NASA Technical Memorandum 107601

Simulation Model of a Twin-Tail, High Performance Airplane

Carey S. Buttrill and P. Douglas Arbuckle
Langley Research Center
Hampton, Virginia

and

Keith D. Hoffler
ViGYAN, Inc.
Hampton, Virginia

July 1992

NASA
National Aeronautics and Space Administration

Langley Research Center
Hampton, VA 23665

SUMMARY

A mathematical model and associated computer program (*f18bas*) to simulate a twin-tailed high performance fighter airplane (McDonnell Douglas F/A-18) are described. The simulation is intended to support advanced control law research for application to high performance aircraft. The modeled flight envelope is extensive, allowing investigations in the high-angle-of-attack portion of the flight regime. Included in the simulation mathematical model are the nonlinear, six degree-of-freedom rigid-body equations, an engine model, and simple sensor and actuator models. A simplified form of the F/A-18 digital control laws (version 8.3.3) is implemented, including only the Auto Flap Up (AFU) flight mode used for air combat maneuvering. Aerodynamic forces and moments are calculated from a wind-tunnel-derived database using table look-ups with linear interpolation. The aerodynamic database has an angle-of-attack range of -10° to +90° and a sideslip range of -20° to +20°. Elastic deformation effects are incorporated in a quasi-static-elastic manner, with no dynamic simulation of the elastic degrees of freedom.

CONTENTS

1. INTRODUCTION

The mathematical model and associated computer program (*f18bas*) to simulate a twin-tailed high performance fighter airplane (McDonnell Douglas F/A-18) are described. The simulation is intended to support advanced control law research for application to high performance aircraft. The modeled flight envelope is extensive, allowing investigations in the high-angle-of-attack portion of the flight regime. Included in the simulation mathematical model are the nonlinear, six degree-of-freedom rigid-body equations, an engine model, and simple sensor and actuator models. A simplified form of the F/A-18 digital control laws (version 8.3.3) is implemented, including only the Auto Flap Up (AFU) flight mode used for air combat maneuvering. Aerodynamic forces and moments are calculated from a wind-tunnel-derived database using table look-ups with linear interpolation. The aerodynamic database has an angle-of-attack range of -10° to +90° and a sideslip range of -20° to +20°. Elastic deformation effects are incorporated in a quasi-static-elastic manner, with no dynamic simulation of the elastic degrees of freedom. In the engine model, the throttle-commanded steady-state thrust level and the dynamic response characteristics of the engine are based on airflow rate as determined from a table look-up. Afterburner dynamics are switched in at a threshold based on the engine airflow and commanded thrust. The control surfaces are driven by simple transfer functions with constant rate and position limiting.

1.1 Comments on ACSL

The simulation program written in the Advanced Continuous Simulation Language (ACSL). The ACSL simulation system, described in *ACSL* (1987), consists of a special purpose high level language, a translator, and various libraries to satisfy the commands available in the language. ACSL functions according to specifications established under the auspices of Simulation Councils, Inc. in 1967 [Format Subcommittee (1967)]. In a typical application of the ACSL system, a simulation model is first constructed using the special ACSL language. A translator is invoked that converts a model written in ACSL to Fortran 77 code. Fortran subroutines can be called from the simulation model with certain restrictions. The Fortran code is compiled and linked with the ACSL libraries. The resulting executable program will prompt for user input and produce plots and printed output.

1.2 Simulation Development History

In 1986, a team was established at the NASA Langley Research Center to address some of the control-related problems of designing highly integrated aircraft. This team was composed of members of the Aircraft Guidance and Control Branch and the Aeroservoelasticity Branch at LaRC and was dubbed the Functional Integration Technology (FIT) team. The focus of the FIT team's effort was highly maneuverable aircraft with minimal structural weight. It was anticipated that such aircraft would require active structural control. The original focus vehicle was the X-29 research airplane. About 6 months after the FIT team was formed, the focus vehicle was changed to the F/A-18. Focusing on an F/A-18 allowed the FIT team to leverage off NASA's High Angle-of-Attack Research Vehicle (HARV) program. Because of the HARV program, NASA would own a preproduction F/A-18 aircraft that would be used in an extensive flight test program. The X-29, on the other hand, was a joint program over which NASA had less control.

One of the first tasks of the FIT team was to develop a simulation of an F/A-18 that incorporated: (1) nonlinear, six degree-of-freedom, rigid body dynamics, (2) linear elastic modes, and (3) unsteady aerodynamics. At that time the team had access to a batch simulation computer program (*mdcf18*) and documents written by the McDonnell Aircraft Company of the McDonnell Douglas Corporation (MDC) for the F/A-18. The *mdcf18* simulation was a complete, nonlinear, six degree-of-freedom batch simulation that ran on a Digital Equipment Corporation VAX computer. The *mdcf18* simulation included the 8.3.3 flight control system, hinge-moment calculations, and a

full aerodynamic database. A considerable portion of the *mdcf18* simulation was devoted to failure modes of the propulsion and actuation systems. The FIT team also had access to a real-time simulation computer program of an F/A-18 that executed on a CDC Cyber 175 and was written by Unisys personnel for use with the Differential Maneuvering Simulator (DMS) at LaRC. The LaRC-DMS real-time simulation (*dmsf18*) was derived from the MDC simulation and documents provided by MDC. The FIT team chose to focus on dynamics issues rather than failure modes. There was also a requirement to interface closely with the LaRC real-time facilities and to begin setting up procedures whereby the transfer of specifications and code between researchers and the real-time simulation software development staff could be streamlined. The original FIT F/A-18 simulation [Arbuckle, et al. (1987) and Buttrill, et al. (1987)] was a batch simulation written in ACSL that included 6 nonlinear degrees-of-freedom and 20 elastic degrees-of-freedom, but no control system. The rigid-body aerodynamic model and the engine model of the original FIT F/A-18 simulation were largely derived from the LaRC DMS real-time simulation (*dmsf18*).

Another research focus at LaRC is agility at high angle-of-attack and low-to-moderate dynamic pressures. To support researchers working in this area, the fully elastic simulation developed by the FIT team was modified. The elastic dynamics were removed. The code that simulated the 8.3.3 flight control software in the LaRC-DMS real-time simulation was obtained and interfaced with the ACSL batch simulation. This report describes the resulting simulation code (*f18bas*).

The initial simulation code (*f18bas*) developed to support the Aircraft Guidance and Control Branch 's high-angle-of-attack research was frozen so that this document could be written. The *f18bas* simulation models a preliminary-design version of the thrust-vectoring system proposed for the HARV program. The preliminary-design thrust vectoring system employed two paddles per engine nozzle, whereas the actual system subsequently installed on the HARV demonstrator has three paddles per nozzle. The thrust vectoring capability of the *f18bas* simulation will only be described in a limited fashion as required to explain parameter lists. The simulation model without thrust-vectoring can be recovered by setting the logical, LTHVEC, to false.

One outgrowth of the *f18bas* simulation was the *f18harv* simulation. The *f18harv* simulation represents the Aircraft Guidance and Control Branch's best current model of the HARV demonstrator. The mathematical model for *f18harv* simulation is documented in a variety of internal memoranda and will be more formally described in a future report.

1.3 Code Development Philosophy

ACSL allows one to call Fortran subroutines from within ACSL source code. Since ACSL source code gets translated into Fortran before compilation, this is not surprising. If one chose, the calculation of all the state derivatives could be done in a Fortran subroutine. The derivatives could be returned as outputs from the subroutine and integrated by ACSL. ACSL could plot and record variables in the parameter list. The equations of motion would then be a black box. Linear analysis on the ACSL-integrated states could be performed. Alternatively, if one had an existing Fortran simulation, the entire Fortran simulation could be made into a subroutine and treated as, from the perspective of ACSL, a discrete process that returned outputs at current time, t, based on inputs at a previous time, t-h. The states of the Fortran subroutine would be hidden from ACSL and none of the linear analysis capability of ACSL could be used.

The goals adopted by the authors for building the simulation were: (1) all integrations arising from continuous processes would result in ACSL-defined states, and (2) the equations of motion, to the extent possible, would be coded in the ACSL language. Goal (2) was compromised slightly in that the calculation of aerodynamic forces and moments and engine steady state performance are done in Fortran subroutines. The authors had available software that greatly facilitated creating code to implement large table look-up type databases characteristic of aerodynamic force and moment

build-ups and the ACSL capabilities in this area are limited. Since not everyone has ACSL, coding the aerodynamic and engine static models in Fortran served to make major portions of the simulation model more accessible to the research community. The continuous dynamics of the simulation include actuator and sensor dynamics, engine dynamics, and the rigid-body equations of motion. Physically discrete dynamics, such as a digital flight control system, are not coded in ACSL. These dynamics are coded in Fortran subroutines and accessed by the simulation through subroutine calls which occur at discrete time increments using the DISCRETE block feature of ACSL. DISCRETE blocks are explained in the next section (Section 2).

2. SIMULATION STRUCTURE

The general structure of an ACSL simulation is described in section 3 of the ACSL reference manual [ACSL (1987)]. Section 2 of this report summarizes key elements of section 3 of the ACSL manual and will cast the specific structure of this simulation in the context of the discussion and figures of the ACSL reference manual.

ACSL imposes some limitations and programming rigor on its users. ACSL works best when the state derivatives are explicitly defined. If absolutely required, the IMPLICIT operator provided by ACSL can be invoked and the resulting simulation will attempt to solve numerically the implicit formulation, but this would result in unacceptable computing performance in all but the simplest models. Implicit loops are best solved by inserting fast filters or solving them algebraically.

ACSL creates a large common block (ZZCOM) that contains all the simulation variables. Included in ZZCOM are three arrays: the state array, the state derivative array, and the state initial condition array. A state is the output of an integration. A state variable in the state array cannot also be in the state derivative array.

2.1 Overview

The ACSL code structure shown in figure 2.1 is somewhat generic yet also contains the salient features of the F18 simulation. The DYNAMIC...END block of figure 2.1 results in the main simulation loop. The main simulation loop is indicated by the lower shaded box in the overview flow diagram of figure 2.2. In the main loop, if the termination condition, $T \geq TSTP$, is not met, the program will integrate forward in time over one communication interval, CINT [*ACSL* (1987)]. The communication interval determines how often the code in the dynamic block that is not part of either the derivative block or the discrete blocks gets executed. The default value for CINT in the *f18bas* simulation is 0.025 (1/40) seconds.

The maximum integration step for the DERIVATIVE block is defined by MXSTP in the *f18bas* simulation and has a default value of 0.00625 (1/160) seconds. The integration algorithm is selected by setting IALG. A variety of integration schemes are offered (IALG = 1 through 7) and are described in *ACSL* (1987). The default choice for *f18bas* simulation is IALG=4, a Runge-Kutta second-order method. The Runge-Kutta second-order method was recommended by the author of ACSL for most mechanical systems and has been found to work satisfactorily. No systematic investigation of integration methods has been made for this simulation. Using the Runge-Kutta second-order method, two derivative evaluations are made to advance time over one integration step, MXSTP.

The DISCRETE blocks are executed according to their INTERVAL or SCHEDULE statements. DISCRETE block *name1* in figure 2.1 is an example of an interval block that gets executed every *Tname* seconds. DISCRETE block *name2* in figure 2.1 is executed whenever the logical condition defined in its SCHEDULE command becomes true. This type of block might be used to reverse the velocity of a ball bouncing on a table. In the *f18bas* simulation, the digital control law DISCRETE block, FCS, is an INTERVAL block and is called every TFCS seconds. The default value for TFCS is 0.025 seconds. The choice of a value for TFCS is discussed in section 9 of this report. When different blocks are scheduled for execution at the same time on the event list, such as when a DISCRETE block interval is an even multiple of MXSTP or some other DISCRETE block interval, the order of execution is determined by order of appearance in the model definition code. It is not required that the DISCRETE interval period be a multiple of the maximum integration size for the DERIVATIVE block. An INTERVAL type DISCRETE block sets up a time barrier on an event list. ACSL will shorten the last integration step to insure all simulation variables are synchronized at the correct interrupt time.

PROGRAM

INITIAL

Includes statements to be executed at time zero after the runtime command "START" has been entered. Values in the the state IC array have not yet been transferred to the state array.

END

DYNAMIC

DERIVATIVE

Includes the statements that define the continuous system dynamic model. The elements of the state derivative array, the variables to be integrated, are defined here. A state is the output of an integrator.

SCHEDULE *name2* .XP.*var2*

END

DISCRETE *name1*

INTERVAL *Tname* = *number*

An INTERVAL scheduled DISCRETE includes statements that are executed every *Tname* seconds.

END

DISCRETE *name2*

Includes statements to be executed when the scheduling conditions calculated in the DERIVATIVE block (in this case) for DISCRETE *name2* become .true., .i.e. when variable *var2* crosses zero in the positive direction.

END

The remainder of the DYNAMIC block includes statements that define auxiliary variables that are required for output only. These statements are executed every CINT seconds. CINT is the default name for the communication interval.

TERMT(T.GE.TSTP)

END

TERMINAL

Includes statements to be executed when the input logical variable to the MACRO TERMT becomes .TRUE.

END

END

Figure 2.1. Outline of Typical ACSL Program

The paths in figure 2.2 associated with the variables LTR, ITFLG, LWRTR and LDBTR arise from code installed to trim the simulation. This path does not arise from any of the typical ACSL constructs found in figure 2.1. The trim logic is discussed in more detail in section 2.2 and figure 2.4 of this report

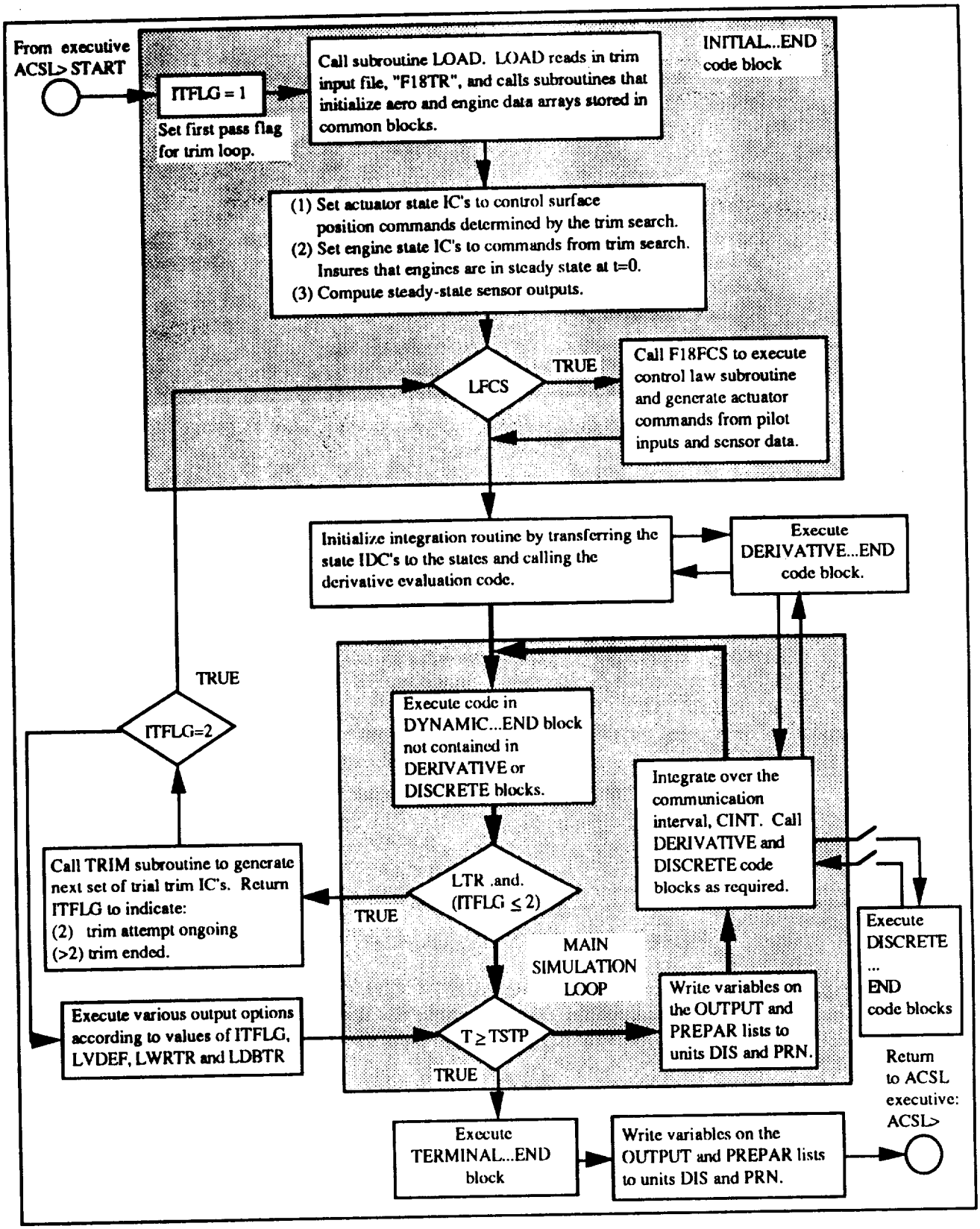


Figure 2.2. Overview of Simulation Flow

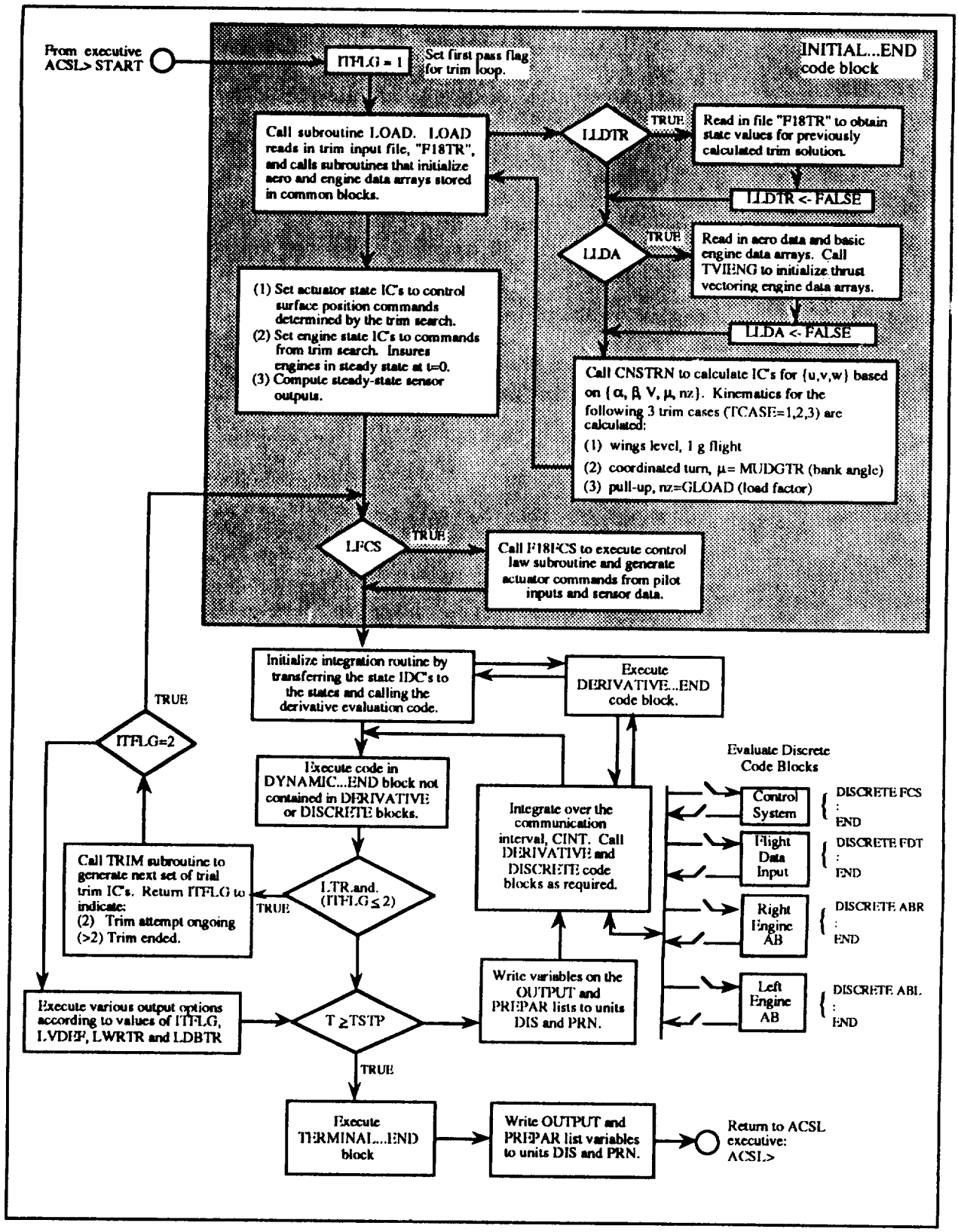


Figure 2.3. Initialization and Discrete Block Details

In figure 2.3, two sections of the simulation flow shown in figure 2.2 are expanded. The actions produced by calling subroutine LOAD in the INITIAL...END code block are detailed and the four discrete blocks in the simulation are shown. The discrete blocks that model the flight control system (FCS) and provide a mechanism for making comparisons with flight data (FDT) are invoked by INTERVAL commands. The discrete blocks associated with the engine afterburner models (ABR & ABL) are invoked by SCHEDULE statements.

2.2 Finding a Trim Solution

ACSL has a TRIM command that works with purely continuous models. To use the ACSL TRIM command, a subset of the simulation states (outputs of integrations) is selected using the command

ANALYZ 'CLEAR','RELEAS'={state list}.

The ACSL trim function works by adjusting the selected states until their state derivatives are zero. A combination of a Newton step and a steepest descent step is used (section 5.2.4 of ACSL (1987) to adjust the states. Without some modification, this is not what one generally wants in an airplane trim problem. For example, a simple longitudinal level-flight trim problem might be:

Find,

$$\{u\} = |\delta_e, T, \alpha, \theta\ ]' \tag{2.1}$$

such that

$$\{y\} = [\ \dot{\alpha},\ \dot{V},\ \dot{q},\ \dot{H}\ ]' = \{0\} \tag{2.2}$$

for fixed V and H, where δ_e, T, H, α, θ, V, q are elevator deflection, thrust, altitude, angle-of-attack, pitch angle, total airspeed and pitch rate, respectively.

In this case, elevator deflection and thrust are control inputs, and altitude, angle-of-attack, pitch angle, total airspeed and pitch rate are states. Trying to find α, V, q and H such that $\{y\}=\{0\}$ for fixed, arbitrary, values of δ_e, T and θ is almost always fruitless. To use the ACSL trim capability to solve the trim problem described in equations (2.1) and (2.2), one could add the following equations to the simulation model:

$$\dot{\delta}_e = k_1\, \dot{q} \tag{2.3}$$

$$\dot{T} = k_2\, \dot{V} + k_3\, \dot{H} \tag{2.4}$$

When attempting to trim, k_1, k_2 and k_3 would be set to something nonzero. The states $\{\delta_e, T, \alpha, \theta, q\}$ can then be released and the ACSL command "ANALYZ TRIM" invoked. ACSL will then drive $\{\ \dot{\delta}_e,\ \dot{T},\ \dot{\alpha},\ \dot{\theta},\ \dot{q}\ \}$ to zero and in the process, by virtue of equations (2.3) and (2.4), assure that $\dot{V}$, $\dot{q}$, $\dot{H}$ are zero. Equations (2.3) and (2.4) mimic the idea of a trim button or elevator trim-wheel. Generalizing this procedure to handle additional trim cases, such as trimming in a steady level turn or in a symmetric pull-up, could get cumbersome. In addition, it is not clear whether the ACSL trim feature could be made to work with a discrete-block control law in the loop. For these reasons, the authors chose to implement the alternative trimming strategy shown in figure 2.4. The trim strategy employed in *f18bas* is not without drawbacks, and a strong case can be made for retaining the ACSL structure and search algorithms as much as possible.

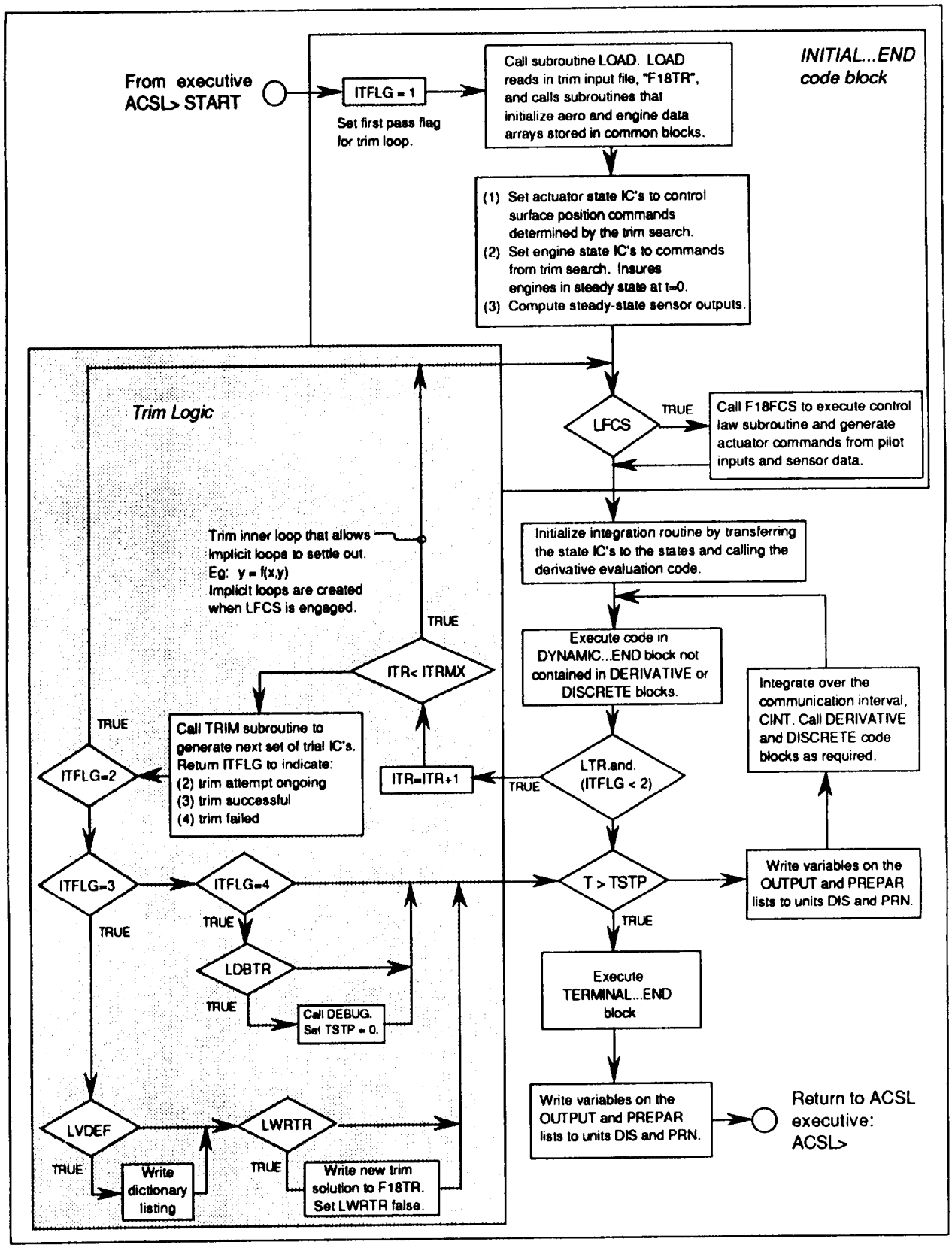


Figure 2.4. Trim loop expanded.

Details of the logic used to trim the simulation are shown in figure 2.4. Subroutine TRIM defines an array called XTRIM and an array called YTRIM. The first NXTR elements of XTRIM are varied until the the first NXTR elements of YTRIM are within some tolerance of zero. The allowable elements of XTRIM and YTRIM are shown in tables 2.1 and 2.2, respectively.

Table 2.1. Selectable members of XTRIM array

1	MACHTR	Mach number, n.d.
2	BETTR	sideslip angle, β, radians
3	ALFTR	angle of attack, α, radians
4	PIC	body axis roll rate, rad/sec
5	QIC	body axis pitch rate, rad/sec
6	RIC	body axis yaw rate, rad/sec
7	THETR	Euler pitch angle, radians
8	PHITR	Euler roll angle, radians
9	PSITR	Euler heading angle, radians
10	GAMTR	longitudinal flight path angle, γ, radians
11	DTVL	left engine thrust vectoring nozzle deflection, degrees
12	DTVR	right engine thrust vectoring nozzle deflection, degrees
13	PCATR	lateral stick deflection (pilot commanded aileron), inches
14	PCSTR	longitudinal stick deflection (pilot commanded stabilator), inches
15	PCRTR	rudder pedal force (pilot commanded rudder), lbs
16	DPSYTR	throttle position, symmetric, [31,127]
17	DPASTR	throttle position, antisymmetric, [96,–96]
18	DSSYTR	stabilator position, symmetric, degrees
19	DSASTR	stabilator position, antisymmetric, degrees
20	DASYTR	aileron position, symmetric, degrees
21	DAASTR	aileron position, antisymmetric, degrees
22	DRSYTR	rudder position, symmetric (both t.e. in), degrees
23	DRASTR	rudder position, antisymmetric (both t.e. left), degrees
24	DNSYTR	leading-edge flap position, symmetric, degrees
25	DNASTR	leading-edge flap position, antisymmetric, degrees
26	DFSYTR	trailing-edge flap position, symmetric, degrees
27	DFASTR	trailing-edge flap position, antisymmetric, degrees
28	CSB	commanded speed brake, degrees
29	MUDGTR	bank angle, μ, degrees
30		open for future expansion

Table 2.2. Selectable members of YTRIM array

1	UD	d/dt x-body component of inertial velocity, g's
2	VD	d/dt y-body component of inertial velocity, g's
3	WD	d/dt z-body component of inertial velocity, g's
4	PD	d/dt body-axis roll rate, rad/sec^2
5	QD	d/dt body-axis pitch rate, rad/sec^2
6	RD	d/dt body-axis yaw rate, rad/sec^2
7	GAMZR	= GAM - GAMTR = $\gamma - \gamma_{TR}$, radians
8	PHIZR	= PHI - PHITR = $\gamma - \gamma_{TR}$, radians
9	THE	Euler pitch angle, radians
10	LAMBDA	lateral flight path angle, λ, radians
11	FVRA(2)	aerodynamic a_y, g's
12	CLERR	= CLRFSF - CLTR = actual – target lift coefficient, n.d.
13	CMRFSF	steady flow pitch coefficient, n.d.
14,15		open for future expansion

The elements of XTRIM and YTRIM are selected by setting the first NXTR elements of the IXSEL and IYSEL arrays. A typical 4×4 longitudinal open-loop trim case could be set up as indicated in figure 2.5. In this example, Mach and altitude are selected a priori and the variables {α, θ, $\delta_{S_{sy}}$, power setting} are varied until {$\dot{u}$, $\dot{w}$, $\dot{q}$, γ-γ_{TR}} are driven to zero.

```
ACSL> NXTR = 4
ACSL> HIC=10000.,MACHTR=.5, GAMTR=0.0,LFCS=.F.
ACSL> IXSEL=3,7,16,18
ACSL> IYSEL=1,3,5,7
ACSL> START
```

Figure 2.5. Trim example 1

Trim example 2 in figure 2.6 is similar except that a trim with the flight control system engaged is sought. The pilot longitudinal stick command is substituted for stabilator position in the list of variables to be varied.

```
ACSL> NXTR = 4
ACSL> MACHTR=.5, GAMTR=0.0,LFCS=.T.
ACSL> IXSEL=3,7,16,14
ACSL> IYSEL=1,3,5,7
ACSL> START
```

Figure 2.6. Trim example 2

Achieving trim requires that all state derivatives are zero, not just the states associated with aircraft motion. The states associated with actuator, sensor, and control law dynamics must be trimmed. For the continuous actuator and sensor states, one approach would be to include these additional states explicitly in the nonlinear trim problem. This would serve to unnecessarily increase the dimension of the trim search. In addition, the trim architecture would become dependent on actuator and sensor transfer functions. For the transfer-function-based dynamics typically asociated with actuator and sensors, the calculation of the required state initial conditions to achieve trim for a given input is simple. Therefore, when in the trim loop, the dynamics of the actuator,

sensor, and control system are trimmed locally, not as part of the nonlinear search. Integrators are turned off and steady state values for all filters are calculated and passed through. Actuator and sensor state initial conditions are set so that the state derivatives are zero for the given inputs. However, local trimming of actuator and sensor dynamics creates an implicit loop in the global trim problem when the flight control system is engaged (LFCS set true - see section 9.2). For example, suppose trim pitch stick position (PCSTR) is one of the admissible XTRIM variables. A change in PCSTR produces a change in the control surface commands, which instantaneously produces (since the actuator dynamics are suppressed) a new control deflection. The change in control deflection produces a change in the vertical accelerometer output, n_{z_s}. Since n_{z_s} is a feedback, a change in n_{z_s} produces an additional change in the control surfaces, which produces new, slightly different n_{z_s}. The trim search assumes the following functional relationship:

$$\{\text{YTRIM}\} = f\{\text{XTRIM}\}$$

The effect of the implicit loop is to make {YTRIM} dependent not only on {XTRIM}, but on the past values of {YTRIM}. The degradation the functional relationship will impede or prevent convergence.

The somewhat ad hoc solution adopted by the authors was to implement an inner trim loop to converge the implicit loop (see figure 2.4). The inner loop increments a counter, ITR. The program loops through the control and flight dynamics code with constant trim driver and initial conditions values until ITR = ITRMX. The default value for ITRMX is 6.

Subroutine TRIM is called after the inner trim loop completes ITRMX passes. The trim routine perturbs the control commands and flight states specified for XTRIM and checks for convergence of the YTRIM values. When convergence is achieved, ITFLG is set to 3. If convergence fails, either because the maximum outer iteration count (MXITR) is exceeded or the sensitivity matrix becomes singular, ITFLG is set to 4. ITFLG will be set back to 1 at the first pass of each run (as shown in figure 2.4), and trimming will occur each time the program is started at the ACSL prompt, unless the variable LTR is set false. If LTR is false, the program will pass once through the initial section and go directly into the main program loop, bypassing the trim loop.

Trim convergence is determined in the ACTRML subroutine, which is called from the TRIM subroutine. The variable TRTOL (default = 0.00005) sets the tolerance criterion for convergence.

3. MASS AND GEOMETRY DATA

3.1 General Arrangement

The general arrangement of a single place F/A-18 is depicted in figure 3.1. This report and the MDC documents listed in the References locate points on the airframe using fuselage station (FS), buttock line (BL), and water line (WL) coordinates. The (FS,BL,WL) coordinates are defined in table 3.1. At the plane of symmetry, the BL coordinate is zero.

Table 3.1. (FS,BL,WL) coordinates of figure 3.1

FS	= positive aft, inches
BL	= positive out the right wing, inches
WL	= positive up, inches

The F/A-18 is a single place, mid-wing, high performance twin engine fighter capable of carrying a variety of armament loadings. For the F/A-18, the primary mission is that of "Fighter Escort". The Fighter Escort mission loading is two wingtip mounted Sidewinder missiles, two fuselage mounted Sparrow missiles, and the M61A1 gun. Each engine is a General Electric, F404-GE-400 rated at 16,100 lbs of uninstalled, static, sea level thrust. The total internal fuel capacity is 10,860 lbs while 6,732 lbs of fuel can be carried in external tanks. The design load factors are +7.5/-3.0 g's. The design sinking speed at touchdown is 25.1 ft/sec.

A moderately swept wing and a highly swept leading-edge extension (LEX) form the primary lifting surface. Leading-edge and trailing-edge flaps on the wing are used for carrier operations, subsonic maneuvering, and cruise. The flap deflections are scheduled with Mach number and angle of attack to provide improved drag and stability characteristics. All moving horizontal tail surfaces are mounted behind and below the wing to provide positive stability at high angles of attack. Twin vertical fins are both canted out and toed out (figure 3.1).

Pitch control is provided by the horizontal tails. Lateral control is accomplished with a blend of ailerons, differential flaps, and differential horizontal tail. Directional control is provided by a rudder on each vertical fin. A speedbrake is also available for deceleration control in the maneuvering configuration. [MDC A7247, p2-1]

3.2 Sign Conventions

While the (FS,BL,WL) coordinate system is used in various MDC reports and the MDC simulation to locate positions on the aircraft, the equations of motion make use of a right-hand body-fixed frame aligned with the (FS,BL,WL) and typically called the body frame. The body frame origin is at the center-of-gravity (c.g.) of the aircraft and is oriented so that X_B is positive forward, Y_B is positive out the right wing, and Z_B is positive down. For example, a sensor package that is located at (FS_σ, BL_σ, WL_σ) would have the following coordinates in the body frame,

$$\begin{Bmatrix} X_{B_\sigma} \\ Y_{B_\sigma} \\ Z_{B_\sigma} \end{Bmatrix} = \begin{Bmatrix} -(FS_\sigma - FS_{cg}) \\ BL_\sigma - BL_{cg} \\ -(WL_\sigma - WL_{cg}) \end{Bmatrix} \tag{3.1}$$

where (FS_{cg}, BL_{cg}, WL_{cg}) are the coordinates of the c.g. in the (FS, BL, WL) axes system.

The body frame, along with the stability frame, control surface sign conventions, and coefficient sign conventions, is depicted in figure 3.2. The stability frame is denoted by the subscript "S" in 3.2. A rotation of α of the stability frame about the Y_S axis brings the stability and body frames into alignment. The roll, pitch and yaw aerodynamic moment coefficients, C_l, C_m and C_n, are defined about body frame axes, X_B, Y_B, and Z_B, respectively. The lift, drag, and side force coefficients, C_L, C_D and C_Y, are defined as positive along the $-Z_S$ axis, the $-X_S$ axis and the $+Y_S$ axis, respectively. Note that Y_S is aligned with Y_B.

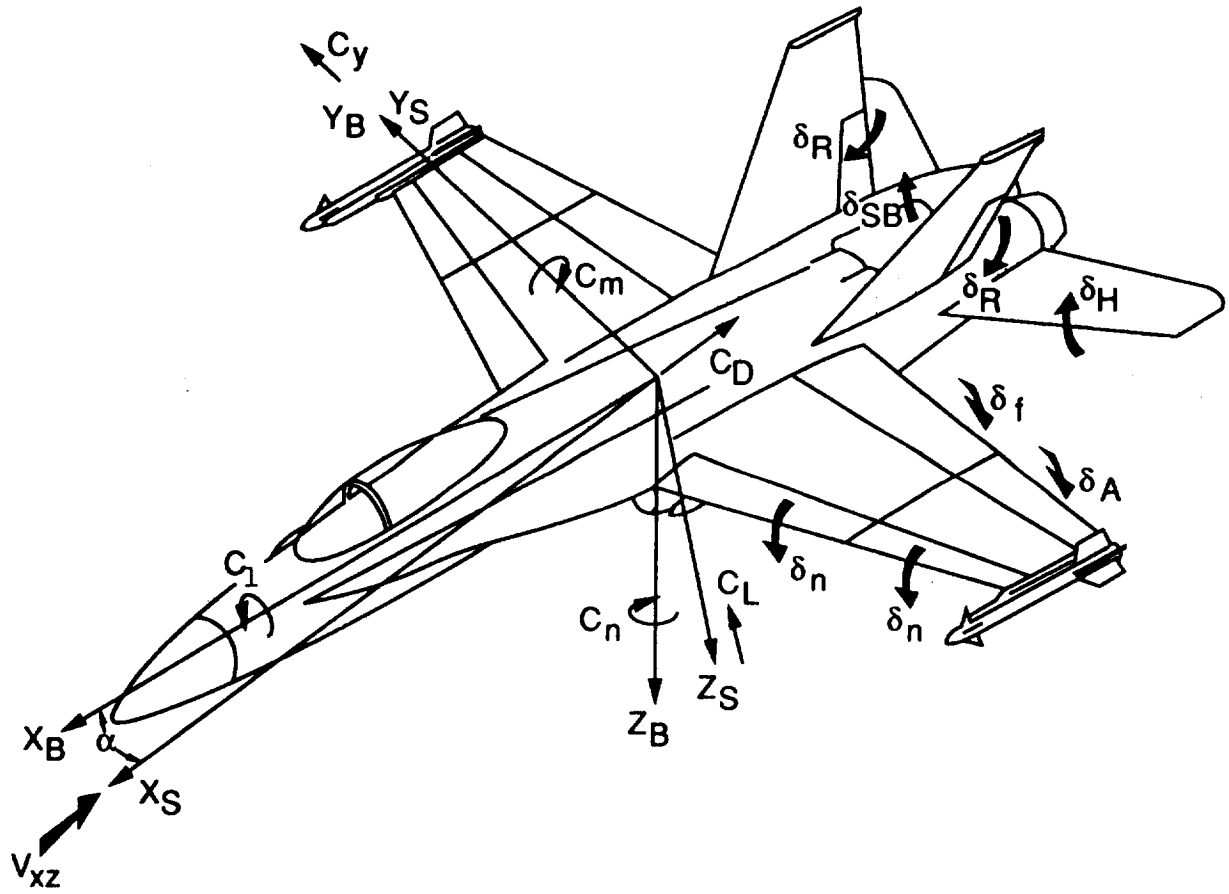


Figure 3.2. Axis System and Sign Convention

The control surface sign conventions, depicted in figure 3.2, apply to each member of a control surface pair and are described in table 3.2.

Table 3.2. Control Surface Sign Conventions

δ_n	leading-edge flap	positive leading edge down
δ_f	trailing-edge flap	positive trailing edge down
δ_A	aileron	positive trailing edge down
δ_H	horizontal tail	positive trailing edge down
δ_R	rudder	positive trailing edge left

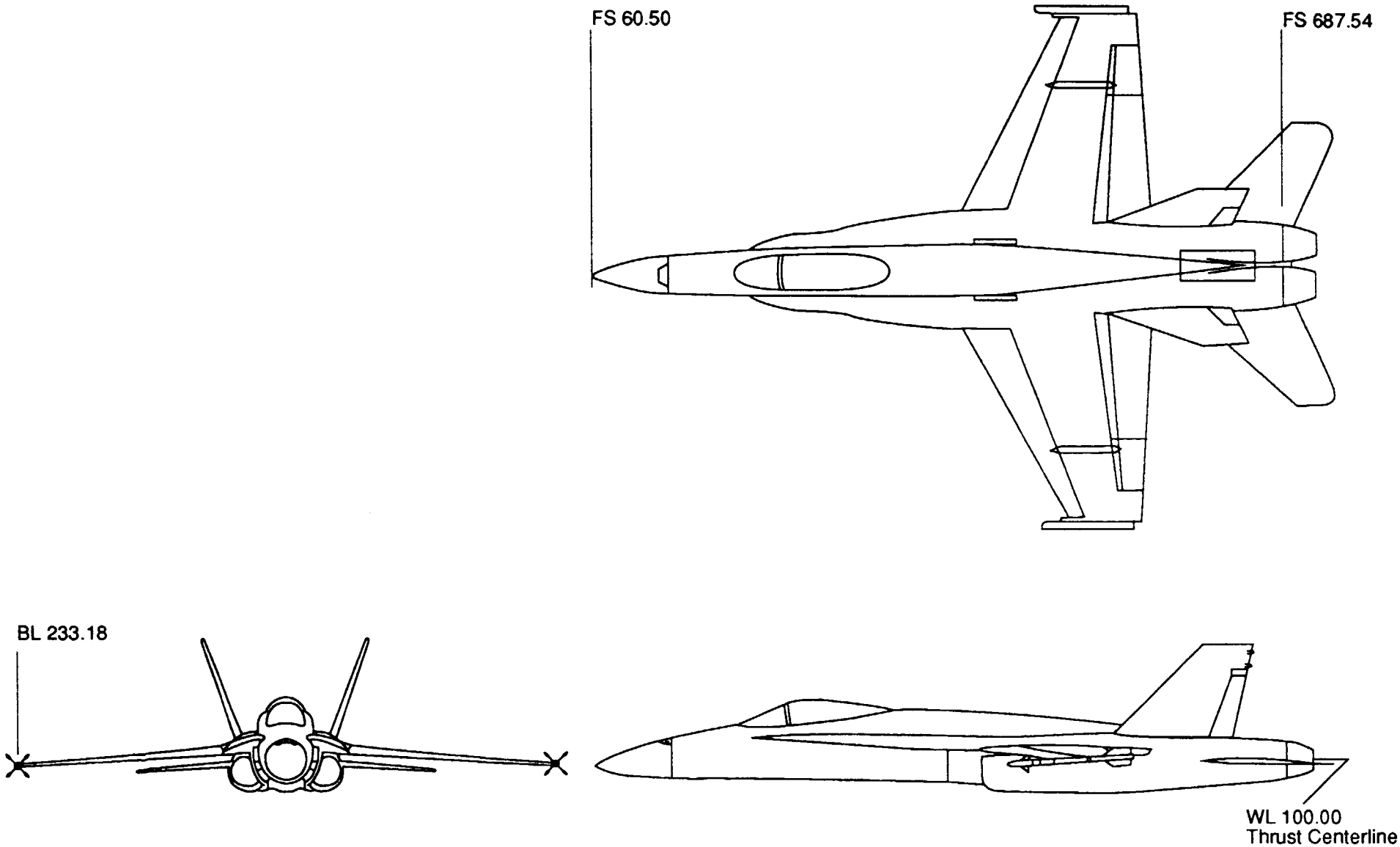


Figure 3.1 3-View Drawing

This page left intentionally blank.

3.3 Dimensional Data

The airplane component dimensional data are given in table 3.2 and pages 2.6-2.8 of MDC report A7247, vol I.

Table 3.3 Airplane Dimensional Data

Total Airplane	
Net Wetted Area (minus engine nozzles)	2028 ft^2
Overall length	56.0 ft
Overall Height	15.3 ft
Wing	
Area, S_{REF}	400 ft^2
Area (Actual)	401.7 ft^2
Wetted Area (Including launchers and aileron actuator fairings)	562 ft^2
Span, b_{REF}	37.42 ft
Aspect ratio, AR	3.5
Chords	
Root Chord (Theoretical)	15.86 ft
Root Chord (Exposed @ X_W 54.0)	13.19 ft
Tip Chord	5.52 ft
MAC, $\bar{c}_{REF}$	11.52 ft
Leading Edge (L.E.) Sweep	26.7°
$\bar{c}/4$ Sweep	20°
Taper Ratio	0.35
Dihedral	-3°
Geometric Twist	0.0° at X_W 145.39 4.0° L.E. down at tip
Incidence	0.0°
Airfoil Section	Modified NACA 65A with sharp L.E.
Airfoil Thickness at	
XW 56.876	5.0 % chord
XW 145.390	3.5 % chord
Tip Chord	3.5 % chord
Leading-Edge Flaps	
Type	Plain
Area	48.4 ft^2 total
Span	13.8 ft per side
Deflection (+ leading edge down)	[0.0°, 34°], maneuvering [12°, 34°],Takeoff and Landing $\pm$ 3° differential
Trailing-Edge Flaps	
Type	Single Slotted w/Drooped Shroud
Area	61.9 ft^2 total
Span	8.72 ft per side
Wing Span Range Covered (percent)	19-68.6 % b/2
Chord	30 % c
Deflections (+ trailing edge down)	[+17°, +30°], Takeoff [+17°, +45°], Landing

Table 3.3 Airplane Dimensional Data (continued)

Trailing-Edge Flaps (continued)	
Shroud	
Area	13.9 ft^2 total
Chord	7 % c
Hinge Line	68.5 % c
Deflection (+ down)	[-10°,+30°]
Ailerons	
Type	Single Slotted w/ Drooped Shroud
Area	24.4 ft^2 total
Span	5.68 ft per side
Percent Wing Span	68.8 -100% b/2
Deflections (+ trailing edge down)	[-25°, +45°], Takeoff and Landing [-25°, +25°], Maneuvering
Shroud	
Area	5.6 ft^2 total
Chord	7 % c
Hinge Line	68.5 % c
Deflection (+ down)	[-10°,+30°]
Leading Edge Extension (LEX)	
Planform Area (Slots Closed)	56.0 ft^2
Wetted Area	210.0 ft^2
Leading Edge Sweep	43°
Incidence (forward of inlet)	6°
LEX Intersection with Wing	F.S. 403.9 B.L. 54.0 W.L. 108.2
Horizontal Tails	
Exposed Area	88.1 ft^2
Wetted Area	176.0 ft^2
Aspect Ratio	2.4
Taper Ratio	.46
L.E. Sweep	47.2°
c/4 Sweep	42.8°
Dihedral	-2°
Span, b_{REF}	14.67 ft
Total Span	21.6 ft
Chords	
Root Chord (Exposed)	8.23 ft
Tip Chord	3.79 ft
MAC, $\bar{c}_{HT}$	6.28 ft
Tail length ($\bar{c}$/4 to $\bar{c}_{HT}$/4)	16.81 ft
Airfoil section	Modified NACA 65A with Sharp L.E.
Airfoil Thickness at	
Root	6 % $\bar{c}_{HT}$
Tip	2 % $\bar{c}_{HT}$
Hinge Line	13 % $\bar{c}_{HT}$

Table 3.3 Airplane Dimensional Data (concluded)

Horizontal Tails (continued)	
Deflections (+ trailing edge down)	[-24°, +8°], Symmetric [-24°, +10.5°], Max
Vertical Tails	
Area	52.0 ft^2 each
Wetted Area	104.0 ft^2 each
Aspect Ratio	1.2
Taper Ratio	0.4
L.E. Sweep	41.3°
c/4 Sweep	35.0°
Cant (Tip out)	20° out
Toe Out	1°
Panel Span	7.92 ft
Chords	
Root Chord (Exposed)	9.42 ft
Tip Chord	3.75 ft
MAC, $\bar{c}_{VT}$	6.99 ft
Tail length ($\bar{c}$/4 to $\bar{c}_{VT}$/4)	10.18 ft
Airfoil section	Modified NACA 65A with Sharp L.E.
Airfoil Thickness at	
Root	5 % $\bar{c}_{VT}$
Tip	3 % $\bar{c}_{VT}$
Rudders	
Area	7.72 ft^2 each
Span	5.21 ft
Percent Chord	20 % c
Deflection	$\pm$ 30°
Fuselage	
Length	53 ft
Maximum Width	7.6 ft
Wetted Area (includes nacelles, canopy, and nozzles)	890 ft^2
Speedbrake	
Planform Area	13.9 ft^2
Span	2.5 ft
Chord	5.57 ft
Maximum Deflection (relative to ML)	60°

3.4 Aerodynamic Reference Dimensions

The aerodynamic reference dimensions are given in table 3.4 below and in MDC report A7247, page 3-2.

Table 3.4. Aerodynamic Reference Dimensions

Wing Area, S_{REF}	400 ft^2
Wing MAC, $\bar{c}_{REF}$ (L.E.MAC @ FS 423.99)	138.275 in = 11.523 ft
Wing Span, b_{REF}	37.42 ft
Horizontal Tail Moment Arm, l_{HT} ($\bar{c}$/4 to $\bar{c}_{HT}$/4)	201.76 in
Horizontal Tail Volume Coefficient, $\bar{V}_{HT}$	.321
Vertical Tail Moment Arm, l_{VT} ($\bar{c}$/4 to $\bar{c}_{VT}$/4)	122.1 in
Vertical Tail Volume Coefficient, $\bar{V}_{VT}$	.0707

3.5 Weight, Center of Gravity, and Inertia Data

The weight, center-of-gravity location, and mass distribution data for the default simulation configuration and for a fighter escort configuration are shown in Table 3.5. The fighter escort description, along with several others, is found in figure 2.3-1 of MDC report A7247. The default configuration represents an early estimate of an F18 HARV configuration after the addition of thrust vectoring hardware. Of all the configurations listed in figure 2.3-1 of MDC report A7247, the data for the fighter escort configuration with 60 percent internal fuel most closely matches the default values.

Table 3.5. Summary of Simulation Weight, Center-of-Gravity, and Inertia

		c.g. Locations		Moments and Products of Inertia			
	Weight	FS	WL	I_{xx}	I_{yy}	I_{zz}	I_{xz}
	(lbs)	(in)		(slug - ft^2)			
Default	33310	455.0	102.8	23000	151293	169945	-2971
Fighter Escort (60% internal fuel)	31665	457.3	101.6	22337	120293	138945	-2430

The I_{xz} product of inertia in table 3.5 is defined as the integral over the volume of the aircraft of the product (xz) (body frame coordinates) weighted with the mass density, σ, of the aircraft. The negative of this quantity is what appears as the (1,3) and (3,1) elements of inertia matrix. Thus,

$$I_{xz} = \iiint \sigma \, xz \, dV \tag{3.2}$$

The product I_{xz} is typically positive for aircraft with vertical fins but is negative in this case.

4. EQUATIONS OF MOTION

The derivation of the equations governing the motion of a rigid airplane over a flat, nonrotating Earth has been performed many times in the literature and is not likely to be improved upon here. Readers interested in the derivation of said equations are referred to Etkin (1972) and Roskam (1979). In the *f18bas* simulation, mass and inertia properties are constants and do not vary with time. If desired, the mass and inertias could be made functions of fuel load according to data presented in MDC-A7247 and MDC-A8575. Gravity is assumed to be aligned with the +Z axis of a local (Earth) right-handed reference frame fixed at sea level. Gravity is assumed constant over the airplane volume, which results in gravity acting as a force at the center of mass and producing no moments. In fact, gravity is constant with altitude with the default value being the sea level value. A reference frame fixed to the body of the airplane (body frame) is the most convenient for expressing the applied aerodynamic and thrust loads. As indicated in figure 3.2, the body frame origin is at the center of gravity and the frame is oriented so that $+X_B$ is out the nose, $+Y_B$ is out the right wing, and $+Z_B$ is out the belly. The airplane is assumed symmetric about the plane formed by the X_B and Z_B axes, so that the I_{xy} and the I_{yz} products of inertia are zero and are not included in the simulation equations.

4.1 Translational Equations

Let the translational inertial velocity of the airplane expressed in body frame components be given by [u,v,w]' and the rotational inertial velocity of the airplane in body frame components be given by [p,q,r]'. The equations governing translational motion can then be written as,

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

$$= \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}_G + \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}_A + \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}_E$$

where the subscripts "G","A", and "E" denote gravitational, aerodynamic and thrust-induced forces. The components of the gravitational force are given by,

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}_G = -mg \begin{Bmatrix} l_{xz} \\ l_{yz} \\ l_{zz} \end{Bmatrix} = mg \begin{Bmatrix} -\sin(\theta) \\ \cos(\theta)\sin(\phi) \\ \cos(\theta)\cos(\phi) \end{Bmatrix} \tag{4.2}$$

where $[l_{xz}\ l_{yz}\ l_{zz}]'$ is the third column of the direction cosine matrix describing a transformation from local frame to body frame components and θ and ϕ are the Euler pitch and roll angles, respectively. The aerodynamic force components are given by,

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}_A = \bar{q}S \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \tag{4.3}$$

The thrust-induced forces are defined in equations (6.8) of section 6.

4.2 Rotational Equations

The body frame rotational equations can be written in a form identical to equation (4.1), with the understanding that moments due to gravity are zero,

$$\frac{d}{dt}\begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}\begin{Bmatrix} p \\ q \\ r \end{Bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}\begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}\begin{Bmatrix} p \\ q \\ r \end{Bmatrix}$$

$$= \begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix}_G + \begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix}_A + \begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix}_E \tag{4.4}$$

and where the aerodynamic moments are given by,

$$\begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix}_A = \bar{q}S \begin{Bmatrix} b\,C_l \\ \bar{c}\,C_m \\ b\,C_n \end{Bmatrix} \tag{4.5}$$

The thrust-induced moments are defined in equations (6.10) of section 6.

4.3 Combined Accelerations

If the body frame components of translational and rotational velocity are combined into a 6×1 generalized velocity vector,

$$\{V\} = \begin{Bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{Bmatrix} \tag{4.6}$$

then equations (4.1) and (4.2) can be combined and written as,

$$[M]_I \{\dot{V}\} + [\Omega][M]_I \{V\} = \{F\}_G + \{F\}_A + \{F\}_E \tag{4.7}$$

where the inertial mass is given by,

$$[M]_I = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & -I_{xz} \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & -I_{xz} & 0 & I_z \end{bmatrix} \tag{4.8}$$

and the matrix representing body frame rotation is,

$$[\Omega] = \begin{bmatrix} 0 & -r & q & 0 & 0 & 0 \\ r & 0 & -p & 0 & 0 & 0 \\ -q & p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -r & q \\ 0 & 0 & 0 & r & 0 & -p \\ 0 & 0 & 0 & -q & p & 0 \end{bmatrix} \tag{4.9}$$

In order to achieve an explicit definition of the airplane accelerations, equation (4.7) must be modified. Observe that the time rate-of-change of angle of attack in still air, given by,

$$\dot{\alpha} = \frac{u\dot{w} - w\dot{u}}{u^2 + w^2} \tag{4.10}$$

is clearly a function of body frame accelerations. In the F/A-18 aerodynamic build-up, as is the case for almost all such aerodynamic models, the coefficients of lift and pitch contain contributions by a nondimensionalized form of the angle-of-attack rate-of-change

$$C_X = \ldots + C_{X_{\dot{\alpha}}} \left(\frac{\dot{\alpha}\bar{c}}{2V} \right) \tag{4.11}$$

where the symbol 'X' in equation (4.11) can be replaced by 'L' (lift) or 'm' (pitch). In the F/A-18 simulation, $C_{X_{\dot{\alpha}}}$ is a function of Mach, altitude, and angle-of-attack. The implication of equations (4.10) and (4.11) is that the generalized aerodynamic force column vector appearing in equation (4.7) should be written as,

$$\{F\}_A = \{F\}_A\Big|_{\dot{\{V\}}=0} - [M]_A\{\dot{V}\} \tag{4.12}$$

where,

$$[M]_A = -\frac{\partial \{F\}_A}{\partial \dot{\{V\}}} \tag{4.13}$$

$[M]_A$ represents an increment to the inertial mass matrix of equation (4.7). If we further define a generalized inertial force as,

$$\{F\}_I = -[\Omega][M]_I\{V\} \tag{4.14}$$

and combine equations (4.7), (4.12) and (4.14), we get the equations of motion as implemented in the simulation,

$$[M]\dot{\{V\}} = \{F\}_{I+G+A+E} \tag{4.15}$$

where the total mass matrix is:

$$[M] = [M]_I + [M]_A \tag{4.16}$$

Equation (4.15) must be solved each simulation time step for the unknown accelerations in $\dot{\{V\}}$. Recall $\{V\} = [u,v,w,p,q,r]'$ (equation 4.6). If the mass matrix, [M], were constant, then the inverse, $[M]^{-1}$, could be formed at time zero and a simple left multiplication of the right-hand-side forces by $[M]^{-1}$ would produce the accelerations. The aerodynamic part of the mass matrix, $[M]_A$, is a function of angle of attack, Mach number, altitude, and dynamic pressure and is therefore time-varying. Forming the inverse of [M] each time step would be prohibitive. The approach used in the simulation documented in this report is to solve (4.15) using Gaussian elimination. In a typical application of Gaussian elimination, row and column switching (pivoting) is performed to insure that the largest possible elements are used as divisors. Both partial and full pivoting strategies exist. In a general application, if the diagonal elements are used as divisors without pivoting, the divisors may become zero. For most airplanes, and for any configuration of the F/A-18, the inertial mass matrix, $[M]_I$, is sufficiently diagonally dominant and $[M]_A$ is sufficiently small throughout the flight envelope, that the diagonal elements of [M] can be used as divisors with impunity. A custom 6×6 equation solver that employs no pivoting strategy is used in the simulation to solve (4.15). The custom solver is implemented in subroutine XRBVDC. In order to improve efficiency, the use of DO-LOOPS in XRBVDC was minimized. The simulation was tested using both the custom solver and a general purpose subroutine that employed a full pivoting strategy (the most robust form) with double precision arithmetic. The custom solver gave identical answers to 5-6 decimals and achieved a substantial reduction in total simulation run time.

An alternate approach to the solution of (4.15) is to use a series approximation to the inverse of [M]. This approach is valid if the time-varying part of the mass matrix is small. Thus if,

$$[M] = [M_I][I + \Delta M] \tag{4.17}$$

then,

$$[M]^{-1} = [I + \Delta M]^{-1}[M_I]^{-1} \approx [I - \Delta M][M_I]^{-1} \tag{4.18}$$

if [ΔM] is small. The inversion of $[M_I]$ could be done once in the initialization portion of the simulation.

4.4 Body Orientation Parameters

Three equivalent sets of parameters to describe the orientation of the body reference frame with respect to the the Earth frame are maintained in the simulation. The three sets are the Euler angles (ψ, θ, ϕ), the quaternions (e_0, e_1, e_2, e_3), and the elements of the direction cosine matrix (l_{xx}, l_{xy}, ..., l_{zz}). These parameters are calculated by two separate methods according to the logical switch, LEULER. If LEULER is selected, then the three sets of orientation parameters are determined by integrating the time derivatives of the Euler angles, $\dot{\psi}$, $\dot{\theta}$, $\dot{\phi}$. If LEULER is not selected, then all the orientation parameters are determined from integrating the time derivatives of the quaternions (Euler parameters). The quaternion method was implemented first and avoids the problem of a singularity that occurs in the Euler angle derivatives when the pitch angle, θ, is 90°. The singularity is associated with θ because of the order of application of the Euler angle rotations in transforming from the Earth frame to the body frame, that order being heading (ψ), pitch (θ), and roll (ϕ). If the $\theta = 90°$ region is avoided, the two methods should produce the same answer to 3-4 decimals. The Euler angle integrations were added later so the Euler angles would occur as states in the simulation and therefore linear analysis using the Euler angles as state variables could be easily performed. Time derivatives are calculated and integrated for both the Euler angles and the quaternions at all times in the simulation. The effect of LEULER is to select one set of integrated outputs and ignore the other. Equations 4.19 through 4.31 below can be found among or derived from the equations in Appendix E of Gainer and Hoffman (1972).

4.4.1 Euler Angles

The time derivatives of the Euler angles are given by:

$$\left.\begin{aligned} \dot{\phi} &= p + \{\cos(\theta)\}^{-1}\left(\sin(\phi)\sin(\theta)q + \cos(\phi)\sin(\theta)r\right) \\ \dot{\theta} &= \cos(\phi)q - \sin(\phi)r \\ \dot{\psi} &= \{\cos(\theta)\}^{-1}\left(\sin(\phi)q + \cos(\phi)r\right) \end{aligned}\right\} \qquad (4.19)$$

In the simulation implementation of (4.19), the following expression is used for $\cos(\theta)$ where division by $\cos(\theta)$ is called for,

$$C'_\theta = \begin{cases} \varepsilon \dfrac{\theta}{|\theta|} & \text{if } |\cos(\theta)| < \varepsilon = 10^{-10} \\ \cos(\theta) & \text{otherwise} \end{cases} \qquad (4.20)$$

so that division by zero when the pitch angle is at 90° is avoided. States created by integrating the derivatives are subscripted with "x" to denote the fact that they may be ignored later depending on the choice of LEULER.

$$\left.\begin{aligned} \psi_x &= \int_0^t \dot{\psi} dt + \psi_{ic} \\ \theta_x &= \int_0^t \dot{\theta} dt + \theta_{ic} \\ \phi_x &= \int_0^t \dot{\phi} dt + \phi_{ic} \end{aligned}\right\} \tag{4.21}$$

4.4.2 Quaternions

The derivatives of the quaternions are given by:

$$\begin{Bmatrix} \dot{e}_0 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{Bmatrix} = \begin{bmatrix} e_0 & -e_1 & -e_2 & -e_3 \\ e_1 & e_0 & -e_3 & e_2 \\ e_2 & e_3 & e_0 & -e_1 \\ e_3 & -e_2 & e_1 & e_0 \end{bmatrix} \begin{Bmatrix} 0 \\ p \\ q \\ r \end{Bmatrix} \tag{4.22}$$

Define

$$\mathbf{e} = e_0 + \mathbf{i}e_1 + \mathbf{j}e_2 + \mathbf{k}e_3 = \begin{Bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{Bmatrix} \quad \text{and} \quad \mathbf{w} = \mathbf{i}p + \mathbf{j}q + \mathbf{k}r = \begin{Bmatrix} 0 \\ p \\ q \\ r \end{Bmatrix}$$

where the following rules of multiplication govern the i, j, k basis vectors.

$$\mathbf{ij} = -\mathbf{ji} = \mathbf{k},\ \mathbf{jk} = -\mathbf{kj} = \mathbf{i},\ \mathbf{ki} = -\mathbf{ik} = \mathbf{j},\ \text{and}\ \mathbf{ii} = \mathbf{jj} = \mathbf{kk} = -1.$$

Note that the complex conjugate of e, the quaternion vector, is given by $\bar{\mathbf{e}} = e_0 - \mathbf{i}e_1 - \mathbf{j}e_2 - \mathbf{k}e_3$. Observing that the product ew is given by,

$$\mathbf{ew} = \begin{aligned} &-pe_1 - qe_2 - re_3 \\ &+ \mathbf{i}[pe_0 - qe_3 + re_2] \\ &+ \mathbf{j}[pe_3 + qe_0 - re_1] \\ &+ \mathbf{k}[-pe_2 + qe_1 + re_0] \end{aligned} \tag{4.23}$$

Equation (4.22) can be rewritten as,

$$\dot{\mathbf{e}} = \mathbf{ew} \text{ or } \dot{\bar{\mathbf{e}}} = \bar{\mathbf{w}}\bar{\mathbf{e}} \text{ where } \mathbf{e} = \begin{Bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{Bmatrix},\ \bar{\mathbf{e}} = \begin{Bmatrix} e_0 \\ -e_1 \\ -e_2 \\ -e_3 \end{Bmatrix} \tag{4.24}$$

where left multiplication by **e** is equivalent to left multiplication by the 4×4 matrix of equation 4.22. The components of the quaternion rate, $\dot{\mathbf{e}}$, are then integrated to produce quaternion states.

$$e_{j0_x} = \int_0^t \dot{e}_j \, dt + e_{j_{ic}} \quad (j = 0,3) \tag{4.25}$$

The quaternions are used as the generating set from which the other two equivalent set of orientation parameters are derived. The quaternions in their unnormalized form are calculated according to the value of LEULER. The additional zero subscript on the quaternions serves to indicate their unnormalized status.

$$e_{0_0} = \begin{cases} \cos(\frac{\psi_x}{2})\cos(\frac{\theta_x}{2})\cos(\frac{\phi_x}{2}) + \sin(\frac{\psi_x}{2})\sin(\frac{\theta_x}{2})\sin(\frac{\phi_x}{2}) & \text{LEULER=.T.} \\ e_{0_0} = e_{0_{0_x}} & \text{LEULER=.F.} \end{cases} \tag{4.26.a}$$

$$e_{1_0} = \begin{cases} \cos(\frac{\psi_x}{2})\cos(\frac{\theta_x}{2})\sin(\frac{\phi_x}{2}) + \sin(\frac{\psi_x}{2})\sin(\frac{\theta_x}{2})\cos(\frac{\phi_x}{2}) & \text{LEULER=.T.} \\ e_{1_0} = e_{1_{0_x}} & \text{LEULER=.F.} \end{cases} \tag{4.26.b}$$

$$e_{2_0} = \begin{cases} \cos(\frac{\psi_x}{2})\sin(\frac{\theta_x}{2})\cos(\frac{\phi_x}{2}) + \sin(\frac{\psi_x}{2})\cos(\frac{\theta_x}{2})\sin(\frac{\phi_x}{2}) & \text{LEULER=.T.} \\ e_{2_0} = e_{2_{0_x}} & \text{LEULER=.F.} \end{cases} \tag{4.26.c}$$

$$e_{3_0} = \begin{cases} \sin(\frac{\psi_x}{2})\cos(\frac{\theta_x}{2})\cos(\frac{\phi_x}{2}) + \cos(\frac{\psi_x}{2})\sin(\frac{\theta_x}{2})\sin(\frac{\phi_x}{2}) & \text{LEULER=.T.} \\ e_{3_0} = e_{3_{0_x}} & \text{LEULER=.F.} \end{cases} \tag{4.26.d}$$

The quaternions are then normalized. Note that if ψ, θ, ϕ are small, then to first order, $\psi = e_3$, $\theta = e_2$, $\phi = e_1$, and $e_0 = 1$. The norm of the quaternion vector is formed,

$$|e| = \left\{ e_{0_0}^2 + e_{1_0}^2 + e_{2_0}^2 + e_{3_0}^2 \right\}^{1/2} \tag{4.27}$$

and then used to scale the quaternion components:

$$e_j = e_{j_0} / |e| \quad (j=0,3) \tag{4.28}$$

While not required for the following development, it is interesting to observe that:

$$\begin{Bmatrix} |e|^2 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = e\bar{e} \tag{4.29}$$

4.4.3 Direction Cosine Matrix

The direction cosine matrix elements and the final Euler angle can then be derived from the normalized quaternions:

$$\left.\begin{aligned} \begin{Bmatrix} 0 \\ l_{xx} \\ l_{yx} \\ l_{zx} \end{Bmatrix} &= \bar{e} \begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{Bmatrix} e \\ \begin{Bmatrix} 0 \\ l_{xy} \\ l_{yy} \\ l_{zy} \end{Bmatrix} &= \bar{e} \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{Bmatrix} e \\ \begin{Bmatrix} 0 \\ l_{xz} \\ l_{yz} \\ l_{zz} \end{Bmatrix} &= \bar{e} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} e \end{aligned}\right\} \tag{4.30}$$

where in expanded form, the first equation of (4.30) is given by,

$$\begin{Bmatrix} 0 \\ l_{xx} \\ l_{yx} \\ l_{zx} \end{Bmatrix} = \begin{bmatrix} e_0 & e_1 & e_2 & e_3 \\ -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix} \begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{Bmatrix} \begin{bmatrix} e_0 & -e_1 & -e_2 & -e_3 \\ e_1 & e_0 & -e_3 & e_2 \\ e_2 & e_3 & e_0 & -e_1 \\ e_3 & -e_2 & e_1 & e_0 \end{bmatrix} \tag{4.31}$$

Finally, the Euler angles can be calculated directly from the elements of the direction cosine matrix.

$$\left.\begin{aligned} \psi &= \tan^{-1}\left\{\frac{l_{xy}}{l_{xx}}\right\} \\ \theta &= \sin^{-1}\{l_{xz}\} \\ \phi &= \tan^{-1}\left\{\frac{l_{yz}}{l_{zz}}\right\} \end{aligned}\right\} \tag{4.32}$$

5. AERODYNAMIC MODEL

5.1 Aerodynamic Data Sources

The aerodynamic characteristics of the F/A-18 are discussed in detail in MDC A7247, vol. 1,2. The low α report, (MDC A7247, vol. 1), describes the force and moment build-up for three basic configurations (maneuvering, catapult and approach) in the angle-of-attack range of -10° to $+40^\circ$. The maneuvering configuration is described in section 3 of MDC A7247, vol 1, while the low speed/high lift configurations (approach and catapult) are described in section 4 of MDC A7247, vol 1. MDC A7247, vol. 2 describes the force and moment build-up for the maneuvering configuration in the angle-of-attack range of $+40^\circ$ to $+90^\circ$. In the MDC batch simulation code (discussed in section 1.2 of this report) four separate force and moment build-ups; "clean", "power approach","take-off", and "high" are implemented. The first three configurations of the MDC batch simulation code, ("clean", "power approach", and "take-off"), correspond to the three configurations of MDC A7247, vol. 1, ("maneuvering", "approach", and "catapult"). The fourth build-up in the MDC batch simulation code corresponds to the high-angle-of-attack, maneuvering configuration of MDC A7247, vol. 2. The "maneuvering" configuration equations of MDC A7247, vol. 1 and the "clean" configuration equations of the MDC batch simulation code are slightly different. The MDC A7247, vol. 1 "maneuvering" build-up has increments due to stores while the MDC batch simulation code "clean" build-up has increments due to gear deployment. The simulation aerodynamic model described in this report is a combination of the maneuvering configurations described in MDC A7247, vol. 1,2 and in the MDC batch simulation code with a few small extensions extracted from the high-lift configurations. In addition, MDC report A8575 contains background material, including a history of configuration development, a history of the substantiating wind-tunnel testing, and propulsion system characteristics.

As required by the simulation equations of motion discussed in section 4 of this report, the total aerodynamic force and moment coefficients are calculated in two parts. The portion of the total coefficient that is an explicit function of airplane velocities or control surface positions, the steady flow part, is calculated in subroutine SFAERRF. The part of the total coefficient that is proportional to airplane acceleration, the unsteady part, is calculated in subroutine USAERRF. The steady flow result is the sum of the static components plus the effect of body angular rates. The unsteady part includes the effect of the time rate of change of angle of attack or sideslip.

The moment coefficients and input velocities are defined about the aerodynamic reference center, (a.r.c.), which is defined in Table 5.1.

Table 5.1. Aerodynamic Reference Center Coordinates

Fuselage Station (FS), in.	458.56 (+ aft)
Buttock Line (BL), in.	0.00 (+ right)
Water Line (WL), in.	100.00 (+ up)

The (FS/BL/WL) coordinates define a body-fixed axes system. The body axes system used in the equations of motion is centered at the center of gravity, aligned with the (FS/BL/WL) system, but has the sign convention of +x forward, + y out the right wing, and +z down (figure 3-1).

There are two logical variables that affect the coefficient calculations in the simulation described in this report, LQSE and LRTE. If the quasi-static-elastic logical, (LQSE), is true, then all the flex/rigid increments and effectiveness factors that were in the original model remain in effect. If LQSE is

false, then all the flex/rigid coefficients are set to the values they approach as air density, and therefore deformation, goes to zero. Setting LQSE false causes the aerodynamic forces and moments appropriate to the wind-tunnel-shape, which is not necessarily the jig-shape, of the airplane to be calculated. The LQSE option allows the use of the same code in a simulation where the flexible modes are actively modeled. The second logical is the real-time-equivalent flag (LRTE). If LRTE is true, then various differences that exist between the aerodynamic model used in the LaRC real-time simulation (*dmsf18*) and the ACSL simulation described in this report are removed. These differences are discussed in section 5.5.2 "(Leading-Edge Flap Effects (Y,l,n)"), section 5.5.5 ("Rudder Effects"), and section 5.5.9 ("Effect of Airplane Asymmetries (Y,l,n)") of this report.

The difference between the two aerodynamic models arises from changes made to the *f18bas* simulation math model that were not duplicated in the LaRC real-time simulation. The first of these changes was the inclusion of a rolling moment increment, $\Delta C_{l_{asm}}$, that is a function only of angle of attack, α, and nonzero for $\alpha > 40°$. The effect is to introduce a rolling disturbance as the simulated airplane moves through the 40° to 60° angle-of-attack region. The term arises from nose asymmetries and is discussed on pages 4-20,21 of MDC A7247, vol. 2 and section 5.5.9 ("Effect of Airplane Asymmetries (Y,l,n)") of this report. The *dmsf18* simulation has the term $\Delta C_{l_{asm}}$ zeroed out. The second change is the inclusion of a nonzero estimate for $C_{L_{\delta_n}}$ (lift coefficient per unit leading-edge flap) for the case of $40° < \alpha \leq 45°$ and leading-edge flap position different from the scheduled value. This change is discussed in section 5.3.4 ("Leading-Edge Flap Effects") of this report. The third difference is the inclusion of rudder toe-in effects on drag, pitch, and lift in the ACSL simulation. Rudder toe-in can induce significant pitch moment and is an appropriate effect to include in a simulation supporting agility research. Symmetric rudder effects on pitch are discussed in section 5.3.9 ("Rudder Effects") of this report.

In the reconstruction equations, the total aerodynamic force and moment coefficients are built up from a large collection of intermediate coefficients. An attempt has been made to adopt symbolic conventions that indicate the source and purpose of the intermediate coefficients. Typically, the total force or moment coefficient consists of a "basic quantity" plus additional coefficients. The "basic contribution" is denoted by a zero subscript and is a function of Mach and either angle of attack or sideslip. For the "basic contribution", the controls are at a nominal position (typically zero) and the airplane is undeformed. Additional coefficients that represent increments to another coefficient (basic or otherwise) will begin with 'Δ'. Increments due to 'x' are the result of evaluating a tabular function where 'x' was an input. Coefficients that resemble classic stability derivatives and are to be multiplied by the appropriate control or nondimensional quantity will have no 'Δ' symbol.

5.2 Control Surface Sign Conventions

The control surface sign convention used in the aerodynamic model follows the convention established in MDC A7247, vol. 1. The terms "stabilator" and "horizontal tail" are used interchangeably in various MDC documents. MDC stability and control reports that describe the aerodynamic model (MDC A7247, vol. 1,2), tend to use the term "horizontal tail". MDC reports describing the flight control laws (MDC A4107, vol. 1 and MDC A7813, vol. 1,2,3), use "stabilator". "Stabilator" will be used in this document. In addition, the symbols δ_H and δ_{HT} are used interchangeably in MDC A7247, vol. 1 to refer to "horizontal tail" deflection.

Table 5.2. Control Deflection Sign Conventions

$\delta_{H_{R,L}}$	= right or left stabilator (horizontal tail), + trailing edge down
$\delta_{A_{R,L}}$	= right or left aileron, + trailing edge down
$\delta_{R_{R,L}}$	= right or left rudder, + trailing edge left
$\delta_{n_{R,L}}$	= right or left leading-edge flaps, + leading edge down
$\delta_{f_{R,L}}$	= right or left trailing-edge flaps, + trailing edge down
δ_H	$= \frac{1}{2}\left(\delta_{H_R} + \delta_{H_L}\right)$, symmetric (average) stabilator
δ_f	$= \frac{1}{2}\left(\delta_{f_R} + \delta_{f_R}\right)$, symmetric trailing-edge flap
δ_n	$= \frac{1}{2}\left(\delta_{n_R} + \delta_{n_L}\right)$, symmetric leading-edge flap
$\Delta\delta_H$	$= \delta_{H_L} - \delta_{H_R}$, differential stabilators (horizontal tails)
$\Delta\delta_f$	$= \delta_{f_L} - \delta_{f_R}$, differential trailing-edge flaps
$\Delta\delta_n$	$= \delta_{n_R} - \delta_{n_L}$, differential leading-edge flaps
δ_{RT}	$= \frac{1}{2}\left\{\delta_{R_R} - \delta_{R_L}\right\}$, symmetric rudder (both t.e.in)
δ_R	$= \frac{1}{2}\left\{\delta_{R_R} + \delta_{R_L}\right\}$, antisymmetric rudder (both t.e.left)
$\delta_{f/A}$	$= \frac{1}{4}\left\{\delta_{f_R} + \delta_{A_R} + \delta_{f_L} + \delta_{A_L}\right\}$, combined t.e. control

5.3 Lift and Pitch (L,m)

The reconstruction equations that are used to compute the total lift and pitch coefficients are almost identical and are combined in a common form in equation (5.1). To the basic lift and pitch reconstruction equations (MDC A7247, vol. 1, section 3.1.1) the following have been added; (1) landing gear effects from the MDC batch simulation code (lines 805-906 of subroutine ARO18CN), and (2), rudder toe-in effects extracted from section 4.1.10 of MDC A7247, vol. 1. Replacing "X" in the equation below with "L" or "m" as appropriate will give the lift and pitch equations, respectively. The total lift/pitch coefficient is given by,

$$C_X = C_{X_{sf}} + C_{X_{\dot{\alpha}}}\left(\frac{\dot{\alpha}\bar{c}}{2V}\right) \tag{5.1}$$

where the steady flow lift/pitch coefficient is given by,

$$C_{X_{sf}} = C_{X_o} + \Delta C^F_{X_o} + \frac{1}{2} R^F_{C_{X\delta_H}} \left\{ \Delta C_{X\delta_{H_L}} + \Delta C_{X\delta_{H_R}} \right\}$$

$$+ R^{FLT}_{C_{L\delta_A}} \left\{ \Delta C_{X\delta_{A_L}} + \Delta C_{X\delta_{A_R}} \right\}$$

$$+ C_{X_{\delta_n}} \delta_{n_1} + \Delta C^F_{X\delta_n} \delta_n + \left\{ C_{X\delta_f} + \Delta C^F_{X\delta_f} \right\} \delta_f$$

$$+ \Delta C_{X\delta_{SB}} + \Delta C_{X\delta_{LG}} \delta_{LG} + \Delta C_{X\delta_R} + \Delta C_{X\delta_{RT}}$$

$$+ \Delta C_{X_{stores}} + \left\{ C_{X_q} + \Delta C^F_{X_q} \right\} \frac{q\bar{c}}{2V} \qquad (5.2)$$

while the unsteady component of lift/pitch, i.e., proportional to nondimensional rate of change of angle of attack is given by,

$$C_{X_{\dot{\alpha}}} = C_{X_{\dot{\alpha}_o}} + \Delta C^F_{X_{\dot{\alpha}}} \qquad (5.3)$$

The individual components of the lift and pitch reconstruction equations, (5.2,5.3), are defined in sections (5.3.1) through (5.3.11). These components are for the most part calculated by performing linear interpolations according to tabulated functions. The variables used as input to the interpolation functions are normally limited to prevent extrapolation. Thus once the actual state or control variable reaches the limit of the data, no further change occurs in the coefficient. The input variables and the limits (if applied) are shown for each function look-up.

5.3.1 Basic Airframe Coefficient (L,m)

The basic airframe lift/pitch coefficient for the rigid airplane with control surfaces undeflected is given by,

$$C_{X_o} = \begin{cases} F_{CXO}(\alpha[^{40}_{-12}, M[^{2.}_{.2}) & \alpha \leq 40°;\ \delta_n = 0° \\ F_{HCXO}(\alpha[^{90}_{40}, M[^{1.1}_{.2}) & \alpha > 40°;\ \delta_n = 34° \end{cases} \qquad (5.4)$$

where the notation is defined as follows,

$$F_{CXO}(\alpha[^{40}_{-12}, M[^{2.}_{.2}) \triangleq \left\{ \begin{array}{l} \text{the function } F_{CXO} \text{ as defined by data tables} \\ \text{is evaluated at an angle of attack and Mach number} \\ \text{that are limited according to the brackets.} \end{array} \right\}$$

At low angle of attack ($\alpha \leq 40°$), C_{X_o} refers to a zero stabilator, retracted flaps configuration. At high angle of attack ($\alpha > 40°$), basic lift/pitch, C_{X_o}, refers to a configuration with zero stabilator

and a leading-edge flap deflection of 34°. The elastic increment to the basic airframe lift/pitch coefficient is:

$$\Delta C^{F}_{X_o} = \begin{cases} \left(\frac{M - .2}{.4}\right)\left[{}^{1}_{0}\, F_{DCXOF}(\alpha\left[{}^{32}_{-8}, M\left[{}^{2.}_{.6}, H_\rho\left[{}^{60000}_{0}\right) & LQSE=.T. \\ 0.0 & LQSE=.F. \end{cases} \right. \tag{5.5}$$

In figures 3.1.1-1 through 3.1.1-22 of MDC A7247, vol. 1, the coefficients C_{X_o} and $\Delta C^{F}_{X_o}$ are defined and combined for presentation into one coefficient.

5.3.2 Stabilator Effects (L,m)

Increments due to left and right stabilator (horizontal tail) deflections are,

$$\Delta C_{X_{\delta_{H_{L,R}}}} = \begin{cases} F_{DCXDH}\left(\alpha\left[{}^{40}_{-12}, \delta_{H_{L,R}}\left[{}^{10.5}_{-24}, M\left[{}^{2.}_{.2}\right) & \alpha \le 40 \\ F_{HDCXDH}\left(\alpha\left[{}^{90}_{40}, \delta_{H_{L,R}}\left[{}^{10.5}_{-24}, M\left[{}^{1.1}_{.2}\right) & \alpha > 40 \end{cases} \right. \tag{5.6}$$

The elastic/rigid stabilator effectiveness factor is,

$$R^{F}_{C_{X_{\delta_H}}} = \begin{cases} F_{FRCXDH}\left(M\left[{}^{2.}_{.2}, H_\rho\left[{}^{60000}_{0}\right) & LQSE=.T. \\ 1.0 & LQSE=.F. \end{cases} \right. \tag{5.7}$$

5.3.3 Aileron Effects (L,m)

The lift/pitch increments due to left and right aileron deflections are,

$$\Delta C_{X_{\delta_{A_{L,R}}}} = \begin{cases} F_{DCXDA}\left(\alpha\left[{}^{40}_{-4}, \delta_{A_{L/R}}\left[{}^{25}_{-25}, M\left[{}^{2.}_{.2}\right) & \alpha \le 40° \\ F_{HDCXDA}\left(\alpha\left[{}^{90}_{40}, \delta_{A_{L,R}}\left[{}^{25}_{-25}, M\left[{}^{1.2}_{.2}\right) & \alpha > 40° \end{cases} \right. \tag{5.8}$$

The elastic/rigid aileron effectiveness factor has been replaced by a flight-test derived quantity, a "flight/rigid" factor. This factor is larger than unity in the "rigid" limit because it reflects the configuration change that occurred between 1979 and 1984 in which the aileron span was increased outboard to the wingtip, (MDC A8575, page 1-4). The data in F_{DCXDA} and F_{HDCXDA} are for the small aileron (MDC A7247, vol. 1, page 3-357). Note that $R^{FLT}_{C_{L_{\delta_A}}}$ is used in both the lift ($C_{L_{sf}}$) and pitch ($C_{m_{sf}}$) equations.

$$R^{FLT}_{C_{L_{\delta_A}}} = \begin{cases} F_{FFCLDA}\left(M\left[{}^{2.}_{.2}, H_\rho\left[{}^{60000}_{0}\right) & LQSE=.T. \\ 1.25 & LQSE=.F. \end{cases} \right. \tag{5.9}$$

5.3.4 Leading-Edge Flap Effects (L,m)

The assumption in MDC A7247, vol. 1,2 and the MDC batch simulation code is that the leading-edge flaps are commanded according to the maneuvering flap schedule and will remain fixed at 34° if angle of attack is above 40°. If angle of attack is above 40°, C_{X_o}, defined in equation (5.4), includes the effect of the nonzero leading-edge flaps. Accordingly, $C_{X_{\delta_n}}$, the lift/pitch per degree of leading-edge flap, is not defined in MDC A7247, vol. 2 and is set to zero in the MDC batch simulation code if $\alpha > 40°$. As a result, when the simulated airplane transitions from the low angle-of-attack database ($\alpha \leq 40°$) to the high angle-of-attack database ($\alpha > 40°$), a potential discontinuity can arise. If the leading-edge flaps are at 34° when the simulation goes through α=40°, no discontinuity in C_{X_o} occurs. However, if the leading-edge flaps are not on schedule, then a discontinuity in C_{X_o} exists. To remove this potential discontinuity in the *f18bas* simulation, $C_{X_{\delta_n}}$ is not set to zero abruptly at α=40°, but rather is ramped to zero as angle of attack goes from 40° to 45°. Furthermore, when angle-of-attack is above 40°, $C_{X_{\delta_n}}$ is multiplied by the difference between leading-edge flaps and 34°, the position at which the high angle-of-attack C_{X_o} data was taken, instead of the total δ_n. Thus,

$$C_{X_{\delta_n}} = \begin{cases} F_{CXDN}(\alpha|_{-4}^{40}, M|_{.2}^{1.2}) & \alpha \leq 40 \\ \left(\frac{45-\alpha}{5}\right)\Big|_0^1 F_{CXDN}(\alpha=40°, M|_{.2}^{1.2}) & \alpha > 40;\ \text{LRTE=.F.} \\ 0.0 & \alpha > 40;\ \text{LRTE=.T.} \end{cases} \tag{5.10}$$

where

$$\delta_{n_1} = \begin{cases} \delta_n & \alpha \leq 40 \\ \delta_n - 34 & \alpha > 40 \end{cases} \tag{5.11}$$

The elastic increments to the leading-edge flap sensitivity for lift and pitch are,

$$\Delta C^F_{L_{\delta_n}} = \begin{cases} \left(\frac{M-.6}{.2}\right)\Big|_0^1 F_{DCLDNF}(\alpha|_{-4}^{32}, M|_{.8}^{1.1}, H_\rho|_0^{60000}) & \text{LQSE=.T.} \\ 0.0 & \text{LQSE=.F.} \end{cases} \tag{5.12}$$

and

$$\Delta C^F_{m_{\delta_n}} = \begin{cases} \left(\frac{M-.2}{.4}\right)\Big|_0^1 F_{DCMDNF}(\alpha|_{-4}^{32}, M|_{.6}^{1.1}, H_\rho|_0^{60000}) & \text{LQSE=.T.} \\ 0.0 & \text{LQSE=.F.} \end{cases} \tag{5.13}$$

The factor $\left(\frac{M-.6}{.2}\right)\Big|_0^1$ serves to ramp in the full value of $\Delta C^F_{L_{\delta_n}}$ as Mach goes from 0.6 to 0.8..

Similarly, $\left(\frac{M-.2}{.4}\right)\Big|_0^1$ ramps in the full value of $\Delta C^F_{m_{\delta_n}}$ by going from zero to one as Mach goes from 0.2 to 0.6.

5.3.5 Trailing-Edge Flap Effects (L,m)

The lift/pitch per degree deflection of symmetric trailing-edge flaps is,

$$C_{X_{\delta_f}} = \begin{cases} F_{CXDF}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}\right], M\left[\begin{smallmatrix}1.1\\.2\end{smallmatrix}\right]\right) & \alpha \le 40° \\ 0.0 & \alpha > 40° \end{cases} \tag{5.14}$$

The elastic increment to the trailing-edge flap sensitivity is,

$$\Delta C^F_{X_{\delta_f}} = \begin{cases} \left(\frac{M-.2}{.4}\right)\Big|_0^1 F_{DCXDFF}\left(\alpha\left[\begin{smallmatrix}32\\-4\end{smallmatrix}\right], M\left[\begin{smallmatrix}.9\\.6\end{smallmatrix}\right], H_\rho\left[\begin{smallmatrix}60000\\0\end{smallmatrix}\right]\right) & \alpha \le 40;\ \text{LQSE=.T.} \\ 0.0 & \text{otherwise} \end{cases} \tag{5.15}$$

5.3.6 Speed Brake Effects (L,m)

The increments due to speed brake deflection are slightly different for lift and pitch. The lift increment is to be multiplied by normalized speed brake deflection, $\frac{\delta_{SB}}{60}$. For the low angle-of-attack case, the pitch increment is itself a function of speed brake and no multiplier is needed.

$$\Delta C_{L_{SB}} = \begin{cases} \left(\frac{\delta_{SB}}{60}\right) F_{DCLSB}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}\right], M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right]\right) & \alpha \le 40° \\ \left(\frac{\delta_{SB}}{60}\right) F_{HDCLSB}\left(\alpha\left[\begin{smallmatrix}90\\40\end{smallmatrix}\right], M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right]\right) & \alpha > 40° \end{cases} \tag{5.16}$$

$$\Delta C_{m_{SB}} = \begin{cases} F_{DCMSB}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}\right], M\left[\begin{smallmatrix}1.6\\.2\end{smallmatrix}\right], \delta_{SB}\left[\begin{smallmatrix}60\\0\end{smallmatrix}\right]\right) & \alpha \le 40° \\ \left(\frac{\delta_{SB}}{60}\right) F_{HDCMSB}\left(\alpha\left[\begin{smallmatrix}90\\40\end{smallmatrix}\right], M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right]\right) & \alpha > 40° \end{cases} \tag{5.17}$$

5.3.7 *Stores (L,m)*

The increment due to stores is discussed in MDC A7247, vol. 1, section 3.1.8, for the cases of Escort, Escort Overload, and Interdiction mission loadings. The quantity is currently set to zero in the *f18bas* simulation described in this report.

$$\Delta C_{X_{stores}} = 0.0 \tag{5.18}$$

5.3.8 *Landing Gear Effects (L,m)*

Increments to lift, pitch, and drag due to landing gear deployment are discussed on pages 4-75 and 4-76 of reference MDC A7247, vol. 1. In MDC A7247, vol. 1, landing gear increments are included only for the high-lift (take-off and approach) configurations. In the MDC batch simulation, landing gear increments are included in the lift, pitch, and drag build-up for the maneuvering ("clean") configuration. The increments used in the MDC batch simulation for the maneuvering configuration are different than those shown on 4-76 of MDC A7247, vol. 1. The increments used in the *f18bas* simulation described in this report are those of MDC batch simulation. These increments are to be multiplied by normalized landing gear deployment. While studies to date have had no requirement for the aerodynamic effects of the landing gear being included, there is virtually no extra computational cost as the code calculating the coefficients is bypassed if the gear is retracted.

$$\Delta C_{X_{LG}} = \begin{cases} F_{CXLG}\left(\alpha\left[\begin{smallmatrix}30\\-2\end{smallmatrix}\right.\right) & \alpha \le 40^\circ \\ 0.0 & \alpha > 40^\circ \end{cases} \tag{5.19}$$

5.3.9 *Rudder Effects (L,m)*

The increment to lift due to antisymmetric rudder is included in the high lift configurations described in section 4.1.10 of MDC A7247, vol. 1, but not in the maneuvering configuration. The lift increment is set to zero in the *f18bas* simulation described in this report. A nonzero pitch increment is included in the *f18bas* simulation,

$$\Delta C_{L_{\delta_R}} = 0.0 \tag{5.20}$$

$$\Delta C_{m_{\delta_R}} = \begin{cases} F_{DCMDR}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}\right., |\delta_R|\left[\begin{smallmatrix}30\\0\end{smallmatrix}\right., M\left[\begin{smallmatrix}2.\\.2\end{smallmatrix}\right.\right) & \alpha \le 40^\circ \\ F_{HDCMDR}\left(\alpha\left[\begin{smallmatrix}90\\40\end{smallmatrix}\right., |\delta_R|\left[\begin{smallmatrix}30\\0\end{smallmatrix}\right., M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right.\right) & \alpha > 40^\circ \end{cases} \tag{5.21}$$

Symmetric rudder is used in approach and landing, but is not part of the nominal maneuvering configuration model. While the pitch channel of the flight control system makes no use of symmetric rudder in the auto-flap-up (AFU), maneuvering mode, future agility-oriented research control laws may employ such a command. Therefore, increments in lift/pitch due to symmetric rudder (toe-in) have been added to the maneuvering configuration by using the increment developed for the high-lift configurations and discussed in section 4.1.10 of MDC A7247, vol. 1. The increments due to symmetric rudder for the *f18bas* simulation are,

$$\Delta C_{X_{\delta_{RT}}} = \begin{cases} F_{DCXDRT}\left(\alpha\left[\begin{smallmatrix}40\\-10\end{smallmatrix}\right., \delta_{RT}\left[\begin{smallmatrix}30\\-15\end{smallmatrix}\right., \delta_{f/A}\left[\begin{smallmatrix}45\\0\end{smallmatrix}\right.\right) & \alpha \leq 40° \\ 0.0 & \alpha > 40° \end{cases} \tag{5.22}$$

Since the rudder toe-in coefficient was developed for the low speed/high lift configurations, there is no associated flex/rigid increment or effectiveness factor (page 4-5 of MDC A7247, vol. 1).

5.3.10 Pitch Rate (L,m)

The pitch rate derivative for lift/pitch is,

$$C_{X_q} = \begin{cases} F_{CXQ}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}\right., M\left[\begin{smallmatrix}2.\\.2\end{smallmatrix}\right.\right) & \alpha \leq 40° \\ F_{HCXQ}\left(\alpha\left[\begin{smallmatrix}90\\40\end{smallmatrix}\right., M\left[\begin{smallmatrix}1.1\\.2\end{smallmatrix}\right.\right) & \alpha > 40° \end{cases} \tag{5.23}$$

The elastic increment to the pitch derivative for lift/pitch is,

$$\Delta C^F_{X_q} = \begin{cases} F_{DCXQF}\left(M\left[\begin{smallmatrix}2.\\.2\end{smallmatrix}\right., H_\rho\left[\begin{smallmatrix}60000\\0\end{smallmatrix}\right.\right) & \text{LQSE=.T.} \\ 0.0 & \text{LQSE=.F.} \end{cases} \tag{5.24}$$

5.3.11 Angle-of-Attack Rate (L,m)

The angle-of-attack rate derivative for lift/pitch for the rigid airplane is,

$$C_{X_{\dot{\alpha}_o}} = \begin{cases} F_{CXAD}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}\right., M\left[\begin{smallmatrix}2.\\.2\end{smallmatrix}\right.\right) & \alpha \leq 40° \\ F_{HCXAD}\left(\alpha\left[\begin{smallmatrix}90\\40\end{smallmatrix}\right., M\left[\begin{smallmatrix}1.1\\.2\end{smallmatrix}\right.\right) & \alpha > 40° \end{cases} \tag{5.24}$$

while the elastic increment to the angle-of-attack rate derivative for lift/pitch is,

$$\Delta C^F_{X_{\dot{\alpha}}} = \begin{cases} \left(\frac{11-\alpha}{3}\right]^1_0 F_{DCXADF}\left(M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right., H_\rho\left[\begin{smallmatrix}60000\\0\end{smallmatrix}\right.\right) & \text{LQSE=.T.} \\ 0.0 & \text{LQSE=.F.} \end{cases} \tag{5.25}$$

5.4 Drag

Drag is not discussed in MDC A7247, vol. 1 for the maneuvering configuration. The total drag coefficient equation is taken from the MDC batch simulation and is repeated below,

$$C_{D_{sf}} = C_{D_o} + \Delta C_{D_{C_L}} + \frac{1}{2}\left\{\Delta C_{D\delta_{H_L}} + \Delta C_{D\delta_{H_R}}\right\} + C_{D\delta_n}\delta_{n_1} + C_{D\delta_f}\delta_f$$

$$+ \Delta C_{D\delta_{SB}} + \Delta C_{D\delta_{LG}}\delta_{LG} + \Delta C_{D\delta_{RT}} \qquad (5.26)$$

5.4.1 Basic Drag Coefficient

The basic drag coefficient is given by,

$$C_{D_o} = \begin{cases} F_{CDO}\left(\alpha\left[{}^{40}_{-12}, M\left[{}^{2.}_{.2}\right)\right. & \alpha \le 40° \\ F_{HCDO}\left(\alpha\left[{}^{90}_{40}, M\left[{}^{1.1}_{.2}\right)\right. & \alpha > 40° \end{cases} \qquad (5.27)$$

When in the low-angle-of-attack regime, an additional drag increment due to lift is calculated. However, C_{D_o} does not represent zero-lift drag. C_{D_o} has the characteristic parabolic shape of the standard drag polar and already includes the basic effects of lift on drag. An additional drag increment due to lift is defined in the MDC batch simulation as

$$\Delta C_{D_{C_L}} = \begin{cases} F_{DCDCL}\left(C_L\left[{}^{2}_{-.5}, M\left[{}^{1.8}_{.2}\right)\right. & \alpha \le 40° \\ 0.0 & \alpha > 40° \end{cases} \qquad (5.28)$$

$\Delta C_{D_{C_L}}$ represents small corrections to C_{D_o} and can be positive or negative. In the MDC batch simulation comments, $\Delta C_{D_{C_L}}$ is described as a correction "from tunnel drag to full-scale" and is presumably the result of flight data. Due to the explicit formulation of the equations of motion (section 4.3 of this document), C_L is not readily available in the *f18bas* simulation. Therefore, in the *f18bas* simulation implementation of (5.28), $\Delta C_{D_{C_L}}$ is a function of $C_{L_{sf}}$ instead of C_L. The difference in aircraft response induced by using $C_{L_{sf}}$ instead of C_L in equation (5.28) is negligible.

5.4.2 Stabilator Drag Increment

The drag increments due to left and right stabilator (horizontal tail) deflection are given by:

$$\Delta C_{D_{\delta_{H_{L,R}}}} = \begin{cases} F_{DCDDH}\left(\alpha\left[{}^{40}_{-12}, \delta_{H_{L,R}}\left[{}^{10.5}_{-24}, M\left[{}^{2.}_{.2}\right)\right.\right. & \alpha \le 40° \\ F_{HDCDDH}\left(\alpha\left[{}^{90}_{40}, \delta_{H_{L,R}}\left[{}^{10.5}_{-24}, M\left[{}^{1.1}_{.2}\right)\right.\right. & \alpha > 40° \end{cases} \qquad (5.29)$$

5.4.3 *Leading-Edge Flap Drag Derivative*

The drag per degree of symmetric leading-edge flap is:

$$C_{D_{\delta_n}} = \begin{cases} F_{CDDN}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}, M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right.\right) & \alpha \le 40° \\ \left(\left(\frac{45-\alpha}{5}\right)\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right.\right) F_{CXDN}\left(\alpha=40, M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right.\right) & \alpha > 40° \end{cases} \tag{5.30}$$

5.4.4 *Trailing-Edge Flap Drag Derivative*

The drag per degree of symmetric trailing-edge flap is,

$$C_{D_{\delta_f}} = \begin{cases} F_{CDDF}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}, M\left[\begin{smallmatrix}1.1\\.2\end{smallmatrix}\right.\right) & \alpha \le 40° \\ 0.0 & \alpha > 40° \end{cases} \tag{5.31}$$

5.4.5 *Speed Brake Drag Derivative*

The drag increment due to full speed brake deflection is,

$$\Delta C_{D_{SB}} = \begin{cases} \left(\frac{\delta_{SB}}{60}\right) F_{DCDSB}\left(\alpha\left[\begin{smallmatrix}40\\-4\end{smallmatrix}, M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right.\right) & \alpha \le 40° \\ \left(\frac{\delta_{SB}}{60}\right) F_{HDCDSB}\left(\alpha\left[\begin{smallmatrix}90\\40\end{smallmatrix}, M\left[\begin{smallmatrix}1.2\\.2\end{smallmatrix}\right.\right) & \alpha > 40° \end{cases} \tag{5.32}$$

5.4.6 *Landing Gear Drag Increment*

The drag increment due to full landing gear deployment is,

$$\Delta C_{D_{LG}} = \begin{cases} F_{DCDLG}\left(\alpha\left[\begin{smallmatrix}30\\-2\end{smallmatrix}\right.\right) & \alpha \le 40° \\ 0.0 & \alpha > 40° \end{cases} \tag{5.33}$$

5.4.7 *Rudder Drag Increment*

As discussed in the lift/pitch term build-up, a drag increment due to symmetric rudder (toe-in) has been added to the maneuvering configuration by using the increment developed for the high-lift configurations and discussed in section 4.1.10 of MDC A7247, vol. 1. Recall that the low speed/high lift configurations are considered low-angle-of-attack configurations. As discussed in section 4.1.2.9, no elastic/rigid adjustments are defined for the rudder toe-in coefficients in MDC A7247, vol. 1.

$$\Delta C_{D_{\delta_{RT}}} = \begin{cases} F_{CDDRT}(\alpha|_{-10}^{40}, \delta_{RT}|_{-15}^{30}, \delta_{f/A}|_{0}^{45}) & \alpha \leq 40° \\ 0.0 & \alpha > 40° \end{cases} \quad (5.33)$$

5.5 Side-Force, Roll, Yaw (Y,l,n)

As in the case of lift/pitch, the same reconstruction equation can be used for side force, roll and yaw. The appropriate equation is recovered by substituting "Y" (side force), "l" (roll) , or "n" (yaw) for "X" in equation (5.34) below.

$$C_{X_{sf}} = C_{X_o} + \Delta C^F_{X_\beta}\beta + \Delta C_{X_{\delta_n}}\left(\frac{\delta_n}{25}\right) + \Delta C_{X_{\delta_f}}\left(\frac{\delta_f}{20}\right) + C_{X_{\Delta\delta_n}}\Delta\delta_n$$

$$+ C_{X_{\Delta\delta_f}}\Delta\delta_f + R^F_{C_{X_{\delta_A}}}\left\{\Delta C_{X_{\delta_{A_L}}} + \Delta C_{X_{\delta_{A_R}}}\right\}$$

$$+ \Delta C_{X_{\delta_R}} R_{C_{X_{\delta_H}}} R^F_{C_{X_{\delta_R}}} + R^F_{C_{X_{\Delta\delta_H}}} C_{X_{\Delta\delta_H}}\Delta\delta_H + \beta\,\Delta C_{X_{\delta_{SB}}}$$

$$+ \Delta C_{X_{asm}} + C_{X_p} R^F_{C_{X_p}}\left(\frac{p\bar{c}}{2V}\right) + \left(C_{X_r} + \Delta C^F_{X_r}\right)\left(\frac{r\bar{c}}{2V}\right) \quad (5.34)$$

5.5.1 Basic Airframe Coefficient (Y,l,n)

The basic airframe coefficient for the rigid airplane with controls at zero is given by,

$$C_{X_o} = \begin{cases} k\,C_{X_{o_1}} + (1-k)\,C_{X_{o_2}} & \alpha \leq 40° \\ F_{HCXO}(\beta, \alpha|_{40}^{90}, M|_{.2}^{.9}) & \alpha > 40° \end{cases} \quad (5.35)$$

where,

$$C_{X_{o_1}} = F_{CXO1}(\beta, \alpha|_{-10}^{40}, M|_{.2}^{1.1}) \quad (5.36.a)$$

$$C_{X_{o_2}} = F_{CXO2}(\beta, \alpha|_{-10}^{20}, M|_{1.2}^{2.}) \quad (5.36.b)$$

$$\text{and } k = \frac{M-1.1}{.1}\Big|_0^1 \quad (5.36.c)$$

5.5.2 Leading-Edge Flap Effects (Y,l,n)

If sideslip is nonzero, then symmetric leading-edge flap deployment produces a change in the lateral/directional coefficients. The increment due to full 25 degree deployment of symmetric leading-edge flaps in the presence of nonzero sideslip is given by,

$$\Delta C_{X_{\delta_n}} = \begin{cases} \frac{\beta}{|\beta|} F_{DCXDN}\left(|\beta|\Big|_0^{20}, \alpha\Big|_0^{40}, M\Big|_{.2}^{1.1} \right) & \alpha \leq 40° \\ 0.0 & \alpha > 40° \end{cases} \tag{5.37}$$

where $\Delta C_{X_{\delta_n}}$ is to be multiplied by a normalized leading-edge flap position in the appropriate reconstruction equation. The change in total coefficient per degree of differential leading-edge flap is given by:

$$C_{Y_{\Delta\delta_n}} = 0.0 \tag{5.38.a}$$

$$C_{l_{\Delta\delta_n}} = \begin{cases} F_{C1FDDN}\left(\bar{q}_1, M\Big|_{.2}^{2.} \right) & \alpha \leq 40° \\ 0.0 & \alpha > 40° \end{cases} \tag{5.38.b}$$

$$C_{n_{\Delta\delta_n}} = 0.0 \tag{5.38.c}$$

where:

$$\bar{q}_1 = \begin{cases} \bar{q} & \text{LQSE=.T.} \\ 0.0 & \text{LQSE=.F.} \end{cases} \tag{5.39}$$

5.5.3 Trailing-Edge Flap Effects (Y,l,n)

Increments due to full 20 degree deployment of symmetric trailing-edge flaps in the presence of nonzero sideslip are,

$$\Delta C_{X_{\delta_f}} = \begin{cases} \frac{\beta}{|\beta|} k_{\delta_f} F_{DCXDF}\left(|\beta|\Big|_0^{20}, \alpha\Big|_0^{40}, M\Big|_{.2}^{.9} \right) & \alpha \leq 40° \\ 0.0 & \alpha > 40° \end{cases} \tag{5.40}$$

where as in the case of the leading-edge flaps, $\Delta C_{X_{\delta_f}}$ is multiplied by normalized trailing-edge flap position in the appropriate reconstruction equation, and the factor k_{δ_f} is defined as,

$$k_{\delta_f} = \frac{1.1\text{-}M}{.2}\Big|_0^1 \tag{5.41}$$

The change in total coefficient per degree of differential trailing-edge flap is given by,

$$C_{Y_{\Delta\delta_f}} = 0.0 \tag{5.42.a}$$

$$C_{l_{\Delta\delta_f}} = \begin{cases} F_{C1FDDF}(\alpha[^{20}_{-4}, H_{SL}[^{60000.}_{0.}, M[^{2.}_{.2}) & \alpha \le 40°;\ \text{LQSE=.T.} \\ F_{C1FDDF}(\alpha[^{20}_{-4}, H_{SL}=60000, M[^{2.}_{.2}) & \alpha \le 40°;\ \text{LQSE=.F.} \\ 0.0 & \alpha > 40° \end{cases} \quad (5.42.b)$$

$$C_{n_{\Delta\delta_f}} = 0.0 \quad (5.42.c)$$

5.5.4 Aileron Effects (Y,l,n)

The increments due to left and right aileron deflection are given by,

$$\Delta C_{X_{\delta_{A_{L,R}}}} = \begin{cases} k\, F_{DCXDAL}\left(\alpha[^{40}_{-4}, \delta_{A_{L,R}}[^{25}_{-25}, M[^{2.}_{.2}\right) & \alpha \le 40° \\ k\, F_{HDCXDA}\left(\alpha[^{90}_{40}, \delta_{A_{L,R}}[^{25}_{-25}, M[^{.9}_{.2}\right) & \alpha > 40° \end{cases} \quad (5.43)$$

where k = 1 for the left aileron and k = -1 for the right aileron.

The elastic/rigid ratio for aileron effectiveness is,

$$R^F_{C_{Y_{\delta_A}}} = \begin{cases} F_{FRCYDA}\left(M[^{2.}_{.2}, H_\rho[^{60000}_{0}\right) & \text{LQSE=.T.} \\ 1.63 & \text{LQSE=.F.} \end{cases} \quad (5.44.a)$$

$$R^F_{C_{l_{\delta_A}}} = \begin{cases} F_{FRCLDA}\left(M[^{2.}_{.2}, H_\rho[^{60000}_{0}\right) & \text{LQSE=.T.} \\ 1.25 & \text{LQSE=.F.} \end{cases} \quad (5.44.b)$$

$$R^F_{C_{n_{\delta_A}}} = \begin{cases} F_{FRCNDA}\left(M[^{2.}_{.2}, H_\rho[^{60000}_{0}\right) & \text{LQSE=.T.} \\ 1.25 & \text{LQSE=.F.} \end{cases} \quad (5.44.c)$$

The elastic/rigid ratios assume values other than one in the rigid limit for the same reasons as discussed in section 5.3.3 of this report. The tabular look-up wind-tunnel data was taken before a configuration change that increased aileron area.

5.5.5 Rudder Effects (Y,l,n)

Increments due to antisymmetric (both t.e.left) rudder deflection are given by:

$$\Delta C_{X_{\delta_R}} = \begin{cases} \dfrac{\delta_R}{|\delta_R|} F_{DCXDR}\left(\alpha[^{40}_{-4}, |\delta_R|[^{30}_{0}, M[^{2.}_{.2}\right) & \alpha \le 40° \\ F_{HDCXDR}\left(\alpha[^{90}_{40}, \beta[^{10}_{-10}, \delta_R\right) & \alpha > 40° \end{cases} \quad (5.45)$$

The flex/rigid ratio for the effect of antisymmetric rudder is:

$$R^{F}_{C_{X\delta_R}} = \begin{cases} F_{FRCXDR}\left(M\left|{}^{2.}_{.2}, H_\rho\right|{}^{60000}_{0}\right) & LQSE=.T. \\ 1.0 & LQSE=.F. \end{cases} \tag{5.46}$$

An additional effectiveness factor that incorporates the effect of symmetric stabilator on the increment due to antisymmetric rudder at high angle of attack is given by,

$$R_{C_{X\delta_H}} = \begin{cases} F_{HRXDRH}\left(\alpha\left|{}^{90}_{40}, \delta_H\right|{}^{0}_{-12}\right) & \alpha > 40°;\ "X"="n" \\ 1.0 & \text{otherwise} \end{cases} \tag{5.47}$$

5.5.6 Stabilator Effects (Y,l,n)

The derivative with respect to a differential stabilator, which is itself a function of angle of attack, symmetric stabilator, and Mach, is given by,

$$C_{X_{\Delta\delta_H}} = \begin{cases} F_{CXDDH}\left(\alpha\left|{}^{40}_{-4}, \delta_H, M\right|{}^{2.}_{.2}\right) & \alpha \le 40° \\ F_{HCXDDH}\left(\alpha\left|{}^{90}_{40}, \delta_H, M\right|{}^{.9}_{.2}\right) & \alpha > 40° \end{cases} \tag{5.48}$$

The elastic/rigid ratio on the differential tail derivative is defined as,

$$R^{F}_{C_{X_{\Delta\delta_H}}} = \begin{cases} F_{FRXDDH}\left(M\left|{}^{2.}_{.2}, H_\rho\right|{}^{60000}_{0}\right) & LQSE=.T.;\ "X"="Y"or"l" \\ F_{FRXDDH}\left(M\left|{}^{2.}_{.2}, H_\rho\right|{}^{60000}_{0}, \alpha\left|{}^{40}_{0}\right.\right) & LQSE=.T.;\ "X"="n" \\ 1.0 & LQSE=.F. \end{cases} \tag{5.49}$$

5.5.7 Dynamic Derivatives (Y,l,n)

The derivatives due to nondimensionalized roll rate are defined as,

$$C_{X_p} = \begin{cases} F_{CXP}\left(\alpha\left|{}^{40}_{-4}, M\right|{}^{2.}_{.2}, H_\rho\right|{}^{60000}_{0}\right) & \alpha \le 40°;\ "X"="l";\ LQSE=.T. \\ F_{CXP}\left(\alpha\left|{}^{40}_{-4}, M\right|{}^{2.}_{.2}, H_\rho=60000\right) & \alpha \le 40°;\ "X"="l";LQSE=.F. \\ F_{CXP}\left(\alpha\left|{}^{40}_{-4}, M\right|{}^{2.}_{.2}\right) & \alpha \le 40°;\ "X"="Y"or"n" \\ F_{HCXP}\left(\alpha\left|{}^{90}_{40}, M\right|{}^{1.2}_{.2}\right) & \alpha > 40° \end{cases} \tag{5.50}$$

while the effect of nondimensional yaw rate is given by,

$$C_{X_r} = \begin{cases} F_{CXR}\left(\alpha\left[{}^{40}_{-4}, M\left[{}^{2.}_{.2}\right.\right.\right) & \alpha \leq 40° \\ F_{HCXR}\left(\alpha\left[{}^{90}_{40}, M\left[{}^{1.2}_{.2}\right.\right.\right) & \alpha > 40° \end{cases} \tag{5.51}$$

The increment to yaw damping due to elastic deformation is given by,

$$\Delta C^F_{X_r} = \begin{cases} F_{FDCXRF}\left(M\left[{}^{2.}_{.2}, H_\rho\right[{}^{60000}_{0}\right) & LQSE=.T. \\ 0.0 & LQSE=.F. \end{cases} \tag{5.52}$$

A separate flex/rigid ratio on roll damping is calculated for side-force and yawing moment. The flexible effect on roll moment due to roll rate is included in the function used to calculate C_{l_p}.

$$R^F_{C_{Y_p}} = \begin{cases} F_{FRCYP}\left(M\left[{}^{2.}_{.2}, H_\rho\right[{}^{60000}_{0}\right) & LQSE=.T. \\ 0.0 & LQSE=.F. \end{cases} \tag{5.53}$$

$$R^F_{C_{n_p}} = \begin{cases} F_{FRCNP}\left(M\left[{}^{2.}_{.2}, \alpha\right[{}^{40}_{0}\right) & LQSE=.T. \\ F_{FRCNP}\left(M=.2, \alpha\right[{}^{40}_{0}\right) & LQSE=.F. \end{cases} \tag{5.54}$$

$$R^F_{C_{l_p}} = 1.0 \quad \text{all conditions} \tag{5.55}$$

5.5.8 Speed Brake Effects (Y,l,n)

The speed brake increments shown below are due to full, normalized, speed brake deployment and are multiplied by sideslip angle in the reconstruction equations.

$$\Delta C_{X_{SB}} = \begin{cases} \left(\frac{\delta_{SB}}{60}\right) F_{DCXSB}\left(\alpha\left[{}^{40}_{-10}, |\beta|\right[{}^{20}_{0}, M\left[{}^{2.}_{.2}\right.\right.\right) & \alpha \leq 40° \\ \left(\frac{45-\alpha}{5}\right)\left[{}^{1}_{0}\right. \left(\frac{\delta_{SB}}{60}\right) F_{HDCXSB}\left(|\beta|\left[{}^{10}_{0}, M\left[{}^{.9}_{.2}\right.\right.\right) & \alpha > 40° \end{cases} \tag{5.56}$$

5.5.9 Effect of Airplane Asymmetries (Y,l,n)

A rolling moment is induced at zero sideslip and symmetric controls for angle of attack between 45° and 60°. This rolling moment is attributed (section 4.1.2 of [MDC A7247, vol. 2]) to asymmetries in the nose. The asymmetric increment is given by,

$$\Delta C_{X_{asm}} = \begin{cases} 0.0 & \alpha \le 40^\circ \\ F_{HC1ASM}\left(\alpha \begin{bmatrix} 90 \\ 40 \end{bmatrix}\right) & \alpha > 40^\circ;\ \text{"X" = "1";LRTE=.F.} \\ 0.0 & \alpha > 40^\circ;\ \text{"X" = "1";LRTE=.T.} \\ 0.0 & \alpha > 40^\circ;\ \text{"X" = "Y"or"n"} \end{cases} \tag{5.57}$$

5.6 Force and Moment Calculations

In this section we describe how the total aerodynamic forces and moments at the c.g. are calculated. The aerodynamic model has been worked up about an aerodynamic reference center. While it is common practice to transform the *outputs* of the aerodynamic model (force and moment coefficients) from the aerodynamic reference center to the c.g., it is less common to transform the *inputs* to the aerodynamic model. Both transformations are performed in the *f18bas* simulation in the interest of completeness. The practice of transforming the inputs has been seen by the authors in a simulation model for the X-29 and in section II of Gainer and Hoffman (1972).

First, the velocity of the air is subtracted from the inertial translational velocity of the airplane to get relative velocity (with respect to the air mass) at the c.g. The body frame components of velocity with respect to the air mass are given by,

$$\begin{Bmatrix} u_a \\ v_a \\ w_a \end{Bmatrix} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} - \begin{Bmatrix} u_{wg} \\ v_{wg} \\ w_{wg} \end{Bmatrix} \tag{5.58}$$

where $[u_{wg}\ v_{wg}\ w_{wg}]'$ is the velocity of the air due to winds, turbulence, and gusts.

The velocity components relative to the air mass at the a.r.c. are given by,

$$\begin{Bmatrix} u_{a_{rf}} \\ v_{a_{rf}} \\ w_{a_{rf}} \end{Bmatrix} = \begin{Bmatrix} u_a \\ v_a \\ w_a \end{Bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} x_{arf} \\ y_{arf} \\ z_{arf} \end{Bmatrix} \tag{5.59}$$

where $[x_{arf}\ y_{arf}\ z_{arf}]'$ locates the a.r.c. in the c.g.-centered body frame. Rotational velocities at the c.g. can induce small changes in translational velocities at the a.r.c. Incidence angles (α_{rf} and β_{rf}), airspeed, and Mach number, are then computed from relative velocities at the a.r.c.

$$\left.\begin{aligned} \alpha_{rf} &= \arctan\left(w_{a_{rf}} / u_{a_{rf}} \right) \\ \beta_{rf} &= \arcsin\left(v_{a_{rf}} / V_{rf} \right) \\ V_{rf} &= \sqrt{u_{a_{rf}}^2 + v_{a_{rf}}^2 + w_{a_{rf}}^2} \\ M_{rf} &= V_{rf} / a \end{aligned}\right\} \tag{5.60}$$

Dynamic pressure ($\bar{q}$) is left referenced to the c.g. The dynamic pressure in lbs/ft^2 is given by,

$$\bar{q} = \frac{1}{2} \rho V_{rf}^2 \tag{5.61}$$

where ρ is the density in slugs/ft^3. A density altitude, H_ρ, is calculated from the air density, ρ. H_ρ is the altitude in feet that is appropriate for ρ in the 1962 Standard Atmosphere tables. Air density can be varied as an independent variable if required (e.g., hot day conditions).

5.6.1 Steady Flow Outputs

The quantities defined in equations (5.60) and (5.61) are then combined with the body frame rotational rates (p,q,r), pressure altitude, the control surface deflections ($\{\delta\}$), and the reference dimensions ($\bar{c}$,b), to serve as inputs to the aerodynamic model described in sections 5.2 through 5.5. Outputs of the aerodynamic model include the force and moment coefficients for steady flow conditions. The process is characterized by equation 5.62.

$$C_{[]_{sf}} = f_{aero}(\alpha_{rf}, \beta_{rf}, V_{rf}, M_{rf}, \bar{q}, H_\rho, \{\delta\}, p, q, r, \bar{c}, b) \tag{5.62}$$

where "[]" = "L", "D", "Y", "l", "m", or "n". The lift and drag coefficients are converted to body axis components via,

$$\left.\begin{aligned} C_{X_{sf}} &= -\cos(\alpha_{rf}) C_{D_{sf}} + \sin(\alpha_{rf}) C_{L_{sf}} \\ C_{Z_{sf}} &= -\sin(\alpha_{rf}) C_{D_{sf}} - \cos(\alpha_{rf}) C_{L_{sf}} \end{aligned}\right\} \tag{5.63}$$

The steady flow aerodynamic forces in the body frame are given by,

$$\{F\}_A\Big|_{\{\dot{V}\}=0} = \begin{Bmatrix} F_{X_{sf}} \\ F_{Y_{sf}} \\ F_{Z_{sf}} \end{Bmatrix}_A = \bar{q}S \begin{Bmatrix} C_{X_{sf}} \\ C_{Y_{sf}} \\ C_{Z_{sf}} \end{Bmatrix} \tag{5.64}$$

The steady flow aerodynamic moments in the body frame are given by,

$$\{M\}_A\Big|_{\{\dot{V}\}=0} = \begin{Bmatrix} M_{X_{sf}} \\ M_{Y_{sf}} \\ M_{Z_{sf}} \end{Bmatrix}_A = \bar{q}S \begin{Bmatrix} b\, C_{l_{sf}} \\ \bar{c}\, C_{m_{sf}} \\ b\, C_{n_{sf}} \end{Bmatrix} + \begin{bmatrix} 0 & -z_{arf} & y_{arf} \\ z_{arf} & 0 & -x_{arf} \\ -y_{arf} & x_{arf} & 0 \end{bmatrix} \begin{Bmatrix} F_{X_{sf}} \\ F_{Y_{sf}} \\ F_{Z_{sf}} \end{Bmatrix}_A \tag{5.65}$$

As a simple sanity check, observe that a positive x_{arf} implies the a.r.c. is ahead of the c.g., and a positive $F_{Z_{sf}}$ force applied at the a.r.c. induces a negative pitch moment, M_Y.

5.6.1 Unsteady Flow Outputs

We seek now to define the 6×6 apparent mass matrix defined in equation (4.13) (repeated below) that arises from the $C_{L_{\dot\alpha}}$ and $C_{m_{\dot\alpha}}$ stability derivatives.

$$[M]_A = -\frac{\partial \{F\}_A}{\partial \{\dot V\}} \tag{4-13}$$

First, in a process similar to equation (5.63), transform the $\dot\alpha$ stability derivatives from stability to body frame components at the a.r.c.,

$$\begin{aligned} C_{X_{\dot\alpha}} &= -\cos(\alpha_{rf})C_{D_{\dot\alpha}} + \sin(\alpha_{rf})C_{L_{\dot\alpha}} \\ C_{Z_{\dot\alpha}} &= -\sin(\alpha_{rf})C_{D_{\dot\alpha}} - \cos(\alpha_{rf})C_{L_{\dot\alpha}} \end{aligned} \tag{5.66}$$

In the *f18bas* simulation, as is typically the case, $C_{D_{\dot\alpha}}$ is zero. Define some intermediate terms that have units of mass and mass-length.

$$m_0 = \bar q\, S \left(\frac{\bar c}{2V_{rf}}\right)\left(\frac{u_{a_{rf}}}{u_{a_{rf}}^2 + w_{a_{rf}}^2}\right), \text{slugs} \tag{5.67}$$

$$\left.\begin{aligned} m_u &= m_0\, C_{X_{\dot\alpha}} && \text{slugs} \\ m_w &= m_0\, C_{Z_{\dot\alpha}} && \text{slugs} \\ m_q &= \bar c\, m_0\, C_{m_{\dot\alpha}} && \text{slug-ft} \end{aligned}\right\} \tag{5.68}$$

For notational convenience, let V and ω denote the 3×1 column vectors associated with translational and rotational velocity, respectively,

$$V = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}, \quad \omega = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{5.69}$$

The apparent mass matrix is given by,

$$[M]_A = -\frac{\partial \{F\}_A}{\partial \{\dot V\}} = \begin{bmatrix} [M_{VV}] & [M_{V\omega}] \\ [M_{\omega V}] & [M_{\omega\omega}] \end{bmatrix} \tag{5.70}$$

where,

$$\hat{w}_{rf} = \left(\frac{w_{a_{rf}}}{u_{a_{rf}}}\right) \tag{5.71}$$

$$[M_{VV}] = \begin{bmatrix} m_u\hat{w}_{rf} & 0 & -m_u \\ 0 & 0 & 0 \\ m_w\hat{w}_{rf} & 0 & -m_w \end{bmatrix}, \text{slugs} \tag{5.71}$$

$$[M_{V\omega}] = \begin{bmatrix} -m_u y_{arf} & m_u(x_{arf}+\hat{w}_{rf}z_{arf}) & -m_u\hat{w}_{rf}y_{arf} \\ 0 & 0 & 0 \\ -m_w y_{arf} & m_w(x_{arf}+\hat{w}_{rf}z_{arf}) & -m_w\hat{w}_{rf}y_{arf} \end{bmatrix}, \text{slug-ft} \tag{5.72}$$

$$[M_{\omega V}] = \begin{bmatrix} m_w\hat{w}_{rf}y_{arf} & 0 & -m_w y_{arf} \\ (m_u z_{arf}+m_w x_{arf}+m_q)\hat{w}_{rf} & 0 & m_u z_{arf}+m_w x_{arf}-m_q \\ -m_u\hat{w}_{rf}y_{arf} & 0 & m_u y_{arf} \end{bmatrix}, \text{slug-ft} \tag{5.73}$$

and,

$$[M_{\omega\omega}] =$$

$$\begin{bmatrix} -m_w y_{arf}^2 & m_w(x_{arf}+\hat{w}_{rf}z_{arf})y_{arf} & -m_w\hat{w}_{rf}y_{arf}^2 \\ (-m_u z_{arf}+m_w x_{arf}-m_q)y_{arf} & (m_u z_{arf}-m_w x_{arf}+m_q)(x_{arf}+\hat{w}_{rf}z_{arf}) & (-m_u z_{arf}+m_w x_{arf}-m_q)\hat{w}_{rf}y_{arf} \\ m_u y_{arf}^2 & -m_u(x_{arf}+\hat{w}_{rf}z_{arf})y_{arf} & m_u\hat{w}_{rf}y_{arf}^2 \end{bmatrix}$$

$$, \text{slug-ft}^2 \tag{5.74}$$

5.7 Implementation of the aerodynamic model

As described in section 5.1, the aerodynamic force and moment coefficients are calculated in subroutines SFAERRF and USAERRF. In sections 5.7.1 and 5.7.2 the input/output variables of the subroutines SFAERRF and USAERF, respectively, are described.

5.7.1 SFAERRF - Input/Output List

The subroutine SFAERRF produces the total force and moment coefficients about the aerodynamic reference center for steady flow conditions (constant angle of attack and sideslip). The names of each input/output parameter as they appear in both the subroutine (SFAERRF) and in the *f18bas.csl* code from which SFAERRF is called, are listed in tables 5.3 and 5.4. The density altitude is used as an input to the flex/rigid multipliers and increments. Ground clearance was used for ground effect calculations. The calculations associated with ground effect have been removed, but the parameter list was left unchanged in the interest of compatibility.

Table 5.3. Input parameters for subroutine SFAERRF

Symbol	Definition	SFAERRF	.csl
a	angle of attack, degrees	ALPDEG	ALDGRF
β	sideslip angle, degrees	BETDEG	BEDGRF
M	Mach number	MACH	MACHRF
p	Body frame roll rate, rad/sec	P	P
q	Body frame pitch rate, rad/sec	Q	Q
r	Body frame yaw rate, rad/sec	R	R
H_ρ	Altitude above sea-level in a standard atmosphere equivalent to a given density, feet	ALT	HRF
H_{gcl}	Ground clearance, feet	HGCL	HGC
$\bar{q}$	Dynamic pressure, lbs/ft^2	QBAR	QBAR
$\bar{c}$	Reference aerodynamic chord, feet	CREF	CWRF
b	Reference aerodynamic span, feet	BREF	BWRF
V	Total airspeed at the aero reference center, ft/sec	VTRF	VTRF
δ_{H_R}	Right stabilator (horizontal tail), positive trailing edge down (+ t.e.d), degrees	DHTR	DSR
δ_{H_L}	Left stabilator (horizontal tail), + t.e.d, degrees	DHTL	DSL
δ_{A_R}	Right aileron, + t.e.d, degrees	DAR	DAR
δ_{A_L}	Left aileron, + t.e.d, degrees	DAL	DAL
δ_{R_R}	Right rudder, + t.e. left, degrees	DRR	DRR
δ_{R_L}	Left rudder, + t.e. left, degrees	DRL	-DRL
δ_{f_R}	Right trailing-edge flap, + t.e.d, degrees	DTFR	DFR
δ_{f_L}	Left trailing-edge flap, + t.e.d, degrees	DTFL	DFL
δ_{n_R}	Right leading-edge flap, + l.e.d, degrees	DLFR	DNR
δ_{n_L}	Left leading-edge flap, + l.e.d, degrees	DLFL	DNL

Table 5.3. Input parameters for subroutine SFAERRF (continued)

Symbol	Definition	SFAERRF	.csl
δ_{SB}	Speed brakes, 0 - 60, degrees	DSBK	DSB
δ_{LG}	Normalized landing gear, 0 - 1, full down	DLG	DLG
logical	(.T.) Detailed print of intermediate variables on unit=20.	LDEBUG	LDEBUG
	(.F.) No action		
logical	(.T.) Flex/rigid multipliers and increments calculated	LQSE	LQSE
	(.F.) Aero for rigid airplane, flex/rigid -> 1.		
logical	(.T.) Real time equivalent aero	LRTE	LRTE
	(.F.) Enhanced aerodynamic model		

Table 5.4. Output parameters for subroutine SFAERRF

Symbol	Definition	SFAERRF	.csl
$C_{D_{sf}}$	Total drag coefficient at a.r.c. evaluated at ($\dot{\alpha}$= 0)	CDREFO	CDRFSF
$C_{Y_{sf}}$	Total side-force coefficient at a.r.c. evaluated at ($\dot{\alpha}$= 0)	CYREFO	CYRFSF
$C_{L_{sf}}$	Total lift coefficient at a.r.c. evaluated at ($\dot{\alpha}$= 0)	CLREFO	CLRFSF
$C_{l_{sf}}$	Total roll coefficient at a.r.c. evaluated at ($\dot{\alpha}$= 0)	C1REFO	C1RFSF
$C_{m_{sf}}$	Total pitch coefficient at a.r.c. evaluated at ($\dot{\alpha}$= 0)	CMREFO	CMRFSF
$C_{n_{sf}}$	Total yaw coefficient at a.r.c. evaluated at ($\dot{\alpha}$= 0)	CNREFO	CNRFSF

5.7.2 USAERRF - Input/Output List

The subroutine USAERRF returns the lift and pitch derivatives with respect to the nondimensional rate-of-change of angle of attack. The names of each input/output parameter as they appear in both the subroutine (USAERRF) and in the *f18bas.csl* code from which USAERRF is called, are listed in tables 5.5 and 5.6.

Table 5.5. Input parameters for subroutine USAERRF

Symbol	Definition	SFAERRF	.csl
α	angle of attack, degrees	ALPDEG	ALDGRF
M	Mach number	MACH	MACHRF
H_ρ	Altitude above sea-level in a standard atmosphere equivalent to a given density, feet	ALT	HRF
logical	(.T.) Detailed print of intermediate variables on unit=20. (.F.) No action	LDEBUG	LDEBUG
logical	(.T.) Flex/rigid multipliers and increments calculated (.F.) Aero for rigid airplane, flex/rigid -> 1.	LQSE	LQSE

Table 5.6. Output parameters for subroutine USAERRF

Symbol	Definition	SFAERRF	.csl
$C_{L_{\dot\alpha}}$	$\partial C_L / \partial \frac{\dot\alpha \bar c}{2V}$	CLADRF	CLADRF
$C_{m_{\dot\alpha}}$	$\partial C_m / \partial \frac{\dot\alpha \bar c}{2V}$	CMADRF	CMADRF

6. ENGINE THRUST MODEL

The engine model takes inputs from the pilot throttle and an autothrottle command, uses current air data (altitude, Mach number, and dynamic pressure), and computes the engine forces acting on the airframe for the equations of motion described in section 4. This engine model is referred to as the "QT Half" model in MDC documentation and correspondence (MDC internal memorandum 338-5250, dated 5-Dec-80, from R.O. Michael and B.R. Williams), and apparently is a refined model (circa 1980) based on installed engine performance flight tests. It should be noted that the authors focused on implementing this model as provided, in order to facilitate comparison with an existing real-time simulation model. The authors are not able to explain some aspects of the architecture of this model -- these will be noted as appropriate in the text below.

The model from pilot/autopilot inputs to thrust determination for a given engine is represented graphically in figure 6.0. Since the airplane has two engines, this model is implemented in code by parallel sets of ACSL equations and external subroutine calls for the left and right engines. Although the autopilot input is provided for future use, in the current simulation this input, CAT, is set to zero. Therefore, PLA, the total throttle command, is equal to CP, the pilot throttle command.

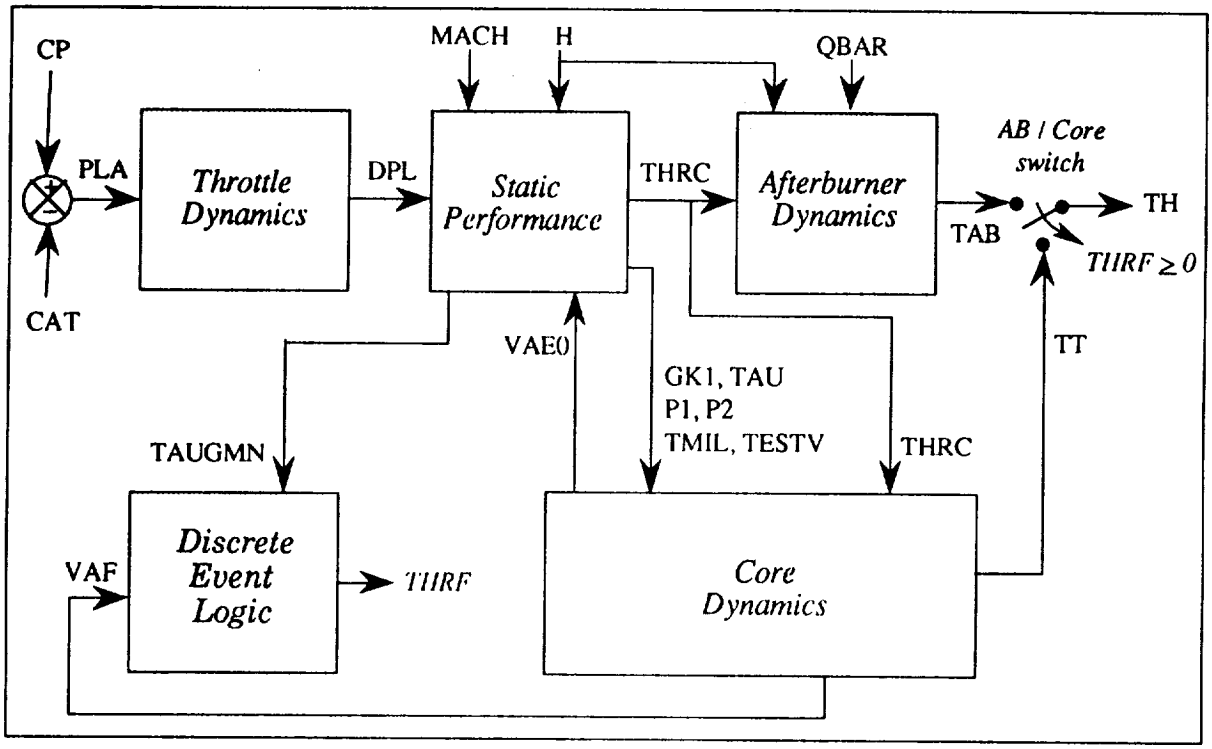


Figure 6.0. Engine Model Block Diagram

6.1 Throttle Dynamics

PLA is input to the *Throttle Dynamics*, a straightforward servo model with throttle position, DPL, as its output. This servo model includes a limited integrator, which functions as a normal integrator as long as its output remains between the indicated limits {31, 130}. If the integrator output equals or exceeds these limits, the state derivative DPLD is set to zero within the limited integrator macro and the integrator output is appropriately limited. The *limited integrator* block shown in figure 6.1 is the implementation provided by ACSL's built-in limited integrator command. This structure is shown for graphical convenience; in the ACSL simulation model, a

custom macro using IF condition tests is actually used. Either implementation should be equivalent; the custom macro was developed by the authors to retain control over the variable names of states and state derivatives [ACSL (1987)].

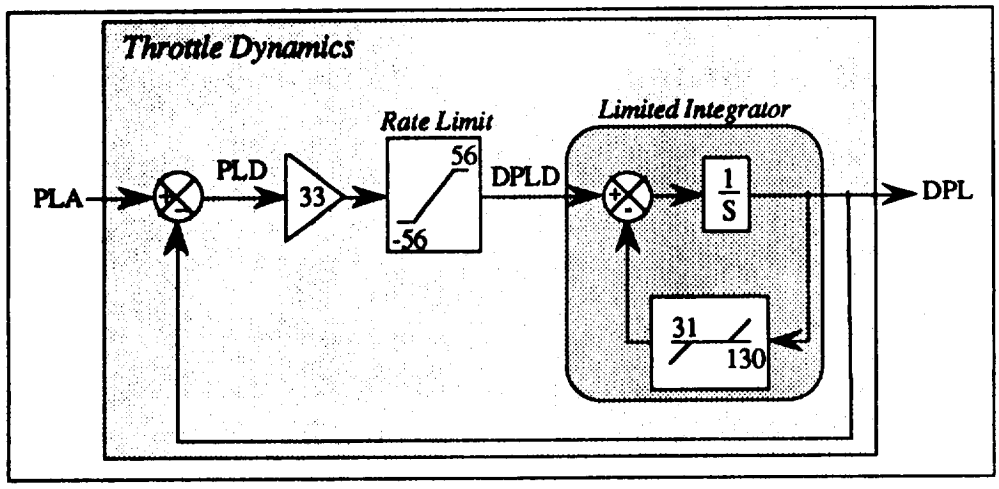


Figure 6.1. Throttle Dynamics

6.2 Static Performance

As shown in figure 6.2, throttle position (DPL), along with altitude (H), Mach number (MACH), and an internal engine dynamics state (VAE0) are input to the static performance module (Fortran subroutine ENG1). VAE0 is an artificial state created to break an algebraic loop for the static performance module. In the original specification, VAE is an input into the static performance module, and VAE is dependent on output THRC. VAE0 is created by passing VAE through a fast first order lag and is discussed further in section 6.5, *Core Dynamics*.

The outputs of the static performance module are engine thrust command (THRC), fuel flow (WF), idle thrust (TIDLE), maximum military power thrust (TMIL), minimum afterburner thrust (TAUGMN), maximum afterburner thrust (TAUGMX), and various gains and parameters for the engine dynamics (TESTV, GK1, P1, P2, TAU). THRC and WF are each computed as a table-lookup function, of H, MACH, and DPL. There are two table-lookup functions for THRC; one for military power throttle settings and one for afterburner throttle settings.

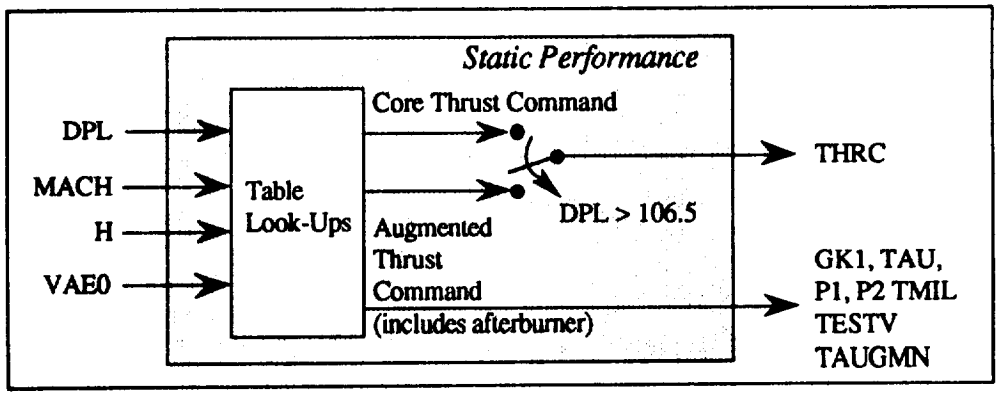


Figure 6.2. Static Engine Performance

The authors cannot fully explain the internal architecture of the static performance module -- the switch between the Core and the Augmented (afterburner) Thrust Command lookup tables occurs

at a DPL (PLA after dynamics) of 106.5°, rather than the expected military/afterburner throttle position boundary of 88-91° (PLA).

TIDLE, TMIL, TAUGMN, and TAUGMX are computed for appropriate fixed throttle positions using the same table-lookup function as for THRC. TESTV, GK1, P1, P2, TAU are computed as table-lookup functions of $\frac{VAE0}{TMIL}$.

THRC is used by two sets of engine dynamics running in parallel: *Afterburner Dynamics* and *Core Dynamics*. This will be described further below. GK1, TAU, TESTV, and TMIL are used in the *Core Dynamics* (see below). TAUGMN is used in the *Discrete Event Logic* (see below). WF, TIDLE, and TAUGMX are not currently used elsewhere in the ACSL simulation model.

6.3 Discrete Event Logic

As described above, the engine thrust command (THRC) is provided to two sets of engine dynamics running in parallel: *Afterburner Dynamics* and *Core Dynamics*. These dynamics provide a single thrust output that is used as the total net thrust for a given engine. The *Discrete Event Logic* (figure 6.3) is used to determine when to switch from one set of dynamics to the other, and implements code:

1) to select the appropriate thrust output from either *Afterburner Dynamics* or *Core Dynamics*;
2) to enable a smooth transition between the two dynamic systems by updating appropriate state variables -- this is accomplished in ACSL using DISCRETE blocks (one per engine) called by the ACSL SCHEDULE operator [ACSL (1987)].

The authors note that the *Discrete Event Logic* uses a thrust comparison test that is inconsistent with the switch test used in the static performance module to select the output of the thrust table lookup. The authors cannot explain why this is so; nor can we explain why this parallel dynamics architecture was chosen. It would seem more physically intuitive to implement an *Afterburner Dynamics* module that outputs an incremental thrust to be added to the *Core Dynamics* thrust to yield total net thrust.

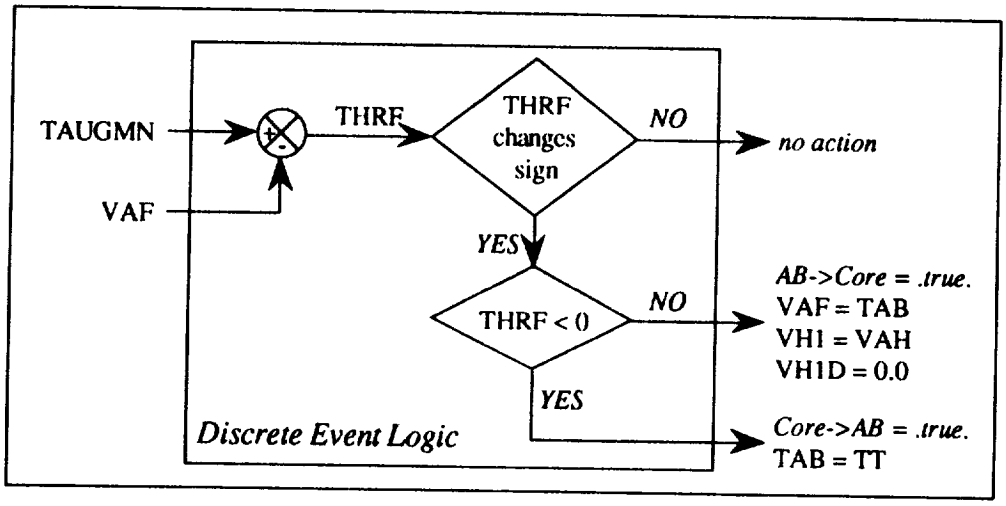


Figure 6.3. Discrete Event Logic

6.4 Afterburner Dynamics

As shown in figure 6.4, THRC is one of three inputs to the *Afterburner Dynamics*. The physics of the MDC model are such that when the thrust command (THRC) exceeds the afterburner thrust (TAB), TAB integrates up at the rate ABUL. Conversely, when TAB is greater than THRC, TAB integrates down at the rate ABLL.

ABUL and ABLL are computed based on current values of altitude (H) and dynamic pressure (QBAR). The lower rate limit, ABLL, is computed as a function of H,

$$ABLL = .0275\,H - 5500 \tag{6.1}$$

The upper rate limit, ABUL, is computed in two steps. The first step is to compute an intermediate value as a function of QBAR,

$$ABUL^* = 1.7\,QBAR^2 - 12.8\,QBAR + 3000 \tag{6.2}$$

ABUL is then computed by limiting ABUL* as,

$$ABUL = \begin{cases} 900 & ABUL^* < 900 \\ ABUL^* & 900 \le ABUL^* \le 4000 \\ 4000 & 4000 < ABUL^* \end{cases} \tag{6.3}$$

In the ACSL implementation of this system, the error signal THRC minus TAB is multiplied by a large gain (200) to yield ZZAB. ZZAB is passed through a rate limiter and is appropriately limited to ABUL or ABLL to yield RTAB. RTAB is then integrated to yield TAB. Synchronization of the *Afterburner Dynamics* with the *Core Dynamics* is achieved by updating TAB with the last value of core engine thrust, TT, when transition from *Core Dynamics* to *Afterburner Dynamics* occurs. This has the effect of "restarting" the afterburner dynamics from the thrust value at transition.

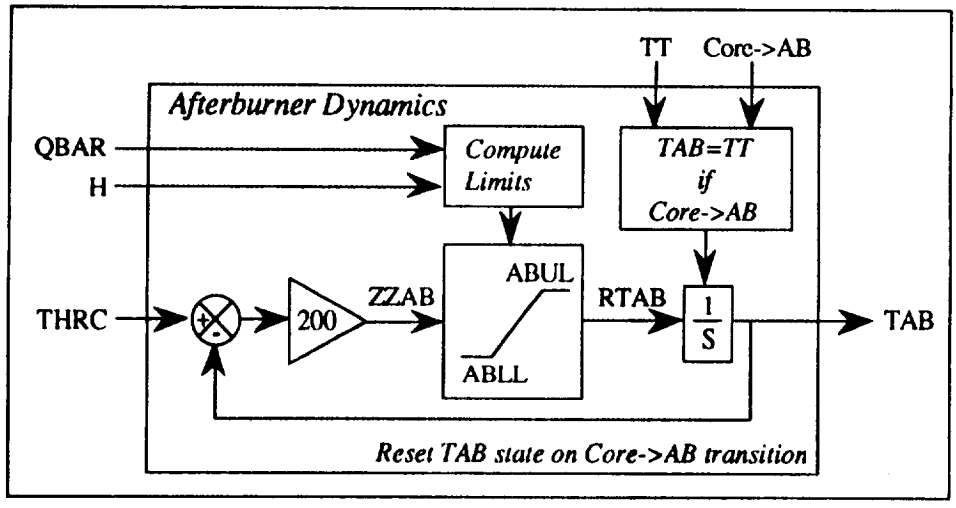


Figure 6.4. Afterburner Dynamics

6.5 Core Dynamics

As shown in figure 6.5, THRC is one of two primary inputs to the *Core Dynamics*, which implements a second order dynamics approximation to a more complicated, higher order dynamic model of the turbomachinery used for batch analysis of the engine by its manufacturer. The other five inputs are variables that have been described previously: TESTV, GK1, P1, P2, TAU.

The error signal VAC is computed by subtracting VAF from THRC. VAC is multiplied by the gain GK1 and is passed through the rate limiter bounded by limits RTEUL and RTELL to yield VAE. The rate limits RTEUL and RTELL are the engine spool-up/-down rate limits and are defined as CONSTANT variables in the ACSL simulation model. Therefore, although they have a "default" value, these limits can be altered at runtime [ACSL (1987)] by the user. The default values are ±5500.

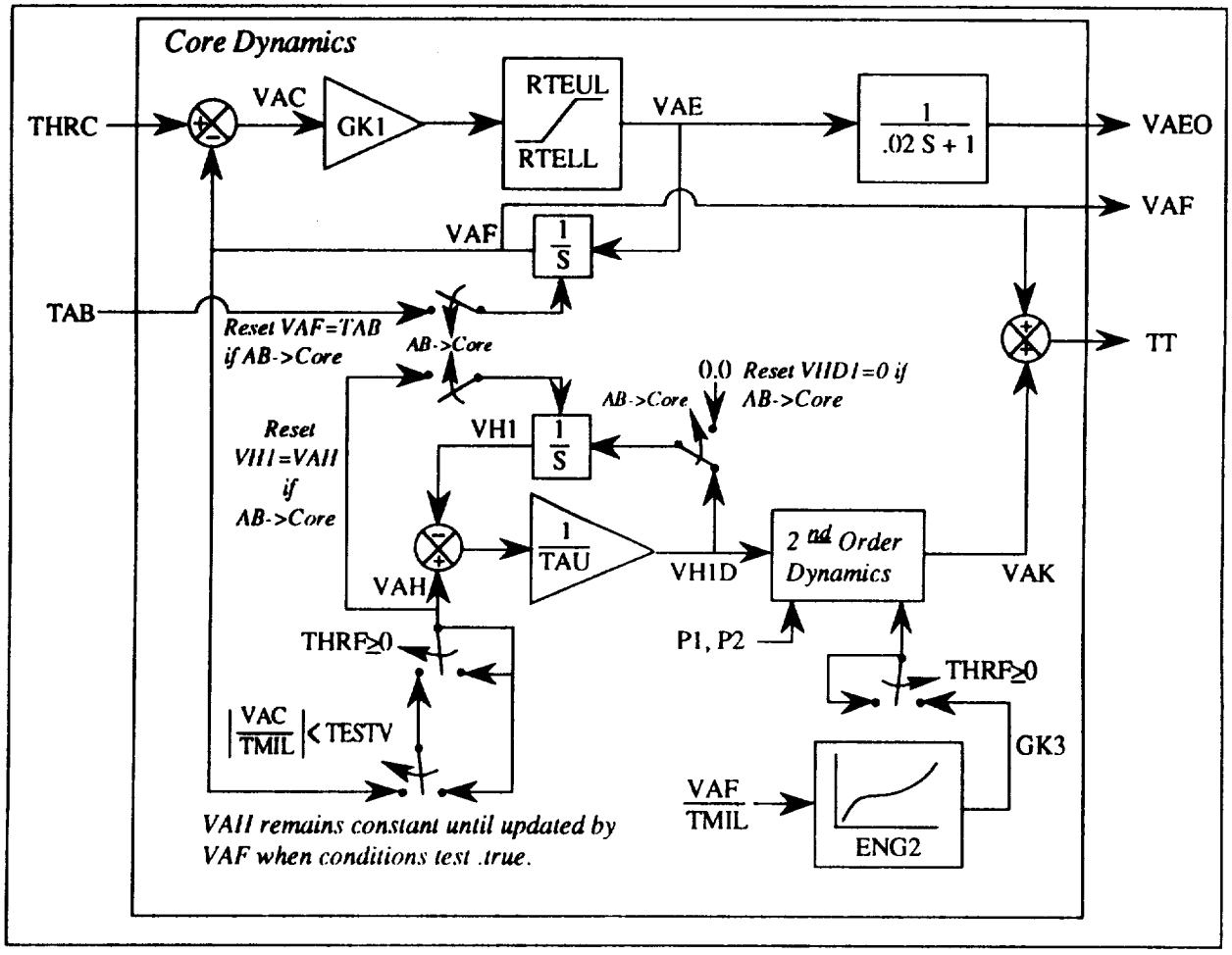


Figure 6.5. Core Dynamics

VAE is input to a low-pass filter to generate the artificial state VAE0 as described earlier to break the algebraic loop for the static performance module. The time constant for this filter is sufficiently small (50 rad/sec) to insure that all engine dynamics are unaffected by the filter. VAE is input to an integrator, yielding VAF. Synchronization of this integrator with the *Afterburner Dynamics* is achieved by updating VAF with the last value of afterburner engine thrust, TAB, when transition occurs from *Afterburner Dynamics* to *Core Dynamics*. This has the effect of "restarting" this integrator with the thrust value at transition.

VAF is the major component of the core engine thrust, represented by a rate-limited first order lag. VAF is used to form the error signal VAC as previously described. VAF is tested against TAUGMN in *Discrete Event Logic* to determine whether to initiate a transition between *Core Dynamics* and *Afterburner Dynamics*. VAF is supplied to the "Nonlinear Switch" [MDC internal memo 338-5250]. The "Nonlinear Switch" will be described below. VAF is summed with VAK to determine core engine thrust, TT.

The "Nonlinear Switch" can only be active when the core dynamics are being used. Inputs to this switch are VAF, VAC, TMIL and TESTV. At each simulation time step for which the condition

$$\mathrm{ABS}\left(\frac{\mathrm{VAC}}{\mathrm{TMIL}}\right) < \mathrm{TESTV} \tag{6.4}$$

is true, the "Nonlinear Switch" assigns VAH to the current value of VAF. Since the "Nonlinear Switch" is the only mechanism for defining VAH, it remains constant when the above condition is not true. TMIL is a function of Mach number and altitude; TESTV is a positive-valued function of VAE/TMIL, varying proportionally with the magnitude of VAE. Therefore VAH only changes when VAC is large relative to VAE (big change in throttle command).

Subroutine ENG2, coded in FORTRAN, is only called when the core dynamics are being used. The input is an internal engine dynamics parameter $\frac{\mathrm{VAF}}{\mathrm{TMIL}}$. The output is a gain for the second order dynamics block (GK3).

VH1 is subtracted from VAH and the result is divided by TAU, yielding VH1D. VH1D is integrated to give VH1. Synchronization of this integrator with the *Afterburner Dynamics* is achieved by setting VH1 equal to VAH when transition occurs from *Afterburner Dynamics* to *Core Dynamics*. By "zeroing" the input signal to this integrator, this part of the dynamics is not active at transition.

VH1D is input to the *2nd Order Dynamics* block, a transfer function with the following Laplace expression,

$$\frac{\mathrm{VAK}}{\mathrm{VH1D}} = \frac{(\mathrm{P2})^2(\mathrm{GK3})}{s^2 + 2(\mathrm{P1})(\mathrm{P2})s + (\mathrm{P2})^2(\mathrm{GK3})} \tag{6.5}$$

The VAK term provides the small "ballooning" effect for the engine described in MDC internal memo 338-5250 (1980). VAK is summed with VAF to give the total thrust for the *Core Dynamics*, TT.

6.6 Total Engine Thrust

The final total thrust for each engine, TH, is determined by the position of the AB/Core switch. If THRF $\geq$ 0, then TH is equal to the output of the *Core Dynamics*, TT; otherwise, TH is equal to the output of the *Afterburner Dynamics*, TAB.

6.7 Calculations for Equations of Motion

The equations of motion for the simulation, described in section 4, require the 3-axis body frame components of force and moment due to the thrust from the engines. Therefore, operations are performed on the final total thrust for each engine, TH, and then added together to yield these required force/moment components. Note that engine gyroscopic moment effects are not modeled.

In keeping with overall assumptions that an x-z plane of symmetry exists, the engines are assumed to be located symmetrically with respect to the body frame x-z plane. However, the thrust axis of each engine is not parallel to the x-axis. Each engine is canted out slightly (1.98°) so that the exit nozzles are closer to the airplane centerline than the front of the engines. The pitch or elevation angle of each engine is zero. Engine cant is measured by an azimuth angle (Ψ_{eng}) that represents rotation about the body frame z-axis. Given this information, the body-axis components of thrust for each engine are computed as,

$$\left.\begin{aligned} TH_{x;L} &= TH_L \cos(\Psi_{eng_L}) & TH_{y;L} &= TH_L \sin(\Psi_{eng_L}) \\ TH_{z;L} &= 0 & TH_{x;R} &= TH_R \cos(\Psi_{eng_R}) \\ TH_{y;R} &= TH_R \sin(\Psi_{eng_R}) & TH_{z;R} &= 0 \end{aligned}\right\} \tag{6.6}$$

where the subscripts "L" and "R" denote left and right engines, and where,

$$\Psi_{eng_L} = -1.98^\circ \text{ and } \Psi_{eng_R} = 1.98^\circ \tag{6.7}$$

The force quantities for the equations of motion can now be computed,

$$\left.\begin{aligned} F_{x_E} &= TH_{x;L} + TH_{x;R} \\ F_{y_E} &= TH_{y;L} + TH_{y;R} \\ F_{z_E} &= TH_{z;L} + TH_{z;R} \end{aligned}\right\} \tag{6.8}$$

The engine thrust is assumed to be applied where the thrust axis intersects the "engine/airframe interface", defined in the engineering drawing, figure 3.1. As can be seen from figure 3.1, the location of the "engine/airframe interface" for the right engine is defined in the FS/BL/WL coordinate frame as (687.5, 18.9, 100.0). The interface point for the left engine is (687.5, -18.9, 100.0). This information is specified in the variables FSENG, BLENGR, BLENGL, and WLENG in CONSTANT statements in the *f18bas* simulation. The vector that locates the engine/airframe interface point in the c.g.-centered body frame is computed in units of feet for use in calculating the moments about the c.g.,

$$\left.\begin{aligned} x_{crf} &= -(FS_{eng} - FS_{cg}) / 12 \\ y_{crf_R} &= (BL_{eng_R} - BL_{cg}) / 12 \\ y_{crf_L} &= (BL_{eng_L} - BL_{cg}) / 12 \\ z_{crf} &= -(WL_{eng} - WL_{cg}) / 12 \end{aligned}\right\} \tag{6.9}$$

The calculations of (6.9) are performed once in the initialization phase of the simulation. Given this information, the moment quantities for the equations of motion can now be computed,

$$\left.\begin{aligned} M_{x_E} &= -z_{crf} F_{y_E} + y_{crf_R} TH_{z;R} + y_{crf_L} TH_{z;L} \\ M_{y_E} &= z_{crf} F_{x_E} - x_{crf} F_{z_E} \\ M_{z_E} &= -(y_{crf_R} TH_{x;R} + y_{crf_L} TH_{x;L}) + x_{crf} F_{Y_E} \end{aligned}\right\} \tag{6.10}$$

7. SENSORS

The sensor code was taken from the LaRC real-time simulation (*dmsf18*). These models can also be found in subroutine FCSENS of the MDC simulation of the F/A-18. A table of sensor characteristics can be found in figure 7-7 of [MDC A7813, vol I, rev A, page 7-16] and selected parts are repeated below as Table 7.1. In the *dmsf18* simulation, the sensor calculations and dynamics are performed in the same subroutines that model the control laws. In keeping with a policy of modeling all continuous dynamics in the derivative portion of the ACSL simulation, sensor MACRO's were written to implement the sensor models, and the sensor code was stripped out of the control law subroutines that are called from the FCS discrete block.

Table 7.1.a . Rate Gyro and Accelerometer Parameter Bounds

Parameter	Pitch/Yaw Rate Gyro	Roll Rate Gyro	Normal Accelerometer	Lateral Accelerometer
Range	± 60 °/sec	± 300 °/sec	± 10 g	± 2 g
Threshold	.005 °/sec	.01 °/sec	.002 g	.002 g
Linearity	1 % full scale	1 % full scale	1 % full scale	1 % full scale
Null	.5 °/sec	1.5 °/sec	.05 g	.01 g
Hysteresis	.05 °/sec	.30 °/sec	.005 g	.001 g
Gradient Tolerance	± 5 %	± 5 %	± 5 %	± 5 %

Table 7.1.b. Pedal Force and Pressure Measurement Parameter Bounds

Parameter	Pedal Transducer	Static Pressure	Dynamic pressure
Range	± 134 lbs	[0,38] in HG	[0,55] in HG
Threshold	.5 lbs	.0005 in HG	.001 in HG
Linearity	2 % full scale	incl in gradient tolerance	incl in gradient tolerance
Null	0.5 % full scale	0.09 in HG	0.105 in HG
Hysteresis	1.0 lb (0 to 60 lbs) 4.0 lb (60 to 134 lbs)	incl in gradient tolerance	incl in gradient tolerance
Gradient Tolerance	± 5 %	± 0.3 %	± 0.3 %

The sensor outputs required by the subroutines that model the flight control laws are given in Table 7.2. The pedal force is an autonomous input to the simulation.

Table 7.2. Sensor Outputs

Symbol	Variable Name	Definition
α_s	ALFS	angle of attack, degrees
n_{y_s}	AYS	lateral acceleration, positive right, g's
n_{z_s}	AZS	vertical acceleration, positive up, g's
p_s	PS	body axis roll rate, °/sec
q_s	QS	body axis pitch rate, °/sec
r_s	RS	body axis yaw rate, °/sec
Qc_s	QCIS	compressible impact pressure, PSF
P_{S_s}	PSTS	static pressure, PSF

The sensor values used by the flight control system model can be either the ideal values or values computed using various sensor models. The variable ISENS determines which type of sensor data is sent to the flight control subroutines. The ISENS variable was made integer type in order to accommodate multiple sensor models. The simulation currently calculates two sets of sensor outputs. The first set consists of the variables created by adding a "0" to the variable names in table 7.2, i.e. {ALFS0, AYS0, ..., PSTS0}, and represents ideal sensors. The second set consists of the variables created by adding a "1" to the list in Table 7.2, i.e. {ALFS1, AYS1, ..., PSTS1}, and represents the sensor models used in the MDC simulation. At one time, a third, more complex, set of sensor models was implemented in the AGCB simulation to produce the variable set {ALFS2, AYS2, ..., PSTS2}. The ISENS=2 option was removed to reduce complexity and because it was not entirely integrated with the simulation trimming algorithm, but could be reinstalled if the need arose. The final output variables in Table 7.2 are determined from the sensor model sets according to the value of the integer switch, ISENS, as shown in figure 7.1.

```
IF (ISENS=0) THEN
   ALFS = ALFS0
   ⋮
   PSTS = PSTS0
ELSE
   ALFS = ALFS1
   ⋮
   PSTS = PSTS1
END IF
```

Figure 7.1. Code block that selects sensor model

The sensor equations appear in both the initial section of the program code and in the derivative section. The initial section contains full code for the computations. The derivative section makes use of macros which appear at the top of the ACSL program as ACCEL1, ALGBT1, PITOT, and RGYRO1.

Ideal static pressure (PSTATC) is computed from the aircraft altitude (H, Height, in the main loop, HIC, Height Initial Condition, in the initial section) through the subroutine call to PSTAT. It interpolates for the static pressure value from an single independent variable (H) ATMAT62 data table. Ideal compressible impact pressure (QC) is computed from the ideal static pressure (PSTATC) and the Mach number (MACH, for main loop, MACHTR, MACH TRim, for the initial section) through the subroutine CPRESS. It interpolates for the compressible impact pressure from a two independent variable (MACH and PSTATC) data table.

Sensed static pressure (PSTS1, ISENS = 1) is computed from the ideal pressure divided by a scaling factor, SPSFXD. SPSFXD is computed as a function of Mach number and sensed angle of attack (ALFS1). The QCDA,QCDB...QCDH variables which appear in the scaling equation are a function of Mach number and are defined as constants in the initial section of the program code. The sensed compressible pressure (QCIS1, ISENS=1) is also determined as a function of Mach and angle of attack, where the static pressure is added to the ideal compressible pressure before the computation and then removed afterwards. A final computation is then made based on a limited angle of attack. These computations model the Pitot tube and static probe mechanics, where the measured value, as determined by pressure differentials, does not exactly reflect the true air pressure values. These pressure values are used in the flight control system for air data scheduling. The air data scheduled gain values were designed to be scheduled from measured pressure values and not the true or ideal pressures.

Sensed angle of attack is computed from the true angle of attack with a gain and bias based on air flow. The bias, (ALFB), is determined by the landing gear position, if the landing gear is retracted (the normal position for any runs) ALFB=1.9, extended ALFB=1.2. The gain, GKAOA, (where it is divided into the true alpha minus the bias) is a function of the Mach number. The measured angle of attack is limited to -14 and 56 degrees.

7.1 Ideal Static Pressure and Compressible Impact Pressure

Static pressure, P_S, is calculated as a function of altitude in feet by the subroutine PSTAT. The function,

$$P_S, \text{ lbs/ft}^2 = F_{PSTAT}(H)$$

implemented in subroutine PSTAT is plotted in figure 7.2.

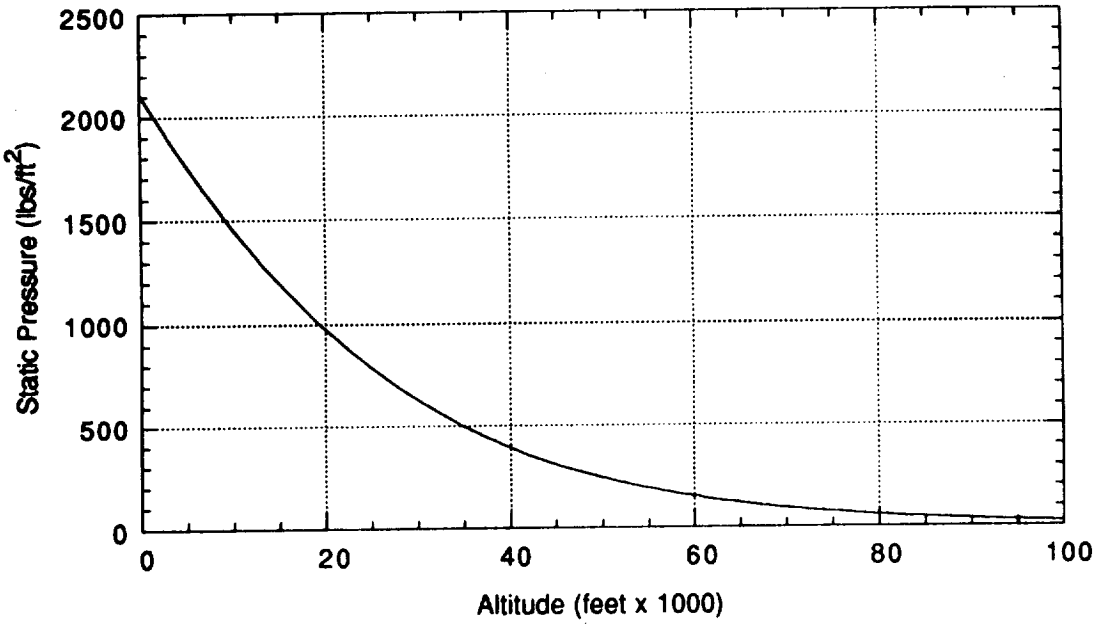


Figure 7.2. Static Pressure as a Function of Altitude

Subroutine CPRESS calculates compressible dynamic pressure as a function of Mach and static pressure, P_S, as follows:

$$Q_C = (Q_C/P_S)\, P_S$$

where the ratio Q_C/P_S is a function of Mach number and is given by figure 7.3. The subroutines PSTAT and CPRESS are called from the DERIVATIVE block.

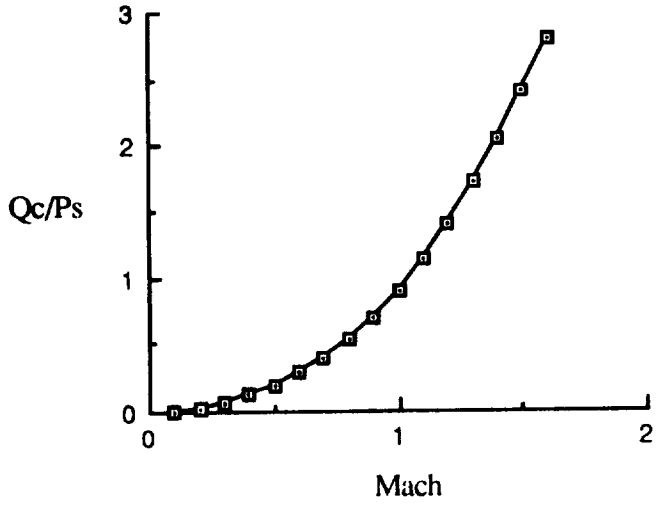


Figure 7.3. Ratio of Compressible to Static Pressure as Function of Mach

7.2 Ideal Sensor Set (ISENS=0)

$$\alpha_{s0} = \alpha, \text{ degrees} \qquad \text{ALFS0}$$

$$n_{y_{s0}} = n_y, \text{ g's} \qquad \text{AYS0}$$

$$n_{z_{s0}} = -n_z - 1, \text{ g's} \qquad \text{AZS0}$$

$$p_{s0} = (180/\pi)p, \text{ °/sec} \qquad \text{PS0}$$

$$q_{s0} = (180/\pi)q, \text{ °/sec} \qquad \text{QS0}$$

$$r_{s0} = (180/\pi)r, \text{ °/sec} \qquad \text{RS0}$$

$$Q_{c_{s0}} = Q_c, \text{ PSF} \qquad \text{QCIS0}$$

$$P_{S_{s0}} = P_S, \text{ PSF} \qquad \text{PSTS0}$$

where,

$$n_y = \{\dot{v} + ru - pw - gl_{yz}\} / g$$

$$n_z = \{\dot{w} + pv - qu - gl_{zz}\} / g$$

7.3 Accelerometers (ISENS=1)

The calculations for the lateral and vertical accelerometers are performed in the macro ACCEL1. The lateral accelerometer output, AYS1, is positive for an acceleration to the right and the vertical accelerometer output, AZS1, is positive for an upward acceleration. Both AYS1 and AZS1 have units of g's. Lever arm effects are calculated and the 1 g bias removed from AZS1. Both outputs are bounded according to Table 7.1.a. Hysteresis, bias, quantization and misalignment are not modeled in the MDC sensor set. The MDC simulation, *mdcf18*, has comments that refer to a transfer function of $G(s) = \frac{1}{(s/220)+1}$ that approximates the accelerometer dynamics, but was deleted from the MDC simulation. The *f18bas* simulation does not model accelerometer dynamics either. The location of the accelerometers in the body frame in feet is given by,

$$x_a = -(FS_a - FS_{cg}) / 12.$$

$$y_a = (BL_a - BL_{cg}) / 12.$$

$$z_a = -(WL_a - WL_{cg}) / 12.$$

where x_a, y_a, z_a represent distances in feet forward from the c.g., right of the c.g. and below the c.g., respectively, and where,

FS_{cg}	= FSCG, Fuselage Station c.g.	=	455.0 inches
BL_{cg}	= BLCG, Buttock Line c.g.	=	0.0 inches
WL_{cg}	= WLCG, Water Line c.g.	=	102.8 inches
FS_a	= NYLOC(1)	=	306.75 inches
BL_a	= NYLOC(2)	=	10.5 inches
WL_a	= NYLOC(3)	=	91.3 inches

The lateral and vertical accelerations with lever arm effects included and 1 g bias removed are,

$$n_{y_s} = n_y + \{(pq + \dot{r})x_a - (p^2+r^2)y_a + (qr - \dot{p})z_a\}/g$$

$$n_{z_s} = -1 - n_z - \{(pr - \dot{q})x_a + (qr + \dot{p})y_a - (p^2+q^2)z_a\}/g$$

The ISENS=1 output accelerations are bounded according to table 7-1a.

$$n_{y_{s1}} = n_{y_s} \left[\begin{matrix}2\\-2\end{matrix}\right. \qquad \text{(AYS1)}$$

$$n_{z_{s1}} = n_{z_s} \left[\begin{matrix}10\\-10\end{matrix}\right. \qquad \text{(AZS1)}$$

7.4 Rate Gyros (ISENS=1)

The outputs of the rate gyros are simply the body frame roll, pitch, and yaw rates in units of degrees/sec with limits applied as indicated in table 7.1a. The calculations are performed in macro RGYRO1 as follows,

$$p_{s1} = \left(\frac{180}{\pi} p\right) \left[\begin{matrix}300\\-300\end{matrix}\right. \text{°/sec} \qquad \text{(PS1)}$$

$$q_{s1} = \left(\frac{180}{\pi} q\right) \left[\begin{matrix}60\\-60\end{matrix}\right. \text{°/sec} \qquad \text{(QS1)}$$

$$r_{s1} = \left(\frac{180}{\pi} r\right) \left[\begin{matrix}60\\-60\end{matrix}\right. \text{°/sec} \qquad \text{(RS1)}$$

7.5 Angle of Attack Measurement (ISENS=1)

The angle-of-attack probe model is implemented in macro ALFBT1. The "probe" quantity is calculated by passing the angle of attack at the c.g. through a filter representing the probe dynamics,

$$\alpha_p = \left(\frac{1}{0.073\ s\ +\ 1}\right)\alpha$$

The local probe angle of attack is adjusted for the induced upwash due to pitch rate,

$$\alpha_{p_1} = \alpha_p - (180/\pi)\frac{l_p q}{V}$$

where l_p = 24 feet. From purely geometric consideration, one would expect l_p to be the distance from the center-of-gravity to the angle-of-attack probe where l_p has units of feet and is positive if the probe is forward of the c.g. A probe located at a fuselage station (FS) of 167 inches is consistent with l_p = 288 inches and a nominal c.g. location of FS_{cg} = 455.0 inches, placing the probe between the nose (FS=60.50) and the leading edge of the cockpit (FS=188.0) (see figure 3.1). However, one must wonder why the probe dynamics are not applied to the total local measurement that includes pitch rate effects instead of to the angle of attack at the c.g. At this point, the only answer is that this is the way it is done in subroutine FCSENS of the MDC simulation.

An angle-of-attack probe bias and gain inverse are calculated. The bias is given by,

$$\alpha_{p_{bias}} = \begin{cases} 1.9 \text{ degrees if gear up} \\ 1.2 \text{ degrees if gear down.} \end{cases}$$

The gain inverse is a function of limited Mach number,

$$GK_\alpha = F_{GKAOA}\left(M\ \begin{bmatrix} 1.4 \\ .9 \end{bmatrix}\right)$$

where the function F_{GKAOA} is defined by figure 7.4.

The final limited sensor output is then,

$$\alpha_{s1} = \frac{\alpha_{p_1} - \alpha_{p_{bias}}}{GK_\alpha}\begin{bmatrix} 56 \\ -14 \end{bmatrix} \qquad \text{(ALFS1)}$$

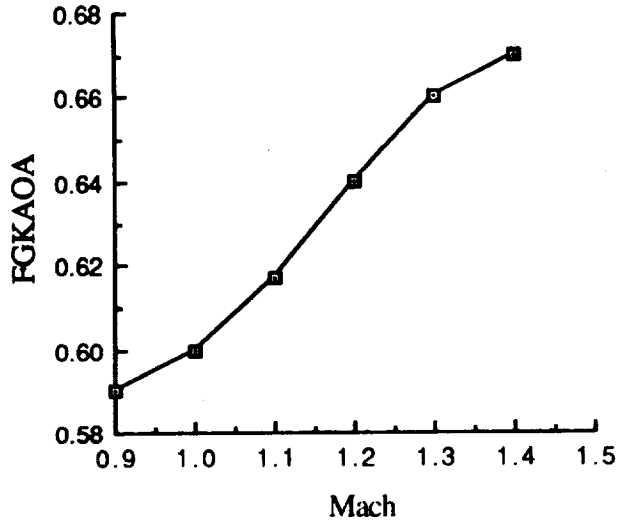


Figure 7.4. Inverse of Sensed AOA Gain

7.6 Air Data Parameters (ISENS=1)

The calculation of indicated static pressure and indicated compressible dynamic pressure requires eight parameters, QCDx, where x={A,B,...,H}. The QCDx are 21 element arrays and the current value for each QCDx parameter is determined as a tabular function of Mach. XHM1 is the similarly sized independent variable array of Mach numbers with a maximum value of 1.65 and a minimum of 0. A search is performed to determine the integer I_{XHM} such that $XHM1(I_{XHM}) \leq$ Mach $< XHM1(I_{XHM}+1)$. The I_{XHM}'th elements of the QCDx arrays are used in the following calculations. No interpolation is done on the QCDx variables. In the notation below, $Q_{CDx} = QCDx(I_{XHM})$. The QCDx parameters as functions of Mach are plotted in figure 7.5.a-h.

The calculation of the sensed static pressure for ISENS=1, $P_{S_{S1}}$, is shown in equations 7.1 through 7.3.

$$S_{PSFXD} = 1. - Q_{CDB}M - Q_{CDA} - \{Q_{CDB}M + Q_{CDC}\}\alpha_{s_1} \tag{7.1}$$

$$P_{S_{S1}} = P_S / S_{PSFXD} \tag{7.2}$$

$$PSTS1 = P_{S_{S1}} \tag{7.3}$$

The calculation of indicated compressible dynamic pressure is shown in equations 7.4 through 7.8.

$$P_T = Q_C + P_S \tag{7.4}$$

$$P_{T_I} = \frac{P_T}{1.0 - Q_{CDF}M - Q_{CDE} - \{Q_{CDH}M + Q_{CDG}\}\alpha_{s_1}} \tag{7.5}$$

An uncorrected initial value for indicated compressible dynamic pressure is calculated as follows,

$$Q_{c_x} = P_{T_I} + P_{S_{s1}} \tag{7.6}$$

The initial value for indicated compressible dynamic pressure is then adjusted for angle-of-attack effects per MDC memo 341-6241 dated "23Feb81".

$$Q_{c_{s1}} = Q_{c_x}\{1 - (0.04\{\alpha\,[^{55}_{30} - 30\})^2\} \tag{7.7}$$

$$QCIS1 = Q_{c_{s1}} \tag{7.8}$$

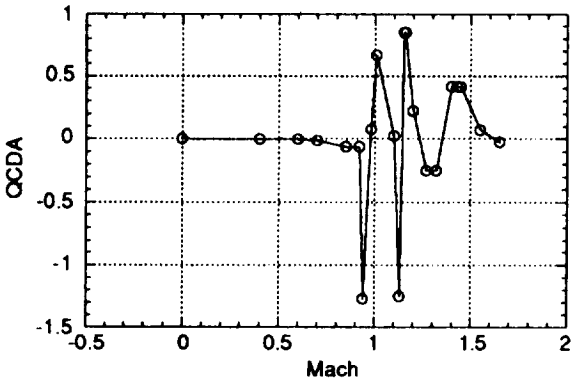


Figure 7.5.a QCDA = f(Mach)

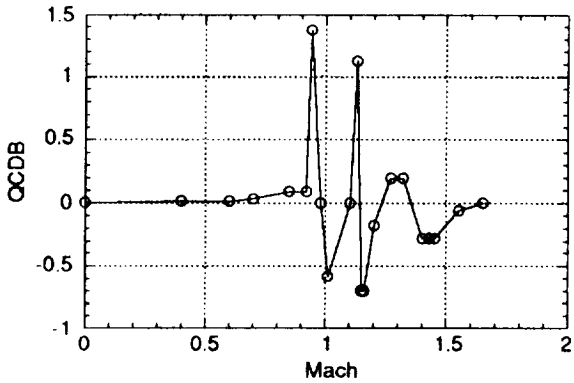


Figure 7.5.b QCDB = f(Mach)

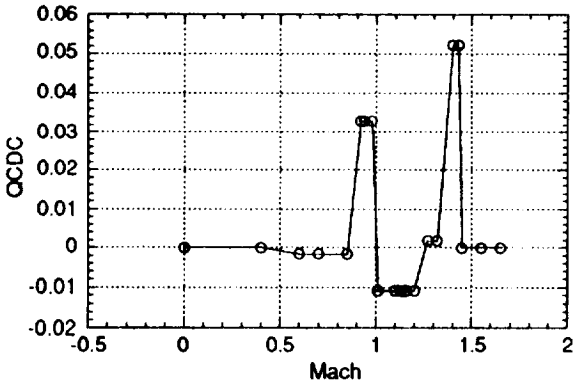


Figure 7.5.c QCDC = f(Mach)

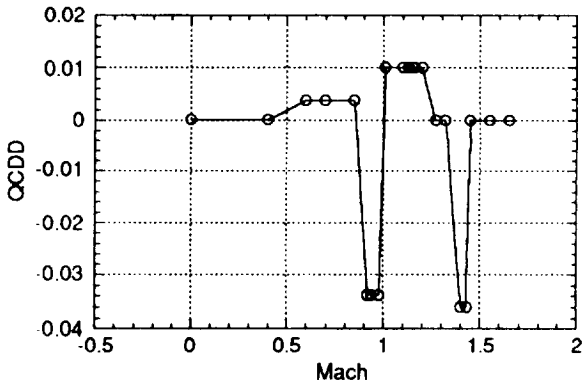


Figure 7.5.d QCDD = f(Mach)

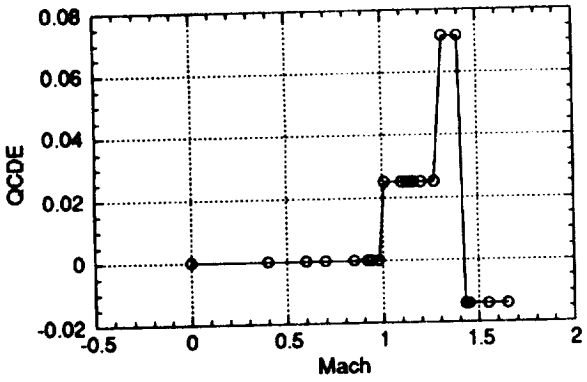


Figure 7.5.e QCDE = f(Mach)

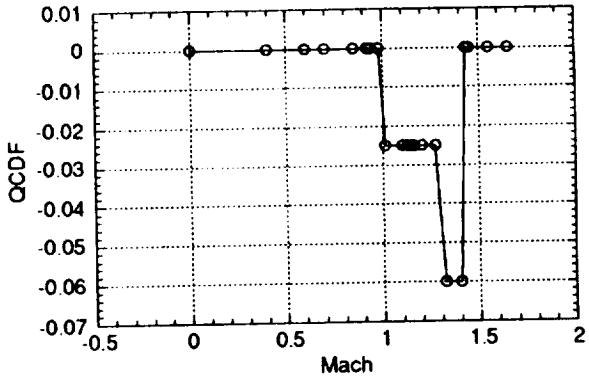


Figure 7.5.f QCDF = f(Mach)

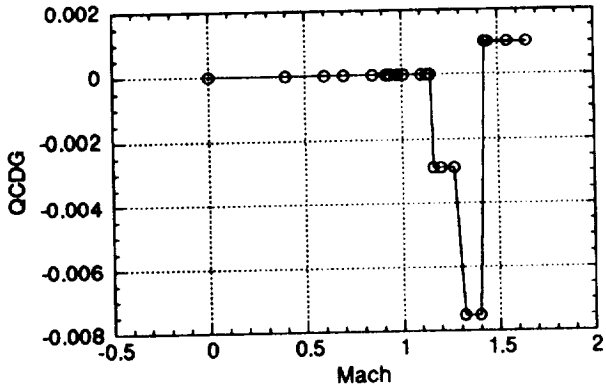


Figure 7.5.g QCDG = f(Mach)

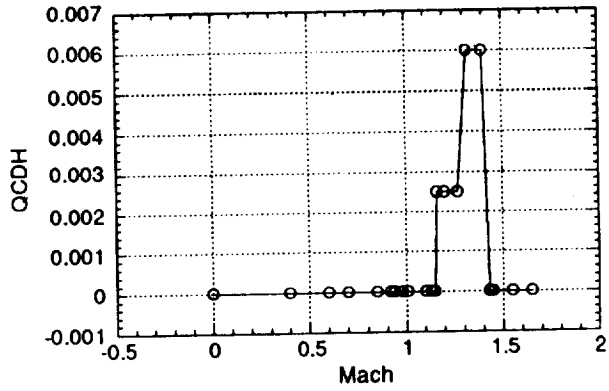


Figure 7.5.h QCDH = f(Mach)

8. ACTUATORS

The simulation described herein includes dynamic models for the following control surfaces,

(1) stabilators (right and left),
(2) rudders (right and left),
(3) ailerons (right and left),
(4) trailing-edge flaps (right and left),
(5) leading-edge flaps (right and left),
(6) speed brake, and
(7) thrust vectoring vanes (2 per engine, 4 total).

The stabilators, rudders, ailerons, trailing-edge, and leading-edge flaps are hydraulically powered and are referred to in both MDC documents and this report as the primary controls. The single speed brake is positioned between and forward of the vertical fins and is driven at a constant rate of 24 degrees per second. The lower and upper limits of the speed brake deflection are 0° and 60°. The primary controls and the speedbrake are found on the basic F/A-18 airframe.

The thrust vectoring vane actuation systems modeled in this simulation reflect a preliminary design of the thrust vectoring system installed on the F-18/HARV and do not agree with the final installation. The documentation of the final system will be undertaken in a separate report. The thrust vectoring vanes are each modeled as first order transfer functions with a steady state gain of one, a time constant of (1/30) seconds, rate limits of 80 °/s and position limits of +/- 30°.

8.1 Linear Dynamics - Primary Controls

The actuation systems for the primary controls in their normal mode of operation are discussed in detail in pages 6-1 through 6-40 of MDC A7813, vol II. The servo loop block diagrams for the primary controls are given in figures 6-2, 6-4, 6-6, 6-8 and 6-10 of MDC A7813, vol II. These block diagrams are repeated in a simplified form in this document as figures 8.1 through 8.5. The high order models represented in these figures describe main ram response with external load.

The transfer functions for the five primary actuators (stabilator, rudder, aileron, leading-edge flaps, and trailing-edge flaps) are presented in Table 8.1 through 8.5. Each table contains three transfer function models of the corresponding actuator type; (a) a high order model; (b) an intermediate order model; and (c) a low order model. For each actuator type, transfer function (a) is derived from the high order block diagrams in figures 8.1 through 8.5. Transfer functions (b) are discussed in pages 6-18 through 6-24 of MDC A7813, vol II. The (b) transfer functions are second order for all except the stabilators, which are fourth order. Transfer functions (c) consists of the lowest order models, the first order transfer functions that are used in both the MDC simulation code and the DMS real-time simulation at LaRC. The (a) transfer functions can be realized by linearizing the nonlinear block diagrams of figures 8.1 to 8.5. Comparisons of gain and phase response of the (a), (b), and (c) transfer functions for the primary controls are shown in figures 8.6 through 8.10. The low order models agree pretty well in phase with the intermediate and high order models out to about 10 radians/second.

An initial form of the ACSL simulation included elastic dynamics (see section 1.2), and in that simulation the (b) transfer functions were implemented. When the elastic dynamics were removed and time histories were compared with results generated by the DMS simulation, significant differences in aircraft responses due to pilot pitch and yaw doublets resulted. These differences were much reduced when the stabilator and rudder actuator models were changed to the first order models. The ACSL simulation now uses the first order actuator models.

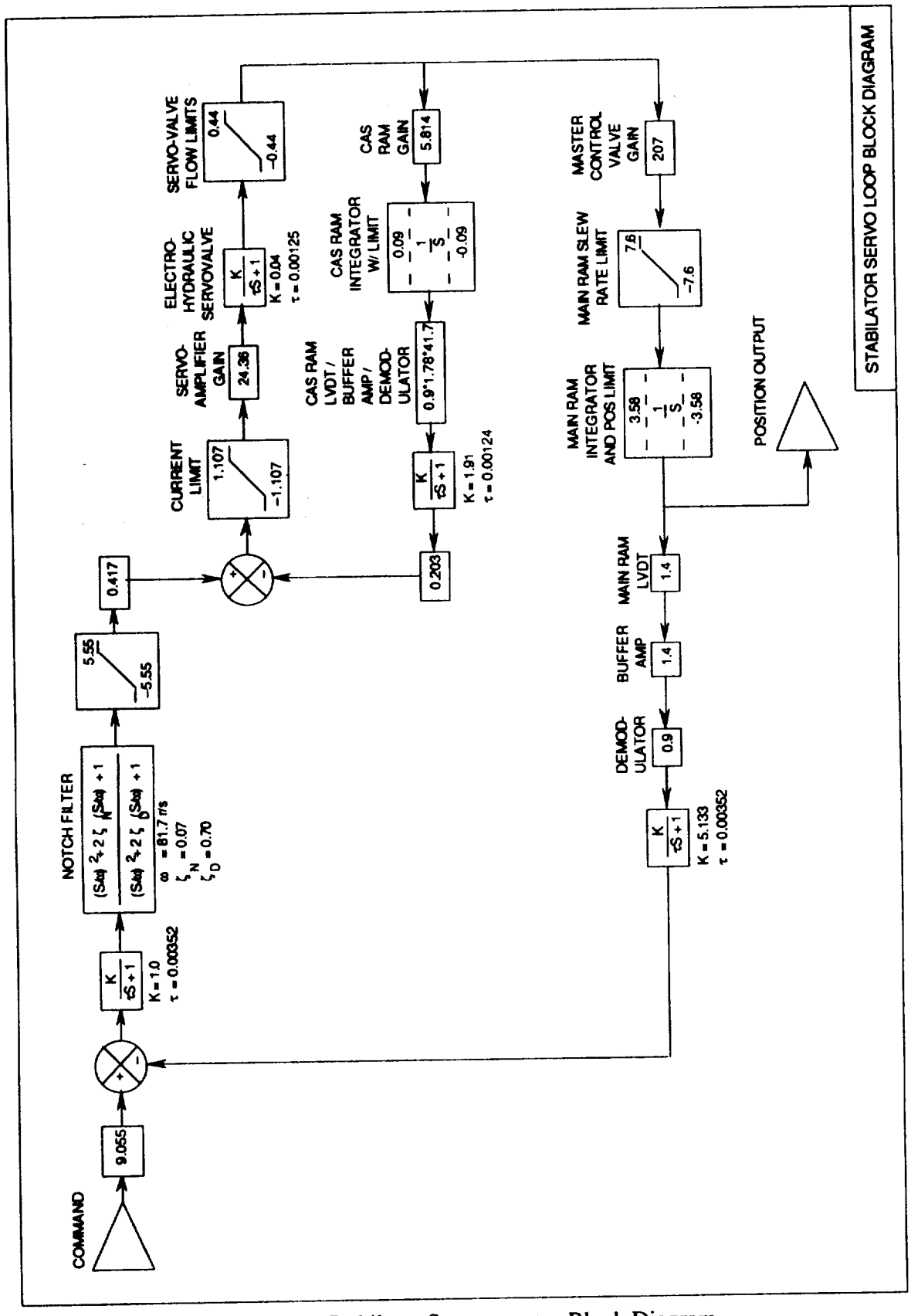


Figure 8.1. Stabilator Servoactuator Block Diagram

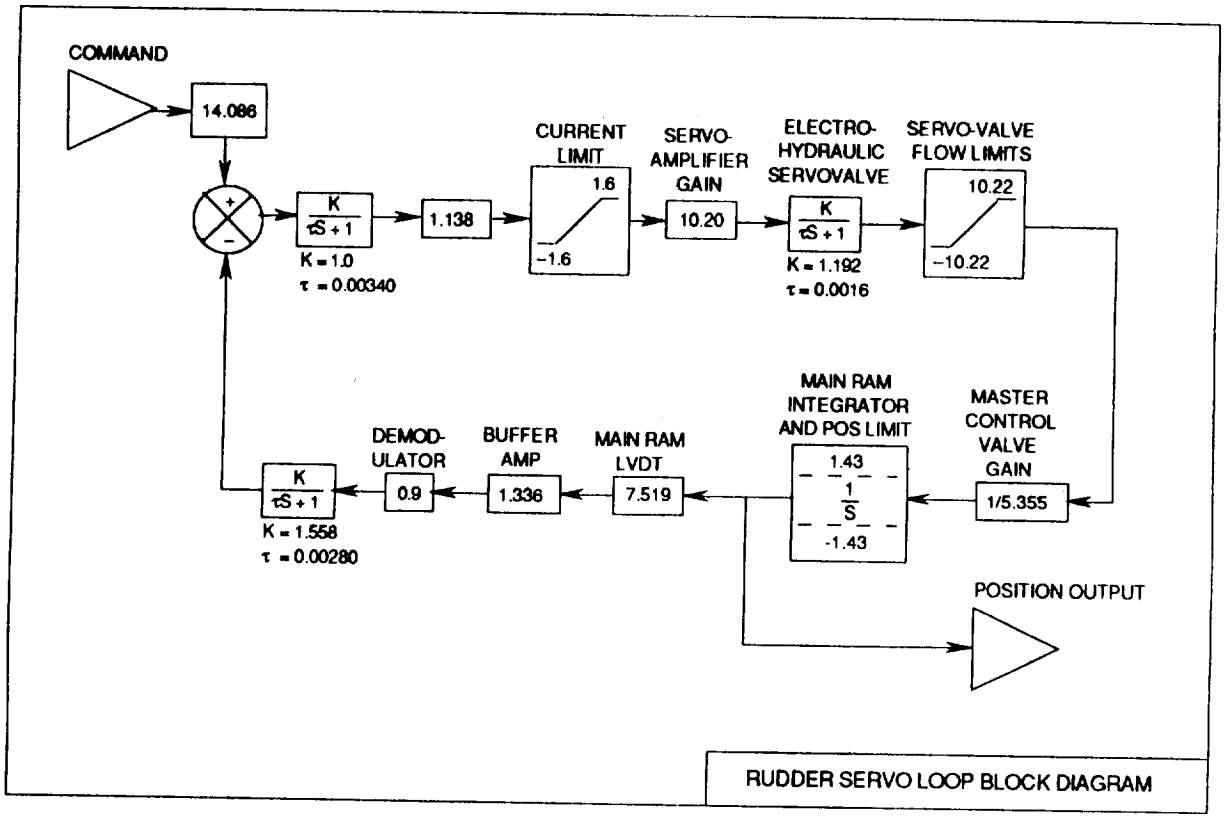


Figure 8.2. Rudder Servoactuator Block Diagram

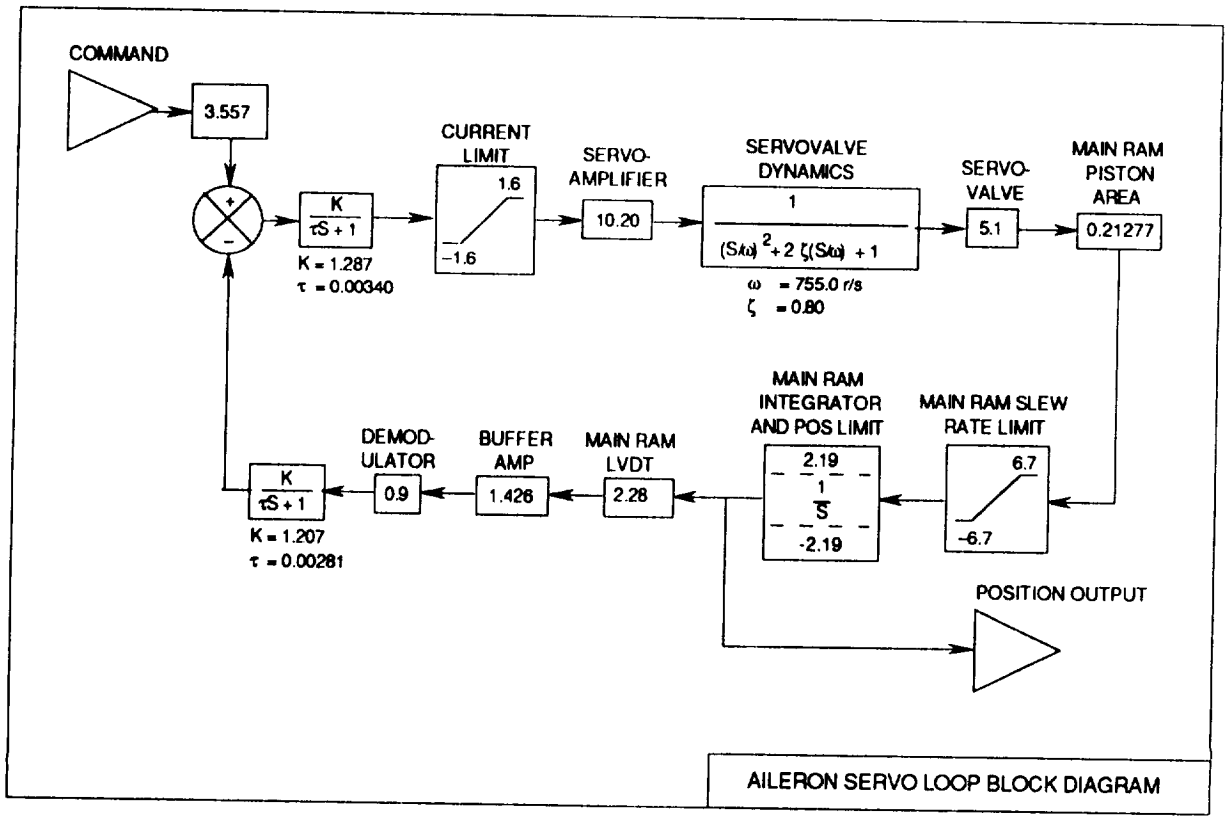


Figure 8.3. Aileron Servoactuator Block Diagram

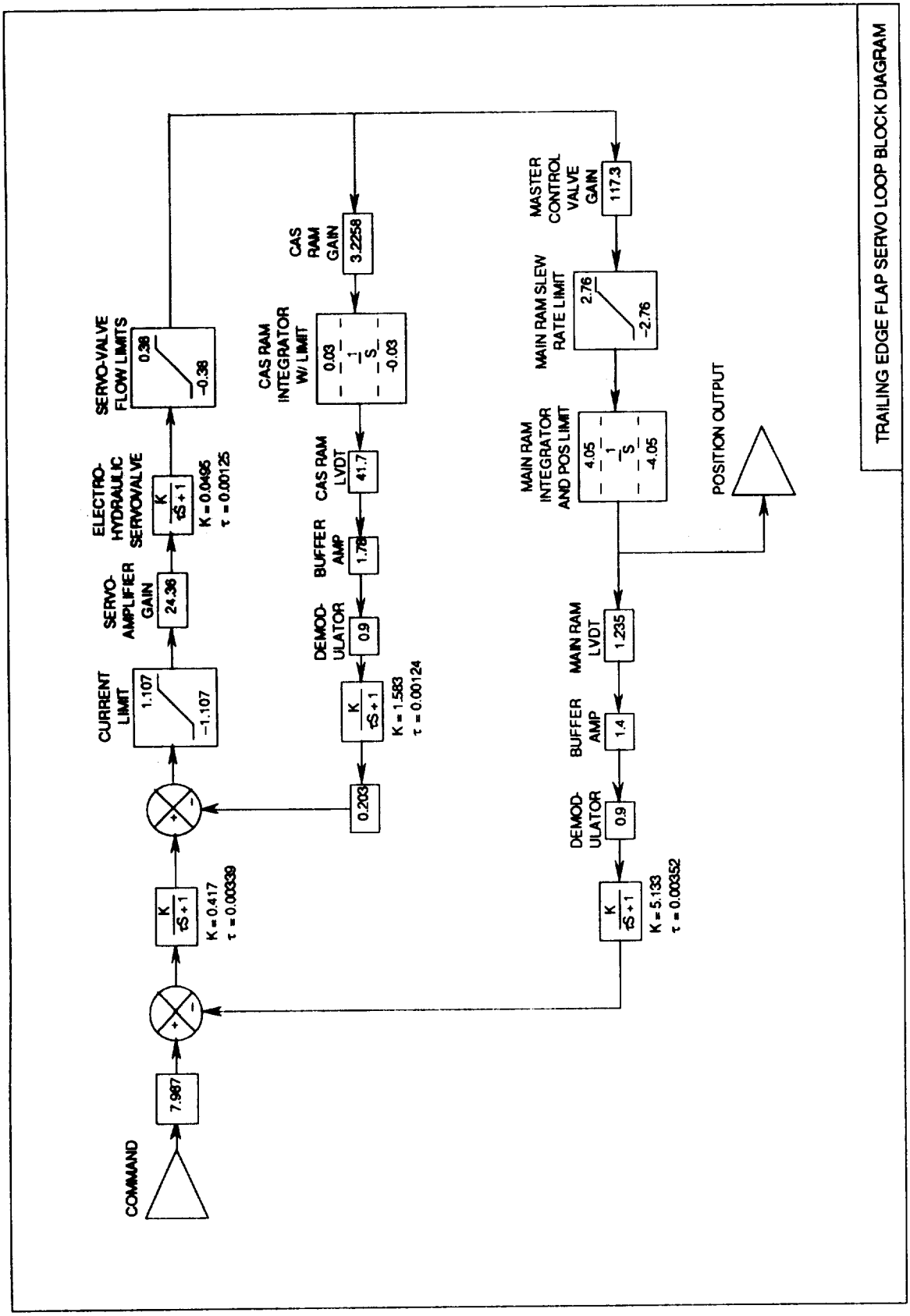


Figure 8.4. Trailing-Edge Flaps Servoactuator Block Diagram

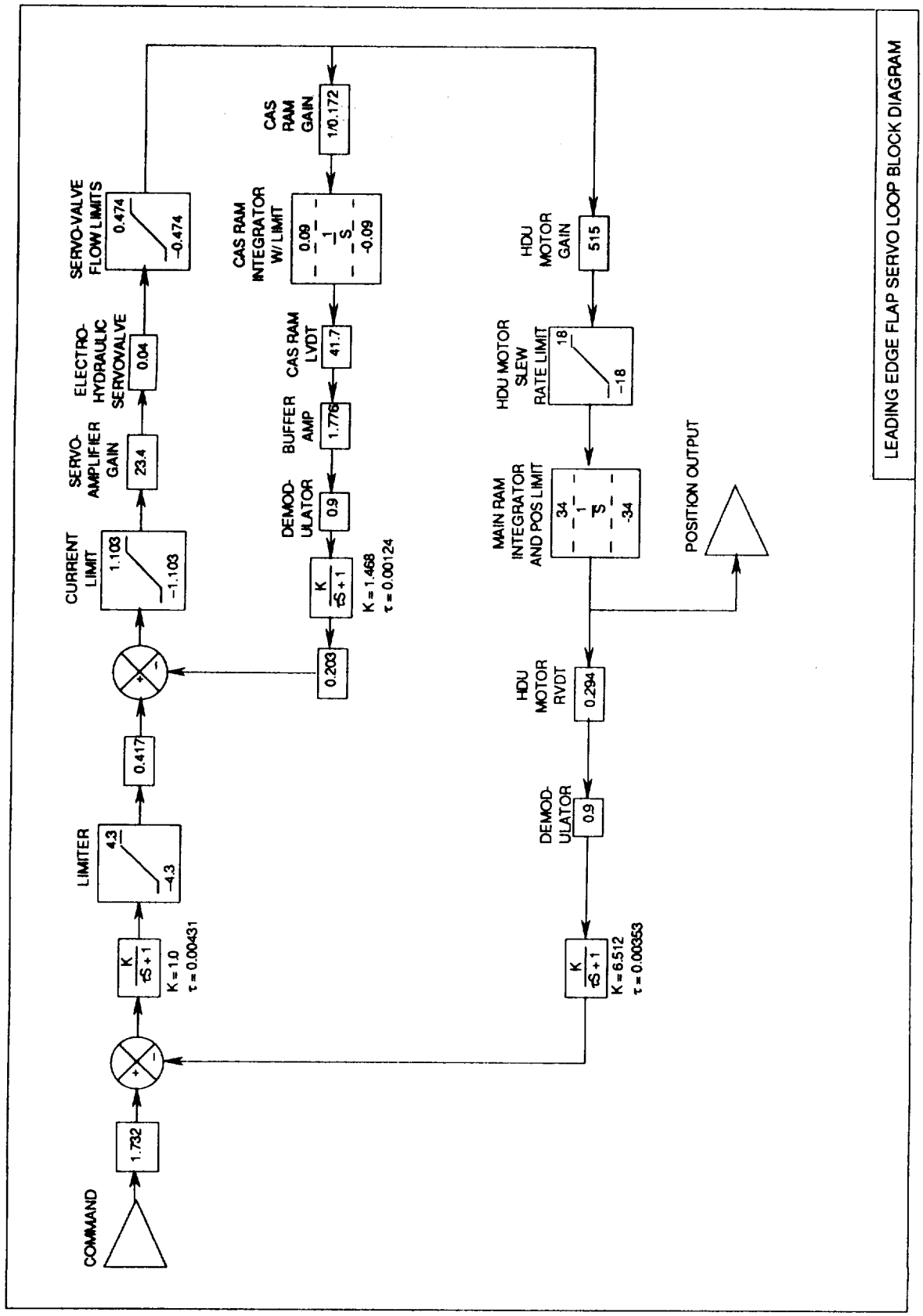


Figure 8.5. Leading-edge flaps Servoactuator Block Diagram

Table 8.1. Linear Stabilator Actuator Models - $\frac{\delta(s)}{\delta_c(s)}$

$-\frac{\left(\left(\frac{s}{81.7}\right)^2+2(.07)\left(\frac{s}{81.7}\right)+1\right)\left(\frac{s}{284}+1\right)\left(\frac{s}{808}+1\right)}{\left(\left(\frac{s}{37.7}\right)^2+2(.359)\left(\frac{s}{37.7}\right)+1\right)\left(\left(\frac{s}{143}\right)^2+2(.637)\left(\frac{s}{143}\right)+1\right)\left(\left(\frac{s}{351}\right)^2+2(.84)\left(\frac{s}{351}\right)+1\right)\left(\frac{s}{404}+1\right)\left(\frac{s}{1098}+1\right)}$	(a)
$\frac{\left(\left(\frac{s}{82.9}\right)^2+2(.068)\left(\frac{s}{82.9}\right)+1\right)}{\left(\left(\frac{s}{36.4}\right)^2+2(.41)\left(\frac{s}{36.4}\right)+1\right)\left(\left(\frac{s}{105.3}\right)^2+2(.59)\left(\frac{s}{105.3}\right)+1\right)}$	(b)
$\frac{1}{\left(\frac{s}{30}+1\right)}$	(c)

Table 8.2. Linear Rudder Actuator Models - $\frac{\delta(s)}{\delta_c(s)}$

$\frac{\left(\frac{s}{357}+1\right)}{\left(\frac{s}{61.9}+1\right)\left(\frac{s}{131}+1\right)\left(\left(\frac{s}{543}\right)^2+2(.997)\left(\frac{s}{543}\right)+1\right)}$	(a)
$\frac{1}{\left(\frac{s}{72.1}\right)^2+2(.69)\left(\frac{s}{72.1}\right)+1}$	(b)
$\frac{1}{\left(\frac{s}{40}\right)+1}$	(c)

Table 8.3. Linear Aileron Actuator Models - $\frac{\delta(s)}{\delta_c(s)}$

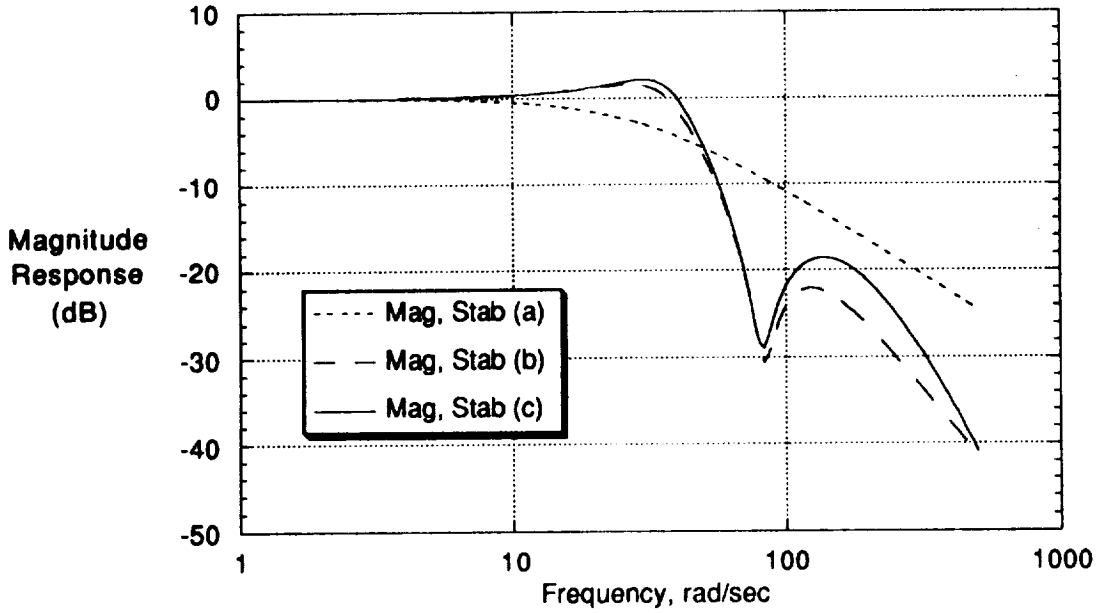
$\frac{.993\left(\frac{s}{356}+1\right)}{\left(\left(\frac{s}{103}\right)^2+2(.810)\left(\frac{s}{103}\right)+1\right)\left(\frac{s}{494}+1\right)\left(\left(\frac{s}{759}\right)^2+2(.789)\left(\frac{s}{759}\right)+1\right)}$	(a)
$\frac{1}{\left(\frac{s}{75}\right)^2+2(.59)\left(\frac{s}{75}\right)+1}$	(b)
$\frac{1}{\left(\frac{s}{48}\right)+1}$	(c)

Table 8.4. Linear Trailing-Edge Flap Actuator Models - $\frac{\delta(s)}{\delta_c(s)}$

$\frac{1.134\ (\frac{s}{284}+1)\ (\frac{s}{808}+1)}{(\frac{s}{33.4}+1)\ (\frac{s}{60.6}+1)\ ((\frac{s}{314})^2+2(.956)(\frac{s}{314})+1)\ (\frac{s}{447}+1)\ (\frac{s}{1045}+1)}$	(a)
$\frac{1}{(\frac{s}{35})^2+2(.71)(\frac{s}{35})+1}$	(b)
$\frac{1}{(\frac{s}{20})+1}$	(c)

Table 8.5. Linear Leading-Edge Flap Actuator Models - $\frac{\delta(s)}{\delta_c(s)}$

$\frac{(\frac{s}{283}+1)\ (\frac{s}{808}+1)}{(\frac{s}{34}+1)\ (\frac{s}{58.5}+1)\ ((\frac{s}{281})^2+2(.980)(\frac{s}{281})+1)\ (\frac{s}{680}+1)}$	(a)
$\frac{1}{(\frac{s}{26.9}+1)\ (\frac{s}{82.9}+1)}$	(b)
$\frac{1}{(\frac{s}{20})+1}$	(c)



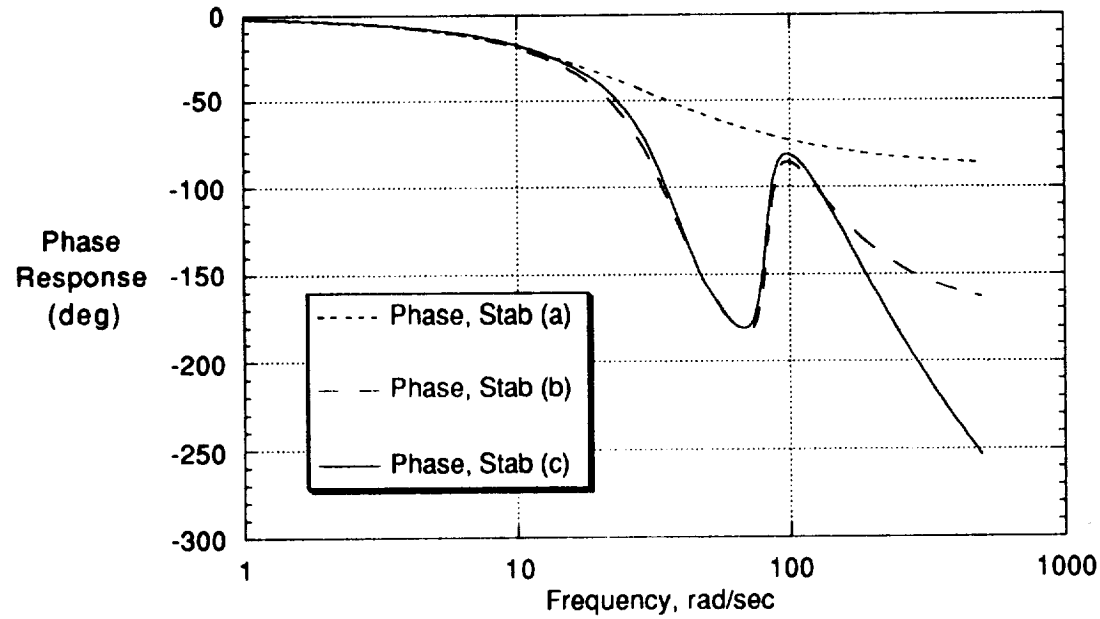


Figure 8.6. Gain and phase response of the (a), (b) and (c) transfer functions for the stabilator

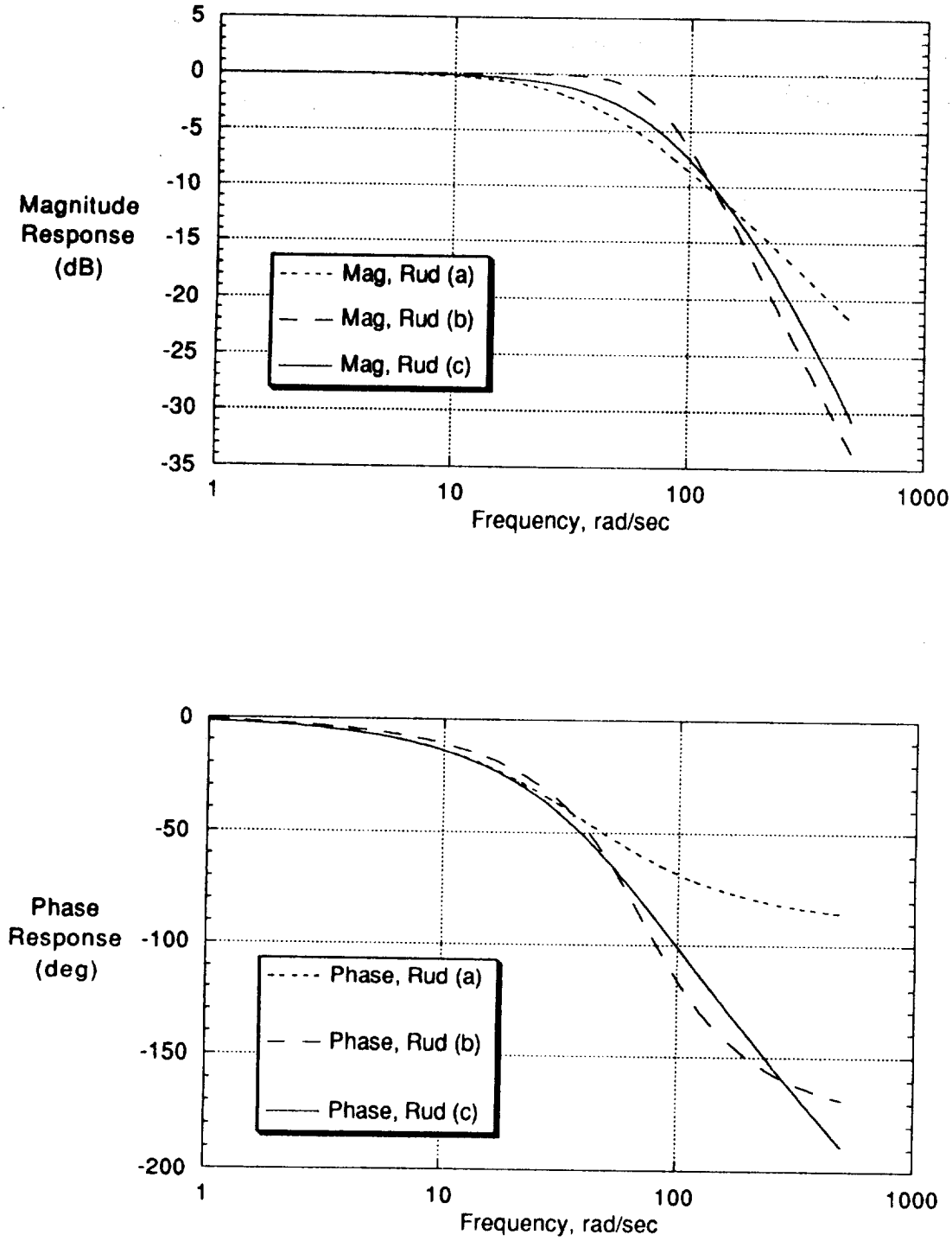


Figure 8.7. Gain and phase response of the (a), (b) and (c) transfer functions for the rudder

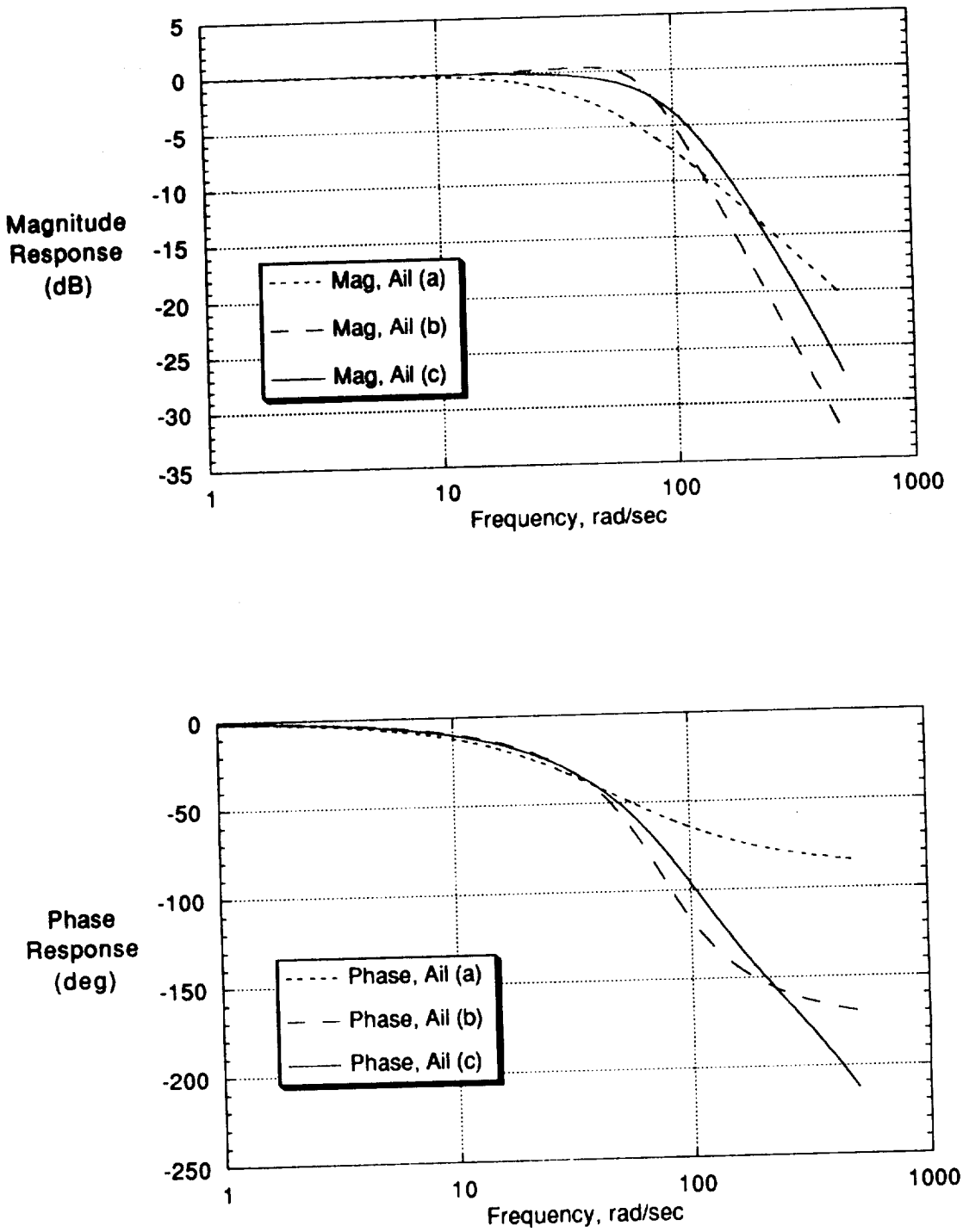


Figure 8.8. Gain and phase response of the (a), (b) and (c) transfer functions for the aileron

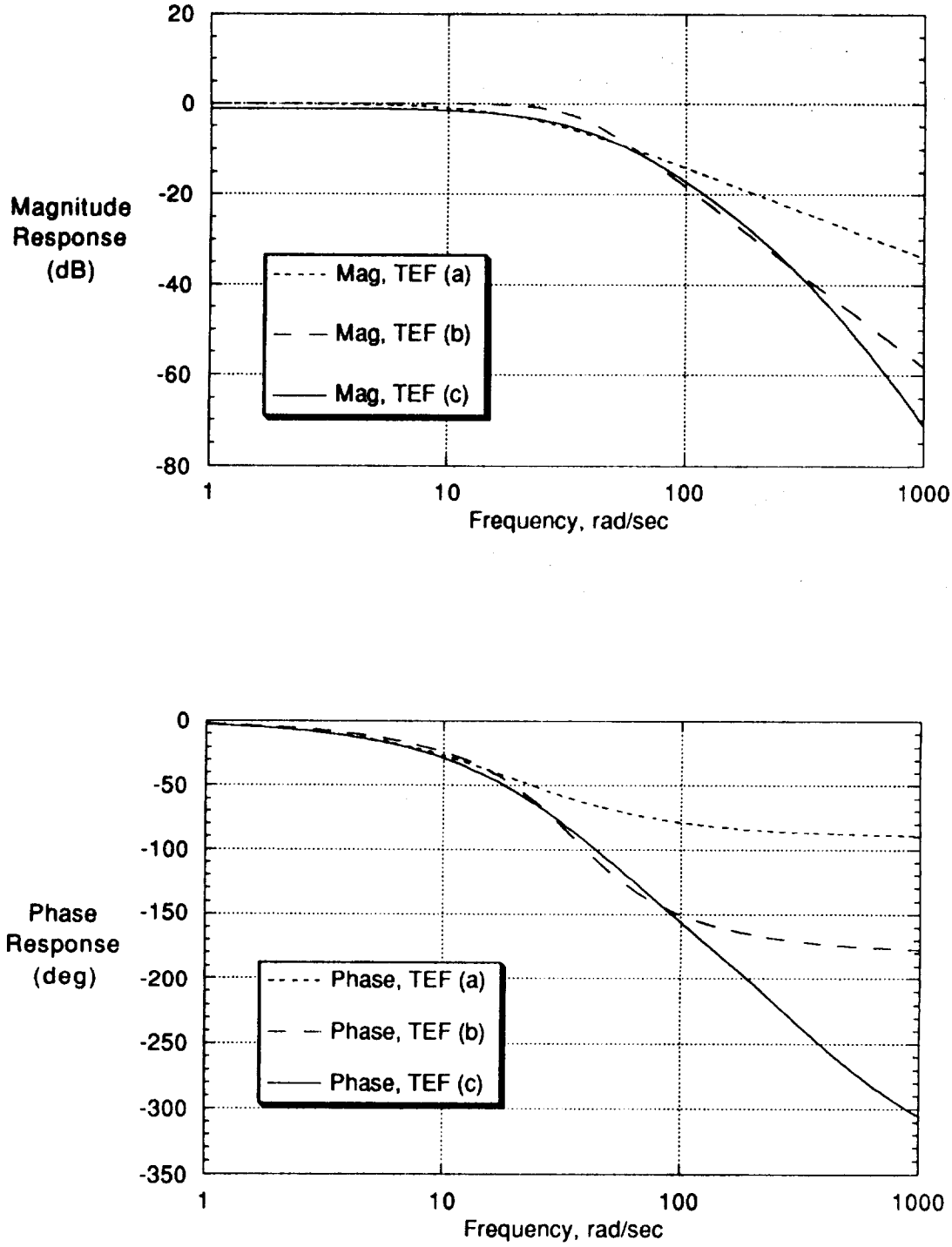


Figure 8.9. Gain and phase response of the (a), (b) and (c) transfer functions for the trailing-edge flaps

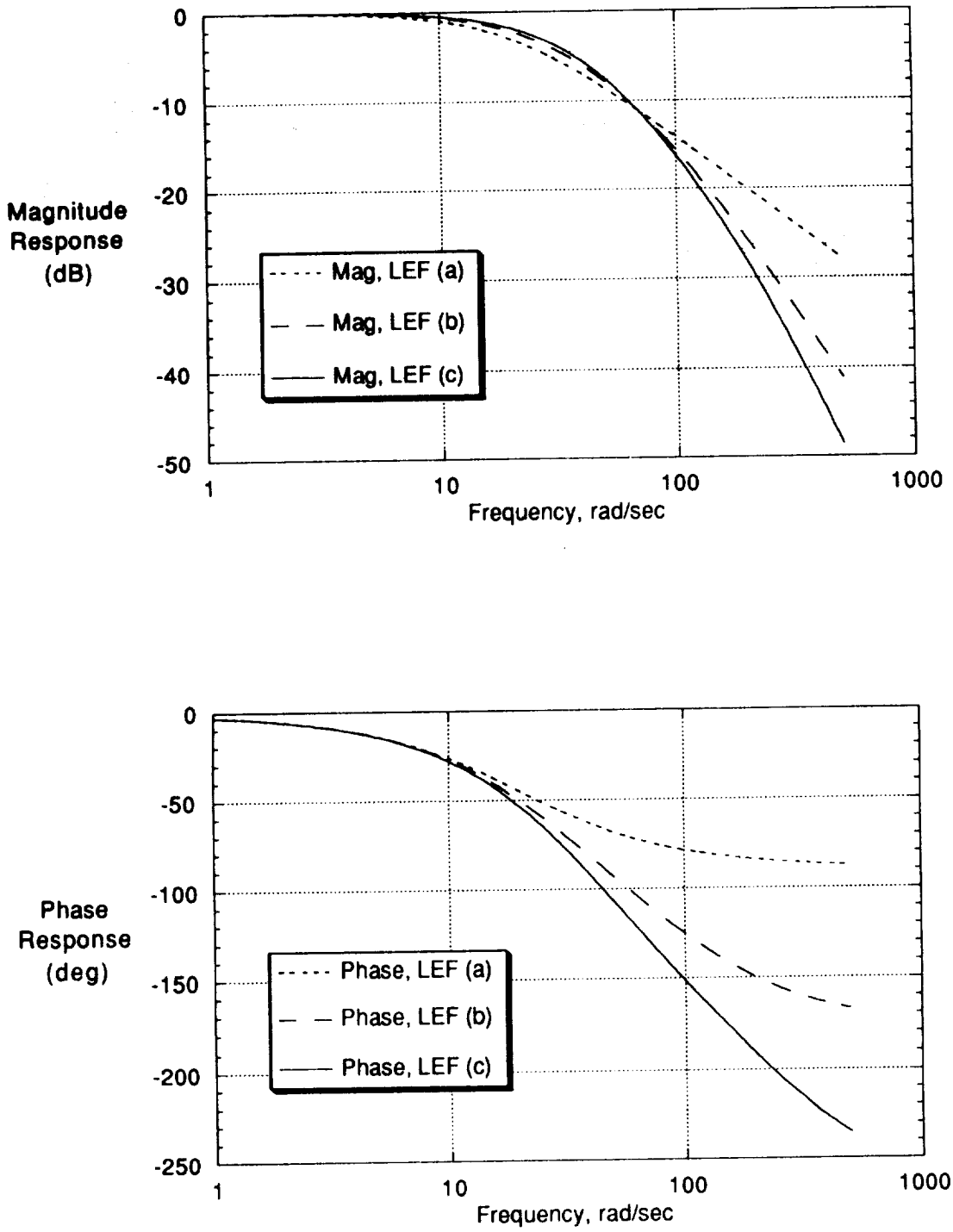


Figure 8.10. Gain and phase response of the (a), (b) and (c) transfer functions for the leading-edge flaps

8.2 Nonlinearities - Primary Controls

The significant nonlinearities in the actuators include a minimum resolution in the digital command due to 12 bit digital-to-analog conversion, hydromechanical thresholds, rate limiting, and nonlinear gearing. The first three of these types of nonlinearities are given in Table 6.2.1-1 of MDC A7813, vol II, and are repeated in Table 8.6 below.

Table 8.6. Actuator Nonlinearities

Actuator	Minimum Digital Command (degrees)	Threshold (fraction of full)	No Load Rate Limit (degrees/second)
Stabilator	0.0083	0.10	40
Rudder	0.0146	0.05	56
Aileron	0.022	0.05	100
Trailing-Edge Flaps	0.0127	0.05	18
Leading-Edge Flaps	0.0106	0.15	18

The main ram slewing rate limit is proportional to $(P_S - P_1)^{1/2}$, where P_S is the supply pressure (3000 PSI) and P_1 is the pressure on the resisting side of the piston due to external load. The maximum slewing rate occurs when P_1 is zero. When P_1 equals P_S, the actuator stalls.

In the ACSL simulation, of the possible nonlinear effects, only the no load rate limits are modeled. In both the MDC simulation and the DMS simulation, the no load rate limit used for the rudder actuators is 61 degrees/second instead of 56 degrees/second as shown in Table 8.6. Another set of specifications for actuator models can be found on page 12 of MDC report A8449 and is repeated below as Table 8.7. The no load rate limit for the rudder actuators is given as 61 degrees/second in Table 8.7. The 61 degree/second limit is used in the *f18bas* simulation.

Table 8.7. MDC F/A-18 Flight Hardware Integration Simulator

Actuator	Rate Limit	Position Limit	Mechanization
Stabilator	+/- 40°/s	-24°,+10.5°	1st order lag, τ = 1/30 s
Rudder	+/- 61°/s	-30,+30°	1st order lag, τ = 1/40 s
Aileron	+/- 100°/s	-25°,+45°	1st order lag, τ = 1/48 s
Trailing-Edge Flaps	+/- 18°/s	-8°,+45°	1st order lag, τ = 1/20 s
Leading-Edge Flaps	+/- 18°/s	-3°,+34°	1st order lag, τ = 1/20 s

For further discussion of gearing nonlinearities and plots of the effect of hinge moment on maximum actuator rate, the reader is referred to section 6.2 of MDC report A7813, vol II.

8.3 Speedbrake

The speedbrake system is described in section 8.7 of MDC A7813, vol I, Rev A. The speedbrake actuator is operated by a selector valve which ports hydraulic pressure to the main ram for extension or retraction. The selector valve is electrically controlled by a three position switch, OFF, RETRACT, and EXTEND, on the right hand throttle. The EXTEND position is a momentary. The OFF command holds the speedbrake in its current position. When the speedbrake is fully retracted, both "OFF" and "RETRACT" switch positions command the selector valve to the retract position.

The speedbrake rate is controlled by a flow regulator and restrictor. "The speedbrake extension time is limited to 2.5 seconds or approximately 20 degrees to 30 degrees per second (no load)" [MDC A7813, p8-14]. Maximum surface deflection is 60 degrees. In the *f18bas* simulation, the speedbrake no-load rate limit, SBNLRL, has been set to 60/2.5 = 24 °/s. No provision is made for reduced rate due to aerodynamic hinge moment. The maximum driving hinge moment available at 60 degrees open with the full 3000 psi of hydraulic pressure available is 296,000 in-lbs.

8.4 Thrust Vectoring Vane Actuators

The four actuators in the preliminary vane actuation system are each modeled with first order lag transfer functions with a time constant of 0.05 seconds, a rate limit of 80 °/sec, and position limits of +/- 30°. Again, this vane configuration of two vanes per nozzle is a preliminary version. The actual system implemented on the HARV F18 has three vanes per nozzle.

9. FLIGHT CONTROL SYSTEM (FCS)

The *f18bas* simulation implements the Control Augmentation System (CAS) of the inner loop F/A-18 Flight Control System defined by the 8.3.3 production programmable read-only memory (PROM) set. The theory of operation and design considerations of the CAS is described in section 9.1. The implementation issues associated with the *f18bas* simulation (as opposed to the actual aircraft) are discussed in section 9.2.

9.1 FCS Theory of Operation

The inner loop flight control law theory of operation for the F/A-18 is described in detail in section 16 of MDC report A7813, Vol I, rev A and section 1 of MDC report A4107, vol I, rev J. The inner loop control laws are divided into three channels: longitudinal, lateral, and directional. A preliminary thrust vectoring option (LTHVEC=.T.) exists in the code wherein additional pitch and yaw commands are generated by the control laws for use by the mixer/predictor. The mixer/predictor resolves the pitch and yaw command into actual vane commands for proposed thrust vectoring hardware. As indicated in section 1.2 of this report, the thrust vectoring option in the *f18bas* simulation represents a preliminary design version that has been superseded and will not be described in this document.

The inner loop CAS is a multi-rate digital controller. The actual flight computers update at a rate of 160 Hz. The inner loop calculations are performed at alternate frames of the flight control computer, so that the fastest inner loop of any of the controllers is 80 Hz. The details of the control law discretization can be found in MDC report A4107, vol I, rev J. All the sampling rates are fast enough that the continuous description offered here will suffice for analysis.

9.1.1 Longitudinal Auto Flap Up CAS

Three modes of leading-edge and trailing-edge flap management are performed in the pitch inner loop control laws of the F/A-18; (1) auto flap up (AFU), (2), half flaps, and (3) full flaps. The AFU mode is for up and away flight. The half and full flap settings are used in take-off and landing, respectively. In the *f18bas* simulation, only the auto flap up mode is supported. A block diagram of the pitch inner-loop control augmentation system in the AFU mode is presented in figure 9.1 (a-d). Figure 9.1 is a simplified version of "FIG 1 / OFP V10.1" on page 1-13 of MDC report A4107, vol I, rev J. Paths in "FIG 1 / OFP V10.1" having to do with outer loops, the direct electric link (DEL) reversion mode, and HALF and FULL flap management modes have been deleted. Also omitted from figure 9.1 are blocks defining stick dynamics, trim functions, and the g-limiter logic. The definition of the pitch gains (PKxx) in terms of pitch functions (FUNxx) are given in table 9.1. Table 9.2 defines the longitudinal filters used in figure 9.1. The pitch functions required to define quantities in both tables 9.1 and 9.2 are defined in figure 9.2 (a-o). The following quote is taken from pages 1-2 through 1-4 of MDC report A4107, Vol I, REV. J. Italics are comments inserted by the authors. Grammatical hyphens have been added.

> Primary control of the F/A-18 aircraft longitudinal motions is accomplished through the CAS. The Direct Electric Link (DEL) and mechanical system provide backup for the pitch CAS. Management of the leading and trailing edge flaps is provided through the use of a cockpit switch which calls for either AUTO, HALF, or FULL flaps. A block diagram of the longitudinal control system is presented in Figure 1 (*figure 9.1 of this document*). Longitudinal functions and logic diagrams are given in Figure 1.1 through 1.44 (*figure 9.2 of this document*). Longitudinal digital filter characteristics are summarized in Table 1-1 (*Table 9.2 of this document*).

Auto Flap Up (CAS On) - The pitch CAS uses a pilot commanded longitudinal stick position input as a command to the CAS. The forward path gain is air data scheduled (Function 32A) to yield a uniform initial pitch acceleration response for sharp inputs. The CAS feedback parameters are a blend of air data scheduled pitch rate (Functions 40 and 68), normal acceleration, and angle of attack. Pitch rate and normal acceleration feedbacks give improved pitch dynamic characteristics and load factor control in the mid to high dynamic pressure portion of the flight envelope. Air combat maneuvering characteristics and increased stick-force-per-g cues in the low to mid dynamic pressure flight regime are provided by the air data scheduled pitch rate feedback. Angle-of-attack feedback provides additional increased stick force cues for low speed high angle-of-attack air combat maneuvering. Roll rate multiplied by yaw rate is fed to the longitudinal control system (Function 107) to reduce the effects of inertial coupling.

The pilot commanded stick position, pitch rate, normal acceleration, and angle-of-attack feedback signals are iterated at the rates shown in Figure 1 (*figure 9.1 of this document*).

The pitch CAS forward patch is comprised of an air data scheduled gain (Function 32A), a forward loop integrator, and the stabilator servo and power cylinder. The aircraft response (feedback) is compared to the maneuver command; the resulting difference drives the servo/power cylinder to reduce the maneuver error to zero through the forward loop integrator. Consequently, the aircraft is automatically kept in steady state trim (hands off) since any aircraft normal acceleration different from the commanded normal acceleration is reduced to zero by the action of the integrator.

The notch structural filter located in the feedback paths attenuates any undesirable motions which are sensed by the motion sensors due to aeroelastic bending of the aircraft structure. The longitudinal forward loop gain (Functions 32A) and pitch rate feedback gains (Functions 40 and 68) are scheduled with air data. A supersonic (air data scheduled time constant, Function 22) lead-lag filter is employed in the pitch rate feedback path. The forward loop integrator gain is air data scheduled (Function 12) and angle-of-attack activated (AOA less than 33 degrees). This implementation is necessary to maintain adequate loop gain margins, phase margins, and handling qualities throughout the flight envelope. Scheduling pitch rate feedback gain with air data results in increased feedback gain with decreasing dynamic pressure or increasing altitude. This increases the stick force cues for mid to low speed flight conditions and provides a pitch attitude type of control law for tracking since pitch rate feedback is nulled via the forward loop integrator.

The stall margin network introduces an aircraft nose-down command proportional to angle of attack (AOA) above 22 degrees AOA. The effect of the stall margin portion of the control laws is to increase the stick force cues for high angle-of-attack maneuvering.

A speedbrake-to-stabilator interconnect (Function 100) is incorporated to alleviate g transients for speedbrake extension/retraction. The interconnect command is on/off in response to speedbrake solenoid valve command. The magnitude of the input command is air data scheduled. The interconnect command is lagged through a first order low-pass filter with the time constant scheduled with air data and as a function of speedbrake extension or retraction. The speedbrake extend time constant is less than the speedbrake retract time constant.

Flap Control - Auto Flap UP - With the flap switch in AUTO, the leading and trailing edge maneuvering flaps are scheduled with angle of attack and air data (Functions 24 and 27) to optimize performance, to improve the high angle of attack characteristics, and provide load alleviation at elevated g's. Flap position limits are computed as a function of air data inputs (Functions 25, 28 and 29) to prevent the application of excessive air loads on the flaps.

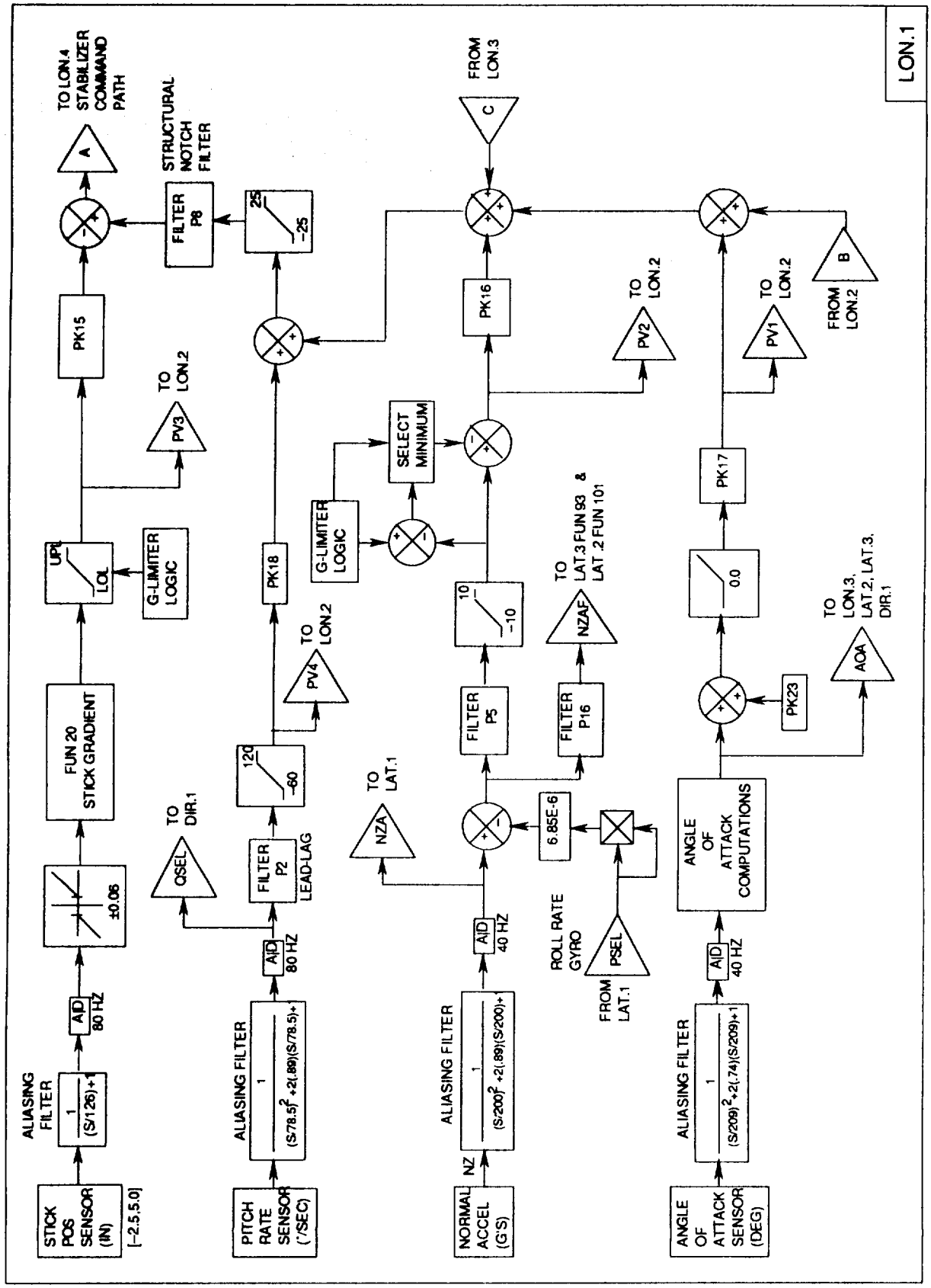


Figure 9.1. Longitudinal CAS - (a) Main feedback paths

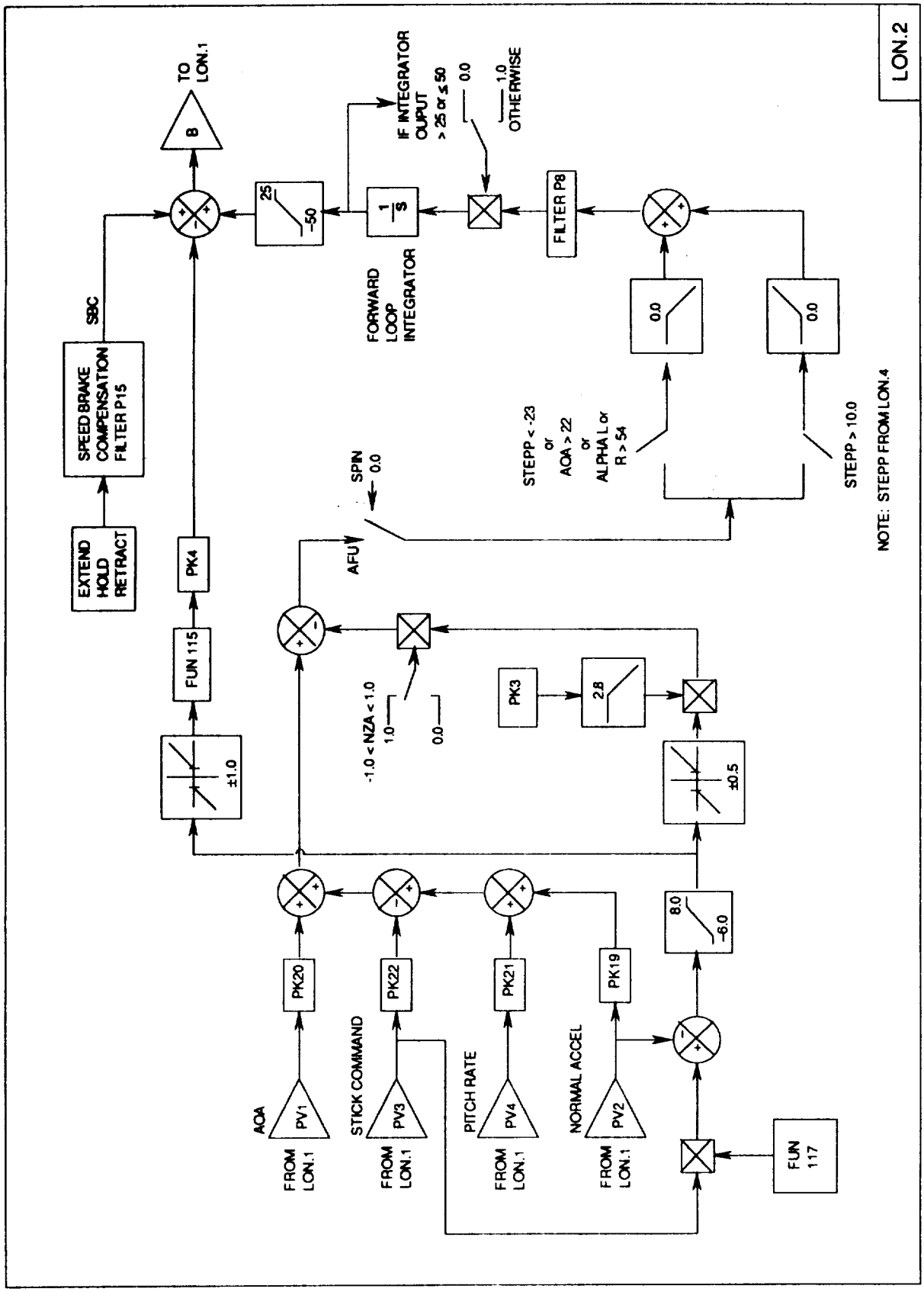


Figure 9.1. Continued - (b) Feedforward integrator

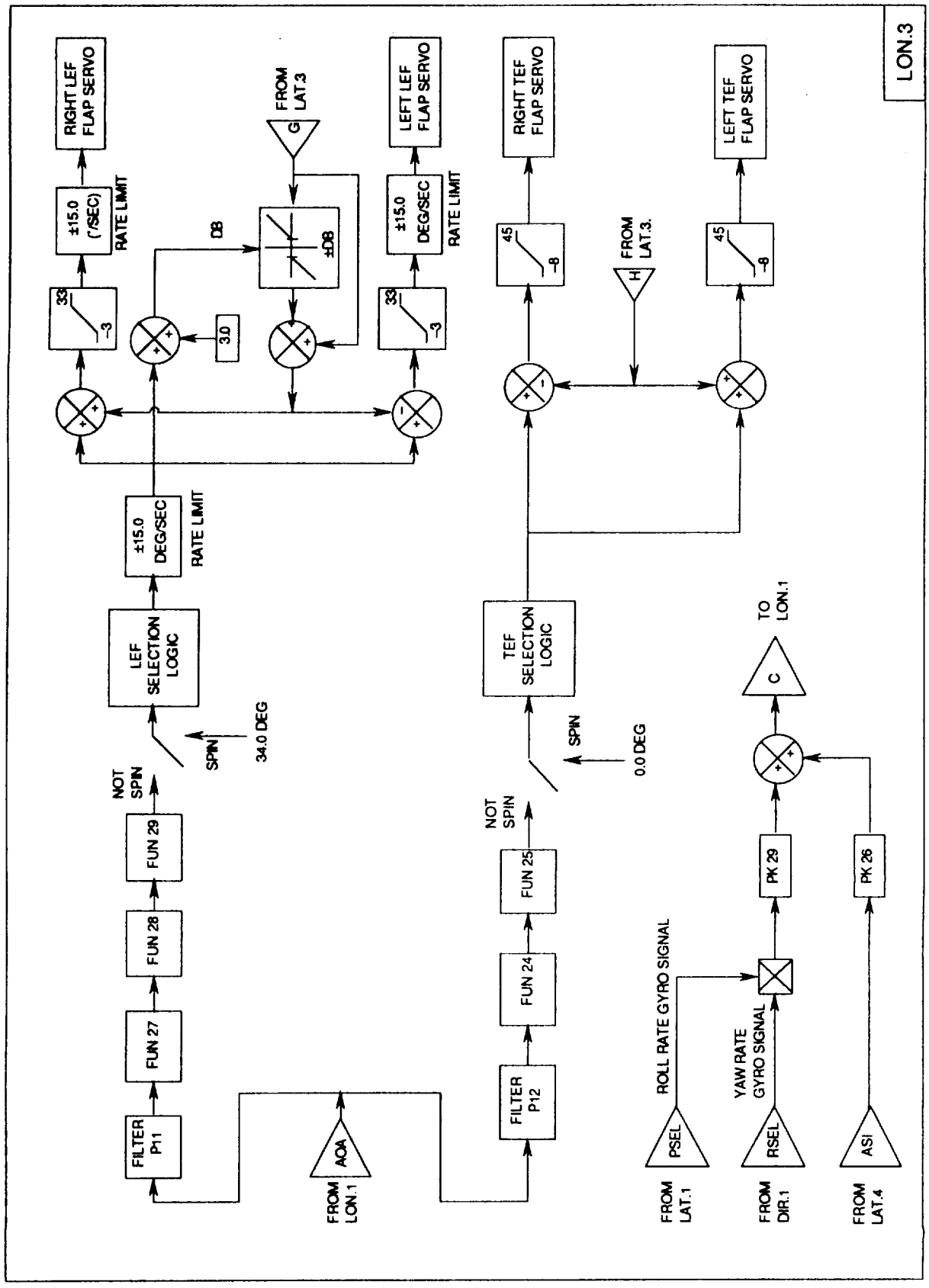


Figure 9.1. Continued - (c) Flap command and inertial compensation logic

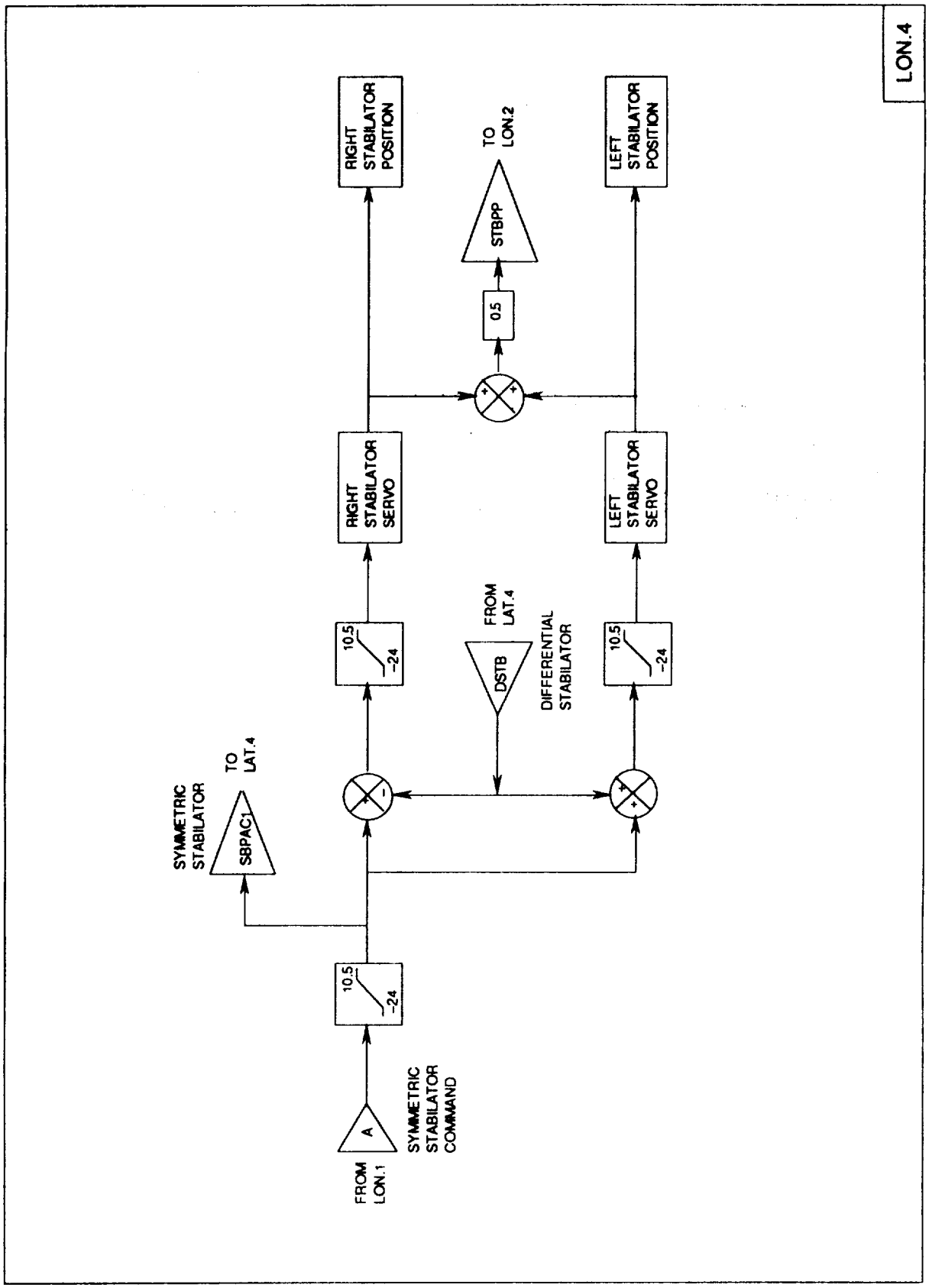


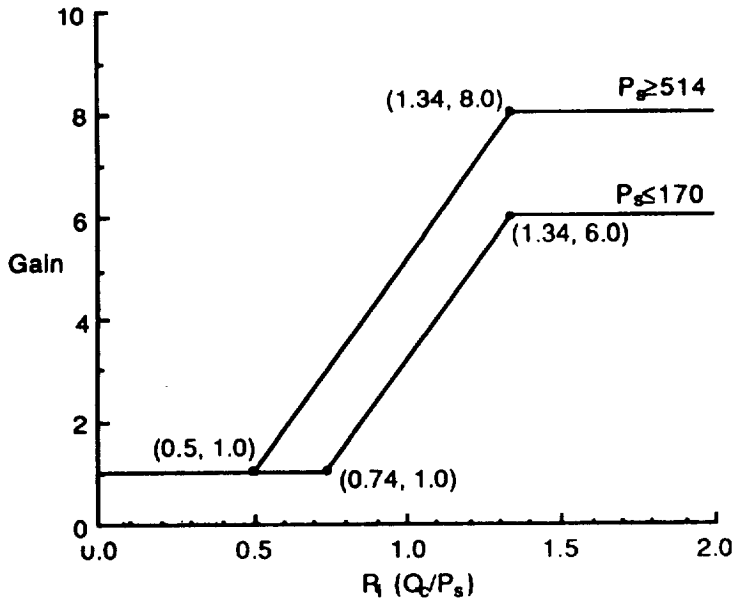
Figure 9.1. Concluded - (d) Stabilator command synthesis

Table 9.1. Pitch gains in 8.3.3 inner loop CAS

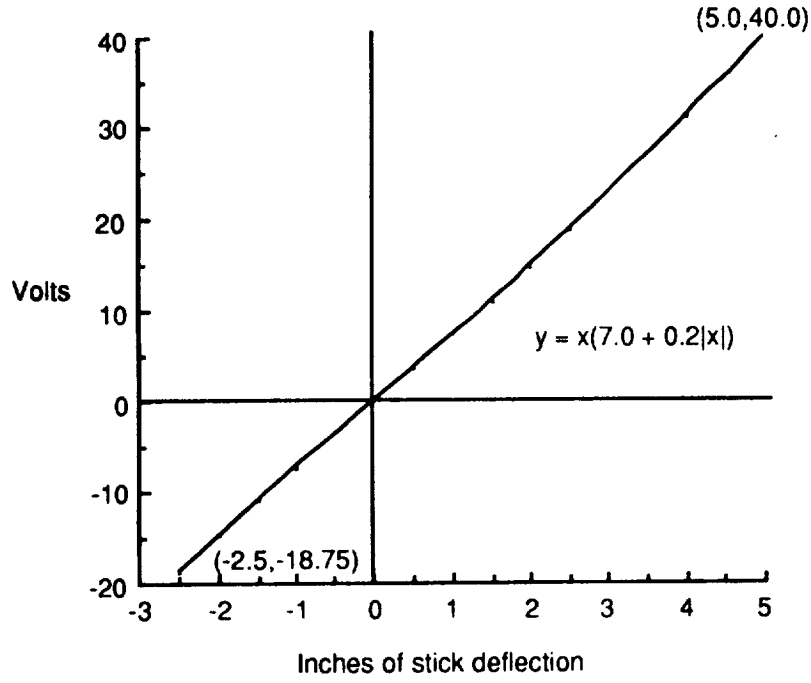
Function	AFU Mode	Spin Mode
PK3	F12*F32A + F116	0.0
PK4	1.5*F32A	0.0
PK15	F32A	0.7
PK16	3.5*F32A	0.0
PK17	F32A	0.0
PK18	F68*F32A + F40	0.0
PK19	8.5*F32A + F12	0.0
PK20	1.0	0.0
PK21	F68*F32A + F12	0.0
PK22	F32A*F12	0.0
PK23	-22.0	NA by virtue of PK17
PK26	0.0	0.1
PK29	F107	0.0

Table 9.2. Pitch filters in 8.3.3 inner loop CAS

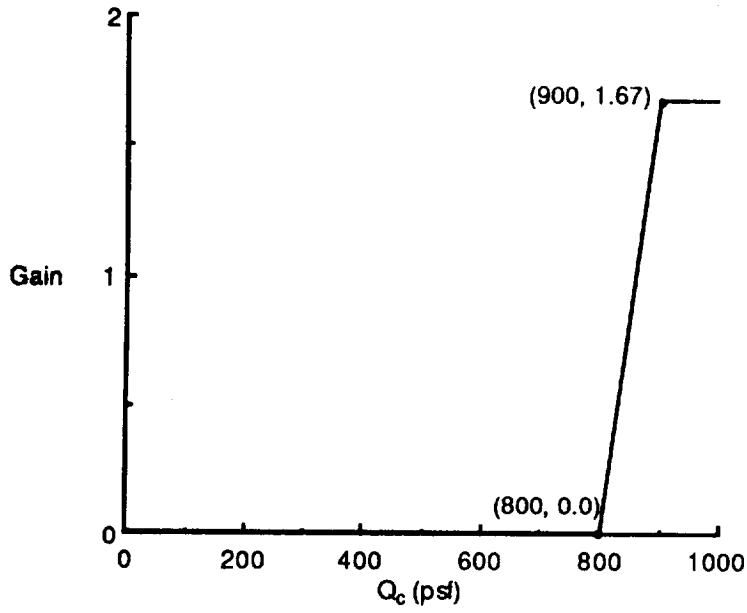
Filter	Transfer Function		Where Used
P2	$\frac{\tau_N s + 1}{\tau_D s + 1}$	$\tau_N = .015(1+FUN22)$ $\tau_D = .015$	Pitch Rate
P5	$\frac{1}{\tau_D s + 1}$	$\tau_D = 0.04$	Normal Acceleration
P8	$\frac{(s/\omega)^2 + 2\zeta_N(s/\omega) + 1}{(s/\omega)^2 + 2\zeta_D(s/\omega) + 1}$	$\zeta_N = 0.03$ $\zeta_D = 0.70$ $\omega = 60.0$ (r/s)	Notch
P11	$\frac{1}{\tau_D s + 1}$	$\tau_D = 0.39$	Leading-Edge Flaps
P12	$\frac{1}{\tau_D s + 1}$	$\tau_D = 0.79$	Trailing-Edge Flaps
P15	$\frac{1}{\tau_D s + 1}$	$\tau_D = 0.60$ for extension $0.49 < \tau_D < 2.0$ for retraction	Speed brake
P16	$\frac{1}{\tau_D s + 1}$	$\tau_D = 0.174$	NZAF



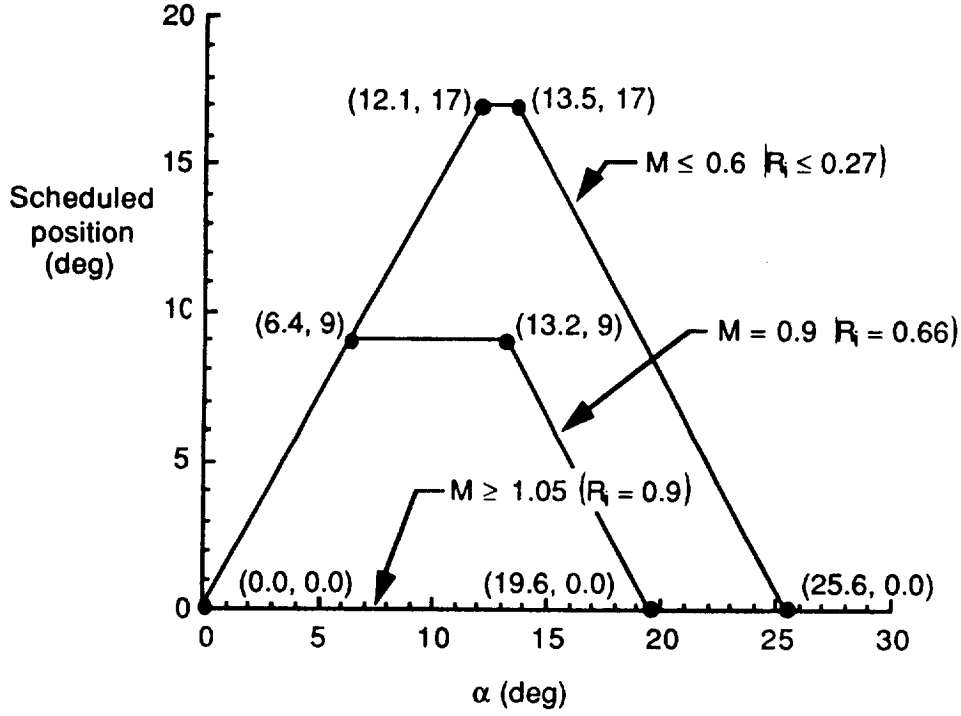
(a) Pitch function 12 - Pitch forward loop integrator gain schedule
Auto Flap Up (R_i, P_S)



(b) Pitch function 20 - Pitch stick gradient
Figure 9.2 Pitch CAS Functions

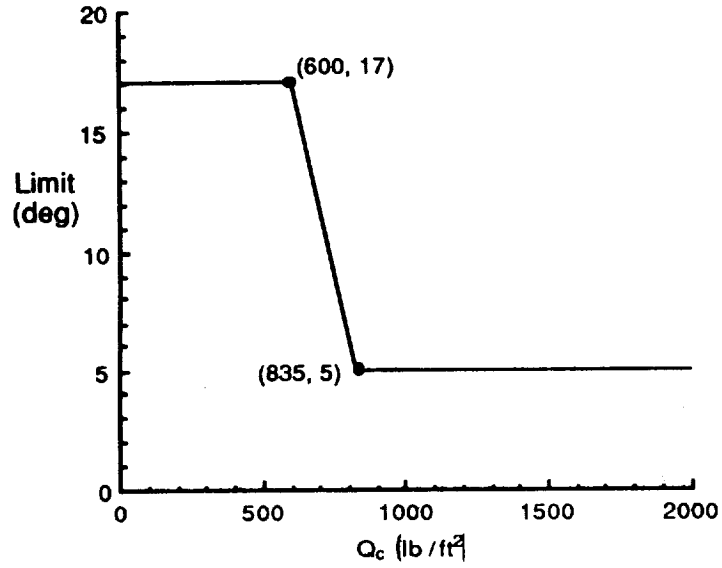


(c) Pitch function 22 - Fader on the supersonic compensation
Auto Flap Up (α, R_i)

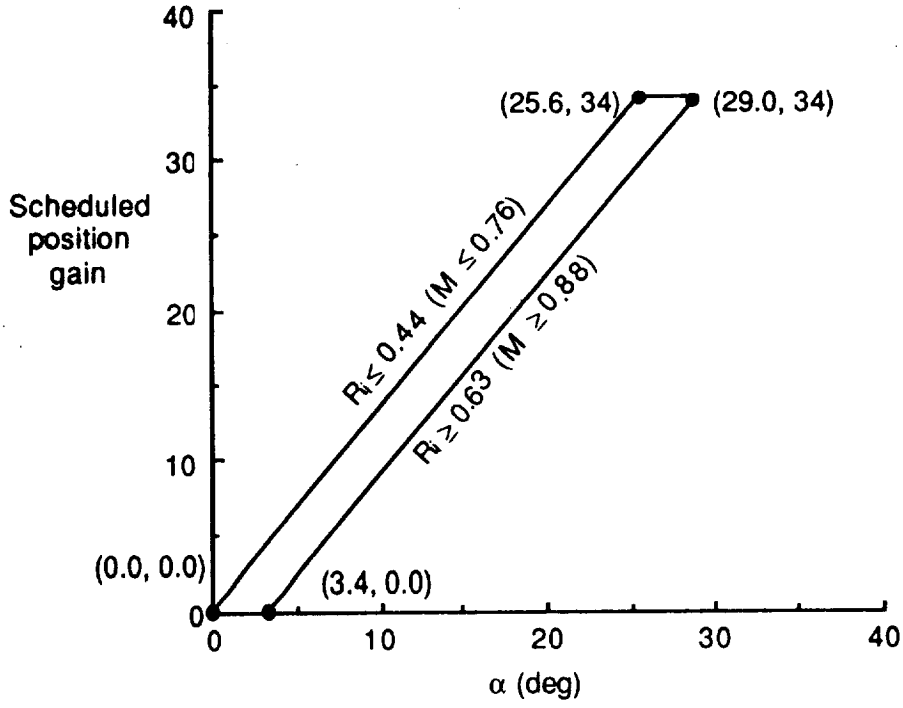


(d) Pitch function 24 - Trailing-edge flap schedule
Auto Flap Up (Q_C)

Figure 9.2 Continued.

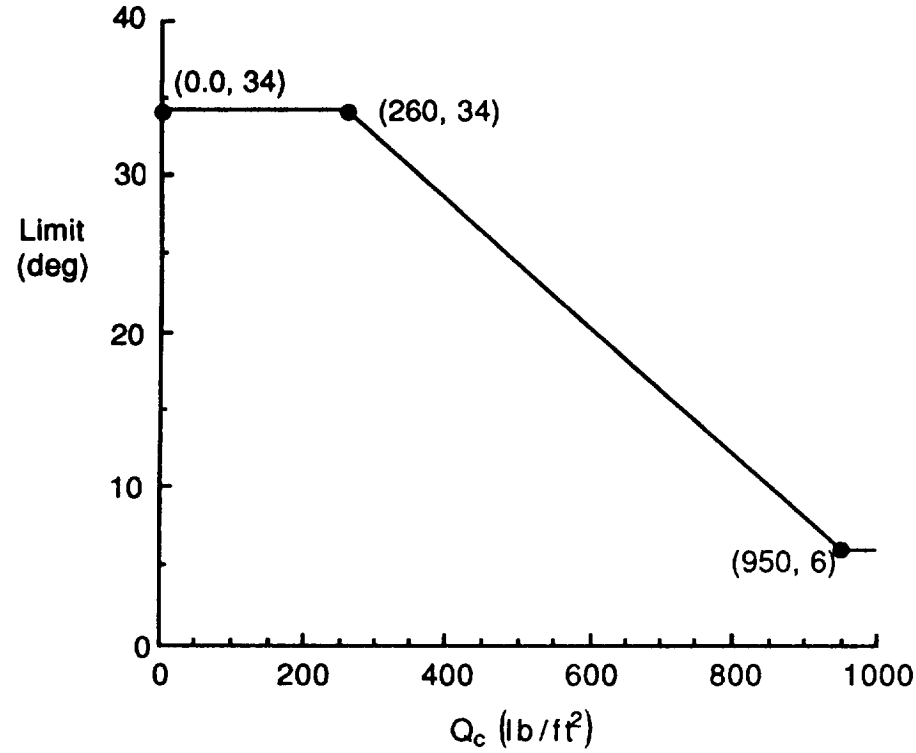


(e) Pitch function 25 - Trailing-edge flap schedule
Auto Flap Up (Q_C)

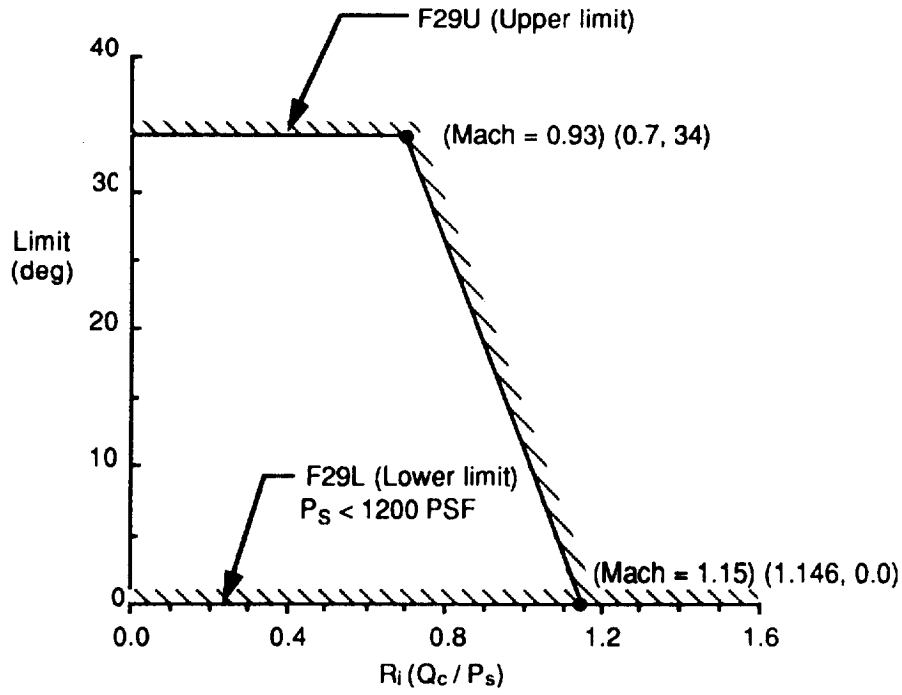


(f) Pitch function 27 - Leading-edge flap schedule
Auto Flap Up (α, R_i)

Figure 9.2 Continued.

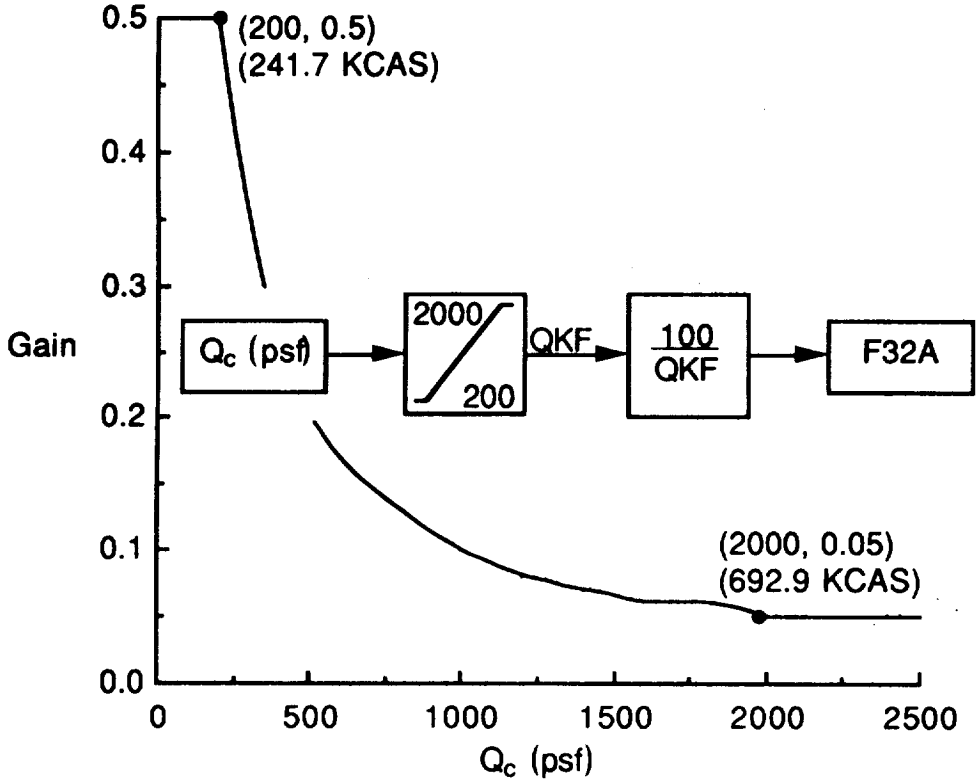


(g) Pitch function 28 - Leading-edge flap schedule
Auto Flap Up (Q_c)

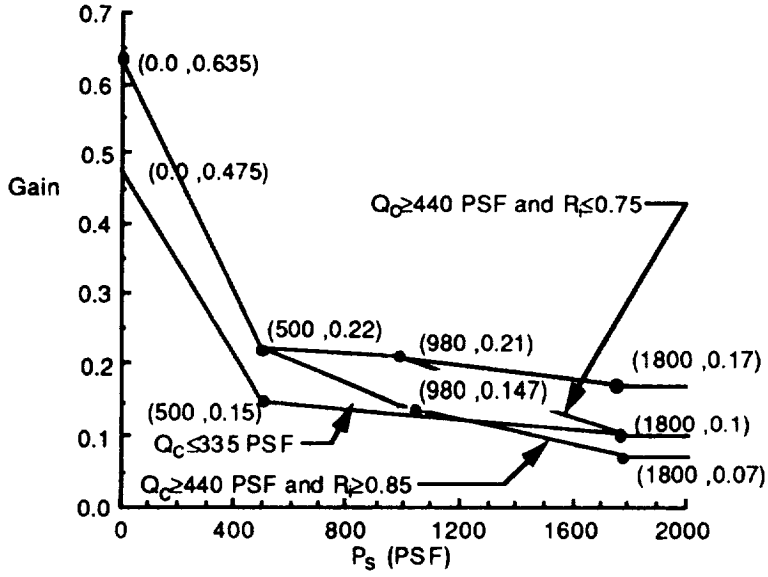


(h) Pitch function 29 - Leading-edge flap schedule
Auto Flap Up (R_i)

Figure 9.2 Continued.



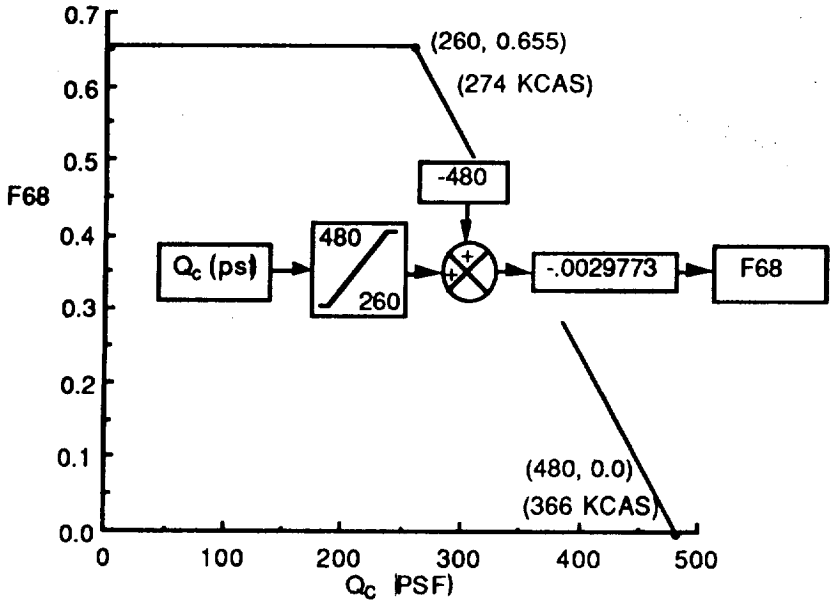
(i) Pitch function 32A - Longitudinal forward loop gain schedule
Auto Flap Up (Q_C)



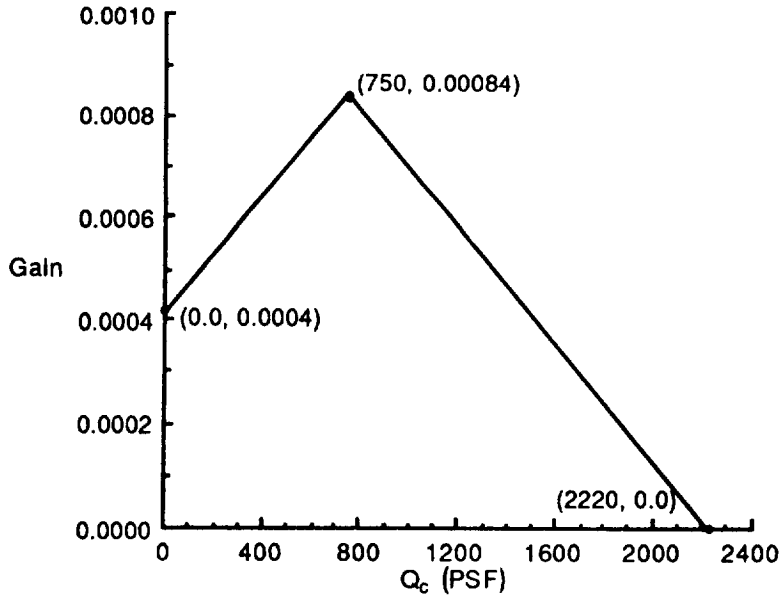
(j) Pitch function 40 - Pitch rate feedback gain schedule
Auto Flap Up (P_S, Q_C, R_i, STORES)

Figure 9.2 Continued.

C-2

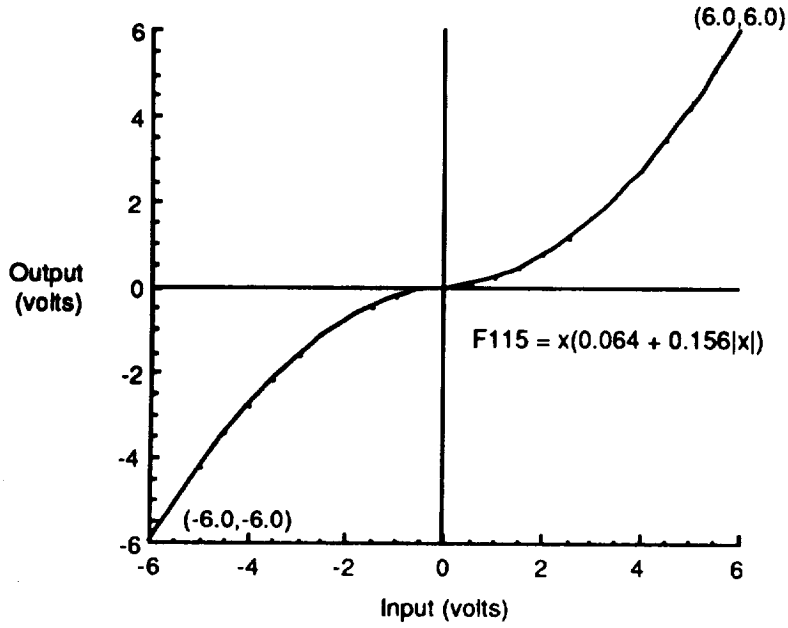


(k) Pitch function 68 - Pitch rate feedback gain schedule
Auto Flap Up (Q_C)

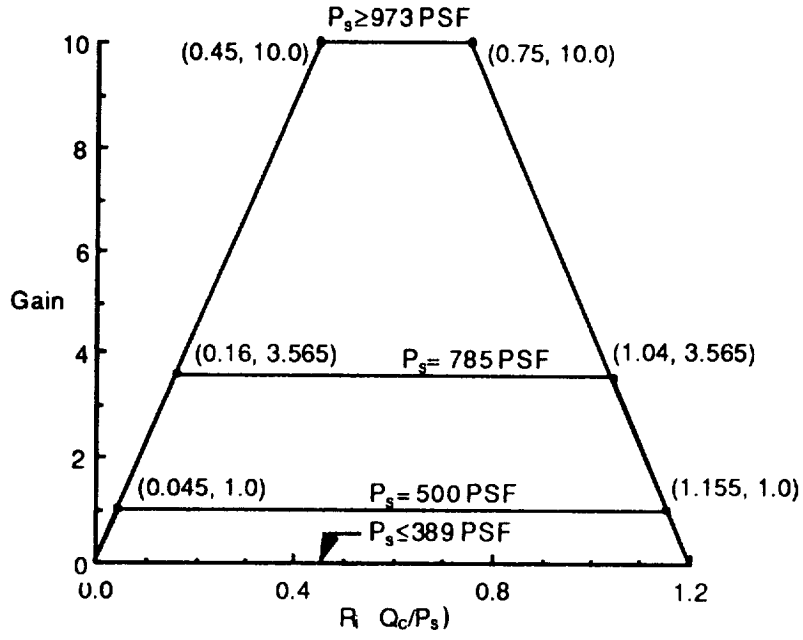


(l) Pitch function 107 - Longitudinal inertial gain schedule
Auto Flap Up (Q_C)

Figure 9.2 Continued.

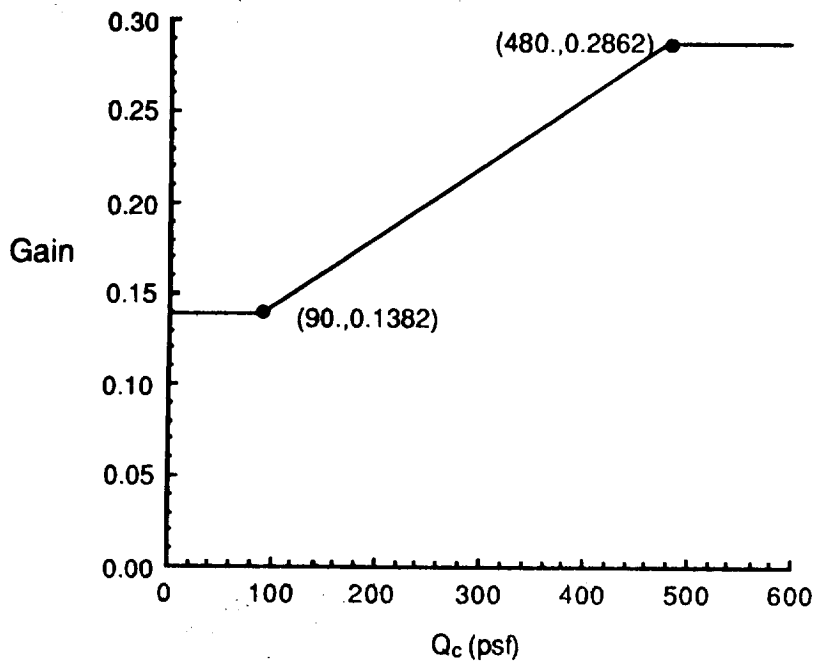


(m) Pitch function 115 - Longitudinal nonlinear stick gradient
Auto Flap Up (Q_C)



(n) Pitch function 116 - Pitch nonlinear integrator gain schedule
Auto Flap Up (R_i, P_S,)

Figure 9.2 Continued.



(o) Pitch function 117 - Nonlinear stick command gain
Auto Flap Up (Q_C)

Figure 9.2 Concluded.

9.1.2 Lateral Auto Flap Up CAS

Figure 9.3 (a-d) is a simplified version of the diagram "Figure 2 / OFP V10.1" on page 1-69 of MDC report A4107, Vol I, REV. J. Paths having to do with outer loops and the direct electric link (DEL) reversion mode have been deleted. The definition of the lateral gains (RKxx) in terms of roll functions (FUNxx) are given in Table 9.3. Table 9.4 defines the lateral filters used in figure 9.3. The roll functions required to define quantities in both tables 9.3 and 9.4 are defined in figure 9.4 (a-l). The following quote is from pages 1-65, 1-97 of MDC report A4107, Vol I, REV. J. Italics are comments inserted by the authors. Grammatical hyphens have been added.

> Lateral control is accomplished by a control-by-wire system which commands the ailerons, differential stabilators, differential trailing-edge flaps, and differential leading-edge flaps. The controls consist of Control Augmentation System (CAS) and Digital and Analog Direct Electric Link (DEL) functions. A block diagram of the lateral control system is shown in Figure 2 (*Figure 9.3 of this report*). Lateral functions and logic diagram are presented in Figure 2.1 through 2.20 (*Figure 9.4 of this report*). Lateral digital filter characteristics are summarized in Table 1-3 (*Table 9.4 of this report*).
>
> The roll CAS augments the pilot's roll stick inputs and provides roll damping through the aileron, differential stabilator, differential trailing-edge flaps, and differential leading-edge flaps. The center stick position command signal is summed with the roll rate feedback (p) to provide the closed loop control of the ailerons, differential stabilators, differential trailing-edge flaps, and differential leading-edge flaps. Because of the inherent high roll damping of the aircraft at low altitude high speed flight conditions, the roll rate feedback gain is reduced (gain scheduled) in this area. A limited authority roll rate feedback is used to improve trim stability in the area where the CAS roll rate feedback gain is reduced.
>
> The control stick position sensor signal, after it is shaped by a parabolic gradient, provides the desired roll response of the augmented aircraft.
>
> The roll CAS normal mode command path is comprised of an air data scheduled gain (Functions 4, 7, and 13). Function 4 is also the roll rate feedback gain schedule. The gain varies with dynamic and static pressure to provide acceptable loop stability and roll response characteristics throughout the envelope. Notch filters located in the roll rate feedback path attenuate motion sensor inputs due to aeroelastic bending of the aircraft structure. A notch filter is located in the stick position signal to prevent structural mode coupling through the stick position sensor. The frequencies are based on modes determined during Ground Vibration Testing. An additional fourth order roll-off filter is located in the feedback path to provide additional attenuation of the structural mode inputs and high frequency noise, the presence of which was detected during Iron Bird testing.
>
> With wing stores, the roll rate feedback gain is increased for roll rates in excess of 110 degrees per second and the stick command gain (Function 13) is modified in order to limit the maximum roll rate to approximately 150 degrees per second. Without wing stores, the roll rate feedback gain is increased above 190 degrees per second to aid in limiting the maximum roll rate to approximately 220 degrees per second. *(Wing stores are not an option in the f18bas simulation).*
>
> The maximum roll rate capability with or without stores is limited to 150 degrees per second above 5 g's normal load factor to reduce the vertical tail loads and reduce inertial coupling.

The aileron command gain is scheduled with angle of attack and air data (Functions 35 and 36). The angle-of-attack gain schedule eliminates excessive adverse sideslip during rolling maneuvers for high angles of attack at low dynamic pressure flight conditions. The air data gain schedule reduces the aileron command to zero at high dynamic pressure flight conditions where wing flexibility reverses the effective aileron aerodynamic moments. The air data schedule also avoids hinge moment limits. The aileron command is also limited by an angle of attack and air data schedule (Function 41) to optimize roll coordination.

The roll CAS signal which drives the differential stabilator is multiplied by a gain and then is limited by a gain schedule (Functions 6 and 101) and rate limited at 35 degrees per second to reduce the coupling between the roll and pitch axis at high dynamic pressure where the stabilator approaches hinge moment and bending moment limits. Stabilator bending moment alleviation is provided via a differential stabilator surface command gain scheduled with load factor (Function 101). The differential stabilator command is also limited by the stabilator pitch command (pitch priority) and by an angle-of-attack schedule (Function 41). The aileron and differential stabilator command signals also command rudder (for roll coordination) through the rolling surface-to-rudder interconnect (RSRI). The RSRI is gain scheduled with air data and angle of attack (Functions 30 and 38).

The differential trailing-edge flap command gain is scheduled with air data and angle of attack (Functions 31 and 34). The air data schedule permits the maximum differential deflection available without encountering hinge moment limiting or excessive vertical tail loads. The angle-of-attack schedule reduces the available deflection above 5 degrees AOA and eliminates it above 10 degrees AOA. Differential flaps are not necessary to meet roll performance requirements above these angles of attack.

The differential leading-edge flaps are gain scheduled with air data and load factor (Function 93) to provide improved roll rate in the low to mid altitude transonic speed range. The gain is reduced for increasing load factor, increasing altitude, and decreasing Mach in the transonic speed range.

Differential ailerons are used for active suppression of a wing stores aeroelastic oscillation. The forward sensor assembly lateral accelerometer signal is filtered to extract the oscillation component, which then drives the ailerons. The active suppression system is switched on by an air data schedule if the heavy wing stores discrete is set by the Mission Computer.

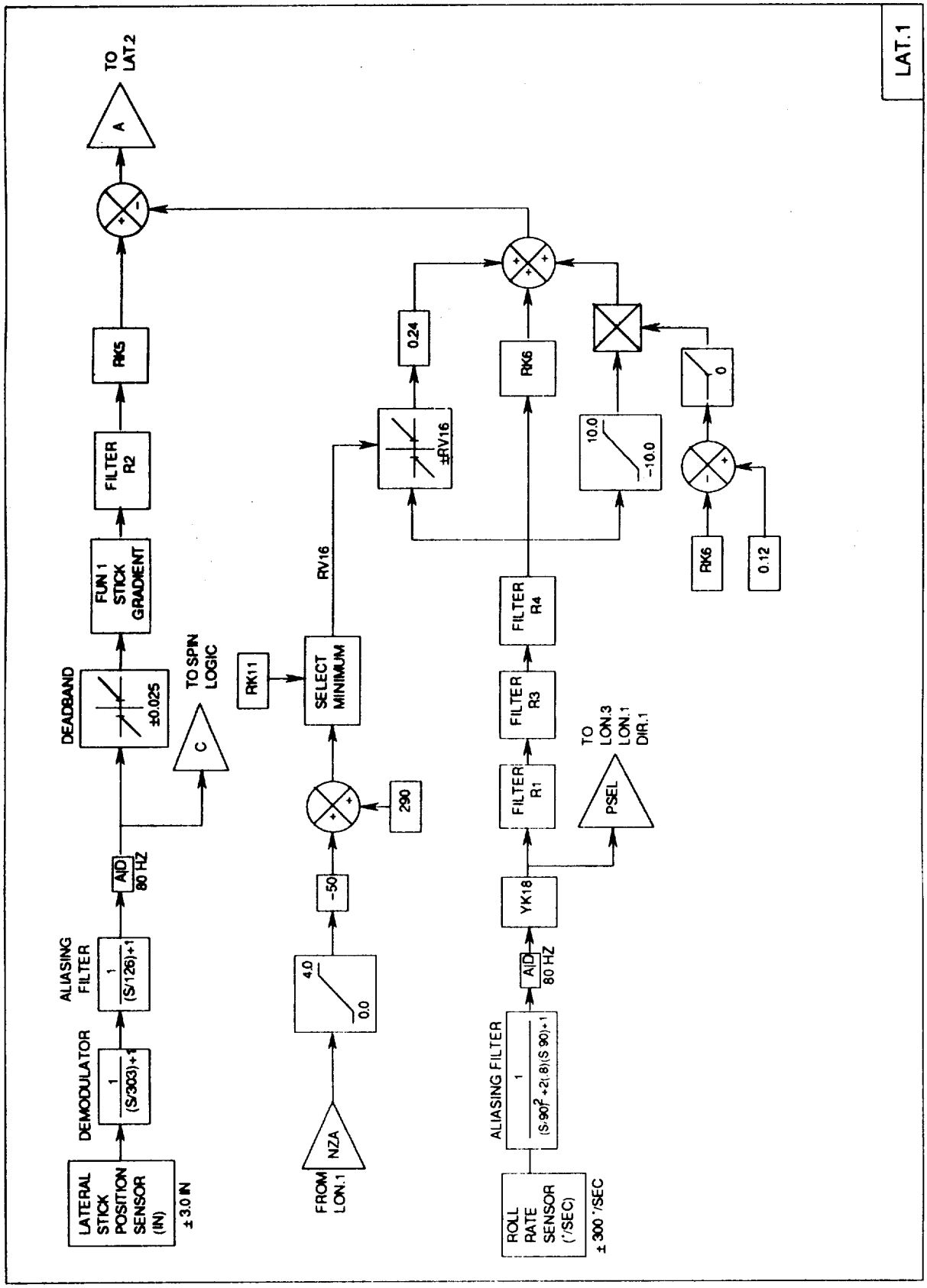


Figure 9.3. Lateral CAS - (a) Main feedback paths

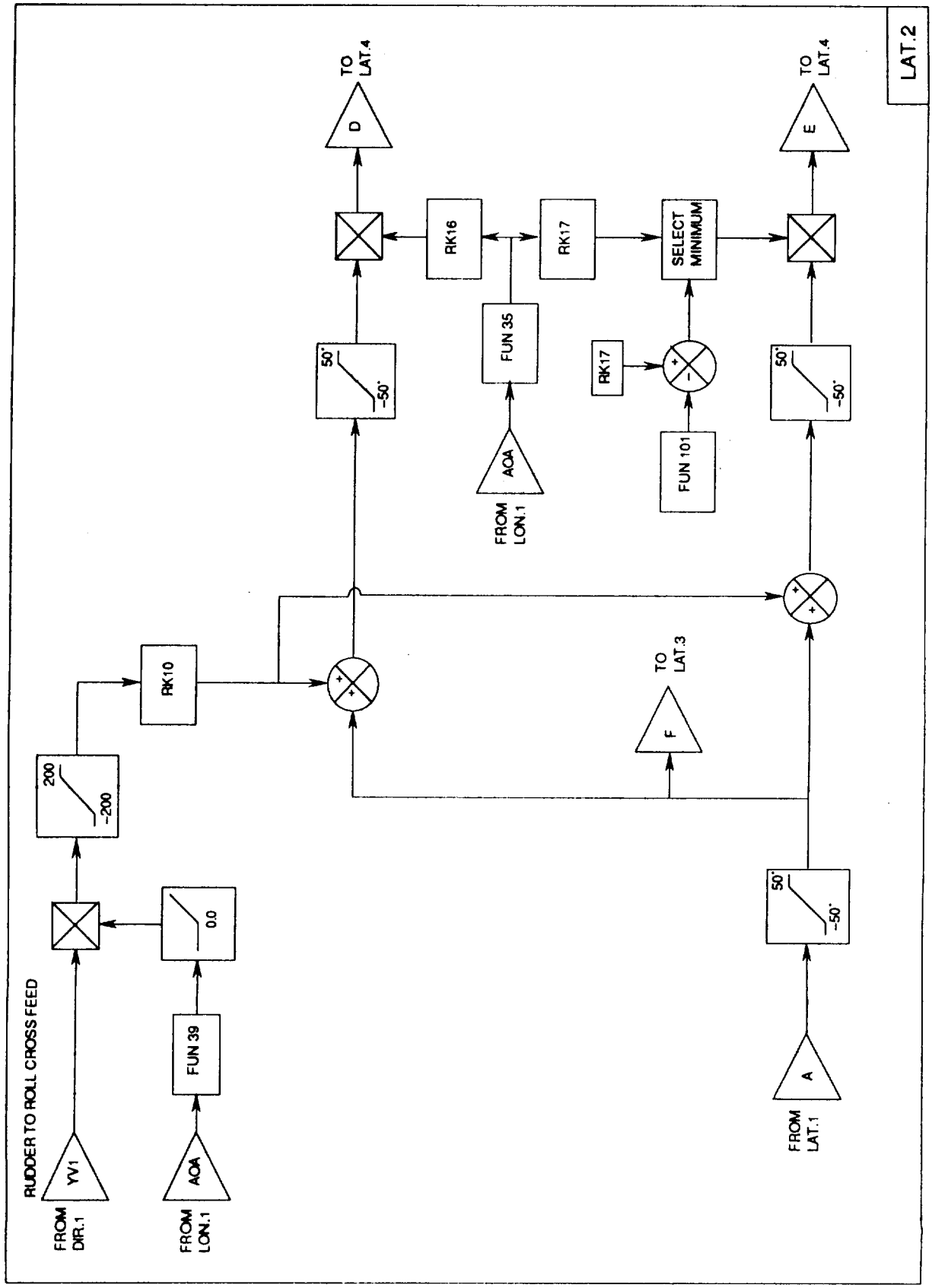


Figure 9.3. Continued - (b) Aileron and differential stabilator commands

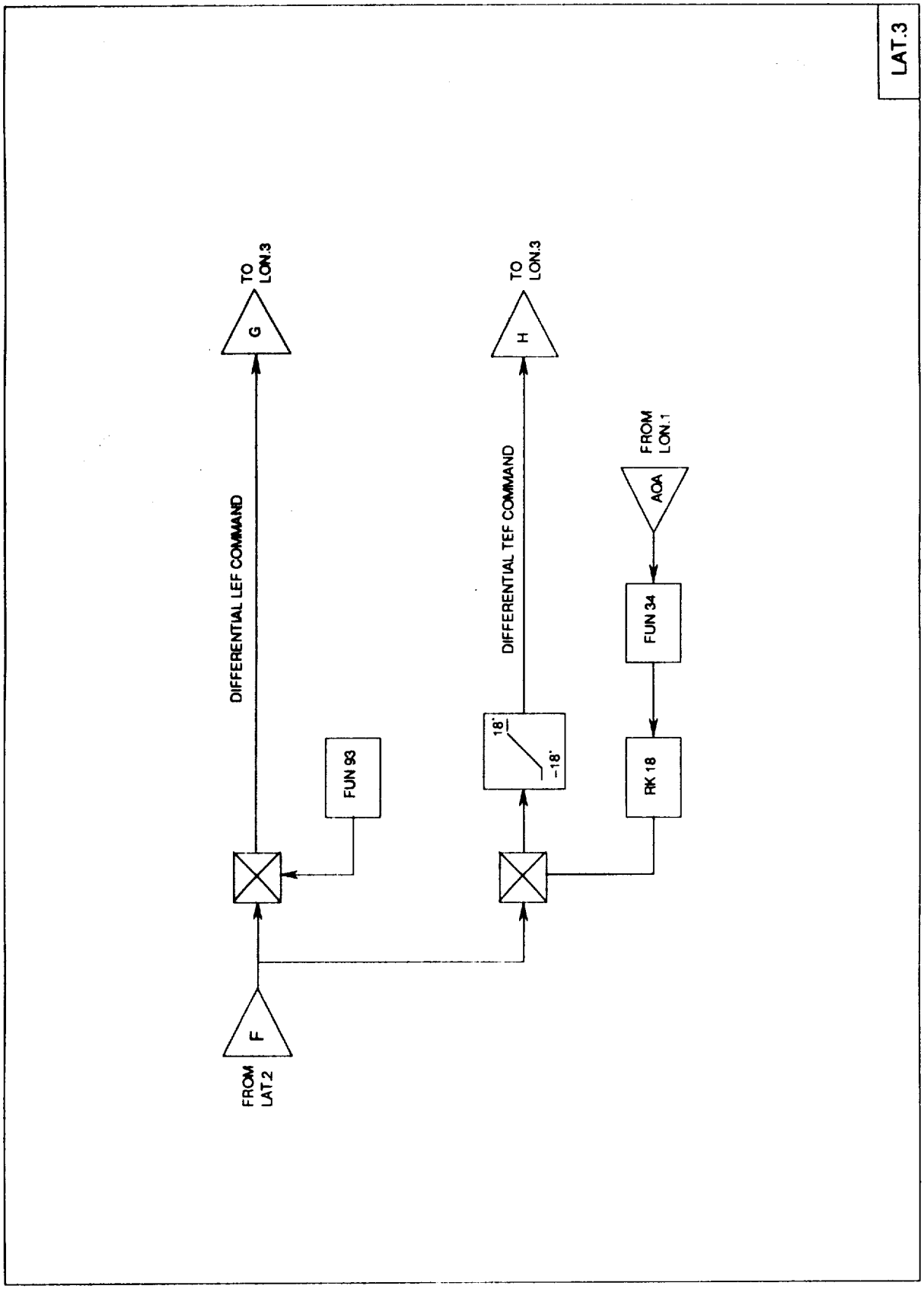


Figure 9.3. Continued - (c) Differential flaps

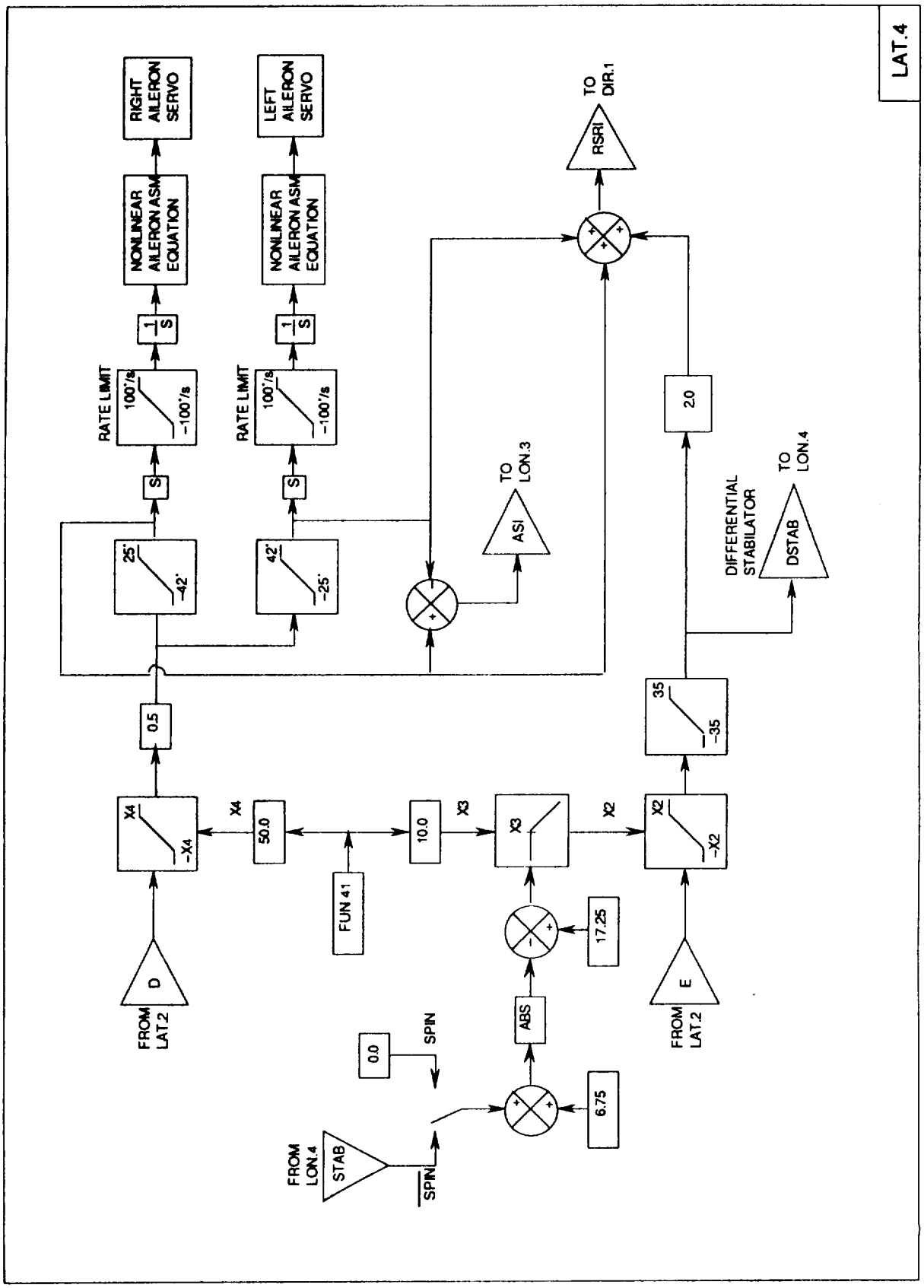


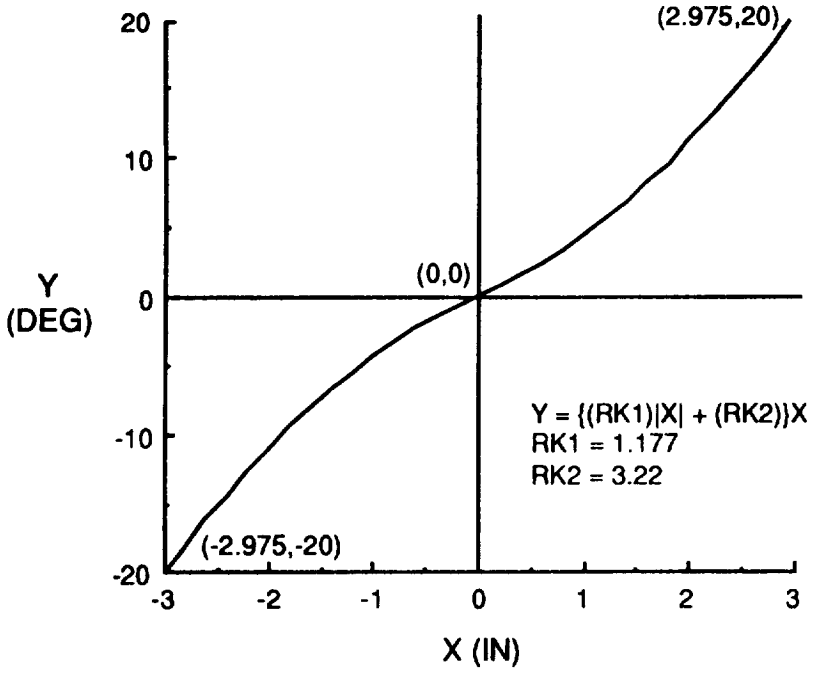
Figure 9.3. Concluded - (d) Servo command synthesis

Table 9.3. Lateral (roll) gains in 8.3.3 inner loop CAS

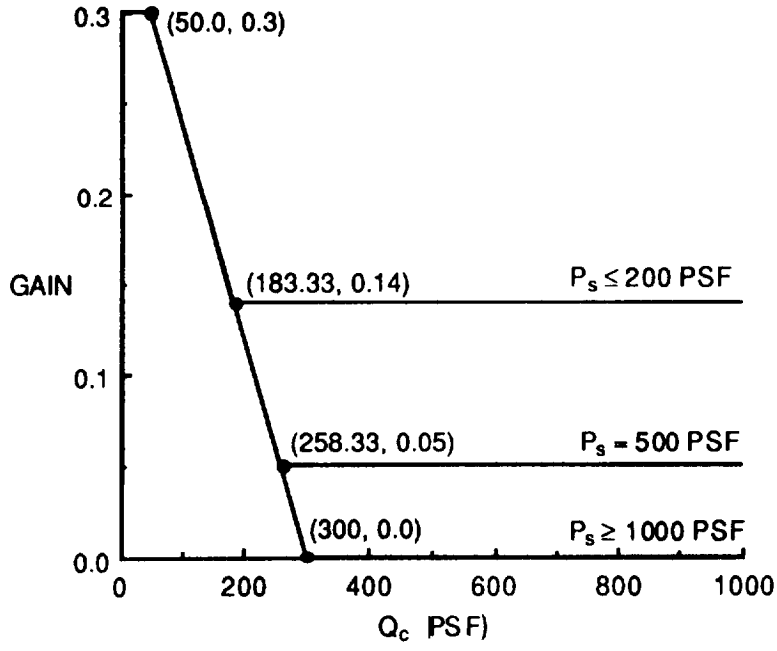
Function	AFU Mode	Spin Mode
RK1	1.117	0.0
RK2	3.22	6.732
RK5	F7*(F4*F13)	2.5
RK6	F4	0.0
RK10	1.33	0.0
RK11	1.0	300.0
RK16	F36	F36
RK17	F6	F6
RK18	F31	0.0
RK22	0.0	0.0
YK18	See directional axis, Table 9.5	

Table 9.4. Lateral (roll) filters in 8.3.3 inner loop CAS

Filter	Transfer Function		Where Used
R1	$\frac{1}{(s/\omega)^2 + 2\zeta(s/\omega) + 1}$	$\zeta = 0.80$ $\omega = 90.0$ (r/s)	2nd order rolloff for roll rate sensor
R2	$\frac{(s/\omega)^2 + 2\zeta_N(s/\omega) + 1}{(s/\omega)^2 + 2\zeta_D(s/\omega) + 1}$	$\zeta_N = 0.05$ $\zeta_D = 0.50$ $\omega = 31.0$ (r/s)	5 Hz notch for stick lateral position sensor
R3	$\frac{(s/\omega)^2 + 2\zeta_N(s/\omega) + 1}{(s/\omega)^2 + 2\zeta_D(s/\omega) + 1}$	$\zeta_N = 0.07$ $\zeta_D = 0.70$ $\omega = 44.0$ (r/s)	7 Hz notch for roll rate sensor
R4	$\frac{(s/\omega)^2 + 2\zeta_N(s/\omega) + 1}{(s/\omega)^2 + 2\zeta_D(s/\omega) + 1}$	$\zeta_N = 0.03$ $\zeta_D = 0.70$ $\omega = 107.0$ (r/s)	17 Hz notch for roll rate sensor

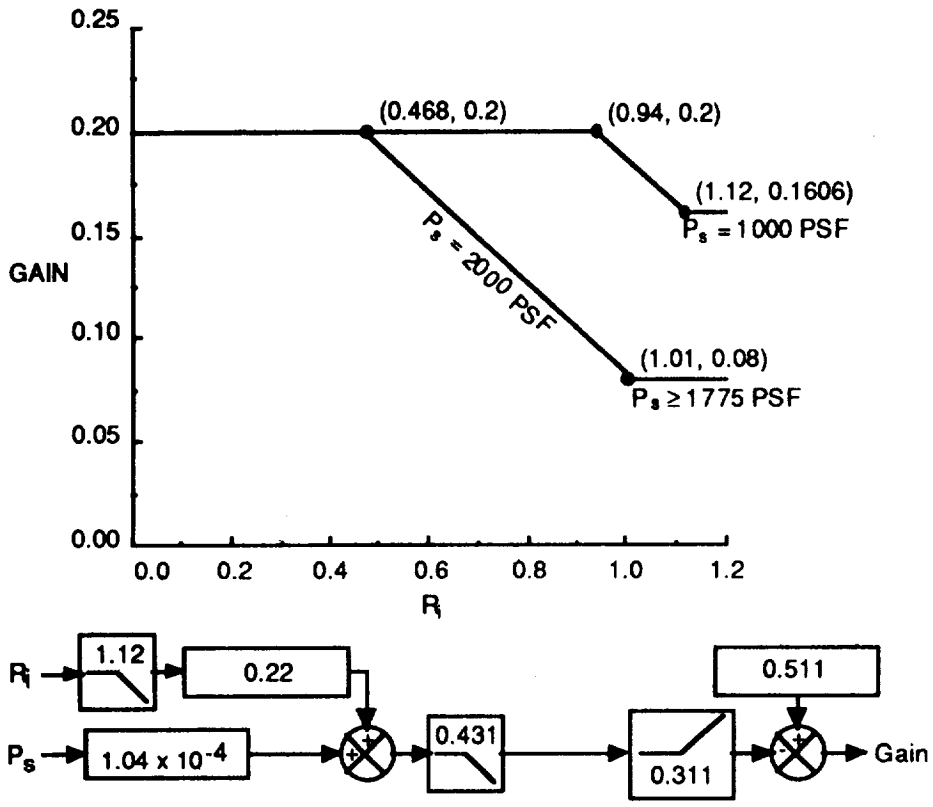


(a) Roll function 1 - Lateral stick gradient, AFU mode

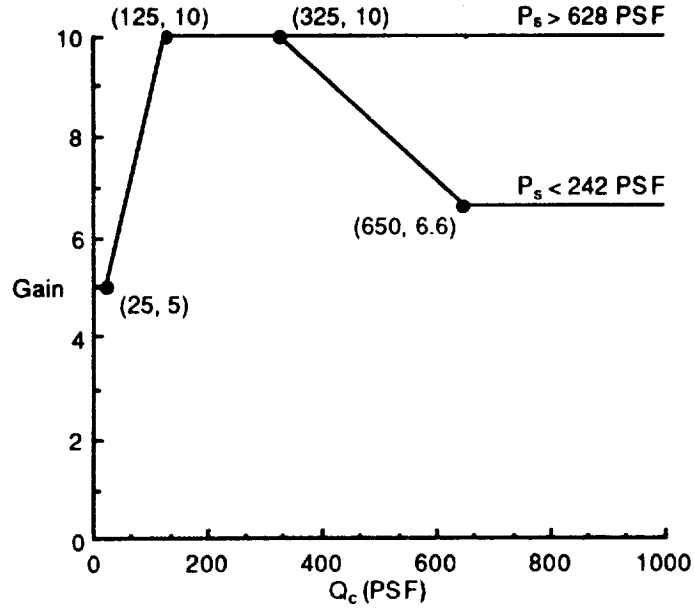


(b) Roll function 4 - Roll rate feedback gain schedule, AFU mode, (Q_C, P_S)

Figure 9.4 Lateral (roll) CAS Functions.

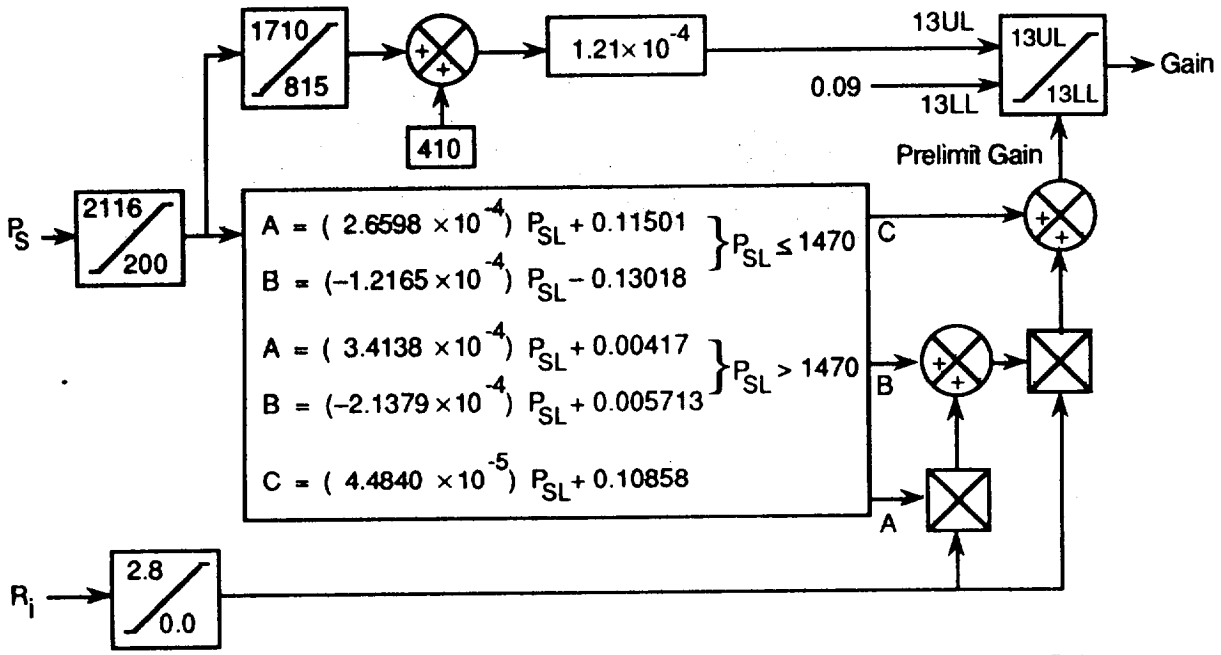


(c) Roll function 6 - Differential stabilator gain schedule, AFU mode, (R_j, P_S)

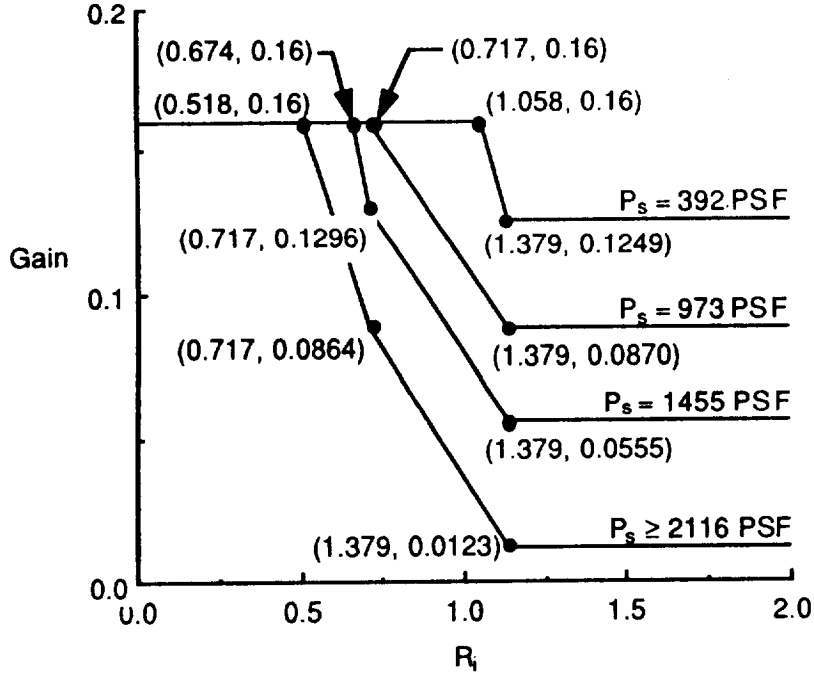


(d) Roll function 7 - Lateral command schedule gain, AFU mode, (Q_C, P_S)

Figure 9.4 Continued.

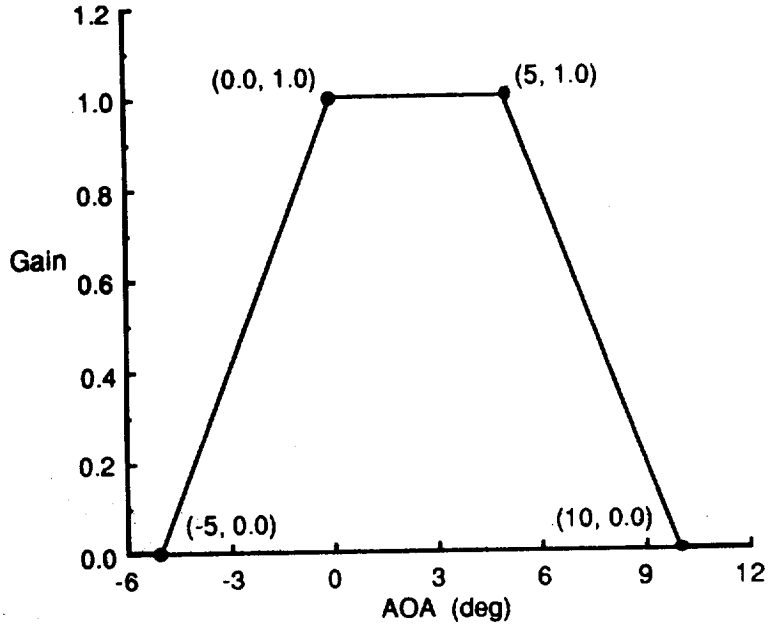


(e) Roll function 13 - Lateral command schedule gain, AFU mode, (R_i, P_S)

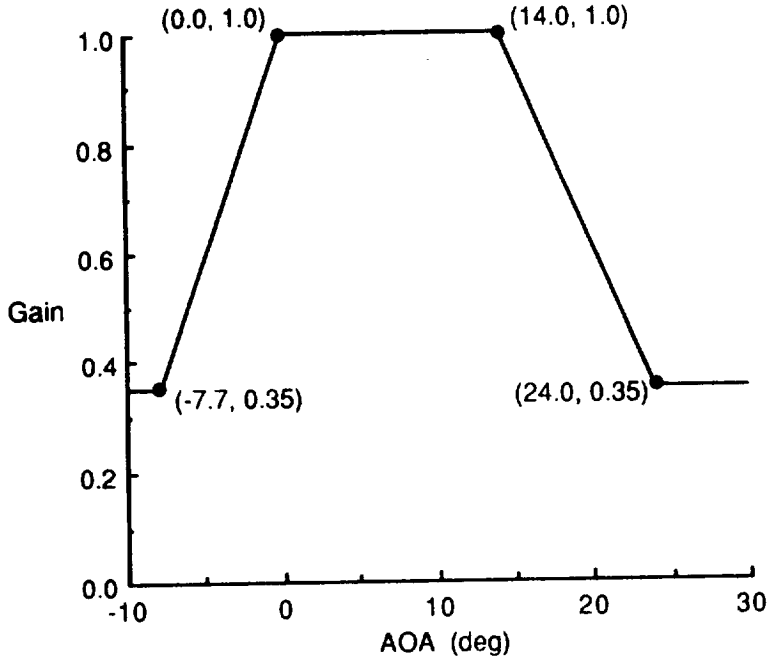


(f) Roll function 31 - Differential trailing-edge flap gain schedule, AFU mode, (R_i, P_S)

Figure 9.4 Continued.

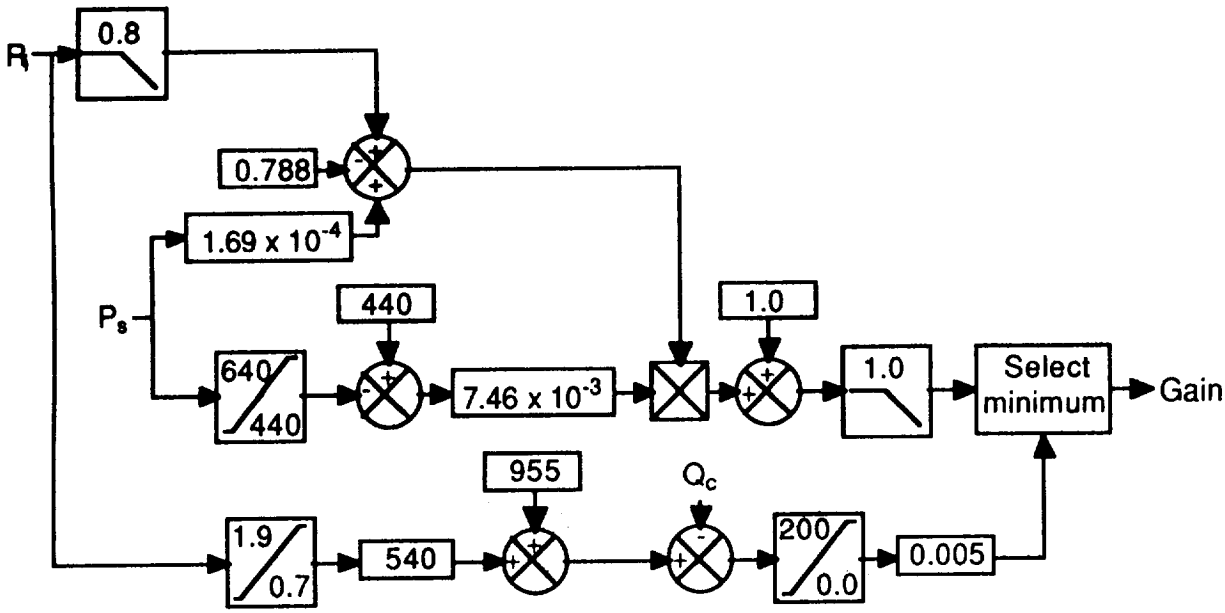


(g) Roll function 34 - Differential trailing-edge flap gain schedule, AFU mode, (α)

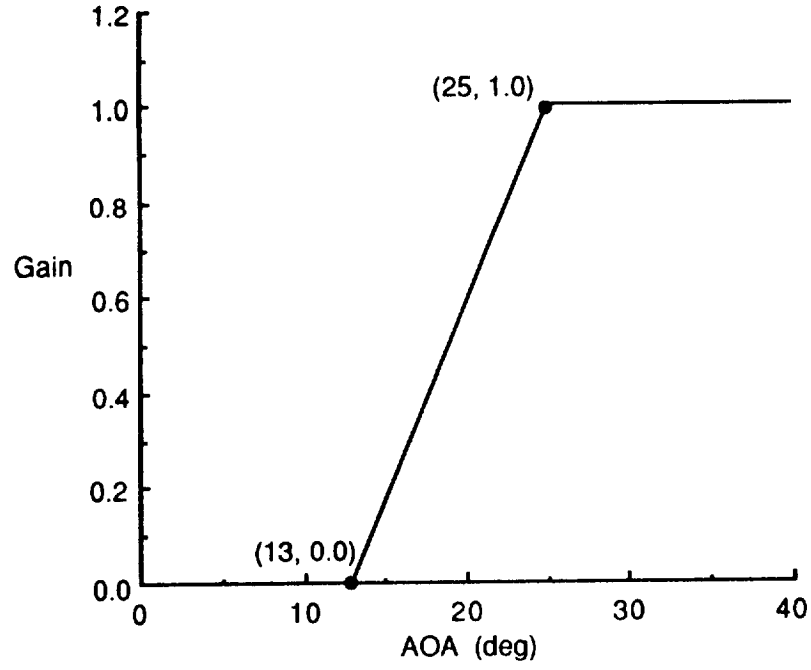


(h) Roll function 35 - Lateral forward loop gain schedule, AFU mode, (α)

Figure 9.4 Continued.

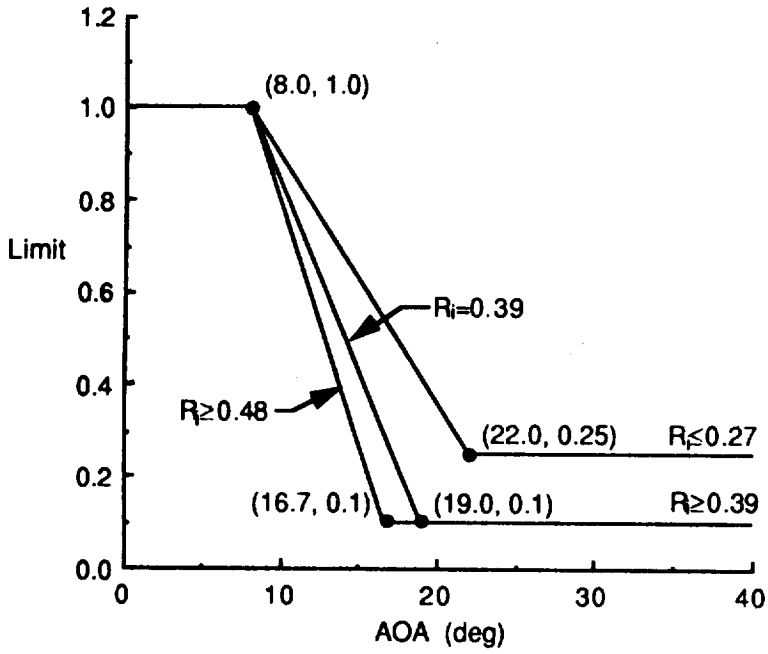


(i) Roll function 36 - Aileron gain schedule, AFU mode, (Q_C, P_S, R_j)

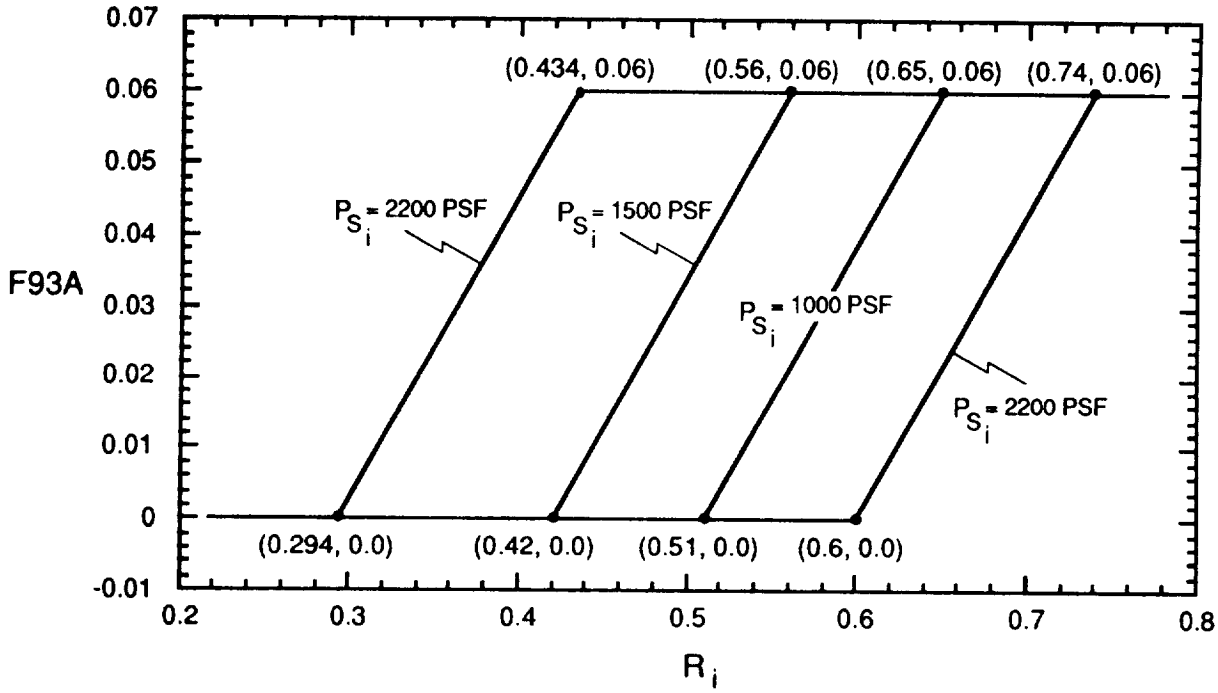


(j) Roll function 39 - Rudder pedal to roll CAS interconnect gain schedule, AFU mode, (α)

Figure 9.4 Continued.

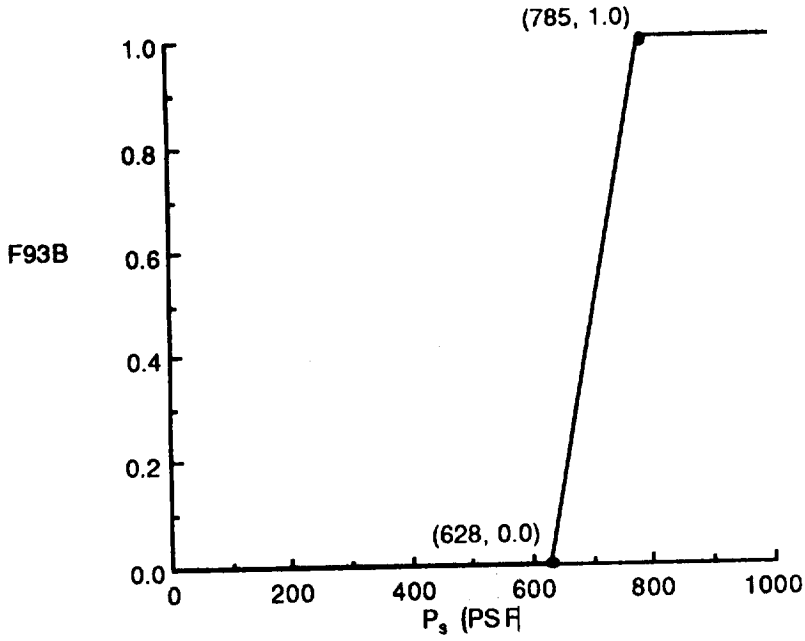


(k) Roll function 41 - Rolling surface limit schedule, AFU mode, (α, R_i)

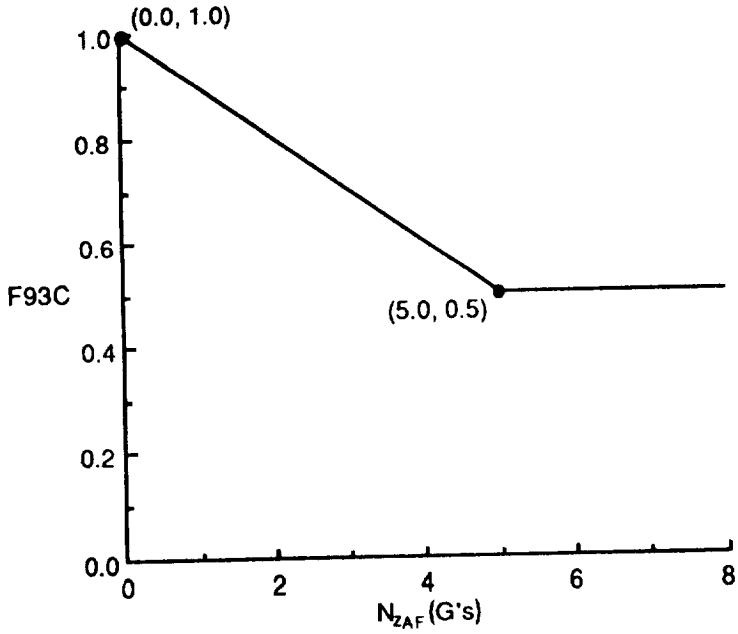


(l) Roll function 93A - Differential leading-edge flap gain schedule component, AFU mode, (R_i, P_{S_i}))

Figure 9.4 Continued.

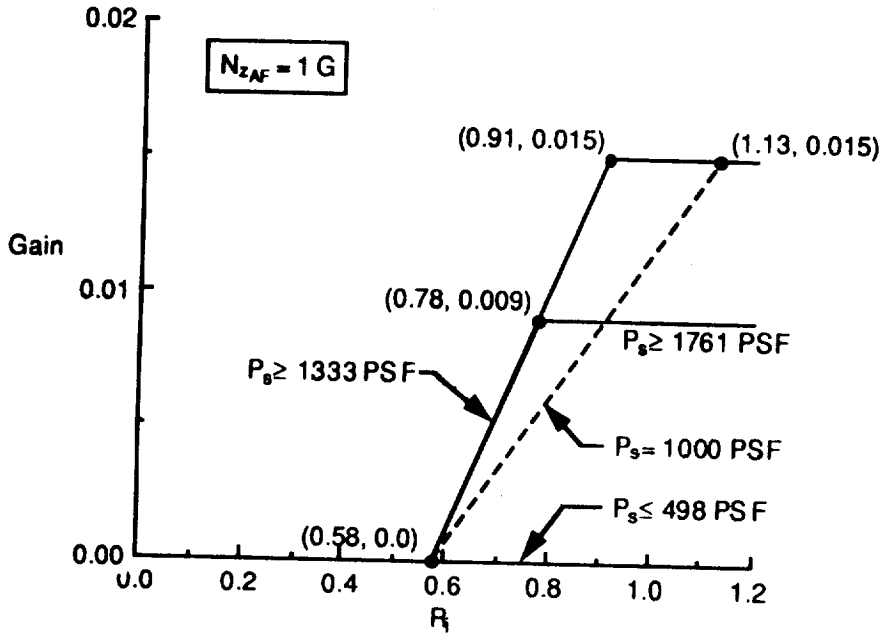


(m) Roll function 93B - Differential leading-edge flap gain schedule component, AFU mode, (P_S)



(n) Roll function 93C - Differential leading-edge flap gain schedule component, AFU mode, ($N_{Z_{AF}}$)

Figure 9.4 Continued.



(o) Roll function 101 - Differential stabilator load alleviation schedule, AFU mode, (R_i, P_s, $N_{Z_{AF}}$)

Figure 9.4 Concluded.

9.1.3 Directional Auto Flap Up CAS

Figure 9.5 (a-b) is a simplified version of the diagram "Figure 3 / OFP V10.1" on page 1-99 of MDC report A4107, Vol I, REV. J. Paths having to do with outer loops and the direct electric link (DEL) reversion mode have been deleted. The definition of the directional gains (YKxx) in terms of yaw functions (FUNxx) are given in Table 9.5. Table 9.6 defines the directional filters used in figure 9.5. The yaw functions required to define quantities in both tables 9.1 and 9.2 are defined in figure 9.6 (a-l). The following quote is from pages 1-93, 1-94 of MDC report A4107, Vol I, REV. J. Comments inserted by the authors are in italics. Grammatical hyphens have been added.

> Directional control is accomplished by a control-by-wire (CBW) system which commands the twin rudders. The yaw CAS is the primary mode of operation with a DEL (*Direct Electric Link*) mode from the rudder pedal force transducers to the rudder actuators available for CAS off operation. A rolling surface-to-rudder interconnect (RSRI) is incorporated to minimize the sideslip accompanying roll maneuvers. The block diagram is shown in Figure 3 (*Figure 9.5 of this document*). Directional functions and logic diagrams are given in Figures 3.1 through 3.27 (*Figure 9.6 of this document*). Directional digital filters are summarized in Table 1-5 (*Table 9.6 of this document*).
>
> The normal mode of control is closed loop employing stability axis yaw rate $\{r\cos(\alpha) - p\sin(\alpha)\}$ and lateral acceleration feedback control.
>
> The rudder pedal force transducer signal, after it is shaped by a dead band followed by a parabolic gradient (Function 14), is air data scheduled (Function 10) to prevent a command which would exceed the vertical tail load limits. Air data and angle-of-attack gain schedules (Function 17 and 114) eliminate aircraft departures for full pedal inputs. Full surface authority is available at high angles of attack and low Mach numbers through the rudder pedal-to-rolling surface and rolling surface-to-rudder interconnects described below.
>
> The CAS feedback loop contains a third order roll-off filter to attenuate high frequency noise and structural feedback modes. The computed stability axis yaw rate feedback signal is multiplied by a gain (Function 96) and then shaped by a washout network. The yaw rate (multiplied by cosine of angle of attack) feedback component is used to augment the Dutch roll damping. The remaining signal component, proportional to roll rate multiplied by the sine of angle of attack, is used to help provide sideslip reduction during high angle of attack maneuvering flight. Lateral acceleration feedback aids in reducing sideslip and provides turn coordination. The lateral acceleration and yaw rate feedback gains are air data scheduled (Functions 90 and 96) to increase directional stability at high altitude supersonic speeds and to optimize Dutch roll characteristics at low to mid altitude and speeds, respectively. Additional lateral acceleration air data and angle of attack gain schedules (Functions 112 and 113) improve directional stability above Mach 0.7 and 15 degrees angle of attack. Roll rate multiplied by pitch rate is fed to the directional control system to reduce the effects of inertial couping (Function 108)
>
> The forward loop consists of a gain scheduled with air data (Function 45). The gain reduces rudder pedal sensitivity in the high dynamic pressure region and optimizes the closed loop directional damping the roll coordination characteristics.
>
> The rolling surface-to-rudder interconnect (RSRI) is used to control rudder deflections to minimize the sideslip accompanying roll maneuvers. The RSRI command is the total commanded differential aileron plus differential stabilator. The RSRI signal is passed through a lead-lag network and multiplied by variable gains, which are functions of angle

of attack and air data (Functions 30 and 38). *The RSRI allows doordinated rolls to be flown with the feet on the floor.*

A rudder pedal-to-rolling surface interconnect is provided to reduce sideslip and angle of attack excursions due to rudder deflection at high angles of attack. The interconnect gain is scheduled with angles of attack and is scheduled to zero at low angles of attack (Function 39).

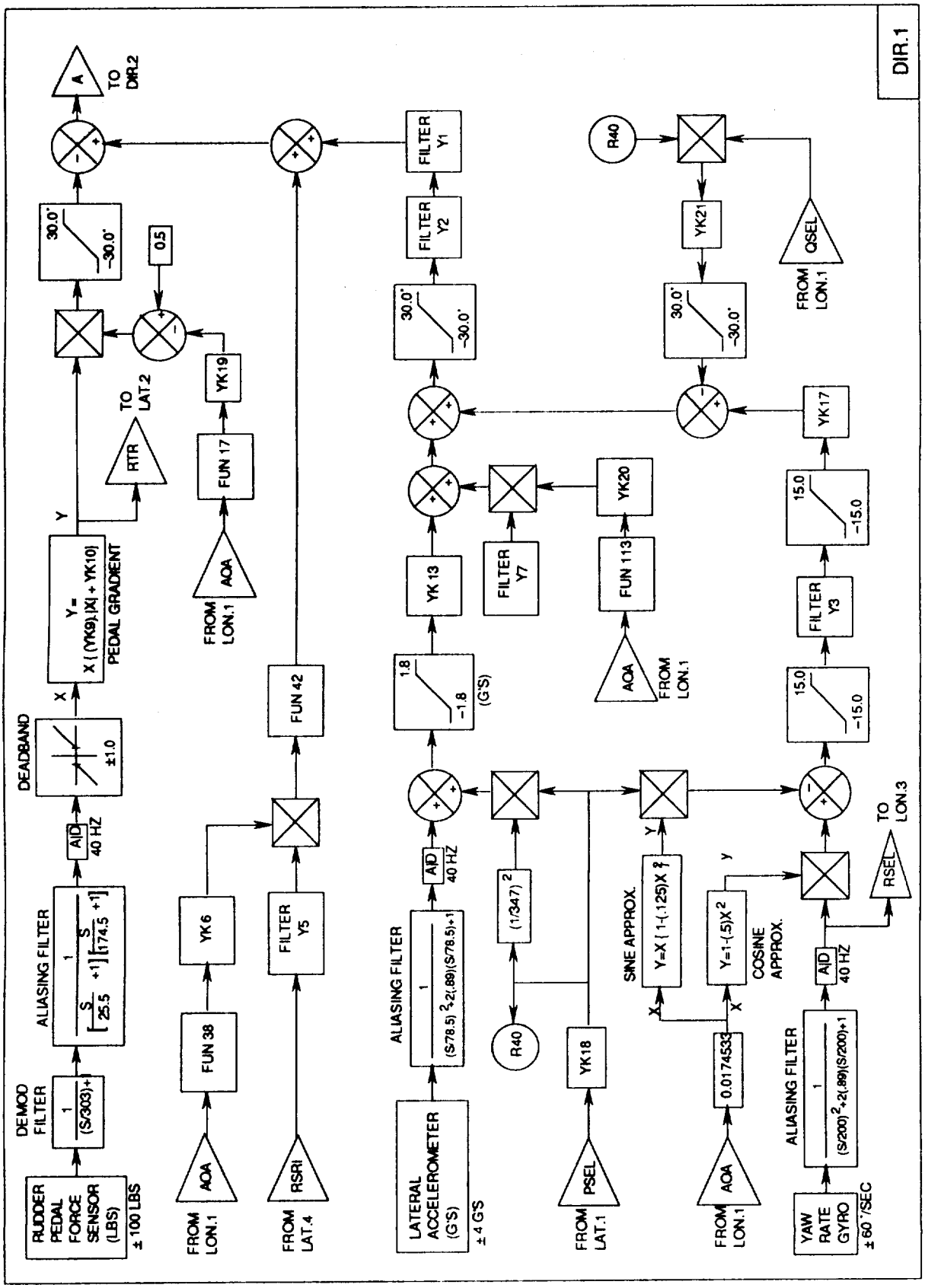


Figure 9.5. Directional CAS - (a) Main feedback paths

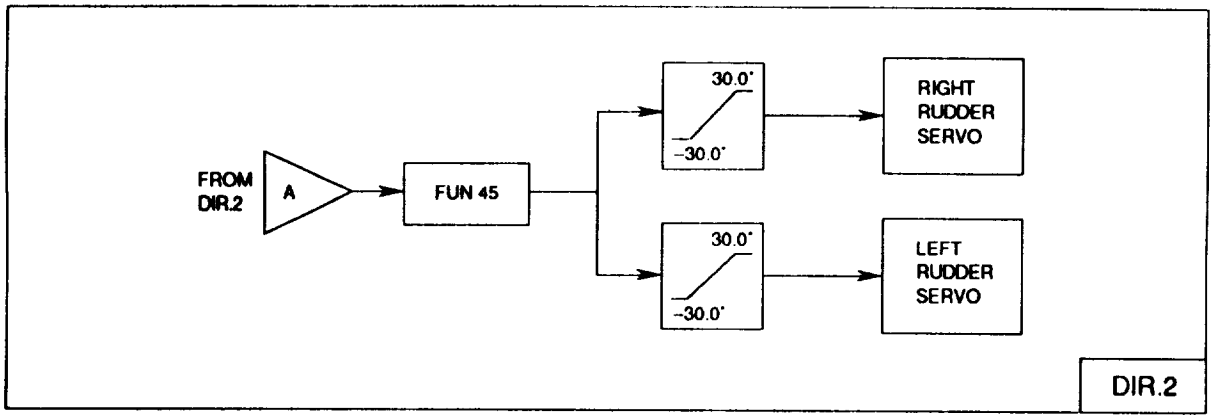


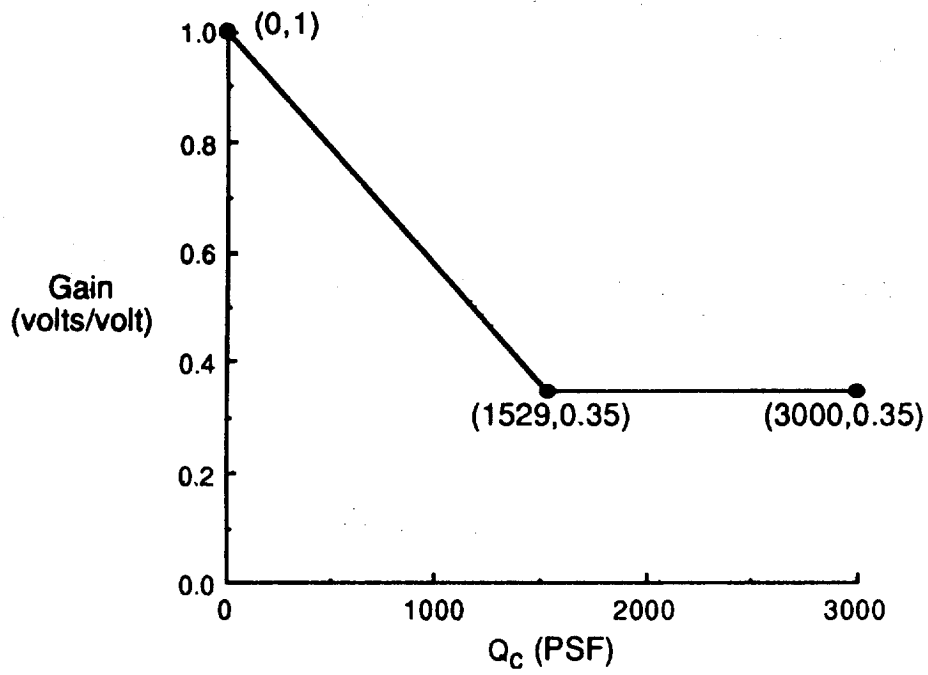
Figure 9.5. Concluded - (b) Rudder servo command synthesis

Table 9.5. Directional (yaw) gains in 8.3.3 inner loop CAS

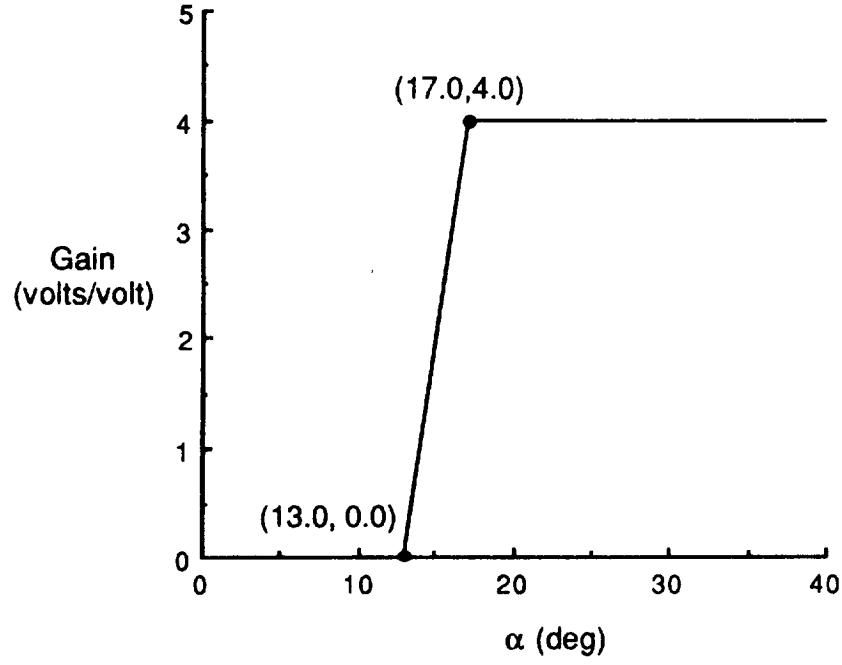
Function	AFU Mode	Spin Mode
YK6	F30	0.0
YK9	0.00072*F10	0.00144
YK10	0.234*F10	0.466
YK13	F90	0.0
YK17	F96	0.0
YK18	1.0	0.0
YK19	F114	0.0
YK20	F112	0.0
YK21	F108	0.0

Table 9.6. Directional (yaw) filters in 8.3.3 inner loop CAS

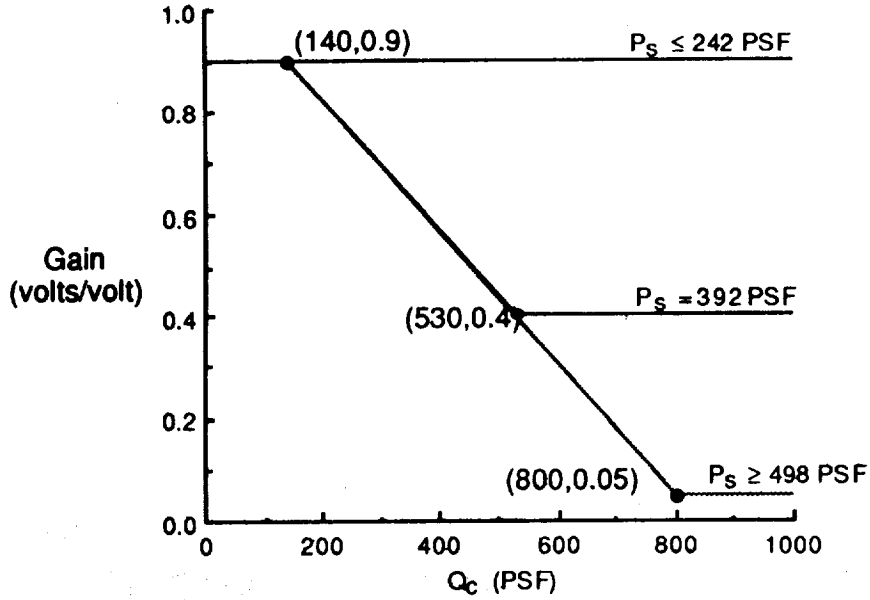
Filter	Transfer Function		Where Used
Y1	$\frac{1}{\tau s + 1}$	$\tau = 1/54.6$	1st order rolloff in lateral accel. feedback path
Y2	$\frac{1}{(s/\omega)^2 + 2\zeta(s/\omega) + 1}$	$\zeta = 0.60$ $\omega = 37.7$ (r/s)	2nd order rolloff in lateral accel. feedback path
Y3	$\frac{s}{\tau s + 1}$	$\tau = 1.0$	washout used in yaw rate gyro path
Y5	$\frac{\tau_N s + 1}{\tau_D s + 1}$	$\tau_N = 0.75$ $\tau_D = 0.50$	RSRI lead-lag
Y7	$\frac{1}{\tau s + 1}$	$\tau = 1/20.45$	1st order rolloff in lateral accel. feedback path



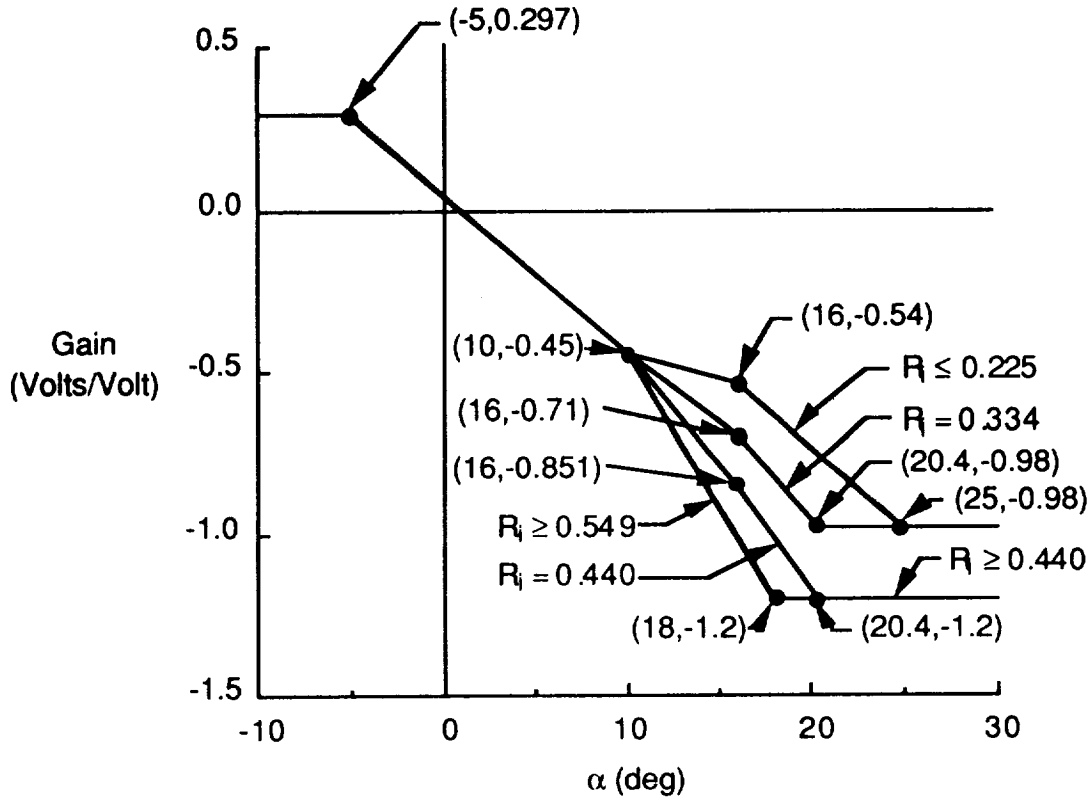
(a) Yaw function 10 - Rudder command gain, AFU mode, (Q_C)



(b) Yaw function 17 - Rudder command gain increment, AFU mode, (α)
Figure 9.6 Directional (yaw) CAS Functions.

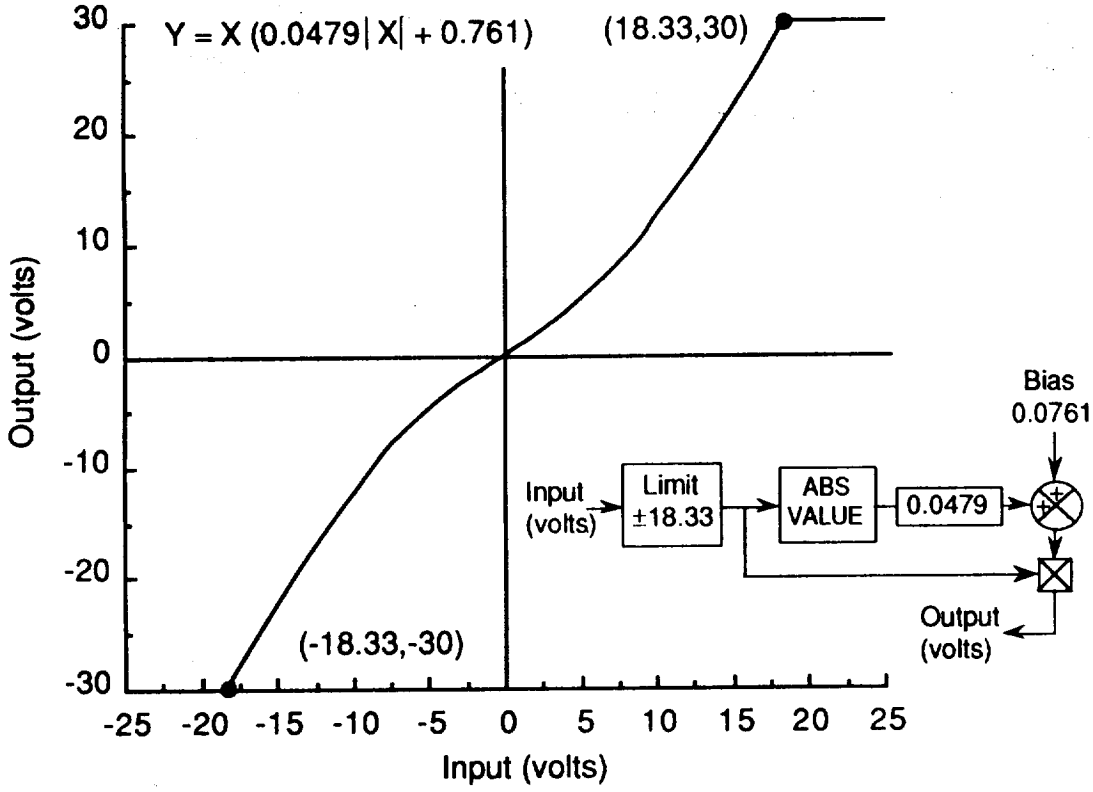


(c) Yaw function 30 - RSRI gain schedule, AFU mode, (Q_C, P_S)

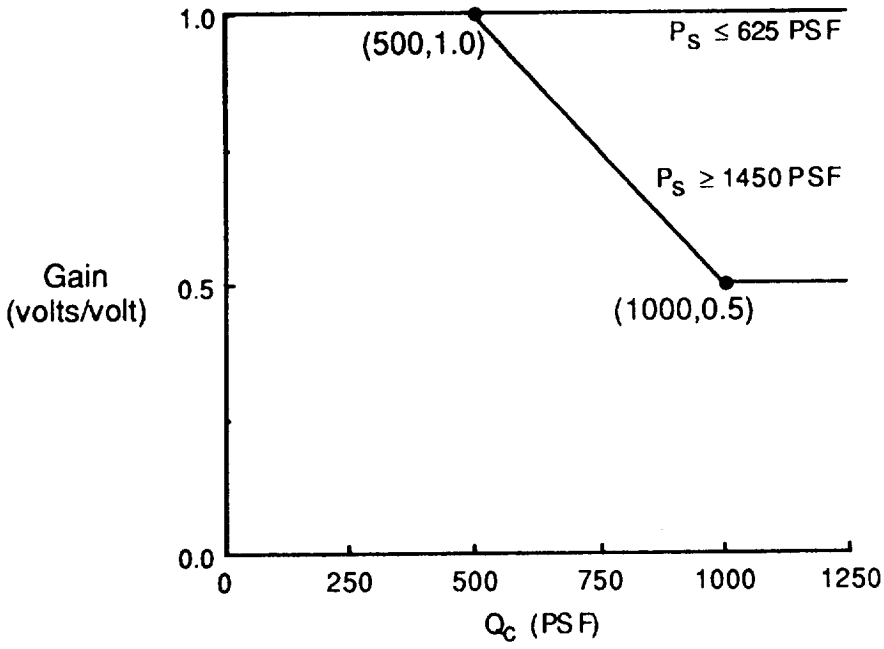


(d) Yaw function 38 - RSRI gain schedule, AFU mode, (α, R_i, P_S)

Figure 9.6 Continued.

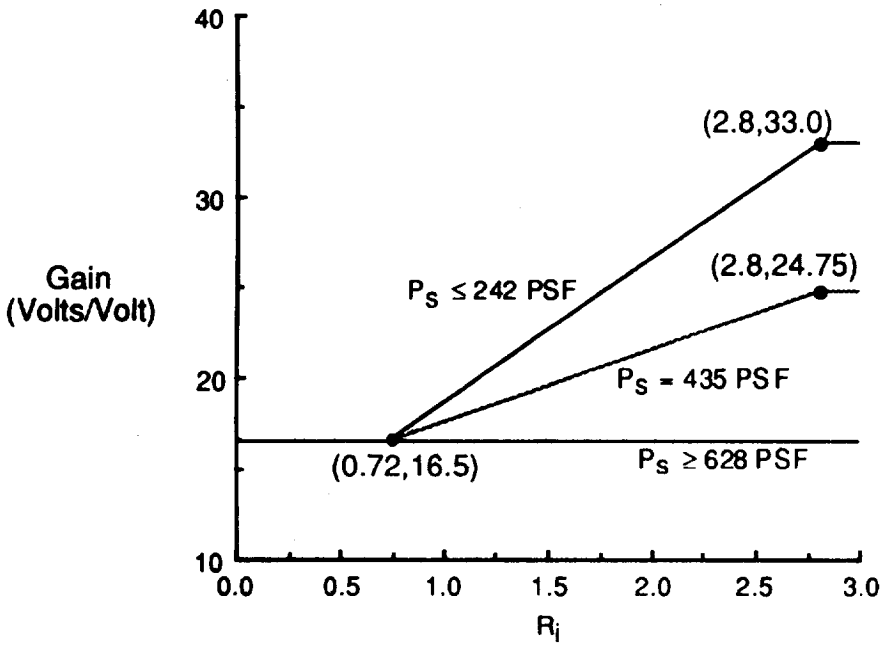


(e) Yaw function 42 - RSRI nonlinear gradient, AFU mode

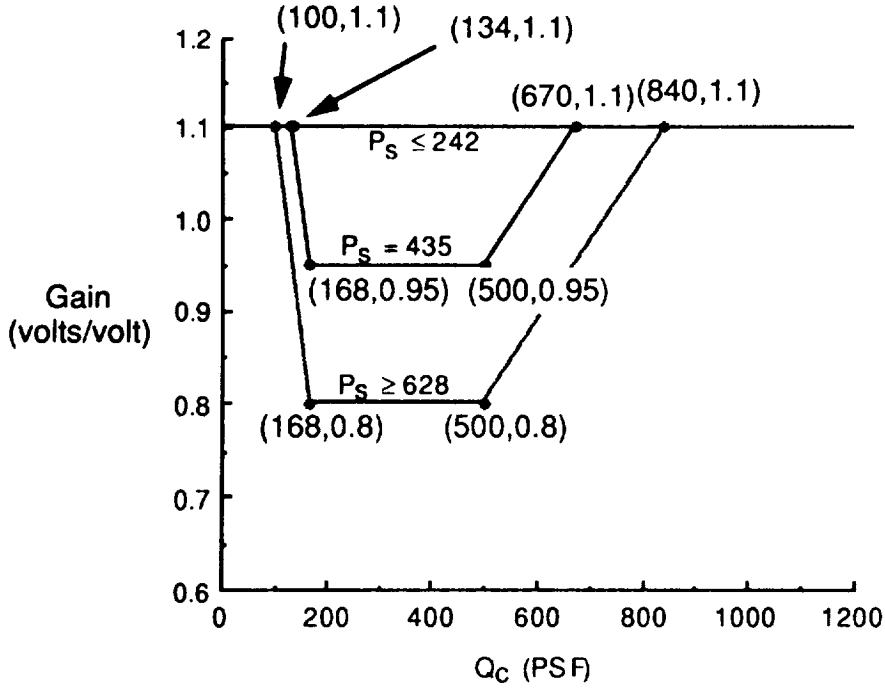


(f) Yaw function 45 - Directional forward loop gain schedule, AFU mode, (Q_C, P_S)

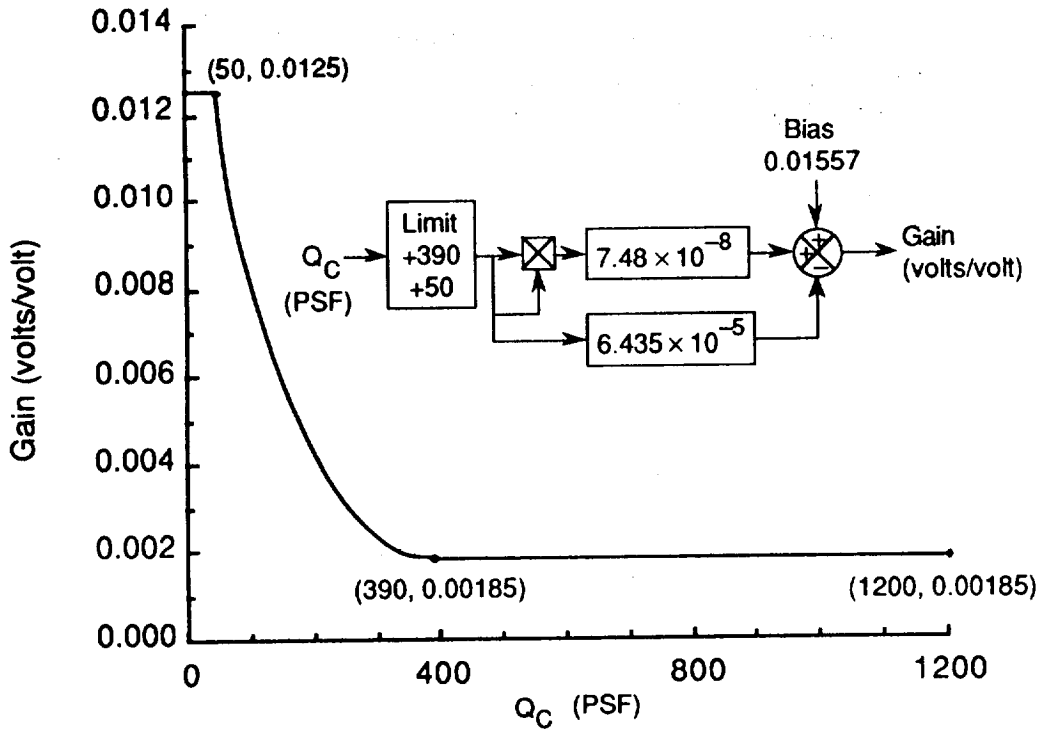
Figure 9.6 Continued.



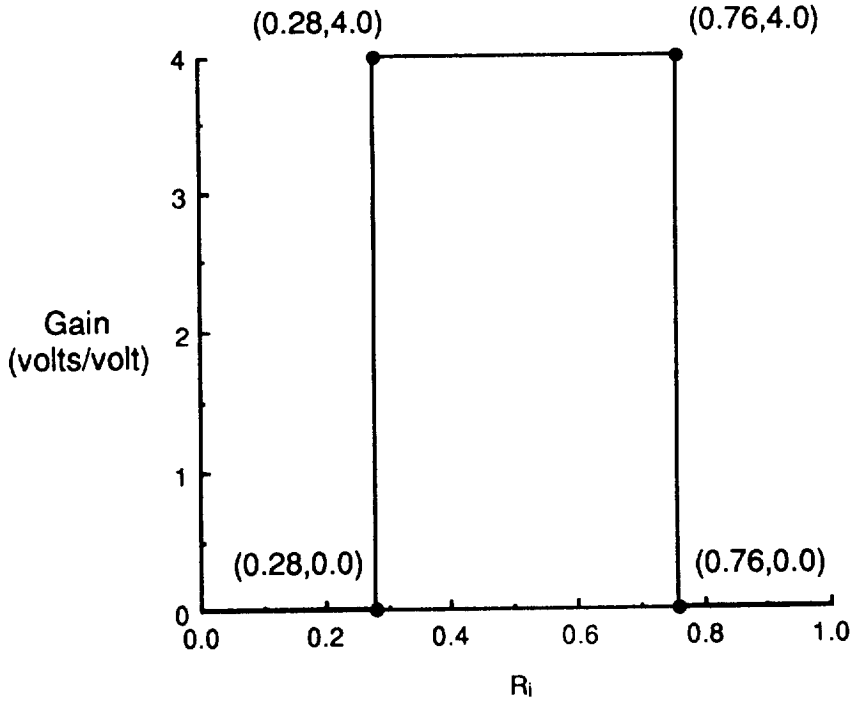
(g) Yaw function 90 - Lateral acceleration feedback gain schedule, AFU mode, (R_j, P_S)



(h) Yaw function 96 - Yaw rate gain schedule, AFU mode, (Q_C, P_S)
Figure 9.6 Continued.

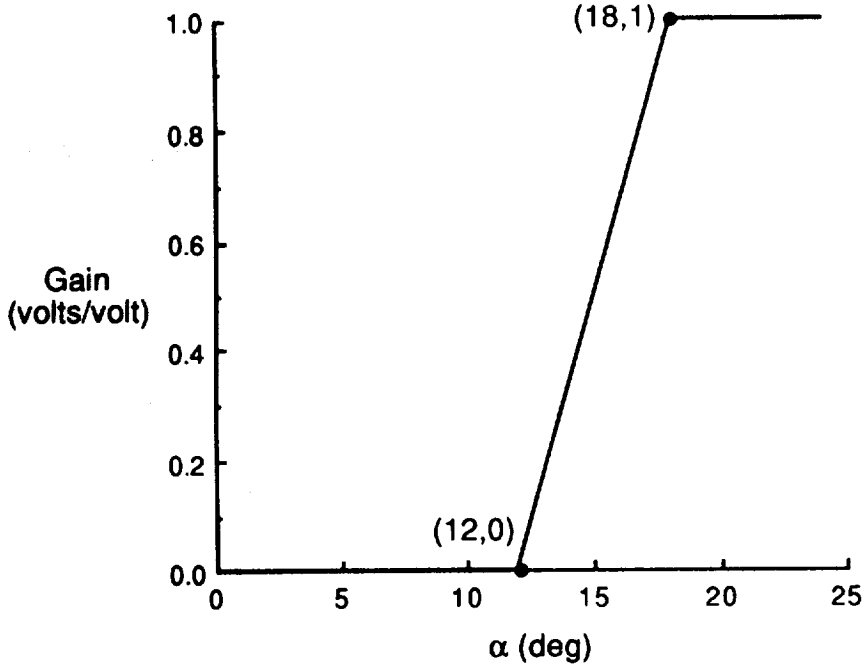


(i) Yaw function 108 - Directional inertial gain schedule, AFU mode, (Q_C)

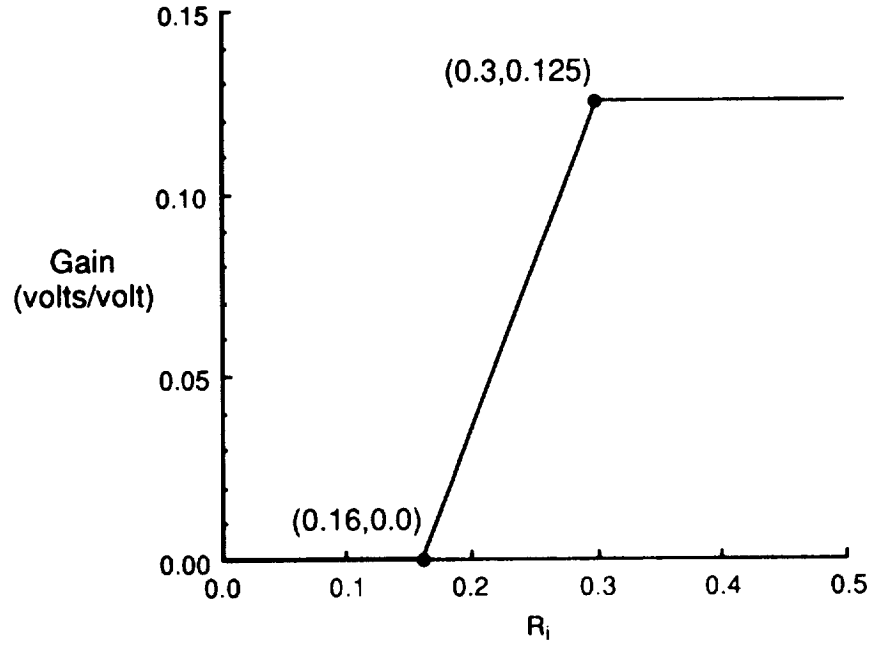


(j) Yaw function 112 - Lateral acceleration feedback gain, AFU mode, (R_i)

Figure 9.6 Continued.



(k) Yaw function 113 - Lateral acceleration feedback gain, AFU mode, (α)



(l) Yaw function 114 - Rudder pedal command gain increment, AFU mode, (R_i)

Figure 9.6 Concluded.

9.2. Flight Control System Implementation

The *f18bas* simulation flight control system (FCS) subroutines were created by starting with existing code which modeled flight controls in the Differential Maneuvering Simulator real-time F18 simulation (*dmsf18*). The code structure and variable nomenclature used in the *dmsf18* simulation were left largely unchanged. The interface portions of the control subroutines were modified to be compatible with the Advanced Continuous Simulation Language (ACSL) structure. When calling FORTRAN subroutines from an ACSL simulation, arguments may be passed through a parameter list or through the global ACSL common block, ZZCOM. Using the global common block is risky because all the ACSL defined variables are in ZZCOM and duplicate names can easily exist in the called subroutine. In the interest of execution speed, real-time subroutines use a multiple common block structure to pass arguments instead of parameter lists. The existing real-time common blocks were replaced with parameter lists containing the feedback and driver values required by the FCS subroutines. Sensor and servoactuator models, which appeared in the real-time FCS subroutines, were removed and coded as continuous models in the ACSL main program. Minor modifications were required for the CDC FORTRAN V to VAX FORTRAN conversion. For example, multiple assignment statements can occur on a single line of code in CDC FORTRAN V (A=B=C), but not in VAX FORTRAN. Logical variables pertaining to real-time systems (i.e. the real-time monitor, cockpit lights, strip chart recorders, etc.) were removed. Library functions and subroutines were modified as required for compatibility. Extraneous code which was present for an alternate, expanded, flight control system was removed. A general cleanup of unused variables and dead end code was performed.

Five subroutines comprise the flight control loop. An executive routine, F18FCS, is called from the discrete block, FCS, in the main ACSL routine. The call is scheduled at a regular time interval specified by the ACSL variable, TSFCS. The default interval value is 1/40 (.025) second which represents a compromise between the various sampling rates of 80, 40, 20 and 10 samples per second used by the actual aircraft flight control computer. Sensor signals and pilot control positions are passed to the control routines through the subroutine call parameter list. Surface position commands are then computed and returned to the ACSL main routine through this parameter list. Figure 10.1 presents the overall flow of the executive routine and shows only the main driver and feedback values and the resulting surface deflections. Specific options and special conditions were not shown in this diagram to reduce the clutter. A complete list of passed parameters for subroutine F18FCS is given in tables 9.7 and 9.8.

The executive routine, F18FCS, computes logical variables indicating mode and filter coefficients. The mode logical variables are: spin mode, stores, the supermaneuver switch, automatic speedbrake retraction, and limiter override commands. The filter coefficients are for the dynamic pressure digital filters and the angle of attack signal digital filters. The remaining four control routines are called from F18FCS in the following order:

(1) the longitudinal control model, F18LONG,
(2) the lateral control model, F18LAT,
(3) the directional control model, F18DIR,
(4) and an auxiliary routine, F18AUX.

The basic inputs for the longitudinal axis control routine, F18LONG, are:

(1) the pilot longitudinal stick command,
(2) normal accelerometer feedback,
(3) pitch rate gyro feedback,
(4) angle-of-attack sensor feedback,
(5) dynamic pressure sensor scheduling input,
(6) and an aileron to stabilator interconnect command from the lateral control routine.

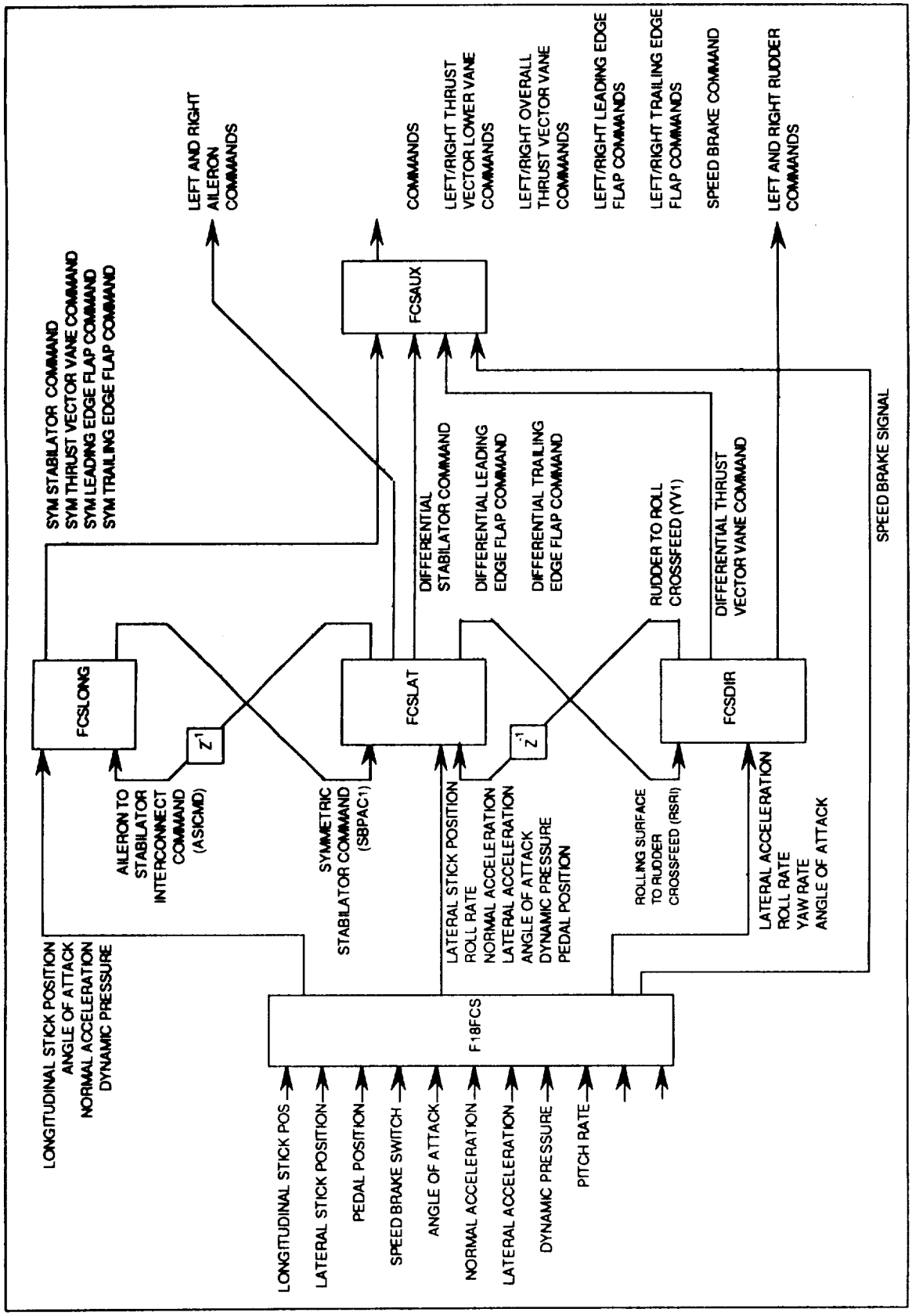


Figure 9.7. Implementation of rhe *f18bas* flight control laws

The basic outputs for the longitudinal axis control routine are:

(1) stabilator surface commands,
(2) the leading and trailing-edge flap commands,
(3) and the symmetric thrust vectoring command.

The basic inputs for the lateral axis control routine, F18LAT, are:

(1) the pilot lateral stick command,
(2) roll rate sensor feedback,
(3) normal accelerometers feedback,
(4) lateral accelerometer feedback,
(5) angle of attack sensor feedback,
(6) dynamic pressure sensor scheduling input,
(7) and a rudder to rolling surface crossfeed command from the directional routine.

The basic outputs for the lateral axis control routine are :

(1) aileron surface commands,
(2) the interconnect command to the stabilator and flaps,
(3) and a rolling surface to rudder interconnect command.

The basic inputs for the directional axis control routine, F18DIR, are:

(1) the pilot pedal command,
(2) lateral accelerometer feedback,
(3) roll rate gyro feedback,
(4) yaw rate gyro feedback,
(5) angle of attack sensor feedback,
(6) and a rolling surface to rudder interconnect command from the lateral routine.

The basic outputs for the directional axis control routine are :

(1) differential rudder command,
(2) symmetric rudder command,
(3) and a differential thrust vectoring command.

The auxiliary routine blends and schedules the thrust vectoring commands and transforms symmetric and differential surface commands into left and right specific surface commands. The auxiliary routine originally contained servoactuator and hinge moment models. Hinge moments were not being computed at the time of this documentation and the servoactuators existed in the continuous portion of the simulation.

Table 9.7. Input Parameters for subroutine F18FCS

f18bas	Local	Definition
ALFD	ALPDOT	d/dt of angle of attack, rad/sec
ALFS	ALFS	angle-of-attack sensor output, degrees
AYS	NYSENS	lateral accelerometer output, + right, g's
AZS	NZSENS	vertical accelerometer output, + up, g's
A81	A81	left engine outlet area, in^2
A82	A82	right engine outlet area, in^2
BETD	BETADOT	d/dt sideslip angle, rad/sec
BETDG	BETADEG	sideslip angle, degrees
BLCG	YBASIC	buttock line coordinates of the c.g., inches
CPL	CPL	left throttle command, degrees
CPR	CPR	right throttle command, degrees

Table 9.7. Input Parameters for subroutine F18FCS (continued)

f18bas	Local	Definition
CSALF	COSALP	cosine of angle of attack
DAASTR	DELATM	calculated antisymmetric aileron trim deflection, degrees
DATMBT	DATMBUT	aileron trim button setting, degrees
DLGPCT	DLGPCT	landing gear state, [0,1], 0=gear up, 1=gear down
DRASTR	DELRTM	calculated antisymmetric rudder trim position, degrees
DSSY	STBPP	symmetric stabilator deflection, degrees
DSSYTR	DELSTM	calculated symmetric stabilator trim position, degrees
DSTMBT	DSTMBUT	stabilator trim button setting, degrees
FCRSET	FCRESET	lateral trim reset logical for spin, T/F
FCSV83	FCSV833	set pitch forward loop gain logical, T/F
FSCG	XBASIC	fuselage station coordinates of the c.g., inches
F7BYPS	F7BYPA	schedule lateral gain logical, T/F
G	G	acceleration due to gravity, ft/sec^2
GLIMTR	GLIMITR	g limit on longitudinal stick logical, T/F
GLORID	GLORIDE	g limit override logical, T/F
HSTORE	HSTORE	heavy stores logical, T/F
ISENS	IDEALS	integer, (0) = ideal sensors, (1) = MDC sensor model
KFS	KFS	stick spring force constant, n.d.
LTHVEC	TREVERS	thrust vectoring engaged, T/F
LTRFLG	TRMMING	logical, true if simulation in trim loop, T/F
MACH	MACH	Mach number
MNSPIN	MANSPIN	logical, manual spin mode, T/F
NTVOPT	NTVOPT	integer, thrust vectoring pitch control blend
ORIDE	ORIDE	logical, override angle-of-attack limiter, T/F
PS	PSENS	output of roll rate sensor, °/sec
PCA	DELTAP	lateral pilot stick input, inches
PCR	DELTRP	rudder pedal force input, lbs
PCS	DELTSP	longitudinal pilot stick input, inches
PCSTR	DLSPTM	calculated trim position for longitudinal stick, in
PD	PDOT	roll acceleration, rad/sec^2
PSAS	PSAS	logical, pitch stability augmentation system engaged, T/F
PSTS	PSTATIC	Sensed static pressure, lbs
QS	QSENS	output of pitch rate sensor, °/sec
QBAR	QBAR	dynamic pressure, lbs/in^2
QCIS	QCIS	sensed compressible impact pressure, lbs
QD	QDOT	pitch acceleration, rad/sec^2
RS	RSENS	output of yaw rate sensor, °/sec
RA2DG	DEGRAD	$(180/\pi)$
RD	RDOT	yaw acceleration, rad/sec^2
RSAS	RSAS	logical, roll stability augmentation system engaged, T/F
SETVDC	SETTVDC	differential thrust vectoring set, degrees
SETVSC	SETTVSC	symmetric thrust vectoring set, degrees
SNALF	SINALP	sine of angle of attack

Table 9.7. Input Parameters for subroutine F18FCS (concluded)

f18bas	Local	Definition
STORES	STORES	logical, stores switch engaged, T/F
T	T	simulation time, sec
TSFCS	H	update frame time for flight control subroutines, sec
TT	TT	total thrust, lbs
TVDBOT	TVDBOUT	logical, thrust vectoring deadband removed, T/F
VT	VTOTAL	total airspeed, ft/sec
WLCG	ZBASIC	waterline coordinate of c.g., inches
WT	WEIGHT	weight of airplane, lbs
YSAS	YSAS	logical, yaw stability augmentation system engaged

Table 9.8. Output Parameters for subroutine F18FCS

f18bas	Local	Definition
CAL	DELTACL	left aileron command, degrees
CAR	DELTACR	right aileron command, degrees
CFL	DELTFCL	left trailing-edge flap command, degrees
CFR	DELTFCR	right trailing-edge flap command, degrees
CNL	DELTNCL	left leading-edge flap command, degrees
CNR	DELTNCR	right leading-edge flap command, degrees
CRL	DELTRCL	left rudder command, degrees
CRR	DELTRCR	right rudder command, degrees
CSB	DELTSBC	speedbrake command, degrees
CSL	DELTSCL	left stabilator command, degrees
CSR	DELTSCR	right stabilator command, degrees
CTVLL	CTVLL	left lower vane (thrust vectoring) command, degrees
CTVLU	CTVLL	left upper vane (thrust vectoring) command, degrees
CTVRL	CTVLL	right lower vane (thrust vectoring) command, degrees
CTVLU	CTVLL	right upper vane (thrust vectoring) command, degrees
DLTVCL	DLTVCL	left overall vane (thrust vectoring) command, degrees
DLTVCR	DLTVCR	right overall vane (thrust vectoring) command, degrees

The call to the F18 flight control system subroutine (F18FCS) appears twice in the main ACSL program (*f18bas*). A flowchart of the main program which illustrates the trimming and main program loops is shown in figures 2.2 through 2.4 in section 2 of this report.

The call to subroutine F18FCS in the *initial...end* code section of the simulation (figure 2.3) is required by the trimming procedure. F18FCS is only called if the "FCS engaged" logical (LFCS) is true. The call to F18FCS in *initial...end* code block is required because the ACSL translator sorts the FCS discrete block and places it after the derivative section where it is can then be checked for an event crossing, (T=TFCS in this case). Actuator models represent physical systems (the servoactuator motors that drive the aircraft surfaces) and appear in the continuous

(derivative) portion of the program code. If the call to F18FCS were not present in the initial section of the ACSL program, the following sequence would occur:

(a) The trim routine would drive a pilot input.
(b) Actuator responses would be computed before the discrete flight control system call.
(c) The surface control command from the previous loop would drive the actuator.
(d) No surface deflection response to the trial pilot input would be computed.
(e) A singularity error would be noted in the trim routine.
(f) 'ESTIMATED [A] MATRIX SINGULAR' would be returned at ACSL prompt.
(g) The simulation would remain untrimmed.

As a result of entering the ACSL runtime command 'START', the simulation process begins execution at the top of the initial section. The top portion of the initial section code is executed once at the beginning of each run (i.e., each time START is entered at the ACSL prompt) and is composed primarily of type declarations (LOGICAL, INTEGER) and constant statements. Constant statements are implemented by the ACSL translator as Fortran DATA statements and the values are entered once into the computer memory when the program is linked. A few direct assignment statements, notably ITFLG=1, also occur in top portion of the initial section code and these values are reset with each run. The continuation statement, I100..CONTINUE, marks the top of the trim loop. Code appearing after this continuation statement is executed with each loop of the trim sequence until the pass on which the variable ITFLG > 2 is true. Successful trim is indicated by IRFLG=3. The subroutine INITD is called following the I100 statement. This call was included to clear the event list and prevent problems with the crossing value setting of the afterburner switch discrete . Initial values for sensor data are computed prior to the initial flight control system call. The first call to the flight control system computes surface deflection commands based on the initial trim values of pilot control commands. These initial trim pilot control values are read on the first pass through the initial section by the subroutine LOAD from the trim data file 'F18TR'. The trial pilot control values are used to compute trim are:

(a) longitudinal stick position, PCSTR (Pilot Control Stabilator TRim),
(b) lateral stick position, PCATR, (Pilot Control Aileron TRim),
(c) and pedal force, PCRTR (Pilot Control Rudder TRim)

The logical variable LFCS must be set true for the flight control system to be called. A false value of LFCS will cause the flight control system to be bypassed. The LFCS false condition is present to enable the simulation to be run from direct surface deflection commands. Trim is achieved when the first NXTR elements of the XTRIM array (see table 2.1 in section 2.) are driven to values that cause first NXTR elements of the YTRIM array (table 2.2) to be zero to within some tolerance. The specific combination of XTRIM values used to achieve trim is dependent on which type of trim condition is required (a straight and level trim should not require pedal and lateral stick adjustment, unless some asymmetry was introduced to the aerodynamics). This combination is set up in the trim condition file ('F18TR'). The procedure for setting up a trim case is more fully described in the section 2.2 of this report.

10. CONCLUSION

A mathematical model and associated computer program (*f18bas*) to simulate a twin-tailed high performance fighter airplane (McDonnell Douglas F/A-18) were described. The simulation was intended to support advanced control law research for application to high performance aircraft. The modeled flight envelope is extensive, allowing investigations in the high-angle-of-attack portion of the flight regime. While no comparisons with F/A-18 flight data are offered, the simulation is sufficiently detailed to support proof-of-concept studies.

The simulation program was written in the Advanced Continuous Simulation Language (ACSL). Wherever possible, continuous system dynamics were modeled in the DERIVATIVE portion of the ACSL program structure. This policy allows important features of ACSL, such as the automatic generation of linear models and error checking on the numerical integration process, to be utilized. Furthermore, ACSL, by implementing continuous and discrete event dynamics via the DERIVATIVE and DISCRETE block conventions, allows a coherent implementation of multirate digital control laws that should not be lightly discarded.

The simulation math model includes the nonlinear, six degree-of-freedom rigid-body equations, an engine model, sensors, and first-order actuators with rate and position limiting. A simplified form of the F/A-18 digital control laws (version 8.3.3) is implemented, including only the Auto Flap Up (AFU) flight mode used for air combat maneuvering. Aerodynamic forces and moments are calculated from a wind-tunnel-derived database using table look-ups with linear interpolation. The aerodynamic database has an angle-of-attack range of -10° to +90° and a sideslip range of -20° to +20°. The effects of elastic deformation were incorporated in a quasi-static-elastic manner. Elastic degrees of freedom are not actively simulated. In the engine model, the throttle-commanded steady-state thrust level and the dynamic response characteristics of the engine are based on airflow rate as determined from a table look-up. Afterburner dynamics are switched in at a threshold based on the engine airflow and throttle position.

The source code can be obtained from the authors.

REFERENCES

Advanced Continuous Simulation Language (ACSL) Reference Manual, Edition 4.2, Mitchell and Gauthier Associates, Concord, MA, 1987.

Arbuckle, P.D., Buttrill, C.S., and Zeiler, T.A.; "A New Simulation Model Building Process for Use in Dynamics Integration Research"; AIAA Paper No. 87-2498-CP; *Proceedings of the 1987 Flight Simulation Technologies Conference*; Monterey, CA, August, 1987.

Buttrill, C.S., Zeiler, T.A., and Arbuckle, P.D.; "Nonlinear Simulation of a Flexible Aircraft in Maneuvering Flight"; AIAA Paper No. 87-2501-CP; *Proceedings of the 1987 Flight Simulation Technologies Conference*; Monterey, CA, August, 1987.

Etkin, B.; *Dynamics of Atmospheric Flight*; John Wiley & Sons, Inc.; New York, 1972

Format Subcommittee; "The SCI Continuous System Simulation Language (CSSL)", SCI Simulation Software Committee, Simulation Councils, INC., *Simulation*, Vol. 9, No. 6, December, 1967.

Gainer, T.G. and Hoffman, S.; "Summary of Transformation Equations and Equations of Motion Used in Free-Flight and Wind-Tunnel Data Reduction and Analysis", NASA SP-3070, 1972.

MDC A4107, "F/A-18 Flight Control Electronic Set Control Laws", Volume I, Issue date: 19 March 76, Revision Date: 1 July 88, Revision letter: J, McDonnell Aircraft Company

MDC A7247, "F/A-18 Stability and Control Data Report, Vol I: Low Angle of Attack", Issue date 31 August 1981, Revision date 15 November 1982, Revision Letter B, McDonnell Aircraft Company

MDC A7247, "F/A-18 Stability and Control Data Report, Vol II: High Angle of Attack", Issue date 31 August 1981, McDonnell Aircraft Company

MDC A8575, "F/A-18 Basic Aerodynamic Data", Issue date 31 March 1984, McDonnell Aircraft Company

MDC A7813 - Volume I, "F/A-18A Flight Control System Design Report, Volume I, System Description and Theory of Operation", Revision date: 28 September 1984, Revision letter A, McDonnell Aircraft Company

MDC A7813 - Volume II, "F/A-18A Flight Control System Design Report, Volume II, Flight Control System Analysis - Inner Loops", Issue date: 15 June 1984, McDonnell Aircraft Company

MDC A7813 - Volume III, "F/A-18A Flight Control System Design Report, Volume III, Flight Control System Analysis - Automatic Flight Modes", McDonnell Aircraft Company

MDC A8575, "F/A-18 Basic Aerodynamic Data", Issue date 31 March 1984, McDonnell Aircraft Company

Roskam, J.; *Airplane Flight Dynamics and Automatic Flight Control*; Roskam Aviation and Engineering Company; Route 4, Box 274, Ottawa, Kansas; 1979

Zeiler, T.A., and Buttrill, C.S.; "Dynamic Analysis of an Unrestrained, Rotating Structure Through Nonlinear Simulation"; AIAA Paper 88-2232-CP; *Proceedings of the AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference*, Williamsburg, VA, 1988.

SYMBOLS

Symbol	Definition	Variable Name	Program Unit
a.r.c.	aerodynamic reference center, in		
b	wing span reference length used for force and moment coefficients, 37.42 ft	BWRF	CSL
BL_a	buttock line coordinate of accelerometers, + right, in. (default = 10.5)	NYLOC(2)	CSL
BL_{cg}	buttock line coordinate of c.g., + right, in. (default = 0.0)	FSCG	CSL
BL_{eng_L}	buttock line coordinate of engine/airframe interface point for left engine, + right, – 18.9, in.	BLENGL	CSL
BL_{eng_R}	buttock line coordinate of engine/airframe interface point for right engine, + right, 18.9, in.	BLENGR	CSL
BL_{rf}	buttock line of aerodynamic reference center, + right, 0.0 in.	BLRF	CSL
$\bar{c}$	wing chord reference length used for force and moment coefficients, 11.52 ft	CWRF	CSL
C_{D_o}	basic drag coefficient, controls at zero, rigid airplane, f(α,M)	CDO	SFAERRF
$C_{D_{sf}}$	total drag coefficient, steady flow, along -x axis in the stability frame, $C_D\|_{\dot{V}=0.0}$	CDREFO	SFAERRF
$C_{D_{\delta_f}}$	drag coefficient per unit symmetric trailing-edge flap, deg^{-1}, f(α,M)	CDDF	SFAERRF
$C_{D_{\delta_n}}$	drag coefficient per unit symmetric leading-edge flap, deg^{-1}, f(α,M)	CDDN	SFAERRF
c.g.	center-of-gravity, in		
C_{l_o}	basic roll coefficient, controls at zero, rigid airplane, f(β,α,M)	C1O	SFAERRF
C_{l_p}	roll coefficient per non-dimensional roll rate, $\partial C_l/\partial\left(\frac{pb}{2V}\right)$, f(α,M)	C1P	SFAERRF

C_{l_r}	roll coefficient per non-dimensional yaw rate, $\partial C_l / \partial\left(\frac{rb}{2V}\right)$, f(α,M)	C1R	SFAERRF
$C_{l_{sf}}$	total roll coefficient, steady flow, about x axis in the body frame, $C_l\|_{\dot{V}=0.0}$	C1REFO	SFAERRF
$C_{l_{\Delta\delta_f}}$	change in roll coefficient per degree of differential trailing-edge flaps	C1FDDF	SFAERRF
$C_{l_{\Delta\delta_H}}$	change in roll coefficient per degree differential stabilator	DC1DDH	SFAERRF
$C_{l_{\Delta\delta_n}}$	change in roll coefficient per degree of differential leading-edge flaps	C1FDDN	SFAERRF
C_L	total lift coefficient, along -z axis in the stability frame		
C_{L_o}	basic lift coefficient, controls at zero, rigid airplane, f(α,M)	CLO	SFAERRF
C_{L_q}	lift coefficient per non-dimensional pitch rate, $\partial C_L / \partial\left(\frac{q\bar{c}}{2V}\right)$, f(α,M)	CLQ	SFAERRF
$C_{L_{sf}}$	total lift coefficient, steady flow, = $C_L\|_{\dot{V}=0.0}$	CLREFO	SFAERRF
$C_{L_{\dot{\alpha}}}$	total lift coefficient per non-dimensional rate-of-change in α, $\partial C_L / \partial\left(\frac{\dot{\alpha}\bar{c}}{2V}\right)$, f(α,M)	CLADRF	USAERRF
$C_{L_{\dot{\alpha}_o}}$	rigid lift coefficient per non-dimensional rate-of-change in α, $\partial C_L / \partial\left(\frac{\dot{\alpha}\bar{c}}{2V}\right)$(rigid airplane), f(α,M)	CLAD	USAERRF
$C_{L_{\delta_f}}$	lift coefficient per unit symmetric trailing-edge flap, rigid airplane, deg^{-1}, f(α,M)	CLDF	SFAERRF
$C_{L_{\delta_n}}$	lift coefficient per unit symmetric leading-edge flap, rigid airplane, deg^{-1}, f(α,M)	CLDN	SFAERRF

C_m	total pitch coefficient, about y axis in the body or stability frame	
C_{m_o}	basic pitch coefficient, controls at zero, rigid airplane, f(α,M)	CMO SFAERRF
C_{m_q}	pitch coefficient per non-dimensional pitch rate, rigid airplane, $\partial C_m / \partial\left(\frac{q\bar{c}}{2V}\right)$, f(α,M)	CMQ SFAERRF
$C_{m_{sf}}$	total pitch coefficient, steady flow, = $C_m\|_{\dot{V}=0.0}$	CMREFO SFAERRF
$C_{m_{\dot{\alpha}}}$	total pitch coefficient per non-dimensional rate-of-change in α, $\partial C_m / \partial\left(\frac{\dot{\alpha}\bar{c}}{2V}\right)$, f(α,M)	CMADRF USAERRF
$C_{m_{\dot{\alpha}_o}}$	rigid pitch coefficient per non-dimensional rate-of-change in α, $\partial C_m / \partial\left(\frac{\dot{\alpha}\bar{c}}{2V}\right)$(rigid airplane), f(α,M)	CMAD USAERRF
$C_{m_{\delta_f}}$	pitch coefficient per unit symmetric trailing-edge flap, rigid airplane, deg^{-1}, f(α,M)	CMDF SFAERRF
$C_{m_{\delta_n}}$	pitch coefficient per unit symmetric leading-edge flap, rigid airplane, deg^{-1}, f(α,M)	CMDN SFAERRF
C_{n_o}	basic yaw coefficient, controls zero, rigid airplane, f(β,α,M)	CNO SFAERRF
C_{n_p}	yaw coefficient per non-dimensional yaw rate, $\partial C_n/\partial\left(\frac{pb}{2V}\right)$, f(α,M)	CNP SFAERRF
C_{n_r}	yaw coefficient per non-dimensional yaw rate, $\partial C_n/\partial\left(\frac{rb}{2V}\right)$, f(α,M)	CNR SFAERRF
$C_{n_{sf}}$	total yaw coefficient, steady flow, about z axis in the body frame, $C_n\|_{\dot{V}=0.0}$	CNREFO SFAERRF
$C_{n_{\Delta\delta_f}}$	change in yaw coefficient per degree differential trailing-edge flaps, (= 0.0)	CNFDDF SFAERRF

$C_{n_{\Delta\delta_H}}$	change in yaw coefficient per degree differential stabilator, $f(\alpha, \delta_H, M)$	DCNDDH	SFAERRF
$C_{n_{\Delta\delta_n}}$	change in yaw coefficient per degree differential leading-edge flaps, (= 0.0)	CNFDDN	SFAERRF
C_{Y_o}	basic side force coefficient, controls at zero, rigid airplane, $f(\beta,\alpha,M)$	CYO	SFAERRF
C_{Y_p}	side force coefficient per non-dimensional roll rate, $\partial C_Y/\partial\left(\frac{pb}{2V}\right)$, $f(\alpha,M)$	CYP	SFAERRF
C_{Y_r}	side force coefficient per non-dimensional yaw rate, $\partial C_Y/\partial\left(\frac{rb}{2V}\right)$, $f(\alpha,M)$	CYR	SFAERRF
$C_{Y_{sf}}$	total side force coefficient, steady flow, along y axis in the body or stability frame, $C_Y\|_{\dot{V}=0.0}$	CYREFO	SFAERRF
$C_{Y_{\Delta\delta_f}}$	change in side force coefficient per degree differential trailing-edge flaps, (=0.0)	CYFDDF	SFAERRF
$C_{Y_{\Delta\delta_H}}$	change in side force coefficient per degree differential stabilator	DCYDDH	SFAERRF
$C_{Y_{\Delta\delta_n}}$	change in side force coefficient per degree differential leading-edge flaps, (= 0.0)	CYFDDN	SFAERRF
F	force, lbs, eqn. (4.1)		
FS_a	fuselage station coordinate of the accelerometers, + aft, in. (default = 306.75)	NYLOC(1)	CSL
FS_{cg}	fuselage station coordinate of c.g., + aft, in. (default = 455.0)	FSCG	CSL
FS_{eng}	fuselage station coordinate of engine/airframe interface point for right or left engine, + aft, 687.5, in.	FSENGR	CSL
FS_{rf}	fuselage station of aerodynamic reference center, + aft, 458.56 in.	FSRF	CSL

F_{x_E}	total x-body component of thrust-induced forces, lbs	FVRE(1)	CSL
F_{y_E}	total y-body component of thrust-induced forces, lbs	FVRE(2)	CSL
F_{z_E}	total z-body component of thrust-induced forces, lbs	FVRE(3)	CSL
g	acceleration due to gravity, 32.174 ft/sec^2	G	CSL
H_ρ	altitude appropriate for current density, ρ, in the 1962 Standard Atmosphere tables, feet	HRF	CSL
I_{xx}	roll moment of inertia, slug-ft^2	IXX	CSL
I_{xz}	roll-yaw cross coupling moment of inertia, slug-ft^2	IXZ	CSL
I_{yy}	pitch moment of inertia, slug-ft^2	IYY	CSL
I_{zz}	yaw moment of inertia, slug-ft^2	IZZ	CSL
l_{xz}	(1,3) entry of the 3x3 direction cosine matrix [L] that transforms a vector in the local (Earth) frame to the body frame.		
l_{yz}	(2,3) entry of the 3x3 direction cosine matrix [L] that transforms a vector in the local (Earth) frame to the body frame.		
l_{zz}	(3,3) entry of the 3x3 direction cosine matrix [L] that transforms a vector in the local (Earth) frame to the body frame.		
m	mass, slugs, eqn. (4-1)	MASS	CSL
M	Mach number		
p	+x component of rotational velocity in the body frame, rad/sec	P	CSL
P_s	static pressure (lbs/ft^2)		
q	+y component of rotational velocity in the body frame, rad/sec	Q	CSL
Q_c	compressible pressure $1.4*P_s*M^2/2.0$ (lbs/ft^2)		
r	+z component of rotational velocity in the body frame, rad/sec	R	CSL
R_i	compressible/static pressure Q_c/P_s (lbs/ft^2)		
$R^F_{Cl_p}$	elastic/rigid multiplier on C_{l_p}, (=1.0)	FRC1P	SFAERRF

$R^{F}_{Cl_{\delta_A}}$	elastic/rigid factor applied to aileron roll increment, = $R^{FLT}_{C_{L_{\delta_A}}}$	FRC1DA SFAERRF
$R^{F}_{Cl_{\delta_R}}$	elastic/rigid effectiveness factor applied to roll due to rudder increment, f(M,H_ρ)	FRC1DR SFAERRF
$R^{F}_{CL_{\delta_H}}$	elastic/rigid factor applied to stabilator lift coefficient increment, f(M,H_ρ)	FRCLDH SFAERRF
$R^{FLT}_{CL_{\delta_A}}$	flight-derived effectiveness factor applied to aileron lift coefficient increment, f(M,H_ρ)	FFCLDA SFAERRF
$R^{F}_{Cm_{\delta A}}$	elastic/rigid effectiveness factor applied to aileron pitch coefficient increment, f(M,H_ρ)	FFCMDA SFAERRF
$R^{F}_{Cm_{\delta H}}$	elastic/rigid factor applied to stabilator pitch coefficient increment, f(M,H_ρ)	FRCMDH SFAERRF
$R^{F}_{Cn_p}$	elastic/rigid multiplier on C_{n_p}, f(M,α)	FRCNP SFAERRF
$R^{F}_{Cn_{\delta A}}$	elastic/rigid factor applied to aileron yaw coefficient increment, f(M,H_ρ)	FRCNDA SFAERRF
$R^{F}_{Cn_{\delta R}}$	elastic/rigid effectiveness factor applied to yaw due to rudder increment, f(M,H_ρ)	FRCNDR SFAERRF
$R^{F}_{CY_p}$	elastic/rigid multiplier on C_{Y_p}, f(M,H_ρ)	FRCYP SFAERRF
$R^{F}_{CY_{\delta A}}$	elastic/rigid factor applied to aileron side force coefficient increment, f(M,H_ρ)	FRCYDA SFAERRF
$R^{F}_{CY_{\delta R}}$	elastic/rigid effectiveness factor applied to side force due to rudder increment, f(M,H_ρ)	FRCYDR SFAERRF

$R_{Cl_{\delta_H}}$	effectiveness factor applied to roll due to rudder accounting for stabilator interference, $f(\alpha,\delta_H)$	RC1DRH	SFAERRF
$R_{Cn_{\delta_H}}$	effectiveness factor applied to yaw due to rudder accounting for stabilator interference, $f(\alpha,\delta_H)$	RCNDRH	SFAERRF
$R_{CY_{\delta_H}}$	effectiveness factor applied to side force due to rudder accounting for stabilator interference, $f(\alpha,\delta_H)$	RCYDRH	SFAERRF
S	wing reference area used for force and moment coefficients, 400 ft^2	SWRF	CSL
t	time, seconds	T	CSL
TH_L	total thrust produced by left engine, lbs	THL	CSL
TH_R	total thrust produced by right engine, lbs	THR	CSL
$TH_{x;L}$	x-body component of force produced by left engine at the engine/airframe interface point (687.5,18.9,100.0), lbs	THXL	CSL
$TH_{x;R}$	x-body component of force produced by right engine at the engine/airframe interface point (687.5,18.9,100.0), lbs	THXR	CSL
$TH_{y;L}$	y-body component of force produced by left engine at the engine/airframe interface point (687.5,18.9,100.0), lbs	THYL	CSL
$TH_{y;R}$	y-body component of force produced by right engine at the engine/airframe interface point (687.5,18.9,100.0), lbs	THYR	CSL
$TH_{z;L}$	z-body component of force produced by left engine at the engine/airframe interface point (687.5,18.9,100.0), lbs	THZL	CSL
$TH_{z;R}$	z-body component of force produced by right engine at the engine/airframe interface point (687.5,18.9,100.0), lbs	THZR	CSL
u	+x component of inertial translational velocity in the body frame, ft/sec	U	CSL
v	+y component of inertial translational velocity in the body frame, ft/sec	V	CSL
V	total airspeed, ft/sec	VT	CSL
w	+z component of inertial translational velocity in the body frame, ft/sec	W	CSL

WL_a	waterline coordinate of accelerometers, + up, in. (default = 91.3)	NYLOC(3)	CSL
WL_{cg}	waterline coordinate of c.g., + up, in. (default = 102.8)	WLCG	CSL
WL_{eng}	waterline coordinate of engine/airframe interface point for right or left engine, + up, 100.0, in.	WLENG	CSL
WL_{rf}	waterline coordinate of aerodynamic reference center, + up, 100.0 in.	WLRF	CSL
x_{arf}	distance from c.g. to the a.r.c. along X-body axis, + forward, $= -(FS_{rf} - FS_{cg})/12$, ft	XARF	CSL
x_{erf}	distance from c.g. to the engine/airframe interface point along x-body axis, + forward, $= -(FS_{eng} - FS_{cg})/12$, ft	XERF	CSL
y_{arf}	distance from c.g. to the a.r.c. along Y-body axis, + right, $= (BL_{rf} - BL_{cg})/12$, ft	YARF	CSL
y_{erf_L}	distance from c.g. to the left engine/airframe interface point along y-body axis, + right, $= (BL_{eng_L} - BL_{cg})/12$, ft	YERFL	CSL
y_{erf_R}	distance from c.g. to the right engine/airframe interface point along y-body axis, + right, $= (BL_{eng_R} - BL_{cg})/12$, ft	YERFR	CSL
z_{arf}	distance from c.g. to the a.r.c. along Z-body axis, + down, $= -(WL_{rf} - WL_{cg})/12$, ft	ZARF	CSL
z_{erf}	distance from c.g. to the engine/airframe interface point along z-body axis, + down, $= -(WL_{eng} - WL_{cg})/12$, ft	YERF	CSL
α	angle of attack, degrees	ALPDEG ALDGRF	SFAERRF CSL
β	sideslip angle, degrees	BETDEG BEDGRF	SFAERRF CSL
$\Delta C_{D_{C_L}}$	increment in drag coefficient due to lift, $f(C_{L_{sf}}, M)$	DCDCL	SFAERRF
$\Delta C_{D_{\delta_{H_L}}}$	increment in drag coefficient due to left stabilator position, rigid airplane, $f(\alpha, \delta_{H_L}, M)$	DCDDHL	SFAERRF

$\Delta C_{D\delta_{H_R}}$	increment in drag coefficient due to right stabilator position, rigid airplane, $f(\alpha,\delta_{H_R},M)$	DCDDHR SFAERRF
$\Delta C_{D\delta_{LG}}$	increment to drag coefficient for fully deployed landing gear, $f(\alpha)$	DCDDLG SFAERRF
$\Delta C_{D\delta_R}$	increment in drag coefficient due to antisymmetric rudder, (=0.0)	DCDDR SFAERRF
$\Delta C_{D\delta_{RT}}$	increment in drag coefficient due to symmetric (toe-in) rudder, $f(\alpha, \delta_{RT}, \delta_{f/A})$	DCDDRT SFAERRF
$\Delta C_{D\delta_{SB}}$	increment to drag coefficient for full 60° deflection in speed brakes, $f(\alpha,M)$	DCDSB SFAERRF
$\Delta C_{l_{asm}}$	roll increment due to airplane asymmetries, $f(\alpha)$	DC1ASM SFAERRF
$\Delta C_{l_r}^F$	increment to C_{l_r} due to elastic deformation, $f(M,H_\rho)$	DC1RF SFAERRF
$\Delta C_{l_\beta}^F$	increment to C_{l_β} due to elastic deformation, $f(M,H_\rho)$	DC1BF SFAERRF
$\Delta C_{l_{\delta_{A_L}}}$	increment in roll coefficient due to left aileron position, rigid airplane, $f(\alpha,\delta_{A_L},M)$	DC1DAL SFAERRF
$\Delta C_{l_{\delta_{A_R}}}$	increment in roll coefficient due to right aileron position, rigid airplane, $f(\alpha,\delta_{A_R},M)$	DC1DAR SFAERRF
$\Delta C_{l_{\delta_f}}$	increment in roll coefficient for full 20 degree deployment of symmetric trailing-edge flaps, rigid airplane, used with sign of β, $f(\lvert\beta\rvert,\alpha,M)$	DC1DF SFAERRF
$\Delta C_{l_{\delta_n}}$	increment in roll coefficient for full 25 degree deployment of symmetric leading-edge flaps, rigid airplane, used with sign of β, $f(\lvert\beta\rvert,\alpha,M)$	DC1DN SFAERRF

$\Delta C_{l_{\delta_R}}$	increment in roll coefficient due to antisymmetric rudder, rigid airplane, used with sign of $\delta_{R_{as}}$, $f(\alpha,\lvert\delta_{R_{as}}\rvert,M)$	DC1DR	SFAERRF
$\Delta C_{l_{\delta_{SB}}}$	increment to roll coefficient for full 60° deflection in speed brakes per degree of sideslip, deg^{-1}, $f(\alpha,\lvert\beta\rvert,M)$	DC1SB	SFAERRF
$\Delta C_{L_o}^F$	increment to basic lift coefficient due to elastic deformation, $f(\alpha,M,H_\rho)$	DCLOF	SFAERRF
$\Delta C_{L_q}^F$	increment to C_{L_q} due to elastic deformation	DCLQF	SFAERRF
$\Delta C_{L_{\dot\alpha}}^F$	elastic increment to $C_{L_{\dot\alpha}}$, $f(\alpha,H)$	DCLADF	USAERRF
$\Delta C_{L_{\delta_f}}^F$	increment to $C_{L_{\delta_f}}$ due to elastic deformation, deg^{-1} ,$f(\alpha,M,H_\rho)$	DCLDFF	SFAERRF
$\Delta C_{L_{\delta_n}}^F$	increment to $C_{L_{\delta_n}}$ due to elastic deformation, deg^{-1} ,$f(\alpha,M,H_\rho)$	DCLDNF	SFAERRF
$\Delta C_{L_{\delta_{A_L}}}$	increment in lift coefficient due to left aileron, rigid airplane, $f(\alpha,\delta_{A_L},M)$	DCLDAL	SFAERRF
$\Delta C_{L_{\delta_{A_R}}}$	increment in lift coefficient due to right aileron, rigid airplane, $f(\alpha,\delta_{A_L},M)$	DCLDAR	SFAERRF
$\Delta C_{L_{\delta_{H_L}}}$	increment in lift coefficient due to left stabilator position, rigid airplane, $f(\alpha,\delta_{H_L},M)$	DCLDHL	SFAERRF
$\Delta C_{L_{\delta_{H_R}}}$	increment in lift coefficient due to right stabilator position, rigid airplane, $f(a,\delta_{H_R},M)$	DCLDHR	SFAERRF
$\Delta C_{L_{\delta_{LG}}}$	increment to lift coefficient for fully deployed landing gear, $f(\alpha)$	DCLDLG	SFAERRF

$\Delta C_{L_{\delta_R}}$	increment in lift coefficient due to antisymmetric rudder, (=0.0)	DCLDR	SFAERRF
$\Delta C_{L_{\delta_{RT}}}$	increment in lift coefficient due to symmetric rudder (rudder toe-in), $f(\alpha,\delta_{RT},\delta_{f/A})$	DCLDRT	SFAERRF
$\Delta C_{L_{\delta_{SB}}}$	increment to lift coefficient for full 60 degree deflection in speed brakes, $f(\alpha,M)$	DCLSB	SFAERRF
$\Delta C^F_{m_o}$	increment to basic pitch coefficient due to elastic deformation, $f(\alpha,M,H_\rho)$	DCMOF	SFAERRF
$\Delta C^F_{m_q}$	increment to C_{m_q} due to elastic deformation	DCMQF	SFAERRF
$\Delta C^F_{m_{\dot\alpha}}$	elastic increment to $C_{m_{\dot\alpha}}$, $f(\alpha,H)$	DCMADF	USAERRF
$\Delta C^F_{m_{\delta_f}}$	increment to $C_{m_{\delta_f}}$ due to elastic deformation, deg^{-1} ,$f(\alpha,M,H_\rho)$	DCMDDF	SFAERRF
$\Delta C^F_{m_{\delta_n}}$	increment to $C_{m_{\delta_n}}$ due to elastic deformation, deg^{-1} ,$f(\alpha,M,H_\rho)$	DCMDNF	SFAERRF
$\Delta C_{m_{LEX}}$	increment in pitch due to the leading-edge wing extensions, $f(\alpha,M)$	DCMLEX	SFAERRF
$\Delta C_{m_{\delta_{A_L}}}$	increment in pitch coefficient due to left aileron, rigid airplane, deg^{-1}, $f(\alpha,\delta_{A_L},M)$	DCMDAL	SFAERRF
$\Delta C_{m_{\delta_{A_R}}}$	increment in pitch coefficient due to right aileron, rigid airplane, deg^{-1}, $f(\alpha,\delta_{A_L},M)$	DCMDAR	SFAERRF
$\Delta C_{m_{\delta_{H_L}}}$	increment in pitch coefficient due to left stabilator position, rigid airplane, $f(\alpha,\delta_{H_L},M)$	DCMDHL	SFAERRF

$\Delta C_{m_{\delta_{H_R}}}$	increment in pitch coefficient due to right stabilator position, rigid airplane, $f(\alpha,\delta_{H_R},M)$	DCMDHR	SFAERRF
$\Delta C_{m_{\delta_{LG}}}$	increment to pitch coefficient for fully deployed landing gear, $f(\alpha)$	DCMDLG	SFAERRF
$\Delta C_{m_{\delta_R}}$	increment in pitch coefficient due to antisymmetric rudder, $f(\alpha,\lvert\delta_R\rvert,M)$	DCMDR	SFAERRF
$\Delta C_{m_{\delta_{RT}}}$	increment in pitch coefficient due to symmetric rudder (rudder toe-in), $f(\alpha,\delta_{RT},\delta_{f/A})$	DCMDRT	SFAERRF
$\Delta C_{m_{\delta_{SB}}}$	increment to pitch coefficient for full 60 degree deflection in speed brakes, $f(\alpha,M)$	DCMSB	SFAERRF
$\Delta C^F_{n_r}$	increment to C_{n_r} due to elastic deformation, $f(M,H_\rho)$	DCNRF	SFAERRF
$\Delta C^F_{n_\beta}$	increment to C_{n_β} due to elastic deformation, $f(M,H_\rho)$	DCNBF	SFAERRF
$\Delta C_{n_{\delta_{A_L}}}$	increment in yaw coefficient due to left aileron position, rigid airplane, $f(\alpha,\delta_{A_L},M)$	DCNDAL	SFAERRF
$\Delta C_{n_{\delta_{A_R}}}$	increment in yaw coefficient due to right aileron position, rigid airplane, $f(\alpha,\delta_{A_R},M)$	DCNDAR	SFAERRF
$\Delta C_{n_{\delta_f}}$	increment in yaw coefficient for full 20 degree deployment of symmetric trailing-edge flaps, rigid airplane, used with sign of β, $f(\lvert\beta\rvert,\alpha,M)$	DCNDF	SFAERRF
$\Delta C_{n_{\delta_n}}$	increment in yaw coefficient for full 25 degree deployment of symmetric leading-edge flaps, rigid airplane, used with sign of β, $f(\lvert\beta\rvert,\alpha,M)$	DCNDN	SFAERRF

$\Delta C_{n_{\delta_R}}$	increment in yaw coefficient due to antisymmetric rudder, rigid airplane, used with sign of $\delta_{R_{as}}$, $f(\alpha,\lvert\delta_{R_{as}}\rvert,M)$	DCNDR	SFAERRF
$\Delta C_{n_{\delta_{SB}}}$	increment to yaw coefficient for full 60 degree deflection in speed brakes per degree of sideslip, deg^{-1}, $f(\alpha,\lvert\beta\rvert,M)$	DCNSB	SFAERRF
$\Delta C^F_{Y_r}$	increment to C_{Y_r} due to elastic deformation, $f(M,H_\rho)$	DCYRF	SFAERRF
$\Delta C^F_{Y_\beta}$	increment to C_{Y_β} due to elastic deformation, $f(M,H_\rho)$	DCYBF	SFAERRF
$\Delta C_{Y_{\delta_{A_L}}}$	increment in side force coefficient due to left aileron position, rigid airplane, $f(\alpha,\delta_{A_L},M)$	DCYDAL	SFAERRF
$\Delta C_{Y_{\delta_{A_R}}}$	increment in side force coefficient due to right aileron position, rigid airplane, $f(\alpha,\delta_{A_R},M)$	DCYDAR	SFAERRF
$\Delta C_{Y_{\delta_f}}$	increment in side force coefficient for full 20 degree deployment of symmetric trailing-edge flaps, rigid airplane, used with sign of β, $f(\lvert\beta\rvert,\alpha,M)$	DCYDF	SFAERRF
$\Delta C_{Y_{\delta_n}}$	increment in side force coefficient for full 25 degree deployment of symmetric leading-edge flaps, rigid airplane, used with sign of β, $f(\lvert\beta\rvert,\alpha,M)$	DCYDN	SFAERRF
$\Delta C_{Y_{\delta_R}}$	increment in side force coefficient due to antisymmetric rudder, rigid airplane, used with sign of $\delta_{R_{as}}$, $f(\alpha,\lvert\delta_{R_{as}}\rvert,M)$	DCYDR	SFAERRF
$\Delta C_{Y_{\delta_{SB}}}$	increment to side force coefficient for full 60 degree deflection in speed brakes per degree of sideslip, deg^{-1}, $f(\alpha,\lvert\beta\rvert,M)$	DCYSB	SFAERRF
θ	pitch angle, radians	THE	CSL
ρ	density of the air, $slugs/ft^3$	RHO	CSL
Φ	roll angle, radians	PHI	CSL

Ψ	yaw angle, radians	PSI	CSL
Ψ_{eng_L}	left engine cant angle in x-y plane, pos nose right, (–1.98°)	AZEL	CSL
Ψ_{eng_R}	right engine cant angle in x-y plane, pos nose right, (1.98°)	AZER	CSL

Subscripts

a	aerodynamic, with respect to the air mass
A	due to air loads
asm	due to asymmetries
E,eng	engine induced
G	due to gravity
rf	quantity computed at the aerodynamic reference center (a.r.c.)
wg	component due to winds, turbulence, or gust
x	component along the X_B (body frame) axis
y	component along the Y_B (body frame) axis
z	component along the Z_B (body frame) axis

Notation

{ }	column vector
[]	matrix or row vector
[ ]'	transpose of [ ]
$x[_b^a$	the variable x restricted to the range, $b \leq x \leq a$, typically used to describe a function argument that is limited.

Acronyms

ACSL	Advanced Continuous Simulation Language
AFU	Auto-Flap Up mode of the F/A-18 Flight Control System (8.3.3 PROM)
AGCB	Aircraft Guidance and Controls Branch at NASA LaRC

BL	Buttock line coordinate, + right
CAS	Control Augmentation System
CDC	Control Data Corporation
DMS	Differential Maneuvering Simulator twin-dome facility at NASA LaRC
dmsf18	Real-time simulation program of an F/A-18 developed at LaRC for use in the DMS facility.
FIT	Functional Integration Technology
FS	Fuselage station coordinate, + aft
f18bas	Simulation model and computer program described in this document
f18harv	Simulation model and computer program resulting from extensive modifications to the *f18bas* program and in support of the HARV research program
HARV	High-Angle-of-Attack Research Vehicle
LaRC	Langley Research Center
LEX	Leading Edge Extension
MDC	McDonnel Douglas Corporation
mdcf18	Batch simulation program of an F/A-18 developed by MDC and obtained for use at LaRC around 1986.
PROM	Programmable Read Only Memory
PSF	pounds per square foot
RSRI	Roll surface to rudder interconnect
SPIN	Spin mode of AFU in F-18 8.3.3 PROM set
STORES	Logical for wing stores on or off (Always off)
WL	Waterline coordinate, + up

APPENDIX A - SIMULATION VARIABLES

Simulation Variable Naming Plan

Development of the simulation model involved the efforts of several engineers and programmers and has included receipt of data, documentation, and several Fortran subroutines from MDC and other sources. This effort has given the authors a keen sense of the desirability of a "standard" variable nomenclature for coding simulation models. Without a "standard" variable nomenclature, it was generally impossible to follow the coded logic without having the documentation in hand. This section describes our efforts at maintaining a consistent variable naming strategy. The creation of meaningful variable names was made challenging by the restriction in ACSL version 9 to six character variable names. ACSL version 10 supports longer variable names.

The authors developed a system for naming variables. This should not be confused with issuing a variable dictionary for project personnel to use. The philosophy behind the variable nomenclature system was that any given variable name could be constructed from a set of predefined mnemonics (base names; prefix and suffix modifiers), given an appropriate set of construction rules. This philosophy is based upon ideas enunciated by Mitchell and documented in Arbuckle (1987). A set of mnemonics developed by the authors is listed in tables A.1 through A.3. The construction rules for creating a variable name are as follows:

1) In keeping with the ANSI Fortran 77 standard and the requirements of ACSL, all variable names are six characters or less.
2) Prefix and suffix modifiers may only be used in combination with a base name.
3) Multiple prefixes may only be used if their nesting levels are different. The highest nesting level prefix will be leftmost, followed by the base name.
4) Multiple suffixes may only be used if their nesting levels are different. The highest nesting level suffix will be rightmost.
5) Base name truncation may be used to meet the requirement that all names be six characters or less. If truncation is required, remove the right character of the base name.

As an example of using the nomenclature system and construction rules, consider a variable defining β in units of degrees at the aerodynamic reference point. The base name for β is BET, the suffixes are DG and RF, respectively. The suffix DG has a nesting level of two, and the suffix RF has a nesting level of three (see table A.3). Constructing the variable name using rule number five results in BETDGRF, which is a seven-character name and violates rule number one. Using rule number six, remove the right character of BET to give us BEDGRF as the variable name which satisfies the construction rules.

When simulation model components were derived from preexisting code, the authors were reluctant to totally discard the variable name structure already in place. Not all of the variable names in the simulation were constructed according to the rules of this section.

Table A.1 Base Variable Naming Plan

Base Name	Description
ALF	angle of attack, α, radians
BET	sideslip angle, β, radians
C	control surface command, degrees
C*xx*	direction cosine, *xx* = {XX, XY, XZ, YX, ... ,ZY, ZZ}
D	control surface deflection, degrees
DG2RD	degrees to radians conversion factor (π/180), rad/°
E*n*	quaternion, *n = 0,1,2,3*
GAM	longitudinal flight path angle, radians
H	-z component of position in Earth frame (altitude above mean sea level), ft
L	logical or integer switch
MACH	Mach number
P	x body component of rotational velocity, rad/sec
PHI	Euler bank angle, ψ, radians
PSI	Euler yaw angle, ϕ, radians
Q	y body component of rotational velocity, rad/sec
QBAR	dynamic pressure, $\bar{q}$, lb/ft^2
R	z body component of rotational velocity, rad/sec
RD2DG	radians to degrees (180/π), °/rad
RHO	air density, slugs/ft^3
TH	total thrust, lb
THE	Euler pitch angle, θ, radians
U	x body frame component of translational velocity, ft/sec
V	y body frame component of translational velocity, ft./sec
VT	total velocity magnitude (inertial), ft/sec
W	z body frame component of translational velocity, ft/sec
X	x component of position in earth frame, ft
Y	y component of position in earth frame, ft

Table A.2 Prefix Naming Plan

Base Name	Description
CS	trigonometric cosine of x
R	ratio
SN	trigonometric sine of x

Table A.3 Suffix Naming Plan

Suffix	Level	Description
0	2	intermediate value of a state
2	2	quantity has been squared
A	1	applies to aileron
A	2	aerodynamic
B	2	bias
CG	2	airplane center-of-gravity
D	1	d/dt, time rate of change
E	2	engine, thrust related
F	1	applies to trailing-edge flaps
I	2	inertial
IC	5	initial condition, variable is in the state variable i.c. array
L	4	lift
LL	2	lower limit
LM	2	limit
LN	5	dummy variable used only to act as a possible input for ACSL ANALYZ functions
MN	2	minimum
MX	2	maximum
P	2	probe
PLA	1	power lever angle
R	2	rudder
R	4	right
RA	2	variable has units of radians or rad/sec
RF	3	reference quantity
S	1	stabilator
S	2	sensor
SB	1	speed brake
SG	4	signal
SL	2	sea level
SY	3	symmetric
TR	5	value calculated for trim, may not be a state variable
UL	2	upper limit
WG	2	aerodynamic disturbance, wind or gust
X	2	x body frame component
Y	2	y body frame component
Z	2	z body frame component
ZR	2	"zero", reference to an i.c. or trim state

Simulation Dictionary Option

ACSL provides for maintaining a simulation dictionary. If the logical LVDEF in the simulation is set true at run time, a call to LISTD(29) is made in the terminal block of the translator source code. Calling LISTD(29) causes a list of variable names and definitions to be read from unit 29 and written to the "PRN" output unit along with the values of all the variables in the simulation [ACSL (1987)]. All variables and their values are listed on the "PRN" output unit. The variables not found on the dictionary (unit 29) are simply listed without definitions. The ACSL executive program is not aware of any variables internal to called Fortran subroutines.

List of Simulation Variables

A81	left engine nozzle exit area, in^2
A82	right engine nozzle exit area, in^2
ABL	marker for left afterburner "state update" discrete block
ABLLL	left engine afterburner thrust lower rate limit, lbs/sec
ABLLR	right engine afterburner thrust lower rate limit, lbs/sec
ABR	marker for right afterburner "state-update" discrete block
ABSWL	.true. if left engine afterburner is activated
ABSWR	.true. if right engine afterburner is activated
ABULL	left engine afterburner thrust upper rate limit, lbs/sec
ABULR	right engine afterburner thrust upper rate limit, lbs/sec
ALDGRF	angle of attack at the aerodynamic reference center (a.r.c.), degrees
ALF	angle of attack at the c.g., radians
ALFB	air data probe local angle-of-attack bias
ALFD	d/dt of ALF, rad/sec
ALFDG	angle of attack at the c.g., degrees
ALFDP	d/dt of ALFP, degree/sec
ALFLM	limited angle of attack, used in Pitot tube sensor model, degrees
ALFP	air data probe angle of attack, degrees
ALFPIC	air data probe angle-of-attack initial condition, degrees
ALFS	sensed angle of attack, equal to ALFS0 or ALFS1 based ISENS = 0 or 1
ALFS0	actual or true angle of attack, equal to ALFDG, degrees
ALFS1	output of air data models, degrees
ALFTR	angle of attack required for trim, radians
AX	acceleration in x-body direction, g's
AY	acceleration in y-body direction, g's
AYS	sensed lateral acceleration, AY, equal to AYS0 or AYS1 based on ISENS = 0 or 1, g's
AYS0	true lateral acceleration, = AY, g's
AYS1	output of lateral accelerometer sensor model, g's
AYW	wind axis lateral acceleration, ft/sec^2
AZ	acceleration in z-body direction, g's
AZEL	azimuth angle of left engine as installed, 1.98 degrees
AZER	azimuth angle of right engine as installed, 1.98 degrees
AZS	sensed normal acceleration, equal to AZS0 or AZS1 based on ISENS, g's
AZS0	true normal acceleration, equal to AZ, g's
AZS1	output of normal accelerometer sensor model, g's
AZW	wind axis normal acceleration, ft/sec^2
BEDGRF	sideslip angle at the aerodynamic reference center (a.r.c.), degrees
BET	sideslip angle at the c.g., radians
BETD	d/dt of BET, rad/sec
BETDG	sideslip angle at the c.g., degrees
BETTR	value of BET required for trim, calculated in trim search, radians
BLCG	buttock line coordinate of the center of gravity, inches
BLENGL	buttock line coordinate of the engine/airframe interface, left engine, –18.9 in.
BLENGR	buttock line coordinate of the engine/airframe interface, right engine, 18.9 in.
BLRF	buttock line coordinate of the aerodynamic reference center, 0.00 inches
BWRF	reference wing span used for nondimensional aero coefficients, 37.42 ft
C1RFSF	total steady-flow roll coefficient about x-body at a.r.c., n.d.
CAASLN	commanded antisymmetric aileron, linearization input only, degrees
CAASP	dynamic check perturbation signal to antisymmetric aileron command, degrees
CAL	commanded left aileron, degrees
CAR	commanded right aileron, degrees

CASYLN	commanded antisymmetric aileron, linearization input only, degrees
CASYP	dynamic check perturbation signal to symmetric aileron command, degrees
CATL	left power lever angle command from auto throttle, degrees
CATR	right power lever angle command from auto throttle, degrees
CDRFSF	drag coefficient at aerodynamic reference center, steady flow ($\dot{\alpha} = 0$), n.d.
CFASLN	commanded antisymmetric trailing-edge flap, linearization input only, degrees
CFASP	dynamic check perturbation signal to antisymmetric trailing-edge flap command, degrees
CFL	command to left trailing-edge flap actuator, degrees
CFR	command to right trailing-edge flap actuator, degrees
CFSYLN	commanded symmetric trailing-edge flap, linearization input only, degrees
CFSYP	dynamic check perturbation signal to symmetric trailing-edge flap command, degrees
CINT	simulation communication interval, time interval at which data is written to the master time history file or available at the screen, seconds
CLRFSF	lift coefficient at aerodynamic reference center, steady flow ($\dot{\alpha} = 0$), n.d.
CLTR	Target trim value for CLRFSF if "12" is in active part of IYSEL array.
CMRFSF	pitch (about y-body axis) coefficient at aerodynamic reference point, steady flow ($\dot{\alpha} = 0$), n.d.
CNASLN	commanded antisymmetric leading-edge flap, linearization input only, degrees
CNASP	dynamic check perturbation signal to antisymmetric leading-edge flap command, degrees
CNL	command to left leading-edge flap actuator, degrees
CNR	command to right leading-edge flap actuator, degrees
CNRFSF	yaw (about z-body axis) coefficient at aerodynamic reference point, steady flow ($\dot{\alpha} = 0$), n.d.
CNSYLN	commanded symmetric leading-edge flap, linearization input only, degrees
CNSYP	dynamic check perturbation signal to symmetric leading-edge flap command, degrees
CO2VT	intermediate quantity, $\frac{\bar{c}}{2V}$, sec
CPASLN	commanded antisymmetric power lever angle, linearization input only, degrees
CPASP	dynamic check perturbation signal to antisymmetric power lever angle command, degrees
CPL	left power lever angle command from the pilot, degrees
CPLTR	left power lever angle command from the pilot required to for trim, degrees
CPR	right power lever angle command from the pilot, degrees
CPRTR	right power lever angle command from the pilot required to for trim, degrees
CPSYLN	commanded symmetric power lever angle, linearization input only, degrees
CPSYP	dynamic check perturbation signal to symmetric power lever angle command, degrees
CPSYSG	commanded symmetric power lever angle, signal array used in IDYNCK=3 option wherein an arbitrary input stream can be constructed
CRASLN	commanded antisymmetric rudder (both t.e.left), linearization input only, degrees
CRASP	dynamic check perturbation signal to antisymmetric rudder, degrees
CRL	command to left rudder actuator, degrees
CRR	command to right ruder actuator, degrees
CRSYLN	commanded symmetric rudder, linearization input only, degrees
CRSYP	dynamic check perturbation signal to symmetric rudder command, degrees
CSALF	cosine of angle of attack at the c.g., $\cos(\alpha)$

CSALRF	cosine of angle of attack at the a.r.c., $\cos(\alpha_{ref})$
CSASLN	commanded antisymmetric stabilator, linearization input only, degrees
CSASP	dynamic check perturbation signal to antisymmetric stabilator command, degrees
CSAZEL	cosine of azimuth angle of left engine as installed, cos(AZEL)
CSAZER	cosine of azimuth angle of right engine as installed, cos(AZER)
CSB	command to speed brake actuator, degrees
CSBET	cosine of sideslip angle at the c.g.
CSL	command to left stabilator actuator, degrees
CSPHI	cosine of the roll angle PHI
CSPHO2	cos(PHITR / 2)
CSPSO2	cos(PSITR / 2)
CSR	command to right stabilator actuator, degrees
CSSYLN	commanded symmetric stabilator, linearization input only, degrees
CSSYP	dynamic check perturbation signal to symmetric stabilator command, degrees
CSTHE	cosine of the pitch angle, $\cos(\theta)$
CSTHIV	= 1. / CSTHLM
CSTHLM	= $\cos(\theta)$ if $\lvert\cos(\theta)\rvert$ is larger than EPS, otherwise equal to EPS with the sign of $\cos(\theta)$. EPS = 10^{-10}
CSTHO2	cos(THETR / 2)
CSTVOA	cosine of the thrust vectoring vane orientation angle
CTVLL	lower left thrust vector vane deflection command, degrees
CTVLU	upper left thrust vector vane deflection command, degrees
CTVRL	lower right thrust vector vane deflection command, degrees
CTVRU	upper right thrust vector vane deflection command, degrees
CWRF	aerodynamic wing reference chord, $\bar{c}$, 11.52 ft
CXRFSF	aerodynamic axial force (x-body) coefficient at a.r.p., steady flow ($\dot{\alpha} = 0$), n.d.
CXX	direction cosine, x body component of unit x Earth frame vector
CXY	direction cosine, x body component of unit y Earth frame vector
CXZ	direction cosine, x body component of unit z Earth frame vector
CYRFSF	aerodynamic side force (y-body) coefficient at a.r.p., steady flow ($\dot{\alpha} = 0$), n.d.
CYX	direction cosine, y body component of unit x Earth frame vector
CYY	direction cosine, y body component of unit y Earth frame vector
CYZ	direction cosine, y body component of unit z Earth frame vector
CZRFSF	aerodynamic normal force (z-body) coefficient at a.r.p., steady flow ($\dot{\alpha} = 0$), n.d.
CZX	direction cosine, z body component of unit x Earth frame vector
CZY	direction cosine, z body component of unit y Earth frame vector
CZZ	direction cosine, z body component of unit z Earth frame vector
DA0L	position output of left aileron actuator, degrees
DA0R	position output of right aileron actuator, degrees
DAAS	antisymmetric aileron deflection, degrees
DAASLN	component of total antisymmetric aileron deflection, linearization input only, degrees
DAASTR	antisymmetric aileron deflection resulting from trim calculations, degrees
DADL	d/dt of left aileron, degrees/sec
DADR	d/dt of right aileron, degrees/sec
DAL	total left aileron position, includes linearization input, degrees
DALIC	left aileron deflection initial condition, degrees
DALLN	component of total left aileron deflection, linearization input only, degrees
DALLN1	= DALLN (if .not.LLNCSY) or (DASYLN - DAASLN) (if LLNCSY).

DAR	total right aileron position, includes linearization input, degrees
DARIC	right aileron deflection initial condition, degrees
DARLN	component of total right aileron deflection, linearization input only, degrees
DARLN1	= DARLN (if .not.LLNCSY) or (DASYLN + DAASLN) (if LLNCSY).
DASY	symmetric aileron deflection, degrees
DASYLN	component of total symmetric aileron deflection, linearization input only, degrees
DASYTR	symmetric aileron deflection resulting from trim calculations, degrees
DATMBT	aileron trim button, pilot input, (1.,0.,-1)
DF0L	position output of left trailing-edge flap actuator, degrees
DF0R	position output of right trailing-edge flap actuator, degrees
DFAS	antisymmetric trailing-edge flap deflection, degrees
DFASLN	component of total antisymmetric trailing-edge flap deflection, linearization input only, degrees
DFASTR	antisymmetric trailing-edge flap deflection resulting from trim calculations, degrees
DFDL	d/dt of left trailing-edge flap, degrees/sec
DFDR	d/dt of right trailing-edge flap, degrees/sec
DFL	total left trailing-edge flap position, includes linearization input, degrees
DFLIC	left trailing-edge flap deflection initial condition, degrees
DFLLN	component of total left trailing-edge flap deflection, linearization input only, degrees
DFLLN1	= DFLLN (if .not.LLNCSY) or (DFSYLN - DFASLN) (if LLNCSY).
DFR	total right trailing-edge flap position, includes linearization input, degrees
DFRIC	right trailing-edge flap deflection initial condition, degrees
DFRLN	component of total right trailing-edge flap deflection, linearization input only, degrees
DFRLN1	= DFRLN (if .not.LLNCSY) or (DFSYLN + DFASLN) (if LLNCSY).
DFSY	symmetric trailing-edge flap deflection, degrees
DFSYLN	component of total symmetric trailing-edge flap deflection, linearization input only, degrees
DFSYTR	symmetric trailing-edge flap deflection resulting from trim calculations, degrees
DG2RA	degrees to radians conversion, (π/180)
DLG	landing gear deployment; assumes values in the interval [0.0,1.0] where 0.0 represents full up, and 1.0 represents full down. **Default is full up.**
DLGPCT	% DLG, (AOA probe)
DLTVCL	left overall thrust vectoring vane deflection command, degrees
DLTVCR	right overall thrust vectoring vane deflection command, degrees
DN0L	position output of left leading-edge flap actuator, degrees
DN0R	position output of right leading-edge flap actuator, degrees
DNAS	antisymmetric leading-edge flap deflection, degrees
DNASLN	component of total antisymmetric leading-edge flap deflection, linearization input only, degrees
DNASTR	antisymmetric leading-edge flap deflection resulting from trim calculations, degrees
DNDL	d/dt of left leading-edge flap, degrees/sec
DNDR	d/dt of right leading-edge flap, degrees/sec
DNL	total left leading-edge flap position, includes linearization input, degrees
DNLIC	left leading-edge flap deflection initial condition, degrees
DNLLN	component of total left leading-edge flap deflection, linearization input only, degrees
DNLLN1	= DNLLN (if .not.LLNCSY) or (DNSYLN - DNASLN) (if LLNCSY).
DNR	total right leading-edge flap position, includes linearization input, degrees
DNRIC	right leading-edge flap deflection initial condition, degrees
DNRLN	component of total right leading-edge flap deflection, linearization input only, degrees
DNRLN1	= DNRLN (if .not.LLNCSY) or (DNSYLN + DNASLN) (if LLNCSY).

DNSY	symmetric leading-edge flap deflection, degrees
DNSYLN	component of total symmetric leading-edge flap deflection, linearization input only, degrees
DNSYTR	symmetric leading-edge flap deflection resulting from trim calculations, degrees
DPASLN	component of total antisymmetric power lever angle input, linearization only, degrees
DPASTR	antisymmetric power lever angle deflection resulting from trim calculations, degrees
DPLDL	d/dt of left throttle boost output signal, degrees/sec
DPLDR	d/dt of right throttle boost output signal, degrees/sec
DPLL	left throttle boost output signal, degrees
DPLLIC	left throttle boost output signal initial condition, degrees
DPLLN	added to DPPL, linearization input only, degrees
DPLLN1	= DPLLN (if .not.LLNCSY) or (DPSYLN - DPASLN) (if LLNCSY).
DPLR	right throttle boost output signal, degrees
DPLRIC	right throttle boost output signal initial condition, degrees
DPRLN	added to DPPR, linearization input only, degrees
DPRLN1	= DPRLN (if .not.LLNCSY) or (DPSYLN + DPASLN) (if LLNCSY).
DPSYLN	component of total symmetric throttle boost output signal, linearization input only, degrees
DPSYTR	symmetric power lever angle deflection resulting from trim calculations, degrees
DR0L	position output of left rudder actuator, degrees
DR0R	position output of right rudder actuator, degrees
DRAS	antisymmetric rudder deflection, both t.e. left, degrees
DRASLN	component of total antisymmetric rudder deflection, linearization input only, degrees
DRASTR	antisymmetric rudder deflection resulting from trim calculations, degrees
DRDL	d/dt of left rudder, degrees/sec
DRDR	d/dt of right rudder, degrees/sec
DRL	total left rudder position (pos t.e.left), includes linearization input, degrees
DRLIC	left rudder deflection initial condition, degrees
DRLLN	component of total left rudder deflection, linearization input only, degrees
DRLLN1	= DRLLN (if .not.LLNCSY) or (DRSYLN - DRASLN) (if LLNCSY).
DRR	total right rudder position (pos t.e.left), includes linearization input, degrees
DRRIC	right rudder deflection initial condition, degrees
DRRLN	component of total right rudder deflection, linearization input only, degrees
DRRLN1	= DRRLN (if .not.LLNCSY) or (DRSYLN + DRASLN) (if LLNCSY).
DRSY	symmetric rudder deflection, both t.e. in, degrees
DRSYLN	component of total symmetric rudder deflection, linearization input only, degrees
DRSYTR	symmetric rudder deflection resulting from trim calculations, degrees
DS0L	position output of left stabilator actuator, degrees
DS0R	position output of right stabilator actuator, degrees
DSAS	antisymmetric stabilator deflection, degrees
DSASLN	component of total antisymmetric stabilator deflection, linearization input only, degrees
DSASTR	antisymmetric stabilator deflection resulting from trim calculations, degrees
DSB	speed brake deflection, degrees
DSBD	d/dt DSB, degrees/sec
DSBIC	initial condition for DSB, degrees
DSBLN	component of total speed brake deflection, linearization input only, degrees
DSDL	d/dt of left stabilator, degrees/sec
DSDR	d/dt of right stabilator, degrees/sec
DSL	total left stabilator position, includes linearization input, degrees
DSLIC	left stabilator deflection initial condition, degrees
DSLLN	component of total left stabilator deflection, linearization input only, degrees

DSLLN1	= DSLLN (if .not.LLNCSY) or (DSSYLN - DSASLN) (if LLNCSY).
DSR	total right stabilator position, includes linearization input, degrees
DSRIC	right stabilator deflection initial condition, degrees
DSRLN	component of total right stabilator deflection, linearization input only, degrees
DSRLN1	= DSRLN (if .not.LLNCSY) or (DSSYLN + DSASLN) (if LLNCSY).
DSSY	symmetric stabilator deflection, degrees
DSSYLN	component of total symmetric stabilator deflection, linearization input only, degrees
DSSYTR	symmetric stabilator deflection resulting from trim calculations, degrees
DSTMBT	pilot stick pitch trim button
DTV0LL	lower left thrust vectoring vane deflection, actuator output, degrees
DTV0LU	upper left thrust vectoring vane deflection, actuator output, degrees
DTV0RL	lower right thrust vectoring vane deflection, actuator output, degrees
DTV0RU	upper right thrust vectoring vane deflection, actuator output, degrees
DTVDLL	d/dt of lower left thrust vectoring vane deflection, degrees/sec
DTVDLU	d/dt of upper left thrust vectoring vane deflection, degrees/sec
DTVDRL	d/dt of lower right thrust vectoring vane deflection, degrees/sec
DTVDRU	d/dt of upper right thrust vectoring vane deflection, degrees/sec
DTVL	total left engine thrust vectoring vane deflection, degrees
DTVLL	left lower thrust vectoring vane deflection, degrees
DTVLLN	added to total left engine thrust vectoring vane deflection, linearization input only, degrees
DTVLU	left upper thrust vectoring vane deflection, degrees
DTVR	total left engine thrust vectoring vane deflection, degrees
DTVRL	right lower thrust vectoring vane deflection, degrees
DTVRLN	added to total right engine thrust vectoring vane deflection, linearization input only, degrees
DTVRU	right upper thrust vectoring vane deflection, degrees
DX1	intermediate variable in pressure sensor function calculations
DX2	intermediate variable in pressure sensor function calculations
E0	quaternion, n.d.
E00	E0 prior to normalization
E000	Direct integral of E0D = d/dt E0
E02	E0**2, used in direction cosine calculations
E0D	d/dt of E0, = .5*(-E1*P-E2*Q-E3*R), rad/sec
E0IC	Initial condition for E0, derived from trim Euler angles
E1	quaternion, n.d.
E10	E1 prior to normalization
E100	Direct integral of E1D = d/dt E1
E12	E1**2, used in direction cosine calculations
E1D	d/dt of E1, = .5*(E0*P-E3*Q+E2*R), rad/sec
E1IC	Initial condition for E1, derived from trim Euler angles
E2	quaternion, n.d.
E20	E2 prior to normalization
E200	Direct integral of E2D = d/dt E2
E22	E2**2, used in direction cosine calculations
E2D	d/dt of E2, = .5*(E3*P+E0*Q-E1*R), rad/sec
E2IC	Initial condition for E2, derived from trim Euler angles
E3	quaternion, n.d.
E30	E3 prior to normalization
E300	Direct integral of E3D = d/dt E3
E32	E3**2, used in direction cosine calculations
E3D	d/dt of E3, = .5*(-E2*P+E1*Q+E0*R), rad/sec
E3IC	initial condition for E3, derived from trim Euler angles

EPS	used to prevent divide by zero, 10^{-10}
EVENT	time at which dynamic check perturbation is applied, sec
F7BYPS	FCS logical for lateral gain schedule bypass, **default = .F.**, (DO NOT CHANGE)
FCRSET	FCS logical to reset lateral trim integrator, **default = .F.**, (DO NOT CHANGE)
FCS	marker for flight control system discrete block
FCSV83	FCS logical for pitch forward loop gain (8.3.3 PROM set), **default = .T.**, (DO NOT CHANGE)
FGKAOA	= Mach number, but limited so that: $0.9 \leq$ FGKAOA ≤ 1.4. Used as independent variable for sensor angle-of-attack gain value look-up.
FGRESL	gross force produced by left engine prior to inclusion of thrust vectoring effects, lbs.
FGRESR	gross force produced by right engine prior to inclusion of thrust vectoring effects, lbs.
FHIL	left engine thrust vectoring vane deflection angle, degrees
FHIR	right engine thrust vectoring vane deflection angle, degrees
FHM1	= Mach number, but limited so that: $0.0 \leq$ FHM1 ≤ 1.65. Used as independent variable in static pressure sensor look-up.
FRML	left engine ram drag, lbs
FRMR	right engine ram drag, lbs
FSCG	fuselage station coordinate of the center of gravity, inches
FSENG	fuselage station coordinate of the engine/airframe interface, either engine, 687.5 in.
FSRF	fuselage station coordinate of the aerodynamic reference center, inches
FTD	event marker for flight test data control input discrete block
FTDAT	array used to hold 1 record of flight data read off a file, 50 elements.
FVR	total generalized force vector for rigid-body dynamics
FVRA	aerodynamic component of FVR
FVRE	thrust induced component of FVR
FVRGR	gravity induced component of FVR
FVRI	inertial component of FVR
FVRWG	aerodynamic disturbance (wind and gust) component of FVR
G	acceleration due to gravity, 31.174 ft/sec^2
GAM	longitudinal flight path angle, radians
GAMDG	longitudinal flight path angle, degrees
GAMTR	target longitudinal flight path angle used in trim search, radians
GAMZR	GAM - GAMTR, driven to zero by trim search
GDTHR	scaling parameter placed on a random number applied to next trim search solution candidate to prevent one component of the solution vector from reaching a solution earlier and remaining fixed. An unchanging element of solution vector will lead to a singularity in the sensitivity matrix estimation.
GINV	1/g, used to scale various variables
GK1L	gain on left engine thrust error, VAC.
GK1R	gain on right engine thrust error, VAC.
GK3L	gain on left engine core dynamics. Second order filter.
GK3R	gain on right engine core dynamics. Second order filter.
GKAOA	angle-of-attack sensor gain value.
GLIMTR	FCS logical used in g-limit on stick input, **default=.T.** (DO NOT CHANGE)
GLOAD	desired load factor used in trim search when TCASE=3, g's
GLORID	FCS logical used for g load override (TK5, TK8), **default = .F.** (DO NOT CHANGE)
H	vertical distance from sea level to aircraft c.g., ft
HD	d/dt of H, ft/sec
HGC	vertical distance from ground to the a.r.c., ft.

HIC	initial condition for H, ft
HRF	1962 standard atmosphere altitude in feet corresponding to a given an air density, ρ, in slugs/ft^3
HSTORE	FCS logical for heavy stores, **default = .F.** (DO NOT CHANGE)
I	summing index used throughout simulation
IALG	integer switch to select ACSL integration algorithm:
(1)	Adams-Moulton, variable step, variable order
(2)	Gear's stiff; variable step, variable order
(3)	Runge-Kutta 1st order or Euler
(4)	**Runge-Kutta 2nd order (default for this simulation)**
(5)	Runge-Kutta 4th order (default for ACSL)
(6)	User supplied subroutine (INTEG)
(7)	Runge-Kutta-Fehlberg 2nd order
(9)	Runge-Kutta-Fehlberg, 5th order
IDYNCK	Integer switch used to invoke dynamic check capability
(0)	No disturbance input
(1)	One sided pulse
(2)	Doublet
(3)	User defined signal, read from array SIG
II	index used for sensor data tables
IN2FT	conversion factor, 1/12 (ft/in)
ISENS	FCS flag for sensor model
(0)	sensor effects off, ideal measurements
(1)	**MDC sensor model (default)**
ITFLG	integer switch used to control program flow when trimming
ITR	loop count used in inner loop of trim search. When a new candidate solution vector is calculated, the resulting derivatives are then calculated ITRMX times. This allows implicit loops in the trim loop to settle out. Implicit loops are caused by the fact that accelerations are inputs the the control system that determine control surface position. An alternate way to break the loop would be to include actuator states in the trim vector. The trim search works best when the trim vector is kept small.
ITRMX	number of passes made through equations of motion in inner trim loop. See ITR. **Default = 6.**
IXSEL	integer pointer array of size NXTRMX used to select elements of trim solution vector, XTRIM. The first NXTR elements of IXSEL select from 29 possible independent variables. XTRIM is formed in subroutine XFORM.
IXX	aircraft moment of inertia about x-body axis, slug-ft^2
IXZ	aircraft xz cross product of inertia, slug-ft^2
IYSEL	integer pointer array of size NXTRMX used to select elements to drive to zero in the trim output vector, YTRIM. The first NYTR elements of IYSEL select from 15 possible output variables. YTRIM is formed in subroutine YFORM.
IYY	aircraft moment of inertia about y-body axis, slug-ft^2
IZZ	aircraft moment of inertia about z-body axis, slug-ft^2
J	simulation summing index
KFS	longitudinal stick force spring, = 7.0 lbs/in, input to F18FCS
LAMDA	difference between ground track angle and heading angle, radians
LCAASP	logical switch
	(.T.) A dynamic check perturbation is added to the antisymmetric aileron command.
	(.F.) no action (default)
LCASYP	logical switch

(.T.) A dynamic check perturbation is added to the symmetric aileron command.
(.F.) no action (default)

LCFASP logical switch
(.T.) A dynamic check perturbation is added to the antisymmetric trailing-edge flap command.
(.F.) no action (default)

LCFSYP logical switch
(.T.) A dynamic check perturbation is added to the symmetric trailing-edge flap command.
(.F.) no action (default)

LCNASP logical switch
(.T.) A dynamic check perturbation is added to the antisymmetric leading-edge flap command.
(.F.) no action (default)

LCNSYP logical switch
(.T.) A dynamic check perturbation is added to the symmetric leading-edge flap command.
(.F.) no action (default)

LCPASP logical switch
(.T.) A dynamic check perturbation is added to the antisymmetric power lever angle command.
(.F.) no action (default)

LCPSYP logical switch
(.T.) A dynamic check perturbation is added to the symmetric power lever angle command.
(.F.) no action (default)

LCRASP logical switch
(.T.) A dynamic check perturbation is added to the antisymmetric rudder command.
(.F.) no action (default)

LCRSYP logical switch
(.T.) A dynamic check perturbation is added to the symmetric rudder command.
(.F.) no action (default)

LCSASP logical switch
(.T.) A dynamic check perturbation is added to the antisymmetric stabilator command.
(.F.) no action (default)

LCSBP logical switch
(.T.) A dynamic check perturbation is added to the speed brake command.
(.F.) no action (default)

LCSSYP logical switch
(.T.) A dynamic check perturbation is added to the symmetric stabilator command.
(.F.) no action (default)

LDBA logical switch - used for debugging aerodynamic calculations.
(.T.) Extended debugging output is written to unit PRN for each call to the aerodynamic coefficient calculations.
(.F.) no action (default)

LDBMI logical switch - used for inertial effects calculation debugging.
(.T.) Extended debugging output is written to unit PRN for each calculation of the inertial components of the generalized force vector and mass matrix.
(.F.) no action (default)

LDBOUT logical switch - primary simulation debugging logical

(.T.) Extended ACSL debugging output is written to unit PRN after the trim search or when time reaches TSTP.
(.F.) no action (default)

LDBSS logical switch - debugging logical for the subroutine that calculates the accelerations by solving the vector equation [M]{a} = {f}
(.T.) Debugging output is written to unit PRN from subroutine XRBVDC.
(.F.) no action (default)

LDBTR logical switch - used for debugging failed trim attempts.
(.T.) Each new solution trial, XTRIM, and the resulting YTRIM vector are written to unit=6 (typically the screen) and also unit=PRN
(.F.) no action (default)

LEULER logical switch:
(.T.) Euler angles rates are integrated directly.
(.F.) quaternion derivatives are integrated. (default)

LFCS logical switch - enables flight control system - read in from file "f18tr.dat"
(.T.) FCS on (default)
(.F.) FCS off

LFTD logical switch - enables use of flight data to drive control surface deflections
(.T.) Control deflections set to data in FTDAT. Actuators bypassed.
(.F.) Normal simulation operation. (default)

LLDTR logical switch
(.T.) When START command is executed, initial conditions for all possible XTRIM variables are read in from file "f18tr.dat". **The default on the first execution of START is .T. Reset to .F. after "f18tr.dat" is read in.**
(.F.) no action

LLIMDF logical switch
(.T.) Assign hardwired values to array of upper and lower bounds on elements of XTRIM variables. Trim algorithm confines search to this space. The hardwired values are defined in array DEFLIM in subroutine TRIM.
(.F.) Use limit values read from "f18tr.dat" file. (default)

LLNCSY logical switch, used to control the 'LN' exogenous inputs that can be created to support the 'ANALYZ' command in ACSL. Dictates whether the exogenous inputs available to the ANALYZ option, which are used to calculate _LN1 inputs, are symmetric/antisymmetric quantities or right/left quantities.
(.T.) _LN1 = _SYLN $\pm$ _ASLN (default)
(.F.) _LN1 = _LN
where _LN1 is summed with actual control surface position in degrees. Examples of _LN1 variables and DSRLN1 (left stabilator), DSLLN1 (right stabilator), and DRLLN1 (left rudder).

LPCAP logical switch
(.T.) Put dynamic check perturbation on lateral stick input.
(.F.) no action (default)

LPCRP logical switch
(.T.) Put dynamic check perturbation on rudder pedal input.
(.F.) no action (default)

LPCSP logical switch
(T.) A dynamic check perturbation is added to the longitudinal stick.
(.F.) no action (default)

LQSE logical switch - passed to subroutine SFAERRF and USAERRF - used to enable/disable the quasi-static-elastic (QSE) modeling option in the aerodynamic model

(T) normal QSE aerodynamics. All flex/rigid ratios and increments apply. (default)
(.F.) All flex/rigid ratios and increments are set to limiting value as altitude becomes large or density goes to zero.

LRDFTI logical switch - used to drive control surface positions according to a data file - as would occur with flight test data
(.T) Subroutine FTDRDF is called. Data is read from file="mhftdi.dat".
(F) no action (default)

LRDWR internal integer switch - intermediate flow control switch set in the simulation and used to control action in call to subroutine RDWRTR - can not be set by runtime commands
(1) causes f18tr.dat trim file to be read in (default)
(2) causes f18tr.dat trim file to be written on

LRTE logical switch - passed to subroutine SFAERRF - used when conducting comparisons to recover the form of the aerodynamic model found in the original DMS simulation
(.T.) activates form of subroutine SFAERRF that agrees with *dmsf18* math model (default)
(.F.) activates changes discussed in section 5.1 - aerodynamic model

LTHCGL left engine thrust moment about body frame x-axis at c.g., ft-lbs

LTHCGR right engine thrust moment about body frame x-axis at c.g., ft-lbs

LTHDMS logical switch - used to duplicate the DMS throttle position
(.T.) use DMS scaled throttle position
(.F.) normal MDC F18 throttle position (default)

LTHVEC logical switch - used to invoke preliminary thrust-vectored engine model
(.T.) preliminary thrust-vectored engine model (default)
(.F.) non-thrust-vectored engine model

LTR logical switch
(.T.) Trim search is initiated after runtime START command. If trim fails, TSTP set = 0. (default)
(.F.) No trim attempted. Integration proceeds with current i.c.'s until T=TSTP.

LTRFLG internal logical switch - used to indicate current TRIM status
(.T.) simulation in trim search loop
(.F.) not in trim search loop

LVDEF logical flag - used to obtain extended output with variable definitions form dictionary
(.T.) All simulation variable values printed out onto unit=PRN with dictionary definitions taken from unit=29.
(.F.) no action (default)

LWRFTO logical switch - used for FLT data comparison
(.T.) subroutine FTDWRF called - data written to file="ftifo.dat"
(.F.) no action (default)

LWRTR logical switch - used to write new file "f18tr.dat"
(.T.) Write the next calculated trim values to file "f18tr.dat". Set LWRTR to .F.
(.F.) no action (default)

MACH Mach number, n.d.

MACHRF Free-stream Mach number at a.r.c., n.d.

MACHTR Mach number to be achieved in trim search if "1" is in the active IXSEL list, n.d.

MARR aerodynamic component of rigid-rigid quadrant of mass matrix

MASS aircraft mass, slugs

MINT minimum integration step size (default value supplied by ACSL)

MIRR inertial component of rigid-rigid quadrant of a mass matrix

MMRR total rigid-rigid quadrant of a mass matrix

MNSPIN FCS logical spin switch. **Default = .F.** (DO NOT CHANGE).

MTHCGL	left engine thrust moment about body frame y-axis at c.g., ft-lbs.
MTHCGR	right engine thrust moment about body frame y-axis at c.g., ft-lbs.
MUDGTR	bank angle in degrees sought in trim search if "29" included in active IXSEL list. Will produce coordinated turn if TCASE = 2.
MUTRMX	In the *f18bas* trim strategy (section 2.2), MUTRMX is equivalent to variable "MUMAX" in ACSL TRIM algorithm (see section 5.2.1, ACSL REFERENCE MANUAL REV. 4.1) - **default = 10**
MXITTR	maximum number of iterations for simulation trimming scheme - **default = 80**
MXSTP	maximum integration step size - default = 0.00625 seconds
NE	square root of sum of squares of quaternion intermediate states, used to normalize quaternions at each time step to prevent numerical problems - n.d
NSTP	minimum number of integration steps in a communication interval. Default = 1. This allows integration step size to be controlled by MXSTP as per recommendation in ACSL manual.
NTHCGL	left engine thrust yaw moment about body z-axis at c.g. (TV), ft-lbs
NTHCGR	right engine thrust yaw moment about body z-axis at c.g. (TV), ft-lbs
NTVOPT	integer switch - FCS flag for thrust vector pitch control blend (1=ON). Default = 1.
NXTR	number of independent variables for simulation trim search ($\leq$ NXTRMX)
NXTRMX	maximum number of independent variables for simulation trim search (**10**)
NYLOC	3 element array describing location of n_y accelerometer in FS/BL/WL coordinates, inches
NYSSG	lateral acceleration sensor signal, g's
NYTR	number of dependent variables for simulation trim algorithm (=NXTR)
NZSSG	normal acceleration sensor signal, g's
NZZR	normal acceleration 1-g steady state removed. NZZR = - AZ - 1. g's
ORIDE	FCS logical to override angle-of-attack limiter. Default = .F. (DO NOT CHANGE).
P	body axis roll rate, rad/sec
P1L	gain in left core engine dynamics. 2nd order filter.
P1R	gain in right core engine dynamics. 2nd order filter
P2L	gain in left core engine dynamics. 2nd order filter
P2R	gain in right core engine dynamics. 2nd order filter
PAGSIZ	Integer that defines page size as number of lines. (45 for NOS, 55 for VAX)
PCA	pilot lateral stick position, inches
PCAFTD	pilot lateral stick position from flight test data, inches
PCASG	pilot lateral stick position signal array for IDYNCK = 3
PCATR	pilot lateral stick position required for trim, inches.
PCR	pilot rudder pedal force, lbs
PCRFTD	pilot rudder pedal force from flight test data, lbs
PCRSG	pilot rudder pedal force signal array for IDYNCK=3
PCRTR	pilot rudder pedal force required for trim, lbs
PCS	pilot longitudinal stick position, inches
PCSFTD	pilot longitudinal stick position flight test data, inches
PCSSG	pilot longitudinal stick position signal array for IDYNCK=3
PCSTR	pilot longitudinal stick position required for trim, inches
PD	body axis roll acceleration, rad/sec^2
PDG	body axis roll rate , degrees/second
PGAIN	gain on dynamic check perturbation command
PHI	Euler roll angle, radians
PHI0	Euler roll angle intermediate state, integral of PHID
PHID	Euler roll angle rate of change, rad/sec
PHIDG	Euler roll angle, degrees

PHILN	Euler roll angle linearization dummy variable, radians
PHITR	Euler roll angle required for trim, radians
PHIZR	Euler roll angle referenced to "PHITR", radians
PIC	body axis roll initial condition, rad/sec
PLAL	left engine total throttle position (power level angle), degrees
PLAR	right engine total throttle position (power level angle), degrees
PLDL	left throttle boost output error term.
PLDR	right throttle boost output error term
PRN	high volume output goes to unit=PRN. Passed to Fortran subroutines that write to an output file. Default = 9.
PS	sensed roll rate , °/sec, = PS0 (if ISENS=0) or = PS1 (if ISENS=1)
PS0	actual roll rate, °/sec
PS1	output of rate sensor model, °/sec
PSAS	FCS logical for pitch SAS on. Default = .T. (DO NOT CHANGE).
PSI	Euler yaw angle, radians
PSI0	Euler yaw angle intermediate state, integral of PSID
PSID	Euler yaw angle rate of change, rad/sec
PSIDG	Euler yaw angle, degrees
PSILN	Euler yaw angle rate linearization dummy variable, radians. [Constant].
PSITR	Euler yaw angle required for trim, radians
PSTATC	static atmospheric pressure, = f(H), PSF
PSTS	output of static atmospheric pressure sensor, PSF.
PSTS0	value assigned to PSTS if ISENS=0 (Ideal sensors).
PSTS1	value assigned to PSTS if ISENS=1 (Sensor model #1)
PT	intermediate variable in compressible pressure computation
PTI	intermediate variable in compressible pressure computation
PTURB	perturbation value used for pulse or double signal if IDYNCK=1 OR 2. Default=0.
PW	wind axis roll rate, rad/sec
PWDG	wind axis roll rate, °/sec
Q	body axis pitch rate, rad/sec
QBAR	aircraft dynamic pressure, $\bar{q}$, lbs/ft^2
QBARS	intermediate variable, = $\bar{q}S$, lbs
QBARSB	intermediate variable, = $\bar{q}Sb$, lbs
QBARSC	intermediate variable, = $\bar{q}S\bar{c}$, lbs
QC	compressible impact pressure, PSF
QCDA	function used in compressible impact pressure computation
QCDB	function used in compressible impact pressure computation
QCDC	function used in compressible impact pressure computation
QCDD	function used in compressible impact pressure computation
QCDE	function used in compressible impact pressure computation
QCDF	function used in compressible impact pressure computation
QCDG	function used in compressible impact pressure computation
QCDH	function used in compressible impact pressure computation
QCIS	output of compressible pressure sensor model, PSF
QCIS0	compressible pressure--ideal sensor, PSF
QCIS1	compressible pressure--sensor model #1, PSF
QCX	intermediate value in sensed compressible pressure computation
QD	body axis pitch acceleration, rad/sec^2
QDG	body axis pitch rate, °/sec
QIC	body axis pitch rate initial condition, rad/sec
QS	sensed pitch rate, °/sec, = QS0 or =QS1 based on "ISENS"
QS0	actual pitch rate, °/sec

QS1	output of rate gyro model, °/sec
QW	wind axis pitch rate, rad/sec
QWDG	wind axis pitch rate, °/sec
R	body axis yaw rate, °/sec
RA2DG	radians to degrees conversion factor, °/rad
RD	body axis yaw acceleration, rad/sec^2
RDG	body axis yaw rate, °/sec
RHO	air density, slugs/ft^3
RHOSL	air density at sea level, 0.00237688 slugs/ft^3
RIC	body axis yaw rate initial condition, rad/sec
RRHO	ratio of actual air density to RHOSL, n.d.
RS	sensed yaw rate,.°/sec, = RS0 OR =RS1 based on "ISENS"
RS0	actual yaw rate, °/sec
RS1	output of rate gyro model, °/sec
RSAS	FCS logical for SAS on. Default = .T. (DO NOT CHANGE).
RTABL	limited left afterburner thrust rate of change, lbs/sec
RTABR	limited right afterburner thrust rate of change, lbs/sec
RTELLL	left engine core spool-up/-down rate lower limit, = - 5500 lbs/sec
RTELLR	right engine core spool-up/-down rate lower limit, = - 5500 lbs/sec
RTEULL	left engine core spool-up/-down rate upper limit, = 5500 lbs/sec
RTEULR	right engine core spool-up/-down rate upper limit, = 5500 lbs/sec
RW	wind axis yaw rate, rad/sec
RWDG	wind axis yaw rate, °/sec
SBERR	error term between speed brake position and command, degrees
SBNLRL	speed brake no load rate limit, = 60/2.5, degrees/second
SETVDC	set thrust vectoring differential command, input to subroutine F18FCS, default = -101. [Constant]
SETVSC	set thrust vectoring symmetric command, input to subroutine F18FCS, default = -101. [Constant]
SIG	array of input values, in order of increasing time. Use SIG for open loop control surface deflections independent of of simulated pilot input SIG(1 ->100) = Signal amplitude values SIG(101->200) = Time breakpoint of corresponding amplitude values
SIGCPS	value of throttle signal at current time, = CSPYSG(T)
SIGNAL	value of generic signal at current time, = SIG(T)
SIGPCA	value of lateral stick signal at current time, = PCASG (T)
SIGPCR	value of rudder pedal signal at current time, = PCRSG (T)
SIGPCS	value of longitudinal stick signal at current time, = PCSSG (T)
SNALF	sin(α = angle of attack at c.g.)
SNALRF	sin(α_{rcf} = angle of attack at a.r.c.)
SNAZEL	sine of azimuth angle of installed left engine n.d.
SNAZER	sine of azimuth angle of installed right engine n.d.
SNBET	sin(β = sideslip angle at c.g.)
SNPHI	sin(ϕ)
SNPHO2	sin(PHITR / 2), n.d.
SNPSO2	sin(PSITR / 2), n.d.
SNTHE	sin(THETA), n.d.
SNTHO2	sin(THETR / 2), n.d.
SNTVOA	sine of thrust-vectoring vane orientation angle, n.d.
SPSFXD	sensed static pressure gain value
STORES	FCS logical for stores logic (default = F.) (DO NOT CHANGE).

SWRF	aerodynamic (wing) reference area, 400 ft^2
T	time, sec
T1	temporary variable used in rotational acceleration equations, = IXX*P – IXZ*R
T2	temporary variable used in rotational acceleration equations, = IYY*Q
T3	temporary variable used in rotational acceleration equations, = – IXZ*P + IZZ*R
TABL	left afterburner thrust, lbs
TABLIC	left afterburner thrust initial condition, lbs
TABR	right afterburner thrust, lbs
TABRIC	right afterburner thrust initial condition, lbs
TANTHE	tan(θ)
TAUGMN	minimum engine thrust with afterburner engaged, lbs
TAUGMX	maximum engine thrust with afterburner engaged, lbs
TAUL	left engine time constant for VAH first order lag dynamics
TAULS	saved value of TAUL from last time in core engine dynamics
TAUR	right engine time constant for VAH first order lag dynamics
TAURS	saved value of TAUR from last time in core engine dynamic
TCASE	integer switch to determine trim solution sought (1) no action - used for normal wings-level trim (2) p,q,r set to values required for coordinated turn at selected bank angle in degrees, MUDGTR (3) q set to value required for symmetric pull-up at load factor in g's selected by GLOAD
TEMPQ	coefficient = QBAR / 100
TESTVL	test value for left engine core dynamic nonlinear switch logic
TESTVR	test value for right engine core dynamic nonlinear switch logic
TH	total engine thrust for thrust vectored engine, lbs
THE	Euler pitch angle, radians
THE0	Euler pitch angle intermediate state, integral of THED
THED	Euler pitch angle rate of change, rad/sec
THEDG	Euler pitch angle, degrees
THELN	Euler pitch angle linearization dummy variable, radians
THETR	Euler pitch angle rate required for trim, radians
THL	left engine net thrust, lbs
THL0	left engine net thrust intermediate state, lbs = TTL if in military power range = TABL if in afterburner
THLLN	left engine thrust linearization dummy variable, lbs
THR	right engine net thrust, lbs
THR0	right engine net thrust intermediate state, lbs = TTR if in military power range = TABR if in afterburner
THRCL	thrust command sent to left engine, lbs
THRCR	thrust command sent to right engine, lbs
THRFL	left engine reference thrust. < 0 if in afterburner dynamics, lbs
THRFR	right engine reference thrust. < 0 if in afterburner dynamics, lbs
THRLN	right engine thrust linearization dummy variable, lbs
THXL	left engine net thrust component in x-direction (NTV), lbs
THXR	right engine net thrust component in x-direction (NTV), lbs
THYL	left engine net thrust component in y-direction (NTV), lbs
THYR	right engine net thrust component in y-direction (NTV), lbs
THZL	left engine net thrust component in z-direction (NTV), lbs
THZR	right engine net thrust component in z-direction (NTV), lbs
TIDLE	engine thrust at idle throttle setting, lbs

TMIL	maximum military thrust, lbs
TNETL	miscellaneous output - left engine net thrust (TV), lbs
TNETR	miscellaneous output - right engine net thrust (TV), lbs
TRTOL	convergence requirement for successful trim - default = 0.00005
TSFCS	time interval at which flight control system discrete is invoked, seconds
TSFTD	time interval between flight test data points, sec
TSTP	simulation stop time - default = 0.0 seconds
TT	total thrust from left and right engine thrust, lbs
TTL	sum of left engine thrust terms VAKL & VAFL (NTV), lbs
TTR	sum of right engine thrust terms VAKL & VAFL (NTV), lbs
TVDBOT	FCS logical . Used to removed thrust vectoring deadband. Makes trimming possible when thrust vectoring is selected. Default = .T.
TVLLIC	CTVLL initial condition
TVLUIC	CTVLU initial condition
TVOA	thrust -vectoring vane orientation angle, degrees
TVRLIC	CTVRL initial condition
TVRUIC	CTVRU initial condition
U	body axis inertial velocity in x-body direction, ft/sec
UA	body axis atmospheric velocity in x-body direction, ft/sec
UA2	UA squared, ft^2/sec^2
UARF	UA at a.r.c., ft/sec
UD	body axis inertial acceleration in x-direction, ft/sec^2
UDA	d/dt of UA, ft/sec^2
UDWG	d/dt of aerodynamic disturbance component in x-body axis, ft/sec^2
UIC	body axis inertial velocity in x-direction i.c., ft/sec
UWG	x-body component of aerodynamic disturbance velocity, ft/sec
V	body axis inertial velocity in y-body direction, ft/sec
VA	body axis atmospheric velocity in y-body direction, ft/sec
VA2	VA squared, ft^2/sec^2
VACL	left engine error term. = THRCL - VAFL
VACR	right engine error term. = THRCL - VAFL
VAE0L	rate-limited left engine spool-up-/-down intermediate state, lbs/sec
VAE0R	rate-limited right engine spool-up-/-down intermediate state, lbs/sec
VAEDL	artificial fast derivative inserted to achieve explicit formulation, the integral of which is VAE0L
VAEDR	artificial fast derivative inserted to achieve explicit formulation, the integral of which is VAE0R
VAEL	rate-limited left engine spool-up-/-down , lbs/sec
VAELIC	rate-limited left engine spool-up-/-down initial condition, lbs/sec
VAER	rate-limited right engine spool-up-/-down , lbs/sec
VAERIC	rate-limited right engine spool-up-/-down initial condition, lbs/sec
VAFL	main component of left engine core dynamics, lbs
VAFLIC	main component of left engine core dynamics initial condition, lbs
VAFR	main component of right engine core dynamics, lbs
VAFRIC	main component of right engine core dynamics initial condition, lbs
VAHL	output of left engine core dynamics nonlinear switch, lbs
VAHLS	saved value of VAHL from last time in core engine dynamic
VAHR	output of right engine core dynamics nonlinear switch, lbs
VAHRS	saved value of VAHR from last time in core engine dynamic
VAK1DL	intermediate left core engine dynamic, saved past value, lbs/sec
VAK1DR	intermediate right core engine dynamic, saved past value, lbs/sec
VAKDDL	left core engine spool up input, lbs/sec^2

VAKDDR	right core engine spool up input, lbs/sec^2
VAKDL	intermediate left core engine dynamic, lbs/sec
VAKDR	intermediate right core engine dynamic, lbs/sec
VAKL	output of left engine core 2nd order dynamics, lbs
VAKLIC	initial condition of integrator for VAKL
VAKR	output of right engine core 2nd order dynamics, lbs
VAKRIC	initial condition of integrator for VAKR
VARF	VA at a.r.c., ft/sec
VD	body axis inertial acceleration in y-direction , ft/sec^2
VDA	d/dt of VA, ft/sec^2
VDBA	deadband applied to thrust vectoring vane deflection, radians
VDWG	d/dt of aerodynamic disturbance component in y-body axis, ft/sec^2
VGS	aircraft ground speed, ft/sec
VH1DL	VH1L rate of change, lbs/sec
VH1DR	VH1R rate of change, lbs/sec
VH1L	left engine state created by 1st order lag mechanization, lbs
VH1LIC	left engine core thrust at initial condition, lbs
VH1LS	saved value of 'VH1L' from last time in core engine dynamic, lbs
VH1R	right engine state created by 1st order lag mechanization, lbs
VH1RIC	right engine core thrust at initial condition, lbs
VH1RS	saved value of 'VH1R' from last time in core engine dynamic., lbs
VIC	body axis inertial velocity in y-direction i.c., ft/sec
VKDLIC	initial condition on first integral in left core engine dynamic, lbs/sec^2
VKDRIC	initial condition on first integral in right core engine dynamic, lbs/sec^2
VS	speed of sound, ft/sec
VT	aircraft total atmosphere velocity, ft/sec
VTD	aircraft total atmosphere acceleration, ft/sec^2
VTINV	coefficient to normalize by total velocity, sec/ft
VTRF	total airspeed at a.r.c., ft/sec
VTRINV	coefficient to normalize by total velocity at a.r.c., sec/ft
VTTR	aircraft total atmosphere velocity after successful trim, ft/sec
VWG	y-body component of aerodynamic disturbance velocity, ft/sec
W	body axis inertial velocity in z-body direction, ft/sec
WA	body axis atmospheric velocity in z-body direction, ft/sec
WA2	WA squared, ft^2/sec^2
WARF	WA at a.r.c., ft/sec
WD	body axis inertial acceleration in z-direction, ft/sec^2
WDA	(d/dt) of WA, ft/sec^2
WDWG	d/dt of aerodynamic disturbance component in z-body axis, ft/sec^2
WFL	left engine fuel flow
WFR	right engine fuel flow
WIC	body axis inertial velocity in z-direction i.c., ft/sec
WIDTH	time span to hold step input dynamic check perturbation command,.default = 1.0 seconds
WLCG	water line coordinate of the center of gravity, inches
WLENG	water line coordinate of the engine/airframe interface, right/left engine, 100.0 in.
WLRF	water line coordinate of the aerodynamic reference center, = 100.0 inches
WT	aircraft weight, lbs
WWG	z-body component of aerodynamic disturbance velocity, ft/sec
XALF	air data probe alpha with source noise added, degrees
XARF	x-direction distance from c.g. to a.r.c., ft

XBAR	normalized x-location of accelerometer from c.g., n.d.
XD	aircraft inertial ground track velocity in track direction , ft/sec
XDDWG	d/dt of XDWG, ft/sec^2
XDWG	aerodynamic disturbance velocity component in x-earth frame direction, ft/sec
XDWGIC	aerodynamic disturbance velocity component in x-earth frame direction - initial condition, ft/sec
XHM1	sensed static pressure breakpoint vector
XRFCG	'x' component of the location of right (left) engine/airframe interface point in c.g.-centered body frame, = -(FSENG - FSCG)/12., ft.
XTHCG	total thrust force at c.g. in body x-direction (TV), lbs
XTHCGL	left engine thrust force at c.g. in body x-direction (TV), lbs
XTHCGR	right engine thrust force at c.g. in body x-direction (TV), lbs
XTRLM	2 by 30 array with upper and lower limits on XTRIM independent variables
XTVL	left engine thrust force in body x-direction (TV), lbs
XTVR	right engine thrust force in body x-direction (TV), lbs
YARF	y-direction distance from c.g. to a.r.c., ft
YBAR	normalized y-location of accelerometer from c.g., n.d.
YD	aircraft inertial ground track velocity in crosstrack direction, ft/sec
YDDWG	d/dt of YDWG, ft/sec^2
YDWG	aerodynamic disturbance velocity component along earth frame y-axis, ft/sec
YDWGIC	aerodynamic disturbance velocity component along earth frame y-axis - initial condition, ft/sec
YGKAO2	sensed angle-of-attack gain table
YGKAOA	sensed angle-of-attack gain table
YRFCGL	'y' component of the location of left engine/airframe interface point in c.g.-centered body frame, = (BLENGL - BLCG)/12., ft.
YRFCGR	'y' component of the location of right engine/airframe interface point in c.g.-centered body frame, = (BLENGR - BLCG)/12., ft.
YSAS	FCS logical for yaw SAS on. Default = .T. (DO NOT CHANGE)
YTHCGL	left engine thrust force at c.g. in body y-direction (TV), lbs
YTHCGR	right engine thrust force at c.g. in body y-direction (TV), lbs
YTVL	left engine thrust force at nozzle in y-body direction (TV), lbs
YTVR	right engine thrust force at nozzle in y-body direction (TV), lbs
ZARF	z-direction distance from c.g. to a.r.c., ft
ZBAR	normalized z-location of accelerometer from c.g., n.d.
ZDDWG	(d/dt) of ZDWG, ft/sec^2
ZDWG	aerodynamic disturbance velocity component along earth frame z-axis, ft/sec
ZDWGIC	aerodynamic disturbance velocity component along earth frame z-axis - initial condition, ft/sec
ZRFCG	'z' component of the location of right (left) engine/airframe interface point in c.g.-centered body frame, = -(WLENG - WLCG)/12., ft.
ZTHCGL	left engine thrust force at c.g. in body z-direction (TV), lbs
ZTHCGR	right engine thrust force at c.g. in body z-direction (TV), lbs
ZTVL	left engine thrust force in body z-direction (TV), lbs
ZTVR	right engine thrust force in body z-direction (TV), lbs
ZZABL	d/dt of left engine thrust in AB before rate limiting, lbs/sec
ZZABR	d/dt of right engine thrust in AB before rate limiting, lbs/sec

APPENDIX B - EXAMPLE CASES

Trim to Level Flight (TCASE=1)

The following wings level (TCASE=1) trim cases are demonstrated by an example terminal session:

(1) M=.6, H=10,000 (ft),flight path angle (γ) = 0.

(2) M=.8, H=10,000 (ft),flight path angle (γ) = 0.

(3) α = 5°, H=10,000 (ft),flight path angle (γ) = 0.

(4) α = 10°, H=10,000 (ft),flight path angle (γ) = 0.

Commands entered at the keyboard are in **bold Courier**. The program and VMS responses are in plain Courier. Comments inserted after the session to explain actions are in *italics*.

```
AGCB9$  @setup_bas                          Assign data files to names read by program.
$       assign [CSB.F18BAS.DATA_FILES]T5AERO.F18BAS     t5aero
$       assign [CSB.F18BAS.DATA_FILES]T5ENG.F18BAS      t5eng
$       assign [CSB.F18BAS.DATA_FILES]F18BAS.DCT        for029
$       set noverify
AGCB9$  type  f18tr.dat                     Show copy of f18tr.dat.dat file used to store
                                            trim information.
F18 WITH 60% FUEL, TIP MISSLES, LEVEL TURN

------------------------------ MASS, INERTIAS, AND CG LOCATION
  33310.0     WT     (LBS)          WEIGHT
  23000.0     IXX    (SLUG*FT**2)   INERTIA ABOUT X AXIS
 151293.0     IYY    (SLUG*FT**2)   INERTIA ABOUT Y AXIS
 169945.0     IZZ    (SLUG*FT**2)   INERTIA ABOUT Z AXIS
  -2971.0     IXZ    (SLUG*FT**2)   XZ PLANE INERTIA PRODUCT
    455.0     FSCG   (IN)           CG LOC - FUSELAGE STATION
      0.0     BLCG   (IN)           CG LOC - BUTTOCK LINE
    102.8     WLCG   (IN)           CG LOC - WATER LINE

----------------------- FLIGHT CONDITIONS AND MODELING OPTIONS
        F    LTHVEC | THRUST VECTORING ON
        T    LFCS   | FLT CONTROL SYTEM ON
        F    LTHDMS | USE DMS PLA SCHEDULE
        T    LRTE   | USE R/T EQV AERO
  0.60000    MACHTR (N.D.)
 10000.00    HIC    (FT)
    30.00    MUDGTR (DEG)
  1.00000    GLOAD  (G)

------------------------------------ TRIM SETUP IF LTR = .TRUE
  1          TCASE | 1=LEVEL, 2=COORD TURN (MUDGTR), 3=PULL UP(GLOAD)
  4          NUMBER OF DRIVER VARIABLES
 ELEMENT     LIMITS                          VARIABLE NAME, UNITS
 14          -2.500       5.000              PCSTR   (INCHES )
 16          31.000     127.000              DPSYTR  (% POWER)
  3          -0.100       1.500              ALFTR   (RADIANS)
  7          -1.500       1.500              THETR   (RADIANS)

  4          NUMBER OF DRIVEN VARIABLES
 ELEMENT     VARIABLE NAME, UNITS
  1          UD       (FT/SEC2)
```

```
 3          WD       (FT/SEC2)
 5          QD      (RAD/SEC2)
 7          GAMZR    (RADIANS)

INITIAL CONDITIONS :
0.60000002           MACHTR   (N.D.)
0.00000000E+00       BETTR    (RADIANS)
0.43141015E-01       ALFTR    (RADIANS)
0.00000000E+00       PIC      (RAD/SEC)
0.00000000E+00       QIC      (RAD/SEC)
0.00000000E+00       RIC      (RAD/SEC)
0.43145921E-01       THETR    (RADIANS)
0.00000000E+00       PHITR    (RADIANS)
0.00000000E+00       PSITR    (RADIANS)
0.00000000E+00       GAMTR    (RADIANS)
0.00000000E+00       DTVL     (DEGREES)
0.00000000E+00       DTVR     (DEGREES)
0.00000000E+00       PCATR    (INCHES )
0.23838108E-01       PCSTR    (INCHES )
0.00000000E+00       PCRTR    (LBS    )
 60.480717           DPSYTR   (% POWER)
0.00000000E+00       DPASTR   (% POWER)
0.35326593E-01       DSSYTR   (DEGREES)
0.00000000E+00       DSASTR   (DEGREES)
0.00000000E+00       DASYTR   (DEGREES)
0.00000000E+00       DAASTR   (DEGREES)
0.00000000E+00       DRSYTR   (DEGREES)
0.00000000E+00       DRASTR   (DEGREES)
 3.2825463           DNSYTR   (DEGREES)
0.00000000E+00       DNASTR   (DEGREES)
 3.4605167           DFSYTR   (DEGREES)
0.00000000E+00       DFASTR   (DEGREES)
0.00000000E+00       CSB      (DEGREES)
 30.000000           MUDGTR   (DEGREES)
0.00000000E+00       future expansion


$END OF DATA READING SECTION

CASE SELECTIONS :
TCASE     1   STRAIGHT & LEVEL STEADY STATE
          2   COORDINATED TURN STEADY STATE
          3   PULL-UP          STEADY STATE

TRIM VALUE SELECTIONS:
        TRIM DRIVER ARRAY                    TRIM OUTPUT ARRAY
          1    MACHTR   (N.D.)                1 UD       (FT/SEC2)
          2    BETTR    (RADIANS)             2 VD       (FT/SEC2)
          3    ALFTR    (RADIANS)             3 WD       (FT/SEC2)
          4    PIC      (RAD/SEC)             4 PD      (RAD/SEC2)
          5    QIC      (RAD/SEC)             5 QD      (RAD/SEC2)
          6    RIC      (RAD/SEC)             6 RD      (RAD/SEC2)
          7    THETR    (RADIANS)             7 GAMZR    (RADIANS)
          8    PHITR    (RADIANS)             8 PHIZR    (RADIANS)
          9    PSITR    (RADIANS)             9 THE      (RADIANS)
         10    GAMTR    (RADIANS)            10 LAMDA    (RADIANS)
         11    DTVL     (DEGREES)            11 FYTOT        (G S)
         12    DTVR     (DEGREES)            12 CL-CLTR     (N.D.)
```

```
          13     PCATR    (INCHES )                    13 CM          (N.D.)
          14     PCSTR    (INCHES )
          15     PCRTR    (LBS    )
          16     DPSYTR   (% POWER)
          17     DPASTR   (% POWER)
          18     DSSYTR   (DEGREES)
          19     DSASTR   (DEGREES)
          20     DASYTR   (DEGREES)
          21     DAASTR   (DEGREES)
          22     DRSYTR   (DEGREES)
          23     DRASTR   (DEGREES)
          24     DNSYTR   (DEGREES)
          25     DNASTR   (DEGREES)
          26     DFSYTR   (DEGREES)
          27     DFASTR   (DEGREES)
          28     CSB      (DEGREES)
          29     MUDGTR   (DEGREES)
          30     future expansion
AGCB9$ acsl/run  f18bas                    Execute file 'f18bas.exe'.

 The ACSL Licence will terminate in  3 months
 Ask your systems manager to contact Mitchell and Gauthier for new key
 Telephone (USA) 508/369-5115

 Switching CMD unit to 4 to read F18BAS.CMD
ACSL>start                                 Start case (1). TSTP=0.0 so program stops
                                           immediately after trim attempt. LTR true
                                           causes trim attempt to be made.
 >>> READING VALUES TO INIT TRIM SEARCH (F18TR FILE)
 >>> READING AERO DATA
        TIME ELAPSED =     7.238281     SECONDS
 >>> READING ENGINE DATA
        TIME ELAPSED =     1.339844     SECONDS
NORM OF Y ON FIRST PASS =    3.0825590E-03
COLUMN             2 NOT REPEATABLE. ERROR =    6.519258022308350E-009
TRIM ACHIEVED. ERROR =    4.4315234E-06
ITERATIONS REQUIRED =             27
t exceeds tstp
ACSL>s  tcwprn=80                          Set display column width=80 (required only
                                           for this demo).
ACSL>d  machtr,hic,vt,ud,vd,wd,pd,qd,rd,thedg,alfdg,...
ACSL>pcstr,dpsytr,phi
      MACHTR 0.60000000          HIC 10000.0000          VT 646.442000
          UD-2.3931E-04           VD 0.                  WD 1.3487E-04
          PD 0.                   QD-1.3995E-05          RD 0.
       THEDG 2.47138000        ALFDG 2.47177000      PCSTR-0.06876620
      DPSYTR 60.4787000          PHI 0.
ACSL>d ixsel                               Display selected XTRIM variables.
       IXSEL     14                        16                      3
                  7                        0                       0
                  0                        0                       0
                  0
ACSL>s  machtr=.8                          Change desired Mach for new trim.
ACSL>start                                 Start case (2)
NORM OF Y ON FIRST PASS =    0.2624109
TRIM ACHIEVED. ERROR =    1.0789349E-05
ITERATIONS REQUIRED =             23
t exceeds tstp
```

```
ACSL>d  machtr,hic,vt,ud,vd,wd,pd,qd,rd,thedg,alfdg,...
ACSL>pcstr,dpsytr
      MACHTR 0.80000000          HIC 10000.0000          VT 861.923000
          UD-2.7995E-04           VD 0.                  WD 5.5112E-04
          PD 0.                   QD 3.8318E-05          RD 0.
       THEDG 1.32924000        ALFDG 1.32953000       PCSTR-1.00055000
      DPSYTR 80.5041000
ACSL>s  ixsel=14,16,1,7                    Change XTRIM select list by replacing
                                           ALFTR with MACHTR. This causes
                                           MACHTR to be varied while ALFTR is left
                                           fixed. Useful for trimming to a desired angle
                                           of attack.
ACSL>s  alftr=.087266462                   Set ALFTR in radians to produce 5 degrees
                                           α.
ACSL>start                                 Start case (3)
NORM OF Y ON FIRST PASS =    1.022277
COLUMN            2 NOT REPEATABLE. ERROR =   4.951133405484143E-007
COLUMN            4 NOT REPEATABLE. ERROR =   2.402740051365981E-007
TRIM ACHIEVED. ERROR =   2.0998859E-06
ITERATIONS REQUIRED =           16
t exceeds tstp
ACSL>d  machtr,hic,vt,ud,vd,wd,pd,qd,rd,thedg,alfdg,...
ACSL>pcstr,dpsytr,phi
      MACHTR 0.42064100          HIC 10000.0000          VT 453.200000
          UD-3.4658E-05           VD 0.                  WD 7.4657E-06
          PD 0.                   QD 7.1250E-06          RD 0.
       THEDG 5.00025000        ALFDG 5.00000000       PCSTR 0.21334900
      DPSYTR 55.0637000           PHI 0.
ACSL>s  alftr=.1745329                     Set ALFTR in radians to produce 10 degrees
                                           α.
ACSL>start                                 Start case (4)
NORM OF Y ON FIRST PASS =   0.2984953
TRIM ACHIEVED. ERROR =   2.2749262E-05
ITERATIONS REQUIRED =           11
t exceeds tstp
ACSL>d  machtr,hic,vt,ud,vd,wd,pd,qd,rd,thedg,alfdg,...
ACSL>pcstr,dpsytr,phi
      MACHTR 0.29906200          HIC 10000.0000          VT 322.210000
          UD-5.2243E-04           VD 0.                  WD 8.9943E-04
          PD 0.                   QD 8.4760E-05          RD 0.
       THEDG 10.0004000        ALFDG 10.0000000       PCSTR 0.50991900
      DPSYTR 60.9197000           PHI 0.
ACSL>stop                                  End session
End ACSL run
AGCB9$
```

Trim to Bank Angle (TCASE=2)

The following trim to bank angle (TCASE=1) trim cases are demonstrated by an example terminal session. The session is begun by reading in the TCASE=1 trim data file f18tr.dat, then using run time commands to set up the bank angle trim.

(1) M=.6, H=10,000 (ft), wings level, TCASE=1

(2) M=.6, H=10,000 (ft), TCASE=2, Bank angle (μ) = 30°

(3) M=.6, H=10,000 (ft), TCASE=2, Bank angle (μ) = 60°

```
AGCB9$ acsl/run  f18bas

 The ACSL Licence will terminate in  3 months
 Ask your systems manager to contact Mitchell and Gauthier for new key
 Telephone (USA) 508/369-5115

 Switching CMD unit to 4 to read F18BAS.CMD
ACSL>start                                          Start case (1)
 >>> READING VALUES TO INIT TRIM SEARCH (F18TR FILE)
 >>> READING AERO DATA
        TIME ELAPSED =     7.230469     SECONDS
 >>> READING ENGINE DATA
        TIME ELAPSED =     1.339844     SECONDS
NORM OF Y ON FIRST PASS =    3.0825590E-03
COLUMN            2 NOT REPEATABLE. ERROR =    6.519258022308350E-009
TRIM ACHIEVED. ERROR =    4.4315234E-06
ITERATIONS REQUIRED =            27
t exceeds tstp
ACSL>s  tcwprn=80
ACSL>d  machtr,hic,tcase,alfdg,thedg,pcstr,dpsytr
      MACHTR 0.60000000          HIC 10000.0000          TCASE       1
       ALFDG 2.47177000        THEDG 2.47138000        PCSTR-0.06876620
      DPSYTR 60.4787000
ACSL>d  ud,vd,wd,pd,qd,rd
         UD-2.3931E-04            VD 0.                    WD 1.3487E-04
         PD 0.                    QD-1.3995E-05            RD 0.
ACSL>s  tcase=2,llimdf=.t.,nxtr=8      Convert from wings level to a trim at bank
                                       angle using the TCASE=2 option. Expand
                                       trim from 4x4 to 8x8 . NYTR will be
                                       automatically set to NXTR.

ACSL>s  ixsel=15,13,2,8,7,3,14,16
ACSL>s  iysel=6,4,2,11,1,3,5,7
ACSL>s  mudgtr=30                      Choose desired bank angle.
ACSL>start                             Start case (2)
NORM OF Y ON FIRST PASS =    7.2131425E-02
MUMAX REACHED. CALCULATE NEW SENSITIVITY MATRIX.
NORM OF Y ON FIRST PASS =    3.7534259E-02
TRIM ACHIEVED. ERROR =    2.8616143E-05
ITERATIONS REQUIRED =            46
t exceeds tstp
ACSL>d  machtr,hic,phidg,mudgtr,az,alfdg,thedg,phidg,dpsytr
      MACHTR 0.60000000          HIC 10000.0000        PHIDG 30.0299000
      MUDGTR 30.0000000           AZ-1.15330000        ALFDG 2.81582000
       THEDG 2.42915000        PHIDG 30.0299000      DPSYTR 61.2118000
ACSL>d  ud,vd,wd,pd,qd,rd
         UD 2.3536E-06            VD 6.9801E-05            WD-1.3113E-05
         PD-2.2889E-04            QD-1.9547E-06            RD 1.6206E-06
```

```
ACSL>s  mudgtr=60
ACSL>start                                        Start case (3)
NORM OF Y ON FIRST PASS =    0.1437438
COLUMN            8 NOT REPEATABLE. ERROR =    1.303851604461670E-008
TRIM ACHIEVED. ERROR =    1.1091912E-05
ITERATIONS REQUIRED =            26
t exceeds tstp
ACSL>d  machtr,hic,phidg,mudgtr,az,alfdg,thedg,phidg,dpsytr
      MACHTR 0.60000000          HIC 10000.0000         PHIDG 60.0811000
      MUDGTR 60.0000000           AZ-1.99337000         ALFDG 4.67035000
       THEDG 2.30884000        PHIDG 60.0811000       DPSYTR 76.6398000
      2 g's of normal load (stability frame) are produced by 60° of bank angle.  AZ is in the
      body frame and is slightly different from NZ in the stability frame.
ACSL>s  lwrtr=.t.                                 Set flag so next successful trim results are
                                                  saved to f18tr.dat.
ACSL>start                                        Repeat case (3) to cause trim data to be
                                                  written to file 'f18tr.dat'.
NORM OF Y ON FIRST PASS =    1.1091912E-05
ALREADY TRIMMED.
t exceeds tstp
ACSL>stop                                         End session
End ACSL run
```

Steady State Pull-Up (TCASE=3)

The following steady-state pull-up (TCASE=3) trim cases are demonstrated by an example terminal session. The session is begun by reading in the TCASE=1 trim data file f18tr.dat, then using run time commands to set up the pull-up trim.

(1) M=.6, H=10,000 (ft), wings level, TCASE=1, flight path angle (γ) = 0.
(2) M=.6, H=10,000 (ft), TCASE=3, GLOAD (n_z) = 2 g's.
(3) M=.6, H=10,000 (ft), TCASE=3, GLOAD (n_z) = 4 g's.

```
AGCB9$ copy  f18tr_tc1.dat  f18tr.dat   Copy f18tr.dat file used for TCASE=1 trim
                                        cases into generic file read by f18bas
                                        simulation.
AGCB9$ acsl/run  f18bas

 The ACSL Licence will terminate in  3 months
 Ask your systems manager to contact Mitchell and Gauthier for new key
 Telephone (USA) 508/369-5115

 Switching CMD unit to 4 to read F18BAS.CMD
ACSL>start                              Start case (1)
 >>> READING VALUES TO INIT TRIM SEARCH (F18TR FILE)
 >>> READING AERO DATA
       TIME ELAPSED =    7.351563    SECONDS
 >>> READING ENGINE DATA
       TIME ELAPSED =    1.492188    SECONDS
NORM OF Y ON FIRST PASS =   3.0825590E-03
COLUMN            2 NOT REPEATABLE. ERROR =   6.519258022308350E-009
TRIM ACHIEVED. ERROR =   4.4315234E-06
ITERATIONS REQUIRED =           27
t exceeds tstp
ACSL>s  tcwprn=80
ACSL>d  tcase,alfdg,thedg,machtr,hic
       TCASE      1            ALFDG 2.47177000        THEDG 2.47138000
      MACHTR 0.60000000          HIC 10000.0000
ACSL>d ixsel                            Selected XTRIM variables
       IXSEL     14                    16                    3
                  7                     0                    0
                  0                     0                    0
                  0
ACSL>d iysel                            Selected YTRIM variables
       IYSEL      1                     3                    5
                  7                     0                    0
                  0                     0                    0
                  0
ACSL>d az
         AZ-0.99906600
ACSL>s  tcase=3,gload=2                 TCASE=3 sets pitch rate automatically for a
                                        pull-up with load (g's) = GLOAD.
ACSL>start                              Start case (2)
NORM OF Y ON FIRST PASS =   0.2461748
COLUMN            2 NOT REPEATABLE. ERROR =   1.220281404812340E-007
TRIM ACHIEVED. ERROR =   7.6116430E-06
ITERATIONS REQUIRED =           10
t exceeds tstp
ACSL>d  tcase,alfdg,thedg,machtr,hic,az
       TCASE      3            ALFDG 4.67142000        THEDG 4.67179000
      MACHTR 0.60000000          HIC 10000.0000          AZ-1.99336000
```

```
ACSL>d  ud,wd,qd
         UD 5.5424E-04           WD-2.6588E-04           QD-2.2835E-05
ACSL>s  gload=4
ACSL>start                                        Start case (3)
NORM OF Y ON FIRST PASS =    0.4928297
COLUMN            3 NOT REPEATABLE. ERROR =    3.632492280303268E-007
TRIM ACHIEVED. ERROR =    5.9509330E-07
ITERATIONS REQUIRED =           20
t exceeds tstp
ACSL>d  dpsytr,q                                  Show required throttle (%) and pitch rate
                                                  (r/s).
      DPSYTR 112.504000            Q 0.14931300
ACSL>d  tcase,alfdg,thedg,machtr,hic,az
       TCASE      3            ALFDG 8.99947000          THEDG 8.99940000
      MACHTR 0.60000000           HIC 10000.0000             AZ-3.95076000
ACSL>stop
End ACSL run
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1992	Technical Memorandum

4. TITLE AND SUBTITLE

Simulation Model of a Twin-Tail, High Performance Airplane

5. FUNDING NUMBERS

505-64-30-01

6. AUTHOR(S)

Carey S. Buttrill, P. Douglas Arbuckle, and Keith D. Hoffler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23665-5225

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

NASA TM-107601

11. SUPPLEMENTARY NOTES

Buttrill and Arbuckle: Langley Research Center, Hampton, VA; Hoffler: ViGYAN, Inc., Hampton, VA
This research was supported in part by the National Aeronautics and Space Administration under contract number NAS1-18585.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Unclassified - Unlimited

Subject Category 08, 05

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The mathematical model and associated computer program to simulate a twin-tailed high performance fighter airplane (McDonnel Douglas F/A-18) are described. The simulation program is written in the Advanced Continuous Simulation Language language. The simulation math model includes the non-linear, six degree-of-freedom rigid-body equations, an engine model, sensors, and first order actuators with rate and position limiting. A simplified form of the F/A-18 digital control laws (version 8.3.3) are implemented. The simulated control law includes only inner loop augmentation in the up and away flight mode. The aerodynamic forces and moments are calculated from a wind-tunnel-derived database using table look-ups with linear interpolation. The aerodynamic database has an angle-of-attack range of -10° to +90° and a sideslip range of -20° to +20°. The effects of elastic deformation are incorporated in a quasi-static-elastic manner. Elastic degrees of freedom are not actively simulated. In the engine model, the throttle-commanded steady-state thrust level and the dynamic response characteristics of the engine are based on airflow rate as determined from a table look-up. Afterburner dynamics are switched in at a threshold based on the engine airflow and commanded thrust.

14. SUBJECT TERMS

twin-tailed high performance fighter, simulation, six degree-of-freedom quasi-rigid mathematical model, high angle-of-attack

15. NUMBER OF PAGES

180

16. PRICE CODE

A09

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		