

AD-A128 579

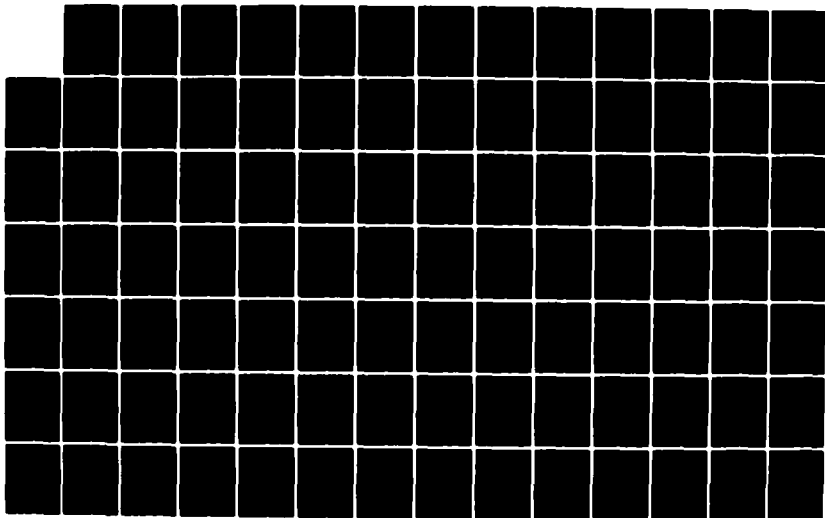
A LATERAL-DIRECTIONAL CONTROLLER FOR
HIGH-ANGLE-OF-ATTACK FLIGHT(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH W A EHRENSTROM MAR 83
AFIT/C1/NR-83-12T

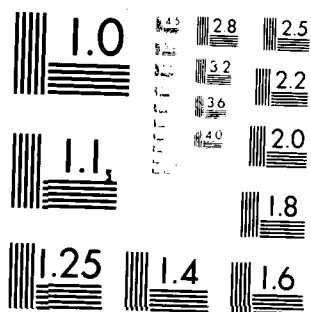
1/3

UNCLASSIFIED

F/G 1/2

NL





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

READ INSIDE COVER
FOR MORE INFORMATION

67582

144

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**



A LATERAL-DIRECTIONAL CONTROLLER
FOR HIGH-ANGLE-OF-ATTACK FLIGHT

by

2LT William A. Ehrenstrom

Princeton University
School of Engineering and Applied Science
Department of Mechanical and Aerospace Engineering

Submitted in partial fulfillment of the requirements for the degree
of Master of Science in Engineering from Princeton University, 1982

Prepared by:

William A. Ehrenstrom

Approved by:

Robert F. Stengel

Professor Robert F. Stengel
Thesis Advisor

H.C. Curtin

Professor
Thesis Reader

March, 1983

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Dist	
Avail and/or Special	
A	

ABSTRACT

A digital flight control system based on microprocessor technology has been designed, developed, and flight tested using the Avionics Research Aircraft (ARA) at Princeton University. The control system utilizes the existing microprocessor system available in the aircraft's fly-by-wire control system. The command and stability augmentation control law was developed using modern control theory and is incorporated into existing flight control computer programs. Development of the model and control law, the gain scheduling procedure, and the flight test results are presented.

The objective of the study was to provide lateral-directional stability during high-angle-of-attack flight and into the stall regime. Flight test results show that it is indeed possible to design a control system which will eliminate lateral-directional instabilities and do so at a level higher than the pilot was able to attain. In addition, gain scheduling proved to be capable of providing satisfactory control throughout a wide range of flight conditions. Additional work, however, is required to correct a number of control system inadequacies before the control system can become operational.

ACKNOWLEDGMENTS

I would like to thank the following for their assistance in completing this study:

The Schultz Foundation for their sponsorship of the study;

Professor Robert F. Stengel for his assistance in the selection and development of the control system;

W. Barry Nixon, senior technical staff member, for his assistance in conducting both ground and flight tests;

George E. Miller, senior technical stall member, for his assistance in preparing the final control system for flight test;

Robert V. Walters, graduate student, for development of control system software which was modified for the purposes of this project;

David B. Glade, graduate student, for his assistance in learning the microprocessor systems;

and Marion E. Sandvik for typing much of the final report.

This thesis carries 1583-T in the records of the Department of Mechanical and Aerospace Engineering.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	ii
ACKNOWLEDGMENTS	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
 I. INTRODUCTION	 1
1.1 Description of the Problem	1
1.2 Organization of the Thesis	3
 II. DEVELOPMENT OF THE MODEL	 5
2.1 Nonlinear, Time-Varying Equations	7
2.2 Linearization	8
2.2.1 Calculation of the Trim Condition	11
2.2.2 Selection of the Command Vector	13
2.3 Nondimensional Force and Moment Coefficients	14
2.3.1 Constant Components	16
2.3.2 Stability Derivative Components	19
2.3.3 Rotary Derivative Components	20
2.3.4 Control Derivative Components	29
2.4 Open-Loop Results	32
2.5 Model Reevaluation	33
2.5.1 Inertia Matrix	34
2.5.2 Vertical Component of Velocity	37
 III. DEVELOPMENT OF THE CONTROL LAW	 40
3.1 Singular Command Equilibrium	42
3.2 Calculation of Optimal Gains	49
3.2.1 Sampled-Data System Equations	50
3.2.2 Sampled-Data State- and Control- Weighting Matrices	52
3.2.3 Solution to the Discrete Riccati Equation	53

TABLE OF CONTENTS - Continued

	<u>Page</u>
3.2.4 Computation of the Closed-Loop System	53
3.3 Calculation of Control Gains	54
3.4 Closed-Loop Results	56
3.4.1 Selection of Q_c and R_c	56
3.4.2 Closed-Loop Simulation: Nominal Flight Condition	62
3.4.3 Summary of Closed-Loop Results	67
IV. GAIN SCHEDULING	70
4.1 Flight Condition Functions	72
4.2 Selection of Solution Forms	76
4.3 Computation of the Coefficient Matrices	81
4.4 Gain Scheduling Simulation	84
V. FLIGHT TESTING	88
5.1 Description of the Micro-DFCS	89
5.2 CAS Software	90
5.3 Ground Tests	96
5.4 Flight Tests	97
5.4.1 Airframe Tests	100
5.4.2 Pilot Tests	108
5.4.3 CAS Tests	111
5.5 Analysis of Results	124
VI. CONCLUSIONS AND RECOMMENDATIONS	128
APPENDIX	
A. AVIONICS RESEARCH AIRCRAFT	133
A.1 Description of the Aircraft	133
A.2 Aircraft Data	134
B. GAIN COMPUTATION SOFTWARE	144
C. CAS SOFTWARE	174
REFERENCES	191

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Open-Loop Results	38
2. Response Characteristics Varying Sideslip and Roll Angle Weightings	59
3. Response Characteristics Varying Sideslip Weighting	61
4. Closed-Loop Results	68
5. Flight Condition and Gain Matrices	75
6. Coefficient Gain Matrices	82
7. Flight Test Documentation	99
8. Constant Component Data	135
9. Stability Derivative Component Data	136
10. Longitudinal Coefficient Data	138
11. Rudder Derivative Component Data	139
12. Aileron Derivative Component Data	141
13. Aircraft Constants	142
14. Gain Computation Listings	147
15. CAS Version 6.5 Listing	177

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Control Law Configuration	14
2. Eigenvalue Plot: Open-Loop vs. Closed-Loop	64
3. Closed-Loop Simulation: Roll Rate Command	66
4. Closed-Loop Simulation: Sideslip Command	66
5. Gain Sensitivities to Changes in Angle of Attack . . .	78
6. Gain Sensitivities to Changes in Throttle Setting . .	78
7. Gain Sensitivities to Changes in Dynamic Pressure . .	79
8. Gain Schedule Simulation: Roll Rate Command	87
9. Gain Schedule Simulation: Sideslip Command	87
10. Overview of Aircraft Systems Configuration	90
11. Control System Execution Cycle	93
12. Flowchart of Control Sequence	95
13. Flight Test Run 1-1 Results	104
14. Flight Test Run 1-2 Results	105
15. Flight Test Run 1-3 Results	106
16. Flight Test Run 1-4 Results	107
17. Flight Test Run 2-1 Results	112
18. Flight Test Run 2-2 Results	113
19. Flight Test Run 2-3 Results	114
20. Flight Test Run 2-4 Results	115
21. Flight Test Run 3-1 Results	120
22. Flight Test Run 3-2 Results	121
23. Flight Test Run 3-3 Results	122
24. Flight Test Run 3-4 Results	123
25. Gain Computation Flowchart	146

Chapter I

INTRODUCTION

1.1 DESCRIPTION OF THE PROBLEM

The study of aircraft in high angles of attack is of critical importance to the understanding of aircraft flight and to design for aircraft safety. It is in this regime that the maximum lift is reached and stall encountered -- one of the primary reasons for aircraft accidents. By understanding the high angle-of-attack regime, it might be possible to design safety features into an aircraft which could delay the onset of stall, warn the pilot of an impending stall, or enable the pilot to recover more quickly. This topic has been the subject of ongoing study at the Princeton Flight Research Laboratory (PFL).

While stall is primarily a problem of the longitudinal mode of the aircraft, the lateral-directional mode can be seriously affected in the regime near stall. With lateral-directional controls greatly reduced, the aircraft can experience instabilities such as wing rock. Indeed, these instabilities have been noted in the research aircraft at Princeton (Ref. 1). In addition, it has been found that increasing the throttle tends to aggravate the unstable condition. The lateral-directional instabilities can cause difficulties in obtaining data in the

high angle-of-attack regime and thus severely hamper efforts in dealing with aircraft stall. Therefore, a lateral-directional command augmentation system (CAS) would be helpful to steady the aircraft and to isolate the critical aspects of high angle-of-attack flight from lateral-directional disturbances. The topic of this study, then, is the design and testing of a lateral-directional CAS capable of operating at all flight conditions which the aircraft might encounter but with special emphasis on high angle-of-attack conditions.

The FRL has at its disposal two research aircraft, one of which is the Avionics Research Aircraft (ARA) Navion. The Navion has been tested extensively in the wind tunnel at NASA's research facility at Hampton, Virginia and the results are summarized in NASA TN D-5857 (Ref. 2). The availability of the data simplified the process of developing a model, and the presence of the aircraft enhanced the ability to validate the control system once it was designed.

The ARA has been fitted with a microprocessor for digital flight control. The CAS, then, was designed to be digital to take advantage of the microprocessor. In addition, FRL has at its disposal a ground-based microprocessor system for software development and testing.

The CAS itself was designed as a two-input, two-output, single command mode controller. Lateral stick and pedals were selected as inputs and were scaled to be roll rate and sideslip commands,

respectively. The outputs were commands to the ailerons and rudder. Three longitudinal variables -- angle of attack, throttle setting, and dynamic pressure -- were selected as the flight condition. Since the CAS was required to work at all flight conditions, the gains were scheduled with these three variables.

1.2 ORGANIZATION OF THE THESIS

The thesis is presented to cover the critical steps that were taken to design and test the CAS. In particular, Chapter 2 deals with the development of the model of the AKA. In addition, the linearization procedure is discussed. Finally, a set of open-loop results is presented for comparison with known aircraft behavior.

Chapter 3 discusses the calculation of the control gains using linear-quadratic, sampled-data regulator theory. Singular command equilibrium is discussed, as is the calculation of the optimal gains. In addition, the control law is developed and presented. Finally, results are included showing the selection of the continuous-time weighting matrices, a detailed description of the CAS performance for a nominal flight condition, and a summary of closed-loop simulation results for a variety of flight conditions.

Chapter 4 discusses the gain scheduling process. The form of the gain equations is discussed, as is the solution of the flight

condition functions which make up the gain equations. Also included is the computation of the gain coefficient matrices. Finally, results of a simulation using the gain schedules are presented.

Chapter 5 covers the flight tests. The microprocessor system used and the development of software for CAS implementation are discussed. In addition, ground tests performed before flight tests are covered. Finally, the flight test procedures and a detailed analysis of the results are included.

Chapter 6 presents some conclusions and possible recommendations for further work in the area of gain scheduling.

The appendices cover the extraneous areas of study which were required for successful completion of the project.

Appendix A presents the aircraft data used in the development and testing of the model. Appendix B summarizes the gain calculation software while Appendix C presents the microprocessor software. References lists the sources of information used during the course of this research.

Chapter II

DEVELOPMENT OF THE MODEL

The requirements of the control system specified that the CAS should provide satisfactory control for not just one flight condition, but for the whole spectrum of flight conditions the aircraft might encounter. This in turn required the development of a model which could accurately predict aircraft behavior throughout this same range of flight conditions. In other words, the model was developed not only as a function of the state and controls, but also of the flight condition.

The model could have taken two forms. The first stores the aerodynamic data in a table. To use this method, the model would be looked up for the particular flight condition from a set of tables containing a number of possible models. Since a model for every possible flight condition combination could not be listed, some flight conditions would have to be approximated by the closest model available or by interpolation between models. This idea was rejected as requiring too much computer storage and excessive run time.

The second choice was to find a set of polynomials to calculate the model based on the specified value of the flight condition. There is a considerable amount of aerodynamic data

from which the polynomials could be derived. All that was required was to reduce the available data to a set of equations. This was the method selected in this study.

The experimental data were compiled in NASA TN D-5857 (Ref. 2) and that which were used are summarized in Appendix A. The NASA report included data on stability and control derivatives. Experimental values of rotary derivatives were given in NASA TN D-6643 (Ref. 3) but these were considered insufficient since they did not include variations of these variables with the changing flight condition. Instead, the USAF STABILITY AND CONTROL DATCOM (Ref. 4) methods were used to derive the set of rotary derivative equations, and the values available in TN D-6643 were used for comparison.

The selection of the flight condition variables was made on the basis of available data and control system needs. Since the control system was to be angle-of-attack sensitive, angle of attack, α , was chosen as a variable. Fortunately, the flight data presented the variations of the lateral-directional parameters against variations in angle of attack. Since the application of throttle resulted in aggravating the unstable condition, throttle setting, T_c , as chosen as a flight condition variable. Data were also available showing the effect of throttle setting on the lateral-directional parameters. Finally, given the overall effect of dynamic pressure, \bar{q} (velocity and altitude effects), it was chosen as the final flight condition

variable. The ranges of each variable were based on typical operating conditions and available data: angle of attack, -4 to 24 degrees; throttle setting (representing nondimensional value of thrust, $T/\bar{q}S$), $.03$ to $.23$; and dynamic pressure, which was based on typical velocities for the aircraft of 100 to 200 feet per second and an altitude of 5000 feet.

This chapter covers the model development process from the linearization of the nonlinear equations to the data reduction and formulation of the nondimensional force and moment coefficients. Open-loop results for 27 different flight conditions are presented to help verify the model.

2.1 NONLINEAR, TIME-VARYING EQUATIONS

The nonlinear, lateral-directional equations can easily be found in the literature (Ref. 3) and are summarized as follows:

$$\begin{aligned} \dot{y} = & u \cos(\theta_c) \sin(\Psi) + v (\sin(\phi) \sin(\theta_c) \sin(\Psi) - \cos(\phi) \cos(\Psi)) \\ & w (\cos(\phi) \sin(\theta_c) \sin(\Psi) - \sin(\phi) \cos(\Psi)) \end{aligned} \quad (2-1)$$

$$\dot{v} = -ru + pw + g \cos(\theta_c) \sin(\phi) + (\bar{q}S/m)C_y \quad (2-2)$$

$$\dot{p} = (\bar{q}Sb/l_x)C_l \quad (2-3)$$

$$\dot{r} = (\bar{q}Sb/l_z)C_n \quad (2-4)$$

$$\dot{\phi} = p + (q_c \sin(\phi) + r \cos(\phi)) \tan(\theta_c) \quad (2-5)$$

$$\dot{\Psi} = q_c \sin(\phi) \sec(\theta_c) + r \cos(\phi) \sec(\theta_c) \quad (2-6)$$

The Ψ angular position and the y position variables give information on the aircraft with respect to some fixed axis

system, but they have no effect on the aircraft's stability. Hence, these two variables were dropped, leaving a fourth-order model. In addition, the lateral velocity variable, v , does not have as much physical meaning to the pilot as sideslip angle, B . To get the sideslip rate equation, the \dot{v} equation was divided by the total velocity, V_0 . Finally, the equations were rotated through the angle-of-attack (to the stability axis system). Hence, the final nonlinear equations of motion were:

$$\dot{r} = (\bar{q}Sb/I_z)C_n \quad (2-7)$$

$$\dot{B} = (1/V_0)(-ru + g\cos(\gamma_0)\sin(\phi) + (\bar{q}S/m)C_y \quad (2-8)$$

$$\dot{p} = (\bar{q}S/I_x)C_l \quad (2-9)$$

$$\dot{\phi} = p + (q_0\sin(\phi) + r\cos(\phi))\tan(\alpha_0) \quad (2-10)$$

These equations were rewritten as a vector differential equation:

$$\dot{\underline{x}} = \underline{f}(\underline{x}(t), \underline{u}(t)) \quad (2-11)$$

where \underline{x} is the state vector (r, B, p, ϕ) and \underline{u} is the control vector ($\delta k, \delta k$).

2.2 LINEARIZATION

The development of a control system based on the nonlinear equations of motion would be quite difficult. Therefore, it was imperative that the equations be linearized. This was done using a Taylor series approximation. The approximation was taken about a trim point -- the forces and moments add up to zero, and the aircraft is at equilibrium. The Taylor series approximation of the nonlinear equations, \underline{f} , is as follows:

$$\dot{\underline{x}} = \underline{f}(\underline{x}_0, \underline{u}_0) + \frac{\partial \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{x}} \Delta \underline{x} + \frac{\partial \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{u}} \Delta \underline{u} + \frac{\partial^2 \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{x}^2} \Delta \underline{x}^2 + \frac{\partial^2 \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{u}^2} \Delta \underline{u}^2 + \dots (2-12)$$

where \underline{x}_0 and \underline{u}_0 are the trim values of the states and controls, respectively. By dropping the higher order terms and rearranging, the approximation becomes:

$$\underline{x}_c = \underline{f}(\underline{x}_c, \underline{u}_c) \quad (2-13)$$

$$\Delta \dot{\underline{x}} = \frac{\partial \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{x}} \Delta \underline{x} + \frac{\partial \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{u}} \Delta \underline{u} \quad (2-14)$$

The \underline{x}_c equation represents the nominal trim solution and the \underline{x} equation represents the deviations from trim. It was this perturbation equation that was of the most interest. The derivative terms form a Jacobian matrices and were redefined as:

$$F = \frac{\partial \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{x}} \quad G = \frac{\partial \underline{f}(\underline{x}_0, \underline{u}_0)}{\partial \underline{u}}$$

The final linearized perturbation equations of motion (in matrix form) were as follows:

$$\Delta \dot{\underline{x}} = F(t) \Delta \underline{x} + G(t) \Delta \underline{u} \quad (2-15)$$

where F is the system dynamics matrix and G is the control dynamics matrix.

The nondimensional force and moment coefficients (C_y , C_n , C_l) are functions of the states and controls and were rewritten in linearized form as:

$$C_y = C_{y_c} + C_{y_B} \Delta E + (b/(2V_0)) C_{y_r} \Delta r + (b/(2V_0)) C_{y_p} \Delta p + C_{y_{\delta k}} \Delta \delta k + C_{y_{\delta A}} \Delta \delta A \quad (2-16)$$

$$C_n = C_{n_O} + C_{n_E} \Delta B + (b/(2V_O)) C_{n_R} \Delta r + (b/(2V_O)) C_{n_P} \Delta p + C_{n_{dK}} \Delta dK + C_{n_{dA}} \Delta dA \quad (2-17)$$

$$C_l = C_{l_O} + C_{l_E} \Delta B + (b/(2V_O)) C_{l_R} \Delta r + (b/(2V_O)) C_{l_P} \Delta p + C_{l_{dK}} \Delta dK + C_{l_{dA}} \Delta dA \quad (2-18)$$

where the equation coefficients are stability derivatives (C_{Y_E} , C_{n_E} , C_{l_E}), rotary derivatives (C_{Y_P} , C_{n_P} , C_{l_P} , C_{Y_R} , C_{n_R} , C_{l_R}), and control derivatives ($C_{Y_{dK}}$, $C_{n_{dK}}$, $C_{l_{dK}}$, $C_{Y_{dA}}$, $C_{n_{dA}}$, $C_{l_{dA}}$). The linearized equations in matrix form became (in dimensional terms):

$$I = \begin{bmatrix} N_R & N_L & N_P & C \\ Y_R/V_C - 1 & Y_L/V_C & Y_P/V_C & \epsilon^* \cos(\gamma_C)/V_O \\ L_R & L_L & L_P & C \\ \tan(\gamma_C) & C & 1 & C \end{bmatrix}$$

$$C = \begin{bmatrix} N_{dK} & N_{dA} \\ Y_{dK}/V_O & Y_{dA}/V_O \\ L_{dK} & L_{dA} \\ C & C \end{bmatrix}$$

where $()_L = C()_E \bar{q} S / m$, etc.

In addition to the equations of motion, an output equation was required to keep track of certain quantities over which explicit control was desirable. This was done using an output equation of the form:

$$y(t) = h(\underline{x}(t), \underline{u}(t)) \quad (2-19)$$

Performing the Taylor series approximation and redefining, the perturbation output equation became:

$$\Delta \underline{y} = H_x \Delta \underline{x} + H_u \Delta \underline{u} \quad (2-20)$$

where

$$H_x = \frac{\partial h(\underline{x}_0, \underline{u}_0)}{\partial \underline{x}} \quad H_u = \frac{\partial h(\underline{x}_0, \underline{u}_0)}{\partial \underline{u}}$$

The selection of \underline{h} is discussed later.

2.2.1 Calculation of the Trim Condition

At the trim condition, all of the forces and moments sum to be zero; hence, the state is in equilibrium ($\Delta \underline{x} = \underline{0}$). Therefore, to solve for the trim condition, \underline{f} was set to zero and the states computed.

$$0 = (\bar{q}Sb/l_z)C_n \quad (2-21)$$

$$0 = (1/V_0)(-ru_0 + q\cos(\gamma_0)\sin(\phi) + (\bar{q}S/m)C_y) \quad (2-22)$$

$$0 = (\bar{q}Sb/l_x)C_l \quad (2-23)$$

$$0 = p + (q_0\sin(\phi) + r\cos(\phi))\tan(\gamma_0) \quad (2-24)$$

Multiplying by terms that were unchanging, equations (2-21) to (2-24) simplified to:

$$0 = C_n \quad (2-25)$$

$$0 = -ru_0 + q\cos(\gamma_0)\sin(\phi) + (\bar{q}S/m)C_y \quad (2-26)$$

$$0 = C_l \quad (2-27)$$

$$0 = p + (q_0\sin(\phi) + r\cos(\phi))\tan(\gamma_0) \quad (2-28)$$

With no initial roll rate or yaw rate and zero C_n and C_l , the nominal values of p and r were zero. Thus, from (2-16), (2-17), and (2-18), equations (2-25) to (2-28) simplified to:

$$C = C_{n_o} + C_{n_E} \Delta E + C_{n_{dR}} \Delta dR + C_{n_{dA}} \Delta dA \quad (2-29)$$

$$0 = C_{y_o} + C_{y_E} \Delta E + C_{y_{dR}} \Delta dR + C_{y_{dA}} \Delta dA + ((gm)/(\bar{q}S)) \cos(\gamma_o) \sin(\phi) \quad (2-30)$$

$$C = C_{l_o} + C_{l_E} \Delta E + C_{l_{dR}} \Delta dR + C_{l_{dA}} \Delta dA \quad (2-31)$$

$$C = 0 \quad (2-32)$$

By examining (2-29) to (2-31), it was apparent that the aircraft could have a trim sideslip and zero roll angle, a trim roll angle and zero sideslip, or a combination of roll angle and sideslip. In every case the trim controls were non-trivial.

For this study, roll angle was set to zero, leaving a trim sideslip. Doing this and manipulating equations (2-29) to (2-31), the following matrix solution for E , dR , and dA was obtained:

$$\begin{bmatrix} \Delta E \\ \Delta dR \\ \Delta dA \end{bmatrix} = \begin{bmatrix} C_{n_E} & C_{n_{dR}} & C_{n_{dA}} \\ C_{y_E} & C_{y_{dR}} & C_{y_{dA}} \\ C_{l_E} & C_{l_{dR}} & C_{l_{dA}} \end{bmatrix}^{-1} \begin{bmatrix} -C_{n_o} \\ -C_{y_o} \\ -C_{l_o} \end{bmatrix} \quad (2-33)$$

The solution of (2-33) provided the trim condition about which the equations were linearized and the perturbations measured.

2.2.2 Selection of the Command Vector

While the selection of the command vector, \underline{h} , has no effect on the eventual stability of the closed loop system, it can drastically affect the response of the system. If the command vector is chosen such that the state contains an integral of one of the command vector elements, then the response will be singular. Singular command equilibrium does not necessarily imply state and control equilibrium. Furthermore, computation of the control gains will result in a set of integral feedforward elements, known as proportional-integral filtering, in addition to the usual feedforward and feedback gains.

Comparison of singular versus nonsingular command modes shows that the singular command mode has the effect of adding a third state to the two-state command vector, namely the integral state (Ref. 5). Selection of the singular command mode, then, appears to be a wise choice and the extra work involved in computation of such a control law is made up for in the improved control over the response of the system.

In this study, the command vector was chosen as roll rate ($\dot{\phi}$) and sideslip angle (β)--both variables considered valuable in aircraft control. In addition, since the state vector contains roll angle (ϕ), an integral of roll rate, control of that same variable was effectively added. The control law configuration, then, is represented in Figure 1.

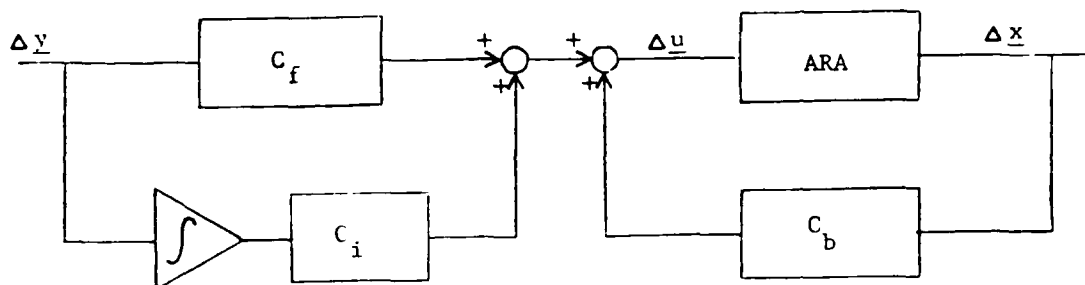


Figure 1: Control Law Configuration

2.3 NONDIMENSIONAL FORCE AND MOMENT COEFFICIENTS

Perhaps the most important part of modelling is the determination of the nondimensional coefficients--side force (C_y), yaw moment (C_n), and roll moment (C_l). The coefficients are affected by the states, the controls, and the flight condition, and they are nonlinear. The task, then, was to reduce the available data to a set of polynomials from which the coefficients could be calculated.

In the form of equations (2-16) to (2-18), the nondimensional coefficients were linear functions of the state and controls. However, the coefficients of these equations (which for sake of clarity, shall be called coefficient components) were nonlinear functions of the flight condition. Equations for these components were derived which would allow the coefficients to be calculated for any specified flight condition and state and control vectors. The coefficient components, for the purposes of this report, are broken down and discussed in four categories:

the constant components (C_{y_0} , C_{n_0} , C_{l_0}); the stability derivative components (C_{y_B} , C_{n_B} , C_{l_B}); the rotary derivative components (C_{y_P} , C_{n_P} , C_{l_P} , C_{y_R} , C_{n_R} , C_{l_R}); and the control derivative components ($C_{y_{dK}}$, $C_{n_{dK}}$, $C_{l_{dK}}$, $C_{y_{dA}}$, $C_{n_{dA}}$, $C_{l_{dA}}$).

The aircraft for which this control system was designed was tested extensively in the wind tunnel and the data presented in NASA TN D-5857. The NASA report lists data from which the constant components, the stability derivative components, and the control derivative components could be derived directly. The method used to reduce the data was generally the same in each case and is discussed in detail in the constant component section. Since there was a considerable amount of data on these components, certain assumptions were made to facilitate the data reduction. In particular, throttle effects were considered linear, velocity was assumed to affect only the dimensional components (through dynamic pressure), and state and control variables were assumed to affect the coefficients linearly. The validity of each of these assumptions is discussed below.

The data which were available for the other components were not available for the rotary derivatives. The report NASA TN D-6643 did present some experimental values of these components but for only one flight condition. For this study, it did not seem wise to assume that the rotary derivatives would stay constant throughout the range of flight conditions. Therefore, the USAF DATCOM methods were used, making it possible to reduce

even the rotary derivative components to functions of the flight condition.

2.3.1 Constant Components

The static flight data included in NASA TN D-5857 were presented as sets of curves showing how each coefficient changes as the state, controls, and flight condition vary. In general, there was one plot per throttle setting with each plot having several curves plotted against angle of attack. The curves each correspond to a different value of a state or control variable. Each curve was plotted with ten explicit data points (corresponding to angles of attack of -4, 0, 4, 8, 12, 14, 16, 18, 20, and 22 deg). The first step taken for data reduction was to make a table of eight data points (the explicit points at -4, 0, 4, 8, 12, 16, and 20 deg angle of attack and one at 24 deg found by extrapolating the curves) at each throttle position and angle of attack. The data are presented in Appendix A.

The data were reduced to a linear equation where the coefficients of that equation were functions of angle of attack only. The form of these equations was

$$C_{Y_O} = C_{Y_{O_T}} T_c + C_{Y_{O_C}} \quad (2-34)$$

$$C_{n_O} = C_{n_{O_T}} T_c + C_{n_{O_C}} \quad (2-35)$$

$$C_{l_O} = C_{l_{O_T}} T_c + C_{l_{O_C}} \quad (2-36)$$

The value of the slope and constant terms were found by doing a linear regression at each angle of attack. The assumption of throttle linearity proved to be valid in this case. Curves were fitted to the slope and constant points, which were functions of angle of attack alone. (In the case of C_{l_0} , however, it was apparent that curve fitting would be very difficult for the slope term, C_{l_0} . Therefore, an average C_{l_0} was calculated for the three throttle settings used, essentially neglecting throttle effects).

The curve fitting method used assumed a solution in the form of an n-th order polynomial:

$$C = A_n a^n + A_{n-1} a^{n-1} + \dots + A_1 a^1 + A_0 \quad (2-37)$$

In this case, a 7-th order equation was used such that eight values of the dependent and independent variables were required. A matrix equation was then set up in the following manner:

$$\begin{bmatrix} C(a_1) \\ C(a_2) \\ . \\ . \\ . \\ C(a_8) \end{bmatrix} = \begin{bmatrix} 1 & a_1 & a_1^2 & . & . & . & a_1^7 \\ 1 & a_2 & a_2^2 & . & . & . & a_2^7 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 1 & a_8 & a_8^2 & . & . & . & a_8^7 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ . \\ . \\ . \\ A_7 \end{bmatrix} \quad (2-38)$$

where $C(a_i)$ is the value of the coefficient at angle of attack, a_i . More simply:

$$\underline{C} = \underline{M} \underline{A} \quad (2-39)$$

To solve for the coefficients, all that was required was a matrix inversion and multiplication by the coefficient vector:

$$\underline{A} = M^{-1} \underline{C} \quad (2-40)$$

(Since the same eight angles of attack -- -4, 0, 4, 8, 12, 16, 20, 24 deg -- always were used whether calculating the constant components, the stability derivative components, or the control derivative components, M and hence M^{-1} were always the same. Thus, it was necessary only to assemble \underline{C} to find the coefficients, \underline{A} , for any component coefficient).

The final reduced constant component equations are summarized as follows:

$$C_{Y_{C_T}} = 1.094E-8 a^7 - 7.104E-7 a^6 + 1.630E-5 a^5 - 1.490E-4 a^4 + \\ 2.979E-4 a^3 + 2.446E-3 a^2 - 1.715E-2 a + 0.0$$

$$C_{Y_{C_O}} = -1.996E-9 a^7 + 1.298E-7 a^6 - 2.974E-6 a^5 + 2.713E-5 a^4 - \\ 5.525E-5 a^3 - 4.614E-4 a^2 + 1.812E-3 a + 0.0$$

$$C_{r_{C_T}} = 5.553E-9 a^7 - 3.259E-7 a^6 + 6.695E-6 a^5 - 5.416E-5 a^4 + \\ 7.633E-5 a^3 + 1.025E-3 a^2 - 4.504E-3 a - 0.03281$$

$$C_{r_{C_O}} = 3.817E-10 a^7 - 2.568E-8 a^6 + 6.104E-7 a^5 - 5.881E-6 a^4 + \\ 1.520E-5 a^3 + 8.592E-5 a^2 - 5.593E-4 a + 0.001822$$

$$C_{l_O} = -7.099E-10 a^7 + 4.458E-8 a^6 - 9.659E-7 a^5 + 7.877E-6 a^4 - \\ 6.971E-6 a^3 - 1.637E-4 a^2 + 2.167E-4 a + 0.00717$$

2.3.2 Stability Derivative Components

The method for finding the stability derivative component equations was essentially the same as for the constant components except that one initial step -- the computation of the stability derivatives -- was included. The data were presented as one plot per throttle setting with seven curves per plot. The curves (each corresponding to a different value of sideslip -- -15, -10, -5, 0, 5, 10, and 15 deg) showed how the coefficients varied with angle of attack. Once again, eight points per curve were selected, and these are presented in Appendix A.

The first task was to compute the stability derivatives for each angle of attack and throttle setting. Since the coefficients were assumed linear with respect to the states, a linear regression was used for this task. The assumption of linear sideslip effects was valid except for C_n and C_l at high angles of attack. The stability derivatives were reduced to the form:

$$C_{y_E} = C_{y_{E_T}} \tau_c + C_{y_{E_O}} \quad (2-47)$$

$$C_{n_E} = C_{n_{E_T}} \tau_c + C_{n_{E_O}} \quad (2-48)$$

$$C_{l_E} = C_{l_{E_T}} \tau_c + C_{l_{E_O}} \quad (2-49)$$

Since only two throttle settings were given in the data, no analysis of throttle linearity was required. Once again, linear regression was used to determine the equations for the stability derivatives. The curve fitting method previously discussed was

used to determine the equations for the slope and constant terms. The final reduced stability derivative component equations are summarized as follows:

$$C_{Y_{E_7}} = 1.274E-9 a^7 - 8.630E-8 a^6 + 2.089E-6 a^5 - 2.044E-5 a^4 + 4.791E-5 a^3 + 3.116E-4 a^2 - 1.607E-3 a - 0.0188$$

$$C_{Y_{E_0}} = 2.380E-4 a - 0.01249 \quad (C_{Y_{E_0}} \text{ was assumed a straight line})$$

$$C_{n_{E_7}} = -2.004E-11 a^7 + 2.584E-9 a^6 - 8.590E-8 a^5 + 1.123E-6 a^4 - 4.370E-6 a^3 - 2.051E-5 a^2 + 3.196E-4 a + 0.001$$

$$C_{n_{E_0}} = 3.033E-12 a^7 - 4.747E-10 a^6 + 1.815E-8 a^5 - 2.599E-7 a^4 + 1.068E-6 a^3 + 3.907E-6 a^2 - 8.049E-5 a + 0.002$$

$$C_{l_{E_7}} = -3.283E-10 a^7 + 2.310E-8 a^6 - 5.742E-7 a^5 + 5.638E-6 a^4 - 1.109E-5 a^3 - 9.987E-5 a^2 + 4.262E-4 a - 0.00106$$

$$C_{l_{E_0}} = 6.697E-11 a^7 - 4.466E-9 a^6 + 1.063E-7 a^5 - 1.012E-6 a^4 + 1.851E-6 a^3 + 1.483E-5 a^2 - 1.463E-5 a - 0.00155$$

2.3.3 Rotary Derivative Components

Since no flight test or wind tunnel data were available showing the variation of the nondimensional coefficients with changes in roll rate and yaw rate, the USAF STABILITY AND CONTROL DATCOM methods were used to find the rotary derivatives. In general, the rotary derivatives are functions of the asymmetrical distribution of lift and drag over the wing panels caused by

rolling and yawing. Because of this, the nondimensional lift and/or drag coefficients, C_L and C_D , were found in every equation for the rotary derivative components. Therefore, these coefficients were derived first, as functions of the flight condition. Each derivative is discussed in turn.

Every rotary derivative experiences compressibility effects at sufficiently high Mach numbers. However, since the velocity of the aircraft is decidedly subsonic ($M < .2$), a constant Mach number was used in the equation formulation. Since the static data in the NASA TN D-5857 was taken at 93 feet/second, the equivalent Mach number of .085 was used.

A number of other terms were common to all the rotary derivative component equations, as well. One of these terms, $(\Delta C_{Y_E})_{v(wEH)}$, is a tail-body sideslip derivative specified by:

$$(\Delta C_{Y_E})_{v(wEH)} = -k(C_{L_a})_{v(EH)}(1 + \partial \sigma / \partial \beta)(q_v/q_w)(S_v/S_w) \quad (2-56)$$

Using aircraft constants and charts available in USAF DATCOM,

$(\Delta C_{Y_E})_{v(wEH)}$ reduced to:

$$(\Delta C_{Y_E})_{v(wEH)} = -0.0181$$

In addition, a number of other constants relating to the aircraft (e.g. z , z_P , l_P , b_w , b_H , s_w , s_H) were present in the equations and can be found in Appendix A.

Longitudinal Nondimensional Coefficients

Since neither C_L nor C_D are significantly affected by the lateral-directional parameters, only variations with respect to the flight condition were considered. Variations with respect to changes in the longitudinal states and controls were not required. The data values are listed in Appendix A.

First, the coefficients were reduced to linear functions of the thrust coefficient:

$$C_L = C_{L_T} T_C + C_{L_O} \quad (2-57)$$

$$C_L = C_{L_T} T_C + C_{L_O} \quad (2-58)$$

where the coefficients of these equations were found at each angle of attack using linear regression. The assumption of linear throttle effects was valid in this case.

The curve fitting method previously discussed was used to find the equations of the slope and constant terms for the longitudinal coefficients. The equations are summarized as follows:

$$C_{L_T} = 2.780E-8 \alpha^7 - 1.810E-6 \alpha^6 + 4.299E-5 \alpha^5 - 4.463E-4 \alpha^4 + \\ 1.530E-3 \alpha^3 + 6.761E-3 \alpha^2 - 1.321E-3 \alpha + 0.357$$

$$C_{L_O} = -5.571E-10 \alpha^7 + 8.952E-8 \alpha^6 - 3.664E-6 \alpha^5 + 5.420E-5 \alpha^4 - \\ 2.531E-4 \alpha^3 - 9.214E-4 \alpha^2 + 0.06337 \alpha + 0.121$$

$$C_{L_T} = -1.401E-8 \alpha^7 + 1.045E-6 \alpha^6 - 2.756E-5 \alpha^5 + 2.968E-4 \alpha^4 -$$

$$9.036E-4 a^3 - 0.003391 a^2 + 0.02157 a - 1.053$$

$$C_{D_0} = -0.539E-10 a^7 + 2.606E-8 a^6 - 1.176E-7 a^5 - 5.244E-6 a^4 + 7.751E-5 a^3 - 2.278E-5 a^2 + 4.268E-5 a + 0.0551$$

Side-Force-Due-to-Rolling Component

The side-force-due-to-rolling component, C_{y_F} , was computed using the following equation:

$$(C_{y_F})_{WB_T} = (C_{y_F})_{WB} + 2((z-z_P)/b_w)(\Delta C_{y_B})_v(WBL) \quad (2-63)$$

where $(C_{y_F})_{WB_T}$ is the component for the entire wing-body-tail combination and $(C_{y_F})_{WB}$ is the component for the wing-body combination.

$(C_{y_F})_{WB}$ was calculated using the following equation:

$$(C_{y_F})_{WB} = K((C_{y_F}/C_L)C_L) + (\Delta C_{y_F})_T \quad (2-64)$$

where

$$K = \frac{\frac{\partial}{\partial \alpha}(C_L \tan(\alpha)) - \frac{\partial}{\partial \alpha}(C_D - C_{D_0})}{\frac{\partial}{\partial \alpha}(C_L \tan(\alpha)) - \frac{\partial}{\partial \alpha}(C_L^2/\pi AR)} \quad (2-65)$$

$$\frac{C_{y_F}}{C_L} \Big|_{C_L=0, M_L} = \frac{AR+4 \cos(\Lambda_{c/4})}{AR \beta + 4 \cos(\Lambda_{c/4})} \quad (2-66)$$

and

$$(\Delta C_{y_F})_T = (3 \sin(\Gamma) (1 - 2 \frac{z}{b/2})) (C_{y_F})_T \Big|_{C_L=0} \quad (2-67)$$

Since C_L and C_D are functions of the flight condition, K was reduced to a polynomial function of angle of attack and is summarized below:

$$K = 4.722E-6 \alpha^7 - 1.83E-6 \alpha^6 + 4.235E-6 \alpha^5 + 5.208E-4 \alpha^4 - 0.00590 \alpha^3 + 0.007230 \alpha^2 + 0.0870 \alpha + 1.00$$

By assuming constant Mach number and using aircraft constants and charts available in USAF DATCOM, (C_{Y_F}/C_L) and (C_{Y_F}) reduced to:

$$\begin{aligned} (C_{Y_F}/C_L) &= -.06 \\ (C_{Y_F}) &= .161 \end{aligned}$$

Yaw-Moment-Due-to-Roll Component

The yaw-moment-due-to-roll component, C_{n_P} , was computed using the following equation:

$$(C_{n_P})_{WB T} = (C_{n_P})_{WB} - (z/l_w)(l_F \cos(\alpha) + z_P \sin(\alpha))((z-z_P)/l_w) * (\Delta C_{Y_F})_v(WBH) \quad (2-69)$$

where $(C_{n_P})_{WB T}$ is the component for the wing-body-tail combination and $(C_{Y_F})_{WB}$ is the component for the wing-body combination.

$(C_{n_P})_{WB}$ was calculated using the following equation:

$$(C_{n_P})_{WB} = -C_{l_F} \tan(\alpha) - K(-C_{l_F} \tan(\alpha) - (C_{n_P}/C_L)C_L) + (C_{n_P}/\epsilon)\epsilon \quad (2-70)$$

where

$$\frac{C_{n_p}}{C_L} \bigg|_{\substack{C_L=0 \\ M=0}} = -\frac{1}{6} \frac{AR + 6 (AR + \cos(\Lambda_{c/4})) \left(\frac{\bar{x}}{c} \frac{\tan(\Lambda_{c/4})}{AR} + \frac{\tan(\Lambda_{c/4})}{12} \right)}{AR + 4 \cos(\Lambda_{c/4})} \quad (2-71)$$

K was previously discussed, and C_{1_p} is the roll damping component to be discussed later. By assuming constant Mach number and using aircraft constants and charts available in USAF DATCOM, (C_{n_p}/C_L) and (C_{n_p}/ϵ) reduced to:

$$(C_{n_p}/C_L) = -.1003$$

$$(C_{n_p}/\epsilon) = 0.0$$

Roll Damping Component

The roll damping component, C_{1_p} , was computed using the following equation:

$$(C_{1_p})_{wLT} = (C_{1_p})_{wL} + .5(C_{1_p})_E \left(\frac{s_E/s_w}{b_E/L_w} \right)^2 + 2 \left(\frac{z}{L_w} \right) * \left(\frac{(z-z_P)}{L_w} \right) (\Delta C_{Y_E})_{V(wLL)} \quad (2-72)$$

where $(C_{1_p})_{wLT}$ is the component for the wing-body-tail combination and $(C_{1_p})_{wL}$ is the component for the wing-body combination.

$(C_{1_p})_{wL}$ was calculated by the following equation:

$$(C_{1_p})_w = (\beta C_{1_p}/k)_{C_L=0} (k/\beta) (C_{L_a})_{C_L=0} / (C_{L_a})_{C_L=0} (C_{1_p})_T / (C_{1_p})_{T=C} + (\Delta C_{1_p})_{drag} \quad (2-73)$$

where

$$(C_{1P})_{\Gamma}/(C_{1P})_{\Gamma=0} = 1 - 2(z/(b/2))\sin(\Gamma) + 3(z/(b/2))^2\sin^2(\Gamma) \quad (2-74)$$

$$(\Delta C_{1P})_{\text{drag}} = (C_{1P})_{C_{D_L}}/C_L^2 * C_L^2 - (1/8)C_{L_0} \quad (2-75)$$

$$\beta = \sqrt{1 - M^2} \quad (2-76)$$

$$k = (C_{1a})_M / (2\pi/\beta) \quad (2-77)$$

$$(C_{1a})_M = (1.05/\beta)(C_{1a}/(C_{1a})_{\text{theory}})(C_{1a})_{\text{theory}} \quad (2-78)$$

Assuming constant Mach number and using aircraft constants and charts available in USAF DATCOM, $(C_{1P})_{\Gamma}/(C_{1P})_{\Gamma=0}$, β , k , and $(\beta C_{1P}/k)$ reduced to:

$$(C_{1P})_{\Gamma}/(C_{1P})_{\Gamma=0} = .985$$

$$\beta = .990$$

$$k = .992$$

$$(\beta C_{1P}/k) = -.42$$

Since both $(\Delta C_{1P})_{\text{drag}}$ and (C_{1a}/C_{1a}) are functions of the lift and drag coefficients, it was possible to reduce the equations to a set of polynomial equations in angle of attack, summarized below:

$$\frac{(C_{L_a})}{(C_{L_a})_{C_{L=0}}} = -4.844E-10 a^7 + 1.797E-7 a^6 - 6.524E-6 a^5 + 6.649E-5 a^4 -$$

$$2.235E-5 a^3 - 1.326E-3 a^2 - 1.720E-3 a + 0.99$$

$$\Delta C_{1P} = 8.453E-10 a^7 - 5.086E-8 a^6 + 1.000E-6 a^5 - 9.040E-6 a^4 +$$

$$1.985E-5 a^3 + 2.109E-4 a^2 + 9.369E-5 a + 0.1329$$

$$\Delta C_{1P_0} = 2.325E-11 a^7 - 5.324E-10 a^6 - 1.689E-8 a^5 + 4.544E-7 a^4 - 3.006E-6 a^3 + 5.380E-5 a^2 + 2.386E-4 a - 0.00075$$

Side-Force-Due-to-Yaw Component

The USAF DATCOM lists no method for the side-force-due-to-yaw component. In addition, it states that the term is usually negligible for angles of attack up to stall. Therefore, this term was neglected for this study.

Yaw Damping Component

The yaw damping component, C_{n_r} , was computed using the following equation:

$$(C_{n_r})_{WB} = (C_{n_r})_{WL} + (2/b_w^2)(l_P \cos(\alpha) + z_P \sin(\alpha))^2 (\Delta C_{Y_E})_v(WL) \quad (2-79)$$

where $(C_{n_r})_{WB}$ is the component for the wing-body-tail combination and $(C_{n_r})_{WL}$ is the component for the wing-body combination.

$(C_{n_r})_{WB}$ was calculated using the following equation:

$$(C_{n_r})_{WB} = (C_{n_r}/C_L^2)C_L^2 + (C_{n_r}/C_{D_0})C_{D_0} \quad (2-80)$$

where

$$C_{D_0} = C_D - C_L^2/(\pi AR) \quad (2-81)$$

Assuming constant Mach number and using aircraft constants and charts available in USAF DATCOM, (C_{n_r}/C_L^2) and (C_{n_r}/C_{D_0}) reduced to:

$$\begin{aligned}(C_{n_r}/C_L^2) &= -.02 \\ (C_{n_r}/C_{D_0}) &= -.32\end{aligned}$$

Roll-Moment-Due-to-Yaw Component

The roll-moment-due-to-yaw component, C_{l_r} , was calculated using the following equation:

$$\begin{aligned}(C_{l_r})_{WET} &= (C_{l_r})_{WB} - (z_w^2/b_w^2)(l_p \cos(\alpha) + z_p \sin(\alpha)) * \\ &\quad (z_F \cos(\alpha) - l_F \sin(\alpha))(\Delta C_{y_E})_{(WET)}\end{aligned}\quad (2-82)$$

where $(C_{l_r})_{WET}$ is the component for the wing-body-tail combination and $(C_{l_r})_{WB}$ is the component for the wing-body combination.

$(C_{l_r})_{WB}$ was computed by the following equation:

$$(C_{l_r})_{WB} = (C_{l_r}/C_L)C_L = C_{L_r} + (C_{l_r})C_L + (\Delta C_{l_r}/\Gamma)\Gamma \quad (2-83)$$

where

$$\begin{aligned}\frac{C_{l_r}}{C_L} \bigg|_{C_L^2=0} &= \frac{1 + \frac{AR(1-\epsilon^2)}{2} + \frac{AR + 2 \cos(\lambda_c/4)}{AR\beta + 4 \cos(\lambda_c/4)} \frac{\tan^2(\lambda_c/4)}{8}}{1 + \frac{AR + 2 \cos(\lambda_c/4)}{AR + 4 \cos(\lambda_c/4)} \frac{\tan^2(\lambda_c/4)}{8}} \frac{C_{l_r}}{C_L} \bigg|_{C_L^2=0} \quad (2-84)\end{aligned}$$

$$\frac{\Delta C_{l_r}}{\Gamma} = \frac{1}{12} \frac{\pi AR \sin(\lambda_c/4)}{AR + 4 \cos(\lambda_c/4)} \quad (2-85)$$

Assuming constant Mach number and using aircraft constants and charts available in USAF DATCOM, these terms reduced to:

$$(C_{l_r}/C_L) = .241$$
$$(\Delta C_{l_r}/\Gamma) = .001079$$

2.3.4 Control Derivative Components

As in the cases of the constant and stability derivative components, static data were available for the control derivatives. Because the data were not presented in the same way, rudder and aileron are discussed separately.

Rudder Derivative Components

The method for finding the rudder derivative component equations was the same as the stability derivative components. Static data were presented in one plot per throttle setting with five curves per plot. Each curve (corresponding to a different value of rudder deflection -- -17.5, -9.0, 0.0, 7.0, and 13.2 deg) showed how the coefficients varied with angle of attack. Once again, eight points per curve were selected, and they are presented in Appendix A.

The first task was to compute the rudder derivatives at each angle of attack and throttle setting using a linear regression. The linear assumption was valid for both C_y and C_n but not so

good for C_1 . Since it was desirable to use as simple a model as possible, linearity for C_1 was assumed, although the assumption could be reconsidered if the results were not satisfactory. The rudder derivatives then were reduced to the form:

$$C_{Y_{dK}} = C_{Y_{dK_T}} \tau_c + C_{Y_{dK_O}} \quad (2-86)$$

$$C_{n_{dK}} = C_{n_{dK_T}} \tau_c + C_{n_{dK_O}} \quad (2-87)$$

$$C_{l_{dK}} = C_{l_{dK_T}} \tau_c + C_{l_{dK_C}} \quad (2-88)$$

Since only two throttle points were used, no investigation of the linearity of throttle was necessary. A linear regression was also used for this task. The slope and constant terms were then curve fitted using the method previously discussed. The final reduced equations are summarized below:

$$C_{Y_{dF_T}} = 6.225E-10 a^7 - 4.079E-8 a^6 + 9.409E-7 a^5 - 8.651E-6 a^4 + 1.017E-5 a^3 + 1.582E-4 a^2 - 2.147E-4 a + 0.0008$$

$$C_{Y_{dF_C}} = 4.687E-11 a^7 - 3.079E-9 a^6 - 7.235E-8 a^5 - 6.912E-7 a^4 + 1.095E-6 a^3 + 6.535E-6 a^2 - 1.208E-5 a + 0.00290$$

$$C_{n_{dF_T}} = -8.158E-11 a^7 + 4.476E-9 a^6 - 7.650E-8 a^5 + 5.046E-7 a^4 - 1.104E-6 a^3 - 4.531E-6 a^2 - 8.110E-5 a - 0.0040$$

$$C_{n_{dF_C}} = -7.206E-11 a^7 + 4.832E-9 a^6 - 1.141E-7 a^5 + 1.077E-6 a^4 - 2.106E-6 a^3 - 1.690E-5 a^2 + 2.946E-5 a - 0.00150$$

$$C_{l_{dF_T}} = 3.278E-10 a^7 - 2.428E-8 a^6 + 0.533E-7 a^5 - 7.240E-6 a^4 + 2.069E-5 a^3 + 1.294E-4 a^2 - 6.396E-4 a + 0.00166$$

$$C_{l_{dF_O}} = -1.381E-10 a^7 + 9.568E-9 a^6 - 2.394E-7 a^5 + 2.467E-6 a^4 -$$

$$6.880E-6 a^3 - 4.086E-5 a^2 + 1.717E-4 a + 0.000247$$

Aileron Derivative Components

The only difference between the aileron derivatives and the rudder derivatives was that the aileron derivatives were presented in NASA TN D-5857 with only one throttle setting; hence, one step in the process was deleted. The data were presented as five curves per plot (corresponding to aileron values of -42, -21, 0, 21, and 42 deg) showing the effect of angle of attack and are summarized in Appendix A. The curve fitting method previously discussed was used to reduce the data to a final set of equations summarized as follows:

$$C_{Y_{d\alpha}} = 2.433E-11 a^7 - 1.603E-9 a^6 + 3.859E-8 a^5 - 3.817E-7 a^4 + \\ 8.667E-7 a^3 + 5.518E-6 a^2 - 2.085E-5 a - 0.000267$$

$$C_{l_{d\alpha}} = -1.037E-11 a^7 + 1.088E-9 a^6 - 2.590E-8 a^5 + 2.497E-7 a^4 - \\ 5.133E-7 a^3 - 4.023E-6 a^2 + 3.243E-5 a - 0.000005$$

$$C_{l_{d\delta}} = -1.381E-11 a^7 + 9.800E-10 a^6 - 2.405E-8 a^5 + 2.227E-7 a^4 - \\ 2.285E-7 a^3 - 5.377E-6 a^2 + 3.619E-6 a - 0.00127$$

2.4 OPEN-LOOP RESULTS

The final step in the model development was verification of the model. To do this, a number of test flight conditions were used to determine the linearized system dynamic equations, F and G . From these, system eigenvalues and dimensional derivatives were determined. Table 1 lists the results.

Twenty-seven flight condition combinations were used representing three different values for each of the three flight condition variables. The three values were chosen as the maximum, minimum, and midpoint values for each of the variable's typical ranges. In particular, the values for angle of attack were -4 , 10 , and 24 deg; for throttle setting, they were $.03$, $.13$, and $.23$; and for dynamic pressure, they were 9.731 , 21.894 , and 38.922 pounds per square foot (corresponding to velocities of 100 , 150 , and 200 feet per second).

The eigenvalues give a good representation of the basic aircraft. At low angles of attack (-4°), no instabilities are noted. At moderate angles of attack (10°), an unstable spiral mode with a long time constant is encountered. Finally, at high angles of attack (24°), an unstable roll-spiral develops. This corresponds to the wing rock instability which has been noted in this flight condition regime.

In addition, the rotary derivatives were checked against experimental data in NASA TN D-6643. The comparison was done

using for the model at a nominal condition (angle of attack = 10° ; throttle = .13; velocity = 150 ft/sec). The yaw damping (C_{n_r}) and the roll damping (C_{l_p}) terms correspond well with accepted values. The yaw damping for the model was -0.125 compared with -0.125 for the data, while the roll damping was -0.376 compared with -0.41. The cross-coupling derivatives, however, both were off by a factor of two. Yaw-due-to-roll (C_{n_p}) was -0.110 for the model compared to -0.056 in the data, while roll-due-to-yaw (C_{l_r}) was 0.236 compared to 0.107. Since no error could be found in the calculations and since the complete model appeared to give good results despite this discrepancy, the cross-coupling derivatives were left as derived. This assumption could be reconsidered if later results were not satisfactory.

2.5 MODEL REEVALUATION

After this project was finished, an inspection of the model was done and two major errors were found along with other problems noted in later Chapters, these errors may prove to be the reason the control system was not a complete success. This section has been added after the fact to point out these problems and is included in this chapter to be consistent with the thesis organization.

Both model errors pertain to assumptions that were made which are not valid at high angles of attack. The first involved the inertia matrix; the second involved the vertical component

of the velocity vector. The model errors in turn affect the control law and the eventual outcome of the project.

2.5.1 Inertia Matrix

Due to the emphasis on angle of attack that this project used, the stability axes provided a better basis for an axis system than did the body axes. Indeed, much of the aerodynamic data that was used to formulate the model was based on the stability-axis system. The body-axis equations of motion were rotated through the angle of attack for development of the control law. In doing so, however, the rotation effects on the inertia matrix were not considered. Hence, the model essentially used stability-axis aerodynamics and body-axis inertias. At low angles of attack (less than 5 degrees), this effect can be considered negligible. However, this control system was required to perform at high angles of attack and the inertia changes are significant.

To repair this oversight, the inertia matrix needs to be rotated to the stability axis system and can be done by the following equation:

$$I_S = H_B^S I_B H_S^B$$

I_B is defined as the body-, or principle-axis inertia matrix as follows:

$$I_B = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}$$

H_B^S and H_S^B are defined as follows:

$$H_B^S = \begin{bmatrix} \cos\alpha_O & 0 & \sin\alpha_O \\ 0 & 1 & 0 \\ -\sin\alpha_O & 0 & \cos\alpha_O \end{bmatrix}$$

$$H_S^B = H_B^{S^{-1}} = H_B^{ST}$$

By working through the equation, the following relationships result:

$$I_{xx_S} = I_x \cos^2\alpha_O + I_z \sin^2\alpha_O$$

$$I_{yy_S} = I_y$$

$$I_{zz_S} = I_x \sin^2\alpha_O + I_z \cos^2\alpha_O$$

$$I_{xz_S} = \frac{1}{2}(I_z - I_x) \sin 2\alpha_O$$

and the stability-axis inertia matrix is as follows:

$$I_S = \begin{bmatrix} I_{xx_S} & 0 & -I_{xz_S} \\ 0 & I_{yy_S} & 0 \\ -I_{xz_S} & 0 & I_{zz_S} \end{bmatrix}$$

Perhaps the most significant effect of this rotation is the appearance of the product-of-inertia term, I_{xz_S} . This term affects the rotational dynamics of the aircraft.

The rotational dynamics are defined as follows:

$$\dot{\omega}_S = I_S^{-1} [M_S - \tilde{\omega}_S I_S \omega_S]$$

where the terms are as follows:

$$M_S = \begin{bmatrix} C_\ell \bar{q} S b \\ C_m \bar{q} S c \\ C_n \bar{q} S b \end{bmatrix}_S$$

$$\tilde{\omega}_S = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}_S$$

$$\omega_S = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_S$$

Working through the equations and assuming pitch rate (q) is negligible, the following equations result:

$$\dot{p}_S = \left(\frac{1}{I_{xx_S} I_{zz_S} - I_{xz_S}^2} \right) [C_\ell \bar{q} S b I_{zz_S} + C_n \bar{q} S b I_{xz_S}]$$

$$\dot{r}_S = \left(\frac{1}{I_{xx_S} I_{zz_S} - I_{xz_S}^2} \right) [C_l \bar{q} S b I_{xz_S} + C_n \bar{q} S b I_{xx_S}]$$

Comparing these equations to (2-7) and (2-9), the inertia effects are readily apparent: the higher the angle of attack, the larger the inertia effects.

2.5.2 Vertical Component of Velocity

The rate of change of the lateral velocity component, \dot{v} , and hence the rate of change of sideslip, $\dot{\beta}$, are functions of the vertical component of velocity, w . At low angles of attack, w is very small compared to u , the forward velocity component, but at high angles of attack, w takes on significance. This term, however, was inadvertently neglected (assumed zero) and hence does not appear in equation (2-8) for $\dot{\beta}$.

The result of this oversight is the miscalculation of the F matrix. The Δp coefficient of the $\Delta \dot{\beta}$ equation changes from Y_p/V_O to $Y_p/V_O + w/V_O$. At high angles of attack (e.g. 24 degrees), the coefficient becomes

$$Y_p/V_O + w/V_O = 0.4$$

instead of

$$Y_p/V_O = 0.008$$

The significance of this error, then, is readily apparent.

Table 1

Open-Loop Results

FLIGHT CONDITION	EIGENVALUES	STABILITY DERIVATIVES				ROTARY DERIVATIVES				CONTROL DERIVATIVES						
		N_R	Y_R/V_o	L_R	N_r	Y_r/V_o	L_r	N_p	Y_p/V_o	L_p	$N_{\delta R}$	$Y_{\delta R}/V_o$	$L_{\delta R}$	$N_{\delta A}$	$Y_{\delta A}/V_o$	$L_{\delta A}$
-4 .03 9.731	-1782 ± 1.5991j -3.3327 -.0741	2.363	-.157	-4.816	-.339	0	.170	.117	.005	-3.277	-1.601	.033	.908	-.075	-.003	-3.387
-4 .03 21.894	-3336 ± 2.3074j -4.9038 -.0539	5.316	-.235	-10.836	-.494	0	-.255	.176	.005	-4.916	-3.602	.049	2.043	-.169	-.004	-7.620
-4 .03 38.922	-4964 ± 3.0328j -6.4921 -.0417	9.450	-.313	-19.264	-.659	0	-.340	.234	.005	-6.554	-6.404	.065	3.632	-.301	-.007	-13.547
-4 .13 9.731	-1798 ± 1.6123j -3.2458 -.0765	2.369	-.177	-5.224	-.329	0	-.132	.108	.005	-3.175	-1.998	.039	1.535	-.075	-.003	-3.387
-4 .13 21.894	-3534 ± 2.3178j -4.7598 -.0562	5.330	-.265	-11.754	-.494	0	-.197	.162	.005	-4.763	-4.496	.059	3.453	-.169	-.005	-7.620
-4 .13 38.922	-5136 ± 3.0419j -6.2931 -.0437	9.476	-.354	-20.895	-.659	0	-.263	.217	.005	-6.351	-7.992	.078	6.139	-.301	-.007	-13.547
-4 .23 9.731	-1804 ± 1.6262j -3.1606 -.0789	2.375	-.197	-5.632	-.329	0	-.093	.100	.005	-3.074	-2.395	.046	2.161	-.075	-.003	-3.387
-4 .23 21.894	-3623 ± 2.3292j -4.6174 -.0585	5.344	-.296	-12.671	-.494	0	-.139	.149	.005	-4.610	-5.389	.068	4.863	-.169	-.005	-7.620
-4 .13 38.922	-5298 ± 3.0524j -6.0956 -.0456	9.501	-.395	-22.527	-.659	0	-.186	.199	.005	-6.147	-9.580	.091	8.645	-.301	-.007	-13.547
10 .03 9.731	-2177 ± 1.5760j -3.1487j -.0406	1.608	-.122	-4.320	-.390	0	1.674	-.312	.006	-3.032	-1.869	.036	.350	.242	-.004	-3.563
10 .03 21.894	-3691 ± 2.2532j -4.6074 -.0303	3.619	-.182	-9.720	-.586	0	2.511	-.467	.006	-4.547	-4.205	.054	.788	.545	-.006	-8.017
10 .03 38.922	-5125 ± 2.9488j -6.0859 -.0237	6.433	-.243	-17.280	-.781	0	3.349	-.623	.006	-6.063	-7.476	.071	1.400	.970	-.008	-14.252
10 .13 9.731	-2516 ± 1.6844j -3.0235 -.0691	1.897	-.149	-4.074	-.392	0	1.835	-.33	.006	-2.916	-2.467	.047	.495	.242	-.004	-3.563
10 .13 21.894	-4124 ± 2.4206j -4.4127 -.0513	4.268	-.224	-9.167	-.588	0	2.753	-.506	.006	-4.374	-5.551	.070	1.113	.545	-.006	-8.017
10 .13 38.922	-5663 ± 3.1742j -5.8225 -.0400	7.588	-.299	-16.296	-.784	0	3.671	-.675	.006	-5.832	-9.868	.093	1.978	.970	-.008	-14.252

Table 1
continued

FLIGHT CONDITION	T_c	q	EIGENVALUES	STABILITY DERIVATIVES				ROLLING DERIVATIVES								CONTROL DERIVATIVES					
				N_B	Y_B/V_o	L_B	N_T	Y_T/V_o	L_T	N_P	Y_P/V_o	L_P	$N_{\delta R}$	$Y_{\delta R}/V_o$	$L_{\delta R}$	$N_{\delta A}$	$Y_{\delta A}/V_o$	$L_{\delta A}$			
10	.23	9.731	$-2.851 \pm 1.7894j$ -2.8981 .0965	2.186	-.177	-3.828	-.394	0	1.996	-.364	.006	-2.801	-3.065	.057	.639	.242	-.004	-3.563			
10	.23	21.894	$-4.564 \pm 2.5809j$ -4.2162 .0713	4.917	-.266	-8.613	-.591	0	2.994	-.545	.006	-4.201	-6.897	.086	1.438	.545	-.006	-8.017			
10	.23	38.922	$-.6217 \pm 3.3894j$ -5.5558 .0557	8.742	-.354	-15.313	-.787	0	3.993	-.727	.006	-5.602	-12.261	.115	2.556	.970	-.008	-14.252			
24	.03	9.731	$-.9436 \pm .8959j$.3567 \pm .6514j	1.400	-.080	-8.616	-.714	0	2.321	-.399	.008	-.380	-.867	.035	-.035	.828	.003	-2.965			
24	.03	21.894	$-1.5151 \pm 1.7827j$.7609 .5050	3.149	-.120	-19.385	-1.071	0	3.481	.599	.008	-.569	-1.950	.053	-1.864	.797	.005	-6.672			
24	.03	38.922	$-2.0765 \pm 2.5363j$ 1.5868 .2191	5.599	-.160	-34.462	-1.428	0	4.641	.799	.008	-.759	-3.466	.070	-3.313	1.416	.006	-11.860			
24	.13	9.731	$-.8778 \pm .7933j$.3502 \pm .5483j	1.279	-.094	-7.139	-.718	0	2.564	.391	.008	-.244	-1.085	.033	-.743	.354	.003	-2.965			
24	.13	21.894	$-1.4257 \pm 1.6231j$.9757 .2928	2.877	-.140	-16.062	-1.076	0	3.847	.586	.008	-.366	-2.441	.050	-1.673	.797	.005	-6.672			
24	.13	38.922	$-1.9604 \pm 2.3177j$ 1.6548 .1555	5.115	-.187	-28.555	-1.435	0	5.129	.782	.008	-.488	-4.339	.066	-2.973	1.416	.006	-11.860			
24	.23	9.731	$-.7944 \pm .6651j$.3258 \pm .3785j	1.158	-.107	-5.662	-.721	0	2.808	.382	.008	-.109	-1.303	.031	-.658	.354	.003	-2.965			
24	.23	21.894	$-1.3331 \pm 1.4401j$ 1.1209 .1395	2.605	-.161	-12.739	-1.082	0	4.212	.573	.008	-.163	-2.932	.047	-1.482	.797	.005	-6.672			
24	.23	38.922	$-1.8413 \pm 2.0684j$ 1.7271 .0808	4.632	-.214	-22.648	-1.443	0	5.617	.765	.008	-.217	-5.212	.062	-2.634	1.416	.006	-11.860			

Chapter III

DEVELOPMENT OF THE CONTROL LAW

The design of the command/stability augmentation system was accomplished using linear-quadratic control theory for a sampled data regulator. This method calculates an optimal feedback matrix, C , by minimizing a sampled-data cost function, J , specified by:

$$J = 1/2 \sum_{k=0}^{t_f} \left(\Delta \underline{x}_k^T Q_c \Delta \underline{x}_k + \Delta \underline{u}_k^T R_c \Delta \underline{u}_k \right) dt \quad (3-1)$$

(where Q_c and R_c are the continuous-time state and control weighting matrices, respectively) subject to a linear constraint specified by the sampled-data system dynamic equations:

$$\Delta \underline{x}_{k+1} = \tilde{G} \Delta \underline{x}_k + \tilde{\Gamma} \Delta \underline{u}_k \quad (3-2)$$

where \tilde{G} and $\tilde{\Gamma}$ are the sampled-data equivalents of the system dynamics equations, F and C .

A non-zero set point regulator was formulated as follows:

$$\Delta \underline{u} = \Delta \underline{u}^* - C (\Delta \underline{x} - \Delta \underline{x}^*) \quad (3-3)$$

where $\Delta \underline{x}^*$ and $\Delta \underline{u}^*$ are the equilibrium set points for the states and controls determined by the command, Δy^* , and where $\Delta \underline{x}$ and $\Delta \underline{u}$ are the current values of the states and controls. The objective of this regulator, then, is to drive the states and controls to their equilibrium values.

The controller design task was two-fold. First, the equilibrium values of the states and controls were determined given a specified command input. Since the state includes an integral of a command vector element, the singular command equilibrium method was used. The second task was determination of the optimal feedback gain matrix for a sampled-data regulator, C . Once these tasks were completed, the control law specified by (3-3) was developed.

Since the CAS was designed as a sampled-data controller, a sampling time had to be selected. The criterion for the selection was that the sampling time had to be long enough to enable all calculations for the control law to be completed and short enough that aircraft handling was not degraded. A sampling time of 0.1 seconds appeared to make a good compromise between the two conflicting objectives.

This chapter covers the determination of the singular command equilibrium, the calculation of the optimal gains, and the development of the control law. In addition, results are included showing: the selection process of the continuous-time matrices, Q_c and R_c ; a detailed description of the controller operation for a nominal flight condition, with simulation results; and a summary of closed-loop simulation results for 27 flight conditions.

3.1 SINGULAR COMMAND EQUILIBRIUM

The first step toward the design of the controller is the determination of the equilibrium point. The equilibrium point is defined as the desired value of the aircraft's states plus the assorted control settings. It is determined by the input command vector, Δy^* , and it is represented by Δx^* and Δu^* , the equilibrium values of the states and controls.

The system equations are denoted by:

$$\dot{\Delta x} = F \Delta x + G \Delta u \quad (3-4)$$

$$\Delta y = h_x \Delta x + h_u \Delta u \quad (3-5)$$

To examine the equilibrium, it would seem correct to set $\Delta x = 0$ and manipulate (3-4) and (3-5) as follows:

$$\begin{bmatrix} 0 \\ \Delta y \end{bmatrix} = \begin{bmatrix} F & G \\ h_x & h_u \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix} \quad (3-6)$$

$$\begin{bmatrix} \Delta x^* \\ \Delta u^* \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix} = \begin{bmatrix} F & G \\ h_x & h_u \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \Delta y^* \end{bmatrix} \quad (3-7)$$

(where $\Delta y^* = \Delta y$).

However, in the case of a singular command, $\begin{bmatrix} F & G \\ h_x & h_u \end{bmatrix}^{-1}$ does not exist, and a different approach (Ref.6) must be taken.

Singular equilibrium occurs when the state vector, $\Delta \underline{x}$, contains an integral of a command vector (Δy) element. While such a case has desirable aspects, it does mean that no true equilibrium can exist in the states and controls. A singular command variable (in this case, roll rate) is the derivative of a state variable (roll angle), so a non-zero value of the former prevents the latter from reaching any steady-state value (hence $\dot{x}_2^* \neq 0$). However, the disequilibrium in the singular variable may affect the nonsingular variables such that they, too, do not reach steady-state. While the disequilibrium in the nonsingular variables is small, it is still significant enough to affect the results and should not be neglected. Indeed, singular equilibrium implies no equilibrium at all (Ref. 5).

To develop the singular equilibrium, the singular and nonsingular variables are partitioned, resulting in the following equations:

$$\Delta \underline{x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} \text{nonsingular variables} \\ \text{singular variables} \end{bmatrix}$$

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \Delta u \quad (3-8)$$

$$\Delta y = \begin{bmatrix} h_{x_1} & h_{x_2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + h_u \Delta u \quad (3-9)$$

Thus, the following equations are applicable for finding the equilibrium values:

$$\dot{\Delta \underline{x}}_1^* = F_1 \Delta \underline{x}_1^* + F_{12} \Delta \underline{x}_2^* + G_1 \Delta \underline{u}^* \quad (3-10)$$

$$\dot{\Delta \underline{x}}_2^* = F_{21} \Delta \underline{x}_1^* + F_2 \Delta \underline{x}_2^* + G_2 \Delta \underline{u}^* \quad (3-11)$$

$$\Delta \underline{y}^* = H_{x_1} \Delta \underline{x}_1^* + H_{x_2} \Delta \underline{x}_2^* + H_u \Delta \underline{u}^* \quad (3-12)$$

and the following values are assumed:

$$\Delta \underline{x}_2^*(0) = \underline{0}$$

$$\Delta \underline{y}^* = \text{constant}$$

$$\dot{\Delta \underline{x}}_2^* = K \Delta \underline{y}^* \quad (\text{thus } \Delta \underline{x}_2^* = \text{constant})$$

Solving for \underline{x}_1^* and \underline{u}^* in (3-10) and (3-12), the following matrix equation results:

$$\begin{bmatrix} \Delta \underline{x}_1^* \\ \Delta \underline{u}^* \end{bmatrix} = \begin{bmatrix} F_1 & G_1 \\ H_{x_1} & H_u \end{bmatrix}^{-1} \begin{bmatrix} \dot{\Delta \underline{x}}_1^* - F_{12} \Delta \underline{x}_2^* \\ \Delta \underline{y}^* - H_{x_2} \Delta \underline{x}_2^* \end{bmatrix} \quad (3-13)$$

S can be defined as,

$$\begin{bmatrix} F_1 & G_1 \\ H_{x_1} & H_u \end{bmatrix}^{-1} = S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$$

where the partitions of S can alternately be found by:

$$s_{11} = F_1^{-1} (-G_1 s_{21} + 1) \quad (3-14)$$

$$s_{12} = -F_1^{-1} G_1 s_{22} \quad (3-15)$$

$$s_{21} = -s_{22} H_{x_1} F_1^{-1} \quad (3-16)$$

$$s_{22} = (-H_{x_1} F_1^{-1} G_1 + H_u)^{-1} \quad (3-17)$$

Thus, multiplied out, equation (3-13) becomes:

$$\Delta \underline{x}_1^* = s_{12} \Delta \underline{y}^* - (s_{11} F_{12} + s_{12} H_{x_2}) \Delta \underline{x}_2^* + s_{11} \dot{\Delta \underline{x}}_1^* \quad (3-18)$$

$$\Delta \underline{u}^* = S_{22} \Delta \underline{y}^* - (S_{21} F_{12} + S_{22} H_{x_2}) \Delta \underline{x}_2^* + S_{21} \Delta \underline{x}_1^* \quad (3-19)$$

Substituting (3-18) and (3-19) into (3-11) and gathering terms, the following equation results:

$$\begin{aligned} \Delta \underline{x}_2^* = & (F_{21} S_{12} + G_2 S_{22}) \Delta \underline{y}^* + (F_{21} S_{11} + G_2 S_{21}) \Delta \underline{x}_1^* \\ & (F_2 - F_{21} (S_{11} F_{12} + S_{12} H_{x_2}) - G_2 (S_{21} F_{12} + S_{22} H_{x_2})) \Delta \underline{x}_2^* \end{aligned} \quad (3-20)$$

Equation (3-20) can be simplified by making the following definitions:

$$K_Y = F_{21} S_{12} + G_2 S_{22} \quad (3-21)$$

$$K_X^* = F_{21} S_{11} + G_2 S_{21} \quad (3-22)$$

$$K_X = F_2 - F_{21} (S_{11} F_{12} + S_{12} H_{x_2}) - G_2 (S_{21} F_{12} + S_{22} H_{x_2}) \quad (3-23)$$

such that:

$$\Delta \underline{x}_2^* = K_Y \Delta \underline{y}^* + K_X^* \Delta \underline{x}_1^* + K_X \Delta \underline{x}_2^* \quad (3-24)$$

Recalling the assumption that $\Delta \underline{x}_2^* = K \Delta \underline{y}^*$:

$$\begin{aligned} \Delta \underline{x}_2^* &= K \Delta \underline{y}^* \\ &= K \begin{bmatrix} H_{x_1} & H_{x_2} \end{bmatrix} \begin{bmatrix} \Delta \underline{x}_1^* \\ \Delta \underline{x}_2^* \end{bmatrix} + K H_u \Delta u^* \\ &= K H_{x_1} \Delta \underline{x}_1^* + K H_{x_2} \Delta \underline{x}_2^* + K H_u \Delta u^* \end{aligned} \quad (3-25)$$

The following relationships are noted by comparing equation (3-25) to (3-11):

$$F_{21} = K H_{x_1} \quad (3-26)$$

$$F_2 = KH_{x_2} \quad (3-27)$$

$$C_2 = KH_u \quad (3-28)$$

These relationships coupled with equations (3-14) to (3-17) allowed (3-24) to be simplified. First, K_x^* is eliminated by manipulating (3-22):

$$\begin{aligned} K_x^* &= F_{21}S_{12} + C_2S_{22} \\ &= KH_{x_1}F_1^{-1}(-G_1S_{21} + 1) + KH_uS_{21} \\ &= KH_{x_1}F_1^{-1} - KH_{x_1}F_1^{-1}G_1S_{21} + KH_uS_{21} \\ &= K(H_{x_1}F_1^{-1} + (-H_{x_1}F_1^{-1}G_1 + H_u)S_{21}) \\ &= K(H_{x_1}F_1^{-1} + S_{22}^{-1}(-S_{22}H_{x_1}F_1^{-1})) \\ &= K(H_{x_1}F_1^{-1} - H_{x_1}F_1^{-1}) \\ &= 0 \end{aligned}$$

By the same reasoning, K_x is eliminated by manipulating (3-23):

$$\begin{aligned} K_x &= F_2 - F_{21}(S_{11}F_{12} + S_{12}H_{x_2}) - C_2(S_{21}F_{12} + S_{22}H_{x_2}) \\ &= KH_{x_2} - KH_{x_1}S_{11}F_{12} - KH_{x_1}S_{12}H_{x_2} - KH_uS_{21}F_{12} - KH_uS_{22}H_{x_2} \\ &= KH_{x_2} - KH_{x_1}F_1^{-1}(-G_1(-S_{22}H_{x_1}F_1^{-1}) + 1)F_{12} - \\ &\quad KH_{x_1}(-F_1^{-1}G_1S_{22})H_{x_2} - KH_u(-S_{22}H_{x_1}F_1^{-1})F_{12} - KH_uS_{22}H_{x_2} \\ &= KH_{x_2} - KH_{x_1}F_1^{-1}C_1S_{22}(H_{x_1}F_1^{-1}F_{12} - H_{x_2}) - KH_{x_1}F_1^{-1}F_{12} \\ &\quad KH_uS_{22}(H_{x_1}F_1^{-1}F_{12} - H_{x_2}) \\ &= KH_{x_2} + (KH_u - KH_{x_1}F_1^{-1}C_1)S_{22}(H_{x_1}F_1^{-1}F_{12} - H_{x_2}) - KH_{x_1}F_1^{-1}F_{12} \\ &= KH_{x_2} + K(H_u - H_{x_1}F_1^{-1}C_1)(-H_{x_1}F_1^{-1}C_1 + H_u)^{-1*} \\ &\quad (H_{x_1}F_1^{-1}F_{12} - H_{x_2}) - KH_{x_1}F_1^{-1}F_{12} \\ &= KH_{x_2} + K(H_{x_1}F_1^{-1}F_{12} - H_{x_2}) - KH_{x_1}F_1^{-1}F_{12} \\ &= 0 \end{aligned}$$

Thus, $\Delta \ddot{x}_2^*$ reduced to:

$$\begin{aligned}\Delta \dot{x}_2^* &= K_Y \Delta Y^* + \cancel{K_X \dot{\Delta x}_2^*} + \cancel{K_X \dot{\Delta x}_1^*} \\ &= K_Y \Delta Y^*\end{aligned}\quad (3-29)$$

From (3-29), the equilibrium equation for Δx_2^* is found, noting that:

$$\begin{aligned}\Delta x_2^* &= \Delta x_2^*(0) + \int \Delta \dot{x}_2^*(\tau) d\tau \\ &= \int \Delta \dot{x}_2^*(\tau) d\tau\end{aligned}\quad (3-30)$$

Thus

$$\Delta x_2^* = K_Y \int \Delta Y^* d\tau \quad (3-31)$$

A relationship for Δx_1^* is found by differentiating equation (3-18), such that:

$$\begin{aligned}\Delta \dot{x}_1^* &= S_{12} \Delta \dot{Y}^* - (S_{11} F_{12} + S_{12} H_{x_2}) \Delta \dot{x}_2^* + S_{11} \Delta \ddot{x}_1^* \\ &= -(S_{11} F_{12} + S_{12} H_{x_2}) \Delta \dot{x}_2^* + S_{11} \Delta \ddot{x}_1^*\end{aligned}\quad (3-32)$$

Differentiating again:

$$\begin{aligned}\Delta \ddot{x}_1^* &= -(S_{11} F_{12} + S_{12} H_{x_2}) \Delta \ddot{x}_2^* + S_{11} \Delta \ddot{\ddot{x}}_1^* \\ &= 0\end{aligned}\quad (3-33)$$

Thus, from (3-33), equation (3-32) becomes:

$$\begin{aligned}\Delta \ddot{x}_1^* &= -(S_{11} F_{12} + S_{12} H_{x_2}) \Delta \ddot{x}_2^* + S_{11} \Delta \ddot{\ddot{x}}_1^* \\ &= -(S_{11} F_{12} + S_{12} H_{x_2}) \Delta \ddot{x}_2^* \\ &= -(S_{11} F_{12} + S_{12} H_{x_2}) K_Y \Delta Y^*\end{aligned}\quad (3-34)$$

By substituting (3-34) into equation (3-18), a relationship for the nonsingular variables, $\Delta \underline{x}_1^*$, in terms of $\Delta \underline{y}^*$ and $\Delta \underline{x}_2^*$ results:

$$\begin{aligned}\Delta \underline{x}_1^* &= S_{12} \Delta \underline{y}^* - (S_{11} F_{12} + S_{12} H_{x_2}) \Delta \underline{x}_2^* - S_{11} \Delta \dot{\underline{x}}_1^* \\ &= S_{12} \Delta \underline{y}^* - (S_{11} F_{12} + S_{12} H_{x_2}) \Delta \underline{x}_2^* - \\ &\quad S_{11} (S_{11} F_{12} + S_{12} H_{x_2}) K_Y \Delta \underline{y}^* \\ &= (S_{12} - S_{11} (S_{11} F_{12} + S_{12} H_{x_2}) K_Y) \Delta \underline{y}^* - \\ &\quad (S_{11} F_{12} + S_{12} H_{x_2}) \Delta \underline{x}_2^* \quad (3-35)\end{aligned}$$

Finally, a relationship for the equilibrium controls is found by substituting (3-34) into (3-19) as follows:

$$\begin{aligned}\Delta \underline{u}^* &= S_{22} \Delta \underline{y}^* - (S_{21} F_{12} + S_{22} H_{x_2}) \Delta \underline{x}_2^* + S_{21} \Delta \dot{\underline{x}}_1^* \\ &= S_{22} \Delta \underline{y}^* - (S_{21} F_{12} + S_{22} H_{x_2}) \Delta \underline{x}_2^* - S_{21} (S_{11} F_{12} + S_{12} H_{x_2}) K_Y \Delta \underline{y}^* \\ &= (S_{22} - S_{21} (S_{11} F_{12} + S_{12} H_{x_2}) K_Y) \Delta \underline{y}^* - (S_{21} F_{12} + S_{22} H_{x_2}) \Delta \underline{x}_2^* \quad (3-36)\end{aligned}$$

Since this CAS is designed as a sampled-data controller, the corresponding sampled-data equilibrium equations are:

$$\Delta \underline{x}_{2k}^* = \Delta \underline{x}_{2k-1}^* + K_Y \Delta t \Delta \underline{y}_k^* \quad (3-37)$$

$$\Delta \underline{x}_{1k}^* = (S_{12} - S_{11} (S_{11} F_{12} + S_{12} H_{x_2}) K_Y) \Delta \underline{y}_k^* - (S_{11} F_{12} + S_{12} H_{x_2}) \Delta \underline{x}_{2k}^* \quad (3-38)$$

$$\Delta \underline{u}_k^* = (S_{22} - S_{21} (S_{11} F_{12} + S_{12} H_{x_2}) K_Y) \Delta \underline{y}_k^* - (S_{21} F_{12} + S_{22} H_{x_2}) \Delta \underline{x}_{2k}^* \quad (3-39)$$

Finally, equations (3-37) through (3-39) is simplified further by defining the following quantities:

$$S_{\lambda Y} = S_{12} - S_{11}(S_{11}F_{12} + S_{12}H_{x_2})K_Y \quad (3-40)$$

$$S_{\lambda 1} = -(S_{11}F_{12} + S_{12}H_{x_2}) \quad (3-41)$$

$$S_{UY} = S_{22} - S_{21}(S_{11}F_{12} + S_{12}H_{x_2})K_Y \quad (3-42)$$

$$S_{U1} = -(S_{21}F_{12} + S_{22}H_{x_2}) \quad (3-43)$$

Thus:

$$\Delta x_{2k}^* = \Delta x_{2k-1}^* + K_Y \Delta t \Delta y_k^* \quad (3-44)$$

$$\Delta x_{1k}^* = S_{\lambda Y} \Delta y_k^* + S_{\lambda 1} \Delta x_{2k}^* \quad (3-45)$$

$$\Delta u_k^* = S_{UY} \Delta y_k^* + S_{U1} \Delta x_{2k}^* \quad (3-46)$$

3.2 CALCULATION OF OPTIMAL GAINS

Using sampled-data, linear-quadratic control theory, the optimal gains, C , are,

$$C = (\hat{K} + \Gamma^T P_{SS})^{-1} (\Gamma^T F_{SS} \bar{Q} + \hat{M}^T) \quad (3-47)$$

where \bar{Q} and Γ are the sampled-data system equations, \hat{K} and \hat{M} are sampled-data weighting matrices and F_{SS} is the solution to the discrete Riccati equation. To solve for the optimal gains, then, the sampled-data system equations and weighting matrices and the solution to the discrete Riccati equation had to be found.

3.2.1 Sampled-Data System Equations

The continuous-time, system equations are of the following form:

$$\Delta \dot{\underline{x}} = F \Delta \underline{x} + G \Delta \underline{u} \quad (3-48)$$

By neglecting the control effects such that $\Delta \underline{u} = \underline{0}$ (this can be done by superposition), the equation becomes:

$$\Delta \dot{\underline{x}} = F \Delta \underline{x} \quad (3-49)$$

Solving for $\Delta \underline{x}$ (using the Laplace transform method), the equation reduced to:

$$\begin{aligned} (sI - F) \Delta \underline{x} &= \Delta \underline{x}(0) \\ \Delta \underline{x} &= (sI - F)^{-1} \Delta \underline{x}(0) \end{aligned} \quad (3-50)$$

or, in the time domain:

$$\Delta \underline{x} = e^{Ft} \Delta \underline{x}(0) \quad (3-51)$$

The equivalent recursive equation for propagating the state from one instant to the next is,

$$\Delta \underline{x}(t_1) = e^{F(t_1 - t_0)} \Delta \underline{x}(t_0) \quad (3-52)$$

The state transition matrix, Φ , can be defined as:

$$\Phi = e^{F(t_1 - t_0)} \quad (3-53)$$

or it can be defined over an interval, Δt , such that:

$$\Delta \underline{x}(t + \Delta t) = \Phi(\Delta t) \Delta \underline{x}(t) \quad (3-54)$$

The calculation of the state transition matrix involves the use of the series representation for e :

$$e^{a\Delta t} = 1 + a\Delta t + (a\Delta t)^2/2 + (a\Delta t)^3/3 + \dots \quad (3-55)$$

In matrix notation for F :

$$\Phi(\Delta t) = 1 + F\Delta t + 1/2 (F\Delta t)^2 + 1/3 (F\Delta t)^3 + \dots \quad (3-56)$$

In the case where $\Delta u \neq 0$:

$$\Delta x(t) = \Phi(\Delta t)\Delta x(t_0) + \int \Phi(\Delta t, \tau)G(\tau)\Delta u(\tau) d\tau \quad (3-57)$$

The control effects matrix, Γ , is defined as:

$$\begin{aligned} \Gamma &= \int \Phi(\Delta t, \tau)G(\tau) d\tau \\ &= \int e^{F(\Delta t - \tau)} d\tau G \\ &= \Phi(\Delta t) \int e^{-F\tau} d\tau G \end{aligned} \quad (3-58)$$

Solving $\int e^{-F\tau} d\tau$:

$$\begin{aligned} \int e^{-F\tau} d\tau &= \int (1 - F\tau + 1/2 (F\tau)^2 - 1/3 (F\tau)^3 + \dots) d\tau \\ &= \Delta t - 1/2 F\Delta t^2 + 1/6 F^2\Delta t^3 - 1/24 F^3\Delta t^4 + \dots \\ &= \Delta t(1 - 1/2 F\Delta t + 1/6 (F\Delta t)^2 - 1/24 (F\Delta t)^3 + \dots) \end{aligned} \quad (3-59)$$

Therefore, the control effects matrix is found as follows:

$$\Gamma = \Phi(\Delta t)\Delta t(1 - 1/2 F\Delta t + 1/6 (F\Delta t)^2 - 1/24 (F\Delta t)^3 + \dots) G \quad (3-60)$$

3.2.2 Sampled-Data State- and Control-Weighting Matrices

The sampled-data cost function, J , defined in (3-1) as:

$$J = 1/2 \sum_{k=0}^{\infty} \int_{t_k}^{t_{k+1}} (\Delta \underline{x}^T Q_C \Delta \underline{x} + \Delta \underline{u}^T R_C \Delta \underline{u}) dt$$

may also be defined as:

$$J = 1/2 \sum_{k=0}^{\infty} (\Delta \underline{x}_k^T \hat{Q} \Delta \underline{x}_k + \Delta \underline{x}_k^T \hat{M} \Delta \underline{u}_k + \Delta \underline{u}_k^T \hat{K} \Delta \underline{u}_k) \quad (3-61)$$

where \hat{Q} , \hat{M} , and \hat{K} are sampled-data, state- and control-weighting matrices and are defined in terms of the continuous-time weighting matrices as follows:

$$\hat{Q} = \int_0^{\Delta t} \tilde{\Phi}^T(\tau) Q_C \tilde{\Phi}(\tau) d\tau \quad (3-62)$$

$$\hat{M} = \int_0^{\Delta t} \tilde{\Phi}^T(\tau) Q_C \Gamma(\tau) d\tau \quad (3-63)$$

$$\hat{K} = \int_0^{\Delta t} (R_C + \Gamma^T(\tau) Q_C \Gamma(\tau)) d\tau \quad (3-64)$$

The integrals are solved using Simpson's rule, which is:

$$\int_0^{\Delta t} f(t) dt = ((k-\alpha)/(3n)) (f(t_0) + 4f(t_1) + 2f(t_2) + \dots + 2f(t_{n-2}) + 4f(t_{n-1}) + f(t_n)) \quad (3-65)$$

In this case, $(k-\alpha)$ is defined as the sampling time, Δt , and n is the number of subintervals in the sample (10).

The calculations are simplified noting that:

$$\tilde{\Phi}(\Delta t) \tilde{\Phi}(\Delta t) = \tilde{\Phi}(2\Delta t) \quad (3-66)$$

Thus, only one state transition matrix (for $t = \Delta t/10$) needed to be calculated instead of calculating one at every nt (Ref. 6).

3.2.3 Solution to the Discrete Riccati Equation

The discrete Riccati equation (Ref. 7) is as follows:

$$P_{k-1} = \Phi^T P_k \Phi + \hat{Q} - (\Gamma^T P_k \Phi + \hat{M}^T) (\hat{R} \Gamma^T P_k \Gamma)^{-1} (\Gamma^T P_k \Phi + \hat{M}^T) \quad (3-67)$$

The equation is iterated until a steady-state solution is reached, i.e.:

$$P_k = P_{k-1} = P_{ss}$$

3.2.4 Computation of the Closed-Loop System

Once the optimal gains are calculated, the equivalent closed-loop system dynamics equation, F_{cl} , is found and its stability characteristics investigated. First, the closed-loop state transition matrix is found by:

$$\Phi_{cl} = \Phi - \Gamma c \quad (3-68)$$

F_{cl} is then found using the series representation for the natural log:

$$\begin{aligned} a &= (1/\Delta t) \ln(e^{a\Delta t}) \\ &= (1/\Delta t) ((e^{a\Delta t} - 1) - (1/2)(e^{a\Delta t} - 1)^2 + (1/3)(e^{a\Delta t} - 1)^3 - \dots) \end{aligned} \quad (3-69)$$

Therefore,

$$\begin{aligned} F_{c1} &= (1/\Delta t) \ln(\Phi_{c1}) \\ &= (1/\Delta t)((\Phi - 1) - (1/2)(\Phi - 1)^2 + (1/3)(\Phi - 1)^3 - \dots) \end{aligned} \quad (3-70)$$

3.3 CALCULATION OF CONTROL GAINS

To summarize the results up to this point, the steady-state estimates $(\Delta \underline{x}_1^*, \Delta \underline{x}_2^*, \Delta \underline{u}^*)$ are developed, and those equations are:

$$\Delta \underline{x}_{2k}^* = \Delta \underline{x}_{2k-1}^* + K_Y \Delta t \Delta \underline{y}_k^* \quad (3-44)$$

$$\Delta \underline{x}_{1k}^* = SXY \Delta \underline{y}_k^* + S\lambda 1 \Delta \underline{x}_{2k}^* \quad (3-45)$$

$$\Delta \underline{u}_k^* = SUY \Delta \underline{y}_k^* + S\lambda 1 \Delta \underline{x}_{2k}^* \quad (3-46)$$

where all terms are previously defined. In addition, the optimal gains, C , are calculated. Therefore, by substituting into the control law:

$$\Delta \underline{u}_k = \Delta \underline{u}_k^* - C (\Delta \underline{x}_k - \Delta \underline{x}_k^*) \quad (3-5)$$

the control law becomes:

$$\begin{aligned} \Delta \underline{u}_k &= \Delta \underline{u}_k^* - C \Delta \underline{x}_k - C^* \begin{bmatrix} \Delta \underline{x}_{1k}^* \\ \Delta \underline{x}_{2k}^* \end{bmatrix} \\ &= \Delta \underline{u}_k^* - C \Delta \underline{x}_k - C_1 \Delta \underline{x}_{1k}^* - C_2 \Delta \underline{x}_{2k}^* \\ &= SUY \Delta \underline{y}_k^* + S\lambda 1 \Delta \underline{x}_{2k}^* - C \Delta \underline{x}_k - C_1 (SXY \Delta \underline{y}_k^* + S\lambda 1 \Delta \underline{x}_{2k}^*) - C_2 \Delta \underline{x}_{2k}^* \end{aligned}$$

$$= (S_{LY} - C_1 S_{\lambda Y}) \Delta y_k^* + (S_{L1} - C_1 S_{\lambda 1} - C_2) \Delta x_{2k}^* - C \Delta x_k \quad (3-71)$$

By defining:

$$C_f = S_{LY} - C_1 S_{\lambda Y} \quad (3-72)$$

$$C_s = S_{L1} - C_1 S_{\lambda 1} - C_2 \quad (3-73)$$

$$C_b = -C \quad (3-74)$$

the control law simplified to:

$$\Delta u_k = C_f \Delta y_k^* + C_s \Delta x_{2k}^* + C_b \Delta x_k \quad (3-75)$$

Since

$$\Delta x_{2k}^* = K_Y \int \Delta y_k^* dt \quad (3-44)$$

The control law is rewritten as:

$$\Delta u_k = C_f \Delta y_k^* + C_s K_Y \int \Delta y_k^* dt + C_b \Delta x_k \quad (3-76)$$

By defining:

$$C_i = C_s K_Y \quad (3-77)$$

the final control law is then written as:

$$\Delta u = C_f \Delta y_k^* + C_i \int \Delta y_k^* dt + C_b \Delta x_k \quad (3-78)$$

where Δy_k^* is the command, Δx_k is the current value of the state, and:

$$C_f = S_{22} - S_{21}(S_{11}F_{12} + S_{12}H_{x_2})K_Y -$$

$$C_1(S_{12} - S_{11}(S_{11}F_{12} + S_{12}H_{x_2})K_Y) \quad (3-79)$$

$$C_i = -(S_{21}F_{12} + S_{22}H_{x_2}) + C_1(S_{11}F_{12} + S_{12}H_{x_2})K_Y \quad (3-80)$$

$$C_b = -C \quad (3-81)$$

3.4 CLOSED-LOOP RESULTS

Once the control law design method was set, the selection of the continuous-time weighting matrices, Q_c and R_c , was necessary. Since these matrices affected response, desired response characteristics had to be determined before Q_c and R_c could be chosen.

After the controller was subsequently formulated, verification of its operation through simulation also needed to be carried out. Results of the verification were presented in two ways. First, a detailed description of a nominal flight condition was presented. Second, a summary of 27 different flight condition simulations was presented including closed-loop eigenvalues and response characteristics to two different commands.

3.4.1 Selection of Q_c and R_c

Before the sampled-data weighting matrices could be calculated, the continuous-time weighting matrices, Q_c and R_c , had to be selected. Since there was no method for determining the weights which give the desired step response, the selection was based on a trial-and-error iteration. Tradeoffs in response

characteristics were examined and the weights which gave the "best" result were used.

Three step response characteristics were considered in the selection--rise time, defined as the amount of time to go from 10 to 90 percent of final value; overshoot, the percentage over final value that the response reached at first peak; and settling time, the amount of time to settle to within one percent of final value. The test was run using a nominal flight condition (angle of attack = 10 deg, throttle = .15, velocity = 150 feet per second) and two commands (10 deg/sec roll rate with zero sideslip and zero roll rate with 2 deg of sideslip).

The desired response was selected for two types of commands. The first was a roll rate command (with zero sideslip) for which minimum rise time, overshoot, and settling time were wanted. The second command was a sideslip command (with zero roll rate) for which minimum overshoot and settling time and a rise time around 1 sec were desirable. The selection of Q_c and R_c were based on the results closest to these criteria.

Before starting the selection process, an initial Q_c and R_c were chosen. In particular, only variations in the Q_c elements corresponding to sideslip (Q_β) and roll angle (Q_ϕ) were found to be important in arriving at suitable responses. The other elements in the weighting matrices were set to the inverses of the maximum mean-square values of the states and controls (Ref. 5). Those mean values used are:

$$r = 10 \text{ deg/sec}$$

$$p = 10 \text{ deg/sec}$$

$$dR = 10 \text{ deg}$$

$$dA = 40 \text{ deg}$$

Using these values and rounding to one significant figure, the weighting matrices are:

$$Q_c = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & Q_E & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & Q_\phi \end{bmatrix} \quad R_c = \begin{bmatrix} 1.0 & 0 \\ 0 & .10 \end{bmatrix}$$

The Q_E and Q_ϕ terms then were varied to find the best choice of those weights. At first, each was set at five different values (1, 25, 50, 75, 100), such that 25 different combinations were tested. The results were listed in Table 2. The roll rate command seemed to be best when Q_ϕ was 25 but the sideslip command was too slow for $Q_E = 1$ and too fast for $Q_E = 25$. Another set of weights were tested using $Q_\phi = 25$ and $Q_E = (1, 5, 10, 15, 20, 25)$. These results were listed in Table 3. From these tests, $Q_E = 10$ appeared to be the best choice.

Therefore, the final continuous-time weighting matrices are:

$$Q_c = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 10.0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 25.0 \end{bmatrix} \quad R_c = \begin{bmatrix} 1.0 & 0 \\ 0 & .10 \end{bmatrix}$$

Table 2

Response Characteristics Varying Sideslip and Roll Angle Weightings

DUTCH ROLL ANGLE WEIGHTING		EIGENVALUES (Dutch Roll, Roll, Spiral)		$\Delta y = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$		$\Delta y = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$			
Q_ϕ	LOOP			RISE TIME (sec)	OVERSHOOT (%)	SETTLING TIME (sec)	RISE TIME (sec)	OVERSHOOT (%)	SETTLING TIME (sec)
OPEN	LOOP	-3.124 ± 2.420j -4.4127 .0513		.388	7.05	-	.51	3.55	-
1	1	-2.3535 ± .2625j -8.6816 -.9795		.174	6.97	2.70	1.47	-	3.20
1	25	-2.1375 ± .3124j -5.5666 ± 1.6542j		.152	14.65	1.15	1.48	-	3.30
1	50	-2.1206 ± .2923j -5.5782 ± 2.7347j		.149	16.07	1.00	1.48	-	3.35
1	75	-2.1080 ± .2845j -5.6270 ± 3.1518j		.147	16.81	.95	1.48	-	3.40
1	100	-2.0906 ± .2690j -5.6795 ± 3.3946j		.146	17.23	.90	1.49	-	3.40
25	1	-3.1665 ± 2.6614j -8.4946 -.9785		.174	7.11	2.70	.58	2.35	1.60
25	25	-3.2144 ± 2.5423j -5.2477 ± 1.5524j		.153	14.84	1.15	.58	2.15	1.60
25	50	-3.1569 ± 2.4779j -5.3575 ± 2.7104j		.150	16.02	1.00	.59	2.10	1.60
25	75	-3.1183 ± 2.4479j -5.4182 ± 3.1309j		.148	16.74	.95	.59	2.05	1.60
25	100	-3.0897 ± 2.4236j -5.4626 ± 3.3691j		.147	17.14	.95	.59	2.00	1.65
50	1	-3.3670 ± 3.0928j -8.4105 -.9786		.174	7.16	2.70	.50	3.20	1.45
50	25	-3.4425 ± 3.0281j -5.1833 ± 1.4623j		.153	14.94	1.15	.50	2.95	1.45

Table 2

continued

ROLL SLIP WEIGHTING Q_ϕ	ROLL ANGLE Q_ϕ	EIGENVALUES (Dutch Roll, Roll, Spiral)	$\Delta y = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$		$\Delta y = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$	
			RISE TIME (sec)	OVERSHOOT (%)	SETTLING TIME (sec)	OVERSHOOT (%)
50	50	-3.4308 ± 2.9406j	.150	16.07	1.00	.50
		-5.2466 ± 2.6403j				2.90
50	75	-3.3991 ± 2.8907j	.148	16.79	.95	.50
		-5.2982 ± 3.0745j				2.80
50	100	-3.3716 ± 2.8565j	.148	17.19	.95	.50
		-5.3387 ± 3.3168j				2.80
75	1	-3.4605 ± 3.3038j	.174	7.18	2.70	.48
		-8.3633 - .9786				3.55
75	25	-3.5342 ± 3.2687j	.153	15.00	1.10	.48
		-5.1648 ± 1.4176j				3.40
75	50	-3.5512 ± 3.1940j	.150	16.10	1.00	.48
		-5.1993 ± 2.5796j				3.35
75	75	-3.5361 ± 3.1370j	.148	16.83	.95	.49
		-5.2331 ± 3.0174j				3.30
75	100	-3.5156 ± 3.0963j	.148	17.23	.95	.49
		-5.2650 ± 3.2633j				3.25
100	1	-3.5158 ± 3.4307j	.174	7.19	2.70	.48
		-8.3322 - .9786				3.85
100	25	-3.5837 ± 3.4122j	.153	15.04	1.10	.48
		-5.1570 ± 1.3942j				3.70
100	50	-3.6127 ± 3.3529j	.150	16.11	1.00	.48
		-5.1793 ± 2.5373j				3.60
100	75	-3.6116 ± 3.2987j	.149	16.85	.95	.48
		-5.1985 ± 2.9700j				3.55
100	100	-3.5996 ± 3.2561j	.148	17.25	.95	.48
		-5.2210 ± 3.2160j				3.50

Table 3
Response Characteristics Varying Sideslip Weighting

$\Delta y = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$
 $\Delta y = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$

Sideslip Weighting β	Roll Angle Weighting ϕ	EIGENVALUES (Dutch Roll, Roll-Spiral)	Rise Time (sec)	Over Shoot (%)	Settling Time (sec)	Rise Time (sec)	Over Shoot (%)	Settling Time (sec)
1	25	-2.1375 ± .3124j -5.5066 ± 1.6542j	.152	14.65	1.15	1.48	-	3.30
5	25	-2.5555 ± 1.4279j -5.4384 ± 1.6474j	.152	14.68	1.15	.937	.4	1.10
10	25	-2.8270 ± 1.8860j -5.3694 ± 1.6289j	.152	14.73	1.15	.871	1.05	1.70
15	25	-3.0012 ± 2.1701j -5.3163 ± 1.6043j	.153	14.77	1.15	.666	1.05	1.70
20	25	-3.1240 ± 2.3786j -5.2768 ± 1.5778j	.153	14.81	1.15	.596	1.85	1.65
25	25	-3.2144 ± 2.5423j -5.2477 ± 1.5524j	.153	14.84	1.15	.580	2.15	1.60

3.4.2 Closed-Loop Simulation: Nominal Flight Condition

Verification of the controller was done by simulations of command responses. The nominal flight condition was used for the simulations; thus, at the nominal flight condition, the linearized system dynamic and output equations were determined as follows:

$$\begin{bmatrix} \Delta r \\ \Delta b \\ \Delta p \\ \Delta \phi \end{bmatrix} = \begin{bmatrix} -.588 & 4.268 & -.506 & 0.0 \\ -.999 & -.224 & 0.000 & 0.214 \\ 2.753 & -9.167 & -4.374 & 0.0 \\ 0.0 & 0.0 & 1.000 & 0.0 \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta b \\ \Delta p \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} -5.551 & 0.545 \\ 0.070 & -.000 \\ 1.113 & -8.017 \\ 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} \Delta \delta R \\ \Delta \delta A \end{bmatrix}$$

$$\begin{bmatrix} \Delta p_c \\ \Delta E_c \end{bmatrix} = \begin{bmatrix} 0.0 & 0.0 & 1.000 & 0.0 \\ 0.0 & 1.000 & 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta b \\ \Delta p \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} \Delta \delta R \\ \Delta \delta A \end{bmatrix}$$

where Δp_c and ΔE_c were roll rate and sideslip commands, respectively. The characteristic equation was found as follows:

$$C = (s + .4124 - 2.4206j)(s + .4124 + 2.4206j)(s + 4.4127)(s - .0513)$$

Thus, from the characteristic equation, a spiral instability with a long time constant (19.5 seconds) was noted.

The continuous-time system dynamic equations were converted to the following sampled-data system equations:

$$\begin{bmatrix} \Delta r_{k+1} \\ \Delta B_{k+1} \\ \Delta P_{k+1} \\ \Delta \phi_{k+1} \end{bmatrix} = \begin{bmatrix} 0.916 & 0.425 & -.039 & 0.005 \\ -.095 & 0.956 & 0.004 & 0.021 \\ 0.252 & -.677 & 0.640 & -.008 \\ 0.013 & -.038 & 0.081 & 1.000 \end{bmatrix} \begin{bmatrix} \Delta r_k \\ \Delta B_k \\ \Delta P_k \\ \Delta \phi_k \end{bmatrix} + \begin{bmatrix} -.535 & 0.069 \\ 0.034 & -.004 \\ 0.015 & -.640 \\ 0.002 & -.035 \end{bmatrix} \begin{bmatrix} \Delta d k_k \\ \Delta d h_k \end{bmatrix}$$

The sampled-data weighting matrices were found using the sampled-data system equations and the continuous-time weighting matrices. The sampled-data weighting matrices that were calculated were summarized below:

$$\hat{Q} = \begin{bmatrix} 0.140 & -.142 & 0.033 & 0.157 \\ -.142 & 0.949 & -.073 & -.293 \\ 0.033 & -.073 & 0.074 & 0.255 \\ 0.157 & -.293 & 0.255 & 2.461 \end{bmatrix}$$

$$\hat{R} = \begin{bmatrix} -.128 & -.084 \\ 0.050 & 0.167 \\ -.008 & -.088 \\ -.070 & -.304 \end{bmatrix} \quad \hat{K} = \begin{bmatrix} 0.312 & 0.037 \\ 0.037 & 0.226 \end{bmatrix}$$

The optimal gains and, subsequently, the control gains were calculated. The control gains were as follows:

$$C_b = \begin{bmatrix} 0.724 & -.820 & -.016 & 0.084 \\ 0.288 & -.073 & 0.001 & 2.908 \end{bmatrix}$$

$$C_f = \begin{bmatrix} -0.167 & 1.027 \\ -1.172 & -.386 \end{bmatrix} \quad C_i = \begin{bmatrix} -0.253 & 0.0 \\ -2.898 & 0.0 \end{bmatrix}$$

The equivalent closed-loop system characteristic equation was found from the optimal gains and the sampled-data system equations; it is:

$$C = (s+2.827-1.886j)(s+2.827+1.886j)(s+5.369-1.629j)(s+5.369+1.629j)$$

As can be seen, the closed-loop system had no instabilities and fairly quick time constants. Also, the roll and spiral modes were no longer separate. Figure 2 shows a plot of the open- and closed-loop eigenvalues for comparison.

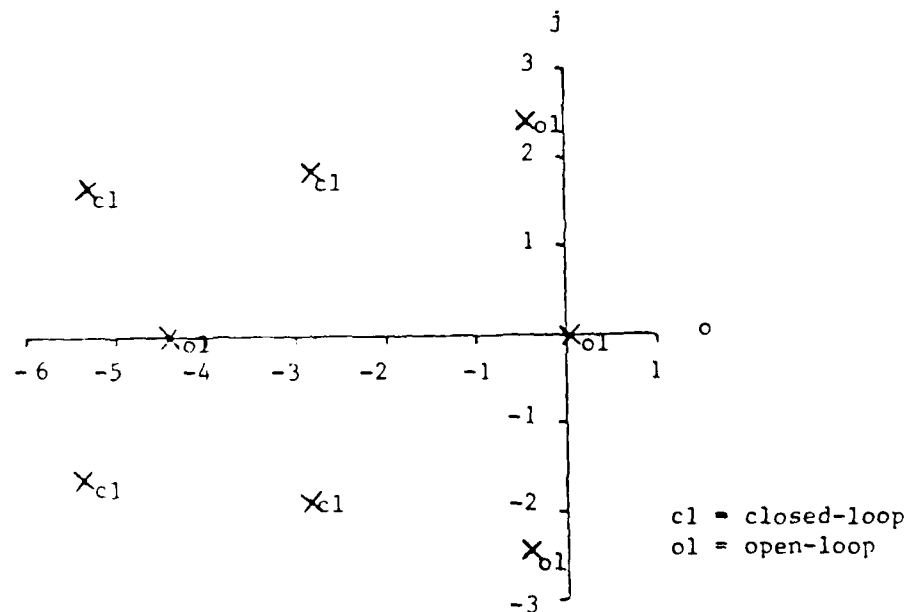


Figure 2: Eigenvalue Plot: Open-Loop vs. Closed-Loop

Finally, simulations were run to insure the proper operation of the control system. Linear simulations using state transition matrices were used for ease of comparison with later simulations.

Figure 3 shows the response of the aircraft for a roll rate command of 10 degrees per second. The rise time for the roll rate response is 0.152 seconds while the settling time was 1.15 seconds. Overshoot was 14.73 percent over final value. The sideslip experienced some steady-state error even though commanded to be zero but that error was negligible (0.006 degrees). The yaw rate response demonstrated that when command equilibrium was reached, even nonsingular variables do not necessarily reach equilibrium.

Figure 4 shows the system response for a sideslip command of 2 degrees. Rise time for the sideslip response was 0.751 seconds while settling time is 1.70 seconds. Overshoot was limited to only 1.05 percent over final value. All variables reached some steady-state value, as opposed to the roll rate command response, since the sideslip command was not singular.

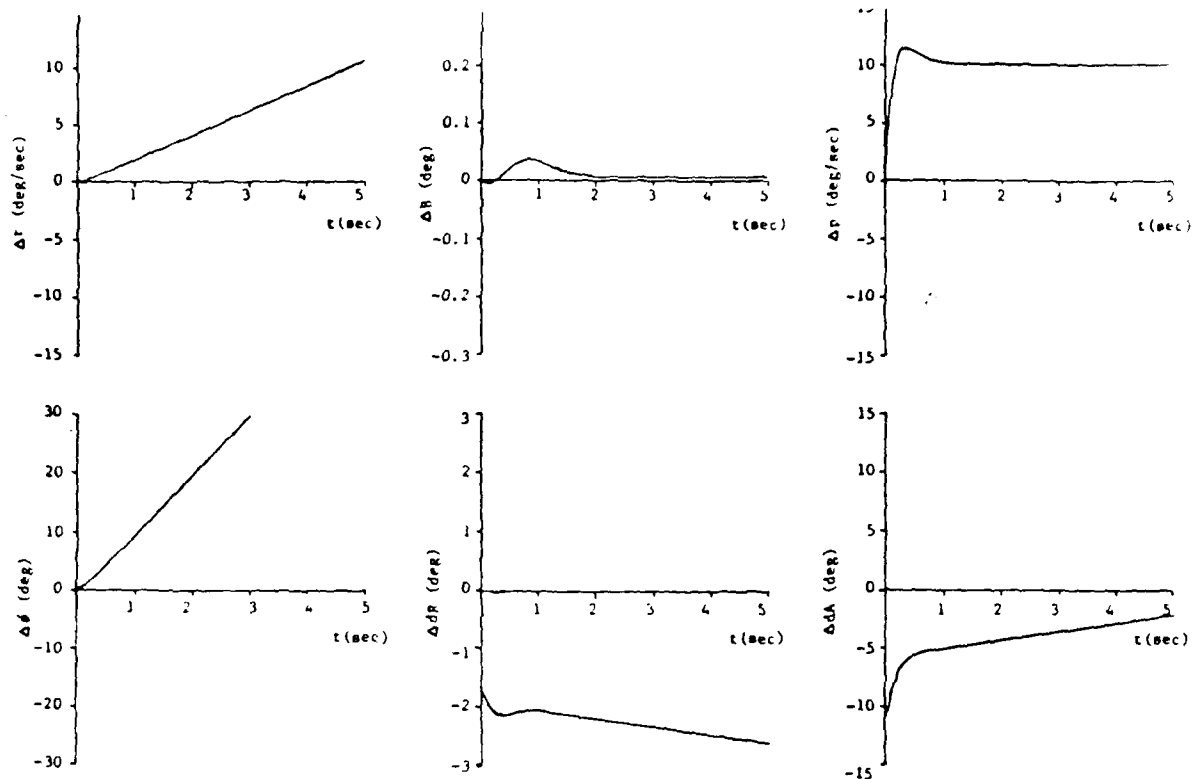


Figure 3: Closed-Loop Simulation - Roll Rate Command

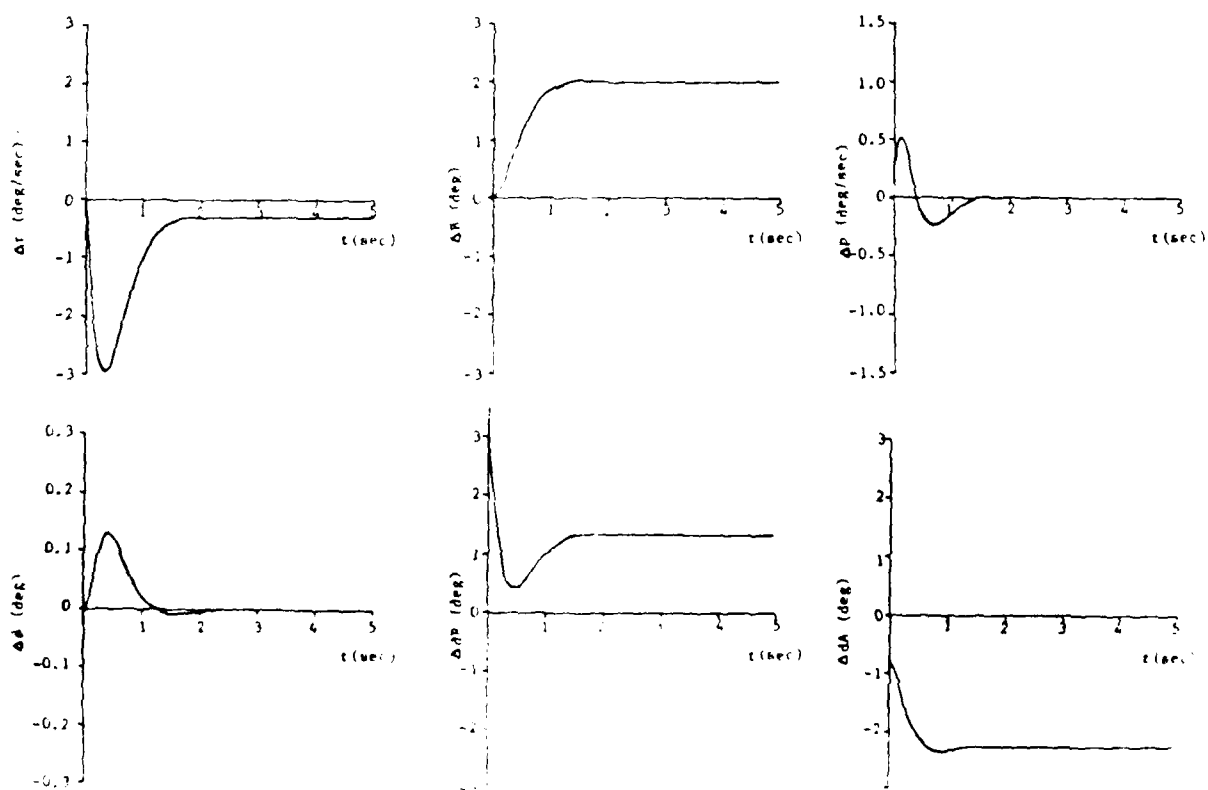


Figure 4. Closed-Loop Simulation - Sideslip Command

3.4.3 Summary of Closed-Loop Results

In addition to the nominal flight condition, 26 other cases were run using three values of each flight condition variable. The results were listed in Table 4. Listed were the closed-loop eigenvalues and the response characteristics for two command vectors.

The two commands were chosen for typical values which may be encountered in actual flight testing. In particular, a roll rate of 10 deg/sec with zero sideslip and a sideslip of 2 deg with zero roll rate were chosen.

The response characteristics of interest are: rise time, the time to go from 10 to 90 percent of final value; overshoot, the percentage over final value at the first peak; settling time, the time required to settle to within 1 percent of final value; steady-state error in roll rate, as a percentage of final value for a roll rate command and as an absolute number for a sideslip command; and steady-state error in sideslip, as an absolute number for a roll rate command and as a percentage of final value for a sideslip command.

Table 4

Closed-Loop Results

FLIGHT CONDITION	EIGENVALUES			RESPONSE $\Delta y = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$			RESPONSE $y = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$		
	α (DEG)	T_c (sec)	\ddot{q} (lb/ft ²)	Dutch Roll Roll-Spiral	Rise Time (sec)	Overshoot (%)	Settling Time (sec)	Steady State Error P (deg)	Steady State Error B (deg/sec)
-4 .03 9.731	-1.4797 ± 1.7081j	.161	15.78	1.20	0	.011	.823	7.50	2.75
-4 .03 21.894	-4.5205 ± 1.9247j -2.2968 ± 2.2659j	.152	14.23	1.10	0	.006	.663	4.15	2.05
-4 .03 38.922	-5.4832 ± 1.1206j -3.0265 ± 2.4610j	.148	13.66	1.05	0	.004	.597	2.05	1.65
-4 .13 9.731	-7.9104 ± 4.5891j -1.7252 ± 1.8320j	.162	15.85	1.20	0	.009	.822	5.20	2.60
-4 .13 21.894	-4.4837 ± 1.9563j -2.5920 ± 2.1860j	.153	14.36	1.10	0	.005	.679	2.40	1.95
-4 .13 38.922	-5.4084 ± 1.2512j -3.3053 ± 2.2722j	.148	13.77	1.05	0	.004	.662	.95	.90
-4 .23 9.731	-7.6598 ± 4.6443j -1.9354 ± 1.8470j	.162	15.91	1.20	0	.008	.823	3.75	2.45
-4 .23 21.894	-4.4479 ± 1.9877j -2.8180 ± 2.0989j	.153	14.48	1.10	0	.005	.685	1.45	1.80
-4 .23 38.922	-5.3324 ± 1.3711j -3.5194 ± 2.1048j	.148	13.85	1.05	0	.004	.673	.45	.80
10 .03 9.731	-7.3650 ± 4.6839j -1.7604 ± 1.6023j	.161	16.15	1.20	.04	.012	.915	3.40	2.75
10 .03 21.894	-4.4552 ± 2.0454j -2.5788 ± 1.8842j	.152	14.56	1.15	.04	.007	.760	1.50	2.00
10 .03 38.922	-5.4218 ± 1.4695j -3.1336 ± 1.9070j	.147	13.99	1.05	.04	.005	.687	.70	.85
10 .13 9.731	-7.5308 ± 5.1357j -2.0346 ± 1.7041j	.161	16.27	1.20	.05	.010	.844	2.55	2.45
10 .13 21.894	-4.4126 ± 2.0939j -2.8270 ± 1.8860j	.152	14.73	1.15	.05	.006	.751	1.05	1.70
10 .13 38.922	-5.3694 ± 1.6289j								

Table 4
continued

LIGHT CONDITION	EIGENVALUES	RESPONSE $\Delta y = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$		RESPONSE $y = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$				
		Rise Time (sec)	Overshoot (%)	Settling Time (sec)	Steady State Error B (deg)	Rise Time (sec)	Overshoot (%)	Settling Time (sec)
α (Hz)	T_c (lb/ft ²)	Dutch Roll Roll-Spiral						
10	.13 38.922	-3.2743 ± 1.8810j	.148	14.06	1.05	.05	.004	.681
		-7.0280 ± 5.5406j					.55	.85
10	.23 9.731	-2.2582 ± 1.7584j	.162	16.38	1.20	.05	.009	.835
		-4.3712 ± 2.1450j					1.95	2.25
10	.23 21.894	-2.9833 ± 1.8696j	.153	14.89	1.10	.05	.005	.694
		-5.3235 ± 1.7775j					.80	.85
10	.23 38.922	-3.3465 ± 1.8689j	.148	14.14	1.10	.05	.004	.677
		-6.2365 ± .5758j					.50	.80
24	.03 9.731	-1.2879 ± 1.2096j	.177	17.94	1.35	.06	.025	1.25
		-3.5074 ± 2.3667j					3.50	3.70
24	.03 21.894	-1.9609 ± 1.3669j	.168	16.70	1.30	.06	.007	1.03
		-3.9132 ± 2.6147j					1.10	2.45
24	.03 38.922	-2.3413 ± .8606j	.160	15.65	1.50	.06	-.001	1.17
		-4.6852 ± 3.0252j					-	1.35
24	.13 9.731	-1.4211 ± 1.2856j	.178	17.97	1.35	.07	.020	1.17
		-3.4911 ± 2.3133j					3.10	3.40
24	.13 21.894	-2.0789 ± 1.4265j	.170	16.67	1.40	.07	.003	.958
		-3.9289 ± 2.5560j					1.00	1.15
24	.13 38.922	-2.3614 ± 1.0300j	.162	15.17	1.55	.07	-.005	1.15
		-4.7409 ± 2.8196j					.05	1.30
24	.23 9.731	-1.5393 ± 1.3444j	.179	17.99	1.40	.08	.017	1.10
		-3.4526 ± 2.2971j					2.75	3.20
24	.23 21.894	-2.1530 ± 1.4844j	.172	16.66	1.45	.08	.001	.949
		-3.9561 ± 2.4773j					.95	1.15
24	.23 38.922	-2.3306 ± 1.2170j	.165	14.66	1.60	.08	-.008	1.07
		-4.7904 ± 2.5881j					.20	1.20

Chapter IV

GAIN SCHEDULING

Up to this point, the control system consisted of 27 different flight conditions and hence 27 different sets of gains. The problem was to find a scheme to schedule the gains such that the microprocessor would have the correct set of gains for the particular flight condition. The limitations of the microprocessor and the requirements of the CAS each had an effect on the eventual gain scheduling solution.

As mentioned above, the microprocessor limitations affected the form of the gain schedules. In particular, the memory space afforded the CAS program was limited to about 26K bytes. In addition, the microprocessor could only do a limited number of calculations during any interval of time. It was desirable to continuously update the gains to account for changes in the flight condition, but all calculations would have to be done within the sampling time, 0.1 seconds. Hence, the gain schedule had to be as small as possible without sacrificing accuracy, which the CAS required for proper operation.

Two methods could have been used for gain scheduling. The first was a table lookup method where the computer senses the flight condition and looks up the appropriate gains. In order

for this scheme to work effectively, there would have had to be a large table covering the variety of flight conditions which the aircraft might encounter. Such a method was not possible, since it required large memory space not available in the microprocessor. Therefore, this method of scheduling was rejected.

The second method involved calculating the gains as functions of the flight condition. This method included investigation into the sensitivities of the gains to changes in the flight condition and selection of a suitable solution form to match those sensitivities. By using the same solution form for as many gains as possible, the coefficients could be put into a set of matrices to simplify calculations and reduce the computation time. This method, too, had its drawbacks, in that if the gains were to be updated every sampling time, a scheme had to be developed to do all calculations within the sampling time. However, the drawbacks in this method did not seem as serious as those of the other method; therefore, this method was selected.

This chapter covers the formulation of the gain equations. In particular the flight condition functions, which are used in the gain equations, are discussed, as is the selection of the solution forms for the flight condition functions. In addition, the gain coefficient matrix computation method is covered. Finally, results are included showing a simulation for a nominal flight condition using the gain schedules.

4.1 FLIGHT CONDITION FUNCTIONS

At each flight condition, a different set of gains was required to provide satisfactory control of the aircraft. Thus, the gains were functions of the flight condition variables and could be represented as follows:

$$C = f(a) f(T_c) f(\bar{q}) \quad (4-1)$$

The gain scheduling task was to find suitable flight condition functions-- $f(a)$, $f(T_c)$, $f(\bar{q})$ --whose solutions were accurate compared to the actual gains at any condition.

The flight condition functions were required to reflect the sensitivities of the gains to each flight condition variable. By investigating these sensitivities, it was possible to narrow down the different solution forms. This was done by plotting the gains versus one flight condition variable, holding the other flight condition variables constant.

It was desirable to find solution forms which could be used to schedule more than a single gain. By doing this, implementation of the schedules in the CAS was simpler and matrix manipulation could be done with less memory space than otherwise. Therefore, selection of flight condition functions was limited to those which could be used for several gains.

The data used for gain scheduling included the 27 sets of gains and flight conditions reflecting all possible combinations

of the three values for each flight condition. The results are presented in Table 5. Since only three values of each flight condition variable were used, the polynomial functions,

$$f(a) = B_0 + B_1 a^1 + B_2 a^2 \quad (4-2)$$

$$f(T_c) = C_0 + C_1 T_c^1 + C_2 T_c^2 \quad (4-3)$$

$$f(\bar{q}) = D_0 + D_1 \bar{q}^1 + D_2 \bar{q}^2 \quad (4-4)$$

scheduled the gains exactly. (Any three points can be described by a second-order function). When these flight condition functions were multiplied in (4-1) and put into matrix form, a gain coefficient matrix of 14 by 27 resulted. Though this was not necessarily too large for use in the microprocessor, it did require a lot of computation time and a lot of memory space that could have been used more effectively by the CAS. It was felt that it would be better to reduce computation time and accept some error in the gains than to risk not being able to do the required computations within the sampling interval. Nevertheless, the exact solution did give a starting point for comparison.

In addition, a correlation factor was computed for the gain schedules to determine how well the gain schedule approximated the actual gains. The factor was computed as follows:

$$\text{CORRELATION} = 1 - ((C_{\text{actual}} - C_{\text{scheduled}}) / C_{\text{actual}})^2 \quad (4-5)$$

It was found that a correlation below 0.8 resulted in simulations which did not reach command equilibrium, while correlations above

that figure gave good results. Using the second-order solutions for all flight condition functions (as discussed above) resulted in a correlation of .9996 -- the error due to roundoff.

To facilitate computation of the gain equations, the gains were numbered as follows:

$$C_b = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \\ c_5 & c_6 & c_7 & c_8 \end{bmatrix} \quad C_f = \begin{bmatrix} c_9 & c_{10} \\ c_{11} & c_{12} \end{bmatrix} \quad C_i = \begin{bmatrix} c_{13} & c_{14} \\ c_{15} & c_{16} \end{bmatrix}$$

Table 2

[illegible]

Year	10,000	10,000	10,000	24,000	24,000	24,000	24,000	24,000	24,000	24,000
0.230	0.230	0.230	0.030	0.030	0.130	0.130	0.130	0.250	0.230	0.230
9.731	21.694	38.922	9.731	21.694	38.922	9.731	21.694	38.922	9.731	21.694

1.371	0.660	0.587	1.275	0.781	0.498	1.183	0.695	0.425	1.435	0.883	0.561	1.250	0.724	0.435
-1.245	-0.662	-0.131	-1.265	-0.616	-0.095	-1.245	-0.557	-0.064	-1.685	-1.070	-0.536	-1.527	-0.626	-0.527
-0.071	-0.055	-0.050	-0.080	-0.057	-0.049	-0.086	-0.057	-0.048	0.026	-0.012	-0.024	0.015	-0.016	-0.023
-0.526	-0.450	-0.434	-0.544	-0.434	-0.399	-0.551	-0.414	-0.367	0.222	0.131	0.073	0.154	0.084	0.048
0.424	0.324	0.238	0.640	0.442	0.307	0.821	0.534	0.358	0.342	0.270	0.198	0.415	0.288	0.203
-1.572	-1.459	-1.277	-1.884	-1.636	-1.372	-2.177	-1.793	-1.458	-0.601	-0.776	-0.785	-0.565	-0.673	-0.670
1.328	0.593	0.324	1.320	0.586	0.322	1.311	0.580	0.318	1.316	0.597	0.359	1.324	0.601	0.541
5.252	2.962	1.971	5.097	2.850	1.897	4.946	2.746	1.624	5.136	2.946	2.029	5.088	2.508	1.995
-0.017	0.065	0.077	0.003	0.061	0.068	0.017	0.058	0.061	-0.470	-0.223	-0.137	-0.357	-0.167	-0.103
2.530	2.342	1.795	2.668	2.006	1.463	2.461	1.753	1.232	2.541	1.916	1.364	2.347	1.627	1.105
-2.520	-1.236	-0.802	-2.293	-1.210	-0.783	-2.263	-1.186	-0.764	-2.212	-1.188	-0.780	-2.192	-1.172	-0.700
0.613	0.506	0.323	0.590	0.745	0.474	1.304	0.517	0.569	-0.552	-0.371	-0.363	-0.496	-0.386	-0.380
0.023	0.234	0.324	0.085	0.245	0.307	0.130	0.247	0.288	-0.728	-0.340	-0.175	-0.585	-0.253	-0.127
5.421	-3.046	-2.017	-5.337	-2.960	-1.955	-5.244	-2.876	-1.891	-5.100	-2.938	-2.024	-5.061	-2.198	-1.991

1.107	0.609	0.358	1.917	1.260	0.651	1.787	1.146	0.735	1.676	1.641	0.0
-1.366	-0.633	-0.165	-2.345	-1.670	-1.194	-2.244	-1.702	-1.029	-2.145	-1.552	-0.6
0.607	-0.018	-0.021	0.207	0.014	-0.066	0.168	0.005	-0.081	0.174	0.003	-0.0
0.111	0.058	0.036	-0.614	-0.972	-1.177	-0.587	-0.914	-1.054	-0.562	-0.846	-0.9
0.464	0.299	0.206	0.281	0.208	0.175	0.370	0.247	0.205	0.443	0.277	0.2
-0.495	-0.563	-0.567	-1.820	-1.930	-2.083	-1.355	-1.458	-1.625	-0.876	-0.981	-1.1
1.332	0.605	0.344	1.785	0.685	0.589	1.613	0.699	0.594	1.636	0.910	0.5
5.038	2.870	1.966	4.510	2.550	1.960	4.425	2.480	1.870	4.344	2.404	1.7
-0.288	-0.133	-0.083	-0.190	0.110	0.210	-0.171	0.101	0.186	-0.157	0.090	0.1
2.165	1.406	0.931	2.907	2.429	1.740	2.797	2.249	1.554	2.695	2.090	1.4
-2.171	-1.157	-0.756	-1.918	-1.005	-0.686	-1.899	-0.980	-0.661	-1.877	-0.554	-0.0
-0.476	-0.410	-0.411	-1.240	-1.127	-0.972	-1.196	-1.092	-0.922	-1.167	-1.067	-0.8
-0.492	-0.200	-0.100	-0.139	0.642	1.006	-0.096	0.621	0.908	-0.068	0.583	0.8
5.012	-2.856	-1.945	-4.311	-2.466	-1.516	-4.235	-2.397	-1.827	-4.161	-2.319	-1.7

4.2 SELECTION OF SOLUTION FORMS

The plots of the gain sensitivities are shown in Figures 5 through 7. The plots were made by holding two flight condition variables constant and plotting the variation of the gains versus variations in the third flight condition variable. Figure 5 shows the angle of attack sensitivities of the gains. In general, the gains followed no particular pattern; hence, it seemed simplest to use the second-order solution discussed previously.

Figure 6 shows the gain variations with respect to throttle setting. All gains appeared to be linear in throttle (due to the linear assumption used in the model formulation). Thus, an appropriate solution was a first-order equation:

$$f(T_c) = C_0 + C_1 T_c \quad (4-6)$$

In addition, a lot of the gains sloped towards zero as throttle setting increased (implying proportionality to the inverse of T_c). Since several gains showed this characteristic, another possible solution form was:

$$f(T_c) = e^{-C_c T_c} \quad (4-7)$$

Indeed, this turned out to be the case for half the gains. (In those gains where this form was appropriate, the constant term, C_c , had to be the same for every gain. It was found that $C_c = 0.001$ was satisfactory).

Figure 7 shows the gain sensitivities to changes in dynamic pressure. Almost all the gains behaved similarly. In particular, the gains curve toward zero as dynamic pressure increased. Such a form suggested a solution of:

$$f(\bar{q}) = D_0/\bar{q} \quad (4-8)$$

or

$$f(\bar{q}) = e^{-L_0 \bar{q}} \quad (4-9)$$

Both (4-8) and (4-9) required constants which had to be the same for at least several gains or the solution form was not desirable. However, constants could not be found in either case which would schedule any more than a few gains. Hence, the solution form used for the dynamic pressure equation was the second-order solution previously discussed.

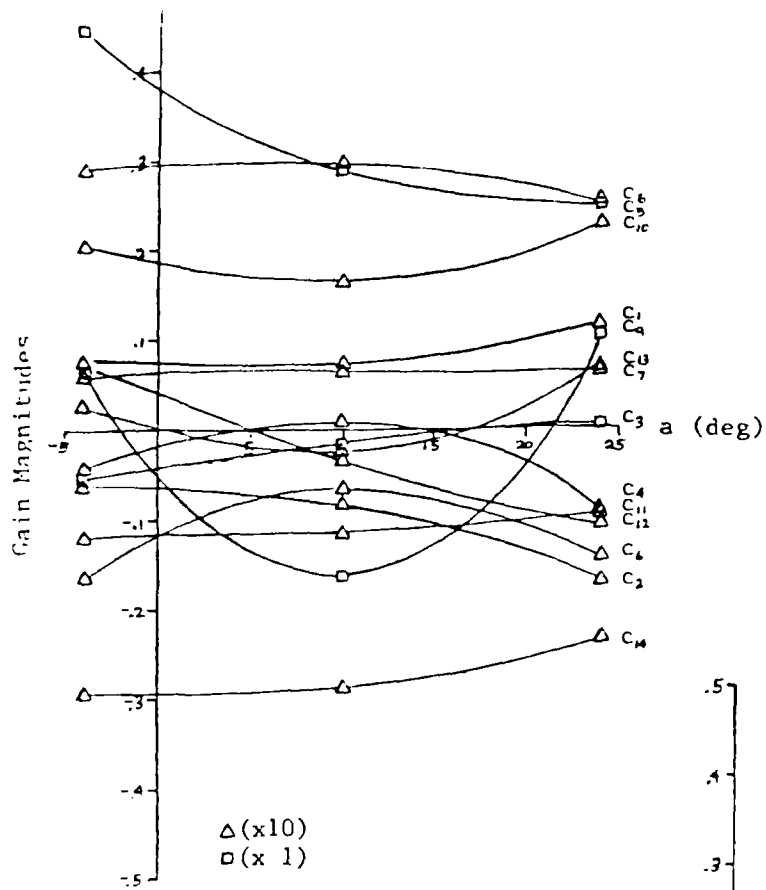


Figure 5: Gain Sensitivities to Changes in Angle-of-Attack

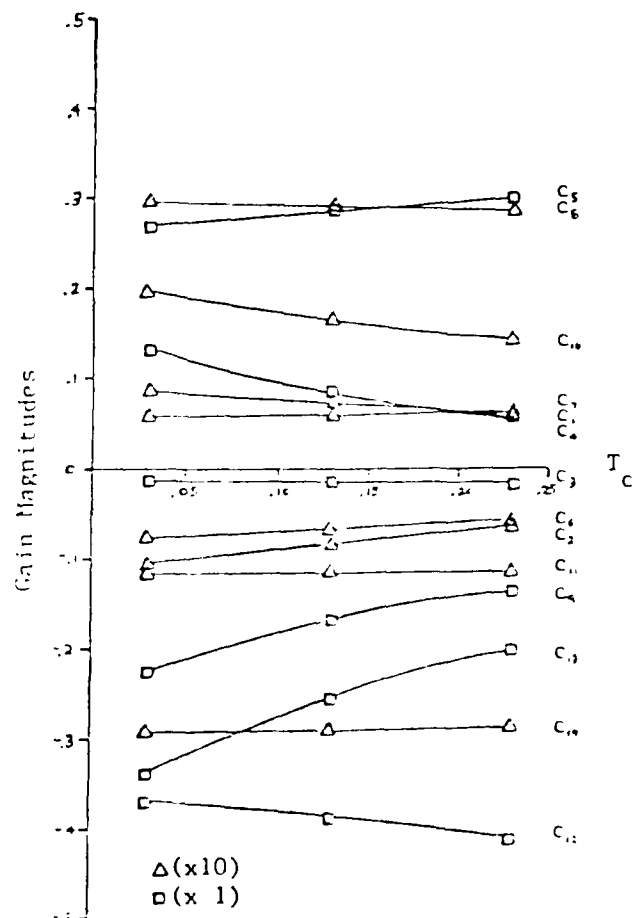


Figure 6: Gain Sensitivities to Changes in Throttle Setting

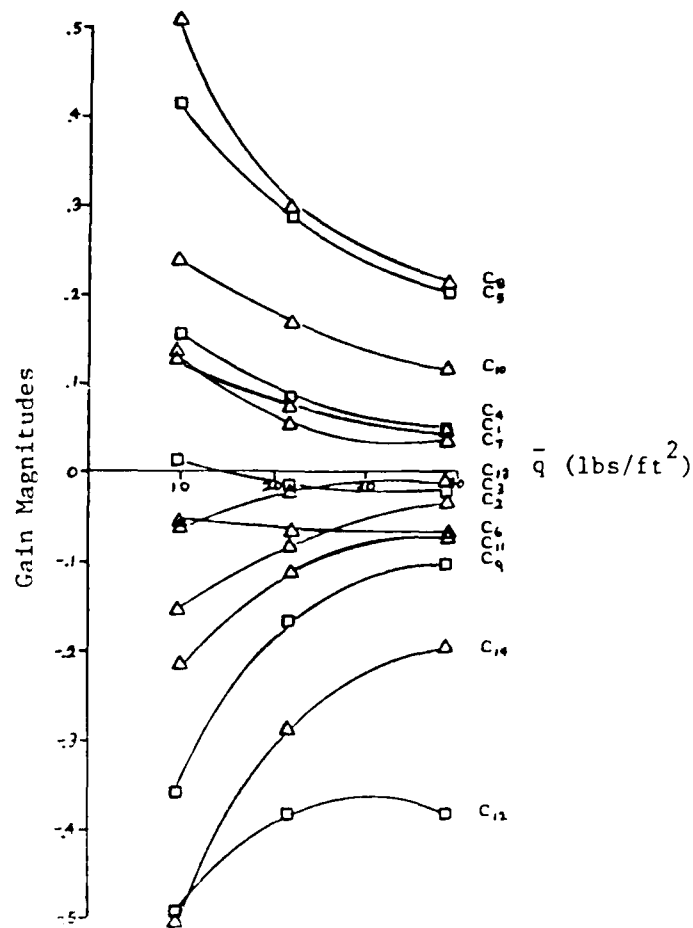


Figure 7: Gain Sensitivities to Changes in Dynamic Pressure

Thus the two solution forms for the throttle function resulted in two gain equations -- each solution form was used for half the gains. The forms are as follows:

1. For gains 1, 5, 7, 8, 9, 11, and 12:

$$C = (E_C + B_1 a + B_2 a^2) e^{-C T_C} (D_0 + D_1 \bar{q} + D_2 \bar{q}^2) \quad (4-10)$$

2. For gains 2, 3, 4, 6, 10, 13, and 14:

$$C = (E_C + B_1 a + B_2 a^2) (C_C + C_1 T_C) (D_0 + D_1 \bar{q} + D_2 \bar{q}^2) \quad (4-11)$$

When the equations were multiplied out, they took the form:

$$\begin{aligned}
 C = & A_0 e^{C_0 T_c} + A_1 e^{C_0 T_c} \bar{q}_T + A_2 e^{C_0 T_c} \bar{q}^2 + \\
 & A_3 e^{C_0 T_c} a + A_4 e^{C_0 T_c} \bar{q} a + A_5 e^{C_0 T_c} \bar{q}^2 a + \\
 & A_6 e^{C_0 T_c} a^2 + A_7 e^{C_0 T_c} \bar{q} a^2 + A_8 e^{C_0 T_c} \bar{q}^2 a^2 \quad (4-12)
 \end{aligned}$$

and

$$\begin{aligned}
 C = & A_0 + A_1 T_c + A_2 \bar{q} + \\
 & A_3 T_c \bar{q} + A_4 \bar{q}^2 + A_5 T_c \bar{q} + \\
 & A_6 a + A_7 T_c a + A_8 \bar{q} a + \\
 & A_9 T_c \bar{q} a + A_{10} \bar{q}^2 a + A_{11} T_c \bar{q} a + \\
 & A_{12} a + A_{13} T_c a^2 + A_{14} \bar{q} a + \\
 & A_{15} T_c \bar{q} a^2 + A_{16} \bar{q}^2 a^2 + A_{17} T_c \bar{q}^2 a^2 \quad (4-13)
 \end{aligned}$$

In this form, the equations could be written as a matrix equation:

$$\underline{C} = [A] \underline{fc} \quad (4-14)$$

where \underline{C} is the gain matrix (14 x 7 in either case), A contains the coefficients of the gain equations (14 x 9 for equation (4-12) and 14 x 18 for equation (4-13), and \underline{fc} is the flight condition vector consisting of all the combinations of the flight condition variables (9 x 1 for equation (4-12) and 18 x 1 for equation (4-13)).

4.3 COMPUTATION OF THE COEFFICIENT MATRICES

Once the flight condition functions were determined, the coefficients of the gain equations, $[A]$, could be found. To simplify the calculations, all the gains with the same form of gain equation were assembled into a matrix equation; thus only two matrix manipulations needed to be accomplished -- one for each gain equation. The matrices were set up as follows:

$$[\underline{C}_{fc_1} \quad \underline{C}_{fc_2} \quad \dots \quad \underline{C}_{fc_{27}}] = [A][\underline{fc}_1 \quad \underline{fc}_2 \quad \dots \quad \underline{fc}_{27}] \quad (4-15)$$

or more simply

$$[C] = [A][FC] \quad (4-16)$$

where the column vectors of C are the gains at one particular flight condition, the column vectors of FC are the flight condition combinations for one particular flight condition, and A contains the coefficients of the equations. Since FC was not necessarily square, the solution of A involved a "pseudo-inverse" of FC (Ref. 8). The pseudo-inverse was defined as follows:

$$FC^\# = FC^T (FC FC^T)^{-1} \quad (4-17)$$

such that:

$$[A] = [C][fc]^\#$$

The final solution of the two coefficient matrices, A_1 and A_2 , is presented in Tables 6 and 7. The matrix A_1

Table 6

Coefficient Gain Matrices

Matrix A_1

```

0.1790E+01 -.6704E-01 0.8409E-03 -.7062E-02 -.2718E-03 0.5783E-05 0.1574E-02 -.2029E-04 0.5743E-07
0.7297E+00 -.2105E-01 0.2326E-03 -.2591E-01 0.8608E-03 -.1064E-04 0.7078E-03 -.3056E-04 0.4830E-06
0.2155E+01 -.1052E+00 0.1480E-02 -.1192E-01 0.5802E-03 -.8376E-05 0.1896E-02 -.7971E-04 0.1183E-05
0.7689E+01 -.3227E+00 0.4372E-02 0.5735E-02 0.4320E-03 -.7653E-05 -.1968E-02 0.3013E-04 0.1266E-06
0.3254E+01 -.8524E-01 0.9080E-03 -.2486E-01 -.1404E-02 0.2740E-04 0.1056E-02 0.1207E-03 -.2462E-05
-.3595E+01 0.1556E+00 -.2146E-02 0.9345E-02 -.6244E-03 0.9687E-05 0.5536E-03 -.5157E-05 -.9062E-07
0.5348E+00 -.6276E-02 -.7038E-04 -.1531E+00 0.4123E-02 -.4705E-03 0.3235E-02 -.1480E-03 0.2176E-05

```

Matrix A_2

```

-.1986E+01 -.2263E+00 0.6908E-01 0.1142E+00 -.6239E-03 -.2183E-02 -.3905E-01 0.1234E+00 0.8762E-03
-.5978E-01 -.1526E+00 0.1118E-02 0.9144E-02 -.1153E-04 -.1301E-03 0.8944E-02 -.2450E-03 -.3950E-05
-.1687E+00 -.6572E+00 0.5632E-02 0.3318E-01 -.9288E-04 -.4252E-03 0.9175E-01 -.3717E-01 -.1522E-02
-.9162E+00 -.3385E+01 -.1312E-01 0.1550E+00 0.3222E-03 -.2074E-02 0.1234E+00 0.3041E+00 -.3296E-02
-.4223E+00 0.9430E+00 0.2269E-01 -.4901E-01 -.3330E-03 0.7071E-03 -.6199E-01 0.1051E+00 0.1961E-02
-.6580E+00 0.1412E+01 0.3657E-01 -.6358E-01 -.5010E-03 0.8178E-03 -.8771E-01 0.6820E-01 0.1415E-02
-.6224E+01 0.6892E+00 0.3341E+00 0.1572E-02 -.4504E-02 -.1901E-03 0.3650E-01 -.5388E-01 -.2292E-02

```

Copy available to DTIC does not
 permit fully legible reproduction

```

0.3238E-02 -.1553E-04 -.7741E-04 0.3322E-03 -.4530E-02 -.9593E-04 -.1424E-03 0.1853E-05 0.3766E-
-.7969E-04 0.6372E-05 0.1957E-05 0.4981E-03 -.1815E-03 -.3247E-04 0.5954E-05 0.3536E-06 0.4980E-
-.1522E-02 0.2116E-04 0.2149E-04 -.3840E-02 0.2660E-02 -.3302E-04 0.5423E-04 0.3215E-06 0.2687E-
-.6569E-02 0.4196E-04 0.4993E-04 -.6807E-02 0.1183E-02 0.1429E-03 0.3267E-04 -.2300E-05 0.9122E-
-.3469E-02 -.2631E-04 0.4469E-04 0.2308E-02 -.5170E-02 -.4179E-04 0.1639E-03 0.5309E-06 -.2211E-
0.8433E-04 -.2001E-04 -.2753E-05 0.2882E-02 -.3604E-02 0.7385E-04 -.1119E-04 -.8737E-06 -.5247E-
0.1104E-02 0.3346E-04 -.8684E-05 0.1139E-02 0.2541E-02 0.2660E-05 -.7000E-04 -.7202E-06 0.1551E-

```

(where $f(T_c) = \epsilon C_0 T_c$) was 7 by 9 while A_2
 (where $f(T_c) = C_0 + C_1 T_c$) was 7 by 18. Taken together, the
 matrices reflected a reduction of one half over the exact
 solution. The average correlation was .9089 with the lowest
 being .8200 on gain 12.

4.4 GAIN SCHEDULING SIMULATION

In order to verify the accuracy of the gain schedules, a simulation was run at the nominal flight condition using the gain schedules, and the results were compared to those results obtained using the actual gains. Some slight differences were noted between the actual and scheduled gains and, consequently, the closed-loop eigenvalues and responses were slightly different.

The actual gains and the schedule gains at the nominal flight condition are listed below for comparison.

Actual Gains				Scheduled Gains					
	0.724	-.826	-.016	0.084		0.739	-.846	-.015	0.089
$C_b =$	0.288	-.673	0.601	2.908	$C_b =$	0.286	-.672	0.601	2.910
		-0.167	1.627				-0.174	1.650	
$C_f =$	-1.172	-.386			$C_f =$	-1.174	-.389		
		-0.253	0.0				-0.263	0.0	
$C_i =$	-2.898	0.0			$C_i =$	-2.900	0.0		

In cross checking the individual gains, small differences were noted between the actual and scheduled gains. This, in turn, led to the expectation that there would be differences, though hopefully small, in the eigenvalues and, hence, responses. The characteristic equations for the actual and scheduled closed-loop systems were as follows:

Actual Gains

$$C = (s+2.827-1.886j)(s+2.827+1.886j)(s+5.369-1.629j) \\ (s+5.369+1.629j)$$

Scheduled Gains

$$C = (s+2.868-1.848j)(s+2.868+1.848j)(s+5.370-1.656j) \\ (s+5.370+1.656j)$$

As expected, the eigenvalues did show only slight variations; thus, the scheduled gains give a good representation of the actual ones.

Since the gains and the eigenvalues of the actual and scheduled gain systems were only slightly different, it was reasonable to expect only slight variations when comparing the responses of the two. Figure 8 shows the response of the system to a roll rate command of 10 degrees per second. The rise time for the roll rate response was 0.152 seconds compared to 0.152 seconds for the actual gains. Settling time was 1.15 seconds compared to 1.15 seconds and overshoot was 14.91 percent over final value compared to 14.73. So the roll rate responses were almost identical. In addition, the gain schedule sideslip response did not reach steady-state as it did in the actual gain case. The rate of change is very small, however, so that the disequilibrium is not a significant factor.

Figure 9 shows the system response to a sideslip command of 2 degrees. Rise time for the sideslip response was 0.766 seconds compared to 0.751 seconds in the actual gain case. Settling time was 0.90 seconds compared to 1.70 seconds and overshoot was 0.95 percent over final value compared to 1.05. The settling time difference was attributable to the smaller overshoot and the way settling time is defined (time to within 1 percent of final value).

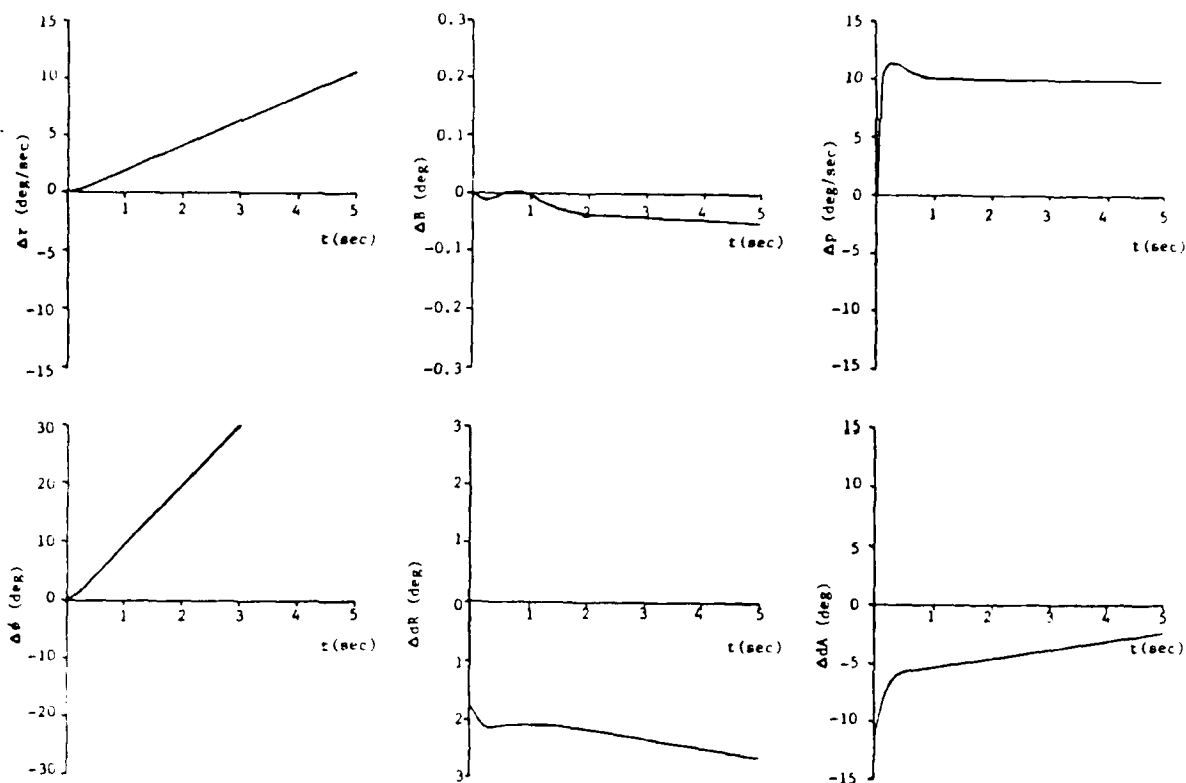


Figure 8: Gain Schedule Simulation - Roll Rate Command

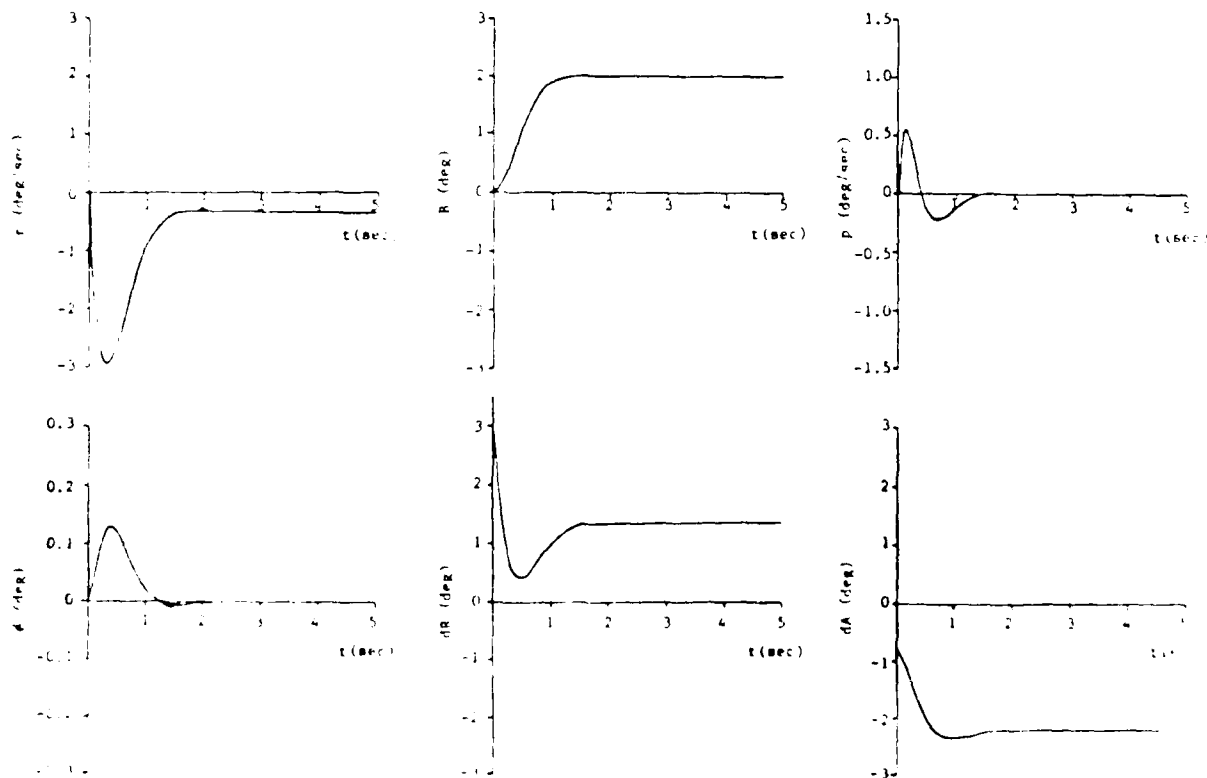


Figure 9: Gain Schedule Simulation - Sideslip Command

AD-A128 579

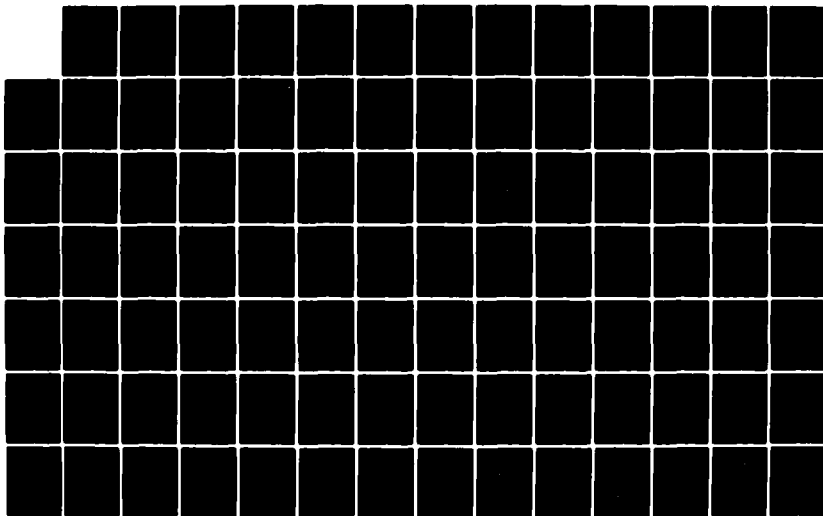
A LATERAL-DIRECTIONAL CONTROLLER FOR
HIGH-ANGLE-OF-ATTACK FLIGHT(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH W A EHRENSTROM MAR 83
AFIT/C1/NR-83-12T

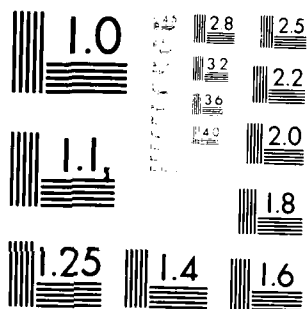
2/3

UNCLASSIFIED

F/G 1/2

NL





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

Chapter V

FLIGHT TESTING

The final test of the CAS designed in this study was to use the CAS in actual flight. To do this, CAS software compatible with the microprocessor systems was developed, implementing the control law developed in Chapter 3 and the gain schedules calculated in Chapter 4. Microprocessor limitations in memory space and computational speed as well as control system requirements of sampling time and accuracy were considered in the software development.

Once the software was developed, a limited number of ground tests were performed before the actual flight tests. The ground tests were used to check the CAS software against known simulation results and to insure that the correct signs on the outputs were generated in an actual run-time situation.

Finally, actual flight tests were performed. The flight tests included tests on the basic airframe to get an understanding of the lateral-directional characteristics in the stall regime and to help verify the model; tests of the pilot's ability to control the aircraft during the stalled conditions for comparison with the CAS; and finally, tests of the CAS operation wherein the control system was required to maintain a wings-level attitude throughout the stall maneuver.

This chapter, then describes the Microprocessor-based Digital Flight Control System (Micro-DFCS) and the software that was developed to implement the control system. In addition, a discussion of the ground tests is included. Finally, the actual

flight test results are covered including the flight test procedures, the results of the individual set of tests, and an analysis of the overall results.

5.1 DESCRIPTION OF THE MICRO-DFCS

There were four Micro-DFCS functions: accept analog inputs of aircraft states and pilot commands; update the gains; compute the control law; and output commands to the control surfaces. The microprocessor needs certain characteristics to perform these functions. It needs to have a reasonably fast computation time capability. The bit precision should be at least as good as that available from the A/D converters. Finally, the A/D and D/A converters must have a resolution compatible with aircraft sensors.

The Micro-DFCS is built around a Monolithic Systems Corporation (MSC) 8004 board. The MSC 8004 board has 32K of random-access memory (RAM), of which 28K is available for the CAS software; a programmable read-only memory (PROM) containing the Uniform Monitor (UFM) for loading, running, and dumping the CAS program; a Zilog Z80 central processing unit (CPU); and an AM-9511 high-speed mathematics unit. In addition, the Micro-DFCS has A/D and D/A converter boards. All three boards are put into a card cage and are connected to a hand-held control-display unit (CDU).

The ARA is equipped with analog, digital, and mechanical control systems. An overview of the control system inter-relationship with the aircraft and pilots is presented in Figure 10. The analog and digital control systems can operate simultaneously such that digital control of the lateral-

directional mode can be accomplished without affecting the longitudinal mode.

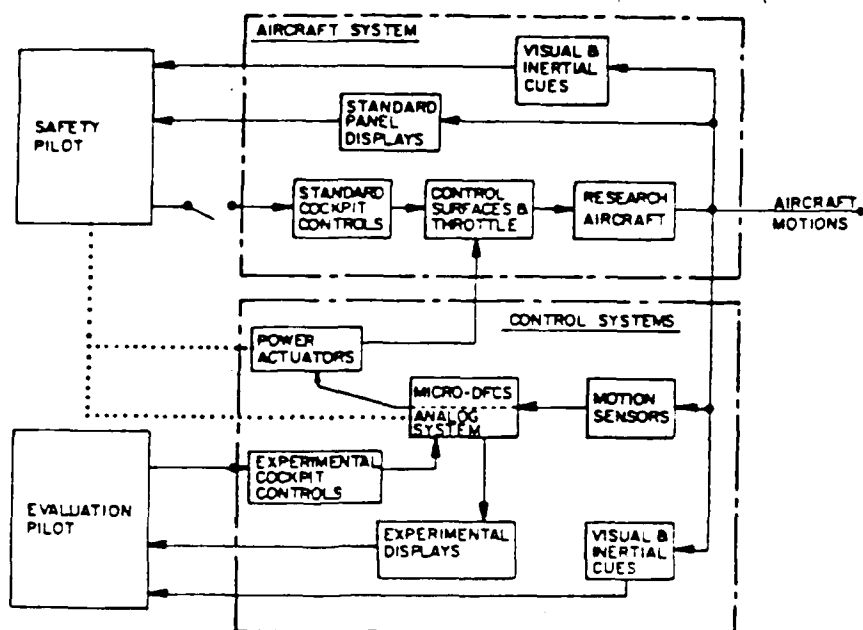


Figure 1C: Overview of Aircraft Systems Configuration

5.2 CAS SOFTWARE

The CAS software was required to perform the Micro-DFCS functions for control of the aircraft. The control system required that all tasks in the control sequence (including the gain updates and the control law calculations) be accomplished

within the sampling time. Because the microprocessor had limitations concerning the number of computations that could be done within any specific time period, the CAS software had to compromise these conflicting requirements.

The software was developed using the MSC 8009 disk system available at FRL. The 8009 system consisted of a MSC 8009 board, a card cage, two SMS floppy-disk drives, an ADM-31 terminal, and an Anadex 9501 line printer. The 8009 board, card cage, and disk drives are mounted, with a power supply, in a cabinet which also houses FRL's Telemetry Monitoring system. The 8009 computer uses the CP/M system monitor. The monitor can be configured to run with 32k or 64K of RAM, 32K provided by the 8009 board and the additional 32k provided by other circuit boards.

There were two problems with implementation that had to be resolved for the control system to work. The first was the integration of the roll rate command to multiply by the integral gain matrix, C_i . This was easily resolved by noting that the integral of roll rate is the roll angle so that the integral could be found by the following equation:

$$\begin{aligned}\Delta p_c &= \Delta \dot{p}_c + \Delta p_c * \Delta t \\ \Delta \phi_c &= \int \Delta \dot{p}_c\end{aligned}\tag{5-1}$$

where $\Delta \dot{p}_c$ is the roll rate command and Δt is the sampling time. The second problem was the determination of the trim condition about which the perturbations would be measured. This was

resolved by allowing the operator to set a flag to reset the nominal condition anytime the aircraft was at a new trim. The input was put in as part of the background program which ran when the control law sequence was not running.

The control sequence of CAS was initiated by an interrupt from the clock at each sampling time. When the control sequence was not being executed, a background routine accepted inputs from the CLU and had the capability of performing a limited number of tasks.

The computation of the control law began by entering the current values of the states. The perturbation values then were found by subtracting the nominal state values (i.e., the trim condition, which was set when the program was initialized) from the current value of the state. The rudder and aileron commands then were computed with the perturbations and the control gains. Finally, the commands were sent to the control surfaces.

The sequence described above did not present a problem with computation time. However, it was desirable to update the gains continuously to account for changes in the flight condition. The number of calculations required for such a task proved to be time consuming indeed. The control law calculations and update of the 14 gains required 0.25 seconds -- two and a half times the sampling interval. To run at this rate would seriously degrade the control law effectiveness, particularly since the gains were all computed with a sampling interval of 0.1 seconds.

To circumvent this problem, the gains were updated over a number of control cycles such that only one gain per sampling interval was updated. Thus, each gain was updated every 1.4 seconds. Updating more than one gain caused the control sequence to use more than its allotted time. Figure 11 shows the typical execution of the CAS program. The low break corresponds to entering the control sequence, while the high break corresponds to leaving it. As can be seen in the figure, the control sequence used about 0.08 seconds in this form.

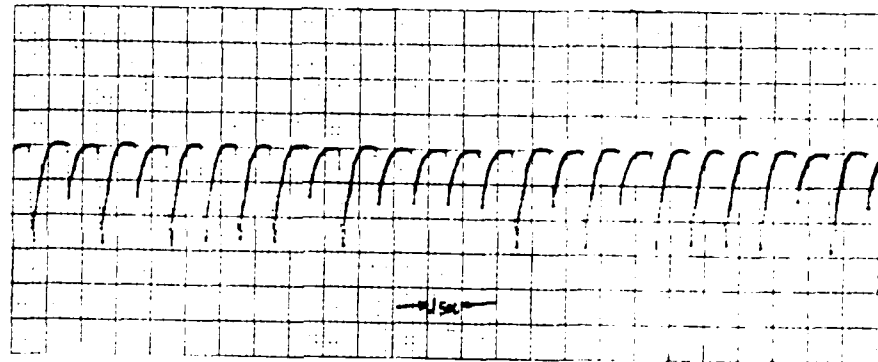


Figure 11: Control System Execution Cycle

The gain updates were accomplished prior to computation of the control law so that the most recent gains would be used. Rudder gains were updated first, followed by aileron gains. A flowchart for the control sequence is depicted in Figure 12 .

The background routine allowed the pilot to input commands for a limited number of options including: reinitialization, halting,

and breaking the program execution; a test for A/D and L/I functions; and a reset for the nominal condition state values. A detailed description of the background routine and the control sequence is presented in Appendix C.

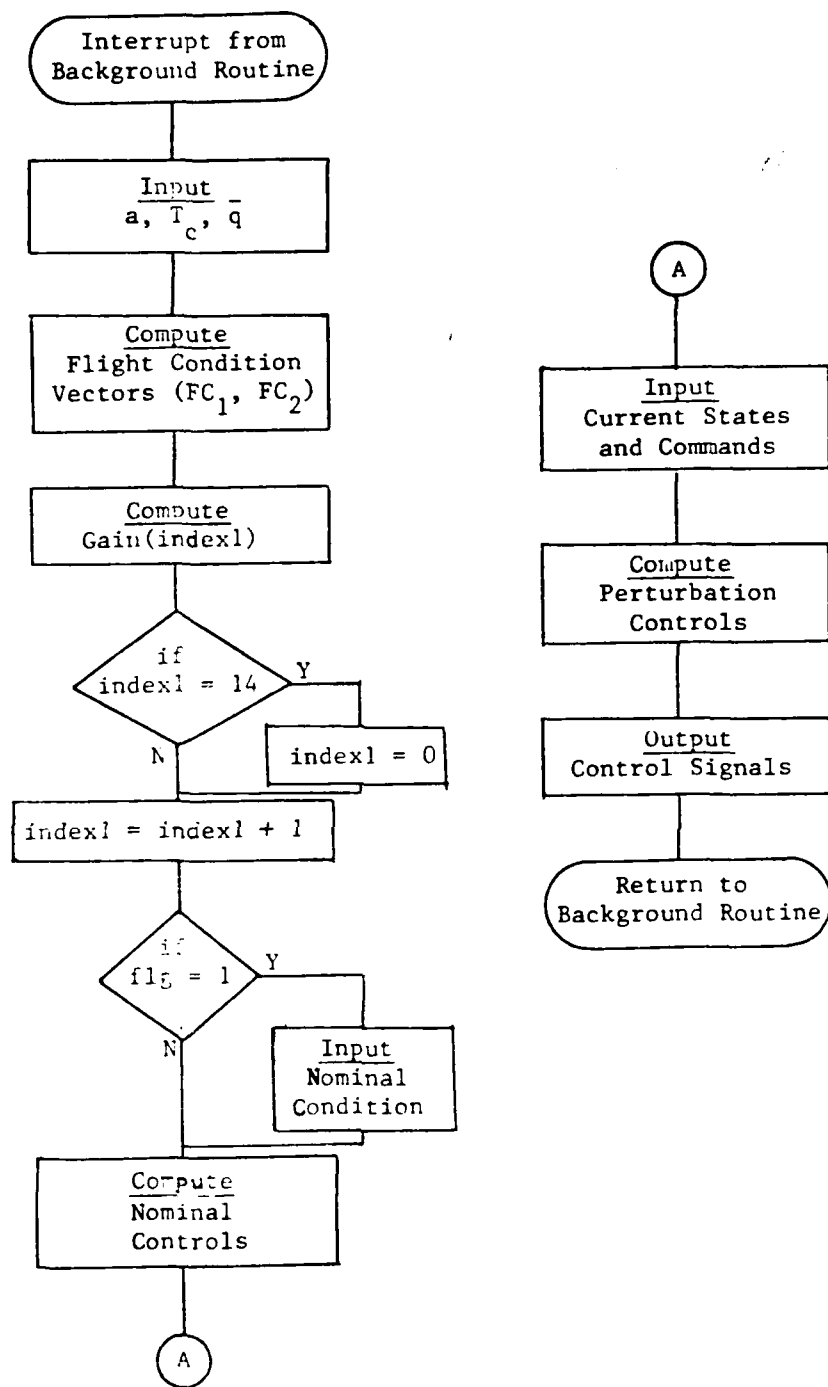


Figure 12: Flowchart of the Control Sequence

5.3 GROUND TESTS

Before the flight tests were performed, a preliminary set of tests was accomplished on the ground to insure that the software computations were correct and that the correct signals were being sent to the control surfaces. Two sets of ground tests were used to verify the CAS operation.

The first test included interfacing the microprocessor to the analog computer and sending voltages from the analog to the microprocessor. The voltages corresponded to known values of the state and flight condition variables found from simulations. Outputs were then checked to insure that the correct voltages were being sent out from the microprocessor to the control surfaces. To facilitate this check, the software had a ground test mode which enabled each step of the control law calculation to be sent to the line printer. These results were then cross-checked against simulation results and the discrepancies corrected.

Once it was ascertained that the control system calculations were being done correctly, a second set of ground tests was performed to insure that the signs of the outputs were correct. The aircraft control surfaces were capable of being operated on the ground using auxiliary power and hydraulic sources. The CAS was loaded onto the airplane as it would be during the actual flight tests and then inputs were generated for the microprocessor using the aircraft sensors. The tests were used to prove that for a given input, certain control surface deflections would be generated. With the completion of these tests, the control system was ready for actual flight situations.

5.4 FLIGHT TESTS

The objective of the flight tests was to examine the capabilities of the CAS into the stall regime and compare the results to those of the pilot in a similar situation. It was hoped that the control system would maintain lateral-directional stability better than the pilot could and thus free the pilot to perform other tasks. To accomplish this objective, a series of flight tests were developed and flown to examine different aspects of the aircraft, pilot, and control system.

The first set of tests was set up to examine the stability of the basic aircraft in the stall regime. To do this, the lateral-directional controls were locked in a trim condition just prior to stalling the aircraft. Based upon the results of the model developed in Chapter 2, the aircraft was expected to exhibit lateral-directional instabilities once the stall was encountered. The results of this set of tests pointed out the problems that the pilot and CAS were required to overcome.

The second set of tests looked at the pilot's abilities during the stall. In this case, the pilot attempted to maintain a wings-level attitude while the aircraft was stalled. Based on previous stall testing, the pilot was expected to have trouble overcoming the instabilities. The results from these tests were used in comparison with the CAS operation.

The final set of tests examined the control system capabilities. In these tests, the pilot stalled the aircraft while

the CAS attempted to maintain lateral-directional trim. From the results of the CAS development, the control system was expected to provide stability for the aircraft into the stall regime. These tests were used to determine the success or failure of the CAS based on its ability to maintain stability.

For each of these tests, a couple of conditions were introduced which were known to affect the stability of the aircraft. Since it had been found that throttle setting affected stability (Ref. 1), two throttle settings were examined. In addition, "pitch pumping", or the rapid oscillation of the angle of attack, was also found to radically affect lateral-directional stability. In this case, it was also interesting to find out how well the CAS could keep up with the oscillations since the gains were angle-of-attack sensitive but were rescheduled completely only once every 1.4 seconds. Thus, each set of tests included four test runs including: low power setting with no pitch pumping; low power with pitch pumping; high power with no pitch pumping; and, finally, high power with pitch pumping. The tests outlined above are summarized in Table 7.

Documentation of the test results for analysis and inclusion into the report was accomplished using a data telemetry system already incorporated into the aircraft and ground station. In this case, the aircraft telemetry system received data on the aircraft attitude from aircraft sensors. These data then were transmitted to a recording system at the ground station.

Table 7

FLIGHT TEST DOCUMENTATION

Trim Conditions: V=70 kts., $\delta f=0^\circ$, MAP=20", RPM=2500,
Mixture=Normal

<u>Run</u>	<u>Lat/Dir Controls</u>	<u>Power</u>	<u>Pitch Pulse</u>
1-1	Locked	15"/2500 RPM	N
1-2			Y
1-3		25"/2500 RPM	N
1-4			Y
2-1	Manual	15"/2500 RPM	N
2-2			Y
2-3		25"/2500 RPM	N
2-4			Y
3-1	CAS	15"/2500 RPM	N
3-2			Y
3-3		25"/2500 RPM	N
3-4			Y

For these tests, the data which could be recorded were limited to four channels. The flight condition variables, angle of attack and velocity, were recorded to ascertain the onset and severity of the stall. In addition, these two variables (plus throttle setting which was constant for each test run) were used to schedule the gains. Along with these variables, two lateral-directional attitude variables were recorded: sideslip and roll angle. These variables were used to determine the performance of the pilot and CAS during the test runs.

5.4.1 Airframe Tests

The purpose of these tests was to get a feel for the stability of the basic airframe in a stalled condition. To accomplish these tests, the aircraft's lateral-directional controls were trimmed and locked into position. The aircraft was then stalled and its response recorded. From there, the data were analyzed and a picture of the lateral-directional stall characteristics was formed.

From the model, a preliminary idea of how the aircraft should react in a stall was obtained. The eigenvalues for the aircraft at a stall angle of attack, low power setting and low airspeed point to an unstable roll-spiral mode. In this case, a divergent roll angle with some oscillations was expected. For high power setting, the eigenvalues predict

the same type of response with a slightly longer time constant and slower oscillation rate. No information was available from the model pertaining to the effect of pitch pumping.

Figures 13 through 16 present the results of this set of tests. In each test, the stall is characterized by the aircraft exceeding the critical angle of attack, which from NASA TN D-5758 was found to be 18 deg angle of attack. (The spikes which occur throughout the data were the result of telemetry dropouts.)

In the case of the low power settings (Figures 13 and 14), the results were as expected. In each case, as the stall was encountered, the aircraft began to oscillate around a slowly divergent roll angle. In Figure 13, with no pitch pumping, the aircraft was trimmed for a roll angle of 3.2 deg and a sideslip of 4.5 deg. Twenty seconds after the stall was encountered, the roll angle reached a maximum of 61.5 deg before the aircraft was recovered. Sideslip, while apparently not divergent, was very erratic, ranging from -9.5 deg to 5.8 deg -- as much as 14.2 deg from trim. The maximum angle of attack encountered was 33.1 deg and the minimum velocity was 61.6 knots.

In Figure 14, low power setting with pitch pumping, the results were similar. The trim conditions were 4.8 deg for roll angle and 7.7 deg for sideslip. The maximum roll angle prior to aircraft recovery was 69.4 deg occurring 28 sec after the

stall was encountered. Sideslip ranged from 0.0 deg to 11.9 deg and as much as 7.7 deg from trim. The maximum angle of attack was 34.7 deg. During pitch pumping, the aircraft was subjected to changes in angle of attack as high as 15.6 deg/sec. The minimum velocity encountered was 61.6 knots.

The problem of stalling at a higher power setting is shown in Figures 15 and 16. In both cases, the aircraft showed very little oscillation compared to the low power setting but the divergences were of larger magnitude with higher roll rates.

In Figure 15, with no pitch pumping, the aircraft was trimmed with a roll angle of 6.5 deg and a sideslip of 7.1 deg. Once the stall was encountered, the aircraft remained somewhat stable for a few seconds as the angle of attack increased. Sideslip departed trim first, oscillating from -16.5 deg then back to 18.7 deg. Maximum divergence from trim was 23.6 deg. Roll angle hesitated prior to departing the trim condition and then simply rolled onto one wing. Maximum roll angle was undetermined since the plot went off the scale but was in excess of 75.0 deg. Large roll rates as high as 36.5 deg/sec were encountered. Maximum angle of attack was also undetermined but was in excess of 40.0 deg. The minimum velocity was 50.0 knots.

In Figure 16, high power setting with pitch pumping, the results are similar. The aircraft was trimmed for a roll angle of 6.5 deg and a sideslip of 7.1 deg. After the aircraft was

stalled, sideslip once again departed trim first, ranging from 15.2 deg to -17.1 deg -- as much as 24.2 deg from trim. Roll angle shows the aircraft rolling one direction and then back the other before being recovered 21 sec after the stall was encountered. Maximum roll angle was 79.9 deg and was as low as -37.9 deg. The maximum change in angle of attack was 18.2 deg/sec. Minimum velocity was 50.0 knots.

In comparing the results of including pitch pumping against those where pitch pumping was not done, there seems to be little effect on the stability of the aircraft. Indeed, with the exception of the roll left in Run 1-4 prior to the roll right, the same power setting gave similar results regardless of pitch pumping.

It is also interesting to note that the aircraft always ends up rolling towards the right wing and that the maximum deviation from trim sideslip always occurs with negative sideslip. These results are due to torque effects of the engine and point to problems for both the pilot and CAS in trying to maintain stability of the aircraft into the stall regime.

Overall, the aircraft performed as predicted by the model. At the low power settings, the aircraft went into a divergent oscillatory roll once stall was encountered. At the higher power settings, less oscillation was encountered. The maximum divergence occurred around the same time in both cases.

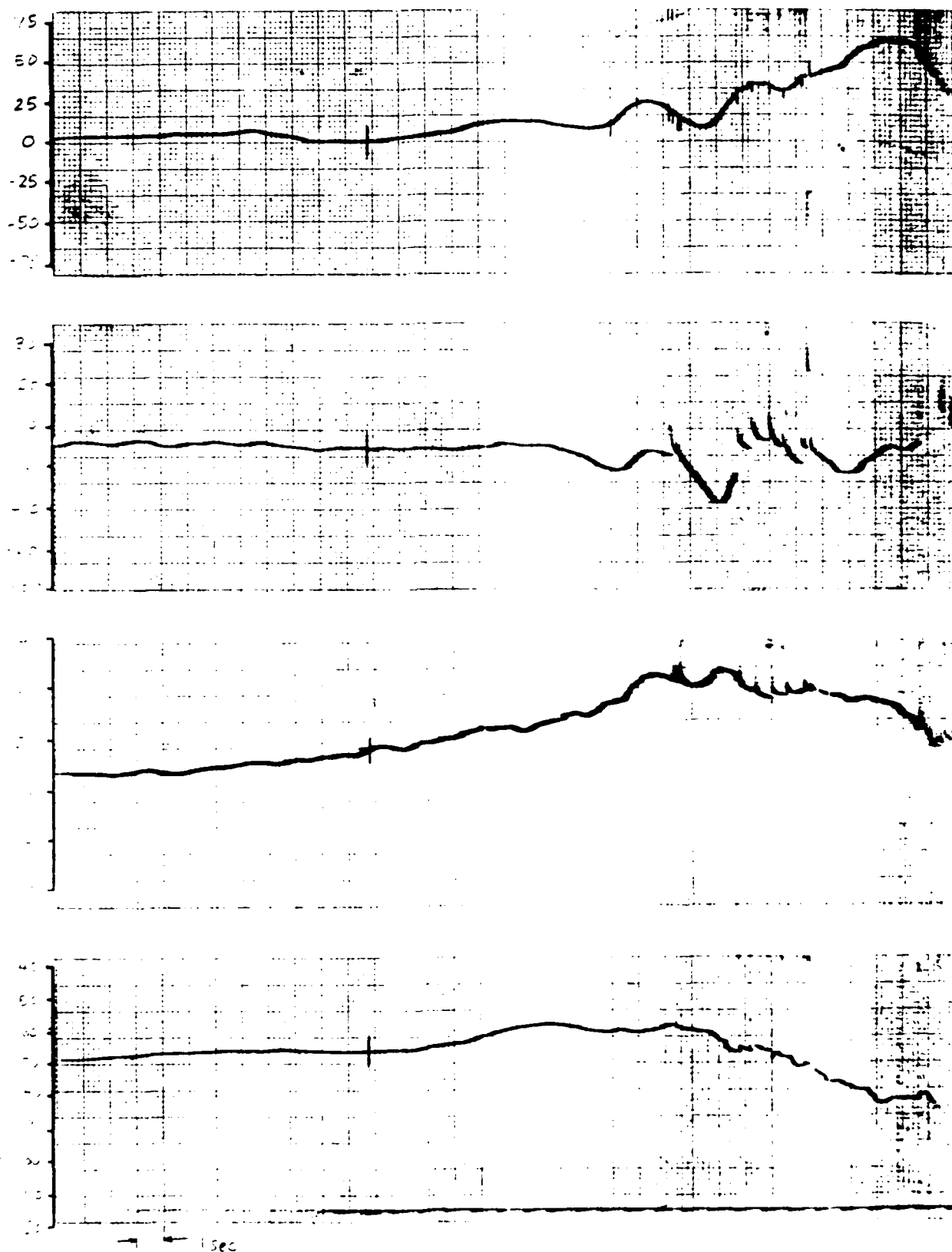


Figure 13. Flight Test Run 1-1 Results.

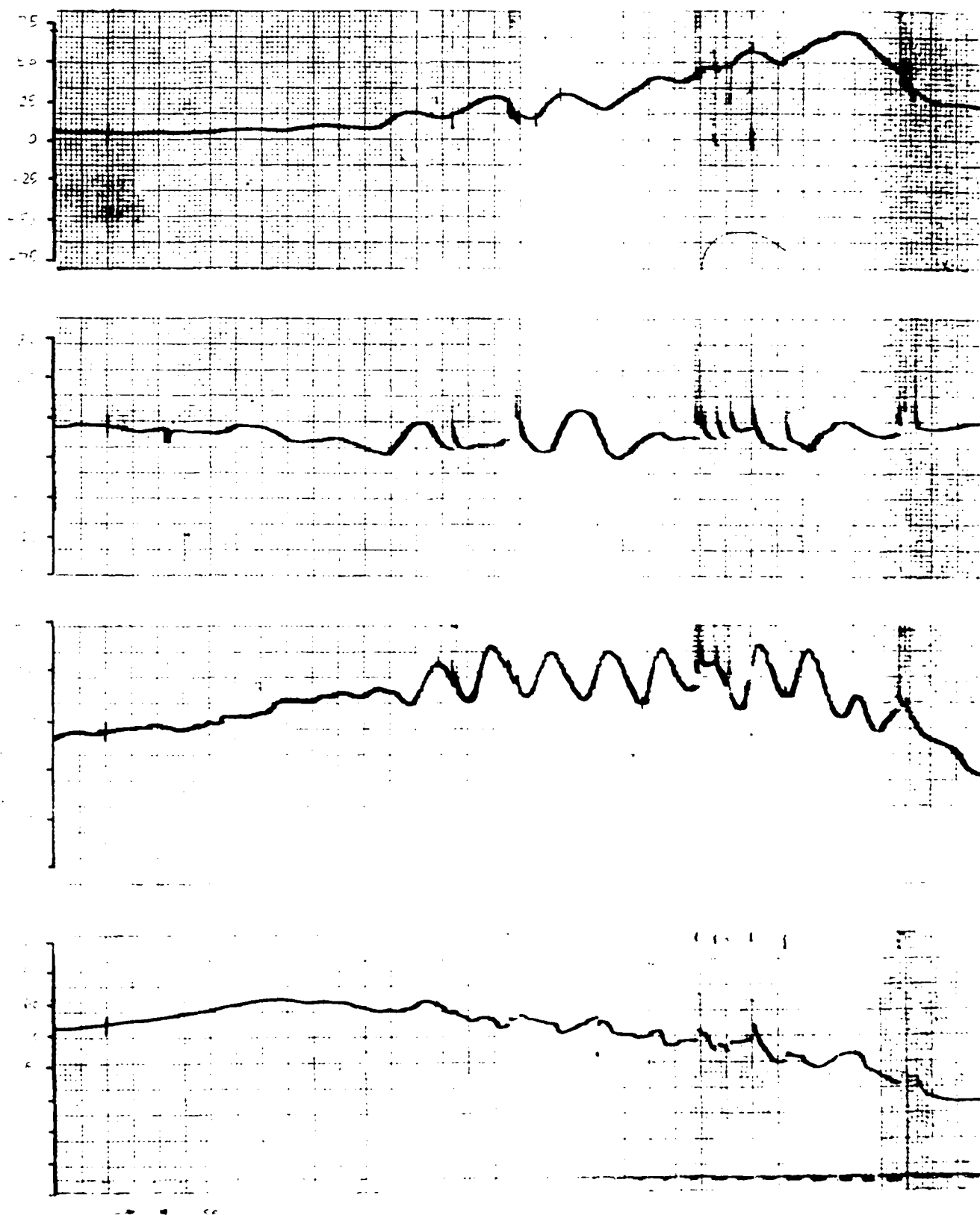


Figure 14. Flight Test Run 1-2 Results.

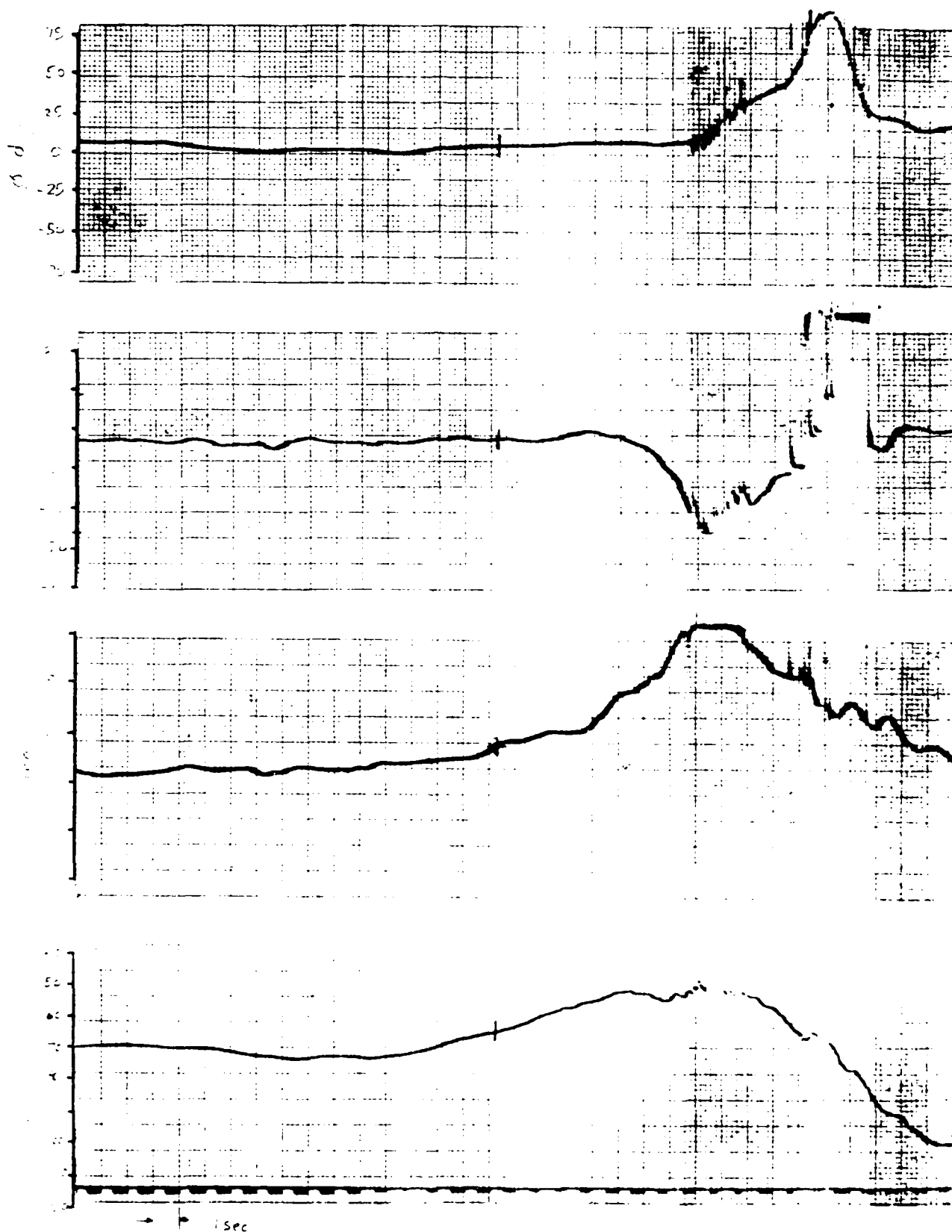


Figure 15. Flight Test Run 1-3 Results.

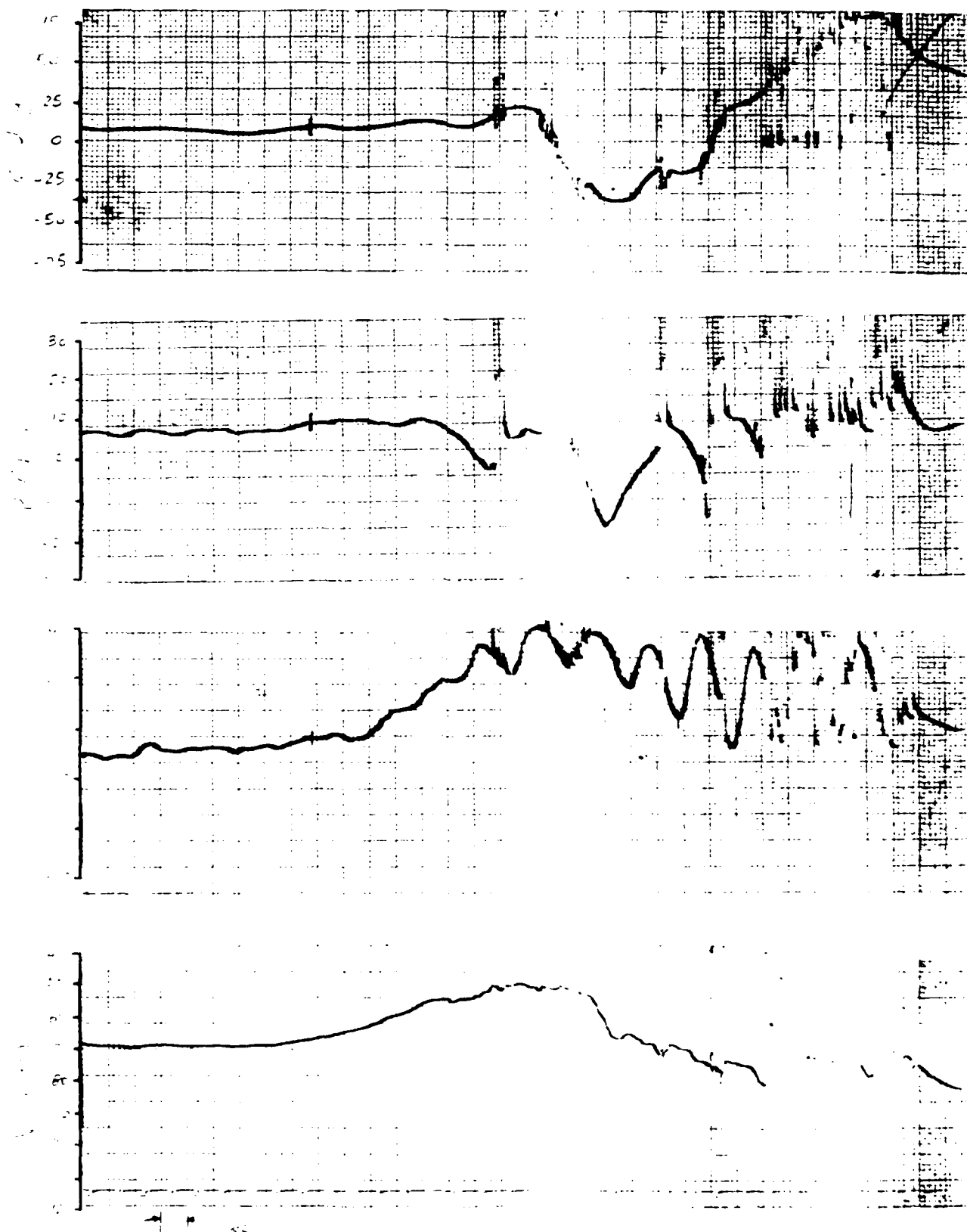


Figure 16. Flight Test Run 1-4 Results.

5.4.2 Pilot Tests

The purpose of this set of tests was to determine how well the pilot was able to maintain lateral-directional stability when the aircraft was stalled. The results were used as a baseline for comparison with the CAS in a similar situation. Figures 17 through 20 present the results.

At low power settings (Figures 17 and 18), the pilot was able to control the instabilities fairly well. In Figure 17, low power setting without pitch pumping, the aircraft's lateral-directional attitude showed only minor deviations. Roll angle ranged from 4.9 deg to 12.9 deg and was in general pretty steady. Sideslip was also fairly steady, ranging from 1.3 deg to 9.4 deg. The maximum angle of attack encountered was 32.6 deg and the minimum velocity was 53.2 knots.

Figure 18 shows the results of low power setting with pitch pumping. Once again, the pilot was able to control the instabilities well into the stall regime, though the addition of pitch pumping caused larger deviations. Roll angle varied anywhere from -1.6 deg to 33.1 deg while sideslip varied from 0.0 deg to 16.8 deg. The maximum angle of attack encountered was 37.8 deg with rates of change as high as 15.6 deg/sec. Minimum velocity was 53.2 knots.

The higher power settings proved to be much more difficult for the pilot. Both roll angle and sideslip deviations became large compared to low power settings. In particular,

the sideslip deviations became unmanageable once the stall was encountered.

Figure 19 shows the results of high power setting without pitch pumping. Roll angle deviations were large though manageable, varying from -9.7 deg to 26.6 deg. Sideslip variations were also large and much more erratic. Sideslip angles from -9.4 deg to 12.6 deg were encountered. Maximum angle of attack was 37.9 deg while minimum velocity was 51.6 knots.

Figure 20 presents the results of high power setting with pitch pumping. In this case, the aircraft's instabilities became unmanageable for the pilot. Roll angle varied anywhere from -34.7 deg to 43.5 deg and would have continued to increase had the aircraft not been recovered. Sideslip, too, showed large deviations ranging from -25.8 deg to 20.7 deg and was continuing to increase up until aircraft recovery. The highest angle of attack was 41.0 deg, changing at rates as high as 24.3 deg/sec during pitch pumping. Minimum velocity encountered was 52.1 knots.

In comparing the results of no pitch pumping to those where pitch pumping was added, some definite differences were noted. At low power settings without pitch pumping, only small deviations in lateral-directional attitude were noted. When pitch pumping was included, the deviations became markedly larger, particularly in the case of sideslip. Total variation of roll angle without pitch pumping (measured by the angle

between the minimum and maximum roll angle) was 8.0 deg compared to 34.7 deg with pitch pumping. For sideslip, total variation without pitch pumping was 8.1 deg compared to 16.8 deg with pitch pumping.

The results were similar in the case of high power setting. Total roll angle variation without pitch pumping was 36.3 deg compared to 78.2 deg with pitch pumping. Total sideslip deviation without pitch pumping was 22.0 deg compared to 46.5 deg when pitch pumping was included.

In contrast to the airframe tests, then, pitch pumping made a discernable difference in the results where the pilot was required to maintain lateral-directional stability. This could be attributed to a number of reasons. First, in trying to maintain stability at low airspeeds, the pilot was using sluggish control surfaces. Since the aircraft's attitude was changing rapidly (due to the inclusion of pitch pumping), the combination of relatively slow reaction time of the pilot and sluggish control surfaces would make it difficult to overcome the instabilities.

Second, whereas the pilot had outside references for maintaining wings level, there were no similar references for sideslip. If the pitch pumping caused sideslip deviations, the pilot would not have been able to sense and correct them. Flying in a sideslip would tend to aggravate the aircraft's instabilities.

Finally, just the effort of including pitch pumping into the results may have taken away from the pilot's attention enough that he was unable to maintain stability as well as when pitch pumping was not included.

Overall, this set of tests showed that differences in power setting and the inclusion of pitch pumping made a real difference in how well the pilot was able to maintain stability. Though no simulation results were available for comparison, the pilot did perform as had been found in previous studies.

5.4.3 CAS Tests

The final set of tests was run to determine how effective the control system designed in this study was in eliminating the instabilities encountered when the aircraft was stalled. To accomplish this task, the aircraft was trimmed prior to actuating the control system, and the aircraft was stalled. The control system then was required to maintain the lateral-directional trim into the stall.

The control system was designed with relatively rapid response times. Roll rate response was selected to be as quick as possible, while the sideslip response was chosen to be about one second. In both cases, the response times were found to be sufficient to provide stability into the stall regime in all simulation results. The results from this set of tests

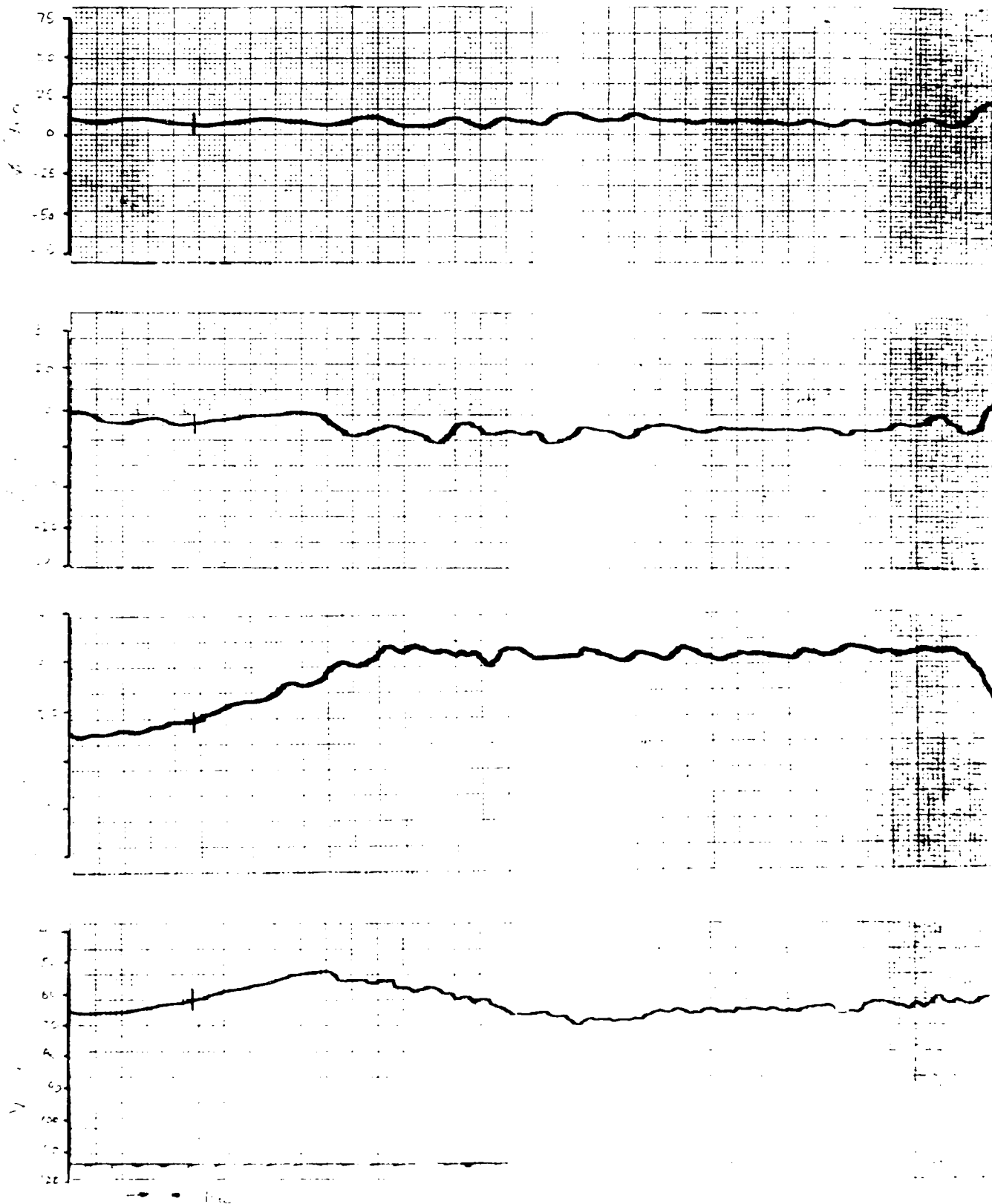


Figure 17. Flight Test Run 2-1 Results.

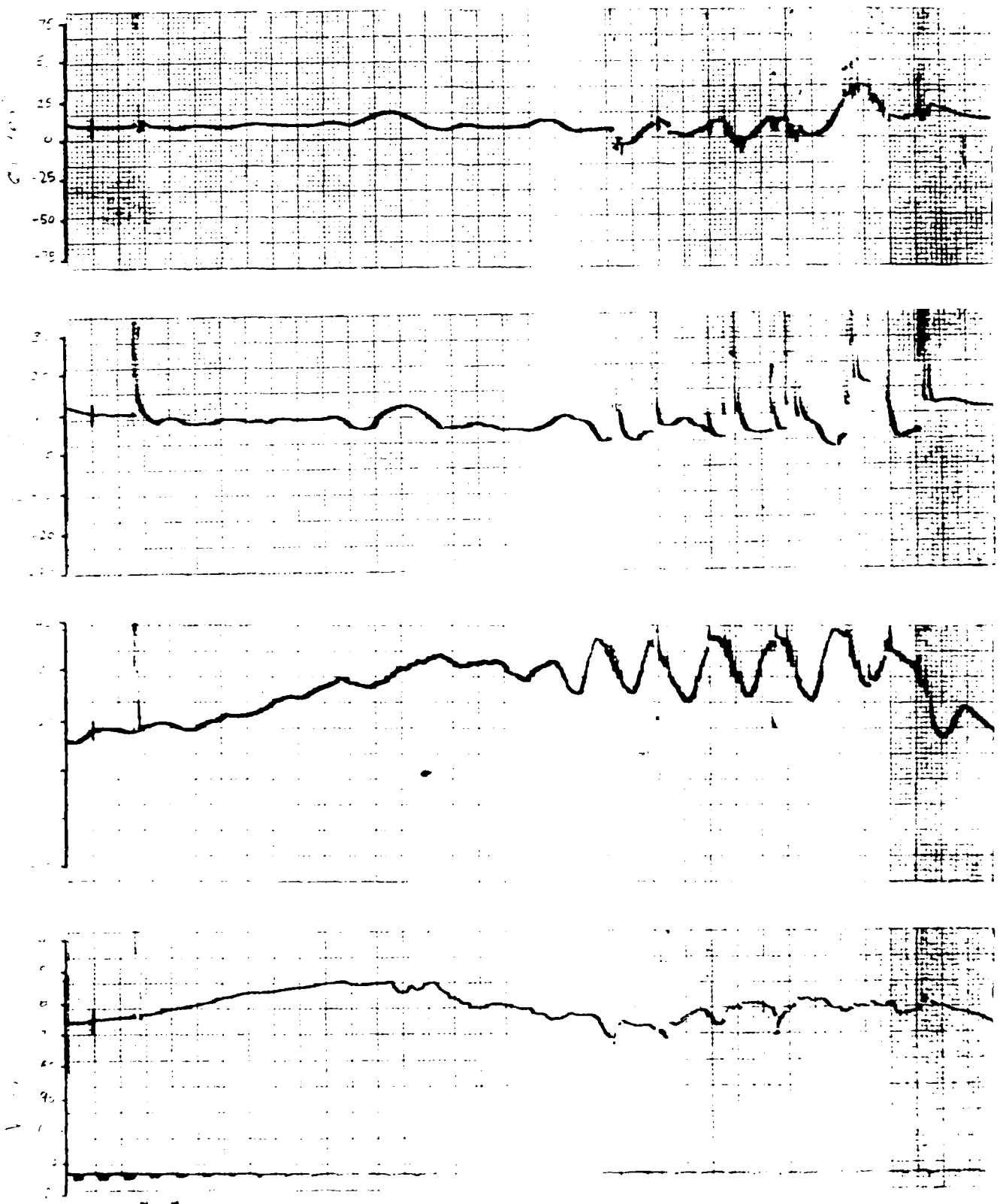


Figure 18. Flight Test Run 2-2 Results.

Copy available to DTIC does not
permit fully legible reproduction

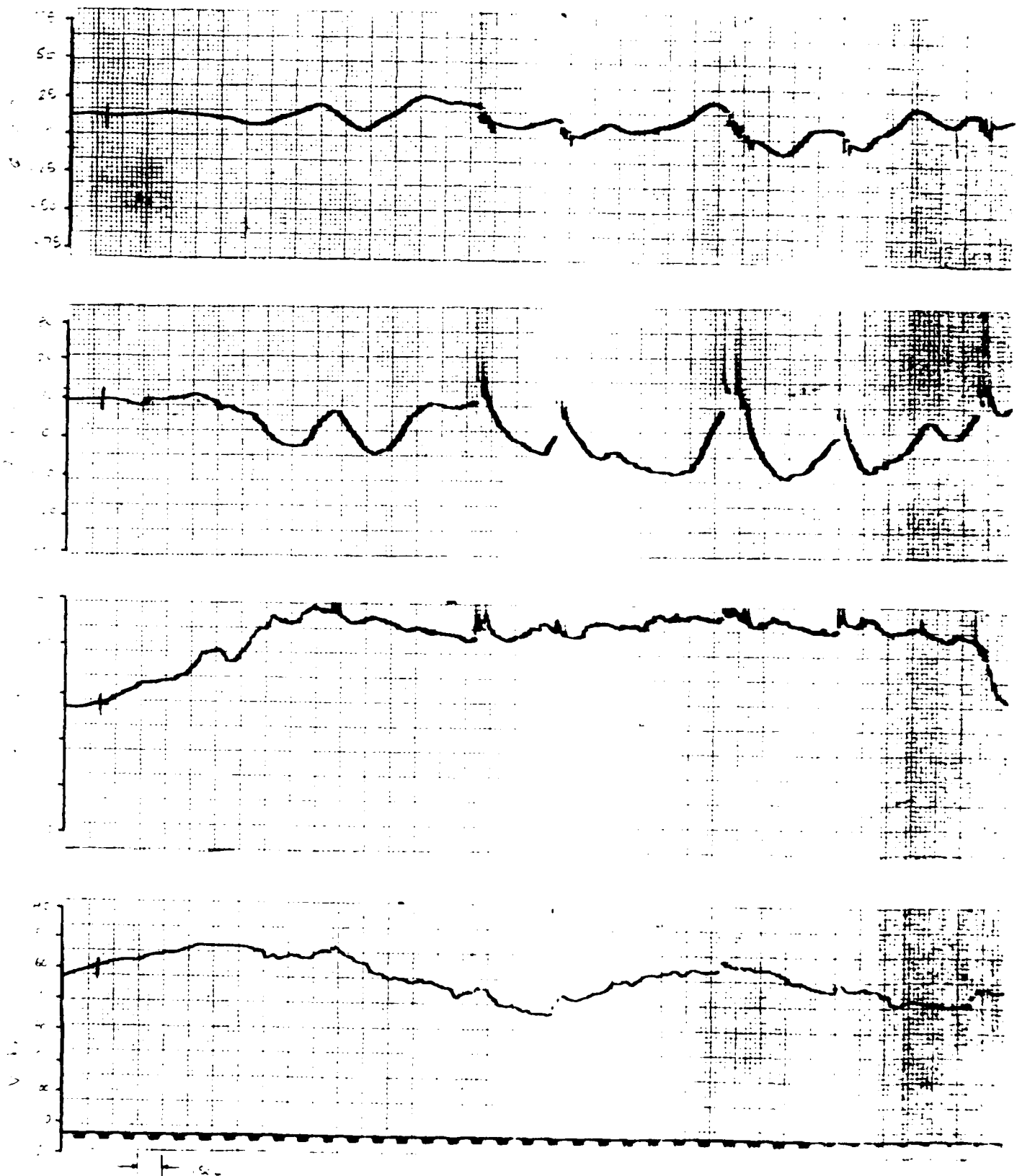


Figure 19. Flight Test Run 2-3 Results.

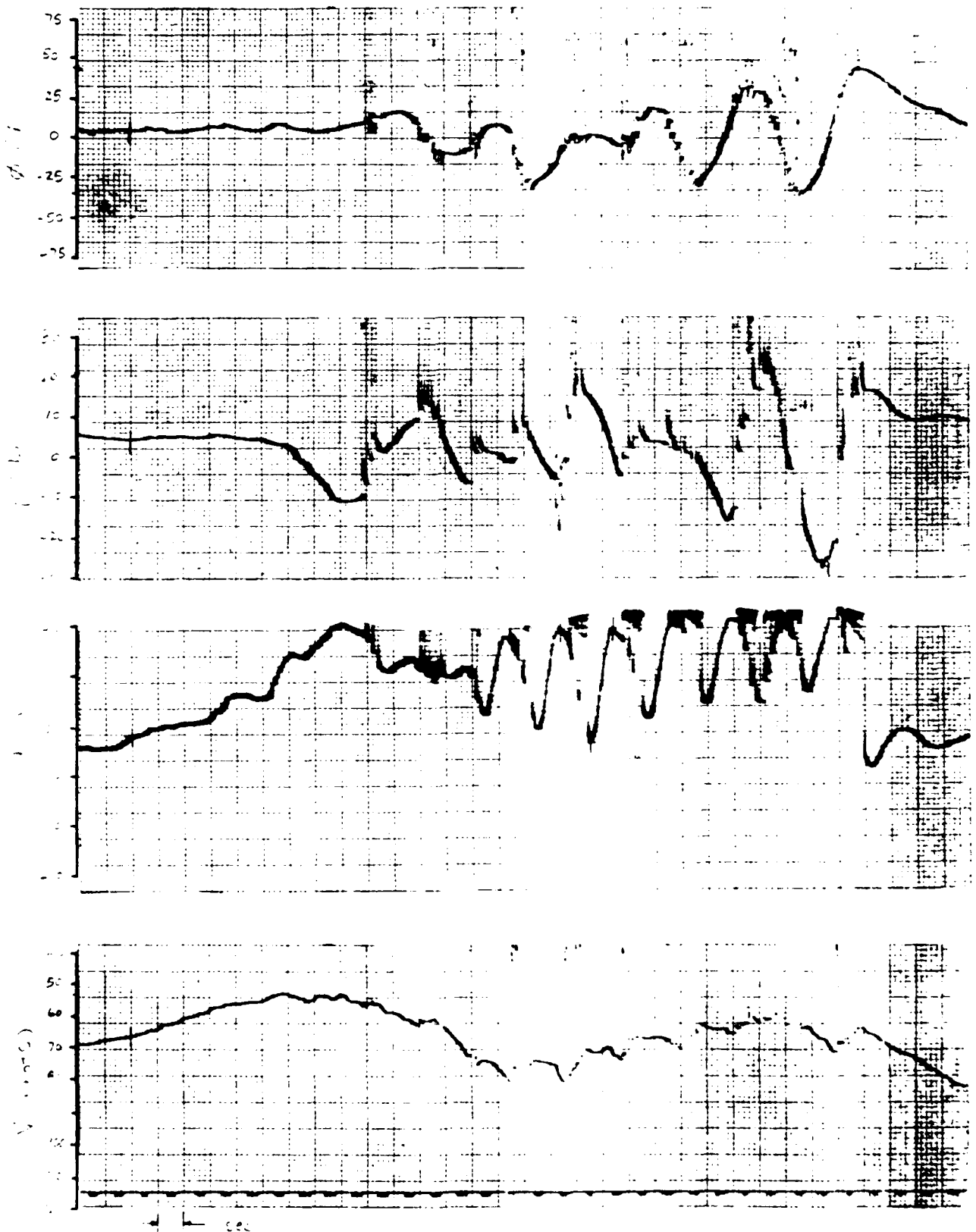


Figure 20. Flight Test Run 2-4 Results.

are presented in Figures 21 through 24.

The low power setting results are shown in Figures 21 and 22. When no pitch pumping was present (Figure 21), the CAS had no trouble at all maintaining stability. Roll angle showed negligible variation from the trim value of 8.1 deg. Sideslip did show some deviations from the 8.1 deg trim value, ranging from 8.7 deg to 3.2 deg -- as much as 4.9 deg from trim. This deviation is probably due to the slower response time demanded for sideslip. The maximum angle of attack was 30.0 deg while the minimum velocity was 50.0 knots.

When pitch pumping was included (Figure 22), the CAS still responded to maintain lateral-directional stability. Roll angle was trimmed at 4.9 deg. Once stall was encountered and the pitch pumping begun, the aircraft did a slow, counterclockwise roll to -6.5 deg. This slow roll was probably due to the inability of the control system to keep up with the rapidly changing flight condition. Sideslip also showed some minor deviations during pitch pumping. Sideslip was trimmed for 5.2 deg but was as high as 11.3 deg or 6.1 deg from trim. The maximum angle of attack was 31.0 deg while the maximum rate of change of the angle of attack was 20.8 deg/sec. The minimum velocity encountered was 51.6 knots.

Higher power settings also did not pose a problem for the control system though larger deviations from trim were noted than at the lower power settings. In Figure 23, high power

setting without pitch pumping, roll angle was trimmed at 9.7 deg and varied from 16.2 deg to 6.5 deg or as much as 6.5 from trim. Sideslip was trimmed at 8.7 deg and varied from 11.3 deg to 5.2 deg or as much as 3.5 deg from trim. The maximum angle attack encountered was 30.0 deg while the minimum velocity was 50.0 knots.

When pitch pumping was included with a higher power setting (Figure 24), the results were still more than adequate. Roll angle was trimmed at 9.7 deg and varied between 16.2 deg and 3.2 deg. Maximum deviation from trim was 6.5 deg. An oscillation occurred when the pitch pumping was ended and the aircraft recovered and was probably due to the control system catching up to the current flight condition. Sideslip was trimmed at 11.3 deg and varied from 13.2 deg to 6.8 deg -- as much as 4.5 deg from trim. The largest deviations occurred during the pitch pumping phase of the test and, once again, are probably due to the inability of the control system to keep up with the changing flight condition. The maximum angle of attack was 36.9 deg while the largest rate of change for angle of attack encountered during pitch pumping was 13.4 deg/sec. Minimum velocity was 47.6 knots.

As was in the case of the pilot tests, pitch pumping did have an effect on the results, though not as drastic. At low power settings, the test without pitch pumping showed no variation in roll angle and a total sideslip variation of 5.5 deg.

When pitch pumping was included, total roll angle variation went up to 11.3 deg and total sideslip variation to 6.1 deg. The same kind of results occurred at the higher power settings as well. Total roll angle variation without pitch pumping was 9.7 deg compared to 12.9 deg with pitch pumping. Total sideslip variation without pitch pumping was 6.1 deg while with pitch pumping, it was 6.5 deg. These results were due to the slow speed at which the control system updated the gains and the fact that the angle of attack was changing at rates as high as 20.8 deg/sec. Regardless, the CAS provided adequate control throughout the stall.

One problem with the control system was encountered which makes it unusable in its present form. The aircraft's control surfaces are set up with a safety feature to cutout if the CAS commands excessively large deflections at any one time. This was a problem for the CAS since it was operating in a very unstable region where the control effects were sluggish. Large control surface deflections were required to maintain aircraft control. In the case of several test runs, numerous attempts were needed to get one "successful" run where cutouts did not occur too early.

Several possibilities exist for why this problem arose. First, when the weighting matrices were selected, a minimum roll rate response was considered desirable. However, this may force the CAS to correct deviations from trim too fast and cause excessively large aileron commands. Another possible problem is

that of sampling time. At high angles of attack, the aircraft is very unstable in the roll-spiral mode. The aircraft can diverge quite rapidly, as was demonstrated in the first set of tests, so the control system must be capable of reacting equally as fast. The CAS would have to make up for a slow sampling time by using large control surface deflections. Finally, the safety limits on the aircraft may be too tight for the regime in which the control system was tested and could be widened somewhat. Thus, the solution to the cutout problem would be to make changes in all three potential problem areas -- slow down roll rate response, increase the samples per interval of time, and relax the control surface safety limits on the aircraft.

Another answer to the cutout problem might be low-pass filtering. This scheme includes not only control surface deflection into the gain computation process, but also the deflection rate. The deflection rate could be limited to an acceptable level where the control surfaces would not cutout. It could also reduce the jerkiness associated with large control surface deflections.

Overall, the control system proved to be a success in eliminating the aircraft's instabilities encountered in the stall regime. It was able to overcome the problems of sluggish control response and slow gain updating and still provide adequate control. Control surface cutouts did detract from the performance of the system but can be remedied with some minor changes in the gain updating scheme.

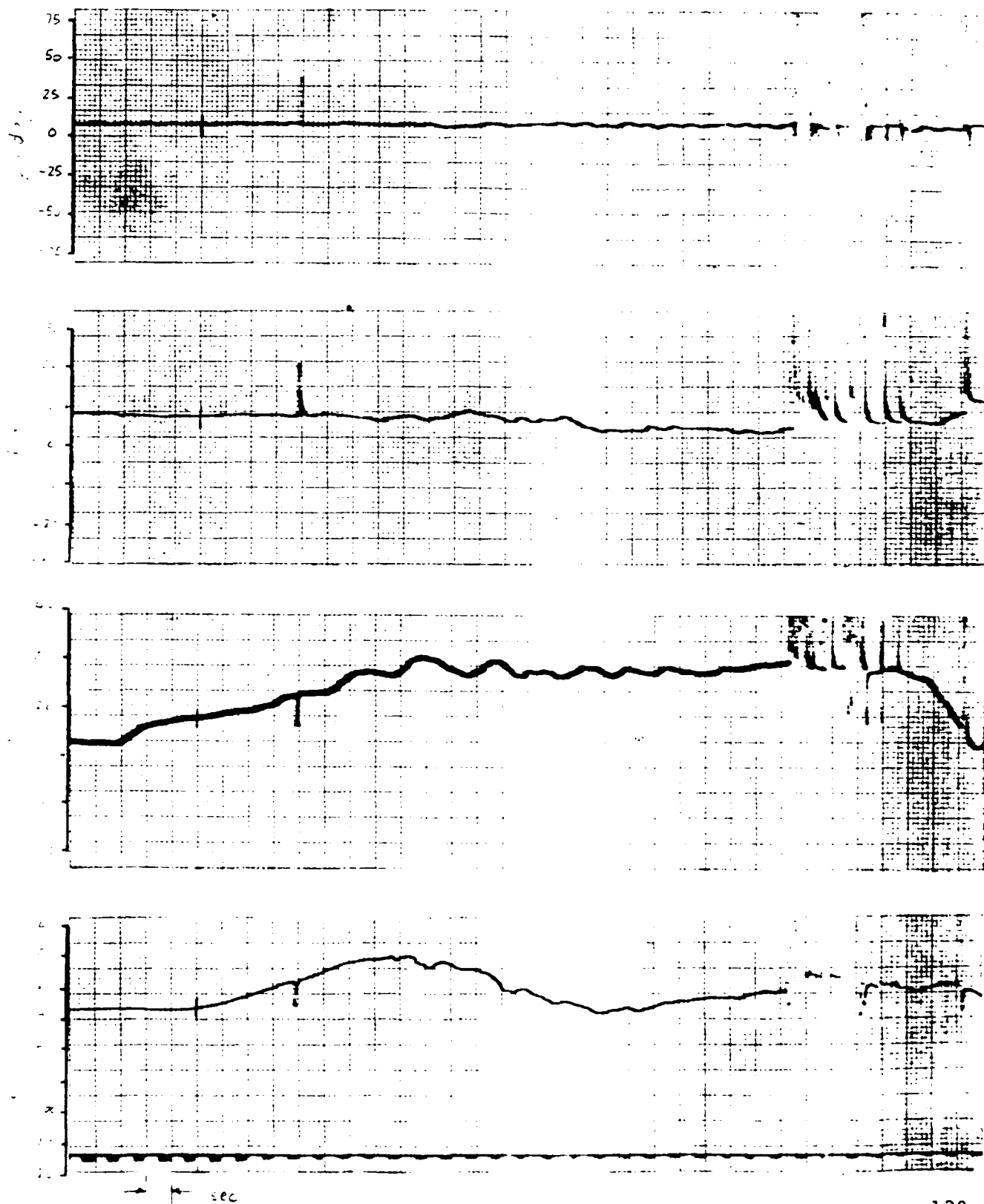


Figure 21. Flight Test Run 3-1 Results.

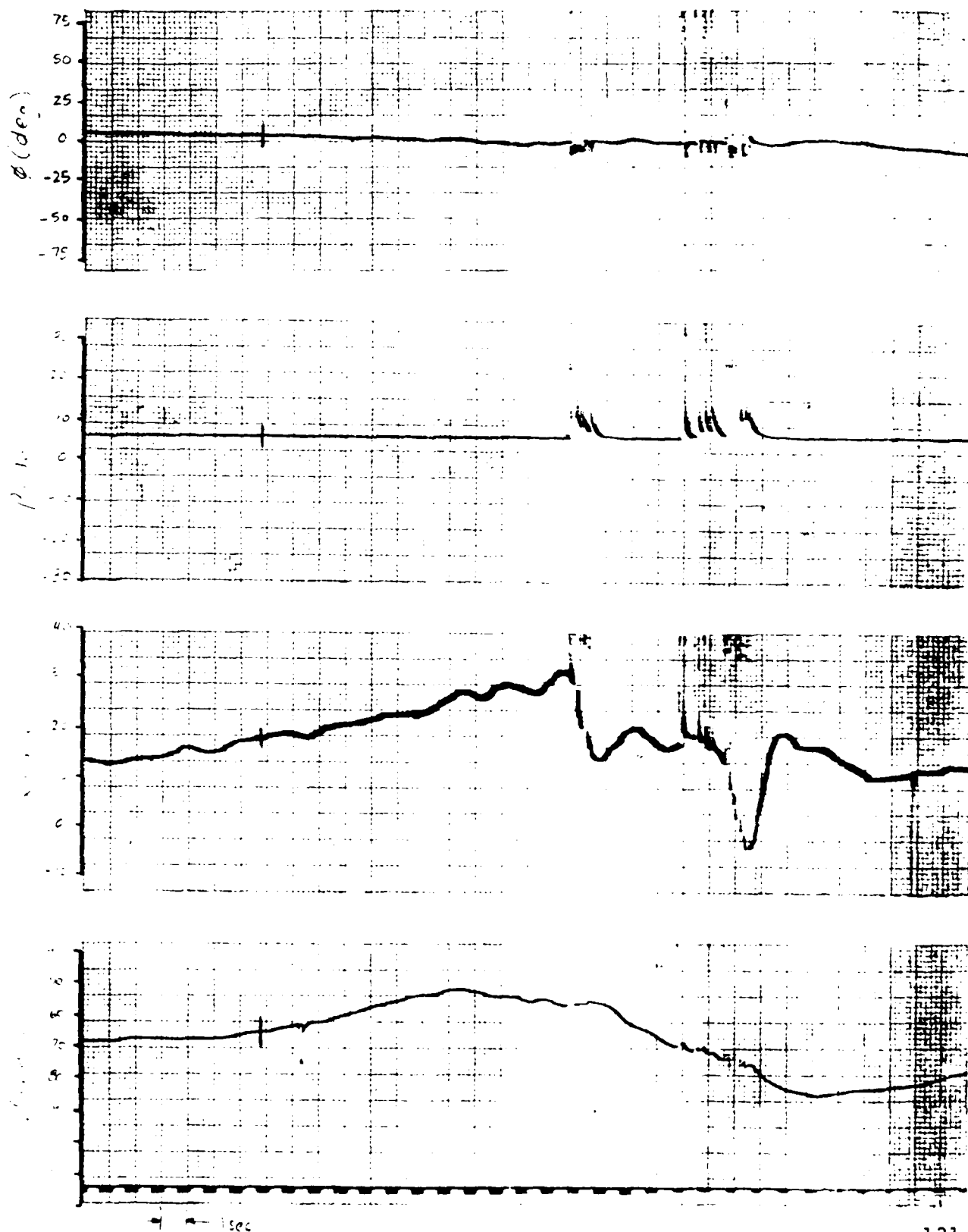


Figure 22. Flight Test Run 3-2 Results.

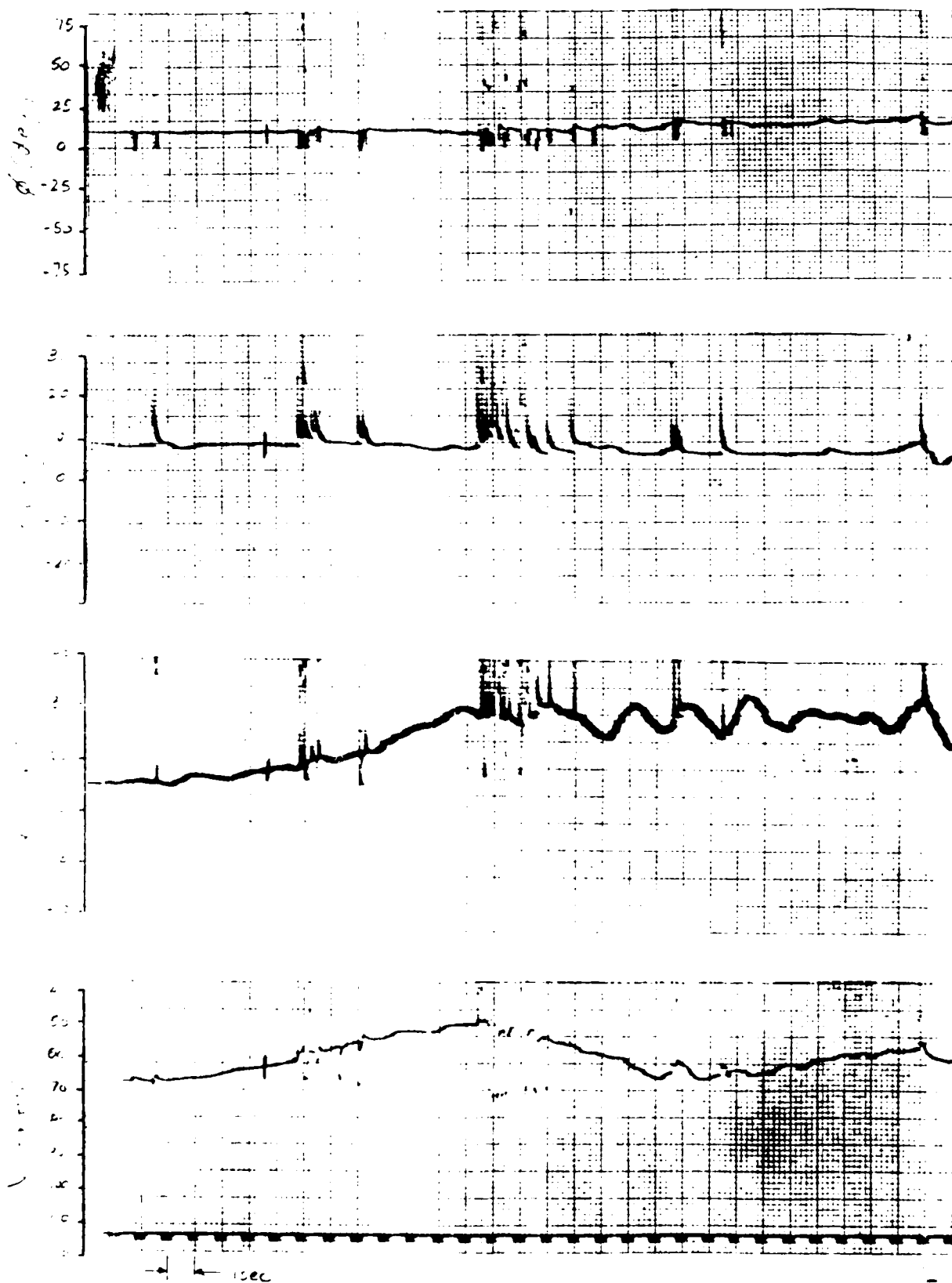
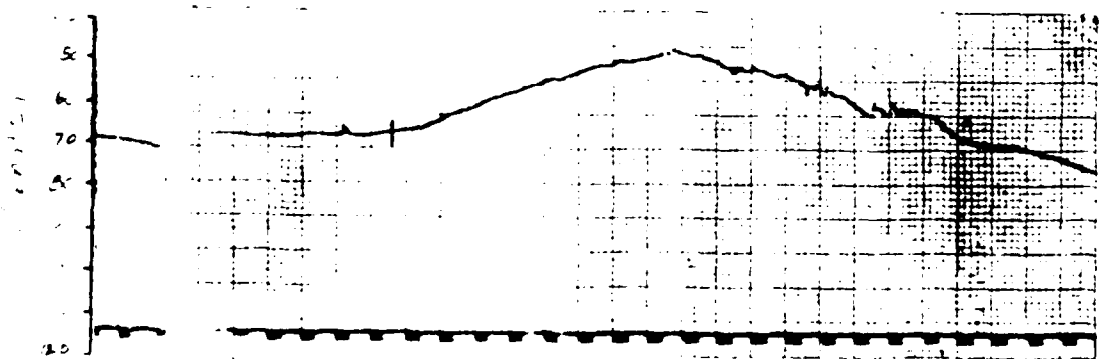
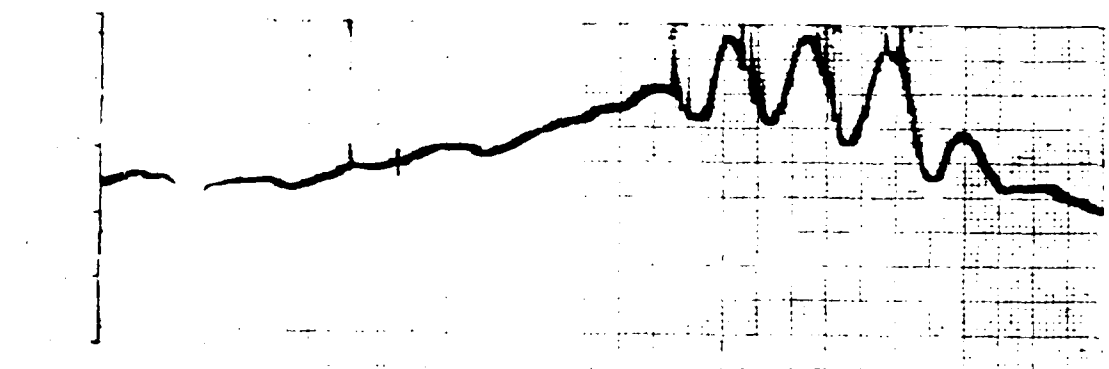
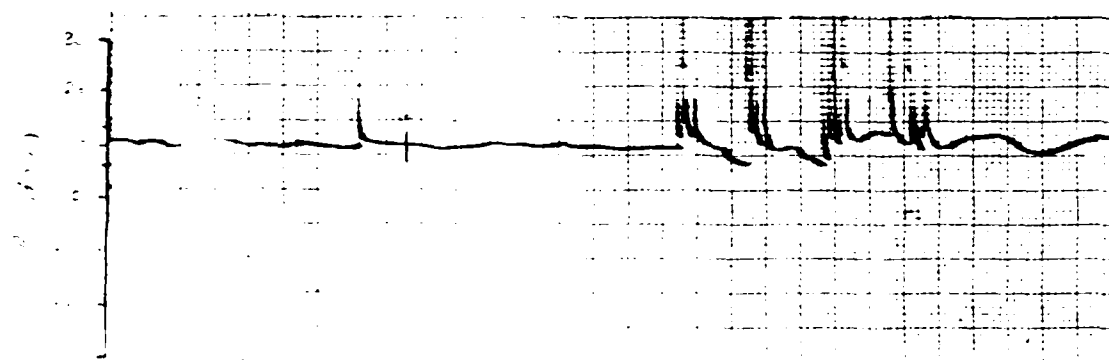
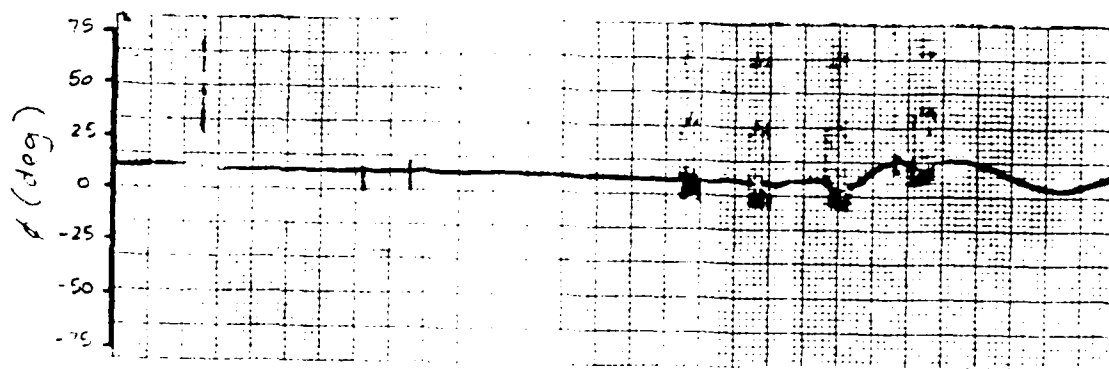


Figure 23. Flight Test Run 3-3 Results.



1 sec

Figure 24. Flight Test Run 3-4 Results.

5.5 ANALYSIS OF RESULTS

To summarize the results, the flight tests were successful at providing an understanding of the aircraft in the stall regime and how to eliminate the instabilities encountered there. The airframe tests pointed out the problems of stalled flight while the pilot tests showed how difficult it was for the pilot to overcome the instabilities experienced in the stall. The CAS tests proved that it is indeed possible to eliminate the instabilities and provide safer and more controlled stall flight characteristics.

The primary purpose of the airframe tests was to get a better understanding of how the aircraft responds in the stall regime. Based on the model, the roll-spiral mode was found to be unstable, which would point to a divergent, oscillatory roll. This was found to be the case in the actual tests. Higher power setting was found to lessen the oscillatory portion of the roll-spiral instability but increase the speed at which the divergences occur. Low power settings caused roll rates as high as 12.9 and 16.2 deg/sec for cases without and with pitch pumping, respectively, while at higher power settings, those same maximum roll rates became 35.5 and 44.4 deg/sec, respectively.

It was also interesting to note that the aircraft always departed controlled flight in the same direction due to the torque effects of the propeller. In this case, the aircraft

always ended up rolling toward the right wing and flying with negative sideslip. These observations point to problems that both the pilot and control system had to overcome.

Finally, it was found that the inclusion of pitch pumping had only a small effect on the stability of the aircraft. The more interesting effect of pitch pumping was how well the pilot and CAS were able to handle the aircraft when pitch pumping was included.

The purpose of the pilot and CAS tests was to determine how well the control system was able to help the pilot control the unstable aircraft. When the results of the tests were compared, the CAS did provide a much larger degree of stability than the pilot in the same situation. Marked reductions in the total variation are found in both roll angle and sideslip.

In the case of low power setting with no pitch pumping, the pilot had a total roll angle variation of 8.1 deg compared to no variation for the control system. For sideslip, the pilot allowed 8.1 deg total variation while the CAS allowed only 5.5 deg. When pitch pumping was included, the pilot had a total roll angle variation of 34.7 deg compared to only 11.3 deg for the CAS. Total pilot sideslip variations, in this case, were 16.8 deg compared to only 6.1 deg for the control system.

The comparisons for the high power setting are even more striking. Without pitch pumping, the pilot allowed a total roll

angle variation of 36.3 deg compared with only 9.7 deg total variation allowed by the control system. Total sideslip variation for the pilot was 22.0, while the CAS had only 6.1 deg total variation. With pitch pumping, the pilot had total roll angle variation of 78.2 deg compared with 12.9 deg for the CAS. The total sideslip variation for the pilot was 46.5 deg compared to 6.5 deg for the control system. Also at the high power settings, the pilot results showed that the variations would have continued to increase had the aircraft not been recovered, while the control system results show no such tendency. It was readily apparent, then, that the control system could be an invaluable aid to the pilot in controlling an otherwise unstable aircraft.

One major problem was discovered with the control system implementation after the tests had been run. The control system was designed around the stability axes of the aircraft. However, the aircraft sensors, from which the aircraft attitude was measured and sent to the computer, are mounted on the aircraft and are not angle-of-attack sensitive. No accommodations were made for this fact within the control system software. Thus the control system utilized stability-axis software with body-axis inputs. To remedy this problem, the sensor inputs should be transformed to stability axes, based on the current angle of attack, prior to the computation of the control surface outputs.

In addition, there were some minor problems noted in the actual flight tests. In particular, the cutout problem previously mentioned needs to be worked out. Some jerkiness also was experienced in the control surface actuations due to the large control surface deflections and the relatively long sampling time. One last problem was that the control system had some trouble keeping up with a rapidly changing flight condition. It is possible that these problems may be the result of the modelling or control law implementation errors. They may also point to some fine tuning required in the control law.

It should be pointed out that regardless of the errors encountered in this project, the control system was still a success and did accomplish its goal. Overall, the flight tests proved that it is possible to provide stability to an otherwise unstable aircraft, even into the stall regime. Such a control system could provide additional safety for pilots, particularly unskilled pilots, when operating in the regions near stall.

Chapter VI

CONCLUSIONS AND RECOMMENDATIONS

The study of aircraft in the high-angle-of-attack regime can provide great rewards in terms of increasing the knowledge of flight and subsequently enhancing aircraft safety. However, that same regime many challenging problems which must be overcome before the rewards can be attained. By putting together a test platform to investigate the different aspects of stalled flight, it would be possible to overcome the problems and make important steps toward improving flight safety.

The purpose of this study was to design and test a control system to eliminate the lateral-directional disturbances from the longitudinal mode in stalled flight. At high angles of attack, the lateral-directional mode becomes unstable and difficult for even an experienced pilot to overcome. By designing a control system to provide stability, even into the stall regime, it would be possible to isolate the longitudinal mode and take steps toward improving aircraft safety.

Perhaps the most challenging part of the problem was that the aircraft could be stalled at an infinite number of flight conditions. Designing a CAS that only provided control for one flight condition would be useless at best. Therefore, the control system was designed for a whole range of flight conditions where the gains were made a function of the flight condition.

Thus, a model of the aircraft had to be computed to represent the aircraft at any flight condition.

The model was assembled using wind tunnel data for the aircraft found in NASA TN D-5758. For those terms for which data were not available, the USAF DATCOM calculation method was used. The data were reduced to a number of equations where the stability, rotary, and control derivatives were calculated from the current flight condition. This method proved to be an effective one. The results of the simulations compare well with actual results of airframe flight tests as explained in Chapter 5. The model then was used to derive control gains for the CAS.

The CAS was designed as a sampled-data, linear-quadratic controller. A number of sample flight conditions were chosen for which suitable control gains were calculated. Simulations showed satisfactory control of the aircraft at each of the sample flight conditions. The gains then were scheduled as functions of the flight condition. These scheduled gains were then tested in simulation and were also found to provide adequate control for each flight condition provided that a minimum correlation was maintained between the actual gains and the scheduled gains.

Finally, operational control system software was formulated for the purpose of flight tests. The flight tests were broken

down into three parts. The first was a test of the basic airframe to get a better understanding of the lateral-directional mode in a stall. The results of the tests pointed out the instabilities that the pilot and CAS were required to overcome. The results also verified the model since the aircraft reacted as predicted from simulation results.

The second part of the flight tests included pilot tests to see how well the pilot could handle the instabilities in a stall. The results were also used as a baseline for comparison with the CAS results. The tests showed that the pilot could maintain some control at lower power settings but that the control was lost at higher power settings.

The final part of the flight tests was the test of the CAS during a stall. The results show that the control system was able to maintain stability well into the stall regime. Deviations from trim were present but were not near the magnitude of those encountered with the pilot in control. Therefore, the study was a success at proving that it is indeed possible to provide stability for an otherwise unstable aircraft, even into the stall regime. The ramifications for this type of CAS point to improved aircraft safety, especially for unskilled pilots, not only in the high-angle-of-attack and stall regimes, but also throughout the whole range of flight conditions the aircraft may encounter.

In reexamining the project, errors were found in the modelling and control law implementation. In particular, the angle-of-attack effects on the inertia matrix and the vertical component of velocity were neglected and resulted in an erroneous model. This, in turn, affected the computation of the control gains. In addition, no provision was made for converting raw sensor data to the stability-axis system in the computer software. These problems would have to be remedied before the control system could be used effectively. It should be noted, however, that even with these errors, the control system did accomplish its goal of stabilizing the aircraft in a stall.

Further work is needed to make some improvements in the control system to make it more workable. In particular, a more efficient gain scheduling scheme would be desirable to reduce the size of the gain matrices while retaining the needed accuracy. This, in turn, would allow a reduction in the sampling time with several added benefits: elimination of cutouts, smoothing of the jerkiness, and better response during rapid changes in the flight condition. A look into the area of proportional filtering might also eliminate large control surface deflections and their associated problems.

In conclusion, then, this study has shown that it is indeed possible to improve lateral-directional stability into the stall regime of the aircraft. Such a find promises to enhance

the study of high-angle-of-attack flight. It also provides a starting point for the design of safety systems into the aircraft to help eliminate instabilities experienced during high-angle-of-attack flight.

Appendix A
AVIONICS RESEARCH AIRCRAFT

A.1 DESCRIPTION OF THE AIRCRAFT

The aircraft used in this study is the Avionics Research Aircraft (ARA), a modified Navion. The ARA is capable of three modes of control -- direct, analog, and digital -- which can operate simultaneously for a particular application. In addition, the aircraft has been equipped with inertial, air data, and navigation sensors. The aircraft has been used in stall-spin research as well as control system design.

The sensors which are available for telemetry and control system usage include angular rate gyros and linear accelerometers for all three axes, angle-of-attack and sideslip vanes, an airspeed sensor, and control position indicators. In addition, barometric altitude, airspeed, air temperature, and engine manifold pressure also are available to the pilot.

The aircraft is flown with two pilots for reasons of flight test efficiency and safety. The evaluation pilot sits in the left seat which is set up for fly-by-wire control system operation for both digital and analog applications. For analog control systems, pilot inputs are routed through potentiometers to the control surfaces. The potentiometers are located on the

display panel and can be adjusted in flight to vary the aircraft's handling characteristics. For digital control systems, a microprocessor is used. The computer itself is located behind the right seat and is interfaced with the pilot through the hand-held CDU. The safety pilot sits in the right seat and has direct control over the control surfaces through the use of mechanical linkages.

A.2 AIRCRAFT DATA

A Navion airframe was tested extensively in the 30' x 60' full-scale wind tunnel at the NASA Research Center, Hampton, Virginia and the results of those tests were compiled in the report, NASA TN D-5857. The availability of the data simplified the model development process. To use it, the data were reduced to a set of tables of data points from the nondimensional coefficient curves. The tables are presented here and are arranged as follows: Table 8, constant component data; Table 9, stability derivative component data; Table 10, longitudinal coefficient data; Table 11, rudder derivative component data; and Table 12, aileron derivative component data.

In addition, a number of aircraft constants were required for the rotary derivative component development. Those constants are presented in Table 13.

TABLE 8

Constant Component Data

$a \quad T_c$	0.03	0.12	0.23
-4.0	0.001	0.002	0.007
0.0	0.0	0.0	0.0
4.0	0.0	-0.004	-0.007
8.0	-0.002	-0.008	-0.016
12.0	-0.005	-0.015	-0.028
16.0	-0.019	-0.025	-0.044
20.0	-0.018	-0.040	-0.065
24.0	-0.020	-0.045	-0.090

 C_y

$a \quad T_c$	0.03	0.12	0.23
-4.0	0.0010	0.0	-0.0040
0.0	0.0005	-0.0015	-0.0000
4.0	-0.0005	-0.0030	-0.0080
8.0	-0.0010	-0.0035	-0.0090
12.0	-0.0015	-0.0020	-0.0090
16.0	0.0020	0.0	-0.0050
20.0	0.0050	0.0015	0.0030
24.0	0.0020	0.0070	0.0100

 C_n

$a \quad T_c$	0.03	0.12	0.23
-4.0	0.0110	0.0090	0.0020
0.0	0.0090	0.0065	0.0060
4.0	0.0090	0.0055	0.0040
8.0	0.0100	0.0050	0.0020
12.0	0.0070	0.0050	0.0025
16.0	-0.0030	0.0030	0.0030
20.0	0.0090	0.0065	-0.0025
24.0	0.0120	0.0020	0.0100

 C_1

TABLE 9
Stability Derivative Component Data

(throttle setting = 0.03)

α E	-15.0	-10.0	-5.0	0.0	5.0	10.0	15.0
-4.0	0.182	0.134	0.075	0.001	-.066	-.148	-.210
0.0	0.177	0.131	0.069	0.0	-.070	-.145	-.200
4.0	0.170	0.110	0.062	0.0	-.066	-.142	-.202
8.0	0.160	0.100	0.055	-.002	-.060	-.135	-.197
12.0	0.143	0.085	0.044	-.005	-.050	-.110	-.185
16.0	0.121	0.075	0.030	-.019	-.063	-.104	-.171
20.0	0.102	0.050	0.010	-.018	-.054	-.097	-.155
24.0	0.084	0.050	0.005	-.020	-.050	-.097	-.130

C_y

α E	-15.0	-10.0	-5.0	0.0	5.0	10.0	15.0
-4.0	-.0275	-.0210	-.0120	0.0010	0.0135	0.0265	0.0365
0.0	-.0200	-.0200	-.0115	0.0005	0.0130	0.0220	0.0325
4.0	-.0250	-.0180	-.0110	-.0005	0.0105	0.0175	0.0290
8.0	-.0230	-.0160	-.0105	-.0010	0.0075	0.0140	0.0275
12.0	-.0210	-.0135	-.0095	-.0015	0.0030	0.0120	0.0220
16.0	-.0105	-.0110	-.0085	0.0020	0.0040	0.0120	0.0195
20.0	-.0185	-.0080	-.0010	0.0050	0.0105	0.0135	0.0240
24.0	-.0235	-.0150	-.0020	0.0020	0.0100	0.0105	0.0100

C_n

α E	-15.0	-10.0	-5.0	0.0	5.0	10.0	15.0
-4.0	0.0330	0.0280	0.0210	0.0110	-.0050	-.0110	-.0170
0.0	0.0330	0.0240	0.0190	0.0090	-.0025	-.0080	-.0120
4.0	0.0270	0.0225	0.0170	0.0090	-.0040	-.0100	-.0115
8.0	0.0220	0.0220	0.0160	0.0100	-.0070	-.0140	-.0150
12.0	0.0250	0.0230	0.0150	0.0070	-.0120	-.0200	-.0215
16.0	0.0300	0.0190	0.0140	-.0030	-.0175	-.0200	-.0290
20.0	0.0440	0.0370	0.0330	0.0090	-.0055	-.0275	-.0325
24.0	0.0500	0.0430	0.0200	0.0120	-.0000	-.0320	-.0300

C_l

Table 9

continued

(throttle setting = 0.23)

α $^{\circ}$	-15.0	-10.0	-5.0	0.0	5.0	10.0	15.0
-4.0	0.243	0.180	0.106	0.008	-0.080	-0.160	-0.237
0.0	0.230	0.168	0.095	0.0	-0.090	-0.167	-0.246
4.0	0.213	0.155	0.082	-0.008	-0.099	-0.175	-0.252
8.0	0.197	0.142	0.068	-0.016	-0.104	-0.185	-0.256
12.0	0.180	0.127	0.053	-0.029	-0.104	-0.190	-0.257
16.0	0.158	0.104	0.037	-0.043	-0.096	-0.162	-0.250
20.0	0.133	0.070	0.005	-0.065	-0.110	-0.158	-0.221
24.0	0.095	0.004	-0.050	-0.100	-0.108	-0.128	-0.228

 C_y

α $^{\circ}$	-15.0	-10.0	-5.0	0.0	5.0	10.0	15.0
-4.0	-0.0320	-0.0250	-0.0190	-0.0040	0.0100	0.0200	0.0330
0.0	-0.0320	-0.0275	-0.0205	-0.0060	0.0075	0.0190	0.0305
4.0	-0.0320	-0.0280	-0.0220	-0.0075	0.0055	0.0175	0.0285
8.0	-0.0315	-0.0270	-0.0235	-0.0090	0.0035	0.0160	0.0265
12.0	-0.0295	-0.0240	-0.0240	-0.0090	0.0015	0.0140	0.0250
16.0	-0.0260	-0.0200	-0.0220	-0.0050	0.0	0.0105	0.0240
20.0	-0.0215	-0.0130	-0.0110	0.0030	0.0040	0.0140	0.0225
24.0	-0.0160	-0.0015	0.0080	0.0140	0.0105	0.0180	0.0215

 C_n

α $^{\circ}$	-15.0	-10.0	-5.0	0.0	5.0	10.0	15.0
-4.0	0.0385	0.0290	0.0260	0.0020	-0.0130	-0.0130	-0.0170
0.0	0.0330	0.0205	0.0215	0.0000	-0.0100	-0.0120	-0.0130
4.0	0.0280	0.0230	0.0175	0.0040	-0.0085	-0.0115	-0.0130
8.0	0.0245	0.0190	0.0130	0.0020	-0.0080	-0.0130	-0.0160
12.0	0.0220	0.0170	0.0085	0.0025	-0.0095	-0.0160	-0.0200
16.0	0.0260	0.0195	0.0120	0.0030	-0.0140	-0.0235	-0.0240
20.0	0.0350	0.0220	0.0145	-0.0025	-0.0210	-0.0240	-0.0390
24.0	0.0400	0.0265	0.0220	0.0000	-0.0250	-0.0160	-0.0240

 C_l

TABLE 10
Longitudinal Coefficient Data

α	τ_c	0.03	0.12	0.23
-4.0		-.180	-.180	-.140
0.0		0.140	0.150	0.210
4.0		0.455	0.480	0.550
8.0		0.750	0.800	0.895
12.0		1.040	1.120	1.240
16.0		1.270	1.365	1.525
20.0		1.300	1.500	1.550
24.0		1.370	1.580	1.620

C_L

α	τ_c	0.03	0.12	0.23
-4.0		0.015	-.070	-.190
0.0		0.020	-.065	-.190
4.0		0.025	-.060	-.180
8.0		0.040	-.040	-.160
12.0		0.070	0.0	-.115
16.0		0.130	0.055	-.060
20.0		0.270	0.140	0.075
24.0		0.375	0.420	0.280

C_L

TABLE 11
Rudder Derivative Component Data

(throttle setting = 0.03)

α deg	-17.5	-9.0	0.0	7.0	13.2
-4.0	-.050	-.020	0.002	0.018	0.044
0.0	-.054	-.020	0.0	0.018	0.050
4.0	-.058	-.024	-.005	0.014	0.050
8.0	-.061	-.027	-.006	0.011	0.047
12.0	-.063	-.031	-.006	0.008	0.040
16.0	-.070	-.036	-.017	0.003	0.030
20.0	-.077	-.050	-.015	-.004	0.015
24.0	-.060	-.028	-.015	0.002	0.047

C_Y

α deg	-17.5	-9.0	0.0	7.0	13.2
-4.0	0.0280	0.0130	0.0010	-.0000	-.0210
0.0	0.0300	0.0125	0.0	-.0080	-.0255
4.0	0.0315	0.0125	-.0005	-.0095	-.0280
8.0	0.0320	0.0130	-.0010	-.0100	-.0290
12.0	0.0300	0.0145	-.0010	-.0085	-.0250
16.0	0.0390	0.0140	0.0020	-.0060	-.0200
20.0	0.0350	0.0195	0.0055	-.0015	-.0150
24.0	0.0250	0.0090	0.0015	-.0060	-.0150

C_N

α deg	-17.5	-9.0	0.0	7.0	13.2
-4.0	0.0010	0.0070	0.0110	0.0080	0.0135
0.0	0.0065	0.0040	0.0085	0.0115	0.0145
4.0	0.0065	0.0075	0.0095	0.0135	0.0155
8.0	0.0100	0.0080	0.0100	0.0090	0.0150
12.0	0.0020	0.0065	0.0075	0.0080	0.0060
16.0	0.0035	0.0040	-.0035	-.0015	-.0040
20.0	0.0175	0.0090	0.0090	0.0080	0.0045
24.0	0.0155	0.0120	0.0115	0.0135	0.0055

C_L

Table 11

continued

(throttle setting = 0.23)

a dk	-17.5	-9.0	0.0	7.0	13.2
-4.0	-.063	-.018	0.000	0.033	0.070
0.0	-.084	-.033	0.0	0.027	0.001
4.0	-.100	-.040	-.006	0.020	0.055
8.0	-.115	-.058	-.018	0.011	0.050
12.0	-.120	-.068	-.030	0.003	0.040
16.0	-.140	-.074	-.044	0.003	0.044
20.0	-.130	-.100	-.065	-.031	0.015
24.0	-.140	-.080	-.090	-.080	-.080

 C_y

a dk	-17.5	-9.0	0.0	7.0	13.2
-4.0	0.0365	0.0115	-.0040	-.0160	-.0375
0.0	0.0400	0.0110	-.0000	-.0210	-.0430
4.0	0.0425	0.0115	-.0070	-.0230	-.0470
8.0	0.0445	0.0120	-.0080	-.0250	-.0500
12.0	0.0440	0.0130	-.0080	-.0255	-.0505
16.0	0.0475	0.0100	-.0055	-.0200	-.0495
20.0	0.0420	0.0230	0.0040	-.0150	-.0370
24.0	0.0395	0.0170	0.0100	0.0025	-.0170

 C_n

a dk	-17.5	-9.0	0.0	7.0	13.2
-4.0	-.0040	0.0020	0.0020	0.0125	0.0230
0.0	-.0010	-.0010	0.0000	0.0090	0.0190
4.0	0.0	-.0015	0.0035	0.0050	0.0190
8.0	0.0020	0.0	0.0020	0.0055	0.0140
12.0	0.0055	0.0065	0.0010	0.0060	0.0070
16.0	0.0085	0.0075	0.0020	0.0015	-.0030
20.0	0.0140	0.0010	-.0025	-.0015	-.0055
24.0	0.0055	0.0085	0.0085	-.0030	-.0060

 C_l

TABLE 12
Aileron Derivative Component Data

α $d\delta$	-42.0	-21.0	0.0	21.0	42.0
-4.0	0.015	0.012	0.005	0.0	-.010
0.0	0.012	0.010	0.0	-.002	-.010
4.0	0.010	0.010	0.0	-.005	-.011
8.0	0.008	0.008	-.005	-.010	-.015
12.0	0.007	0.018	-.008	-.015	-.024
16.0	-.007	0.0	-.015	-.027	-.040
20.0	0.005	-.018	-.018	-.022	-.055
24.0	-.015	-.015	-.015	-.022	0.020

C_Y

α $d\delta$	-42.0	-21.0	0.0	21.0	42.0
-4.0	0.0050	0.0010	0.0010	0.0	-.0020
0.0	0.0010	0.0	0.0	0.0100	0.0
4.0	-.0025	-.0030	0.0	0.0015	0.0025
8.0	-.0005	-.0000	-.0005	0.0030	0.0055
12.0	-.0110	-.0130	-.0010	0.0005	0.0105
16.0	-.0095	-.0110	0.0020	0.0110	0.0200
20.0	-.0155	-.0020	0.0050	0.0145	0.0285
24.0	-.0100	-.0080	0.0020	0.0130	0.0150

C_n

α $d\delta$	-42.0	-21.0	0.0	21.0	42.0
-4.0	0.0000	0.0400	0.0105	-.0170	-.0450
0.0	0.0555	0.0390	0.0090	-.0150	-.0470
4.0	0.0645	0.0385	0.0100	-.0140	-.0480
8.0	0.0645	0.0425	0.0100	-.0150	-.0480
12.0	0.0630	0.0320	0.0070	-.0180	-.0515
16.0	0.0575	0.0320	-.0030	-.0290	-.0605
20.0	0.0730	0.0390	0.0100	-.0120	-.0535
24.0	0.0620	0.0540	0.0130	-.0090	-.0200

C_l

TABLE 15
Aircraft Constants

Wing Data		
Aspect Ratio	AR	6.04
Area	S_w, S	184 feet ²
Span	b_w	33.38 feet
Chord	c_w	5.7 feet
Centerline distance from c.g.	z_w	1 foot
Taper Ratio	λ_w	0.54
Dihedral	T_w	7.5°
Sweep (at quarter chord)	Λ_w	3.0°

Horizontal Tail Data		
Aspect Ratio	AR_h	4.0
Area	S_h	43 feet ²
Span	b_h	13.11 feet
Chord	c_h	3.28 feet
Centerline distance from c.g.	z_h	0.0 feet
Taper ratio	λ_h	0.67
Sweep (at quarter chord)	Λ_h	6.0°

Vertical Tail Data		
Aspect Ratio	AR_v	2.0
Area	S_v	12.5 feet ²
Span	b_v	5.0 feet
Chord	c_v	3.52 feet
Taper Ratio	λ_v	0.557
Sweep (at quarter chord)	Λ_v	20.0°

Table 13

continued

Mass and Inertia Data

Gross Weight	m	2940 lbs.
Center of Gravity (c.g.)	x/c_w	0.25
Moment of Inertia (x axis)	I_x	1284.1 slug-ft ²
Moment of Inertia (y axis)	I_y	2772.9 slug-ft ²
Moment of Inertia (z axis)	I_z	3234.7 slug-ft ²

Miscellaneous Data

Distance from Moment Center to Vertical Tail c.p. (normal to body ℓ)	z_P	2.87 feet
Distance from Moment Center to Vertical Tail c.p. (parallel to body ℓ)	l_P	17.29 feet
Distance of Vertical Tail Center of Pressure Above or Below Moment Center	z	$z_P \cos(\alpha) + l_P \sin(\alpha)$

Appendix B

GAIN COMPUTATION SOFTWARE

The gain computations were done by coding, in FORTRAN, the model and control law development steps into the program, **CONTRL**. **CONTRL** had the following capabilities: compute the linearized system equations given the flight condition; compute the optimal gains given a set of state control weightings; compute the equivalent closed-loop system equations; and perform a linear and/or nonlinear simulation. A flow chart of **CONTRL** is presented in Figure 13 .

In addition, a number of subroutines were included in the main program to perform intermediate tasks. **LDDYN** computed the lateral-directional dynamic equations given the flight condition and the current value of the states and controls. **FANDG** computed the linearized system and output matrices given the flight condition and the trim values of the states and controls. **STM** computed the state transition matrix given the sample time and the system dynamic matrix. **CEM** computed the control effects matrix given the state transition matrix and the system matrices. **CLOOPE** computed the equivalent closed-loop system dynamics matrix given the closed-loop state transition matrix. **RKINT** performed a nonlinear simulation using a 4-th order Runge-Kutta integration given the flight condition, the control gain matrices, and the

initial condition. Table 14 presents a listing of each of these subroutines as well as the main program, CONTRL.

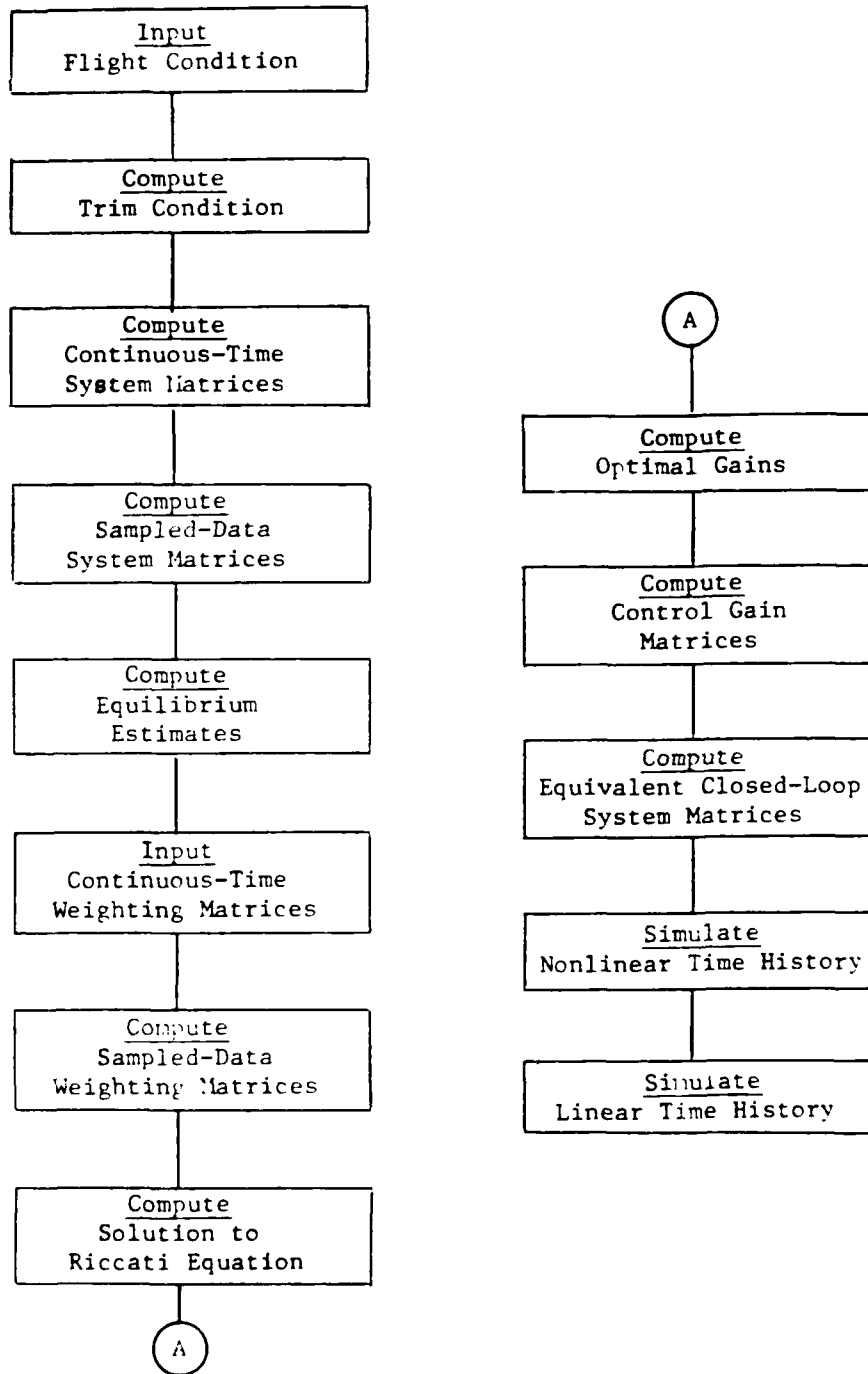


Figure 25: Gain Computation Flowchart

Table 14. Gain Computation Listings

FILE: CONTROL FORTRAN A

PRINCETON UNIVERSITY TIME-SHARING SYSTEM

```

DIMENSION WKF(4), PHI(4,4), F(4,4), FCOFY(4,4), QC(4,4), FINV(4,4),
1      G(4,2), GAMMA(4,2), HX(2,4), HU(2,2), RC(2,2), AREAP(3,3),
2      XO(4), U(2), TEMP1(4,4), TEMP2(2,4), TEMP3(2,2), TEMP4(4,2),
3      TEMP5(4,4), MHAT(4,2), MHATOT(4,2), OHAT(4,4), OHATOT(4,4),
4      Q(4,4), RHAT(2,2), RHATOT(2,2), R(2,2), OTNOW(4,4),
5      MTNOW(4,2), RTNOW(2,2), SINV(5,5), S(5,5), AREAS(4,4),
6      PHIT(4,4), GAMMAT(2,4), MHATT(2,4), PHITEM(4,4),
7      GAMTEM(4,2), PK(4,4), PKMNS1(4,4), APEAR(1,1), XT1(4),
8      C(2,4), S11(3,3), S12(3,2), S21(2,3), S22(2,2), C1(2,3),
9      C2(2,1), CP(2,2), CB(2,4), P1(3,3), P12(3,1), P2(1,1)
DIMENSION F21(1,3), G1(3,2), G2(1,2), TEMP6(2,1), CS(2,1),
1      TEMP7(1,2), TEMP8(1,2), C1(2,2), PHICL(4,4), PCL(4,4),
2      DY(2,1), DYT(2,1), X(4,1), U2(2,1), TEMP10(4,1),
3      TEMP11(2,1), PCLCFY(4,4), XGUT(4), U2OUT(2), TEMP12(1,1),
4      X1STAR(3,1), USTAR(2,1), XUSTAR(5,1), CONVEC(5,1),
5      X2STAR(1,1), TEMP9(3,1), PCHECK(4,4), IDEN(4,4),
6      KY(1,2), SXI(3,1), TEMP13(3,1), SAXY(3,2), SUY(2,2),
7      HX1(2,3), HY2(2,1), SHI(2,1), FC(8,1), A1(7,9), A21(7,9),
8      CFEEED1(7,1), CFEEED2(7,1), PC1(9,1), PC2(18,1), A22(7,9),
9      T1(4,4), T2(4,4), T3(4,4), CTRIM(3,3), CVEC(3,1), AREACT(2,2)
DIMENSION XI(4), UT(4), A2(7,18)
REAL IDEN
COMPLEX WHIGEN(4), ZFIGEN(4,4)
REAL INTVL
COMMON/CYCOFF/CY, CYO, CYB, CYDR, CYDA, CYP, CYR
COMMON/CNCOFF/CN, CNO, CNB, CNDR, CNDA, CNP, CNR
COMMON/CLCOFF/CL, CLC, CLB, CLDR, CLDA, CLP, CLR
DATA A1/1.79, .7297, 2.155, 7.889, 3.254, -3.595, .5348, -.06704,
1  -.02105, -.1052, -.3227, -.08524, .1556, -.006276, .8409E-3, .2326E-3,
2  .00148, .004372, .908E-3, -.002146, -.7038E-4, -.007062, -.02591,
3  -.01192, .005733, -.02486, .009345, -.1531, -.2718E-3, .8608E-3,
4  .5902E-3, .432E-3, -.001404, -.6244E-3, .004123, .5783E-5, -.1064E-4,
5  -.827E-5, -.7653E-5, .274E-4, .9687E-5, -.4765E-4, .001574, .7078E-3,
6  .001894, -.001963, .001056, .0005536, .003235, -.2029E-4, -.3058E-4,
7  -.7971E-4, .3013E-4, .1207E-3, -.5157E-5, -.148E-3, .5743E-7,
8  .4931E-5, .1182E-5, .1268E-6, -.2462E-5, -.9082E-7, .2178E-5/
DATA A2/1.946, -.05978, -.1867, -.9162, -.4223, -.658, -8.224,
1  -.2274, -.1526, -.6572, -3.385, .943, 1.412, .6892,
2  .06903, .001114, .005632, -.01312, .02269, .03657, .3341,
3  .1142, .000144, .03218, .155, -.04901, -.06353, .001572, -.6239E-3,
4  -.1153E-4, -.9239E-4, .3222E-3, -.333F-3, -.501E-3, -.004504,
5  -.002183, -.1301E-3, -.4252E-3, -.002074, .7071E-3, .8178E-3, -.1901E-3,
6  -.03305, .008944, .09175, .1234, -.06199, -.08771, .0363,
7  .1234, -.245E-3, -.03717, .3041, .1051, .0682, -.05388,
8  .8762E-3, -.395E-3, -.001522, -.003296, .001981, .001415, -.002292/
DATA A22/.003239, -.7989E-4, -.001522, -.006569, -.003469, .8433E-4,
1  .001104, -.1553E-4, .6372E-5, .2116E-4, .4196E-4, -.2631E-4, -.2001E-4,
2  .3346E-4, -.7741E-4, .1957E-5, .2149E-4, .4993E-4, .4469E-4, -.2753E-5,
3  .8684E-5, .3322E-3, .4981E-3, -.00384, -.006807, .002309, .002882,
4  .001139, -.00453, -.1815E-3, .00266, .001183, -.00517, -.003604, .002541,
5  -.9593E-4, -.3247E-4, -.3302E-4, .1429E-3, -.4179E-4, .7385E-4,
6  .266E-5, -.1424E-3, .5954E-5, .5423E-4, .3287E-4, .1639E-3, -.1119E-4,
7  -.7E-4, .1853E-5, .353E-6, .3215E-6, -.23E-5, .5309E-6, -.8737E-6,
8  -.7202E-6, .3768E-5, .498E-7, .2687E-7, .9122E-6, -.2211E-5, -.5247E-6,
9  .1551E-5/

```



```

800  FORMAT('-', 'INPUT: FLIGHT CCNDITION (A,IC,VEL,ALT)')
801  FORMAT('-', 'PHI EQUALS:', 4F10.3, 'GAMMA EQUALS:')
802  FORMAT('0', 4F10.3, 13X, 2F10.3)
811  FORMAT('-', 'S EQUALS:')
812  FORMAT('0', 5F10.3)
829  FORMAT('-', 'STEADY STATE ESTIMATE')
860  FORMAT('-', 'OPEN LOOP? (1=YES,0=NO)')
863  FORMAT('-', 'INPUT: Q(SIDESLIP),Q(ROLL ANGLE)')
804  FORMAT('-', 'INPUT: PC DIAGONAL ELEMENTS (2)')
805  FORMAT('-', 'Q(CONTINUOUS):', 31X, 'Q(DISCRETE):')
806  FORMAT('0', 4F8.3, 12X, 4F8.3)
807  FORMAT('-', 'M(DISCRETE):')
808  FORMAT('0', 2F8.3)
809  FORMAT('-', 'R(CONTINUOUS):', 7X, 'R(DISCRETE):')
810  FORMAT('0', 2F8.3, 4X, 2F8.3)
813  FORMAT('-', 'P DID NOT CONVERGE IN ', I4, ' STEPS')
814  FORMAT('-', 'P(STEADY STATE) EQUALS:')
815  FORMAT('0', 4F10.3)
816  FORMAT('-', 'OPTIMAL GAINS (C) EQUAL:')
817  FORMAT('0', 4F10.3)
818  FORMAT('-', 'CB EQUALS:', 27X, 'CF EQUALS:', 11X, 'CI EQUALS:')
819  FORMAT('0', 4F8.3, 5X, 2F8.3, 5X, 2F8.3)
861  FORMAT('-', 'INPUT: SECONDS FOR TIME HISTORY')
820  FORMAT('-', 'DO YOU WISH TO RUN ANOTHER DELY VECTOR?',
1      ' (1=YES,0=NO)')
821  FORMAT('-', 'INPUT: DELY VECTOR (DEGREES) - (Y1,Y2)')
822  FORMAT('1', 'X(TIME) EQUALS: (', 4F10.3, ') DEG')
823  FORMAT('0', 'U(TIME) EQUALS: (', 2F10.3, ') DEG')
825  FORMAT(' ', 'TIME', 4X, 'YAW RATE', 3X, 'SIDESLIP', 3X, 'ROLL RATE',
1      1X, 'ROLL ANGLE', 3X, 'RUDDER', 5X, 'AILERON')
826  FORMAT(' ', '-----', 4X, '-----', 3X, '-----', 3X, '-----',
1      1X, '-----', 3X, '-----', 5X, '-----')
850  FORMAT('0', 'ALL OUTPUT IN DEG OR DEG/SEC')
851  FORMAT('-', 'STATE TRANSITION MATRIX TIME HISTORY')
827  FORMAT(' ', P5.2, 6(3X, P8.3))
830  FORMAT('-', 'DIFFERENT C MATRIX? (1=YES,0=NO)')
831  FORMAT('-', 'NONLINEAR TIME HISTORY? (1=YES,0=NO)')
832  FORMAT('-', 'LINEAR TIME HISTORY? (1=YES,0=NO)')
828  FORMAT('-', 'ANOTHER CASE? (1=YES,0=NO)')
870  FORMAT('-', 'FLIGHT CONDITION MATRIX EQUALS:')
871  FORMAT(' ', 8(F10.3, 1X))
872  FORMAT('-', 'SCHEDULE GAINS? (1=YES,0=NO)')
873  FORMAT('-', 'SCHEDULED GAINS:')
C
C 100-----INITIALIZATION OF MODEL
C
C      INCLUDES: CREATION OF F,G,HX,AND HU MATRICES.
C      ALSO COMPUTATION OF SYSTEM EIGENVALUES
C
C
C INPUT FLIGHT CONDITION:  ANGLE OF ATTACK
C                          THRUSTLE SETTING
C                          VELOCITY
C                          ALTITUDE
C

```

```

CON00560
CON00570
CON00580
CON00590
CON00600
CON00610
CON00620
CON00630
CON00640
CON00650
CON00660
CON00670
CON00680
CON00690
CON00700
CON00710
CON00720
CON00730
CON00740
CON00750
CON00760
CON00770
CON00780
CON00790
CON00800
CON00810
CON00820
CON00830
CON00840
CON00850
CON00860
CON00870
CON00880
CON00890
CON00900
CON00910
CON00920
CON00930
CON00940
CON00950
CON00960
CON00970
CON00980
CON00990
CON01000
CON01010
CON01020
CON01030
CON01040
CON01050
CON01060
CON01070
CON01080
CON01090
CON01100

```

900	WRITE (6,800)	CON01110
	READ (6,*) A, TC, VEL, ALT	CON01120
C		CON01130
C	CREATE FLIGHT CONDITION MATRICES	CON01140
C		CON01150
	RHO=.002377*EXP(-ALT/25000.)	CON01160
	DYNPRS=.5*FHO*VEL**2	CON01170
	FC1(1,1)=EXP(-.001*TC)	CON01180
	FC1(2,1)=FC1(1,1)*DYNPRS	CON01190
	FC1(3,1)=FC1(2,1)*DYNPRS	CON01200
	DO 10 I=1,6	CON01210
10	FC1(I+3,1)=FC1(I,1)*A	CON01220
	FC2(1,1)=1.	CON01230
	FC2(2,1)=TC	CON01240
	DO 15 I=1,4	CON01250
15	FC2(I+2,1)=FC2(I,1)*DYNPRS	CON01260
	DO 20 I=1,12	CON01270
20	FC2(I+6,1)=FC2(I,1)*A	CON01280
C		CON01290
C	INITIALIZE STATE AND CONTROL TO IRIM	CON01300
C		CON01310
	DO 100 I=1,4	CON01320
100	X0(I)=0.	CON01330
	DO 102 I=1,2	CON01340
102	U(I)=0.	CON01350
	CALL DNDYN(A,TC,VEL,ALT,X0,U,DX)	CON01360
	CTRIM(1,1)=CYB	CON01370
	CTRIM(1,2)=CYDB	CON01380
	CTRIM(1,3)=CYDA	CON01390
	CTRIM(2,1)=CNS	CON01400
	CTRIM(2,2)=CNDB	CON01410
	CTRIM(2,3)=CNDA	CON01420
	CTRIM(3,1)=CLB	CON01430
	CTRIM(3,2)=CLDB	CON01440
	CTRIM(3,3)=CLDA	CON01450
	CVFC(1,1)=CYS	CON01460
	CVFC(2,1)=CNS	CON01470
	CVFC(3,1)=CLB	CON01480
	CNVPSN=100./3.14159	CON01490
	CALL SMPLY(3,CTRIM,3,CNVPSN,CTRIM)	CON01500
	CALL INVPS(3,CTRIM,2,APFACT,CTRIM)	CON01510
	CALL SMPLY(3,CVFC,1,-1.,CVFC)	CON01520
	CALL SMPLY(3,CTRIM,3,CVEC,1,CVEC)	CON01530
	XT(1)=0.	CON01540
	XT(2)=CVFC(1,1)	CON01550
	XT(3)=0.	CON01560
	XT(4)=0.	CON01570
	UT(1)=CVFC(2,1)	CON01580
	UT(2)=CVFC(3,1)	CON01590
	DO 110 I=1,4	CON01600
110	XOUT(I)=XT(I)*CNVPSN	CON01610
	DO 112 I=1,2	CON01620
112	U2OUT(I)=UT(I)*CNVPSN	CON01630
	WRITE (6,822) (XOUT(I),I=1,4)	CON01640
	WRITE (6,823) (U2OUT(I),I=1,2)	CON01650

C		CON01660
C	COMPUTE F,G,HX,AND HU MATRICES AND SYSTEM EIGENVALUES	CON01670
C		CON01680
	CALL FANDG(A,TC,VBL,ALT,XT,UT,F,G,HX,HU,1)	CON01690
	CALL EIGEN(F,FCOPY,4,0,WEIGEN,ZEIGEN,WKF)	CON01700
C		CON01710
C	PARTITION F, G, AND HX FOR STEADY STATE RESPONSE AND COMPUTATION	CON01720
C	OF GAINS	CON01730
C		CON01740
C		CON01750
C	F	CON01760
C		CON01770
	DO 150 I=1,3	CON01780
	DO 152 J=1,3	CON01790
152	F1(I,J)=F(I,J)	CON01800
150	CONTINUE	CON01810
	DO 154 I=1,3	CON01820
	F12(I,1)=F(I,4)	CON01830
154	F21(1,I)=F(4,I)	CON01840
	F2(1,1)=F(4,4)	CON01850
C		CON01860
C	G	CON01870
C		CON01880
	DO 156 I=1,3	CON01890
	DO 158 J=1,2	CON01900
158	G1(I,J)=G(I,J)	CON01910
156	CONTINUE	CON01920
	DO 160 I=1,2	CON01930
160	G2(1,I)=G(4,I)	CON01940
C		CON01950
C	HX	CON01960
C		CON01970
	DO 115 I=1,2	CON01980
	DO 117 J=1,3	CON01990
117	HX1(I,J)=HX(I,J)	CON02000
115	CONTINUE	CON02010
	DO 114 I=1,2	CON02020
114	HX2(I,1)=HX(I,4)	CON02030
C		CON02040
C	COMPUTE STATE TRANSITION MATRIX AND CONTROL EFFECTS MATRIX	CON02050
C		CON02060
	CALL STX(4,F,.1,50,PHI,0)	CON02070
	CALL CEM(4,F,2,G,.1,PHI,GAMMA,T1,T2,T3,IDEN)	CON02080
C		CON02090
C	OUTPUT PHI AND GAMMA	CON02100
C		CON02110
	WRITE(6,801)	CON02120
	DO 105 I=1,4	CON02130
105	WRITE(6,802) (PHI(I,J),J=1,4), (GAMMA(I,K),K=1,2)	CON02140
C		CON02150
C	200-----COMPUTATION OF GAINS	CON02160
C		CON02170
C	INCLUDES: COMPUTATION OF OPTIMAL GAINS (C) AND THE GAIN	CON02180
C	MATRICES CB, CP, AND CI. ALSO, THE STEADY STATE ESTIMATES	CON02190
C	(FROM F MATRIX), COMPUTATION OF CONTINUOUS AND DISCRETE	CON02200

C	WEIGHTING MATRICES, AND THE SOLUTION TO THE DISCRETE	CON02210
C	RICCATI EQUATION (P).	CON02220
C		CON02230
C		CON02240
C	S MATRIX	CON02250
C		CON02260
C		CON02270
C	FORM SIN V	CON02280
C		CON02290
	DO 200 I=1,3	CON02300
	DO 202 J=1,3	CON02310
202	SINV(I,J)=F(I,J)	CON02320
200	CONTINUE	CON02330
	DO 204 I=1,3	CON02340
	DO 206 J=1,2	CON02350
	SINV(J+3,I)=HX(J,I)	CON02360
206	SINV(I,J+3)=G(I,J)	CON02370
204	CONTINUE	CON02380
	DO 208 I=1,2	CON02390
	DO 210 J=1,2	CON02400
210	SINV(I+3,J+3)=H(I,J)	CON02410
208	CONTINUE	CON02420
C		CON02430
C	COMPUTE S	CON02440
C		CON02450
	CALL INVS(5,SINV,4,APAS,S)	CON02460
C		CON02470
C	OUTPUT S	CON02480
C		CON02490
	WRITE(6,811)	CON02500
	DO 213 I=1,5	CON02510
213	WRITE(6,812) (S(I,J),J=1,5)	CON02520
C		CON02530
C	PARTITION S FOR GAIN CALCULATIONS	CON02540
C		CON02550
	DO 290 I=1,3	CON02560
	DO 291 J=1,3	CON02570
291	S11(I,J)=S(I,J)	CON02580
290	CONTINUE	CON02590
	DO 292 I=1,3	CON02600
	DO 293 J=1,2	CON02610
	S12(I,J)=S(I,J+3)	CON02620
293	S21(J,I)=S(J+3,I)	CON02630
292	CONTINUE	CON02640
	DO 294 I=1,2	CON02650
	DO 295 J=1,2	CON02660
295	S22(I,J)=S(I+3,J+3)	CON02670
294	CONTINUE	CON02680
C		CON02690
C	KY	CON02700
C		CON02710
	CALL MPLY(1,F21,3,S12,2,TEMP7)	CON02720
	CALL MPLY(1,G2,2,S22,2,TEMP8)	CON02730
	CALL MADD(1,TEMP7,2,TEMP8,KY)	CON02740
C		CON02750

C SXI	CON02760
C	CON02770
CALL MMPLY(3,S11,3,F12,1,SXI)	CON02780
CALL MMPLY(3,S12,2,HX2,1,TEMP13)	CON02790
CALL MADD(3,SXI,1,TEMP13,SXI)	CON02800
CALL SMPPLY(3,SXI,1,-1.,SXI)	CON02810
C	CON02820
C SXY	CON02830
C	CON02840
CALL MMPLY(3,S11,3,SXI,1,TEMP13)	CON02850
CALL MMPLY(3,TEMP13,1,KY,2,SXY)	CON02860
CALL MADD(3,S12,2,SXY,SXY)	CON02870
C	CON02880
C SUY	CON02890
C	CON02900
CALL MMPLY(2,S21,3,SXI,1,TEMP6)	CON02910
CALL MMPLY(2,TEMP6,1,KY,2,SUY)	CON02920
CALL MADD(2,S22,2,SUY,SUY)	CON02930
C	CON02940
C SUI	CON02950
C	CON02960
CALL MMPLY(2,S21,3,F12,1,SUI)	CON02970
CALL MMPLY(2,S22,2,HX2,1,TEMP6)	CON02980
CALL MADD(2,SUI,1,TEMP6,SUI)	CON02990
CALL SMPPLY(2,SUI,1,-1.,SUI)	CON03000
C	CON03010
C AN ADDITIONAL STEADY STATE RESPONSE?	CON03020
C	CON03030
GOTO 215	CON03040
C	CON03050
C INPUT BY VECTOR	CON03060
C	CON03070
215 DY(1,1)=10.	CON03080
DY(2,1)=0.	CON03090
216 DO 220 I=1,2	CON03100
220 DY(1,1)=DY(1,1)*3.14159/180.	CON03110
C	CON03120
C INITIALIZE XSTAR AND USTAR TO ZERO AT TIME=0	CON03130
C	CON03140
DO 222 I=1,3	CON03150
222 X1STAR(1,1)=0.	CON03160
DO 224 I=1,2	CON03170
224 USTAR(I,1)=0.	CON03180
X2STAR(1,1)=0.	CON03190
TIME=0.	CON03200
C	CON03210
C OUTPUT HEADING	CON03220
C	CON03230
WRITE(6,829)	CON03240
WRITE(6,850)	CON03250
WRITE(6,825)	CON03260
WRITE(6,826)	CON03270
C	CON03280
C ITPATH STEADY STATE RESPONSE FOR TWO SECONDS AT .2 SECOND INTERVALS	CON03290
C	CON03300

```

      DO 230 I=1,11
C
C X2STAR
C
      CALL SMPLY(1,KY,2,TIME,KYDT)
      CALL MMPLY(1,KYDT,2,DY,1,X2STAR)
C
C X1STAR
C
      CALL MMPLY(3,SKY,2,DY,1,X1STAR)
      CALL MMPLY(3,SKI,1,X2STAR,1,TEMP13)
      CALL MADD(3,X1STAR,1,TEMP13,X1STAR)
C
C USTAR
C
      CALL MMPLY(2,SUY,2,DY,1,USTAR)
      CALL MMPLY(2,SUI,1,X2STAR,1,TEMP6)
      CALL MADD(2,USTAR,1,TEMP6,USTAR)
C
C CONVERT XSTAR AND USTAR TO DEGREES FOR OUTPUT
C
      DO 236 J=1,3
236  XOUT(J)=X1STAR(J,1)*180./3.14159
      XOUT(4)=X2STAR(1,1)*180./3.14159
      DO 238 J=1,2
238  U2OUT(J)=USTAR(J,1)*180./3.14159
      WRITE(6,827) TIME, (XOUT(J),J=1,4), (U2OUT(K),K=1,2)
C
C STEP TIME .2 SECONDS
C
      TIME=.2*FLOAT(1)
      CONTINUE
230  IF (DY(1,1).EQ.0.) GOTO 240
      DY(1,1)=0.
      DY(2,1)=2.
      GOTO 216
C
C OPEN LOGIC
C
240  WRITE(6,860)
      READ(6,*) IREQUEST
      IF (IREQUEST.EQ.1) GOTO 2403
C
C INITIALIZE STATE AND CONTROL WEIGHTING MATRICES
C
      DO 242 I=1,4
      DO 244 J=1,4
244  QC(I,J)=0.
242  CONTINUE
      DO 246 I=1,2
      DO 248 J=1,2
248  PC(I,J)=0.
246  CONTINUE
      QC(1,1)=1.
      PC(2,2)=10.

```

CON03310
 CON03320
 CON03330
 CON03340
 CON03350
 CON03360
 CON03370
 CON03380
 CON03390
 CON03400
 CON03410
 CON03420
 CON03430
 CON03440
 CON03450
 CON03460
 CON03470
 CON03480
 CON03490
 CON03500
 CON03510
 CON03520
 CON03530
 CON03540
 CON03550
 CON03560
 CON03570
 CON03580
 CON03590
 CON03600
 CON03610
 CON03620
 CON03630
 CON03640
 CON03650
 CON03660
 CON03670
 CON03680
 CON03690
 CON03700
 CON03710
 CON03720
 CON03730
 CON03740
 CON03750
 CON03760
 CON03770
 CON03780
 CON03790
 CON03800
 CON03810
 CON03820
 CON03830
 CON03840
 CON03850

```

      QC(3,3)=1.
      QC(4,4)=25.
      RC(1,1)=1.
      RC(2,2)=.1
C
C COMPUTE THE DISCRETE WEIGHTING MATRICES USING SIMPSON'S RULE
C INTEGRATION.
C
C
C INITIALIZE TEMPOFAPY Q AND P
C
      DO 252 I=1,4
      DO 254 J=1,4
254   Q(I,J)=QC(I,J)
252   CONTINUE
      DO 256 I=1,2
      DO 258 J=1,2
258   R(I,J)=RC(I,J)
256   CONTINUE
C
C INITIALIZATION
C
      TNOW=0.
      INTVL=.1
      INLFX=0
      H=INTVL/10.
C
C QHAT,MHAT, AND RHAT AT T=0
C
      CALL STM(4,F,TNOW,50,PHITEM,0)
      CALL CEM(4,F,2,3,TNOW,PHITEM,GAMTEM,T1,T2,T3,IDEN)
      CALL TRNSPS(4,PHITEM,4,PHIT)
      CALL MMPLY(4,PHIT,4,Q,4,TEMP1)
      CALL MMPLY(4,TEMP1,4,PHITEM,4,QHATOT)
      CALL MMPLY(4,TEMP1,4,GAMTEM,2,MHATOT)
      CALL TRNSPS(4,GAMTEM,2,GAMMAT)
      CALL MMPLY(2,GAMMAT,4,Q,4,TEMP2)
      CALL MMPLY(2,TEMP2,4,GAMTEM,2,TEMP3)
      CALL MADD(2,R,2,TEMP3,RHATOT)
C
C ITERATE TO FIND QHAT,MHAT, AND RHAT FOR TIME INTERVAL = .1 SEC
C
      DO 260 I=1,10
      TNOW=TNOW+FLOAT(I)*H
      CALL STM(4,F,TNOW,50,PHITEM,0)
      CALL CEM(4,F,2,3,TNOW,PHITEM,GAMTEM,T1,T2,T3,IDEN)
      CALL TRNSPS(4,PHITEM,4,PHIT)
      CALL MMPLY(4,PHIT,4,Q,4,TEMP1)
      CALL MMPLY(4,TEMP1,4,PHITEM,4,QTNOW)
      CALL MMPLY(4,TEMP1,4,GAMTEM,2,MTNOW)
      CALL TRNSPS(4,GAMTEM,2,GAMMAT)
      CALL MMPLY(2,GAMMAT,4,Q,4,TEMP2)
      CALL MMPLY(2,TEMP2,4,GAMTEM,2,TEMP3)
      CALL MADD(2,R,2,TEMP3,PTNOW)
      IF(INDEX.EQ.0) GOTO 262

```

CON03860
 CON03870
 CON03880
 CON03890
 CON03900
 CON03910
 CON03920
 CON03930
 CON03940
 CON03950
 CON03960
 CON03970
 CON03980
 CON03990
 CON04000
 CON04010
 CON04020
 CON04030
 CON04040
 CON04050
 CON04060
 CON04070
 CON04080
 CON04090
 CON04100
 CON04110
 CON04120
 CON04130
 CON04140
 CON04150
 CON04160
 CON04170
 CON04180
 CON04190
 CON04200
 CON04210
 CON04220
 CON04230
 CON04240
 CON04250
 CON04260
 CON04270
 CON04280
 CON04290
 CON04300
 CON04310
 CON04320
 CON04330
 CON04340
 CON04350
 CON04360
 CON04370
 CON04380
 CON04390
 CON04400

```

      TERM=2.
      INDEX=0
      GOTO 264
262  TERM=4.
      INDEX=1
264  IF (I.EQ.10) TERM=1.
      CALL SMPLY (4,QTNOW,4,TERM,CTNOW)
      CALL SMPLY (4,MTNCW,2,TERM,MTNOW)
      CALL SMPLY (2,RTNCW,2,TERM,RTNCW)
      CALL MADD (4,QHATOT,4,QTNOW,QHATOT)
      CALL MADD (4,MHATOT,2,MTNOW,MHATOT)
      CALL MADD (2,RHATOT,2,RTNCW,RHATOT)
260  CONTINUE
C
C CALCULATE QHAT, MHAT, AND RHAT
C
      TERM=H/3.
      CALL SMPLY (4,QHATOT,4,TERM,QHAT)
      CALL SMPLY (4,MHATOT,2,TERM,MHAT)
      CALL SMPLY (2,RHATOT,2,TERM,RHAT)
C
C OUTPUT QC, RC, QHAT, MHAT, RHAT
C
      WRITE (6,805)
      DO 266 I=1,4
266  WRITE (6,806) (QC(I,J),J=1,4), (QHAT(I,K),K=1,4)
      WRITE (6,807)
      DO 268 I=1,4
268  WRITE (6,808) (MHAT(I,J),J=1,2)
      WRITE (6,809)
      DO 269 I=1,2
269  WRITE (6,810) (RC(I,J),J=1,2), (RHAT(I,K),K=1,2)
C
C SOLUTION OF THE DISCRETE RICCATI EQUATION
C
      DO 270 I=1,4
      DO 272 J=1,4
      PK(I,J)=0.
272  PHMNS1(I,J)=0.
270  CONTINUE
      CALL TNSPRS (4,PHI,4,PHIT)
      CALL TNSPRS (4,GAMMA,2,GAMMAT)
      CALL TNSPRS (4,MHAT,2,MHATT)
      PDET=1.
      INDOCF=0
      DO 275 I=1,1000
      CALL MMPLY (4,PK,4,PHI,4,TEMP1)
      CALL MMPLY (4,PK,4,GAMMA,2,TEMP4)
      CALL MMPLY (2,GAMMAT,4,TEMP1,4,TEMP2)
      CALL MMPLY (4,PHIT,4,TEMP1,4,TEMP1)
      CALL MMPLY (2,GAMMAT,4,TEMP4,2,TEMP3)
      CALL MADD (2,TEMP2,4,MHATT,TEMP2)
      CALL MADD (2,TEMP3,2,RHAT,TEMP3)
      CALL TNSPRS (2,TEMP2,4,TEMP4)
      CALL INVR5 (2,TEMP3,1,AREAR,TEMP3)

```

CON04410
 CON04420
 CCN04430
 CON04440
 CON04450
 CON04460
 CON04470
 CON04480
 CON04490
 CON04500
 CON04510
 CON04520
 CON04530
 CCN04540
 CON04550
 CON04560
 CON04570
 CON04580
 CON04590
 CCN04600
 CCN04610
 CON04620
 CON04630
 CON04640
 CON04650
 CON04660
 CON04670
 CCN04680
 CON04690
 CON04700
 CON04710
 CON04720
 CON04730
 CON04740
 CON04750
 CON04760
 CCN04770
 CON04780
 CON04790
 CON04800
 CON04810
 CON04820
 CON04830
 CON04840
 CON04850
 CON04860
 CON04870
 CON04880
 CON04890
 CON04900
 CON04910
 CON04920
 CON04930
 CON04940
 CON04950


```

      CALL MMPLY(4,TEMP4,2,TEMP3,2,TEMP4)
      CALL MMPLY(4,TEMP4,2,TEMP2,4,TEMP5)
      CALL SMPLY(4,TEMP5,4,-1.,TEMP5)
      CALL MADD(4,TEMP5,4,QHAT,TEMP5)
      CALL MADD(4,TEMP1,4,TEMP5,PKMNS1)
      PDET2=1.
      DO 276 J=1,4
      XT1(J)=0.
      DO 277 K=1,4
277  XT1(J)=XT1(J)+PKMNS1(J,K)
      IF(XT1(J).EQ.0.) GOTO 276
      PDET2=PDET2*XT1(J)
276  CONTINUE
      DO 278 J=1,4
      DO 279 K=1,4
279  PK(J,K)=PKMNS1(J,K)
278  CONTINUE
      DO 280 J=1,4
      DO 281 K=1,4
      IF(J.GE.K) GOTO 281
      PK(J,K)=.5*(PK(J,K)+PK(K,J))
      PK(K,J)=PK(J,K)
281  CONTINUE
280  CONTINUE
      DP=ABS((PDET2-PDET)/PDET)
      IF(DP.LE.(1.E-6)) GOTO 2401
      PDET=PDET2
275  CONTINUE
C
C ERROR MESSAGE FOR NONCONVERGENCE
C
      WRITE(6,813) I
C
C OUTPUT B(STEADY STATE)
C
2401  WRITE(6,814)
      DO 282 I=1,4
282  WRITE(6,815) (PK(I,J),J=1,4)
C
C COMPUTATION OF OPTIMAL GAINS, C
C
      CALL MMPLY(4,PK,4,GAMMA,2,TEMP4)
      CALL MMPLY(4,PK,4,PHI,4,TEMP1)
      CALL MMPLY(2,GAMMAT,4,TEMP4,2,TEMP3)
      CALL MMPLY(2,GAMMAT,4,TEMP1,4,TEMP2)
      CALL MADD(2,TEMP3,2,RHAT,TEMP3)
      CALL MADD(2,TEMP2,4,WHATT,TEMP2)
      CALL INVPS(2,TEMP3,1,AREAL,TEMP3)
      CALL MMPLY(2,TEMP3,2,TEMP2,4,C)
      GOTO 2402
C
C IF OPEN LOOP, OPTIMAL GAINS ARE ZERO
C
2403  DO 285 I=1,2
      DO 286 J=1,4

```

```

CON04960
CON04970
CON04980
CON04990
CON05000
CON05010
CON05020
CON05030
CON05040
CON05050
CON05060
CON05070
CON05080
CON05090
CON05100
CON05110
CON05120
CON05130
CON05140
CON05150
CON05160
CON05170
CON05180
CON05190
CON05200
CON05210
CON05220
CON05230
CON05240
CON05250
CON05260
CON05270
CON05280
CON05290
CON05300
CON05310
CON05320
CON05330
CON05340
CON05350
CON05360
CON05370
CON05380
CON05390
CON05400
CON05410
CON05420
CON05430
CON05440
CON05450
CON05460
CON05470
CON05480
CON05490
CON05500

```

286	C(I,J)=0.	CON05510
285	CONTINUE	CON05520
C		CON05530
C	OUTPUT OPTIMAL GAINS	CON05540
C		CON05550
2402	WRITE(6,815)	CON05560
	DO 287 I=1,2	CON05570
287	WRITE(6,817) (C(I,J),J=1,4)	CON05580
C		CON05590
C	COMPUTATION OF CB, CF, AND CI	CON05600
C		CON05610
C		CON05620
C	PARTITION F,G,H,C	CON05630
C		CON05640
	DO 297 I=1,2	CON05650
	DO 298 J=1,3	CON05660
298	C1(I,J)=C(I,J)	CON05670
297	CONTINUE	CON05680
	DO 299 I=1,2	CON05690
299	C2(I,1)=C(I,4)	CON05700
C		CON05710
C	COMPUTE GAIN MATRICES - CB,CF,CI	CON05720
C		CON05730
C		CON05740
C	CB	CON05750
C		CON05760
	CALL SMPLY(2,C,4,-1.,CB)	CON05770
C		CON05780
C	CF	CON05790
C		CON05800
	CALL SMPLY(2,C1,3,SXY,2,CF)	CON05810
	CALL XALL(2,CF,2,SXY,CF)	CON05820
C		CON05830
C	CS	CON05840
C		CON05850
	CALL SMPLY(2,C1,3,SXI,1,CS)	CON05860
	CALL XADD(2,CS,1,C2,CS)	CON05870
	CALL XADD(2,SXI,1,CS,CS)	CON05880
C		CON05890
C	CI	CON05900
C		CON05910
	CALL SMPLY(2,CS,1,FY,2,CI)	CON05920
C		CON05930
C	OUTPUT GAIN MATRICES	CON05940
C		CON05950
	WRITE(6,818)	CON05960
	DO 2100 I=1,2	CON05970
2100	WRITE(6,819) (CB(I,J),J=1,4), (CF(I,K),K=1,2), (CI(I,L),L=1,2)	CON05980
C		CON05990
C	SCHEDULED GAINS?	CON06000
C		CON06010
	WRITE(6,872)	CON06020
	READ(6,*) IQUEST	CON06030
	IF(IQUEST.NE.1) GOTO 300	CON06040
	DO 2225 I=1,7	CON06050

```

      DO 2230 J=1,9
      A2(I,J)=A21(I,J)
2230  A2(I,J+9)=A22(I,J)
2225  CONTINUE
875   FORMAT(' ', 'FC1 EQUALS: ', 9(E10.4,1X))
876   FORMAT(' ', 'A1  EQUALS: ', 9(E10.4,1X))
877   FORMAT(' ', ' ', 9(E10.4,1X))
      WRITE(6,875) (FC1(I,1), I=1,9)
      WRITE(6,876) (A1(1,I), I=1,9)
      DO 2200 I=1,6
2200  WRITE(6,877) (A1(I+1,J), J=1,9)
878   FORMAT(' ', 'FC2 EQUALS: ', 9(E10.4,1X))
879   FORMAT(' ', 'A2  EQUALS: ', 9(E10.4,1X))
880   FORMAT(' ', ' ', 9(E10.4,1X))
      WRITE(6,878) (FC2(I,1), I=1,9)
      WRITE(6,879) (A2(1,I), I=1,9)
      DO 2201 I=1,6
2201  WRITE(6,880) (A2(I+1,J), J=1,9)
      WRITE(6,878) (FC2(I,1), I=10,18)
      WRITE(6,879) (A2(1,I), I=10,18)
      DO 2202 I=1,6
2202  WRITE(6,880) (A2(I+1,J), J=10,18)
      CALL MMPLY(7,A1,9,FC1,1,CFEED1)
      CALL MMPLY(7,A2,18,FC2,1,CFEED2)
      CB(1,1)=CFEED1(1,1)
      CB(2,1)=CFEED1(2,1)
      CB(2,3)=CFEED1(3,1)
      CB(2,4)=CFEED1(4,1)
      CF(1,2)=CFEED1(5,1)
      CF(2,1)=CFEED1(6,1)
      CF(2,2)=CFEED1(7,1)
      CB(1,2)=CFEED2(1,1)
      CB(1,3)=CFEED2(2,1)
      CB(1,4)=CFEED2(3,1)
      CB(2,2)=CFEED2(4,1)
      CF(1,1)=CFEED2(5,1)
      CI(1,1)=CFEED2(6,1)
      CI(2,1)=CFEED2(7,1)
      CI(1,2)=0.
      CI(2,2)=0.
      WRITE(6,873)
      WRITE(6,813)
      DO 2220 I=1,2
2220  WRITE(6,810) (CB(I,J), J=1,4), (CF(I,K), K=1,2), (CI(I,L), L=1,2)
C
C 300-----COMPUTATION OF CLOSED LOOP F
C
C   INCLUDES: COMPUTATION OF FCL AND ITS EIGENVALUES
C
300   CALL MMPLY(4,GAMMA,2,CB,4,TEMP1)
      CALL MADD(4,PHI,4,TEMP1,PHICL)
      CALL CLOCF(4,PHICL,.1,FCL,1)
      CALL EIGEN(FCL,FCLCOPY,4,0,WEIGEN,ZEIGEN,WKF)
C
C 400-----TIME HISTORY

```

CON06060
 CON06070
 CON06080
 CON06090
 CON06100
 CON06110
 CON06120
 CON06130
 CON06140
 CON06150
 CON06160
 CON06170
 CON06180
 CON06190
 CON06200
 CON06210
 CON06220
 CON06230
 CON06240
 CON06250
 CON06260
 CON06270
 CON06280
 CON06290
 CON06300
 CON06310
 CON06320
 CON06330
 CON06340
 CON06350
 CON06360
 CON06370
 CON06380
 CON06390
 CON06400
 CON06410
 CON06420
 CON06430
 CON06440
 CON06450
 CON06460
 CON06470
 CON06480
 CON06490
 CON06500
 CON06510
 CON06520
 CON06530
 CON06540
 CON06550
 CON06560
 CON06570
 CON06580
 CON06590
 CON06600

C		CON06610
C		CCN06620
C	NONLINEAR TIME HISTORY (4TH ORDER RUNGA-KUTTA INTEGRATION)	CON06630
C		CCN06640
	WRITE(6,831)	CON06650
	READ(6,*)IQUEST	CON06660
	IF(IQUEST.NE.1) GOTO 460	CON06670
	CALL RKINT(A,TC,VPL,ALT,CB,CF,CI,XT,UT)	CON06680
C		CON06690
C	LINEAR TIME HISTORY (STATE TRANSITION MATRIX)	CON06700
C		CON06710
460	WRITE(6,832)	CON06720
	READ(6,*)IQUEST	CON06730
	IF(IQUEST.NE.1) GOTO 500	CON06740
	IF(IQUEST.NE.1) GOTO 500	CON06750
	INDCTR=0	CON06760
	INTVL=.1	CON06770
	ISFC=5	CON06780
	ITER=10*ISFC	CON06790
	IOUT=ITER/50	CCN06800
	GOTO 405	CON06810
405	DY(1,1)=10.	CON06820
	DY(2,1)=0.	CON06830
402	DO 406 I=1,2	CCN06840
406	DY(I,1)=DY(I,1)*3.14159/180.	CON06850
	DO 410 I=1,4	CON06860
410	X(I,1)=0.	CON06870
	DO 411 I=1,4	CON06880
411	XOUT(I)=X(I,1)*180./3.14159	CON06890
	WRITE(6,851)	CON06900
	WRITE(6,850)	CON06910
	WRITE(6,825)	CCN06920
	WRITE(6,826)	CON06930
	CALL MMPLY(2,C2,2,DY,1,U2)	CON06940
	DO 440 I=1,2	CON06950
440	U2OUT(I)=U2(I,1)*180./3.14159	CCN06960
	TIME=0.	CON06970
	WRITE(6,827)TIME,(XOUT(I),I=1,4),(U2OUT(J),J=1,2)	CON06980
	DO 415 I=1,ITER	CON06990
	INDCTR=INDCTR+1	CON07000
	TIME=INTVL*FLOAT(I)	CON07010
	CALL MMPLY(4,GAMMA,2,U2,1,TEMP10)	CON07020
	CALL MMPLY(4,PHI,4,X,1,X)	CON07030
	CALL MADE(4,X,1,TEMP10,X)	CON07040
	DYT(1,1)=DY(1,1)*TIME	CON07050
	DYT(2,1)=DY(2,1)*TIME	CCN07060
	CALL MMPLY(2,CI,2,DYT,1,TEMP6)	CON07070
	CALL MMPLY(2,CF,2,DY,1,TEMP11)	CON07080
	CALL MADE(2,TEMP11,1,TEMP6,TEMP6)	CON07090
	CALL MMPLY(2,C3,4,X,1,TEMP11)	CON07100
	CALL MADE(2,TEMP6,1,TEMP11,U2)	CCN07110
	IF(INDCTR.NE.IOUT) GOTO 415	CON07120
	DO 421 J=1,4	CON07130
421	XOUT(J)=X(J,1)*180./3.14159	CON07140
	DO 420 J=1,2	CON07150

426	U2OUT(J)=U2(J,1)*180./3.14159	CON07160
	WRITE(6,827) TIME, (XCUT(K), K=1,4), (U2CUT(J), J=1,2)	CON07170
	INDCTR=0	CON07180
415	CONTINUE	CON07190
	IF (DY(1,1).EQ.0.) GOTO 500	CON07200
	DY(1,1)=0.	CON07210
	DY(2,1)=2.	CON07220
	GOTO 402	CON07230
500	WRITE(6,830)	CON07240
	READ(6,*) IQUEST	CON07250
	IF (IQUEST.EQ.1) GOTO 240	CON07260
	WRITE(6,829)	CON07270
	READ(6,*) IQUEST	CON07280
	IF (IQUEST.EQ.1) GOTO 900	CON07290
	STOP	CON07300
	END	CON07310

```

C
C THE FOLLOWING SUBROUTINE COMPUTES THE NONLINEAR EQUATIONS OF
C MOTION BY ACCEPTING THE CURRENT STATE (X) AND THE FLIGHT
C CONDITION (A,TC,VEL) AND RETURNING THE STATE RATES (DX)
C
SUBROUTINE ENDYN(A,TC,VEL,ALT,X,U,DX)
  DIMENSION X(4),DX(4),U(2)
  COMMON/CNTPIS/DF,DA,BETA
  COMMON/CYCCEF/CY,CY0,CYB,CYDR,CYDA,CYP,CYR
  COMMON/CNCOEF/CN,CN0,CNB,CNDR,CNDA,CNP,CNR
  COMMON/CLCOEF/CL,CL0,CLB,CLDR,CLDA,CLP,CLR
  COMMON/CYCCMP/CYDT,CY00,CYLT,CYBO,CYDRT,CYDRO
  COMMON/CNCCMP/CNCT,CN00,CNBT,CNB0,CNERT,CNDR0
  COMMON/CLCCMP/CLDT,CL00,CLBT,CLB0,CLERT,CLDR0
  COMMON/PCOMP/CLPGAM,CLACLA,CLPDGT,CLPDG0,K,CNPCL
  COMMON/RCOMP/CLRCL,CLIFTT,CLIFT0,CDRAGT,CDRAGO,CYPCL
  COMMON/CNSTNT/MACH
  REAL MACH,IX,IZ,K,MASS

C COMPUTE CONSTANTS REQUIRED FOR COMPUTATIONS:
C   SSC - SPEED OF SOUND (FEET/SEC)
C   ALT - ALTITUDE (FEET)
C   RHO - DENSITY (SLUGS/FEET**3)
C   IX - MOMENT OF INERTIA ABOUT X (SLUGS*FEET**2)
C   IZ - MOMENT OF INERTIA ABOUT Z (SLUGS*FEET**2)
C   E - SPAN (FEET)
C   DYNPFC - DYNAMIC PRESSURE (LBS/(FEET**2))
C   MACH - LOCAL MACH NUMBER
C   S - WING AREA (FEET**2)
C   MASS - AIRCRAFT MASS (LBS*SEC**2/FEET)
C   GRAV - GRAVITATIONAL CONSTANT (FEET/(SEC**2))
C
  FPA=0.
  PI = 3.14159
  BETA = X(2)*180/PI
  APAD = A*PI/180.
  RHO = .002377*EXP(-ALT/25000.)
  SSC = 1116. - (.00409)*(ALT)
  MACH = VEL/SSC
  E = 33.33
  IX = 1234.74
  IZ = 3234.72
  S = 144.
  DYNPFC = .5*RHO*(VEL**2)
  GRAV = 32.174
  MASS = 2942./GRAV
  DR=U(1)*180./3.14159
  DA=U(2)*180./3.14159
  ZP=2.87
  LP=17.25
  Z=ZP*COS(APAD)-LP*SIN(APAD)

C
C INITIALIZE LONGITUDINAL STATE VARIABLES (Q,U,W)
C
  Q = 0

```

```

LDD00010
LDD00020
LDD00030
LDD00040
LDD00050
LDD00060
LDD00070
LDD00080
LDD00090
LDD00100
LDD00110
LDD00120
LDD00130
LDD00140
LDD00150
LDD00160
LDD00170
LDD00180
LDD00190
LDD00200
LDD00210
LDD00220
LDD00230
LDD00240
LDD00250
LDD00260
LDD00270
LDD00280
LDD00290
LDD00300
LDD00310
LDD00320
LDD00330
LDD00340
LDD00350
LDD00360
LDD00370
LDD00380
LDD00390
LDD00400
LDD00410
LDD00420
LDD00430
LDD00440
LDD00450
LDD00460
LDD00470
LDD00480
LDD00490
LDD00500
LDD00510
LDD00520
LDD00530
LDD00540
LDD00550

```

```

      VEL1 = (VEL**2-(SIN(X(2))*VEL)**2)**.5
      UVEL = VEL1
      W = 0.
C
C THE FOLLOWING PORTION COMPUTES THE STABILITY AND CONTROL
C DERIVATIVES GIVEN THE CURRENT FLIGHT CONDITION.
C
C CONSTANT COMPONENTS
C
      CYOT = (((((1.094E-8*A-7.104E-7)*A+1.63E-5)*A-1.496E-4)*A
1      +2.979E-4)*A+2.448E-3)*A-1.715E-2)*A
      CYOO = (((((-1.996E-9*A+1.298E-7)*A-2.974E-6)*A+2.713E-5)*A
1      -5.525E-5)*A-4.614E-4)*A+1.812E-3)*A
      CYO = CYOT*TC+CYOO
      CNOT = (((((5.553E-9*A-3.259E-7)*A+6.695E-6)*A-5.416E-5)*A
1      +7.633E-5)*A+1.025E-3)*A-4.504E-3)*A-.03281
      CNOO = (((((-3.817E-10*A-2.568E-8)*A+6.104E-7)*A-5.881E-6)*A
1      +1.52E-5)*A+8.592E-5)*A-5.593E-4)*A+.001822
      CNO = CNOT*TC+CNOO
      CLO = (((((-7.099E-10*A+4.458E-8)*A-9.659E-7)*A+7.877E-6)*A
1      -6.971E-6)*A-1.637E-4)*A+2.167E-4)*A+.00717
C
C SIDESLIP (B) DERIVATIVES
C
      CYBT = (((((1.274E-9*A-8.63E-8)*A+2.089E-6)*A-2.044E-5)*A
1      +4.791E-5)*A+3.116E-4)*A-1.607E-3)*A-.0188
      CYBO = .000239*A-.01249
      CYB = CYBT*TC+CYBO
      CNBT = (((((-2.604E-11*A+2.584E-9)*A-8.596E-8)*A+1.123E-6)*A
1      -4.376E-6)*A-2.051E-5)*A+3.196E-4)*A+.001
      CNBO = (((((-3.633E-12*A-4.747E-10)*A+1.815E-8)*A-2.599E-7)*A
1      +1.068E-6)*A+3.967E-6)*A-8.049E-5)*A+.002
      CNB = CNBT*TC+CNBO
      CLBT = (((((-3.283E-10*A+2.31E-8)*A-5.742E-7)*A+5.639E-6)*A
1      -1.109E-5)*A-9.937E-5)*A+4.282E-4)*A-.00106
      CLBO = (((((-6.697E-11*A-4.466E-9)*A+1.063E-7)*A-1.012E-6)*A
1      +1.852E-6)*A+1.483E-5)*A-1.463E-5)*A-.00155
      CLB = CLBT*TC+CLBO
C
C PITCH (DE) DERIVATIVES
C
      CYDPT = (((((6.225E-10*A-4.079E-8)*A+9.409E-7)*A-8.651E-6)*A
1      +1.617E-5)*A+1.582E-4)*A-2.147E-4)*A+.0068
      CYDPO = (((((4.637E-11*A-3.079E-9)*A+7.235E-8)*A-6.912E-7)*A
1      +1.695E-6)*A+6.535E-6)*A-1.208E-5)*A+.00296
      CYDR = CYDPT*TC+CYDPO
      CNDPT = (((((-2.538E-11*A+4.476E-9)*A-7.65E-8)*A+5.046E-7)*A
1      -1.104E-6)*A-4.531E-6)*A-8.116E-5)*A-.0043
      CNDPD = (((((-7.266E-11*A+4.832E-9)*A-1.141E-7)*A+1.077E-6)*A
1      -2.106E-6)*A-1.69E-5)*A+2.946E-5)*A-.00156
      CNDP = CNDPT*TC+CNDPD
      CLDPT = (((((3.278E-10*A-2.428E-8)*A+6.533E-7)*A-7.27E-6)*A
1      +2.069E-5)*A+1.294E-4)*A-6.396E-4)*A+.00168
      CLDPO = (((((-1.3E-10*A+9.568E-9)*A-2.394E-7)*A+2.487E-6)*A

```

```

1      -6.88E-6)*A-4.086E-5)*A+1.717E-4)*A+.000247
      CLDR = CLDRT*TC+CLDP0

```

AILEPON (DA) DERIVATIVES

```

      CYDA = ((((((2.433E-11*A-1.603E-9)*A+3.859E-8)*A-3.817E-7)*A
1      +8.667E-7)*A+5.518E-6)*A-2.085E-5)*A-.000267
      CNDA = ((((((1.637E-11*A+1.088E-9)*A-2.596E-8)*A+2.497E-7)*A
1      -5.133E-7)*A-4.023E-6)*A+3.243E-5)*A-.000005
      CLDA = ((((((1.381E-11*A+9.8E-10)*A-2.405E-8)*A+2.227E-7)*A
1      -2.285E-7)*A-5.377E-6)*A+3.619E-6)*A-.00127

```

COMPUTE CLIFT AND CDRAG TO HELP CALCULATE P AND R DERIVATIVES

```

      CLIFT = ((((((2.78E-8*A-1.81E-6)*A+4.299E-5)*A-4.463E-4)*A
1      +1.536E-3)*A+6.761E-3)*A-1.321E-3)*A+.357
      CLIFT0 = ((((((5.571E-10*A+8.952E-8)*A-3.684E-6)*A+5.42E-5)*A
1      -2.531E-4)*A-9.214E-4)*A+.08337)*A+.121
      CLIFT = CLIFT*TC+CLIFT0
      CDEAGT = ((((((1.401E-8*A+1.045E-6)*A-2.756E-5)*A+2.968E-4)*A
1      -9.036E-4)*A-.003371)*A+.02157)*A-1.053
      CDEAG0 = ((((((6.539E-10*A+2.608E-8)*A-1.176E-7)*A-5.244E-6)*A
1      +7.751E-5)*A-2.278E-5)*A+4.263E-5)*A+.0551
      CDEAG = CDEAGT*TC+CDEAG0

```

OIL RATE (P) DERIVATIVES

```

      CLPGAY = ((((-4.012*MACH+7.273)*MACH-4.379)*MACH+.8069)*MACH
1      -.426
      CLACLA = ((((((4.84E-10*A+1.797E-7)*A-6.524E-6)*A+6.049E-5)*A
1      -2.235E-5)*A-1.326E-3)*A-1.72E-3)*A+.99
      CLPDST = ((((((8.453E-10*A-5.086E-8)*A+1.066E-6)*A-9.049E-6)*A
1      +1.989E-5)*A+2.109E-4)*A+9.369E-5)*A+.1329
      CLPDG0 = ((((((2.325E-11*A-5.324E-10)*A-1.689E-8)*A+4.544E-7)*A
1      -3.004E-6)*A+5.38E-5)*A+2.386E-4)*A-.00675
      CLPDG = CLPDST*TC+CLPDG0
      CLPW = (-.42)*(.926)*CLACLA*.985+CLPDG
      TERM=2*(Z/B)*(Z-ZP)/B
      IF (TERM.LT.0.) TERM=-TERM
      CLP=CLPW+.5*(-.314)*.234*.155+TERM*(-.181)
      K=(((4.722E-8*A-1.83E-6)*A+4.235E-6)*A+.0005208)*A-
1      .0059)*A+.00723)*A+.087)*A+1.00
      CYPW = K*(-.06*CL)+.161
      CYF=CYPW+2*(Z-ZP)/B*(-.181)
      CNPW = -CLP*TAN(ARAD)-K*(-CLP*TAN(ARAD)+.1003*CLIFT)
      CNF=CNPW-(2/B)*(LP*COS(ARAD)+ZP*SIN(ARAD))*((Z-ZP)/B)*(-.181)

```

RATE (P) DERIVATIVES

```

      CLPW = CLIFT*.241+.001079
      CLF = CLPW-(2/(3**2))*((LP*COS(ARAD)+ZP*SIN(ARAD))*Z*(-.181)
      CDO = CDEAG0-(CLIFT**2)/(6.04*PI)
      CNEW = -.02*CLIFT**2-.32*CDO
      CNF = CNEW+(2/(3**2))*((LP*COS(ARAD)+ZP*SIN(ARAD))**2)*(-.181)
      CYF = 0

```

```

LDD01110
LDD01120
LDD01130
LDD01140
LDD01150
LDD01160
LDD01170
LDD01180
LDD01190
LDD01200
LDD01210
LDD01220
LDD01230
LDD01240
LDD01250
LDD01260
LDD01270
LDD01280
LDD01290
LDD01300
LDD01310
LDD01320
LDD01330
LDD01340
LDD01350
LDD01360
LDD01370
LDD01380
LDD01390
LDD01400
LDD01410
LDD01420
LDD01430
LDD01440
LDD01450
LDD01460
LDD01470
LDD01480
LDD01490
LDD01500
LDD01510
LDD01520
LDD01530
LDD01540
LDD01550
LDD01560
LDD01570
LDD01580
LDD01590
LDD01600
LDD01610
LDD01620
LDD01630
LDD01640
LDD01650

```


C		LDD01660
C	COMPUTE COEFFICIENTS	LDD01670
C		LDD01680
	CY = CY0+CYB*BETA+CYDB*DB+CYDA*DA+(CYP*X(3)+CYL*X(1))*B/(2*VEL)	LDD01690
	CN = CN0+CNE*BETA+CNDA*DB+CNDA*DA+(CNP*X(3)+CNE*X(1))*B/(2*VEL)	LDD01700
	CL = CL0+CLE*BETA+CLDB*DB+CLEA*DA+(CLP*X(3)+CLI*X(1))*B/(2*VEL)	LDD01710
C		LDD01720
C	COMPUTE STATE RATES	LDD01730
C		LDD01740
C	THE STATE RATES ARE CALCULATED IN THE FOLLOWING CODE:	LDD01750
C		LDD01760
C	DX(1) = PDOT (RAD/SEC**2)	LDD01770
C	DX(2) = PDOT (RAD/SEC)	LDD01780
C	DX(3) = PDOT (RAD/SEC**2)	LDD01790
C	DX(4) = PHIDOT (RAD/SEC)	LDD01800
C		LDD01810
	DX(1) = (DYNPRS*S*B*CN)/IZ	LDD01820
	DX(2) = (-X(1)*UVEL+X(3)*W+GRAV*CCS(FPA)*SIN(X(4)) +	LDD01830
1	(DYNPRS*S*CY)/(MASS))/VEL	LDD01840
	DX(3) = (DYNPRS*S*B*CL)/IX	LDD01850
	DX(4) = X(3)+(2*SIN(X(4))+X(1)*COS(X(4)))*TAN(FPA)	LDD01860
C		LDD01870
C		LDD01880
C	END OF NONLINEAR EQUATION SUBROUTINE	LDD01890
C		LDD01900
C		LDD01910
	PRINT	LDD01920
	END	LDD01930

```

C THIS PROGRAM GENERATES F AND G MATRICES FOR DIFFERENT FLIGHT CONDITIONFAN00010
C THE FLIGHT CONDITION IS SPECIFIED BY ANGLE OF ATTACK, A, THROTTLE FAN00020
C SETTING, TC, AND VELOCITY, VEL. FAN00030
C FAN00040
SUBROUTINE FANDG(A,TC,VEL,ALT,X,U,P,G,HX,HU,IOUT) FAN00050
DIMENSION F(4,4),G(4,2),X(4),XT(4),DX(4),DX1(4),HX(2,4),YPLUS(2), FAN00060
1 YMINUS(2),U(2),HU(2,2),UT(2) FAN00070
REAL NR FAN00080
COMMON/CNTRLS/DP,DA FAN00090
COMMON/CYCCEF/CY,CYC,CYB,CYDR,CYDA,CYP,CYR FAN00100
COMMON/CNCOEF/CN,CN0,CNB,CNDR,CNDA,CNP,CNR FAN00110
COMMON/CLCCEF/CL,CL0,CLB,CIDR,CLDA,CIP,CLR FAN00120
100 FORMAT('0','THE FOLLOWING F,G AND H MATRICES WERE GENERATED') FAN00130
101 FORMAT(' ','UNDER THE FOLLOWING FLIGHT CONDITION:') FAN00140
102 FORMAT(' ','10X','ANGLE OF ATTACK = ',P5.2,' DEG') FAN00150
103 FORMAT(' ','10X','THROTTLE SETTING = ',P6.4) FAN00160
104 FORMAT(' ','10X','VELOCITY = ',P8.4,' FEET/SEC') FAN00170
105 FORMAT(' ','10X','ALTITUDE = ',P10.3,' FEET') FAN00180
106 FORMAT(' ','F EQUALS:',47X,'G EQUALS:') FAN00190
107 FORMAT(' ','4F10.3,13X,2F10.3) FAN00200
108 FORMAT(' ','HX EQUALS:',46X,'HU EQUALS:') FAN00210
109 FORMAT(' ','DET(F) EQUALS:',P10.3) FAN00220
110 FORMAT(' ','CYO CYB CYDR CYDA ', FAN00230
1 'CYP CYR') FAN00240
111 FORMAT(' ','CN0 CNB CNDR CNDA ', FAN00250
1 'CNP CNR') FAN00260
112 FORMAT(' ','CL0 CLB CLDR CLDA ', FAN00270
1 'CLP CLR') FAN00280
113 FORMAT('0',6(10.3,2X)) FAN00290
C FAN00300
C COMPUTE F USING PERTURBATION METHOD FAN00310
C FAN00320
DEIX = .1 FAN00330
DO 150 I50=1,4 FAN00340
DO 151 I51=1,4 FAN00350
151 XT(I51) = X(I51) FAN00360
XT(I50) = X(I50) + .5 * DEIX FAN00370
CALL DNDYN(A,TC,VEL,ALT,XI,U,DX) FAN00380
XT(I50) = X(I50) - .5 * DEIX FAN00390
CALL DNDYN(A,TC,VEL,ALT,XI,U,DX1) FAN00400
DO 152 I52=1,4 FAN00410
152 F(I52,I50) = (DX(I52) - DX1(I52))/DEIX FAN00420
150 CONTINUE FAN00430
C FAN00440
C COMPUTE G USING PERTURBATION METHOD FAN00450
C FAN00460
DELU=.1 FAN00470
DO 153 I53=1,2 FAN00480
DO 154 I54=1,2 FAN00490
154 UT(I54) = U(I54) FAN00500
UT(I53) = U(I53) + .5*DELU FAN00510
CALL DNDYN(A,TC,VEL,ALT,X,UT,LX) FAN00520
UT(I53) = U(I53) - .5*DELU FAN00530
CALL DNDYN(A,TC,VEL,ALT,X,UT,DX1) FAN00540
DO 155 I55=1,4 FAN00550

```

1551	G(I55,I53) = (DX(I55) - DX1(I55)) / DELU	PAN00560
153	CONTINUE	PAN00570
C		PAN00580
C	COMPUTE HX USING PERTURBATION METHOD	PAN00590
C		PAN00600
	DO 155 I55=1,4	PAN00610
	DO 156 I56=1,4	PAN00620
156	XT(I56)=X(I56)	PAN00630
	XT(I55)=X(I55) + .5*DELX	PAN00640
	YPLUS(1)=XT(3)	PAN00650
	YPLUS(2)=XT(2)	PAN00660
	XT(I55)=X(I55) - .5*DELX	PAN00670
	YMINUS(1)=XT(3)	PAN00680
	YMINUS(2)=XT(2)	PAN00690
	DO 157 I57=1,2	PAN00700
157	HX(I57,I55) = (YPLUS(I57) - YMINUS(I57)) / DELX	PAN00710
155	CONTINUE	PAN00720
C		PAN00730
C	COMPUTE HY USING PERTURBATION METHOD	PAN00740
C		PAN00750
	DO 158 I58=1,2	PAN00760
	DO 159 I59=1,2	PAN00770
159	UT(I59)=U(I59)	PAN00780
	UT(I58)=U(I58) + .5*DELU	PAN00790
	YPLUS(1)=X(3)	PAN00800
	YPLUS(2)=X(2)	PAN00810
	UT(I58)=U(I58) - .5*DELU	PAN00820
	YMINUS(1)=X(3)	PAN00830
	YMINUS(2)=X(2)	PAN00840
	DO 160 I60=1,2	PAN00850
160	HU(I60,I58) = (YPLUS(I60) - YMINUS(I60)) / .05	PAN00860
158	CONTINUE	PAN00870
C		PAN00880
C	CALCULATE DETERMINANT OF F	PAN00890
C		PAN00900
	CALL DETXNT(4,F,DETF)	PAN00910
C		PAN00920
C	OUTPUT FLIGHT CONDITION AND F AND G MATRICES	PAN00930
C		PAN00940
	IF(IOUT.EQ.1) GOTO 600	PAN00950
	WRITE(6,100)	PAN00960
	WRITE(6,101)	PAN00970
	WRITE(6,102) A	PAN00980
	WRITE(6,103) TC	PAN00990
	WRITE(6,104) VEL	PAN01000
	WRITE(6,105) ALT	PAN01010
	WRITE(6,106)	PAN01020
	DO 161 I61=1,4	PAN01030
161	WRITE(6,107) (F(I61,I62), I62=1,4), (G(I61,I63), I63=1,2)	PAN01040
	WRITE(6,108) DETF	PAN01050
	WRITE(6,109)	PAN01060
	DO 164 I64=1,2	PAN01070
164	WRITE(6,107) (HX(I64,I65), I65=1,4), (HU(I64,I66), I66=1,2)	PAN01080
600	RETURN	PAN01090
	END	PAN01100

```

C
C THIS SUBROUTINE COMPUTES THE STATE TRANSITION MATRIX FOR A SYSTEM
C SPECIFIED BY THE MATRIX A AND THE TIME STEP T. IT ITERATES TO
C FIND A SOLUTION WHICH CONVERGES TO WITHIN .1% AFTER KEND ITERATIONS.
C THE STATE TRANSITION MATRIX IS THEN PLACED IN PHI.
C
      SUBROUTINE STM(ORDER,A,T,KEND,PHI,OUTPUT)
      INTEGER ORDER,OUTPUT
      DIMENSION A(ORDER,ORDER),A1(12,12),A2(12,12),
1      A3(12,12),IDEN(4,4),PHI(ORDER,ORDER),
2      PHI1(12,12),TEMP(12,12),T1(4,4),T2(4,4),T3(4,4)
      REAL IDEN
100  FORMAT(' ','***WARNING-THE STATE TRANSITION MATRIX DID NOT')
101  FORMAT(' ','CONVERGE IN THE ',I2,'TH ORDER***')
102  FORMAT(' ','THE STATE TRANSITION MATRIX CONVERGED IN ',I2,
1      'TH ORDER')
103  FORMAT(' ','THE STATE TRANSITION MATRIX IS AS FOLLOWS:')
104  FORMAT('0',10(2X,F8.3))
C
C INITIALIZE IDENTITY MATRIX TO THE ORDER OF THE SYSTEM
C
      DO 200 I=1,ORDER
      DO 201 J=1,ORDER
      IDEN(I,J)=0.
201  CONTINUE
200  CONTINUE
      DO 202 I=1,ORDER
      IDEN(I,I)=1.
202  CONTINUE
C
C INITIALIZE:      PHI = I
C                  A1 = A*T
C                  A3 = I
C
      DO 203 I=1,ORDER
      DO 204 J=1,ORDER
      PHI(I,J)=IDEN(I,J)
      A3(I,J)=IDEN(I,J)
      A1(I,J)=A(I,J)*T
204  CONTINUE
203  CONTINUE
C
C START LOOP TO CONVERGE TO STATE TRANSITION MATRIX
C
      DO 205 K=1,KEND
C
C A2 = A*T/K
C
      DO 206 I=1,ORDER
      DO 207 J=1,ORDER
      A2(I,J)=A1(I,J)/FLOAT(K)
207  CONTINUE
206  CONTINUE
C
C A3 = (A**K)*(T**K)/K!

```

```

STM00010
STM00020
STM00030
STM00040
STM00050
STM00060
STM00070
STM00080
STM00090
STM00100
STM00110
STM00120
STM00130
STM00140
STM00150
STM00160
STM00170
STM00180
STM00190
STM00200
STM00210
STM00220
STM00230
STM00240
STM00250
STM00260
STM00270
STM00280
STM00290
STM00300
STM00310
STM00320
STM00330
STM00340
STM00350
STM00360
STM00370
STM00380
STM00390
STM00400
STM00410
STM00420
STM00430
STM00440
STM00450
STM00460
STM00470
STM00480
STM00490
STM00500
STM00510
STM00520
STM00530
STM00540
STM00550

```

C		STM00560
	DO 203 I=1,ORDER	STM00570
	DO 209 J=1,ORDER	STM00580
	DO 210 L=1,ORDER	STM00590
	TEMP(I,J)=TEMP(I,J)+A3(I,L)*A2(L,J)	STM00600
210	CONTINUE	STM00610
209	CONTINUE	STM00620
208	CONTINUE	STM00630
	DO 211 I=1,ORDER	STM00640
	DO 212 J=1,ORDER	STM00650
	A3(I,J)=TEMP(I,J)	STM00660
	TEMP(I,J)=0	STM00670
212	CONTINUE	STM00680
211	CONTINUE	STM00690
C		STM00700
C	PHI1 = I + AT + (A**2)*(T**2)/2! + . . . (TO KTH ORDER)	STM00710
C		STM00720
	DO 213 I=1,ORDER	STM00730
	DO 214 J=1,ORDER	STM00740
	PHI1(I,J)=PHI(I,J)+A3(I,J)	STM00750
214	CONTINUE	STM00760
213	CONTINUE	STM00770
C		STM00780
C	IS THE ERROR LESS THAN OR EQUAL TO .1%?	STM00790
C		STM00800
	ERROR=0	STM00810
	DO 215 I=1,ORDER	STM00820
	ERROR=ERROR+(PHI1(I,I)-PHI(I,I))**2	STM00830
215	CONTINUE	STM00840
	IF(ERROR.LE.(.101)) GOTO 218	STM00850
C		STM00860
C	PHI <-- PHI1	STM00870
C		STM00880
	DO 216 I=1,ORDER	STM00890
	DO 217 J=1,ORDER	STM00900
	PHI(I,J)=PHI1(I,J)	STM00910
217	CONTINUE	STM00920
216	CONTINUE	STM00930
205	CONTINUE	STM00940
C		STM00950
C	END OF CONVERGENCE LOOP	STM00960
C		STM00970
	IF(OUTPUT.NE.1) GOTO 219	STM00980
	WRITE(6,100)	STM00990
	WRITEF(6,101)*	STM01000
	GOTO 219	STM01010
218	IF(OUTPUT.NE.1) GOTO 219	STM01020
	WRITEF(6,102)*	STM01030
219	DO 220 I=1,ORDER	STM01040
	DO 221 J=1,ORDER	STM01050
	PHI(I,J)=PHI1(I,J)	STM01060
221	CONTINUE	STM01070
220	CONTINUE	STM01080
	IF(OUTPUT.NE.1) GOTO 900	STM01090
	WRITE(6,103)	STM01100

FILE: STM FORTRAN A

PRINCETON UNIVERSITY TIME-SHARING SYSTEM

```
      DO 300 I=1,ORDEE
      WRITE(6,104) (PHI(I,J),J=1,CRDEP)
300   CONTINUE
900   RETURN
      END
```

```
STM01110
STM01120
STM01130
STM01140
STM01150
```

	SUBROUTINE CEM(NF,F,NG,G,T,PHI,GAM,T1,T2,T3,IDEN)	CEM00010
	DIMENSION F(NF,NF),G(NF,NG),PHI(NF,NF),GAM(NF,NG),T1(NF,NF),	CEM00020
	1 T2(NF,NF),T3(NF,NF),IDEN(NF,NF)	CEM00030
	REAL IDEN	CEM00040
	DO 10 I=1,4	CEM00050
	DO 20 J=1,NF	CEM00060
20	IDEN(I,J)=0.	CEM00070
10	CONTINUE	CEM00080
	DO 30 I=1,NF	CEM00090
30	IDEN(I,I)=1.	CEM00100
	CALL SMPLY(NF,F,NF,T,T1)	CEM00110
	CALL MCOPY(NF,T1,T3)	CEM00120
	DO 100 I=1,10	CEM00130
	XN=1./FLOAT(12-I)	CEM00140
	CALL SMPLY(NF,T3,NF,XN,T2)	CEM00150
	CALL SMPLY(NF,T2,NF,-1.,T2)	CEM00160
	CALL MADD(NF,IDEN,NF,T2,T2)	CEM00170
100	CALL MPLY(NF,T1,NF,T2,NF,T3)	CEM00180
	CALL MPLY(NF,PHI,NF,T2,NF,T3)	CEM00190
	CALL MPLY(NF,T3,NF,G,NG,GAM)	CEM00200
	CALL SMPLY(NF,GAM,NG,T,GAM)	CEM00210
	RETURN	CEM00220
	END	CEM00230

```

C
C
C THIS SUBROUTINE DOES A NONLINEAR SIMULATION USING THE
C PUNGA-KUTTA INTEGRATION TECHNIQUE
C
      SUBROUTINE RKINT(AOA,TC,VEL,ALT,CB,CF,CI,XT,UT)
      DIMENSION PC(8,1),CB(2,4),CF(2,2),CI(2,2),C1(11,1),A1(11,3),
1  A2(3,18),A(14,8),X(4,1),XNEW(4),XOUT(4),U(2,1),U2(2),UOUT(2),
2  DY(2,1),DYT(2,1),DX(4),DELX1(4),DELX2(4),DELX3(4),XT(4),UT(2),
3  DELX4(4),TEMP1(2,1),C2(3,1),FC1(3,1),PC2(18,1),Q0(4),Q1(4),Q2(4),
4  Q3(4),Q4(4),X1(4),X2(4),X3(4),X4(4),X0(4)
      COMMON/CYCCEF/CY,CY0,CYB,CYDR,CYDA,CYP,CYR
      COMMON/CNCEEF/CN,CN0,CNB,CNDR,CNDA,CNP,CNR
      COMMON/CICCEF/CL,CLO,CLB,CIDR,CLEA,CIF,CLR
810  FORMAT(' ','INPUT: SECONDS FOR TIME HISTORY (INTEGER)')
      ISFC=5
      ITER=10*ISFC
      ICNT=ITER/50
      INDTF=0
C
C INPUT COMMAND VECTOR
C
803  FORMAT(' ','INPUT: COMMAND VECTOR - (Y1,Y2)')
      DY(1,1)=10.
      DY(2,1)=0.
605  DO 115 I=1,2
115  DY(I,1)=DY(I,1)*3.14159/180.
C
C INPUT INITIAL CONDITION
C
      DO 110 I=1,4
110  X(I,1)=XT(I)
C
C COMPUTE INITIAL CONTROL
C
      CALL MMDLY(2,CF,2,DY,1,U)
      DO 112 I=1,2
112  U(I,1)=U(I,1)+UT(I)
C
C SET TIME AND TIME STEP
C
      TIME=0.
      TSTEP=.1
804  FORMAT(' ','TIME',4X,'YAW RATE',3X,'SIDESLIP',3X,'ROLL RATE',
1  1X,'ROLL ANGLE',3X,'RUDDER',5X,'AILERON')
805  FORMAT(' ','-----',4X,'-----',3X,'-----',3X,'-----',
1  1X,'-----',3X,'-----',5X,'-----')
806  FORMAT(' ',F5.2,6(3X,F3.3))
815  FORMAT(' ','NONLINEAR TIME HISTORY')
816  FORMAT('0','ALL OUTPUT IN LEGPES OR DEG/SEC')
      WRITE(6,815)
      WRITE(6,816)
      WRITE(6,804)
      WRITE(6,805)
      DO 120 I=1,4

```


120	XOUT(I)=(X(I,1)-XT(I))*180./3.14159	RKI00560
	DO 122 I=1,2	RKI00570
122	UOUT(I)=(U(I,1)-UT(I))*180./3.14159	RKI00580
	WRITE(6,806)TIME,(XOUT(I),I=1,4),(UOUT(J),J=1,2)	RKI00590
	DO 130 I=1,4	RKI00600
130	QO(I)=0.	RKI00610
C		RKI00620
C	ITERATE TIME HISTORY	RKI00630
C		RKI00640
	DO 200 K=1,ITER	RKI00650
	INDCTR=INDCTR+1	RKI00660
C		RKI00670
C	PEDIMENSION X AND U FOR USE IN DNDYN SUBROUTINE	RKI00680
C		RKI00690
	DO 210 I=1,4	RKI00700
210	XO(I)=X(I,1)	RKI00710
	DO 212 I=1,2	RKI00720
212	U2(I)=U(I,1)	RKI00730
C		RKI00740
C	DELX1	RKI00750
C		RKI00760
	CALL DNDYN(ACA,TC,VEL,ALT,XO,U2,DX)	RKI00770
	DO 215 I=1,4	RKI00780
215	DELX1(I)=DX(I)*TSTEP	RKI00790
	DO 217 I=1,4	RKI00800
217	X1(I)=XO(I)+.5*DELX1(I)	RKI00810
C		RKI00820
C	DELX2	RKI00830
C		RKI00840
	CALL DNDYN(ACA,TC,VEL,ALT,X1,U2,DX)	RKI00850
	DO 220 I=1,4	RKI00860
220	DELX2(I)=DX(I)*TSTEP	RKI00870
	DO 222 I=1,4	RKI00880
222	X2(I)=X1(I)+.5*DELX2(I)	RKI00890
C		RKI00900
C	DELX3	RKI00910
C		RKI00920
	CALL DNDYN(ACA,TC,VEL,ALT,X2,U2,DX)	RKI00930
	DO 225 I=1,4	RKI00940
225	DELX3(I)=DX(I)*TSTEP	RKI00950
	DO 227 I=1,4	RKI00960
227	X3(I)=X2(I)+DELX3(I)	RKI00970
C		RKI00980
C	DELX4	RKI00990
C		RKI01000
	CALL DNDYN(ACA,TC,VEL,ALT,X3,U2,DX)	RKI01010
	DO 230 I=1,4	RKI01020
230	DELX4(I)=DX(I)*TSTEP	RKI01030
	DO 232 I=1,4	RKI01040
232	X4(I)=X3(I)+(1./6.)*(DELX1(I)+2*DELX2(I)+2*DELX3(I)+DELX4(I))	RKI01050
C		RKI01060
C	COMPUTE X(K+1)	RKI01070
C		RKI01080
	DO 235 I=1,4	RKI01090
235	X(I,1)=X4(I)-XT(I)	RKI01100

C		RK101110
C COMPUTE NEW TIME		RK101120
C		RK101130
TIME=TSTEP*FLOAT(K)		RK101140
C		RK101150
C COMPUTE NEW CONTROL		RK101160
C		RK101170
CALL MMPLY(2,CB,4,X,1,U)		RK101180
CALL MMPLY(2,CF,2,DY,1,TEMP1)		RK101190
CALL MADD(2,U,1,TEMP1,U)		RK101200
DYT(1,1)=DY(1,1)*TIME		RK101210
DYT(2,1)=DY(2,1)*TIME		RK101220
CALL MMPLY(2,CI,2,DYT,1,TEMP1)		RK101230
CALL MADD(2,U,1,TEMP1,U)		RK101240
DO 237 I=1,2		RK101250
237 U(I,1)=U(I,1)+UT(I)		RK101260
DO 238 I=1,4		RK101270
238 X(I,1)=X(I,1)+XT(I)		RK101280
C		RK101290
C OUTPUT TIME, X, AND U		RK101300
C		RK101310
IF(INDCTR.NE.IOUT) GOTO 200		RK101320
DO 240 I=1,4		RK101330
240 XOUT(I)=(X(I,1)-YT(I))*180./3.14159		RK101340
DO 242 I=1,2		RK101350
242 UOUT(I)=(U(I,1)-UT(I))*180./3.14159		RK101360
WRITE(6,806)TIME,(XOUT(I),I=1,4),(UOUT(J),J=1,2)		RK101370
INDCTR=0		RK101380
200 CONTINUE		RK101390
811 FORMAT(' ', 'ANOTHER COMMAND VECTOR? (1=YES,0=NO)')		RK101400
IF(DY(1,1).EQ.0.) GOTO 610		RK101410
DY(1,1)=0.		RK101420
DY(2,1)=2.		RK101430
GOTO 605		RK101440
610 RETURN		RK101450
END		RK101460

Appendix C

CAS SOFTWARE

The CAS software was developed to implement the control law for actual flight testing aboard the ARA. The software had the following requirements: accept analog inputs on aircraft states and pilot commands; update the gains; compute the control law; and output commands to the control surfaces. The whole software package was limited to 28K of RAM memory in the Micro-DFCS.

The digital flight control system software, pCAS version 6, revision 5, was the CAS program to be implemented on the microprocessor and is presented in Table 15. The software was developed by altering an existing program, pCAS version 4, revision 1 (Ref. 9). It was broken up into four sections--the data structure declarations, the utility routines, the control routines, and the main program. All software was developed using Pascal-MT.

The data structure declaration section set constant values used in hardware initialization and control law calculations. It also declared variable types (real, integer, array, etc) as required by Pascal-MT.

The utility routines were used for interfacing the software with the hardware and are arranged as a set of subroutines.

DELAY10, DELAY30, and Wait1Second created delays in software execution of 10 microseconds, 30 microseconds, and 1 second, respectively. COLDBOOT initialized hardware (clocks, interrupts, input and output ports), zeroed gain matrices and nominal states, and set the gain schedules. WARMBOOT zeroed the controls and armed the interrupts. AnaTEST enabled the operator to check the A/D and D/A converter operation. MDISP22 and MDISP24 displayed 2 by 2 and 2 by 4 matrices, respectively. MMPLY79 and MMPLY718 calculated a single gain each using the gain index and the gain coefficient matrices A_1 and A_2 , respectively.

The control routines were used for updating the gains and computing the control law. SETUP entered the flight condition, computed the flight condition vector, and computed whichever gain was to be updated. It also entered and set the nominal states as required. CONTROL entered the current values of the states, computed the perturbations from the nominal condition, computed the control law, and sent the commands to the control surfaces.

The main program performed the background routine for accepting operator inputs and performing a limited number of tasks including: reinitializing, halting, or breaking the program execution; testing the A/D and D/A operations; and resetting the nominal condition.

The program could operate in two modes--flight test or ground test. The flight test mode operated by accepting operator inputs and outputting short messages on CAS operations. The ground test

mode performed the flight test functions, and it also executed various steps in the control sequence for error checking.

Table 15. CAS Version 6.5 Listing

```

(*****
|
|               | pCAS-1 Digital Flight Control Software System |
|               |
|               | VERSION 6.5 |
|
|*****

```

Program CAS ;

```

(6Z 01700) ( 6K RTP - hardware 9511 )
(60 09000) ( Set Program ORIGIN )
(6R 08100) ( Set location of RAM )

```

```

(*****
( 1. DATA STRUCTURE DECLARATION
(*****

```

```

(----- 1.1 Constant Declarations -----)
CONST

```

```

ADATA = 004;           ( 9511 data port )
ACONTROL = 005;        ( 9511 control/status port )

ADDAH = $3000;         ( memory base for analog board )
ELEloc = $3008;        ( D/A output location- elevator )
FLPloc = $300A;
ALRloc = $3010;        ( ailerons D/A )
RLDloc = $3012;        ( rudder D/A )
SFPloc = $3014;
ENGloc = $3016;

fa = 10;              ( for the of TERMIFLEX display )
prec = 4;              ( number of significant digits )

IMASK = $02;           ( mask for priority interrupt cont )
IMPMASK = $03;         ( enable p.i.c. )
intCONT = $07;         ( device code for p.i.c. )
IAPPloc = $FFE3;       ( RST2 from p.i.c. is vectored to )
                          ( this location by LFM prom )
JMPOP = $C3;           ( 8080/Z80 JMP opcode )

iZER0v = $7FF0;        ( integer rep of 0 volts to D/A )
rZER0v = 2047.0;       ( f.p. rep of 0 volts to D/A ($7FF) )

in2v = 0.0048852;     ( A/D input ==> volts )
v2out = 204.7;         ( volts ==> D/A out )

PCreg = $92;           ( I/O ports command register )
P8255 = $EB;           ( loc of 8255 control/status port )
PORTC = $EA;           ( loc of 8255 data Port C )

Tcontrol = $0F;        ( timers' control word register )
T0counter = $0C;       ( 8253 timer counter #0 )
T1counter = $0D;       ( 8253 timer counter #1 )

Udata = $EC;           ( UART data port number )
Ucontrol = $EB;        ( UART control/status port )
Umode = $4E;           ( UART mode instruction format )
Ucommand = $37;        ( UART command instruction format )

LFM = $0092;           ( re-entry point for LFM monitor )
Zero = 0.0;

Thchan = 1;            ( throttle A/D )
AOAchan = 26;          ( angle of attack )
VELchan = 6;           ( velocity )

RSTchan = 8;           ( roll stick )
PChan = 9;             ( pedals )

```

```

SSchan = 11;
RRchan = 12;      ( roll angle )
RRchan = 13;      ( roll rate )
YRchan = 14;      ( yaw rate )

VZTH = -0.476;    ( A/D volts == TH ca for'd )
VZVEL = 5.1;      ( " == VEL ft/sec )
VZADA = 2.86;     ( " == ADA deg )

VZPD = -0.256;    ( A/D volts == PD in. forward )
VZBST = -8.210;   ( " == RST deg. clockwise )
VZYR = -2.76;     ( " == YR deg/sec )
VZSS = -3.08;     ( " == SS deg )
VZRR = 4.08;      ( " == RR deg/sec )
VZRA = -8.19;     ( " == RA deg )

RLD2v = 0.4150;   ( RLD deg right == " )
ALR2v = -0.5100;  ( ALR deg right == " )

```

```

(-----)
(----- 1.2 Type Declaration -----)
(-----)
TYPE

```

```

TermLine = PACKED ARRAY [0..11] OF CHAR;

```

```

MAT2x2 = ARRAY [1..2,1..2] OF REAL;
MAT2x4 = ARRAY [1..2,1..4] OF REAL;
MAT7x18 = ARRAY [1..7,1..18] OF REAL;
MAT7x9 = ARRAY [1..7,1..9] OF REAL;
VEC9 = ARRAY [1..9] OF REAL;
VEC18 = ARRAY [1..18] OF REAL;

```

```

(-----)
(----- 1.3 Variable Declarations -----)
(-----)
VAR

```

```

chptr : ^char;
intptr : ^integer;
blankline, baden, badprog : termLine;
CR, LF, ACI, END, BUT, comchar : char;
GroundTest : boolean;
MASK, CI, CD : integer;

VEL, ADA, TH, TC, PD, RHD, VEL2, ADA2, TC2, VELck, ADAck : real;
VELnos, ADAnos, THnos, TCck, TCnos : real;
YR, SS, RR, RA, PD, RST, RSTint, RAC, RAnosC : real;
YRnos, SSnos, RRnos, RAnos, PDnos, RSTnos : real;

RLD, ALR, ELE : real;
RLDnos, ALRnos, ELEnos : real;
ELEout, RLDout, ALRout, index, i, index1, flg : integer;
ELEptr, RLDptr, ALRptr : ^integer;

Cf, Cs : MAT2x2;
Cb : MAT2x4;
A1 : MAT7x9;
A2 : MAT7x18;
FC1, CI : VEC9;
FC2 : VEC18;

```

```

(-----)
(-----)
(-----)
(*I Utl65.Pas*)

```

```

(-----)
(-----)
(-----)
(*I Con65.Pas*)

```

```

(-----)
(-----)
(-----)

```

```

(-----)
(-----)
(-----)

```

```

{=====}
BEGIN                                     ( CAS )

    chptr := iJMPloc;                    ( Set interrupt jump )
    chptr := chr(JMPUP);
    intptr := iJMPloc+1;
    intptr := ord( addr(CONTROL) );

    CI := addr(CONIN);                    ( Define addresses for )
    CO := addr(CONOUT);                  ( - Redirected I/O )

    coldboot;                             ( Initialize hardware )
    warmboot;

    REPEAT                                ( MAIN COMMAND LOOP )
        Read( [CI], Comchar );            ( Wait for a command )
        CASE Comchar OF                  ( Interpret the command )
            'A' : anaTEST;
            'B' : INLINE( $F3 / $C3 / UFM );
            'I' : warmboot;
            'H' : writeln( [CO], LF, CR, 'CAS offline!' );
            'R' : BEGIN flg := 1;
                     writeln([CO], CR, 'RESET NIMS!!')
                   END;
            'U' : BEGIN writeln([CO], LF, CR, 'UP AND AWAY');
                     write ([CO], CR, 'HIDE');
                     wait1second;
                   END;
            ELSE BEGIN writeln( [CO], CR, 'baden' );
                     wait1second;
                     warmboot;
                   END;
        END ( case )

    UNTIL Comchar = 'H';                  ( If 'HALT' then exit )
    INLINE( $F3 / $76 )                  ( program and halt Z80 )

END. ( CAS )

{=====}
{===== FINIS !!! =====}
{=====}

```



```

(*****
(
( 2. CAS Utility Subroutines
(
(
(*****

```

```

(-----)
(----- 2.1 Software Delays -----)
(-----)

```

```

Procedure DELAY10; ( delay 10 microseconds )

```

```

BEGIN
    INLINE($E3/$E3)
END;

```

```

Procedure DELAY30; ( delay 30 microseconds )

```

```

BEGIN
    INLINE($E3/$E3/$E3/$E3/$E3/$E3)
END;

```

```

Procedure Wait1Second;

```

```

VAR i : integer;
BEGIN
    FOR i := 1 TO 9999 DO delay30
END; ( Wait10sec )

```

```

(-----)
(----- 2.2 Console I/O -----)
(-----)

```

```

Procedure CENOUT( ch : char );

```

```

BEGIN
    While Input[UDCTRL] & $0001 = 0 DO delay10;
    Output[UDATA] := ch
END;

```

```

Function CENIN : char;

```

```

VAR chin : char;
    ich : integer;
BEGIN
    While Input[UDCTRL] & $0002 = 0 DO delay10;
    ich := ord( Input[UDATA] & $7F );
    chin := chr( ich );
    While Input[UDCTRL] & $0001 = 0 DO delay10;
    Output[UDATA] := chin;
    CENIN := chin
END;

```

```

(-----)
(----- 2.3 Program Initialization -----)
(-----)

```

```

Procedure COLDBOOT ;

```

```

VAR
    i,j,k : integer;
    testmode : char;
    cptr : ^char;

```

```

BEGIN
    Output[TINTCON] := chr( iMASK ); ( mask interrupt controller )
    BLDIE( $ED/$46 ); ( set Z80 interrupt mode 0 )
    DISABLE;

    Output[TCONTR0] := chr( $36 ); ( initialize TIMER0 )
    Output[T0COUNTER] := chr( $8A );
    Output[T0COUNTER] := chr( $00 );
    delay10;

    Output[TCONTR0] := chr( $74 ); ( initialize TIMER1 )
    Output[T1COUNTER] := chr( $10 );
    Output[T1COUNTER] := chr( $27 );
    delay30;

    Output[P&255] := chr( POREG ); ( initialize parallel PORT C )
    Output[PORTC] := chr( $00FF );

    Output[UDCTRL] := chr( $80 ); ( initialize UART )
    delay10;
    Output[UDCTRL] := chr( $80 );
    delay10;
    Output[UDCTRL] := chr( $40 );

```

```

Output[CONTROL] := chr( UNODE );
delay10;
Output[CONTROL] := chr( UCOMPAND );
delay10;

cptr := ADDAM;           { initialize analog board }
cptr := chr( 901 );

FOR i := 1 TO 2 DO BEGIN { zero the 2X2 gain matrices }
  FOR j := 1 TO 2 DO BEGIN
    Cf[i,j] := zero;
    Cs[i,j] := zero;
  END
END;

FOR i := 1 TO 2 DO       { zero the 2X4 gain matrices }
  FOR j := 1 TO 4 DO
    Cb[i,j] := zero;

FOR i := 1 TO 7 DO       { zero the 7x9 gain schedule }
  FOR j := 1 TO 9 DO
    A1[i,j] := zero;

FOR i := 1 TO 7 DO       { zero the 7x18 gain schedule }
  FOR j := 1 TO 18 DO
    A2[i,j] := zero;

```

```

index:=1;
index1:=1;
fig := 1;

```

```

AOAnom:=0.0;
TOnom:=0.0;
VELnom:=0.0;

```

```

YRnom:=0.0;
SSnom:=0.0;
RRnom:=0.0;
RAnom:=0.0;
RAnomC:=0.0;
RAC :=0.0;
PDnom :=0.0;
RSTnom:=0.0;
RSTint:=0.0;

```

(setup gain schedules - A1 and A2)

A1[1,1]:=1.79;	A1[1,2]:=-.06704;	A1[1,3]:=.0008409;
A1[1,4]:=-.007062;	A1[1,5]:=-.0002718;	A1[1,6]:=.000005783;
A1[1,7]:=.001574;	A1[1,8]:=-.00002029;	A1[1,9]:=.00000005743;
A1[2,1]:=.7297;	A1[2,2]:=-.02105;	A1[2,3]:=.0002326;
A1[2,4]:=-.02591;	A1[2,5]:=.0008608;	A1[2,6]:=-.00001064;
A1[2,7]:=.0007078;	A1[2,8]:=-.00003058;	A1[2,9]:=.000000483;
A1[3,1]:=2.155;	A1[3,2]:=-.1052;	A1[3,3]:=.00148;
A1[3,4]:=-.01192;	A1[3,5]:=.0005802;	A1[3,6]:=-.000008376;
A1[3,7]:=.001896;	A1[3,8]:=-.00007971;	A1[3,9]:=.000001183;
A1[4,1]:=7.889;	A1[4,2]:=-.3227;	A1[4,3]:=.004372;
A1[4,4]:=.005739;	A1[4,5]:=.000432;	A1[4,6]:=-.000007653;
A1[4,7]:=-.001968;	A1[4,8]:=.00003013;	A1[4,9]:=.0000001268;
A1[5,1]:=3.254;	A1[5,2]:=-.08524;	A1[5,3]:=.000908;
A1[5,4]:=-.02466;	A1[5,5]:=-.001404;	A1[5,6]:=.0000274;
A1[5,7]:=.001056;	A1[5,8]:=.0001207;	A1[5,9]:=-.000002462;
A1[6,1]:=-3.595;	A1[6,2]:=.1556;	A1[6,3]:=-.002146;
A1[6,4]:=.009345;	A1[6,5]:=-.0006244;	A1[6,6]:=.000009687;
A1[6,7]:=.000536;	A1[6,8]:=-.000005157;	A1[6,9]:=-.00000009082;
A1[7,1]:=.5348;	A1[7,2]:=-.006276;	A1[7,3]:=-.00007038;
A1[7,4]:=-.1531;	A1[7,5]:=.004123;	A1[7,6]:=-.00004765;
A1[7,7]:=.003235;	A1[7,8]:=-.000148;	A1[7,9]:=.000002178;
A2[1,1]:=-1.986;	A2[1,2]:=-.2263;	A2[1,3]:=.06908;
A2[1,4]:=-.1142;	A2[1,5]:=-.0006239;	A2[1,6]:=-.002163;
A2[1,7]:=-.03765;	A2[1,8]:=.1234;	A2[1,9]:=.0008762;
A2[1,10]:=-.00228;	A2[1,11]:=-.00001553;	A2[1,12]:=-.00007741;

```

A2I1,16):=-.0001424;  A2I1,17):=-.000001853;  A2I1,18):=-.000003768;
A2I2,1):=-.05978;      A2I2,2):=-.1526;      A2I2,3):=-.001118;
A2I2,4):=-.009144;      A2I2,5):=-.00001153;  A2I2,6):=-.0001301;
A2I2,7):=-.008944;      A2I2,8):=-.000245;      A2I2,9):=-.000395;
A2I2,10):=-.0000989;    A2I2,11):=-.000006372;  A2I2,12):=-.000001957;
A2I2,13):=-.0004981;    A2I2,14):=-.0001815;    A2I2,15):=-.00003247;
A2I2,16):=-.000005954;  A2I2,17):=-.000003536;  A2I2,18):=-.000000498;
A2I3,1):=-.1887;        A2I3,2):=-.6572;        A2I3,3):=-.005632;
A2I3,4):=-.03318;        A2I3,5):=-.00009288;    A2I3,6):=-.0004252;
A2I3,7):=-.09175;        A2I3,8):=-.03717;        A2I3,9):=-.001522;
A2I3,10):=-.001522;      A2I3,11):=-.00002116;    A2I3,12):=-.00002149;
A2I3,13):=-.00384;        A2I3,14):=-.00266;      A2I3,15):=-.00003302;
A2I3,16):=-.00005423;    A2I3,17):=-.000003215;  A2I3,18):=-.0000002887;
A2I4,1):=-.9162;        A2I4,2):=-3.385;        A2I4,3):=-.01312;
A2I4,4):=-.155;          A2I4,5):=-.0003222;      A2I4,6):=-.002074;
A2I4,7):=-.1234;         A2I4,8):=-.3041;         A2I4,9):=-.003296;
A2I4,10):=-.006569;      A2I4,11):=-.00004196;    A2I4,12):=-.00004993;
A2I4,13):=-.006807;      A2I4,14):=-.001183;      A2I4,15):=-.0001429;
A2I4,16):=-.00003287;    A2I4,17):=-.0000023;    A2I4,18):=-.0000009122;
A2I5,1):=-.4223;        A2I5,2):=-.943;         A2I5,3):=-.02269;
A2I5,4):=-.04981;        A2I5,5):=-.000333;      A2I5,6):=-.0007071;
A2I5,7):=-.06199;        A2I5,8):=-.1051;        A2I5,9):=-.001981;
A2I5,10):=-.003469;      A2I5,11):=-.00002631;    A2I5,12):=-.00004469;
A2I5,13):=-.002308;      A2I5,14):=-.00517;      A2I5,15):=-.00004179;
A2I5,16):=-.0001639;     A2I5,17):=-.0000005309;  A2I5,18):=-.000002211;
A2I6,1):=-.658;         A2I6,2):=-1.412;        A2I6,3):=-.03657;
A2I6,4):=-.06358;        A2I6,5):=-.009501;      A2I6,6):=-.0008178;
A2I6,7):=-.06771;        A2I6,8):=-.0682;        A2I6,9):=-.001415;
A2I6,10):=-.00006433;    A2I6,11):=-.00002001;    A2I6,12):=-.000002753;
A2I6,13):=-.002882;      A2I6,14):=-.003604;      A2I6,15):=-.00007385;
A2I6,16):=-.00001119;    A2I6,17):=-.0000006737;  A2I6,18):=-.0000005247;
A2I7,1):=-8.224;         A2I7,2):=-.6892;        A2I7,3):=-.3341;
A2I7,4):=-.001572;      A2I7,5):=-.004504;      A2I7,6):=-.0001901;
A2I7,7):=-.0363;        A2I7,8):=-.05388;      A2I7,9):=-.002292;
A2I7,10):=-.001104;      A2I7,11):=-.00003346;    A2I7,12):=-.000006604;
A2I7,13):=-.001139;      A2I7,14):=-.002541;      A2I7,15):=-.00000266;
A2I7,16):=-.00007;      A2I7,17):=-.0000007202;  A2I7,18):=-.000001551;

```

```
RND := 0.001946;
```

```

ELEptr := ELEloc;          ( set pointers to D/A locations )
ALRptr := ALRloc;
RUDptr := RUDloc;

```

```

CR := chr($80);  ACK := chr($0A);  END := chr($05);
LF := chr($0A);  EOT := chr($04);

```

```

writeln( [CO], CR, 'Welcome to..');
write ( [CO], CR, ') pCAS ((');
waitisecond;

```

```

writeln( [CO], CR, 'Ground test. ');
write ( [CO], CR, ' or Flight? ');
read ( [CI], testmode );

```

```

IF testmode='G' THEN GroundTest := TRUE
ELSE GroundTest := FALSE;

```

```
END; ( COLDROOT )
```

```

(-----)
(----- 2.4 Program Reinitialization -----)
(-----)

```

```
Procedure WFCROOT ;
```

```

VAR  iptr : 'integer';
      cptr : 'char';

```

```
(-----)
```

```

Procedure ZUT( ip : integer );
VAR  iptr : 'integer';
BEGIN
    iptr := ip;

```

```

END;
(-----)

BEGIN
  DISABLE;
  blankline := ' ';
  baden := 'Entry Error';
  badprog := 'Program Error';

  writeln(CO, CR, blankline);
  writeln(CO, CR, 'ABHIV ');
  write(CO, CR, 'Option ?');

  MASK := 0FFFF;

  ZOUT( ELEloc );          ( zero volts out to controls )
  ZOUT( ALRIloc );
  ZOUT( RLDloc );

  Output(INTCONT) := chr(iUNMASK); ( re-enable p.i.c. )
  ENABLE

END; ( WARMBOOT )
(-----)
(----- 2.5 Analog to Digital Conversion -----)
(-----)
Function ADC(Vchan : integer) : real;

VAR
  status : char;
  cptr : ^char;
  iptr : ^integer;
  istatus, ivar : integer;

BEGIN
  cptr := ADDR+1;
  cptr^ := chr(Vchan);
  cptr := ADDR;
  cptr^ := chr(40);
  iptr := ADDR+4;
  REPEAT
    status := cptr^;
    istatus := ord(status);
  UNTIL TSTBIT(istatus,7);
  ivar := iptr^;
  ivar := SHR(ivar,4);
  ADC := ivar          ( implicit conversion to REAL )

END; ( ADC )
(-----)
(----- 2.6 A/D and D/A Verification -----)
(-----)

Procedure AnaTEST;

VAR
  ach : char;

(-----)
Procedure testA2D;
VAR
  INchan : integer;
  voltsIN : real;

BEGIN
  writeln(CO, CR, '0..9..F ?');
  write(CO, CR, blankline);
  write(CO, CR, 'Chan In = ?');
  read(CI, ach);

  IF (ach)='0') and (ach<'9') THEN inchan:=ord(ach)-ord('0')
  ELSE IF (ach)='A') and (ach<'F') THEN inchan:=ord(ach)-ord('A')+10
  ELSE BEGIN writeln(CO, CR, baden);
             wait1second;
             warmboot;
             exit
           END;

  REPEAT
    VoltsIn:=(ADC(INchan)-rZLvol)*rVvol;
    writeln(CO, CR, 'Volts = ');
    write(CO, CR, VoltsIn);
    read(CI);
  UNTIL (ach)='Q';

```

AD-A128 579

A LATERAL-DIRECTIONAL CONTROLLER FOR
HIGH-ANGLE-OF-ATTACK FLIGHT(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH W A EHRENSTROM MAR 83

3/3

UNCLASSIFIED

AFIT/CI/NR-83-12T

F/G 1/2

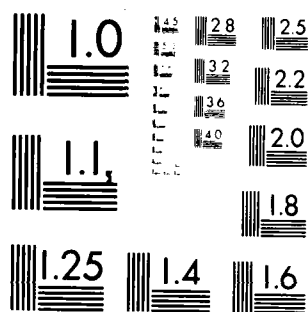
NL



END

DATE
FILMED

* G - R 3
DTIC



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

```

(-----)
Procedure testD2A;
VAR   iptr : integer;
      iVoltsOUT : integer;
      VoltsOUT : real;
BEGIN
  writeln(CD,CR,' 0.5');
  write (CD,CR,blankline);
  write (CD,CR,'CHAN OUT ? ');
  read (CI,ach);
  writeln(CD,CR,'-10.0...+10?');
  write (CD,CR,blankline);
  write (CD,CR,'Volts=');
  readln (CI,VoltsOUT);
  iVoltsOUT := ROUND( (VoltsOUT*v2out) + rZERDv );

  CASE ach OF
    '0' : iptr := ELEloc;
    '2' : iptr := ALRloc;
    '3' : iptr := RUDloc;
  END;

  iptr* := SHL(iVoltsOUT,4) & MASK

END;
(-----)

```

```

BEGIN
  DISABLE;
  writeln(CD,CR,'A/d or B/a');
  write(CD,CR,blankline);
  write(CD,CR,'Which test?');
  read (CI,ach);

  CASE ach OF
    'A' : testA2D;
    'B' : testB2A
  ELSE BEGIN
    writeln(CD,CR,baden);
    wait1second;
    waraboot;
    exit
  END

  END;

  read(CI,ach);
  waraboot
END;

```

```

(-----)
(----- 2.7 Matrix Manipulation -----)
(-----)

```

```

Procedure MDISP22( VAR A : MAT2x2 );
VAR   i,j : integer;
BEGIN
  write(CD,CR);
  FOR i := 1 TO 2 DO BEGIN
    FOR j := 1 TO 2 DO write(CD,A[i,j]:f6:prec,' ');
    writeln(CD,CR)
  END
END;

```

```

Procedure MDISP24( VAR A : MAT2x4 );
VAR   i,j : integer;
BEGIN
  write(CD,CR);
  FOR i := 1 TO 2 DO BEGIN
    FOR j := 1 TO 4 DO write(CD,A[i,j]:f6:prec,' ');
    writeln(CD,CR)
  END
END;

```

```

Procedure MPPLY718(VAR A:MAT7x18; VAR B:MAT7x18; VAR C:MAT7x18; VAR I:integer);
BEGIN
  C[i]:=A[i,1]*B[i,1] +A[i,2]*B[i,2] +A[i,3]*B[i,3] +A[i,4]*B[i,4] +A[i,5]*B[i,5] +A[i,6]*B[i,6] +A[i,7]*B[i,7];

```

```

ACI,10)=BC10) +ACI,11)=BC11) +ACI,12)=BC12) +
ACI,13)=BC13) +ACI,14)=BC14) +ACI,15)=BC15) +
ACI,16)=BC16) +ACI,17)=BC17) +ACI,18)=BC18) ;

```

END;

Procedure PLY79 (VAR A:MAT7x9; VAR B:VEC9; VAR C:VEC9; VAR i:integer);

BEGIN

```

C[i]:=ACI,1)*B[1] +ACI,2)*B[2] +ACI,3)*B[3] +
ACI,4)*B[4] +ACI,5)*B[5] +ACI,6)*B[6] +
ACI,7)*B[7] +ACI,8)*B[8] +ACI,9)*B[9] ;

```

END;

A)


```

(*****)
(      )
(      3. CAS Control Subroutines      )
(      )
(*****)

```

```

(-----)
(----- 3.1 Background Setup Routine -----)
(-----)
Procedure SETUP ;

```

```

(-----)

```

```

Procedure SET_LAT;

```

```

VAR      i : integer;
        ETC,ETC1,ETC2 : real;

```

```

BEGIN

```

```

    TCnom := (ABC(TMchan) - rZERDv) * in2v ;
    ADAnom := (ABC(ADchan) - rZERDv) * in2v ;
    VELnom := (ABC(VELchan) - rZERDv) * in2v ;

```

```

    IF GroundTest THEN BEGIN
    IF index = 100 THEN BEGIN

```

```

        write(CDD,LF,CR,'ADA');
        write(CDD,ADAnom:fs:prec,' ');
        write(CDD,LF,CR,'VEL');
        write(CDD,VELnom:fs:prec,' ');
        write(CDD,LF,CR,'TC');
        write(CDD,TCnom:fs:prec,' ');

```

```

    END;
    END;

```

```

    VELnom := (VELnom * v2VEL + 100.0)*(6.0/3.6);
    ADAnom := (ADAnom * v2ADA + 13.7);
    TCnom := (TCnom * v2TH + 4.53)*(0.23/9.2);
    Q := .5 * VELnom * VELnom * RHD;

```

```

    ( set up flight condition vector )

```

```

    ETC1:= .001 * TCnom;
    ETC2:= ETC1 * ETC1;
    ETC := 1.0 + ETC1 + ETC2/2.0 + (ETC2*ETC1)/6.0 +
           (ETC2*ETC2)/24.0 + (ETC2*ETC2*ETC1)/120.0 +
           (ETC2*ETC2*ETC2)/720.0 ;

```

```

    FC1(1):=ETC;          FC1(2):=FC1(1)*Q;          FC1(3):=FC1(2)*Q;
    FC1(4):=FC1(1)*ADAnom; FC1(5):=FC1(2)*ADAnom;    FC1(6):=FC1(3)*ADAnom;
    FC1(7):=FC1(4)*ADAnom; FC1(8):=FC1(5)*ADAnom;    FC1(9):=FC1(6)*ADAnom;

    FC2(1):=1.0;          FC2(2):=FC2(1)*TCnom;      FC2(3):=FC2(1)*Q;
    FC2(4):=FC2(2)*Q;      FC2(5):=FC2(3)*Q;          FC2(6):=FC2(4)*Q;
    FC2(7):=FC2(1)*ADAnom; FC2(8):=FC2(2)*ADAnom;    FC2(9):=FC2(3)*ADAnom;
    FC2(10):=FC2(4)*ADAnom; FC2(11):=FC2(5)*ADAnom;  FC2(12):=FC2(6)*ADAnom;
    FC2(13):=FC2(7)*ADAnom; FC2(14):=FC2(8)*ADAnom;  FC2(15):=FC2(9)*ADAnom;
    FC2(16):=FC2(10)*ADAnom; FC2(17):=FC2(11)*ADAnom; FC2(18):=FC2(12)*ADAnom;

```

```

    ( calculate gain matrices )

```

```

    IF index = 1 THEN BEGIN

```

```

        i := 1;
        MPPLY79(A1,FC1,C1,i);
        Cx(1,1):=C1(i);

```

```

    END;

```

```

    IF index = 2 THEN BEGIN

```

```

        i := 1;
        MPPLY79(A2,FC2,C1,i);
        Cx(1,2):=C1(i);

```

```

    END;

```

```

IF index1 = 3 THEN BEGIN
    i := 2;
    MPPLY718(A2,FC2,C1,i);
    Cb(1,3):=C1(i);
END;

IF index1 = 4 THEN BEGIN
    i := 3;
    MPPLY718(A2,FC2,C1,i);
    Cb(1,4):=C1(i);
END;

IF index1 = 5 THEN BEGIN
    i := 5;
    MPPLY718(A2,FC2,C1,i);
    Cf(1,1):=C1(i);
END;

IF index1 = 6 THEN BEGIN
    i := 5;
    MPPLY79(A1,FC1,C1,i);
    Cf(1,2):=C1(i);
END;

IF index1 = 7 THEN BEGIN
    i := 6;
    MPPLY718(A2,FC2,C1,i);
    Ca(1,1):=C1(i);
END;

IF index1 = 8 THEN BEGIN
    i := 2;
    MPPLY79(A1,FC1,C1,i);
    Cb(2,1):=C1(i);
END;

IF index1 = 9 THEN BEGIN
    i := 4;
    MPPLY718(A2,FC2,C1,i);
    Cb(2,2):=C1(i);
END;

IF index1 = 10 THEN BEGIN
    i := 3;
    MPPLY79(A1,FC1,C1,i);
    Cb(2,3):=C1(i);
END;

IF index1 = 11 THEN BEGIN
    i := 4;
    MPPLY79(A1,FC1,C1,i);
    Cb(2,4):=C1(i);
END;

IF index1 = 12 THEN BEGIN
    i := 6;
    MPPLY79(A1,FC1,C1,i);
    Cy(2,1):=C1(i);
END;

```

IF index1 = 13 THEN BEGIN

i := 7;
WPLY79(A1,FC1,C1,i);
Cf[2,Z] := C[i];

END;

IF index1 = 14 THEN BEGIN

i := 7;
WPLY79(A2,FC2,C1,i);
Cs[2,i] := C[i];

index1 := 0;

END;

index1 := index1 + 1;

IF GroundTest THEN BEGIN

IF index = 100 THEN BEGIN

write([CO],LF,CR,' ADA:');
write([CO],A0non:fs:prec,');
write([CO],LF,CR,' TC:');
write([CO],TCnon:fs:prec,');
write([CO],LF,CR,' 0:');
write([CO],0:fs:prec,');
write([CO],LF,CR,' VEL:');
write([CO],VELnon:fs:prec,');

writeln([CO],CR);
writeln([CO],CR,' Cf:');
ndisp22(Cf);
writeln([CO],CR,' C1:');
ndisp22(Cs);
writeln([CO],CR,' Cb:');
ndisp24(Cb);

END;

END;

END; (set_lat)

BEGIN

(SETUP)

IF flg = 1 THEN BEGIN

YRnon := (ADC(YRchan) - rZEROv) * in2v;
YRnon := (v2YR * YRnon);
SSnon := (ADC(SSchan) - rZEROv) * in2v;
SSnon := (v2SS * SSnon);
RRnon := (ADC(RRchan) - rZEROv) * in2v;
RRnon := (v2RR * RRnon);
RAnon := (ADC(RAchan) - rZEROv) * in2v;
RAnon := (v2RA * RAnon);

PBnon := (ADC(PBchan) - rZEROv) * in2v;
PBnon := (v2PB * PBnon);
PBnon := PBnon * (15.0/2.38); (scale factor)
RSTnon := (ADC(RSTchan) - rZEROv) * in2v;
RSTnon := (v2RST * RSTnon);
RSTnon := RSTnon * (50.0/80.0); (scale factor)

RAC := 0.0;

flg := 0;

END;

IF GroundTest THEN BEGIN

IF index = 100 THEN BEGIN

write([CO],LF,CR,' YRnon:');
write([CO],YRnon:fs:prec);
write([CO],LF,CR,' SSnon:');
write([CO],SSnon:fs:prec);
write([CO],LF,CR,' RAnon:');

```

write(CD),R0non:fe:prec);
write(CD),LF,CR, R0non );
write(CD),R0non:fe:prec);
write(CD),LF,CR, P0non );
write(CD),P0non:fe:prec);
write(CD),LF,CR, RSTnon );
write(CD),RSTnon:fe:prec);

```

```

END;
END;

```

```

Set_Lat;

```

```

END; ( SETUP )

```

```

(-----)
(----- 4.2 Flight Control Routines -----)
(-----)

```

```

Procedure CONTROL; ( Interrupt Service )

```

```

CONST MINvolts = 00000;
      MAXvolts = 00FFF;

```

```

BEGIN

```

```

  INLINE( 0F5/03E/002/003/007/0E3/0E3/0E3/0E3/0E3/0F1 );
  DISABLE;
  INLINE( 008/009 ); ( Z-80 Register Exchanges )

```

```

  ELEout := MAXvolts;
  ELEptr := SHL(ELEout,4) & MAXL;

```

```

(-----)

```

```

  SETUP;

```

```

  YR := (ADC(YRchan) - rZEROv)*in2v;
  YR := (v2YR * YR);
  SS := (ADC(SSchan) - rZEROv)*in2v;
  SS := (v2SS * SS);
  RR := (ADC(RRchan) - rZEROv)*in2v;
  RR := (v2RR * RR);
  RA := (ADC(RAchan) - rZEROv)*in2v;
  RA := (v2RA * RA);

```

```

  PD := (ADC(PDchan) - rZEROv)*in2v;
  PD := (v2PD * PD);
  PD := PD * (15.4/2.38);
  RST := (ADC(RSTchan) - rZEROv)*in2v;
  RST := (v2RST * RST);
  RST := RST * (50.0/80.0);

```

```

  IF GroundTest THEN BEGIN
  IF index = 100 THEN BEGIN

```

```

    write(CD),LF,CR, YR:');
    write(CD),YR:fe:prec);
    write(CD),LF,CR, SS:');
    write(CD),SS:fe:prec);
    write(CD),LF,CR, RR:');
    write(CD),RR:fe:prec);
    write(CD),LF,CR, RA:');
    write(CD),RA:fe:prec);
    write(CD),LF,CR, PD:');
    write(CD),PD:fe:prec);
    write(CD),LF,CR, RST:');
    write(CD),RST:fe:prec);

```

```

  END;
  END;

```

```

  YR := YRnon - YR;
  SS := SSnon - SS;
  RR := RRnon - RR;
  RA := RAnon - RA;

```

```

  PD := PDnon - PD;
  RST := RSTnon - RST;

```

```

RStint := PAC + RST * .1;
RAC := RStint;

RLD := CFC1,1)*RST+CFC1,2)*PD+Cs1,1)*RStint;
RLD := RLD + (Cb1,1)*YR+Cb1,2)*SS+Cb1,3)*PR+Cb1,4)*RA);

ALR := CFC2,1)*RST+CFC2,2)*PD+Cs2,1)*RStint;
ALR := ALR + (Cb2,1)*YR+Cb2,2)*SS+Cb2,3)*PR+Cb2,4)*RA);
ALR := ALR/2.0;

IF GroundTest THEN BEGIN
  IF index = 100 THEN BEGIN
    write(CO0,LF,CR,' RLD');
    write(CO0,RLD:fw:prec);
    write(CO0,LF,CR,' ALR');
    write(CO0,ALR:fw:prec);

    END;
    END;

    RLD := RLD2v * RLD;
    RLD := (RLD * v2out) + rZER0v;
    ALR := ALR2v * ALR;
    ALR := (ALR * v2out) + rZER0v;

    IF GroundTest THEN BEGIN
      IF index = 100 THEN BEGIN
        write(CO0,LF,CR,' RLDvolts');
        write(CO0,RLD:fw:prec);
        write(CO0,LF,CR,' ALRvolts');
        write(CO0,ALR:fw:prec);

        index := 0;

        END;
        END;

        index := index + 1;

        IF (RLD<=zero) THEN RLDout := MINvolts
          ELSE RLDout := Trunc(RLD);
        IF (RLDout > MAXvolts) THEN RLDout := MAXvolts;

        IF (ALR<=zero) THEN ALRout := MINvolts
          ELSE ALRout := Trunc(ALR);
        IF (ALRout > MAXvolts) THEN ALRout := MAXvolts;

        RLDptr^ := SHL(RLDout,4) & MASK;
        ALRptr^ := SHL(ALRout,4) & MASK;

        (-----)

        ELGout := MINvolts;
        ELEptr^ := SHL(ELGout,4) & MASK;

        INLINE( 0D9/008 );
        INLINE( 0F5/03E/003/0D3/0D7/0F1 );
        ENABLE

        END; ( CONTROL )

        A)

```

REFERENCES

1. Stengel, Robert F., and Nixon, W. Barry, "Investigation of the Stalling Characteristics of a General Aviation Aircraft", Proceedings of the 12th ICAS Congress, Munich, Oct. 1980.
2. Shivers, James P., Fink, Marvin P., and Ware, George M., "Full-Scale Wind Tunnel Investigation of the Static Longitudinal and Lateral Characteristics of a Light Single-Engine Low-Wing Airplane", NASA TN D-5758, June 1970.
3. Suit, William T., "Aerodynamic Parameters of the Navion Airplane Extracted from Flight Data", NASA TN D-6643, March 1972.
4. Anon., USAF Stability and Control Datcom, Wright-Patterson AFB, Ohio: Air Force Flight Dynamics Laboratory, [April 1978].
5. Stengel, Robert F., "Equilibrium Response of Flight Control Systems", Proceedings of the 1980 Joint Automatic Control Conference, Aug. 1980.
6. Foxgrover, John A., "Design and Flight Test of a Digital Flight Control System for General Aviation Aircraft", Princeton University MAE 1559-T, June 1982.
7. Dorato, P., and Levis, A. H., "Optimal Linear Regulators: The Discrete Time Case", IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, Dec. 1971, pp. 613-620.
8. Ben-Israel, A., and Greville, T. N. E., Generalized Inverses, J. Wiley & Sons, New York, 1974.
9. Walters, R. V., informal communications, 1981.

**DATA
FILM**