READM(NM,L,A12)
700= PRINT*," "
710= PRINT*," ENTER A21 ( ",L," BY ",NM," )"
720= CALL READM(L,NM,A21)
730= PRINT*," "
740= PRINT*," ENTER A22 ( ",L," BY ",L," )"
750= CALL READM(L,L,A22)
760=C
770= PRINT*," "
780= PRINT*,"B1 IS A ZERO MATRIX OF DIMENSION ",NM," BY ",L
790=C
800= DO 200 I=1,NM
810= DO 200 J=1,L
820= 200 B1(I,J)=0.
830=C
840= PRINT*," "
850= PRINT*," ENTER B2 ( ",L," BY ",L," )"
860= CALL READM(L,L,B2)
870=C
880= IF(X11.EQ.2) GO TO 9972
890=C
900= GOTO 20
910=C
920= 10 CALL DSGN1(N,L,A11,A12,A21,A22,B1,B2)
930= PRINT*," "
940= PRINT*," SYSTEM DIMENSION = ",N
950= PRINT*," # INPUTS/OUTPUTS = ",L
960= PRINT*," "
970= NM=N-L
980= GOTO 20
990=C
1000= 30 PRINT*," "
1010= PRINT*,"NEW OUTPUT MATRICES ?(1=YES) :"
1020= READ*,X5
1030= IF (X5.NE.1) GOTO 380
1040=C
1050= 20 PRINT*," "
1060= PRINT*," ENTER C1 ( ",L," BY ",NM," )"
1070= CALL READM(L,NM,C1)
1080= PRINT*," "
1090= PRINT*," ENTER C2 ( ",L," BY ",L," )"
1100= CALL READM(L,L,C2)
1110=C
1120= PRINT*," "
1130= PRINT*,"THIS METHOD IS FOR IRREGULAR PLANTS, THAT IS, FOR"
1140= PRINT*,"C2B2 NOT OF FULL RANK"
1150= 9972 CALL VMULFF(C2,B2,L,L,L,10,10,C2B2,10,IER)
1160= D1=1.
1170= CALL LINV3F(C2B2,B,4,L,10,D1,D2,WKAREA,IER)

```

```

1180=      DET=D1*2**D2
1190=      IF (DET.EQ.0) GOTO 280
1200=      PRINT*,"THIS PLANT IS REGULAR - PROGRAM TERMINATED"
1210=      GOTO 9999
1220= 280 PRINT*,"THE PLANT IS IRREGULAR"
1230=      PRINT*," "
1240=      IF(X11.EQ.2) GO TO 9973
1250=C
1260= 380 CONTINUE
1270= 610 PRINT*," "
1280=      PRINT*," "
1290=      PRINT*,"ENTER THE M MATRIX ( ",L," BY ",NM," ) : "
1300=      CALL READM(L,NM,M)
1310=      PRINT*," "
1320= 9973 CALL VMULFF(M,A11,L,NM,NM,10,10,MA11,10,IER)
1330=      CALL MADD(C1,MA11,L,NM,F1)
1340=      CALL VMULFF(M,A12,L,NM,L,10,10,MA12,10,IER)
1350=      CALL MADD(C2,MA12,L,L,F2)
1360=      CALL MADD(C2,MA12,L,L,F2S)
1370=      CALL VMULFF(F2,B2,L,L,L,10,10,F2B2,10,IER)
1380=      PRINT*," "
1390=      PRINT*,"THE MATRIX F1 IS -"
1400=      CALL PRINTM(L,NM,F1)
1410=      PRINT*," "
1420=      PRINT*,"THE MATRIX F2 IS -"
1430=      CALL PRINTM(L,L,F2)
1440=      PRINT*," "
1450=      PRINT*,"THE MATRIX F2*B2 IS -"
1460=      CALL PRINTM(L,L,F2B2)
1470=      D1=1
1480=      CALL LINV3F(F2B2,B,4,L,10,D1,D2,WKAREA,IER)
1490=      DET=D1*2**D2
1500=      IF (DET.NE.0) GOTO 370
1510=      PRINT*,"THE MATRIX F2*B2 IS NOT FULL RANK - TRY AGAIN"
1520=      GOTO 30
1530= 370 CONTINUE
1540=      IF(X11.EQ.2) GO TO 9974
1550=      PRINT*," "
1560=      PRINT*,"NOW VALUES OF SIGMA MUST BE CHOSEN. HOW DO YOU WANT"
1570=      PRINT*,"TO INPUT THESE (0=DIAGONAL, 1=FULL) : "
1580=      READ*,JJ
1590=      IF (JJ.EQ.1) GOTO 390
1600=C
1610=      DO 393 I=1,L
1620=      DO 393 J=1,L
1630= 393 SIGMA(I,J)=0
1640=C
1650=      PRINT*," "
1660=      PRINT*,"ENTER THE ",L," VALUES OF SIGMA (MUST BE POSITIVE) : "
1670=C
1680=      DO 400 I=1,L
1690= 400 READ*,SIGMA(I,I)
1700=C
1710=      GOTO 410

```

```

1720=C
1730= 390 PRINT*," "
1740= PRINT*,"ENTER SIGMA -"
1750= CALL READM(L,L,SIGMA)
1760=C
1770= 9974 CONTINUE
1780= 410 CALL VMULFF(F2,B2,L,L,L,10,10,F2B2,10,IER)
1790=C
1800= DO 392 I=1,L
1810= DO 392 J=1,L
1820= 392 FB2(I,J)=F2B2(I,J)
1830=C
1840= IF(X11.EQ.2) GO TO 9975
1850= CALL LINV1F(F2B2,L,10,FBINV,10,WKAREA,IER)
1860= CALL VMULFF(FBINV,SIGMA,L,L,L,10,10,K0,10,IER)
1870=C
1880= 9975 PRINT*," "
1890= PRINT*,"THE MATRIX K0 IS -"
1900= CALL PRINTM(L,L,K0)
1910=C
1920= DO 460 I=1,L
1930= DO 460 J=1,L
1940= 460 K00(I,J)=-K0(I,J)
1950=C
1960= IF(X11.EQ.2) GO TO 9976
1970= PRINT*," "
1980= PRINT*,"THE MATRIX K1 MUST NOW BE INPUT. HOW DO YOU WISH"
1990= PRINT*,"TO DO THIS (0=SCRATCH , 1=MULTIPLE OF K0) : "
2000= READ*,G2
2010= 9976 IF (G2.EQ.1) GO TO 480
2020= 520 CONTINUE
2030= PRINT*," "
2040= PRINT*," ENTER K1 (",L," BY ",L," )"
2050= CALL READM(L,L,K1)
2060=C
2070= GOTO 510
2080= 480 IF (X11.EQ.2) GO TO 9977
2090= PRINT*," "
2100= PRINT*,"WHAT MULTIPLE OF K0 IS K1 : "
2110= READ*,MULT
2120=C
2130= DO 522 I=1,L
2140= DO 522 J=1,L
2150= 522 K1(I,J)=MULT*K0(I,J)
2160=C
2170= 9977 PRINT*," "
2180= PRINT*," THE K1 MATRIX IS -"
2190= CALL PRINTM(L,L,K1)
2200=C
2210= DO 523 I=1,L
2220= DO 523 J=1,L
2230= 523 K11(I,J)=K1(I,J)
2240=C
2250= 510 CALL EIGZF(K11,10,K00,10,L,0,ALFA,BETA,ZQ,10,WKAREA,IER)

```

```

2260=C
2270=      DO 530 I=1,L
2280=      YY=BETA(I)
2290=      IF (YY.EQ.0) GOTO 540
2300=      LAMBDA(I)=ALFA(I)/BETA(I)
2310=      GOTO 530
2320= 540 LAMBDA(I)=9.E+99
2330= 530 CONTINUE
2340=C
2350=      PRINT*," "
2360=      PRINT*,"THE FIRST SET OF SLOW EIGENVALUES ARE :"
2370=      PRINT*," "
2380=C
2390=      DO 550 I=1,L
2400=      PRINT*,"LAMBDA(",I,") = ",LAMBDA(I)
2410= 550 PRINT*," "
2420=C
2430=      CALL LINV1F(F2S,L,10,F2INV,10,WKAREA,IER)
2440=      CALL VMULFF(F2INV,F1,L,L,NM,10,10,F2IF1,10,IER)
2450=      CALL VMULFF(A12,F2IF1,NM,L,NM,10,10,A12FF,10,IER)
2460=C
2470=      DO 560 I=1,NM
2480=      DO 560 J=1,NM
2490= 560 AFIF(I,J)=-A12FF(I,J)
2500=C
2510=      CALL MADD(A11,AFIF,NM,NM,SLOW2)
2520=      CALL EIGRF(SLOW2,NM,10,0,W,ZQ,10,WKAREA,IER)
2530=C
2540=      PRINT*," "
2550=      PRINT*,"THE SECOND SET OF SLOW EIGENVALUES ARE :"
2560=      PRINT*," "
2570=C
2580=      DO 580 I=1,NM
2590=      J=I+L
2600=      PRINT*,"LAMBDA(",J,") = ",W(I)
2610= 580 PRINT*," "
2620=C
2630=      CALL VMULFF(FB2,K0,L,L,L,10,10,F2B2K0,10,IER)
2640=C
2650=      DO 524 I=1,L
2660=      DO 524 J=1,L
2670= 524 FAST(I,J)=-F2B2K0(I,J)
2680=C
2690=      CALL EIGRF(FAST,L,10,0,W1,ZQ,10,WKAREA,IER)
2700=C
2710=      PRINT*,"THE FAST EIGENVALUES ARE (MUST BE MULTIPLIED BY G) :"
2720=      PRINT*," "
2730=      DO 590 I=1,L
2740=      J=NM+L+I
2750=      PRINT*,"LAMBDA(",J,") = ",W1(I)
2760= 590 PRINT*," "
2770=      IF(X11.EQ.2) GO TO 9978
2780=C
2790=      PRINT*,"FORM CLOSED LOOP MATRICES (1), CHANGE OUTPUTS/EIGENVALUES"

```

```

2800=      PRINT*," (2),  OR STOP (0) : "
2810=      READ*,X3
2820=      IF (X3.EQ.2) GOTO 30
2830=      IF (X3.EQ.0) GOTO 9999
2840= 3999 PRINT*," "
2850=      PRINT*,"PRINT C-L MATRICES (1=YES) : "
2860=      READ*,X25
2870=      PRINT*," "
2880=      PRINT*,"ENTER THE GAIN CONSTANT , G : "
2890=      READ*,G
2900=C
2910=      DO 325 I=1,L
2920=      DO 325 J=1,L
2930= 325 AC(I,J)=0
2940=C
2950=      PRINT*," "
2960=      PRINT*,"ENTER THE ",L," ACTUATOR CONSTANTS : "
2970=      PRINT*,"RUDDER CONSTANT : "
2980=      READ*,AC(1,1)
2990=      PRINT*,"AILERON CONSTANT : "
3000=      READ*,AC(2,2)
3010=      PRINT*,"CANNARD CONSTANT : "
3020=      READ*,AC(3,3)
3030=C
3040= 9978 CONTINUE
3050=      I1=N+L+L
3060=      I2=L+NM
3070=      I3=L+L+NM
3080=C
3090=      DO 600 I=1,I1
3100=      DO 620 J=1,I1
3110= 620 AY(I,J)=0.0
3120=      DO 630 K=1,L
3130=      BEE(I,K)=0.0
3140= 630 CEE(K,I)=0.0
3150= 600 CONTINUE
3160=C
3170=      CALL VMULFF(K0,F2,L,L,L,10,10,KOF2,10,IER)
3180=      CALL VMULFF(AC,K1,L,L,L,10,10,ACK1,10,IER)
3190=      CALL VMULFF(AC,KOF2,L,L,L,10,10,ACKF2,10,IER)
3200=      CALL VMULFF(AC,K0,L,L,L,10,10,ACK0,10,IER)
3210=      CALL VMULFF(ACK0,F1,L,L,NM,10,10,ACKF1,10,IER)
3220=C
3230=      DO 640 I=1,NM
3240=      DO 640 J=1,NM
3250= 640 AY(L+I,L+J)=A11(I,J)
3260=C
3270=      DO 650 I=1,L
3280=      DO 650 J=1,NM
3290=      AY(I,L+J)=-F1(I,J)
3300=      AY(I2+I,L+J)=A21(I,J)
3310=      AY(I3+I,L+J)=-G*ACKF1(I,J)
3320=      AY(L+J,I2+I)=A12(J,I)
3330= 650 CEE(I,L+J)=C1(I,J)

```

```

3340=C
3350=      DO 660 I=1,L
3360=      DO 660 J=1,L
3370=      AY(I,I2+J)=-F2(I,J)
3380=      AY(I2+I,I2+J)=A22(I,J)
3390=      AY(I2+I,I3+J)=B2(I,J)
3400=      AY(I3+I,J)=G*ACK1(I,J)
3410=      AY(I3+I,I2+J)=-G*ACKF2(I,J)
3420=      AY(I3+I,I3+I)=-AC(I,I)
3430=      CEE(I,I2+J)=C2(I,J)
3440=      BEE(I,I)=1.0
3450= 660 BEE(I3+I,J)=G*ACK0(I,J)
3460=C
3470=      NL=I1
3480=      IF (X25.NE.1) GOTO 4310
3490=      PRINT*," "
3500=      PRINT*,"THE CLOSED LOOP AY MATRIX IS -"
3510=      CALL PRINTM(NL,NL,AY)
3520=      PRINT*," "
3530=      PRINT*,"THE CLOSED LOOP BEE MATRIX IS -"
3540=      CALL PRINTM(NL,L,BEE)
3550=      PRINT*," "
3560=      PRINT*,"THE CLOSED LOOP CEE MATRIX IS -"
3570=      CALL PRINTM(L,NL,CEE)
3580= 4310 CONTINUE
3590=      PRINT*," "
3600=C
3610=      DO 4620 I=1,NL
3620=      DO 4620 J=1,NL
3630= 4620 AY1(I,J)=AY(I,J)
3640=C
3650=      PRINT*,"THE OVERALL CLOSED LOOP EIGENVALUES ARE :"
3660=      PRINT*," "
3670=      CALL EIGRF(AY1,NL,10,0,W1,ZQ,10,WKAREA,IER)
3680=C
3690=      DO 4700 I=1,NL
3700= 4700 PRINT*," EIGENVALUE (" ,I," ) = " ,W1(I)
3710=      IF(X11.EQ.2) GO TO 9979
3720=C
3730= 4710 CONTINUE
3740=      PRINT*," "
3750=      PRINT*,"CHANGE OUTPUTS/EIGENVALUES (1), CHANGE ONLY GAIN (2),"
3760=      PRINT*,"CHANGE SIGMA (K0/K1) (3), INTEGRATE C-L EGNS (4),"
3770=      PRINT*," CHANGE PLANT (5), STOP (0) : "
3780=      READ*,X10
3790=      IF (X10.EQ.0) GOTO 9999
3800=      IF (X10.EQ.1) GOTO 30
3810=      IF (X10.EQ.2) GOTO 3999
3820=      IF (X10.EQ.3) GOTO 370
3830=      IF (X10.EQ.5) GOTO 9970
3840=C
3850=C
3860= 9979 PRINT*," "
3870=      PRINT*," *****  ENTER INTEGRATING ROUTINE  *****"

```

```

3880= PRINT*,"FOR THIS SYSTEM THERE ARE ",L," INPUTS AND ",L," OUTPUTS"
3890= PRINT*,"DO YOU WISH TO SEE THE INPUT HISTORIES OR OUTPUT"
3900= PRINT*," HISTORIES ? (1=INPUT,2=OUTPUT) :"
3910= READ*,AA1
3920= PRINT*," ENTER MANEUVER (Y.P.=1,W.L.T.=2,H.T.=3) :"
3930= READ*,AA2
3940= PRINT*," "
3950= PRINT*," ENTER INPUT COMMAND"
3960= PRINT*," FINAL VALUE, RAMP TIME (UNIT STEP = 1,0)"
3970=C
3980= DO 1111 I=1,L
3990= KVR(I)=0
4000= SLOPE(I)=0.0
4010= VST(I)=0.0
4020= KVST(I)=0
4030= PRINT*,I," : "
4040= READ*,VC(I),VDT(I)
4050= VC(I)=VC(I)/57.2957795
4060= IF(VDT(I).EQ.0) GO TO 1160
4070= KVR(I)=1
4080= SLOPE(I)=VC(I)/VDT(I)
4090= GO TO 1111
4100= 1160 PRINT*," ENTER STEP RESPONSE END TIME (INFINITY=0) :"
4110= READ*,VST(I)
4120= IF(VST(I).EQ.0) GO TO 1165
4130= KVST(I)=1
4140= 1165 V(I)=VC(I)
4150= 1111 CONTINUE
4160= TDT=VDT(1)
4170= IF(AA2.EQ.2) TDT=VST(3)
4180= FV=VC(1)*57.2957795
4190=C
4200= PRINT*," "
4210= PRINT*,"ENTER FINAL TIME AND TIME STEP :"
4220= READ*,TF,DELT
4230= PRINT*," "
4240= NEQN=NL
4250= T=0
4260= TOUT=0
4270=C
4280= DO 1113 I=1,NL
4290= X(I)=0
4300= Y(I)=0
4310= 1113 U(I)=0
4320=C
4330= DO 1112 I=1,5
4340= 1112 ER(I)=0
4350=C
4360= RR(1)=0
4370= RR(2)=0
4380= RELERR=1E-06
4390= ABSERR=1E-06
4400= IFLAG=1
4410= NSTEP=INT(TF/DELT)+1

```

```

4420=C
4430=      IF(AA1.EQ.1) GO TO 131
4440=      PRINT*,"          T          OUTPUTS"
4450=      GO TO 132
4460= 131 PRINT*,"          T          CONTROLS"
4470= 132 PRINT*,"-----"
4480=      IF(AA1.EQ.1) GO TO 125
4490=      IF(AA2.EQ.1) GO TO 126
4500= 125 PRINT 9980,TOUT,(U(I),I=1,L)
4510=      WRITE(4,9980) TOUT,(U(I),I=1,L)
4520=      GO TO 127
4530= 126 PRINT 9981,TOUT,(Y(I),I=1,L),RR(1),RR(2)
4540=      WRITE(4,9981) TOUT,(Y(I),I=1,L),RR(1),RR(2)
4550= 127 CONTINUE
4560=C
4570=C
4580=      DO 1115 K=1,NSTEP
4590=C
4600=      TOUT=TOUT+DELT
4610=C
4620=      DO 1150 I=1,L
4630=      IF(KVR(I).EQ.0) GO TO 1155
4640=      V(I)=SLOPE(I)*TOUT
4650=      IF(TOUT.GT.VDT(I)) V(I)=VC(I)
4660=      GO TO 1150
4670= 1155 IF(KVST(I).EQ.0) GO TO 1150
4680=      IF(TOUT.GT.VST(I)) V(I)=0.0
4690= 1150 CONTINUE
4700=C
4710=      CALL ODE(F,NEQN,X,T,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK)
4720=C
4730=      IF(AA1.EQ.2) GO TO 133
4740=C
4750=      DO 1120 I=1,L
4760= 1120 U(I)=X(I3+I)
4770=C
4780=C
4790=C
4800=      DO 1136 I=1,L
4810= 1136 UC(I)=U(I)*57.29578
4820=C
4830=C
4840=      PRINT 9980,TOUT,(UC(I),I=1,L)
4850=      WRITE(4,9980) TOUT,(UC(I),I=1,L)
4860=      GO TO 134
4870=C
4880= 133 DO 1128 I=1,L
4890= 1128 Y(I)=0
4900=C
4910=      DO 135 I=1,L
4920=      DO 135 J=1,NL
4930= 135 Y(I)=Y(I)+CEE(I,J)*X(J)
4940=C
4950=      DO 136 I=1,L
4960= 136 Y(I)=Y(I)*57.2957795

```

```

4970=C
4980=      KK=2
4990=      IF(TOUT.EQ.TDT) KK=1
5000=      IF(AA2.EQ.1) GO TO 137
5010=      IF(AA2.EQ.2) GO TO 139
5020=C
5030=      IF(TOUT.GT.TDT) GO TO 141
5040=      ER(1)=ER(1)+Y(1)*DELT*KK
5050=      ER(2)=0
5060=      ER(4)=0
5070= 141 ER(3)=ER(3)+ABS(Y(2)*DELT*KK)
5080=      E4=(Y(1)-FV)*DELT*KK
5090=      ER(4)=ER(4)+ABS(E4)
5100=      ER(5)=ER(5)+ABS(Y(3)*DELT*KK)
5110=      GO TO 142
5120=C
5130= 137 IF(TOUT.GT.TDT) GO TO 138
5140=      ER(1)=ER(1)+Y(1)*DELT*KK
5150=      ER(2)=ER(2)+Y(3)*DELT*KK
5160=      ER(4)=0
5170=      ER(5)=0
5180= 138 ER(3)=ER(3)+ABS(Y(2)*DELT*KK)
5190=      E4=(Y(1)-FV)*DELT*KK
5200=      ER(4)=ER(4)+ABS(E4)
5210=      ER(5)=ER(5)+ABS(Y(3)*DELT*KK)
5220=      RR(1)=RR(1)+Y(3)*DELT
5230=      RR(2)=RR(1)+Y(1)
5240=      GO TO 142
5250=C
5260= 139 IF(TOUT.GT.TDT) GO TO 140
5270=      ER(1)=0
5280=      ER(2)=ER(2)+Y(3)*DELT*KK
5290=      ER(5)=0
5300= 140 ER(3)=ER(3)+ABS(Y(2)*DELT*KK)
5310=      ER(4)=ER(4)+ABS(Y(1)*DELT*KK)
5320=      ER(5)=ER(5)+ABS(Y(3)*DELT*KK)
5330=C
5340= 142 IF(AA2.EQ.1) GO TO 128
5350=      PRINT 9980,TOUT,(Y(I),I=1,L)
5360=      WRITE(4,9980) TOUT,(Y(I),I=1,L)
5370=      GO TO 129
5380= 128 PRINT 9981,TOUT,(Y(I),I=1,L),RR(1),RR(2)
5390=      WRITE(4,9981) TOUT,(Y(I),I=1,L),RR(1),RR(2)
5400= 129 CONTINUE
5410=C
5420= 134 CONTINUE
5430= 1115 CONTINUE
5440= 9980 FORMAT(5X,F5.2,3(3X,F10.4))
5450= 9981 FORMAT(5X,F5.2,3(3X,F10.4),2X,F7.4,2X,F7.4)
5460=      PRINT*,"-----"
5470=      PRINT*," "
5480=      IF(AA1.EQ.1) GO TO 146
5490=      DO 147 I=1,5
5500= 147 ER(I)=ER(I)/KK
5510=C
5520=      PRINT*," BETA COMMAND =",ER(1)

```

```

5530=      PRINT*, " R COMMAND =", ER(2)
5540=      PRINT*, " PHI =", ER(3)
5550=      PRINT*, " BETA TRANS. =", ER(4)
5560=      PRINT*, " R TRANS. =", ER(5)
5570=      PRINT*, " "
5580= 146  CONTINUE
5590=      GOTO 4710
5600=C
5610=C
5620=C
5630= 9999 CONTINUE
5640=      PRINT*, " "
5650=      PRINT*, "ALL INFORMATION IN PORTER HAS BEEN SAVED IN LOCAL"
5660=      PRINT*, "FILE - PLOT"
5670=      STOP
5680=      END
5690=C
5700=C
5710=C
5720=      SUBROUTINE MADD(A,B,L,N,C)
5730=      DIMENSION A(10,10),B(10,10),C(10,10)
5740=      DO 100 I=1,L
5750=      DO 100 J=1,N
5760= 100  C(I,J)=A(I,J)+B(I,J)
5770=      RETURN
5780=      END
5790=C
5800=      SUBROUTINE READM(L,N,A)
5810=      REAL A(10,10)
5820=      PRINT*, " "
5830=      DO 100 I=1,L
5840=      PRINT*, " ROW ", I, " : "
5850= 100  READ*, (A(I,J), J=1,N)
5860=      RETURN
5870=      END
5880=C
5890=      SUBROUTINE PRINTM(L,N,A)
5900=      REAL A(10,10)
5910=      PRINT*, " "
5920=      DO 100 I=1,L
5930= 100  PRINT 200, (A(I,J), J=1,N)
5940= 200  FORMAT(1X,B(F12.4,3X))
5950=      RETURN
5960=      END
5970=C
5980=C
5990=      SUBROUTINE MSUB(A,B,C,L,N)
6000=      REAL A(10,10),B(10,10),C(10,10)
6010=      DO 100 I=1,L
6020=      DO 100 J=1,N
6030= 100  C(I,J)=A(I,J)-B(I,J)
6040=      RETURN
6050=      END
6060=C
6070=C
6080=      SUBROUTINE DSGN1(N,L,A,B,C,D,E,F)

```

```

6090=      REAL A(10,10),B(10,10),C(10,10),D(10,10),E(10,10),F(10,10)
6100=      N=4
6110=      L=3
6120=      A(1,1)=0.
6130=      B(1,1)=0.0 $B(1,2)=1.0 $B(1,3)=.03184253
6140=      C(1,1)=.04885258 $C(2,1)=0.0 $C(3,1)=0.
6150=      D(1,1)=-.37442438 $D(1,2)=.03182685 $D(1,3)=-.99948763
6160=      D(2,1)=-39.92833732 $D(2,2)=-2.11380581 $D(2,3)=.07969577
6170=      D(3,1)=6.23794780 $D(3,2)=-.00631591 $D(3,3)=-.70740917
6180=      DO 100 I=1,3
6190= 100 E(1,I)=0.0
6200=      F(1,1)=.05309116 $F(1,2)=.00463283 $F(1,3)=.02501018
6210=      F(2,1)=9.79285579 $F(2,2)=-55.73875487 $F(2,3)=2.60326273
6220=      F(3,1)=-5.05439681 $F(3,2)=-1.89267784 $F(3,3)=4.38800282
6230=      PRINT*," "
6240=      PRINT*," THE A11 MATRIX IS -"
6250=      CALL PRINTM(1,1,A)
6260=      PRINT*," "
6270=      PRINT*," THE A12 MATRIX IS -"
6280=      CALL PRINTM(1,3,B)
6290=      PRINT*," "
6300=      PRINT*," THE A21 MATRIX IS -"
6310=      CALL PRINTM(3,1,C)
6320=      PRINT*," "
6330=      PRINT*," THE A22 MATRIX IS -"
6340=      CALL PRINTM(3,3,D)
6350=      PRINT*," "
6360=      PRINT*," THE B1 MATRIX IS A 1 BY 3 ZERO MATRIX "
6370=      PRINT*," "
6380=      PRINT*," THE B2 MATRIX IS -"
6390=      CALL PRINTM(3,3,F)
6400=      RETURN
6410=      END
6420=C
6430=C
6440=      SUBROUTINE F(T,X,XP)
6450=      REAL X(10),XP(10),AY(10,10),BEE(10,10),V(10)
6460=      COMMON AY,BEE,V,NL,L
6470=C
6480=      DO 100 I=1,NL
6490= 100 XP(I)=0
6500=C
6510=      DO 200 I=1,NL
6520=      DO 200 J=1,NL
6530= 200 XP(I)=XP(I)+AY(I,J)*X(J)
6540=C
6550=      DO 300 I=1,NL
6560=      DO 300 J=1,L
6570= 300 XP(I)=XP(I)+BEE(I,J)*V(J)
6580=C
6590=      RETURN
6600=      END
6610=C
6620=C
6630=*EOR
6640=*EOF

```

```

3300= 3999 PRINT*," "
3310= PRINT*,"PRINT C-L MATRICES (1=YES) :"
3320= READ*,X25
3330= PRINT*," "
3340= PRINT*,"ENTER THE GAIN CONSTANT , G :"
3350= READ*,G
3360=C
3370=C FORM CLOSED LOOP A-MATRIX (AY)
3380=C
3390= 9977 DO 4000 I=1,N
3400= DO 4000 J=1,L
3410= 4000 AY(I,J)=0.
3420=C
3430= CALL VMULFF(B2,K1,L,L,L,8,8,B2K1,8,IER)
3440= CALL VMULFF(B2,K0,L,L,L,8,8,B2K0,8,IER)
3450= CALL VMULFF(B2K0,F1,L,L,NM,8,8,B2K0F1,8,IER)
3460= CALL VMULFF(B2K0,F2,L,L,L,8,8,B2K0F2,8,IER)
3470=C
3480= DO 4100 I=1,L
3490=C
3500= DO 4090 J=1,L
3510= BKF2(I,J)=G*B2K0F2(I,J)
3520= 4090 BK1(I,J)=G*B2K1(I,J)
3530=C
3540= DO 4095 KA=1,NM
3550= 4095 BKF1(I,KA)=G*B2K0F1(I,KA)
3560=C
3570= 4100 CONTINUE
3580=C
3590= CALL MSUB(AZ1,BKF1,THRTWO,L,NM)
3600= CALL MSUB(AZ2,BKF2,THRTHR,L,L)
3610=C
3620= DO 4200 I=1,L
3630=C
3640= DO 4190 J=1,L
3650= KA=I+N
3660= LA=J+N
3670= AY(KA,J)=BK1(I,J)
3680= AY(I,LA)=-F2(I,J)
3690= 4190 AY(KA,LA)=THRTHR(I,J)
3700=C
3710= DO 4195 J=1,NM
3720= LB=J+L
3730= AY(I,LB)=-F1(I,J)
3740= 4195 AY(KA,LB)=THRTWO(I,J)
3750=C
3760= 4200 CONTINUE
3770=C
3780= DO 4300 I=1,NM
3790=C
3800= DO 4290 J=1,L
3810= LA=I+L
3820= KA=J+N
3830= 4290 AY(LA,KA)=A12(I,J)
3840=C

```

```

3850=      DO 4295 J=1,NM
3860=      KB=J+L
3870= 4295  AY(LA,KB)=A11(I,J)
3880=C
3890= 4300 CONTINUE
3900=C
3910=      NL=N+L
3920=      IF (X25.NE.1) GOTO 4310
3930=      PRINT*," "
3940=      PRINT*,"THE CLOSED LOOP AY MATRIX IS -"
3950=      CALL PRINTM(NL,NL,AY)
3960= 4310 CONTINUE
3970=C
3980=C      FORM CLOSED LOOP B-MATRIX (BEE)
3990=C
4000=      DO 4400 I=1,NL
4010=      DO 4400 J=1,L
4020= 4400 BEE(I,J)=0.0
4030=C
4040=      DO 4410 I=1,L
4050= 4410 BEE(I,I)=1.0
4060=C
4070=      DO 4420 I=1,L
4080=      DO 4420 J=1,L
4090=      GB2K0(I,J)=G*B2K0(I,J)
4100= 4420 BEE(N+I,J)=GB2K0(I,J)
4110=C
4120=      IF (X25.NE.1) GOTO 4510
4130=      PRINT*," "
4140=      PRINT*,"THE CLOSED LOOP BEE MATRIX IS -"
4150=      CALL PRINTM(NL,L,BEE)
4160= 4510 CONTINUE
4170=C
4180=C      FORM CLOSED LOOP C-MATRIX (CEE)
4190=C
4200=      DO 4500 I=1,L
4210=      DO 4500 J=1,NL
4220= 4500 CEE(I,J)=0.0
4230=C
4240=      DO 4515 I=1,L
4250=C
4260=      DO 4520 J=1,NM
4270= 4520 CEE(I,L+J)=C1(I,J)
4280=C
4290=      DO 4530 J=1,L
4300= 4530 CEE(I,N+J)=C2(I,J)
4310=C
4320= 4515 CONTINUE
4330=C
4340=      PRINT*," "
4350=      IF (X25.NE.1) GOTO 4610
4360=      PRINT*,"THE CLOSED LOOP CEE MATRIX IS -"
4370=      CALL PRINTM(L,NL,CEE)
4380=C

```

VITA

Thomas Lewis was born 27 January 1958 in Albert Lea, Minnesota. He graduated from Suffield Academy, Suffield Connecticut in 1976, and then attended Purdue University, West Lafayette Indiana, where he earned a Bachelor of Science degree in Aeronautical and Astronautical Engineering, graduating May 1981. He also received a reserve commission in the USAF through the ROTC program at Purdue. He immediately pursued a graduate degree by enrolling in the Air Force Institute of Technology, Department of Aeronautical Engineering, WPAFB Ohio, in June 1981.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GAE/EE/82D-1	2. GOVT ACCESSION NO. AD-A124871	3. PCIC/FNT'S CATALOG NUMBER
4. TITLE (and Subtitle) HIGH-GAIN ERROR ACTUATED FLIGHT CONTROL SYSTEMS FOR CONTINUOUS LINEAR MULTIVARIABLE PLANTS		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 2Lt. Thomas Lewis, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December, 1982
		13. NUMBER OF PAGES 137
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release: IAW AFR 150-17. <i>[Signature]</i> LYNN E. WOLAVER Dean for Research and Professional Development Air Force Institute of Technology (AFIT) Wright-Patterson AFB OH 45433 4 JAN 83		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Flight Controller; Decoupling, non-interacting; Singular Perturbations; Advanced Fighter Aircraft, AFTI; Proportional plus Integral Control; Transmission ZEROS, Decoupling Zeros		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The theory of high-gain error actuated feedback control, developed by Potter and Bradshaw, is applied to the design of various lateral-directional decoupling flight control systems for an advanced aircraft. The controllers developed in this report utilize output feedback with proportional plus inte- gral control to produce desirable closed-loop responses with minimal inter- action between outputs. Because of the structure of the system, measurement variables in addition to the outputs are necessary to apply this method.		

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. The report examines controller design robustness by varying the flight conditions or maneuver commands from the ones the controller was specifically designed for, and then judges system performance. The results show that the controller is robust with respect to varying flight conditions, but is not robust with respect to varying maneuver commands. This report also examines the effect of first-order actuator dynamics in the system model. Actuator dynamics are shown to significantly affect the control system response, indicating that a simplified model, without actuators, is not desirable in one's control design scheme. Also a computer program to determine transmission zeros and decoupling zeros is developed.

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

END

FILMED

3-83

DTIC