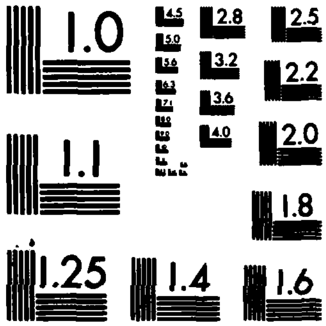
AD-A140 143 COMPUTER PROGRAM TO SIMULATE DIGITAL COMPUTER BASED LONGITUDINAL FLIGHT C..(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J R CARTER DEC 83 1/1

UNCLASSIFIED F/G 1/4 NL



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

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER PROGRAM TO SIMULATE DIGITAL COMPUTER BASED
LONGITUDINAL FLIGHT CONTROL LAWS IN A
HIGH PERFORMANCE AIRCRAFT

by

James Robert Carter

December 1983

Thesis Advisor: M.D. Hewett

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A140 143

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Computer Program to Simulate Digital Computer Based Longitudinal Flight Control Laws in a High Performance Aircraft

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
December 1983

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
James Robert Carter

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93943

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93943

12. REPORT DATE
December 1983

13. NUMBER OF PAGES
72

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Longitudinal Control Laws
Continuous Systems Modeling Program
Component Path
Air Data Schedule
Aliasing Filter
Signal Limiter
Digital Filter

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
The IBM Company's Continuous Systems Modeling Program was used to simulate the longitudinal flight control system of the F/A-18 aircraft. The model is intended for use in investigations of aircraft response to flight conditions which approach spin or stall and is restricted to the automatic flaps up (AFU) flight mode. Program outputs include stabilator deflection, leading and trailing edge flap positions, and cross-axis

20.

interconnect signals. Various stick forces, motion sensor inputs and air pressure inputs were simulated to produce transient control surface responses. These computer generated responses exhibited characteristics corresponding to predicted aircraft control surface movements.

A-1

Approved for public release; distribution unlimited

Computer Program to Simulate Digital Computer Based Longitudinal Flight Control Laws in a High Performance Aircraft

by

James Robert Carter
Lieutenant Commander, United States Navy
B.S., U.S. Naval Academy, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1983

Author: James R. Carter

Approved by: Mark D. Hunt
Thesis Advisor

Donald M. Layton
Chairman, Department of Aeronautical Engineering

J. M. Dyer
Dean of Science and Engineering

ABSTRACT

The IBM Company's Continuous Systems Modeling Program was used to simulate the longitudinal flight control system of the F/A-18 aircraft. The model is intended for use in investigations of aircraft response to flight conditions which approach spin or stall and is restricted to the automatic flaps up (AFU) flight mode. Program outputs include stabilator deflection, leading and trailing edge flap positions, and cross-axis interconnect signals. Various stick forces, motion sensor inputs, and air pressure inputs were simulated to produce transient control surface responses. These computer generated responses exhibited characteristics corresponding to predicted aircraft control surface movements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to acknowledge the the assistance of Adjunct Professor Marle D. Hewett for explanation of theory and definition of the scope of this thesis. Additionally, I wish to acknowledge Lieutenant Scott Graves whose concurrent work in CSMP modeling of the lateral and directional control laws of the F/A-18 contributed to standardization of computer programs.

I. INTRODUCTION

The Navy has experienced the loss of numerous aircraft during recent years due to unintentional departure from controlled flight. The increased cost and complexity of modern aircraft utilizing fly-by-wire flight control systems have placed a renewed emphasis on understanding the performance characteristics of these aircraft, especially near the limits of the flight envelope. The purpose of this thesis is to investigate a method of non-real time computer simulation of the longitudinal flight control system of the F/A-18 aircraft. Future thesis researchers at the Naval Postgraduate School will complete corresponding simulations of the lateral and directional control systems. The combination of these simulations with an existing aerodynamic simulation program will yield a complete aircraft stability and control model. The primary purpose of the model is to investigate methods of designing control augmentation systems which actively inhibit or prevent departures from controlled flight. Other uses of this model would include the capabilities to test new programmable memory configurations, to evaluate new components such as optimal observers, to simulate degraded flight conditions (such as a damaged flight control surface) and to recreate flight conditions during post accident investigations.

The method which was chosen for accomplishment of the objectives of this thesis was the Continuous Systems Modeling Program (CSMP), developed by the IBM Company [Ref. 1]. CSMP is a software package designed to simulate dynamic systems described in terms of differential equations and block diagrams normally encountered in systems theory. CSMP allows programming flexibility through the use of thirty-four pre-programmed functional blocks which are similar to FORTRAN subroutines. These blocks provide rapid access to mathematical functions, switching functions, signal sources, logic functions and FORTRAN functions. Since this thesis represented the first attempt at the Naval Postgraduate School to accurately model the flight control system of a modern, highly augmented tactical aircraft it was deemed important to concentrate on the physical systems rather than become involved in the complexities of numerical analysis.

Alternatives to CSMP which were considered included analog programming and FORTRAN programming. Analog programming was not selected because it is less representative of the systems to be modeled and it is less accurate than CSMP. The concept of programming directly in FORTRAN was carefully considered. Since FORTRAN is the source language for CSMP the capabilities of CSMP are a subset of the capabilities of FORTRAN itself. Additionally, CSMP has restrictions on the number of allowable statements, constants, variables and other parameters. Unlike FORTRAN, when functional blocks in CSMP are used, the programmer has no direct control of mathematical operations internal to the functions. The primary reasons for which CSMP was selected were its simplified input statements, output statements and program control statements which facilitated rapid program writing and testing. Additionally, the automatic time and amplitude scaling, data formatting, and compatibility with graphic display devices which CSMP provides are well suited for prototype program development.

II. METHODOLOGY

A. PROGRAM ORGANIZATION AND NOMENCLATURE

A detailed description of the flight control laws of the F/A-18 aircraft is given in the McDonnell Aircraft Company's system design report [Ref. 2]. The computer program developed in this thesis is based upon figure 16.1 of this report, entitled F/A-18 Longitudinal, Mechanical, CAS, and DEL Control Law Mechanization. This figure contains six pages of block diagrams depicting generation of longitudinal control signals which are valid for all aircraft configurations and failure modes. The version of flight control program incorporated in the flight control computer programmable read only memory (PROM) utilized in this simulation is 8.2.1, dated August 31, 1982.

A brief discussion of the F/A-18 flight control system is necessary to facilitate a discussion of control law modeling. All computations of control laws are accomplished independently by four channels of digital computation. Primary control in the pitch axis is provided by symmetric deflection of the horizontal stabilators. Full span leading edge flaps and trailing edge flaps are scheduled to provide maximum lift to drag ratio during maneuvering, high angle of attack and cruise configurations. Roll control is accomplished by ailerons, differential stabilators and differential leading and trailing edge flap deflection. Directional control is maintained by dual rudders. The thesis computer program which is contained in Appendix A simulates the output of one channel of digital computation and calculates the angular positions of the stabilators, leading edge flaps and trailing edge flaps. This simulation does not calculate rudder or aileron deflection, however, all electrical signals required for cross axis interconnects are provided.

The task of programming the information given in the longitudinal control law mechanization schematic was simplified by two means. First, the program restrictions and assumptions to conditions of flight which are discussed in part B were applied. As a result sections of figure 16.1 which apply to mechanical control laws and spin modes, for example, were deleted. This reduced the number of schematic blocks to be modeled by approximately one third. The second simplification arose through a system of nomenclature in which nine control paths were defined in order to limit the number of input and output signals for any specific path.

When combined, these nine paths form the total longitudinal control law mechanization.

Figure 2.1 is the overall longitudinal control signal block diagram which was used in this simulation. It was derived by applying all program assumptions or restrictions outlined in part B to figure 16.1 of the system design report.

Block diagrams depicting the logic development of the component paths are included in Appendix A. The nomenclature for each control path which is given in Table I is peculiar to this simulation program.

Table II lists nomenclature for control signal groups which are common to both the McDonnell schematic and to this simulation. Signals with common prefixes are numbered consecutively in the feed forward direction. The primary feed forward input to the CSMP simulation is pilot stick force. Feedback signals include pitch rate, roll rate, yaw rate, angle of attack, normal acceleration and differential control surface commands.

B. PROGRAM RESTRICTIONS AND ASSUMPTIONS

The task of obtaining a working computer program to meet the thesis objectives did not require the simulation of all

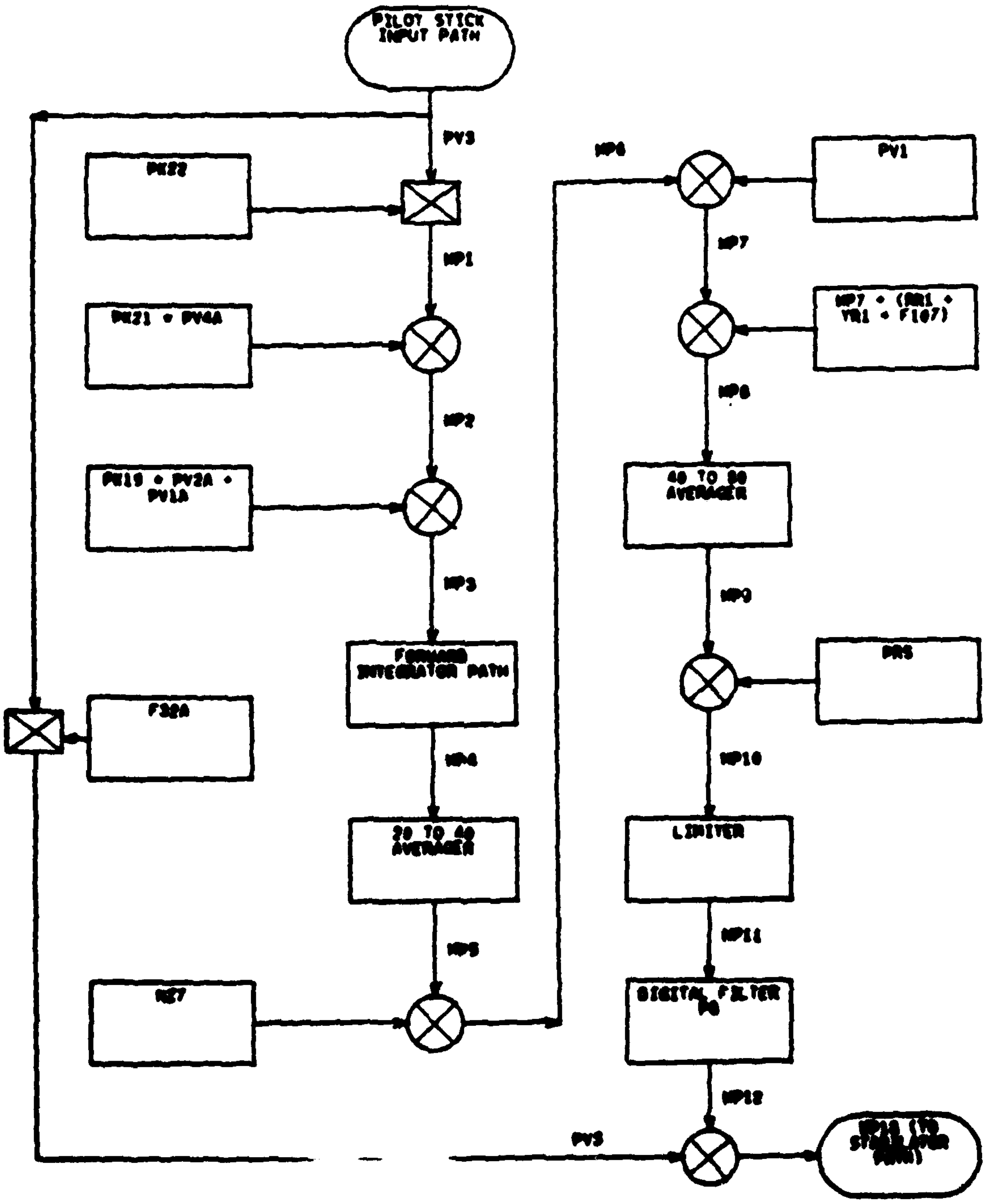


Figure 2.1. Longitudinal Control Laws

TABLE I

Control Signal Paths

PATH	SIGNAL PREFIX
Main path (open loop)	MP
Pilot stick input path	PS
Pitch rate gyro path	PR
Angle of attack sensor path	AA
Normal accelerometer	NZ
Forward integrator path	FI
Stabilator path	ST
Leading edge flap path	LE
Trailing edge flap path	TE

TABLE II

Common Signal Nomenclature

SIGNAL (OR PREFIX)	DESCRIPTION
ALPHAS	Angle of attack (computed)
ALPHAT	Angle of attack (true)
F	Function (air data schedule)
FK	Air data schedule gain parameter
PS	Static pressure
PV	Pitch axis storage location
QC	Compressible dynamic pressure
RI	Pressure quotient (=QC/PS)

possible conditions of flight. Thus, an assumed aircraft configuration and flight condition led to many simplifications in the model. Each major assumption is discussed below. Should future researchers desire to construct a more general model, additional program logic paths could be included in this CSMP simulation without requiring major revisions to the program.

Assumptions:

1. The aircraft flight control electronics set (FCES) is operating normally. Primary components of this set include flight control computers, pitch, roll and yaw rate gyros, accelerometer assemblies and flight control panels. The FCES contains logic sequences for failure detection and corrective control law implementation which were ommitted from the thesis computer program.

2. The aircraft is under inner loop control. In this mode pilot stick force forms the primary control input. Autopilot functions such as heading hold, roll or pitch attitude hold, and altitude hold are not in operation.

3. The pitch control augmentation system (CAS) is providing longitudinal control. Backup control systems which are not in operation include the mechanical backup mode and the direct electric link (DEL) mode which provides an open loop signal from the pilot stick position sensor to the stabilator servoactuator.

4. The aircraft is operating in the up and away flight phase in the auto flaps up (AFU) configuration. This phase requires that the flap switch in the cockpit be in the AUTO position or that the calibrated airspeed be greater than 243

knots. In the AFU mode gain schedules are optimized for combat maneuvering characteristics in the low to mid dynamic pressure regime. Additionally, a trim integrator controls load factor and, at angles of attack above 22 degrees, proportional nose down commands are introduced.

5. The gain switch is in the normal position. The override position of this switch causes flight computers to use fixed values for air data schedules and a predetermined angle of attack in control law computations. The normal position of the gain switch allows measured values of air data and angle of attack to be used.

6. The aircraft is operating with weight off the wheels, speed brake in, and with no external stores.

7. Anti spin functions are not simulated.

III. CONTROL LAW MODELING

A. AIR DATA SCHEDULES

Air data schedules are functions of static pressure (PS), compressible dynamic pressure (QC), and other parameters such as normal acceleration, angle of attack or condition of external stores. The longitudinal control laws contain inputs from 13 different air data schedules which are listed in Table III.

TABLE III

Air Data Schedules

FUNCTION	DESCRIPTION (INPUTS)
F12	Pitch forward loop integrator gain schedule (RI,PS)
F22	Fader on supersonic compensation (QC)
F23	Stall margin on pitch forward loop integrator (AOA)
F24	Trailing edge flap schedule (AOA,RI)
F25	Trailing edge flap schedule (QC limit)
F27	Leading edge flap schedule (AOA,RI)
F28	Leading edge flap schedule (QC limit)
F29U	Leading edge flap schedule (RI limit)
F32A	Longitudinal forward loop gain schedule (QC)
F37	NZ limit on AOA feedback (NZA)
F40	Pitch rate feedback gain schedule (PS,QC,RI)
F68	Pitch rate feedback gain schedule (QC)
F107	Longitudinal inertial gain schedule (QC)

Each schedule controls an output gain for a particular purpose. Function 32A, for example, yields a uniform initial pitch acceleration in response to sharp inputs, with gain decreasing at high values of compressible dynamic pressure. Other functions such as function 29U are used to determine an upper or lower limit to an input signal. Simulation of these functions in the CSMP format required that individual computer programs be written to test each air data schedule. In all cases the final result was given in the flight control system design report, usually in graphical form. Conversion to CSMP format was done by extraction of data points from graphs, use of logic flow charts when available and direct employment of those mathematical formulae which were given. Once programmed in CSMP, the tabular and graphical output data was compared to data given in the design report for the same test parameters. Thus, the output of each function was independently verified before the function was included in the longitudinal control law simulation.

The simulation is written to take advantage of CSMP's SORT and NOSORT capabilities where appropriate. The NOSORT option of CSMP is used for conditional logic and branching. This statement allows the user flexibility in creating sections of the program in which ordinary FORTRAN rules can be used. In NOSORT sections intermediate variables were defined, comparisons made and appropriate branching was executed. The program statements were returned to SORT format as soon as conditional logic was no longer required. An early problem revealed that if two or more SORT sections are separated by a NOSORT section information will not be passed between the separated SORT sections. To prevent errors resulting from this restriction from occuring the following decisions regarding the order of program statements were made.

1. The number of NOSORT sections should be minimized.

2. NOSORT sections should be located close together,allowing fewer and larger SORT sections.

3. Macros should be utilized when possible. Macros, which are similar to subroutines, are discussed in part D under frequency averagers.

The task of arranging program statements to minimize the number of NOSORT sections was greatly simplified through the use of the "output variable sequence" tabulation which is produced as part of the CSMP standard output. The final number of NOSORT statements in the computer simulation could have been further reduced by this method, however, this would have required the movement of large program blocks and caused a deviation from the logic path used in program development. For example, the NOSORT sections of functions

12 and 40 could have been combined. This would have increased difficulty in program debugging and made areas of the program which require combinational logic less apparent to readers.

Appendix C contains documentation for the air data schedules. Each schedule is included in a complete computer program which produces tabular and graphical data to match the characteristics of figures in the design report. In incorporating these air data schedules into the longitudinal control law simulation all schedules except for function 24 were placed before the main computational body of the program. Function 24, the trailing edge flap schedule, requires conditional logic and contains computed angle of attack as an input. Since computed angle of attack is generated in the normal accelerometer path, function 24 was placed immediately before the trailing edge flap path computations.

B. FADERS, ANALOG TO DIGITAL CONVERTERS, MECHANICAL BREAKOUTS

The purpose of faders is to eliminate large discontinuities in gain, permitting gradual change in output from old values to new new values at a desired sample rate.

Faders are located in Figure 16.1 at the outputs of air data schedules and gain schedules which are dependent upon aircraft configuration. Discontinuities may arise as a result of a change in physical measurements such as dynamic pressure, change in aircraft configuration such as speed brake extension, or change in mode of flight. The lower limit for signal MP4, for example, changes from a gain of -50.0 to 0.0 as a result of spin entry. Since the thesis computer program assumes that electrical signals vary smoothly and is restricted to up and away controlled flight, faders are modeled in frequency only.

Analog to digital conversion and frequency matching are obtained by a sample and hold process. Sampling times are generated by the CSMP functional block IMPULS which produces a time series of unit impulse functions with a specified start time and period. Since these pulse trains are used in several areas of the program, impulse functions of 20, 40 and 80 hertz are included immediately following the air data schedules. The zero order hold function ZHOLD keeps signal gain constant throughout the pulse period.

A mechanical breakout force of two pounds is modeled in the pilot stick input path. When large stick forces are applied, the resultant signal is calculated with a reduction in magnitude of two pounds. This modeling is accomplished in CSMP by the deadspace (DEADSP) functional block. Figure 16.1 of the design report depicts electrical signal "deadbands" which are conceptually identical to mechanical breakouts. Appendix D contains a program which displays deadspace outputs in tabular and graphical form.

C. ALIASING FILTERS AND SIGNAL LIMITERS

The longitudinal control laws of the F/A-18 include five aliasing filters which are modeled as first and second order Laplace transforms. The first order transform is of the type $A/(Bs+1)$ and is in the pilot stick input path. This lag type filter with one real pole is converted to CSMP format by the functional block REALPL. A required initial condition is the value of the output signal when time is zero. This may be determined arbitrarily by the user, but was set to zero for this simulation.

Second order filters are present in the pilot stick input path, pitch rate gyro path, angle of attack sensor path and normal accelerometer path. Each is of the form $A/(Bs^2+Cs+D)$ and represents an underdamped system. The CSMP functional block for complex poles (CMPXPL) is used.

Initial conditions are the value of the output signal and time rate of change of the output signal when time equals zero. Natural frequencies varied from 4.34 hertz in the pilot stick input path to 33.3 hertz in the angle of attack sensor path. The appendix contains a computer program which demonstrates the CSMP outputs of both first and second order aliasing filters. Unit step inputs were introduced to each filter to generate transient responses. The first order filter produced an exponential rise to steady state with the correct time constant. Characteristics of the second order filters such as rise time, peak time, maximum overshoot, and settling time compared favorably with theoretical results [Ref. 3].

Signal limiters restrict the maximum or minimum values of an output signal. Stabilator surface deflection, for example, is limited to 10.5 degrees trailing edge down and 24 degrees trailing edge up. The CSMP functional block LIMIT allows direct specification of lower and upper signal limits.

D. FREQUENCY AVERAGERS AND RATE LIMITERS

A characteristic of the F/A-18 flight control system is that various signal paths operate at frequencies of 20, 40 or 80 hertz. When signals are combined mathematically, the inputs are first converted to a common frequency. Normal accelerometer path signals are computed at 40 hertz and combined with outputs of the forward integrator path, which operates at 20 hertz. In this case a 20 to 40 hertz averager is present between the integrator path and the summing junction connected to the normal accelerometer path. The algorithm used for the 20 to 40 hertz averager is based on a procedure given in the Flight Control Electronic System Report [Ref. 4]. The averager was required to generate signal values at twice the rate of the incoming pulses by linear interpolation between amplitudes of the two previous signals at 20 hertz. The formulas used to generate intermediate signals were:

$$Z40 = Z40Z1 + DEL \tag{3.1}$$

$$DEI = (Z20 - Z20Z1) / 2.0 \tag{3.2}$$

where:

Z40 = current value of the 40 hertz signal
Z40Z1 = previous value of the 40 hertz signal
Z20 = current value of the 20 hertz signal
Z20Z1 = previous value of the 20 hertz signal.

Conditional logic was used to keep the output signal equal to the input signal at times when the 20 hertz impuse function was equal to one. The algorithm was initialized by letting "previous values" be equal to input values when time equals zero.

The presence of numerous frequency averagers in the control laws would have required the repetition of many statement blocks without the use of CSMP program MACROS, which are similar to FORTRAN subroutines. Frequency averagers in the thesis computer simulation were included in MACROS and placed at the beginning of the program. A MACRO may be used several times within a program. Input and output variables are given dummy names, yet the MACRO is invoked with a unique name which is assigned in a function definition statement. A limitation to the use of MACROS is that variables which are defined in MACRO structure statements are not available for output unless they are designated as arguments in the function definition statement. Additionally, certain functional blocks such as REALPL, CMPXPL and INTGRL cannot be used as arguments or as parts of a MACRO structure statement.

The PROCEDURE function of CSMP was used in each MACRO to cause statements to be executed in the order of their appearance. Each structure statement of a MACRO will be individually sorted unless PROCEDURE is specified. All statements contained within a PROCEDURE function are treated as a single block which can be moved but not rearranged by the CSMP translator.

FORTRAN subprograms were not used in this simulation because the size of the longitudinal portion of the program did not approach the maximum limits of CSMP. Inclusion of the lateral and directional control systems in this program will require that measures be taken to remain within the allowable number of structure statements, NOSORT sections and MACROS. The CSMP translator is not used to process FORTRAN subprogram statements. Since the number of subprogram statements is not counted the overall size of the program may be increased. The capabilities of the computer system library may be utilized by subprograms through the use of the CALL statement. The CALL statement must, however, be included in a NOSORT or PROCEDURE section. A method to invoke subprograms exists which does not require NOSORT or PROCEDURE sections, but it is valid only for two or more output variables and it requires a specific format for arguments. A final restriction to the use of FORTRAN subprograms is that certain CSMP functional blocks such as ZHOLD, IMPULS, and CMPXPL are not allowable.

The longitudinal control laws contain rate limiters which operate at frequencies of 20 hertz and 80 hertz to restrict the speed of leading edge flap movement. The algorithm for these limiters compares the value of each incoming signal with the value of the previous output signal. The magnitude of the difference between these signals is processed by the LIMIT functional block which generates the current output signal. The 20 hertz rate limiter, for example, allows a maximum change in output signal value of 0.9 degrees during each period of 0.05 seconds. A listing of the program which was used to test the rate limiter is contained in appendix D. Correct outputs were observed for both increasing and decreasing input signals.

E. DIGITAL FILTERS, DIGITAL TO ANALOG CONVERTERS, SERVOMECHANISMS

Three types of digital filters are used in the longitudinal control laws. Z filter number F2 is a lead-lag type controller in the pitch rate gyro path which operates at 20 hertz. Z filter number P9, the forward loop integrator, operates at 20 hertz and compares the aircraft response to the maneuver command. The output signal drives the stabilator servoactuator to reduce the maneuver error to zero. This allows the aircraft to to be automatically kept in a hands off condition since the forward integrator eliminates uncommanded normal acceleration. Z filter number P8 is a structural notch filter which operates at 80 hertz. It attenuates aeroelastic bending which is detected by the motion sensors.

In the simulation each filter was developed in its most general form for inclusion in a MACRO. The following equation for lead-lag filter number P2 is given in the Schematic Design Report [Ref. 5].

$$\frac{PR4}{PR3} = \frac{(1+PK11*(1-PK12))Z - (PK11+1)*(1-PK12)}{Z - (1-PK12)} \qquad (3.3)$$

It is modeled in the thesis simulation as

$$\frac{FOUT}{FIN} = \frac{AZ - B}{Z - C} \qquad (3.4)$$

where A, B and C are constants. The right shifting and linearity properties of the z transform are used to solve explicitly for the variable FOUT [Ref. 6].

$$\frac{FOUT}{FIN} = \frac{A - B(Z-1)}{1.0 - C(Z-1)} \tag{3.5}$$

Cross multiplication and rearrangement yields:

$$FOUT = A*FIN - B*FIN(Z-1) + C*FOUT(Z-1) \tag{3.6}$$

which is described in the simulation as:

$$FOUT = A*FIN - B*FINZ1 + C*FOUTZ1 \tag{3.7}$$

This method, which is termed direct realization programming, was also used in the development of the notch filter and the forward loop integrator. The equivalent Laplace transform for each longitudinal flight control filter is listed in the design report. This permitted a cross check of z filter performance which is included in appendix D.

A specific method of integration may be specified in the terminal portion of a CSMP program. In the case of flight control simulation the Runge-Kutta Fixed Step Size (RKSFX) method was utilized to ensure that integrations would only occur at the desired sampling rate. The highest sampling frequency in any axis of the F/A-18 flight control system is 80 hertz, thus the CSMP integration interval DELT was specified as 0.0125 seconds.

The variable KEEP is used in CSMP to indicate that the end of a valid integration step has been reached. KEEP is set equal to one when this condition is met. During trial or intermediate integration steps KEEP will equal zero. Each MACRO contains conditional logic which allows calculations to be performed only when KEEP equals one.

Conversion of signals from analog to digital form in the F/A-18 occurs as the signal reaches the servomechanism. The quantizer functional block (QNTZR) is employed to accomplish the analog to digital simulation. The transfer functions used by the stabilator and flap servomechanisms are not published by the manufacturer of the aircraft. In order for the thesis computer program to generate control surface positions the response characteristics of a Parker Hannifin fly-by-wire actuation system were incorporated. The selected actuators were designed for use with all-digital flight controls and are modeled as second order systems. The stabilator transfer function is

$$\frac{X}{Ei} = \frac{1600}{S^2 + 56S + 1600} \qquad (3.8)$$

where X is the actuator position in degrees and Ei is the position command. The transfer function for both leading and trailing edge flaps is

$$\frac{X}{Ei} = \frac{400}{S^2 + 28S + 400} \qquad (3.9)$$

The advantages of using the second order model instead of a first order model are that the faster rise time more closely represents the physical actuators and that the natural frequency and damping ratio may be independently modified.

F. PROGRAM TESTING METHODS AND RESULTS

The thesis computer program was tested on three levels. The lowest level involved evaluation of the individual signal blocks of figure 16.1 in the system design report. Sections A through E of this chapter describe signal block modeling techniques and appendicies C through G contain testing programs. To obtain verification of proper program operation it was desired to create input signals of realistic value which would produce a time varying output. The rate limiters, for example, were tested with an input signal which rose exponentially to a limiting value, then decreased exponentially. Output signals at the desired frequency were observed for incoming signals of positive or negative slope and for incoming signals which were within or beyond the rate limit.

The second level of program testing involved the nine signal paths which are listed in Table 1. In most cases a step function was used as the input signal. Intermediate signals were observed to determine continuity, frequency, time constants, and conformance with limiting values.

The highest level of program testing required the generation of control surface deflections for specific combinations of pilot stick force and motion sensor feedback signals. Neither aircraft flight data nor the McDonnell flight control simulation data were available for direct comparison with the outputs of the thesis program. The description of control characteristics given in the design report was used to make a qualitative analysis of program operation.

One specific condition of flight was selected as a basis for comparison of control responses to various combinations of input signals. This "base condition" of flight was defined such that inputs from motion sensors could be superimposed on base condition inputs to enable an investigation of the effects of each parameter. The base condition for program testing was selected to model an

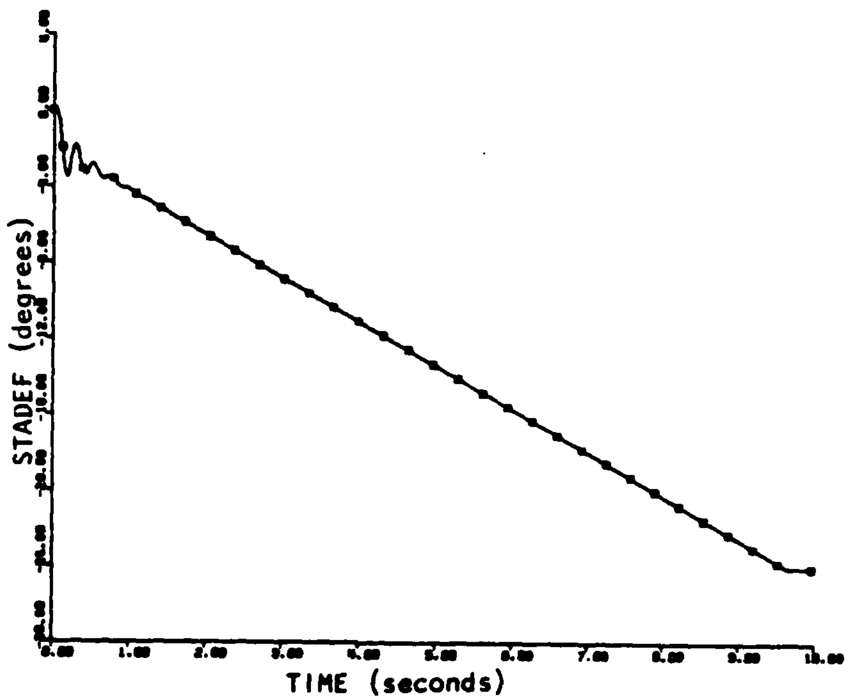


Figure 3.1. Base Condition Stabilator Response

aircraft operating at 20,000 feet and 250 knots. This fixed the values of static and dynamic pressures for each simulation and permitted manual verification of the gains produced by the air data schedules. Additionally, a step function representing six pounds of force in the aft (positive) direction on the control stick was applied at time 0.0 seconds. All motion sensor inputs were held at zero so that their effects could be individually studied. The initial deflections of the stabilator, leading edge flaps and trailing edge flaps were set to zero degrees and all initial conditions for filters were set to zero.

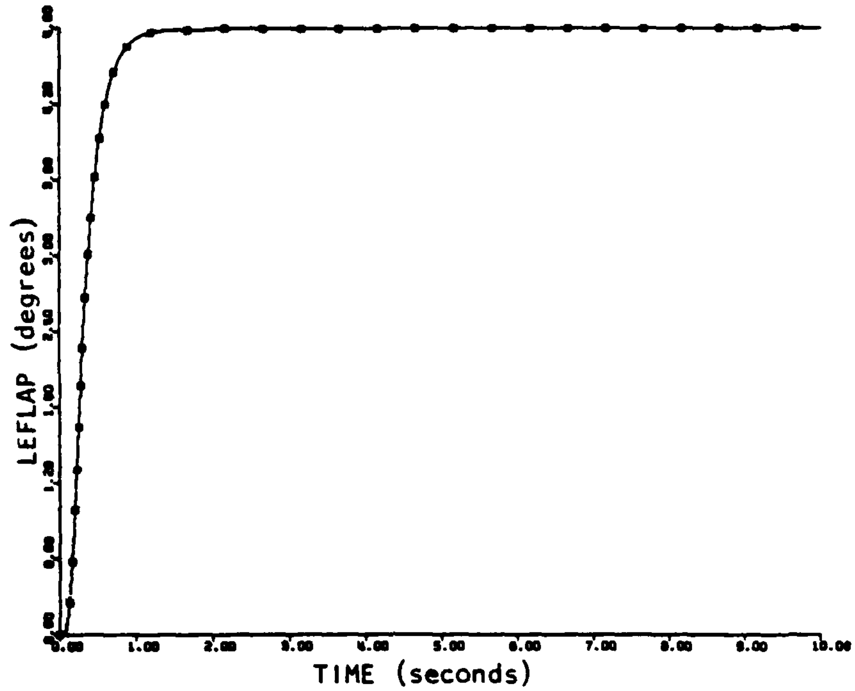


Figure 3.2. Base Condition Leading Edge Flap Response

Figure 3.1 depicts the movement of the stabilator in response to the base flight conditions. A transient oscillation is produced in the first second which results primarily from the second order filter in the pilot stick dynamics path. After this oscillation has decayed the stabilator continues to deflect at a nearly constant rate, since feedback inputs are supressed. The stabilator reaches the limiting value of 24 degrees trailing edge up after 9.6 seconds.

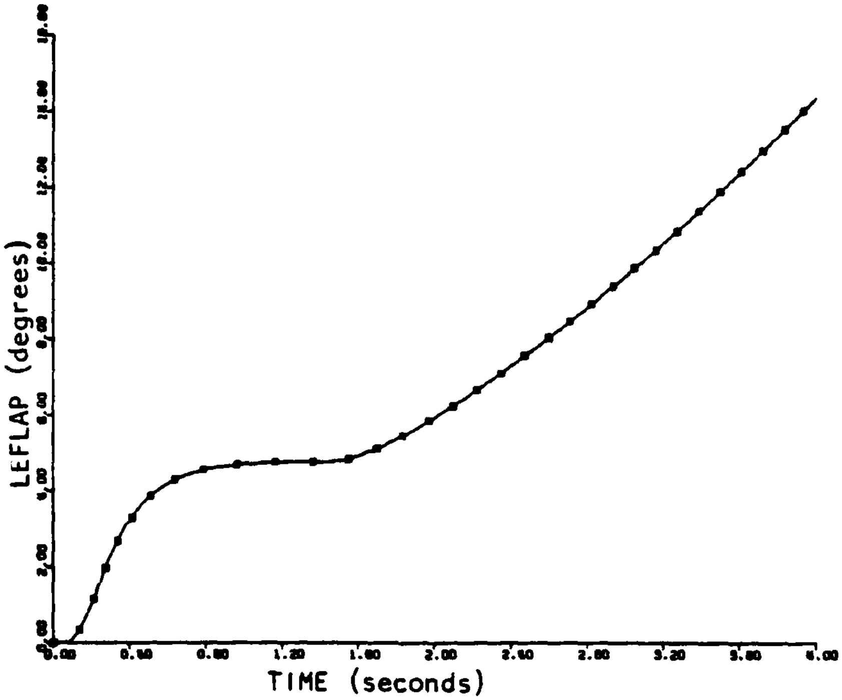


Figure 3.3. Leading Edge Flap Response to AOA Feedback

Figure 3.2 is a plot of leading edge flap deflection (LEFLAP) versus time for the base flight condition. A steady state flap deflection of 4.8 degrees is achieved after 1.5 seconds. Since LEFLAP is a function only of angle of attack, static pressure and dynamic pressure it was desired to observe the variation in LEFLAP with angle of attack. A ramp type increase in angle of attack sensor input (AA1) was superimposed upon the base flight condition beginning at time 1.4 seconds. The resulting schedule of leading edge flap deflection is shown in Figure 3.3 and is consistent with the functional description given in the design report.

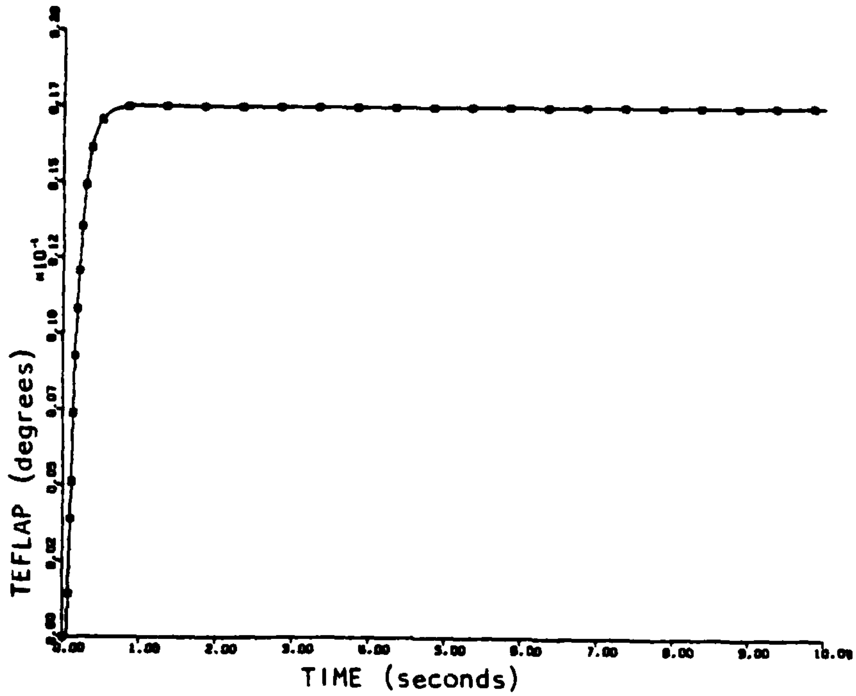


Figure 3.4. Base Condition Trailing Edge Flap Response

Trailing edge flap deflection was modeled similarly in Figure 3.4 for the base filght condition and in Figure 3.5 for the condition in which AA1 is a ramp function starting at time 1.4 seconds.

To determine the effect of forward pressure on the pilot stick a step input of -12.0 pounds was superimposed upon the base flight condition at time 4.0 seconds, which simulated an instantaneous reversal of stick force. Figure 3.6 shows that the direction of stabilator deflection changed abruptly upon introduction of the new stick force, and that the

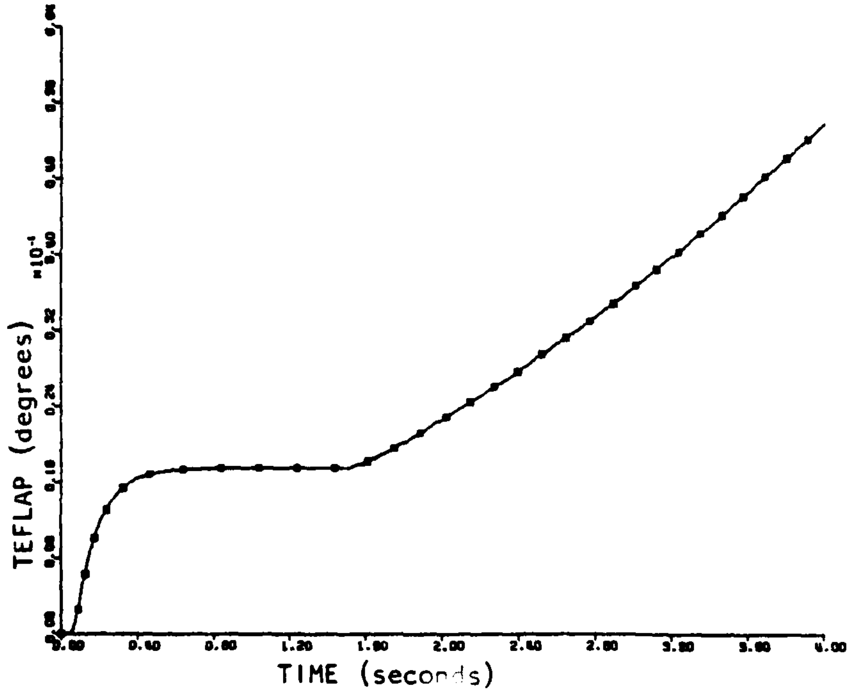


Figure 3.5. Trailing Edge Flap Response to AOA Feedback

magnitude of the steady state rate of stabilator deflection was approximately unchanged. Pilot stick force inputs of larger magnitude were simulated, but the results are not shown. In these cases the rates of stabilator deflection increased while initial oscillatory behavior exhibited characteristics similar to base condition response.

The influence of angle of attack feedback on stabilator position is displayed in Figure 3.7. As described in the design report, angles of attack in excess of 22 degrees will generate a proportional stabilator command causing the nose

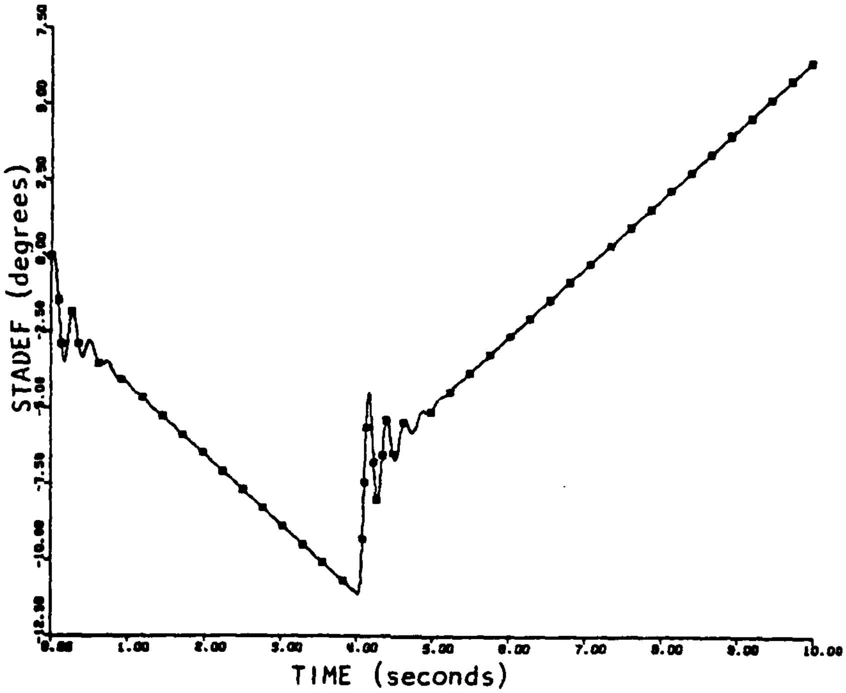


Figure 3.6. Stabilator Response to Stick Force Reversal

of the aircraft to pitch down. In this simulation the angle of attack input function AA1 was set equal to 22 degrees plus a ramp type increase of one degree per second starting at time 4.0 seconds. The resulting stabilator deflection was identical to that for the base condition until time 4.0 seconds due to the action of the -22 degree bias in the AOA feedback path. The nose down pitching effect of AOA feedback is apparent at all later times. In response to the base condition the stabilator had reached its maximum limit of 24 degrees trailing edge down at time 9.5 seconds. The application of AOA feedback restricted stabilator deflection to 20.9 degrees at time 9.5 seconds.

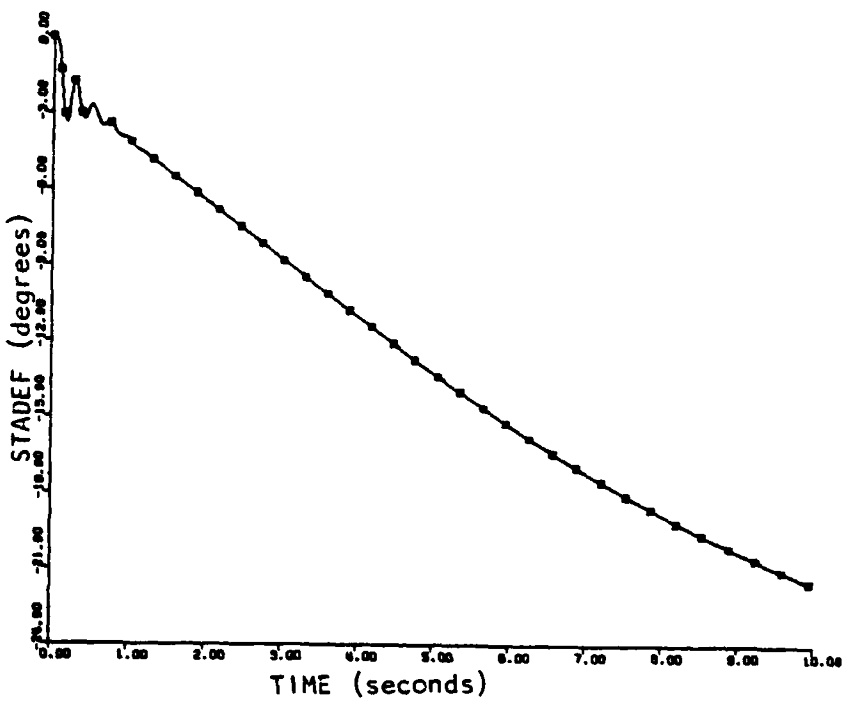


Figure 3.7. Angle of Attack Damping

Normal acceleration damping is shown in Figure 3.8. A ramp increase in normal acceleration equal to 1.0 g's per second was superimposed on the base condition at time 4.0 seconds. The output signal varies smoothly due to the fact that normal acceleration path outputs are processed by the forward integrator path. The stabilator reverses its direction of movement within one second of the time feedback is introduced. In this case the stabilator reached its limit of deflection of 10.5 degrees trailing edge down at time 9.0 seconds.

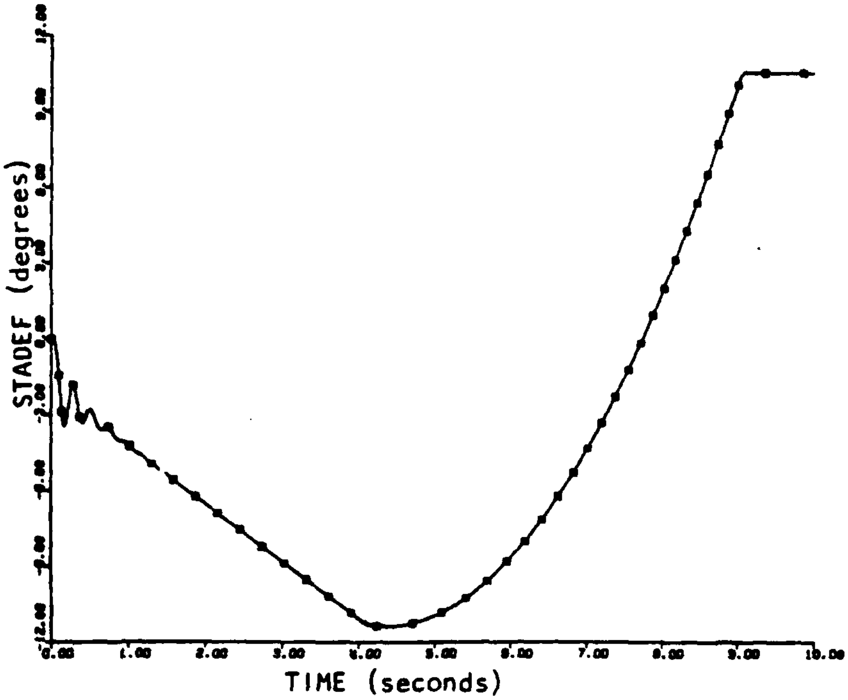


Figure 3.8. Normal Acceleration Damping

According to the design report, the predominant contribution to pitch damping is generated by the pitch rate gyro path. This is because the pitch rate signal PR5 is summed with the main path signal MP9 downstream of the forward integrator as shown in Figure I. Figure 3.9 depicts stabilator position versus time for a ramp increase in pitch rate beginning at time 4.0 seconds. The change in direction of stabilator movement is much more rapid than that produced by angle of attack or normal acceleration damping.

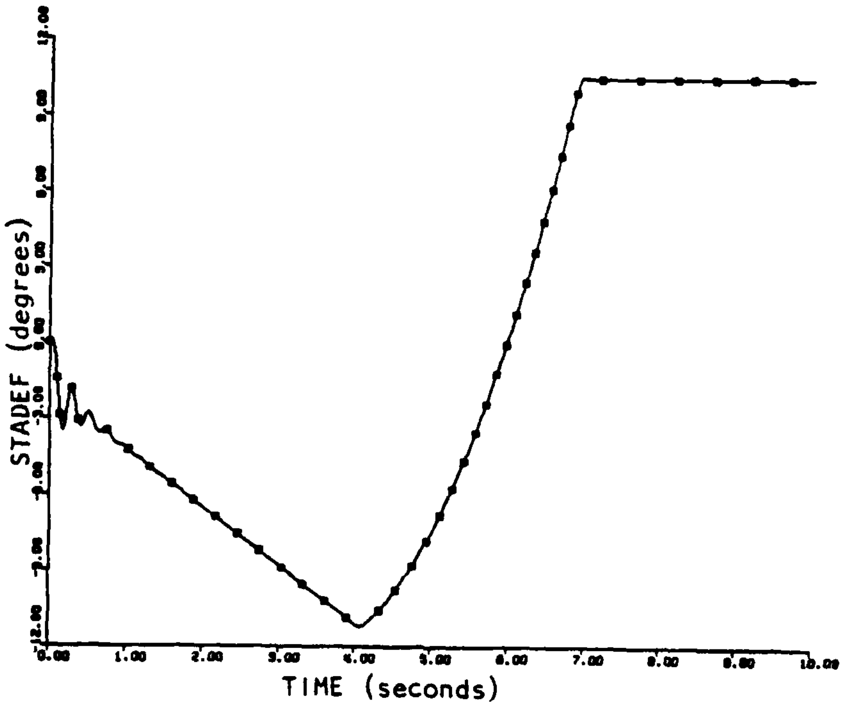


Figure 3.9. Pitch Rate Damping

The thesis computer program used a significant portion of allowable CSMP program size. Table IV, which is extracted from the "translation table" section of CSMP output, lists the areas of the program which most closely approached the size limits of CSMP.

TABLE IV

Program Size Restrictions

Parameter	Current Program	CSMP Maximum
MACRO and statement outputs	206	600
Statement input work area	422	1900
Parameters-function generators	43	400
History and memory block names	21	50
MACRO statement storage	85	125
SORT sections	5	20
Maximum statements in section	171	600

IV. CONCLUSIONS AND RECOMMENDATIONS

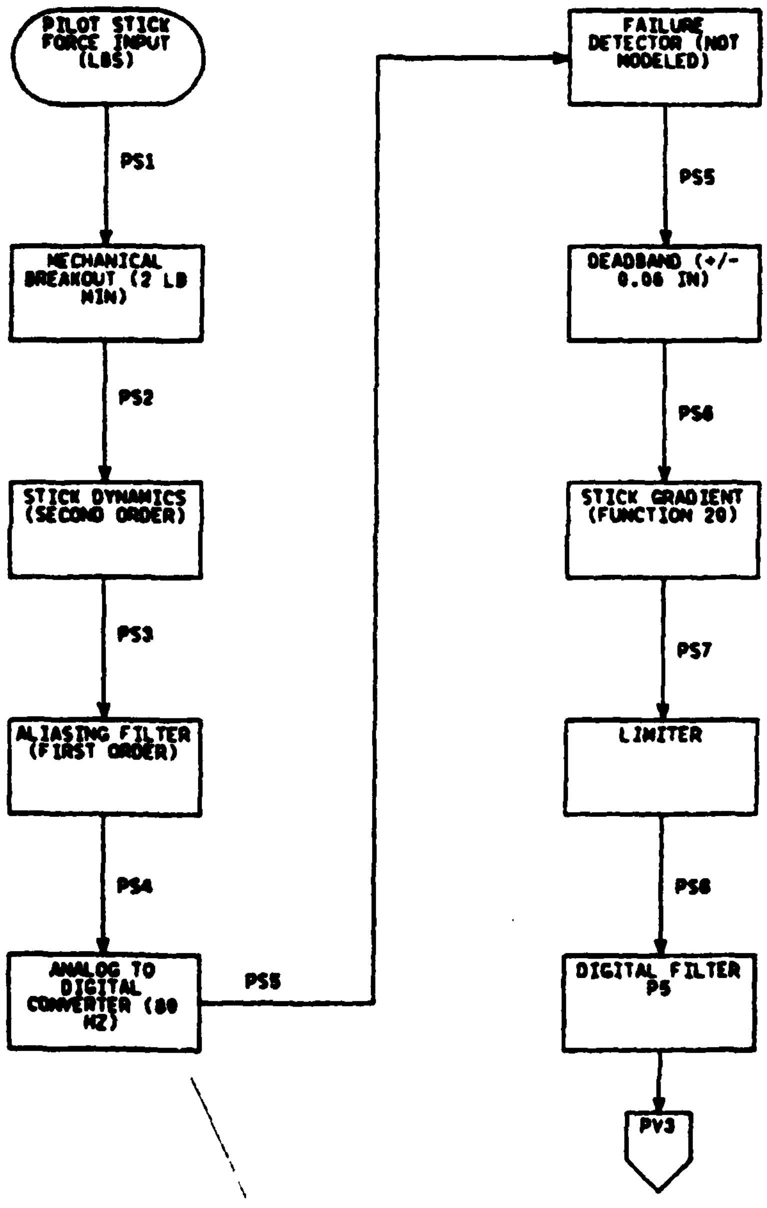
The computer program developed in this thesis simulates the performance of the longitudinal flight control system of the F/A-18 aircraft by generating control surface deflections and cross axis electrical signals. The responses to AOA, normal acceleration and pitch rate feedback correspond to the descriptions given by the aircraft manufacturer.

The use of CSMP simplified the task of system modeling by provision of pre-programmed functional blocks and a flexible format for outputs. The CSMP program size restrictions do not appear to be a factor which would prohibit the addition of the lateral and directional flight control systems to this simulation. It is recommended that future expansion of this program be done using techniques to conserve program size. FORTRAN subroutines and computer library functions should be utilized due to the limited MACRO and NOSORT capabilities of the CSMP translator.

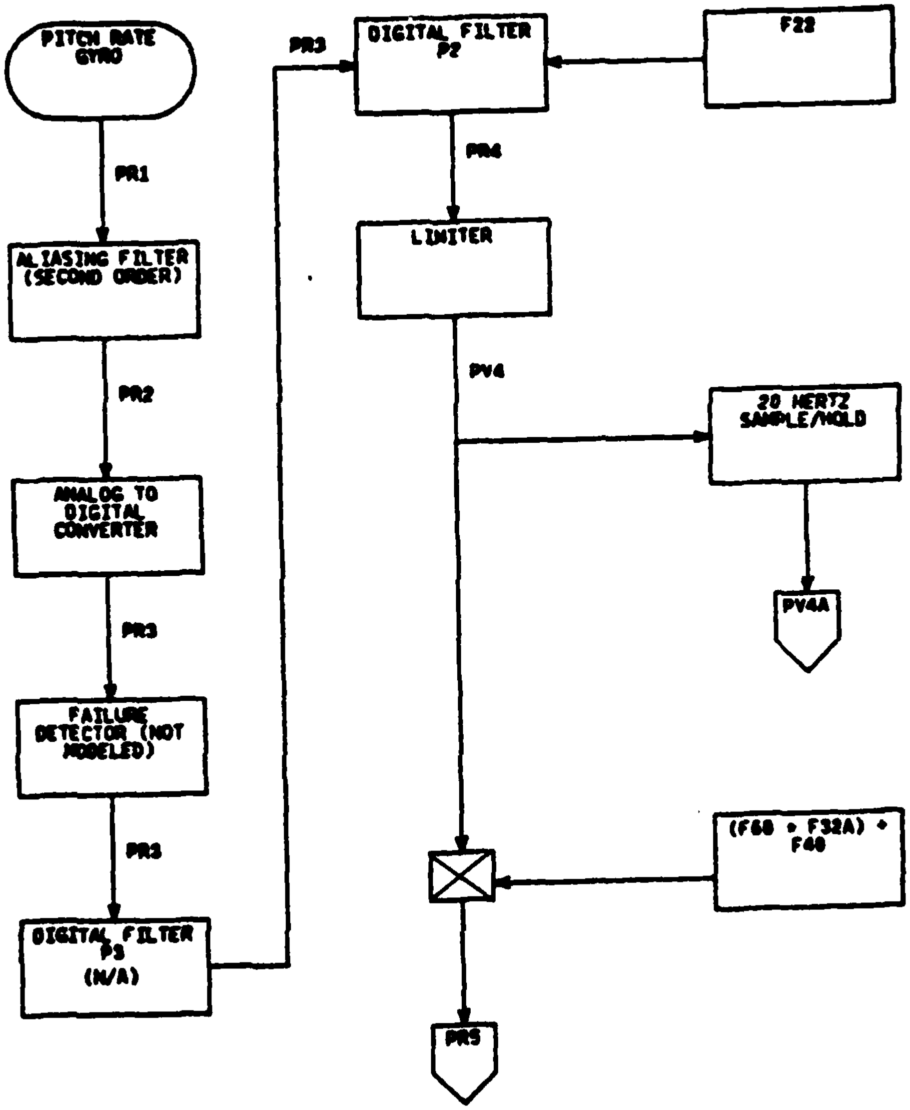
APPENDIX A

FLIGHT CONTROL SYSTEM COMPONENT PATHS

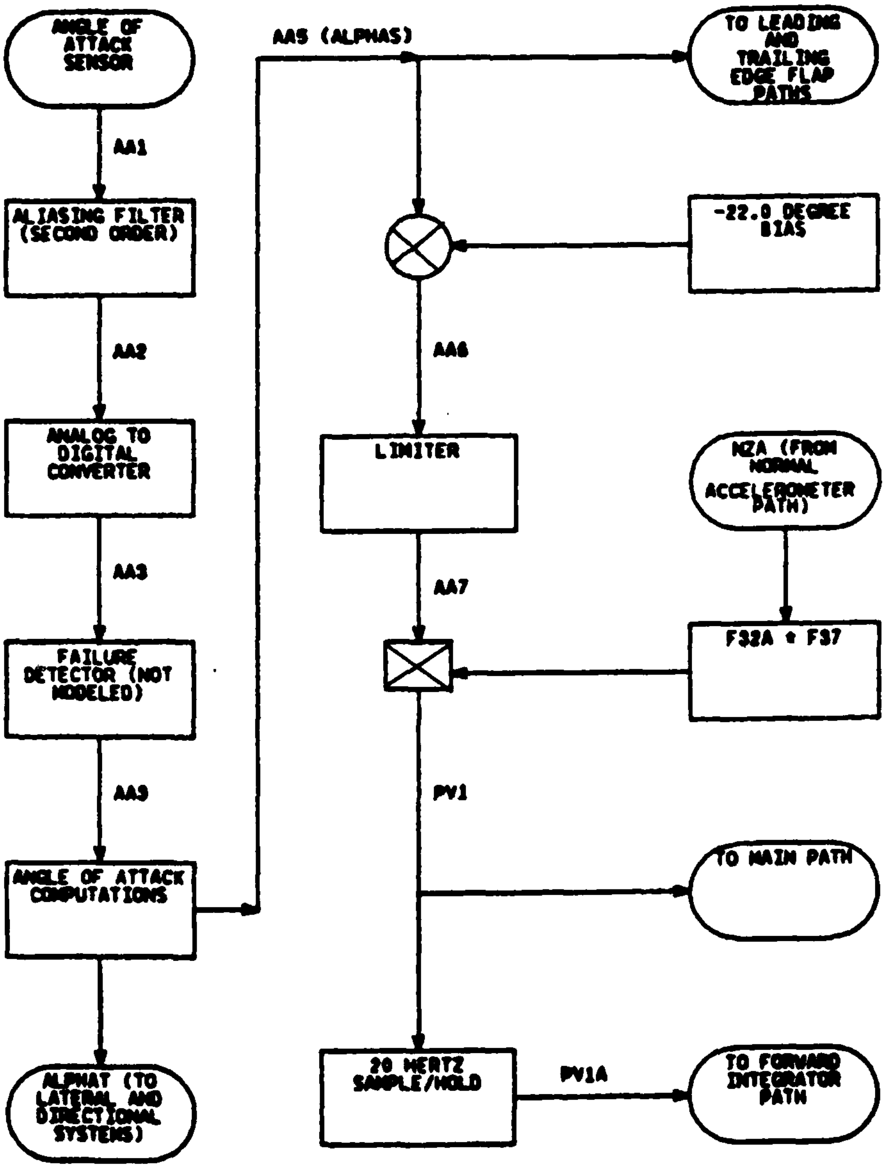
PILOT STICK INPUT PATH



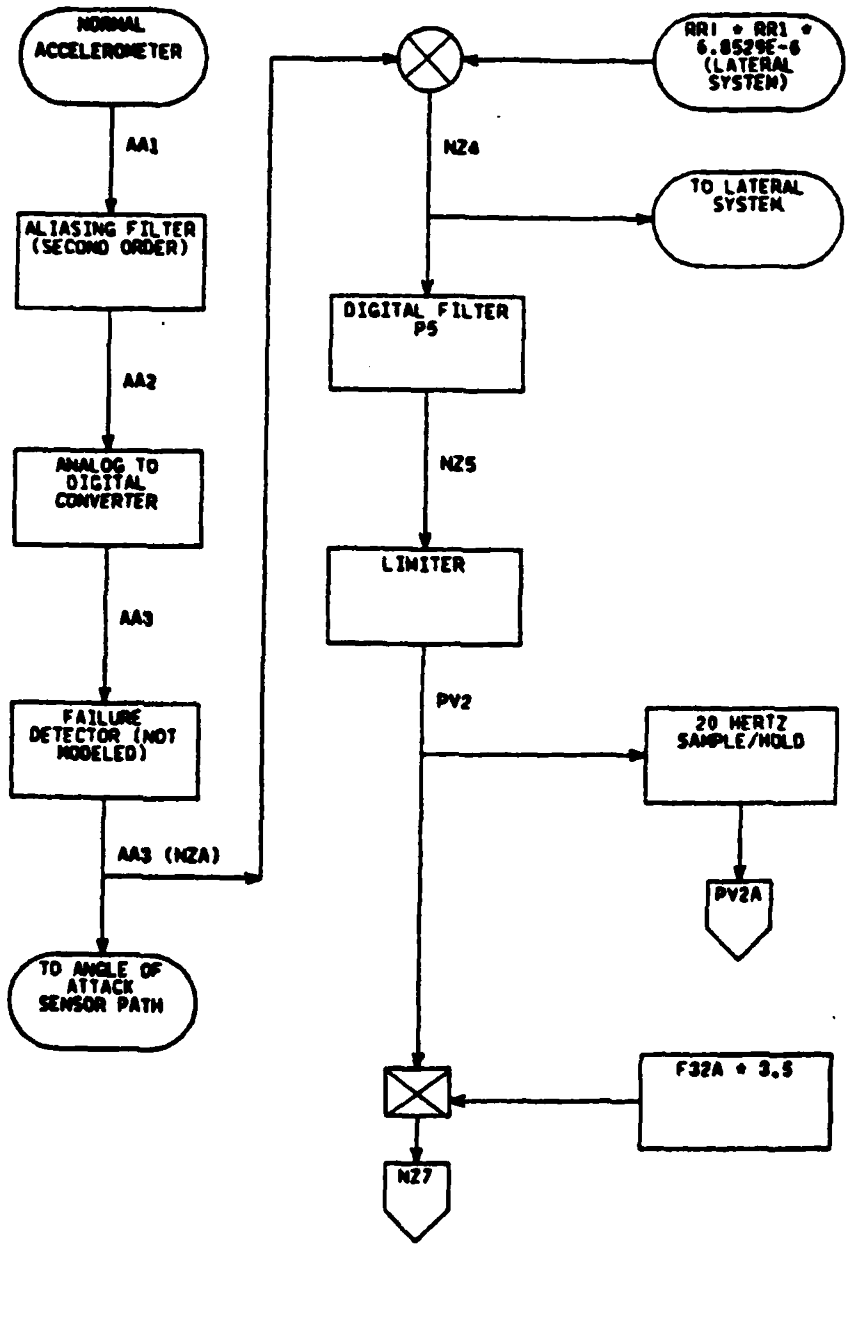
PITCH RATE GYRO PATH



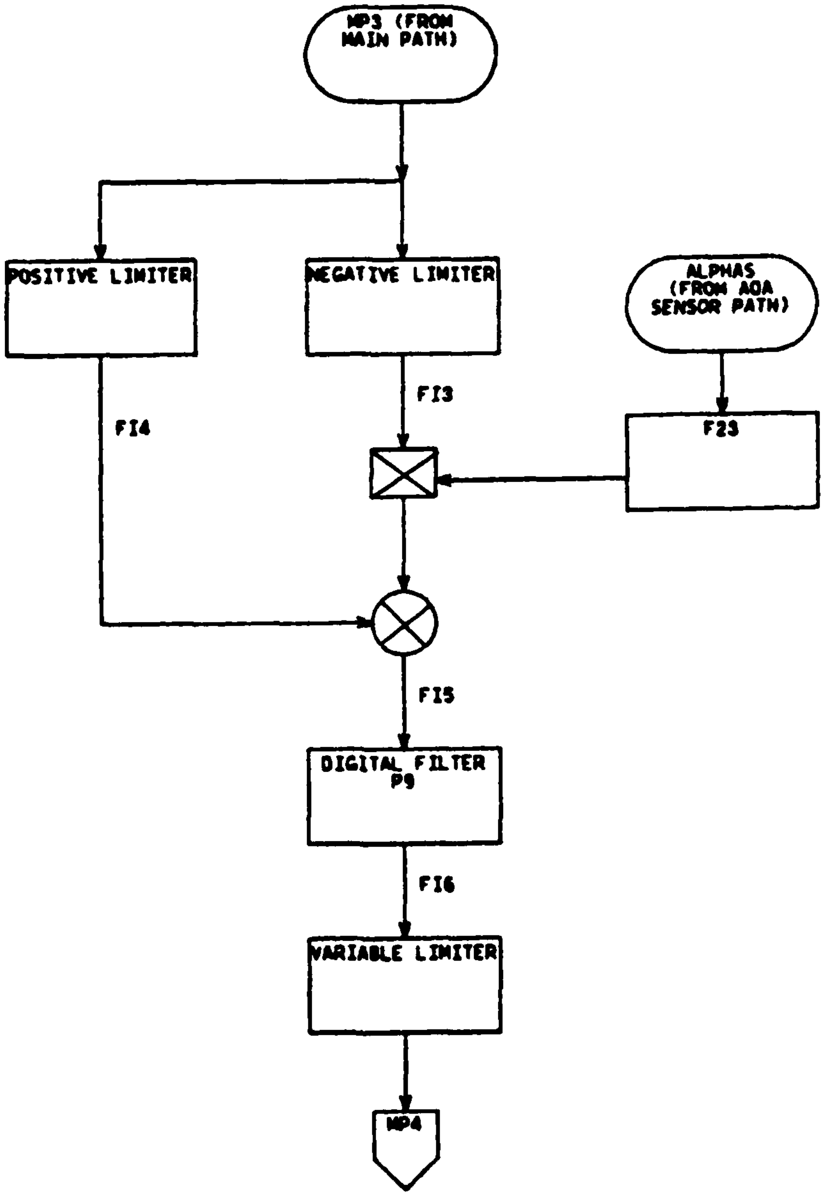
ANGLE OF ATTACK SENSOR PATH



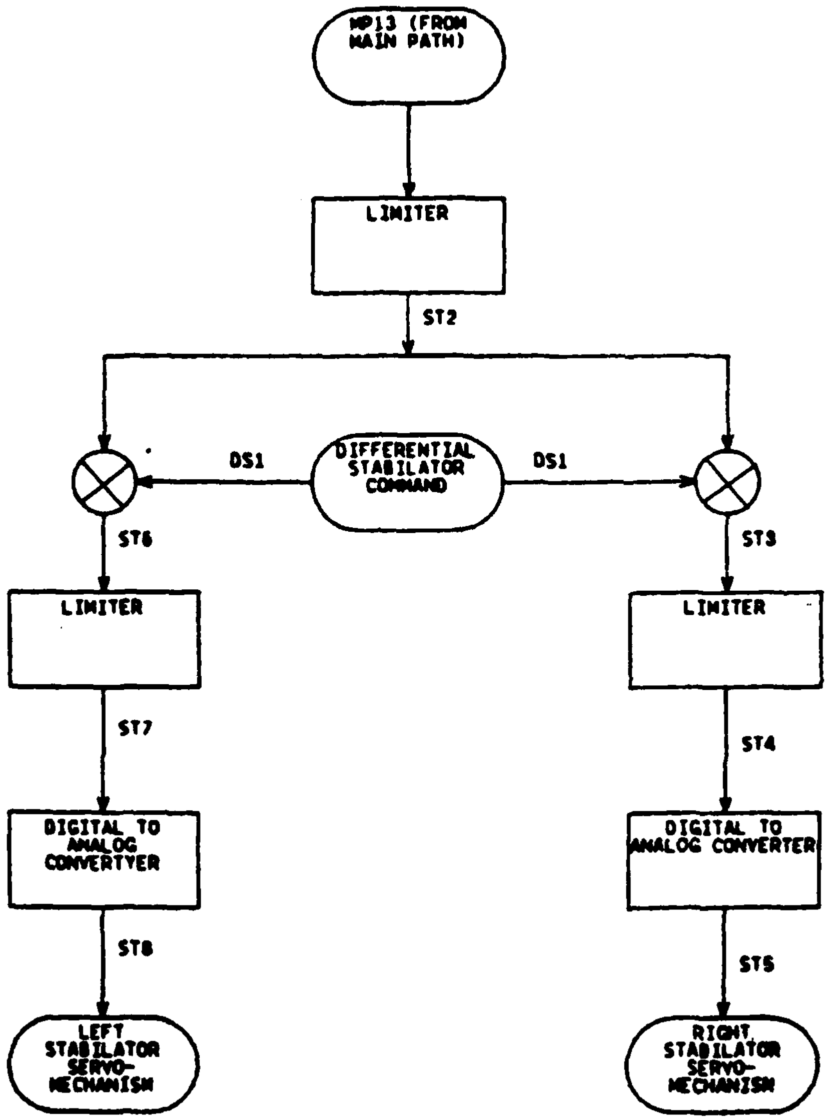
NORMAL ACCELEROMETER PATH



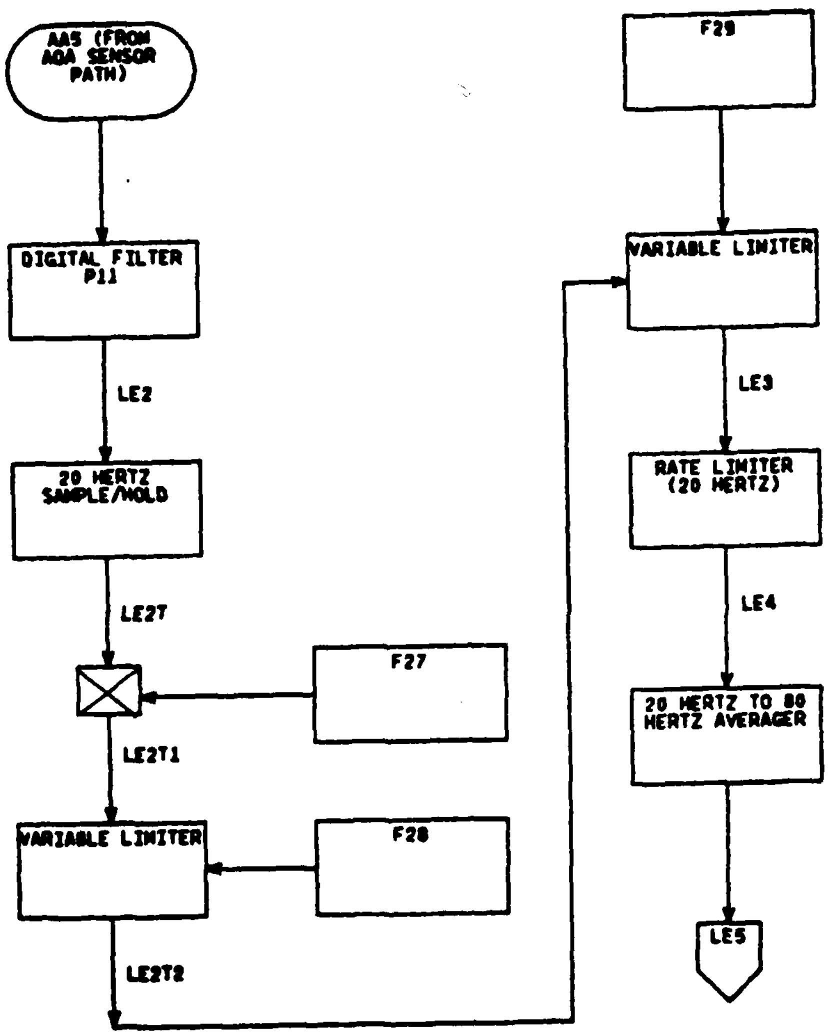
FORWARD INTEGRATOR
PATH



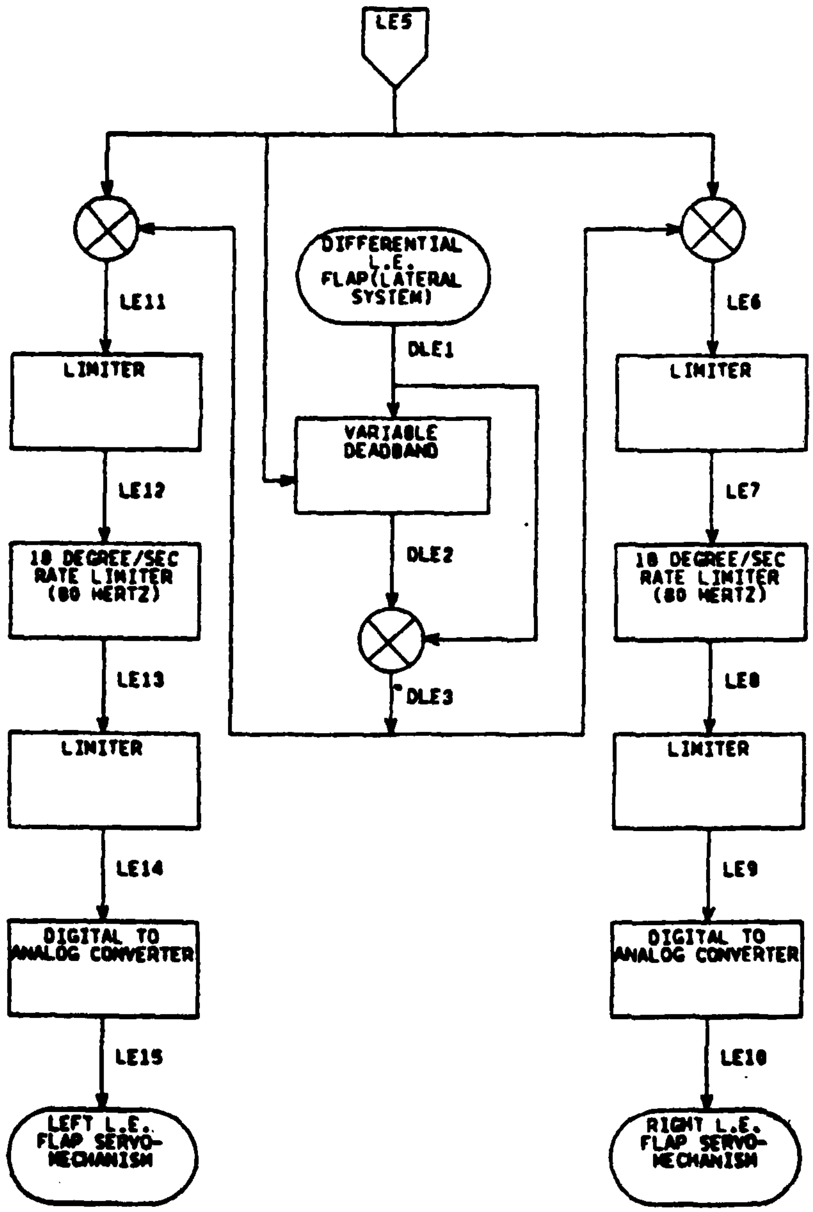
STABILATOR PATH



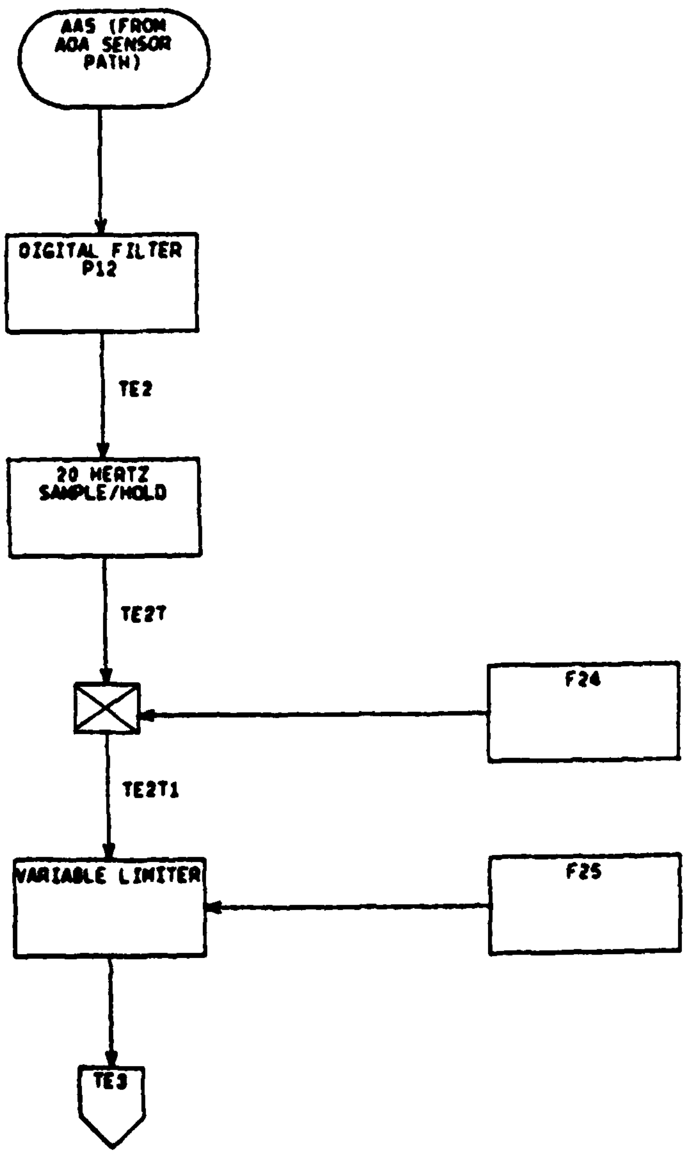
LEADING EDGE
FLAP PATH



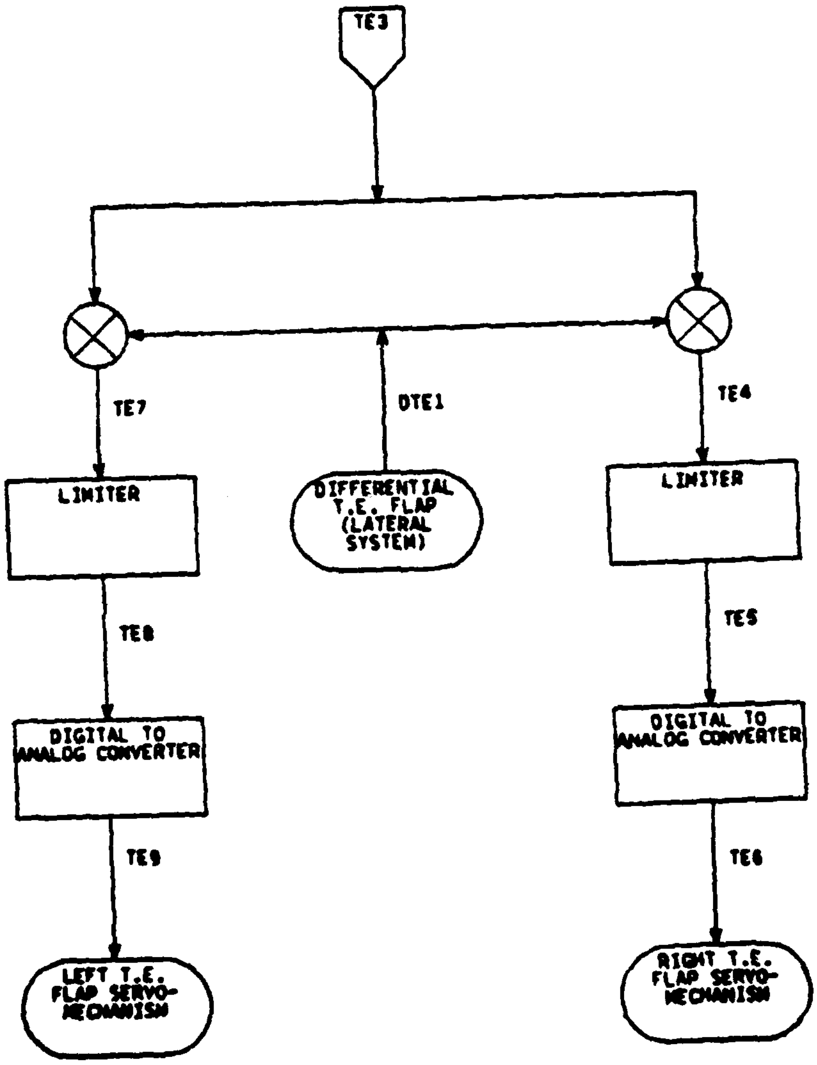
LEADING EDGE FLAP
PATH (CONT'D)



TRAILING EDGE FLAP PATH



TRAILING EDGE FLAP PATH
(CONT'D)



APPENDIX E

FLIGHT CONTROL COMPUTER PROGRAM

```
*   ***********************************************************
*   **                                                       **
*   **     LONGITUDINAL FLIGHT CONTROL LAW SIMULATION        **
*   **                                                       **
*   ***********************************************************
*
*      ****************************************************
*      *****INTEGRATING,LAG,AND NOTCH DIGITAL FILTERS*****
*      ****************************************************
MACRO FOUT=ZINT(FIN,KA,KB,IMP,FOUTZ1)
     PROCEDURAL
     IF(IMP.NE. 1.0) GO TO 10
     IF(KEEP.NE.1.0) GO TO 10
     IF(TIME.EQ.0.0) GO TO 10
     FOUT=KA*FIN + KB*FOUTZ1
     FOUTZ1=FOUT
10   CONTINUE
ENDMAC
MACRO FOUT=ZLAG(FIN,KA,KB,KC,IMP,FINZ1,FOUTZ1)
     PROCEDURAL
     IF(IMP.NE. 1.0) GO TO 10
     IF(KEEP.NE.1.0) GO TO 10
     IF(TIME.EQ.0.0) GO TO 10
     FOUT=KA*FIN - KB*FINZ1 + KC*FOUTZ1
     FOUTZ1=FOUT
     FINZ1=FIN
10   CONTINUE
ENDMAC
MACRO FOUT=ZNOTCH(FIN,KA,KB,KC,KD,KE,IMP,FINZ1,...
     FINZ2,FOUTZ1,FOUTZ2)
     PROCEDURAL
     IF(IMP.NE. 1.0) GO TO 10
     IF(KEEP.NE.1.0) GO TO 10
     IF(TIME.EQ.0.0) GO TO 10
     FOUT=KA*FIN+KB*FINZ1+KC*FINZ2-KD*FOUTZ1-KE*FOUTZ2
     FOUTZ2 = FOUTZ1
     FOUTZ1 = FOUT
     FINZ2 = FINZ1
     FINZ1 = FIN
10   CONTINUE
ENDMAC
*
*      ****************************
*      *****FREQUENCY AVERAGERS*****
*      ****************************
MACRO  Z40=AV2040(Z20,IMP)
     PROCEDURAL
       IF (KEEP.NE.1.0) GO TO 20
       IF (TIME.NE.0.0) GO TO 5
       Z20Z1=Z20
       Z40Z1=Z20
       DEL=0.0
       GO TO 10
5      IF (IMP.EQ.1.0) GO TO 10
       Z40=Z40Z1+DEL
       GO TO 15
10     Z40=Z20
       DEL= (Z20-Z20Z1)/2.0
       Z20Z1=Z20
       Z40Z1=Z40
15     CONTINUE
20     CONTINUE
```

```
ENDMAC
MACRO  Z80=AV4080(Z40,IMP)
     PROCEDURAL
        IF  (KEEP.NE.1.0) GO TO 20
        IF  (TIME.NE.0.0) GO TO 5
        Z40Z1=Z40
        Z80Z1=Z40
        DEL=0.0
        GO TO 10
5       IF  (IMP.EQ.1.0)  GO  TO 10
        Z80=Z80Z1+DEL
        GO  TO  15
10      Z80=Z40
        DEL=(Z40-Z40Z1)/2.0
        Z40Z1=Z40
        Z80Z1=Z80
15      CONTINUE
20      CONTINUE
ENDMAC
MACRO Z80=AV2080(Z20,IMP)
     PROCEDURAL
        IF  (KEEP.NE.1.0) GO TO 20
        IF  (TIME.NE.0.0) GO TO 5
        Z20Z1=Z20
        Z80Z1=Z20
        GO  TO  10
5       IF  (IMP.EQ.1.0)  GO  TC 10
        Z80=Z80Z1+DEL2
        GO  TC  15
10      Z80=Z20
        DEL2=(Z2C-Z20Z1)/4.0
15      CONTINUE
        Z20Z1=Z20
        Z80Z1=Z80
20      CONTINUE
ENDMAC
*
*       ***********************************************
*       *****INITIAL CONDITIONS AND CONSTANTS FOR BASE*****
*       *****FLIGHT CONDITICN OF 20,000 FT AND 250 KTS*****
*       ***********************************************
INITIAL
CONSTANT PS=972.49,QC=112.73,RATE20=0.05,RATE40=0.025,...
PS3IC1=0.0,PS3IC2=0.0,PS4IC1=0.0,PR2IC1=0.0,PR2IC2=0.0,...
AA2IC2=0.0,NZ2IC1=0.0,NZ2IC2=0.0,NZ4Z1=0.0,NZ5Z1=0.0,...
PK10=0.4647,PI6Z1=0.0,MP11Z1=0.0,MP11Z2=0.0,MP12Z1=0.0,...
P8A=0.69084,P8B=-0.99068,P8C=0.66312,P8D=-0.99068,...
DS1=0.0,DLE1=0.0,DTE1=0.0,LE2Z1=0.0,TE2Z1=0.0,LEDIC1=0.0,...
LEDIC2=0.0,TEDIC1=0.0,TEDIC2=0.0,PR3Z1=0.0,PR4Z1=0.0,...
RATE80=0.0125,AA2IC1=0.0,PK9=-1.1543,MP12Z2=0.0,P8E=0.3539
*
DYNAMIC
*
   SORT
*
*       ****************
*       *****INPUTS*****
*       ****************
        PS1=6.0*STEP(0.0)
        PR1=0.0
        AA1=22.0+RAMP(4.0)
        NZ1=0.0
        RR1=0.0
        YR1=0.0
*
*
*       ************************************
*       *****FUNCTIONS REQUIRING NOSORT*****
*       ************************************
```

```
*
*     *********************
*     *****FUNCTION 12*****
*     *********************
      RI=QC/PS
      F12T1=RI**2*9.625-.025*RI+1.0
      F12T2=PS*7.969E-4+0.84
      F12MAX=LIMIT(1.0,8.0,F12T2)
      F12T3=LIMIT(1.0,F12MAX,F12T1)
      F12T5=LIMIT(.5,1.35,RI)
      F12T6=F12T5*(0.00952*PS+4.04) +(-PS*0.00396-1.18)
      F12T4=LIMIT(1.0,8.0,F12T6)
   NCSORT
      IF(.NOT.RI.GT.0.5) GO TO 5
      F12 = F12T4
      GO TO 6
5     CONTINUE
      F12 = F12T3
6     CONTINUE
   SORT
*
*     *********************
*     *****FUNCTION 40*****
*     *********************
      F40T1=3.25-3.0*LIMIT(0.75,0.85,RI)
      F40T2=0.65625-.0013125*LIMIT(500.0,1800.0,PS)
      F40T3=-0.26177+(9.635E-4)*LIMIT(500.0,1800.0,PS)
      PIQ=LIMIT(0.0,1800.0,PS)
      F40T4=LIMIT(0.0,105.0,QC-335.0)
      F40T5=(F40T4*(-PIQ*(1.17428E-6)+(1.5238E-3)))...
            +0.475-(6.5E-4)*PIQ
      F40T6=F40T3+(F40T2*LIMIT(0.75,0.85,RI))
      F40T7=F40T4*(5.6746E-4+1.9841E-7*PIQ)...
            +(0.16923-3.84615E-5*PIQ)
      F40T8=(0.16923-3.86415E-5*PIQ)...
            +(F40T4*(1.67247E-3-9.29152E-7*PIQ))
*
   NCSORT
      IF (.NOT.RI.GE.0.75) GO TO 20
         F40T9=F40T8*F40T1
         GO TO 30
20    CONTINUE
         F40T9=F40T8
30    CONTINUE
      IF (.NOT.(QC.GE.440.0.AND.RI.GE.0.75)) GO TO 40
         F40T10=F40T6
         GO TO 50
40    CONTINUE
         F40T10=F40T7
50    CONTINUE
      IF (.NOT.PIQ.GT.980.0) GO TO 60
         F40=F40T9
         GO TO 80
60    CONTINUE
      IF (.NOT.PIQ.LE.500.0) GO TO 70
         F40=F40T5
         GO TO 80
70    CONTINUE
         F40=F40T10
80    CONTINUE
*
*     *************************
*     *****SORTED FUNCTIONS*****
*     *************************
   SORT
*     *************************
*     *****FUNCTIONS 22,23*****
*     *************************
      F22=0.0167*(-800.0+LIMIT(800.0,900.0,QC))
```

```
*
      F23=3.1435-0.1429*LIMIT(15.0,21.998,ALPHAT)
*
*     ****************************************
*     *****FUNCTIONS 25,27,28,29U,29L,32A,37*****
*     ****************************************
      F25=47.636-0.05106*LIMIT(600.0,835.0,QC)
*
      F27=1.328*(ALPHAT+7.6584-17.86*LIMIT(0.44,0.63,RI))
*
      F28=44.551-0.04058*LIMIT(260.0,950.0,QC)
*
*
      F29U=87.3825-76.25*LIMIT(0.7,1.146,RI)
*
      F29L=0.0
*
      QKF=LIMIT(200.0,2000.0,QC)
      F32A=100.0/QKF
*
      F37=2.5-0.5*LIMIT(3.0,5.0,NZA)
*
*     ************************
*     *****FUNCTIONS 68,107*****
*     ************************
      F68=-0.002977*(-480.0+LIMIT(260.0,480.0,QC))
*
      F107T1=((-5.714E-7)*ABS(QC-750.0))+(8.4E-4)
      F107=LIMIT(0.0,10000.0,F107T1)
*
*     *************************
*     *****IMPULSE FUNCTIONS*****
*     *************************
      IMP20=IMPULS(0.0,RATE20)
      IMP40=IMPULS(0.0,RATE40)
      IMP80=IMPULS(0.0,RATE80)
*
*     *********************
*     *****PK FUNCTIONS*****
*     *********************
      PK11=1.65*F22
      PK12=0.5654
      PK16A=3.5*F32A
      PK17A=F37*F32A
      PK19A=F12*F32A*3.5
      PK21A=F12*F32A*F68
      PK22A=F12*F32A
      PK16=ZHOLD(IMP20,PK16A)
      PK17=ZHOLD(IMP20,PK17A)
      PK19=ZHOLD(IMP20,PK19A)
      PK21=ZHOLD(IMP20,PK21A)
      PK22=ZHOLD(IMP20,PK22A)
*
*     ******************************
*     *****PILOT STICK INPUT PATH*****
*     ******************************
      PS2=DEADSP(-2.0,2.0,PS1)
      PS3=CMPXPL(PS3IC1,PS3IC2,0.14,27.3,PS2*106.47)
      PS4=REALPL(PS4IC1,7.9365E-3,PS3)
      PS5=ZHOLD(IMP80,PS4)
      PS6=PS5
      PS7=PS6*(7.0+(0.2*ABS(PS6)))
      PS8=LIMIT(-25.0,50.0,PS7)
      PV3=PS8
      PV3A=ZHOLD(IMP20,PV3)
      PS9=PV3*F32A
*
*     ***************************
*     *****PITCH RATE GYRO PATH*****
```

```
*       *******************************
        PR2=CMPXFL(PR2IC1,PR2IC2,0.89,78.5,PR1*6162.25)
        PR3=ZHOLD(IMP80,PR2)
        P2A=1.0+PK11*(1.0-PK12)
        P2B=(1.0+PK11)*(1.0-PK12)
        P2C=1.0-PK12
        PR4=ZLAG(PR3,P2A,P2B,P2C,IMP80,PR3Z1,PR4Z1)
        PV4=LIMIT(-80.0,120.0,PR4)
        PR5=PV4*((F68*F32A)+F40)
        PV4A=ZHOLD(IMP20,PV4)
*
*       **************************************
*       *****ANGLE OF ATTACK SENSOR PATH*****
*       **************************************
        AA2=CMPXFL(AA2IC1,AA2IC2,0.74,209.0,AA1*43681.0)
        AA3=ZHOLD(IMP40,AA2)
        ALPHAT=0.59*AA3+1.9
        ALPHAS=ALPHAT
        AA6=ALPHAT-22.0
        AA7=LIMIT(0.0,10000.0,AA6)
        PV1=PK17*AA7
        PV1A=ZHOLD(IMP20,PV1)
*
*       ************************************
*       *****NORMAL ACCELEROMETER PATH*****
*       ************************************
        NZ2=CMPXFL(NZ2IC1,NZ2IC2,0.89,200.0,NZ1*40000.0)
        NZ3=ZHOLD(IMP40,NZ2)
        NZA=NZ3
        NZ4=NZ3-(RR1**2)*(6.8529E-6)
        P5A=(1.0+PK9*(1.0-PK10))
        P5B=(1.0+PK9)*(1.0-PK10)
        P5C=1.0-PK10
        NZ5=ZLAG(NZ4,P5A,P5B,P5C,IMP40,NZ4Z1,NZ5Z1)
        PV2=LIMIT(-10.0,10.0,NZ5)
        NZ7=PV2*PK16
        PV2A=ZHOLD(IMP20,PV2)
*
*       **********************************
*       *****FORWARD INTEGRATOR PATH*****
*       **********************************
        FI2=MP3
        FI3=LIMIT(-10000.0,0.0,FI2)
        FI4=LIMIT(0.0,10000.0,FI2)
        FI5=(FI3*F23)+FI4
        FI6=ZINT(FI5,0.05,1.C,IMP20,FI6Z1)
*
*       ********************
*       *****MAIN PATH*****
*       ********************
        MP1=PV3A*PK22
        MP2=(PV4A*PK21)-MP1
        MP3=(PK19*PV2A)+PV1A+MP2
        MP4=LIMIT(-50.0,25.C,FI6)
        MP5A=AV2040(MP4,IMP20)
        MP5=ZHOLD(IMP40,MP5A)
        MP6=NZ7+PF5
        MP7=MP6+PV1
        MP8A=MP7+((RR1*YR1)*F107)
        MP8=ZHOLD(IMP40,MP8A)
        MP9=AV4080(MP8,IMP40)
        MP10=MP9+PR5
        MP11=LIMIT(-25.0,25.C,MP10)
        MP12=ZNCTCH(MP11,P8A,P8B,P8C,P8D,P8E,IMP80,...
             MP11Z1,MP11Z2,MP12Z1,MP12Z2)
        MP13=MP12-PS9
*
*       **************************
*       *****STABILATOR PATH*****
```

```
*       **************************
        ST2=LIMIT(-24.0,10.5,MP13)
        ST3=ST2-DS1
        ST4=LIMIT(-24.0,10.5,ST3)
        ST5=CNTZR(0.0125,ST4)
        STADEF=CMPXPL(TEDIC1,TEDIC2,0.7,40.0,ST5*1600.0)
*
*       ********************************
*       *****LEADING EDGE FLAP PATH*****
*       ********************************
        LE2=ZINT(ALPHAS,0.0625,0.9375,IMP80,LE2Z1)
        LE2T=ZHOLD(IMP20,LE2)
        LE2T1A=LE2T*F27
        LE2T1=ZHOLD(IMP20,LE2T1A)
        LE2T2=LIMIT(0.0,F28,LE2T1)
        LE3=LIMIT(0.0,F29U,LE2T2)
*                ****************************************
*                *****18 DEG/SEC RATE LIMIT, 20 HZ*****
*                ****************************************
   NOSORT
        IF(TIME.NE.0.0) GO TO 85
        LE4Z1=LE3
85      CONTINUE
        DEL1=LE3-LE4Z1
        DELLIM=LIMIT(-0.9,0.9,DEL1)
        LE4A=LE4Z1+DELLIM
        IF (IMP20.NE.1.0) GO TO 90
        LE4Z1=LE4A
90      CONTINUE
        LE4=ZHOLD(IMP20,LE4A)
*                ****************************************
   SORT
        LE5=AV2080(LE4,IMP20)
        DLE2P1=LE5+3.0
        DLE2=DEADSP(-DLE2P1,DLE2P1,DLE1)
        DLE3=DLE1+DLE2
*       NOTE:POSITIVE LEADING EDGE FLAP PATH ONLY
        LE6=LE5+DLE3
        LE7=LIMIT(-3.0,33.0,LE6)
*                ****************************************
*                *****18 DEG/SEC RATE LIMIT, 80 HZ*****
*                ****************************************
   NOSORT
        IF(TIME.NE.0.0) GO TO 110
        LE8Z1=LE7
110     CONTINUE
        DEL1=LE7-LE8Z1
        DELLIM=LIMIT(-0.225,0.225,DEL1)
        LE8A=LE8Z1+DELLIM
        IF (IMP80.NE.1.0) GO TO 120
        LE8Z1=LE8A
120     CONTINUE
        LE8=ZHOLD(IMP80,LE8A)
*                ****************************************
   SORT
        LE9=LIMIT(-3.0,33.0,LE8)
        LE10=CNTZR(0.0125,LE9)
        LEFLAP=CMPXPL(LEDIC1,LEDIC2,1.4,20.0,LE10*400.0)
*
*       ******************************
*       *****NOSORT FUNCTION 24 *****
*       ******************************
        F24L1=22.538-20.51*LIMIT(0.27,0.66,RI)
        F24L2=32.76-36.0*LIMIT(0.66,0.91,RI)
        F24T1=LIMIT(0.0,10000.0,ALPHAT)
        F24T2=ALPHAT-(14.8769-7.6923*LIMIT(0.27,0.91,RI))
        F24T3=-2.0*LIMIT(0.0,10000.0,F24T2)
        F24T4=1.4*(F24T1+F24T3)
    NOSORT
```

```
       IF (.NOT.RI.GT.0.66) GO TO 10
           F24L=F24L2
       GO TO 15
10     CONTINUE
           F24L=F24L1
15     CONTINUE
       F24=LIMIT(0.0,F24L,F24T4)
*
*      *********************************
*      *****TRAILING EDGE FLAP PATH*****
*      *********************************
       TE2=ZINT(ALPHAS,0.03125,0.096875,IMP80,TE2Z1)
       TE2T=ZHCLD(IMP20,TE2)
       TE2T1=TE2T*F24
       TE3=LIMIT(0.0,F25,TE2T1)
*      NOTE:POSITIVE TRAILING EDGE FLAP PATH ONLY
       TE4=TE3+DTE1
       TE5=LIMIT(-8.0,45.0,TE4)
       TE6=QNTZR(0.0125,TE5)
       TEFLAP=CMPXPL(TEDIC1,TEDIC2,1.4,20.0,TE6*40.0)
TERMINAL
METHOD RKSFX
TIMER FINTIM=10.0,OUTDEL=0.0125,PRDEL=0.1250,DELT=0.0125
      PRINT STADEF,LEFLAP,TEFLAP
LABEL STADEF IN DEGREES
LABEL TIME IN SECONDS
PAGE XYPLOT
      OUTPUT TIME,STADEF
     END
     STOP
ENDJOB
```

APPENDIX C

AIR DATA SCHEDULES

```
LABEL FUNCTION 12
*
INITIAL
       PARAMETER PS= (2000.0,1000.0,500.0,200.0)
*
DYNAMIC
   SORT
      RI=RAMP(0.0)
      F12T1=RI**2*9.625-.025*RI+1.0
      F12T2=PS*7.969E-4+0.84
      F12MAX=LIMIT(1.0,8.0,F12T2)
      F12T3=LIMIT(1.0,F12MAX,F12T1)
      F12T5=LIMIT(.5,1.35,RI)
      F12T6=F12T5*(0.00952*PS+4.04) +(-PS*0.00396-1.18)
      F12T4=LIMIT(1.0,8.0,F12T6)
   NCSORT
      IF(.NOT.RI.GT.0.5) GO TO 5
      F12 = F12T4
      GO TO 6
5     CONTINUE
      F12 = F12T3
6     CONTINUE
*
TERMINAL
      TIMER FINTIM =2.0, OUTDEL=0.05, PRDEL=0.05
      PRINT F12T1,F12T2,F12T3,F12T4,F12T5,F12T6,F12MAX,F12
      PAGE XYPLOT
      OUTPUT RI, F12
    END
    STOP
ENDJOB


LABEL FUNCTION 24
*
INITIAL
      PARAMETER RI=(0.27,0.66,0.91)
*
DYNAMIC
*
   SORT
      ALPHA=RAMP(0.0)
      F24L1=22.538-20.51*LIMIT(0.27,0.66,RI)
      F24L2=32.76-36.0*LIMIT(0.66,0.91,RI)
      F24T1=LIMIT(0.0,10000.0,ALPHA)
      F24T2=ALPHA-(14.8769-7.6923*LIMIT(0.27,0.91,RI))
      F24T3=-2.0*LIMIT(0.0,10000.0,F24T2)
      F24T4=1.4*(F24T1+F24T3)
*
   NOSORT
      IF (.NOT.RI.GT.0.66) GO TO 10
         F24L=F24L2
      GO TO 15
10    CONTINUE
         F24L=F24L1
15    CONTINUE
      F24=LIMIT(0.0,F24L,F24T4)
*
TERMINAL
      TIMER FINTIM=30.0,OUTDEL=0.5,PRDEL=0.5
      PRINT F24L1,F24L2,F24T1,F24T2,F24T3,F24T4,F24L,F24
```

```
        PAGE XYPLOT
        OUTPUT ALPHA,F24
      END
      STOP
ENDJOB


LABEL FUNCTION 29
*
INITIAL
        PARAMETER PS=(1250.0)
*
DYNAMIC
        RI=RAMP(0.0)
*
        F29U=87.3825-76.25*LIMIT(0.7,1.146,RI)
        F29L=0.0
*
TERMINAL
        TIMER FINTIM=1.5, PRDEL=0.05, OUTDEL=0.05
        PRINT RI,F29U,F29L
        PAGE XYPLOT
        OUTPUT RI,F29U,F29L
      END
      STOP
ENDJOB


LABEL FUNCTION 40
*
INITIAL
        PARAMETER QC=(1700.0,440.0,335.0)
*
DYNAMIC
    SORT
        PS= 0.0001 + RAMP(0.0)
        RI=QC/PS
        F40T1=3.25-3.0*LIMIT(0.75,0.85,RI)
        F40T2=0.65625-.0013125*LIMIT(500.0,1800.0,PS)
        F40T3=-0.26177+(9.635E-4)*LIMIT(500.0,1800.0,PS)
        PIQ=LIMIT(0.0,1800.0,PS)
        F40T4=LIMIT(0.0,105.0,QC-335.0)
        F40T5=(F40T4*(-PIQ*(1.17428E-6)+(1.5238E-3)))...
            +0.475-(6.5E-4)*PIQ
        F40T6=F40T3+(F40T2*LIMIT(0.75,0.85,RI))
        F40T7=F40T4*(5.6746E-4+1.9841E-7*PIQ)...
            +(0.16923-3.84615E-5*PIQ)
        F40T8=(0.16923-3.86415E-5*PIQ)...
            +(F40T4*(1.67247E-3-9.29152E-7*PIQ))
*
    NOSORT
        IF (.NOT.RI.GE.0.75) GO TO 20
            F40T9=F40T8*F40T1
            GO TO 30
20      CONTINUE
            F40T9=F40T8
30      CONTINUE
        IF (.NOT.(QC.GE.440.0.AND.RI.GE.0.75)) GO TO 40
            F40T10=F40T6
            GO TO 50
40      CONTINUE
            F40T10=F40T7
50      CONTINUE
        IF (.NOT.PIQ.GT.980.0) GO TO 60
            F40=F40T9
```

```
        GO TO 80
60    CONTINUE
      IF (.NOT.PIQ.LE.500.C) GO TO 70
        F40=F40T5
        GO TO 80
70    CONTINUE
        F40=F40T10
80    CONTINUE
*
TERMINAL
      TIMER FINTIM=2000.0,PRDEL=20.0,OUTDEL=20.0
      PRINT PS,QC,RI,F40T7,F40T9,F40T10,F40
      PAGE XYPLOT
      OUTPUT PS,F40
    END
    STOP
ENDJCB


LABEL GENERATED FUNCTION
*    THIS PROGRAM GENERATES QC FUNCTIONS 22,25,28,32A, AND 68
*
INITIAL
      PARAMETER PS=1000.0
*
DYNAMIC
      QC=RAMP(0.0)
*
      F22=0.0167*(-800.0+LIMIT(800.0,900.0,QC))
*
      F25=47.636-0.05106*LIMIT(600.0,835.0,QC)
*
      F28=44.551-0.04058*LIMIT(260.0,950.0,QC)
*
      QKF=LIMIT(200.0,2000.0,QC)
      F32A=100.0 / QKF
*
      F68=-0.002977*(-480.0+LIMIT(260.0,480.0,QC))
*
TERMINAL
      TIMER FINTIM=2500.0, OUTDEL=50.0, PRDEL=50.0
      PRINT QC,F22,F25,F28,QKF,F32A,F68
      PAGE XYPLOT
      OUTPUT QC,F22
      OUTPUT QC,F25
      OUTPUT QC,F28
      OUTPUT QC,F32A
      OUTPUT QC,F68
    END
    STOP
ENDJOB


LABEL GENERATED FUNCTION
*
INITIAL
      PARAMETER RI=(0.44,0.63)
*
DYNAMIC
      ALPHA= RAMP (0.0)
      NZA=ALPHA
      ALPHAT=ALPHA
*
      F23=3.1435-0.1429*LIMIT(15.0,21.998,ALPHAT)
*
```

```
      F27=1.328*(ALPHA+7.8584-17.86*LIMIT(0.44,0.63,RI))
*
      F37=2.5-0.5*LIMIT(3.0,5.0,NZA)
*
TERMINAL
      TIMER FINTIM=30.0, PRDEL=1.0, OUTDEL=1.0
      PRINT F23,F27,F37
      PAGE XYPLOT
      OUTPUT ALPHAT,F23
      OUTPUT ALPHA,F27
      OUTPUT NZA,F37
   END
   STOP
ENDJOB
```

APPENDIX D

COMPUTER PROGRAMS FOR SIGNAL BLOCK TESTING

DEADSPACE FUNCTION AND ALIASING FILTERS

```
*
INITIAL
    CONSTANT PS2C1=2.0
*
DYNAMIC
    PS1= -5.0 + RAMP(0.0)
    PS2=DEADSP(-PS2C1,PS2C1,PS1)
    PS3=CMPXPL(0.0,0.0,0.14,27.3,PS2*106.47)
    PS4=REALPL(0.0,7.9365E-3,PS3)
*
TERMINAL
    TIMER FINTIM=10.0, OUTDEL=0.25, PRDEL=0.25
    LABEL STICK DEADSPACE PLOT
    PRINT PS1,PS2
    OUTPUT TIME, PS1,PS2,PS3,PS4
    PAGE XYPLOT
    END
    STOP
ENDJOB
```

FREQUENCY AVERAGERS

```
MACRO Z80=AV2080(Z20,IMP)
    PROCEDURAL
      IF (TIME.NE.0.0) GO TO 52
      Z20Z1=Z20
      Z80Z1=Z20
      GO TO 54
52    IF (IMP.EQ.1.0) GO TO 54
      Z80=Z80Z1+DEL2
      GO TO 56
54    Z80=Z20
      DEL2=(Z20-Z20Z1)/4.0
56    CONTINUE
      Z20Z1=Z20
      Z80Z1=Z80
ENDMAC
INITIAL
      CONSTANT  RATE=0.05
DYNAMIC
      Y1= RAMP(0.0)
      Y1A=EXP(Y1)
      Y2= IMPULS(0.0,RATE)
      Y3= ZHOLD(Y2,Y1A)
      Y4= AV2080(Y3,Y2)
TERMINAL
      METHOD RKSFX
      TIMER FINTIM=1.0,OUTDEL=0.0125,PRDEL=0.0125,DELT=0.0125
      LABEL 20 TO 80 AVERAGER
      PRINT Y2,Y3,Y4
    END
    STOP
ENDJOB
MACRO  Z40=AV2040(Z20,IMP)
    PROCEDURAL
      IF (TIME.NE.0.0) GO TO 5
      Z20Z1=Z20
      Z40Z1=Z20
      DEL=0.0
      GO TO 10
5     IF (IMP.EQ.1.0) GO TO 10
      Z40=Z40Z1+DEL
      GO TO 15
10    Z40=Z20
      DEL=(Z20-Z20Z1)/2.0
      Z20Z1=Z20
      Z40Z1=Z40
15    CONTINUE
ENDMAC
INITIAL
      CONSTANT  RATE=0.05
DYNAMIC
      Y1= RAMP(0.0)
      Y1A=EXP(Y1)
      Y2= IMPULS(0.0,RATE)
      Y3= ZHOLD(Y2,Y1A)
      Y4=AV2040(Y3,Y2)
TERMINAL
      METHOD RKSFX
      TIMER FINTIM=1.0,OUTDEL=0.025,PRDEL=0.025,DELT=0.0125
      LABEL 20 TO 40 AVERAGER
      PRINT Y1,Y1A,Y2,Y3,Y4
      PAGE XYPLOT
      OUTPUT TIME,Y1A,Y3,Y4
    END
    STOP
ENDJOB
```

RATE LIMITERS

```
INITIAL
DYNAMIC
        Y1=RAMP(0.0)
        Y2=EXP(Y1)
        Y3=LIMIT(0.0,3.0,Y2)
        Y4=(-1.0+EXP(-TIME+2.0))*STEP(2.0)
        Y5=Y3+Y4
        IMP20=IMPULS(0.0,0.05)
        IMP40=IMPULS(0.0,0.025)
        IMP80=IMPULS(0.0,0.0125)
        Y20=ZHOLD(IMP20,Y5)
        Y40=ZHOLD(IMP40,Y5)
        Y80=ZHOLD(IMP80,Y5)
NOSORT
        IF(TIME.NE.0.0) GO TO 80
        LE8Z1=Y20
80      CONTINUE
        DEL1=Y20-LE8Z1
        DELLIM=LIMIT(-0.1,0.1,DEL1)
        LE8=LE8Z1+DELLIM
        IF (IMP20.NE.1.0) GO TO 90
        LE8Z1=LE8
90      CONTINUE
TERMINAL
        METHOD RKSFX
        TIMER FINTIM=5.0,OUTDEL=0.0125,PRDEL=0.0125,DELT=0.0125
        PRINT Y3,Y4,Y5,DEL1,DELLIM,LE8Z1,LE8,Y20,IMP20
        PAGE XYPLOT
        OUTPUT TIME,Y20,LE8
        OUTPUT TIME,Y20,LE8Z1
      END
      STOP
ENDJOB
```

DIGITAL FILTERS

```
MACRO FOUT=ZINT(FIN,KA,KB,IMP,FOUTZ1)
      PROCEDURAL
      IF(IMP.NE. 1.0) GO TO 10
      IF(KEEP.NE.1.0) GO TO 10
      IF(TIME.EQ.0.0) GO TO 10
      FOUT=KA*FIN + KB*FOUTZ1
      FOUTZ1=FOUT
10    CONTINUE
ENDMAC
MACRO FOUT=ZLAG(FIN,KA,KB,KC,IMP,FINZ1,FOUTZ1)
      PROCEDURAL
      IF(IMP.NE. 1.0) GO TO 10
      IF(KEEP.NE.1.0) GO TO 10
      IF(TIME.EQ.0.0) GO TO 10
      FOUT=KA*FIN - KB*FINZ1 + KC*FOUTZ1
      FOUTZ1=FOUT
      FINZ1=FIN
10    CONTINUE
ENDMAC
MACRO FOUT=ZNOTCH(FIN,KA,KB,KC,KD,KE,IMP,...
           FINZ1,FINZ2,FOUTZ1,FOUTZ2)
      PROCEDURAL
      IF(IMP.NE. 1.0) GO TO 10
      IF(KEEP.NE.1.0) GO TO 10
      IF(TIME.EQ.0.0) GO TO 10
      FOUT=KA*FIN+KB*FINZ1+KC*FINZ2-KD*FOUTZ1-KE*FOUTZ2
      FOUTZ2 = FOUTZ1
      FOUTZ1 = FOUT
      FINZ2 = FINZ1
      FINZ1 = FIN
10    CONTINUE
ENDMAC
INITIAL
PARAM INTN=.0125,INTD=1.0,AA11A1=0.0,AA11A=0.0
PARAM PK9=-1.1543, PK10=0.4647,AA10B1=0.0,AA11B1=0.0
PARAM NA=.69084,NB=-.99068,NC=.66312,ND=-.99068,NE=.35396
PARAM AA10C1=0.0,AA10C2=0.0,AA11C1=0.0,AA11C2=0.0
LA=(1.+PK9*(1.-PK10))
LB=(1.+PK9)*(1.-PK10)
LC=1.-PK10
DYNAMIC
AA9A = STEP(0.0)
IMP=IMPULS(0.0,0.0125)
AA10A = ZHOLD(IMP,AA9A)
AA11A = ZINT(AA10A,INTN,INTD,IMP,AA11A1)
AA11B=ZLAG(AA10A,LA,LB,LC,IMP,AA10B1,AA11B1)
AA11C = ZNOTCH(AA10A,NA,NB,NC,ND,NE,IMP,AA10C1,...
        AA10C2,AA11C1,AA11C2)
TERMINAL
METHOD RKSFX
TIMER FINTIM=1.5,OUTDEL=0.0125,DELT=0.0125
PRINT AA10A,AA11A,AA11B,AA11C
OUTPUT AA10A(0.0,1.5),AA11A(0.0,1.5),...
       AA11B(0.0,1.5),AA11C(0.0,1.5)
END
STOP
ENDJOB
```

LIST OF REFERENCES

1. Groll, D., F/A-18A Flight Control System Design Report, Vol. I, MDC A7813, McDonnell Aircraft Company, December 1982.

2. Speckart, F., and Green, W., A Guide to Using CSMP, 1st Ed., Prentice-Hall, Inc., 1976.

3. Ogata, K., Modern Control Engineering, Prentice-Hall, Inc., 1970.

4. Hess, R., F/A-18 Flight Control Electronic Set Control Laws, MDC A4107, Volumes I and II, McDonnell Aircraft Company, March 1983.

5. Smith, D., and Barnes, B., F-18 Flight Control System Schematic Arrangement Drawings, MDC A4379, McDonnell Aircraft Company, March 1983.

6. Cadzow, J., Discrete-Time Systems, Prentice-Hall, Inc., 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 67 Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Adjunct Professor Marle D. Hewett, Code 67Hj Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	5
5.	LCDR James R. Carter, USN 1389 Valencia Loop Chula Vista, California 92010	2
6.	LT Scott Graves, USN SMC 2419 Naval Postgraduate School Monterey, California 93943	1

END

FILMED

5-84

DTIC