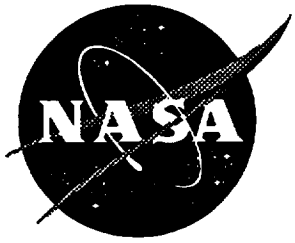
NASA Technical Memorandum 110216



Simulation Model of the F/A-18 High Angle-of-Attack Research Vehicle Utilized for the Design of Advanced Control Laws

Michael D. Messina
Lockheed Martin Engineering & Sciences Company, Hampton, Virginia

Mark E. Strickland
Langley Research Center, Hampton, Virginia

Keith D. Hoffler
ViGYAN, Inc., Hampton, Virginia

Susan W. Carzoo
Unisys Corporation, Hampton, Virginia

W. Thomas Bundick
Langley Research Center, Hampton, Virginia

Jessie C. Yeager
Lockheed Martin Engineering & Sciences Company, Hampton, Virginia

Fred L. Beissner, Jr.
Lockheed Martin Engineering & Sciences Company, Hampton, Virginia

May 1996

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

SUMMARY

This report and a previous report, NASA TM 107601, "Simulation Model of a Twin-Tail, High Performance Airplane," (ref. 1.0), document the aircraft model and computer program which form the nonlinear, six degree-of-freedom, batch simulation of the High Alpha Research Vehicle (HARV), a highly modified F/A-18. Reference 1.0 describes the *f18bas* simulation of a basic F/A-18 modified with a hypothetical thrust-vectoring system. This report describes modifications to that simulation to form the simulation of the HARV called *f18harv*.

The HARV is an F/A-18 airplane modified to incorporate a multi-axis thrust-vectoring system for augmented pitch and yaw control power and actuated forebody strakes for enhanced aerodynamic yaw control power. The HARV is used in NASA's High Alpha Technology Program (HATP) to conduct flight research in advanced control effectors, advanced control law design methodologies, and flight dynamics issues at high angles of attack. An example of advanced control effector research is the Actuated Nose Strakes for Enhanced Rolling (ANSER) experiment.

The *f18harv* simulation is written in Advanced Continuous Simulation Language (ACSL) and FORTRAN languages. Significant modifications to the *f18bas* simulation to form *f18harv* include implementation of the HARV aerodynamic database to account for thrust-vectoring induced effects, implementation of a different engine model, implementation of a model of the HARV thrust-vectoring system, addition of high-fidelity sensor models, addition of the aerodynamic database for ANSER control devices, and addition of control surface actuator model enhancements. Other modifications such as weight and balance updates were required to match the performance of the F/A-18 HARV aircraft more closely. The resulting simulation was used in the design of advanced HARV control laws, which were implemented in the simulation.

(Intentional blank page)

TABLE OF CONTENTS

1.0 INTRODUCTION

The *f18harv* six degree-of-freedom nonlinear batch simulation used to support research in advanced control laws and flight dynamics issues as part of NASA's High Alpha Technology Program is described in this report. This simulation models an F/A-18 airplane modified to incorporate a multi-axis thrust-vectoring system for augmented pitch and yaw control power and actuated forebody strakes for enhanced aerodynamic yaw control power. The modified configuration is known as the High Alpha Research Vehicle (HARV) (fig. 1.1). The *f18harv* simulation was an outgrowth of the *f18bas* simulation (ref. 1.0) which modeled the basic F/A-18 with a preliminary version of a thrust-vectoring system designed for the HARV. The preliminary version consisted of two thrust-vectoring vanes per engine nozzle compared with the three vanes per engine actually employed on the F/A-18 HARV. The modeled flight envelope is extensive in that the aerodynamic database covers an angle-of-attack range of -10° to +90°, sideslip range of -20° to +20°, a Mach Number range between 0.0 and 2.0, and an altitude range between 0 and 60000 feet.

1.1 Simulation Development

Since the initial release of *f18bas* , a continuous stream of updates has been made to the simulation. The major changes were 1) replacement of the hypothetical two-vane thrust-vectoring model with a three-vane model representative of the HARV Thrust-Vectoring System (TVS), 2) replacement of the engine model, 3) addition of the aero database modeling the effects of the TVS on the basic F/A-18 model, 4) addition of the aero database modeling the Actuated Nose Strakes for Enhanced Rolling (ANSER), 5) addition of the Mixer/Predictor to distribute TV commands to the proper vanes, 6) addition of research flight control laws, and 7) refinement of the sensor models to more accurately model the HARV sensors. Additional changes were made for various other reasons that included addition of new capabilities such as generation of output for use in other off-line programs like Getdata, Xplot, Quickplot, Matlab, and MATRIX$_X$®, correction of modeling differences between this simulation and the Dryden Flight Research Center (DFRC) simulations, and correction of implementation discrepancies. Changes required for trim and implementation of a new actuator model were also included.

This report is intended to provide one-source documentation of the changes made from the original *f18bas* to the current *f18harv* simulation. Reference 1.0 documents the aerodynamic model of the basic F/A-18 in detail, and this report does not repeat the information in reference 1.0 but rather complements it.

1.2 HARV Simulations

As previously stated the *f18harv* simulation was an outgrowth of the *f18bas* simulation used in high-angle-of-attack research. Parallel to this simulation development effort, DFRC developed various simulations to support the HARV. Four different simulations resulted from the DFRC work. The first was a UNIX-based batch simulation which runs on a SUN Sparcstation. This simulation contains the basic F/A-18 control system and the ability to run the ADA flight code recompiled using a Sun ADA compiler. This batch simulation contained the fade logic to engage/disengage the Research Flight Control System (RFCS) from the basic F/A-18 control system. The second DFRC simulation is called the All-Software simulation. The configuration is identical to the batch simulation except that the actual Mission Computer (MC) is used instead of the FORTRAN models developed for the batch simulation. The third simulation is called the Hardware-In-the-Loop simulation (HIL). This simulation contains the flight hardware processors designated as the 701E and 1750A. The 701E contains the basic F/A-18 control law, and the 1750A contains the RFCS control laws programmed in ADA. Also included are the 1553 data BUS, Mission Computer, Digital-to-Analog (D/A) converters, Ana-

log-to-Digital (A/D) converters, and analog actuator models. An F/A-18 cockpit is incorporated into this setup. The engine and aero models used in the HIL are the same as those used in the batch simulation. A fourth simulation is called the Iron-Bird (IB) and uses an actual F/A-18 and the actual F/A-18 actuators, thus adding another level of accuracy.

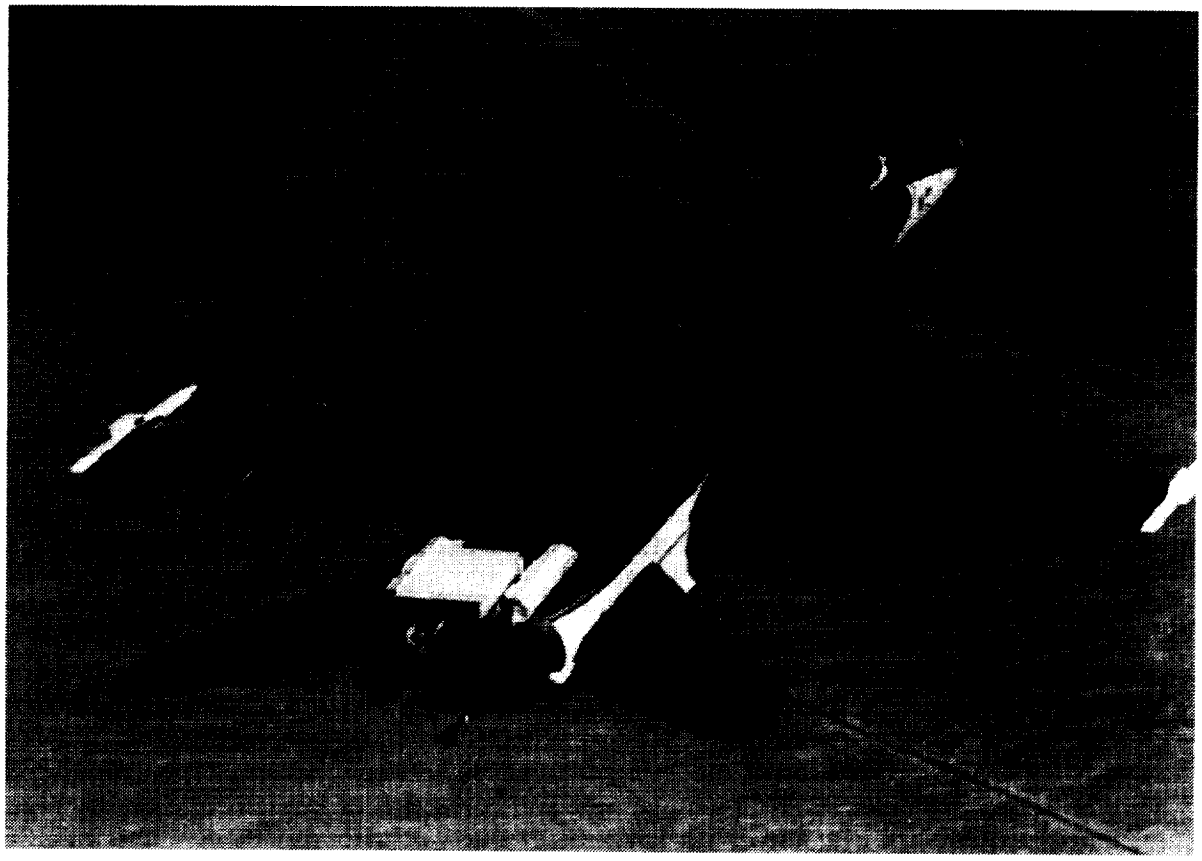


Figure 1.1. Photograph of the F/A-18 HARV showing addition of thrust-vectoring vanes.

The *f18harv* simulation was modified to achieve a match between *f18harv* results and results from the DFRC batch simulation. The Dryden simulation models were considered to be the truth model for the airplane since considerable work was done during model development comparing DFRC batch results to HIL and flight results. This meant that some of the code developed for *f18bas* was replaced with models taken from the DFRC batch simulation. Specifically, the replaced models consisted of the actuator dynamics, sensor models, and engine dynamics, as previously mentioned.

The DFRC Sun-based batch simulation was used also at the Langley Research Center (LaRC) for HARV control law design and analysis, but this report documents the LaRC *f18harv* batch simulation implemented in Advanced Continuous Simulation Language (ACSL) and FORTRAN. A FORTRAN real-time version of *f18harv* was implemented in the LaRC Differential Maneuvering Simulator (DMS) to support piloted simulation, but differences between *f18harv* and the DMS will not be discussed in this report other than to say that aircraft performance in the two is nearly identical.

2.0 HARV SIMULATION STRUCTURE

The general structure of the *f18harv* simulation is the same as that of the *f18bas* simulation. The description presented in Section 2 of reference 1.0 is repeated in the following two sections with minor changes to update the information to the *f18harv* simulation.

2.1 Overview

The ACSL code structure shown in figure 2.1 contains the general outline of the *f18harv* simulation. The DYNAMIC...END block of figure 2.1 defines the main simulation loop of *f18harv* . This 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 (ref. 1.1). The communication interval determines how often the computer executes that code in the DYNAMIC block that is not part of either the DERIVATIVE block or the DISCRETE blocks. The default value for CINT in *f18harv* is 0.0125 (1/80) seconds.

The maximum integration step for the DERIVATIVE block is defined by MXSTP and has a default value of 0.0125 (1/80) seconds. The integration algorithm is selected by setting IALG. A variety of integration schemes are available (IALG = 1 through 7) and are described in reference 1.1. The default choice for the 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 *dname1* in figure 2.1 is an example of an interval block that gets executed every *Time1* seconds. DISCRETE block *dname2* in figure 2.1 is executed whenever the logical condition defined in its SCHEDULE command becomes true. In the *f18harv* simulation, the digital control law DISCRETE block FCS is an INTERVAL block and is called every TFCS seconds. The default value for TFCS is 0.0125 seconds. When different DISCRETE blocks are scheduled for execution at the same time, the order of execution is determined by order of appearance in the code. Scheduling for same-time execution might occur when a DISCRETE block interval is an even multiple of MXSTP or some other DISCRETE block interval. The DISCRETE interval period is not required to be a multiple of the maximum integration size for the DERIVATIVE block. ACSL will shorten the last integration step to insure all simulation variables are synchronized at the correct interrupt time.

Not shown in figure 2.1 are the FORTRAN subroutines that can be called by CALL statements in the ACSL portion of the program. In *f18harv* extensive use is made of FORTRAN subroutines for such tasks as modeling sensors and implementing control laws.

The paths in figure 2.2 associated with the variables LTR, ITFLG, LWRTR and LDBTR represent code installed to trim the simulation. This path is not necessitated by any of the typical ACSL constructs found in figure 2.1. The trim logic is discussed in more detail in section 2 of reference 1.0.

Two sections of the simulation flow shown in figure 2.2 are expanded in figure 2.3. The actions produced by CALLing subroutine LOAD in the INITIAL...END code block are detailed, and some of the discrete blocks in the simulation are shown. The sensor model has five discrete blocks called ADC, FCSENS1, INS, AINS, and TDSENS. The discrete blocks

```
PROGRAM
    MACRO mname1
    MACRO mname2
    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 dname2  .XP. var2
        END
        DISCRETE dname1
            INTERVAL Time1  = number
            An INTERVAL scheduled DISCRETE includes statements that are executed every
            Time1 seconds.
        END
        DISCRETE dname2
            Includes statements to be executed when the scheduling conditions calculated in the
            DERIVATIVE block (in this case) for DISCRETE dname2  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

that model the Research Flight Control System (RFCS) and provide a mechanism for making comparisons with flight data (FDT) are invoked by INTERVAL commands. The discrete block for the engine model which CALLs the Mixer/Predictor (M/P) and Thrust Estimator

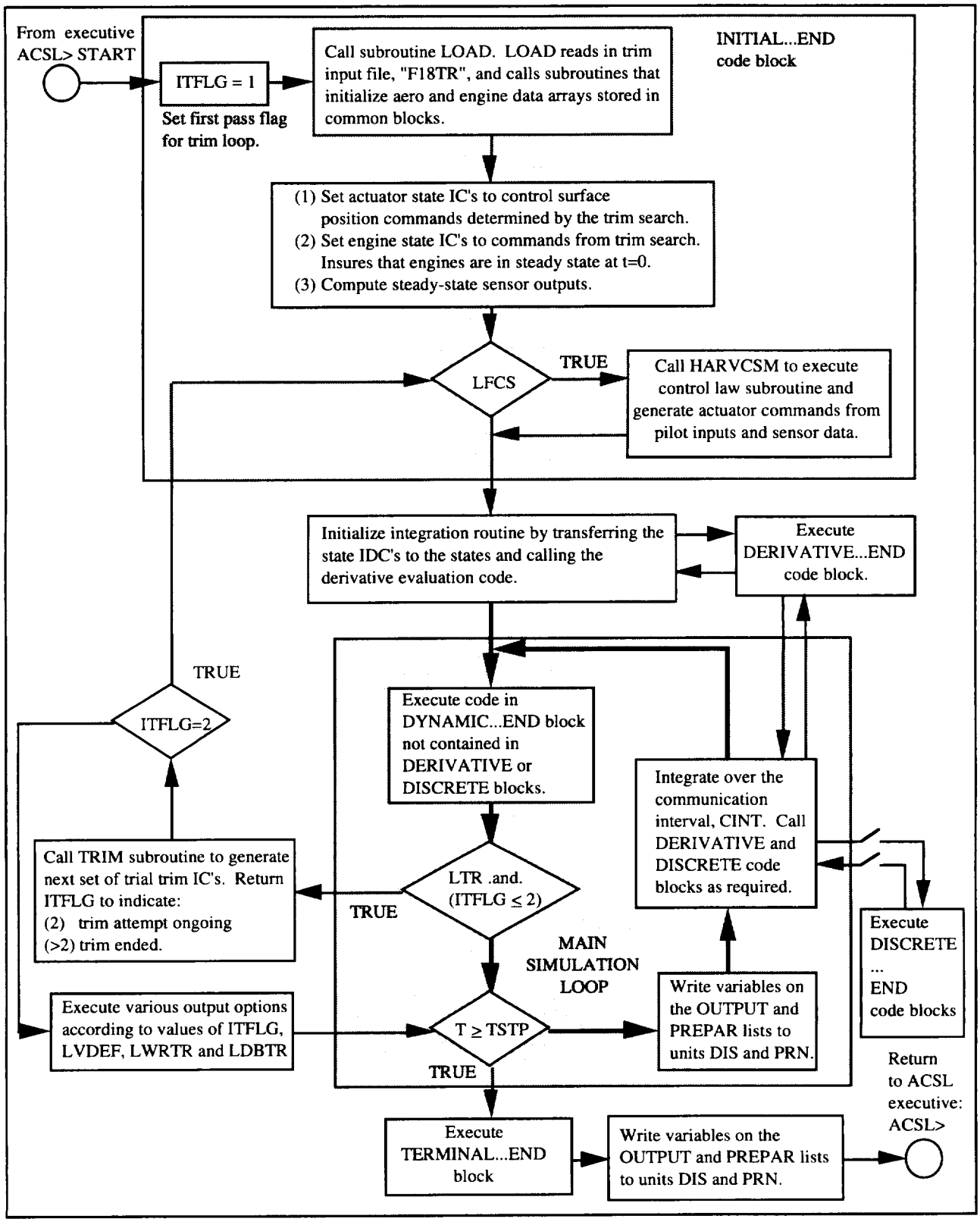


Figure 2.2. - Overview of simulation flow

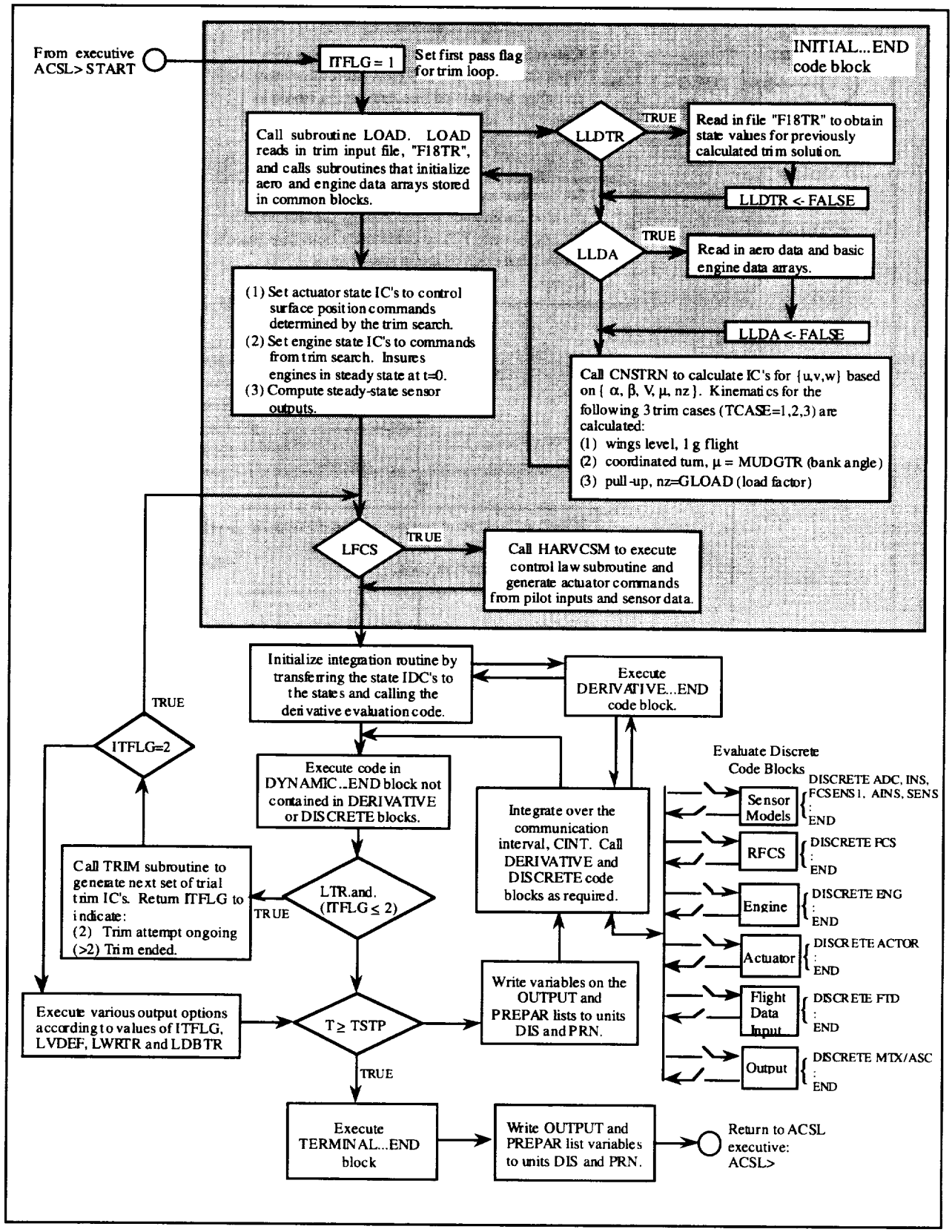


Figure 2.3. - Initialization and Discrete Block Details

(TE) is called ENG. The actuator model is CALLed from the discrete block ACTOR. Two output files in addtion to the ACSL output capability can be generated in the discrete blocks MTX and ASC. More information on output capability is presented in section 8.2.

2.2 Trimming

ACSL has a TRIM command that works with purely continuous models. The ACSL TRIM command selects a subset of the simulation states (outputs of integrations) by 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 to adjust the states. (sec. 5.2.4 of ref. 1.1) For example, a simple longitudinal level-flight trim problem might be to:

find

$$\{u\} = \left[\delta_e, T, \alpha, \theta\right]' \tag{2.1}$$

such that

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

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, while 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. Using 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 a nonzero value. The states $\{\delta_e, T, \alpha\, \theta, q\}$ can then be released and the ACSL command "ANALYZ TRIM" invoked. ACSL will then drive $\left\{\dot{\delta}_e, \dot{T}, \dot{\alpha}, \dot{\theta}, \dot{q}\right\}$ 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. Thus, the trim logic in figure 2.4 was developed for the *f18harv* simulation. Subroutine TRIM defines arrays called XTRIM and YTRIM which have been modified in the *f18harv* simulation. The allowable elements of

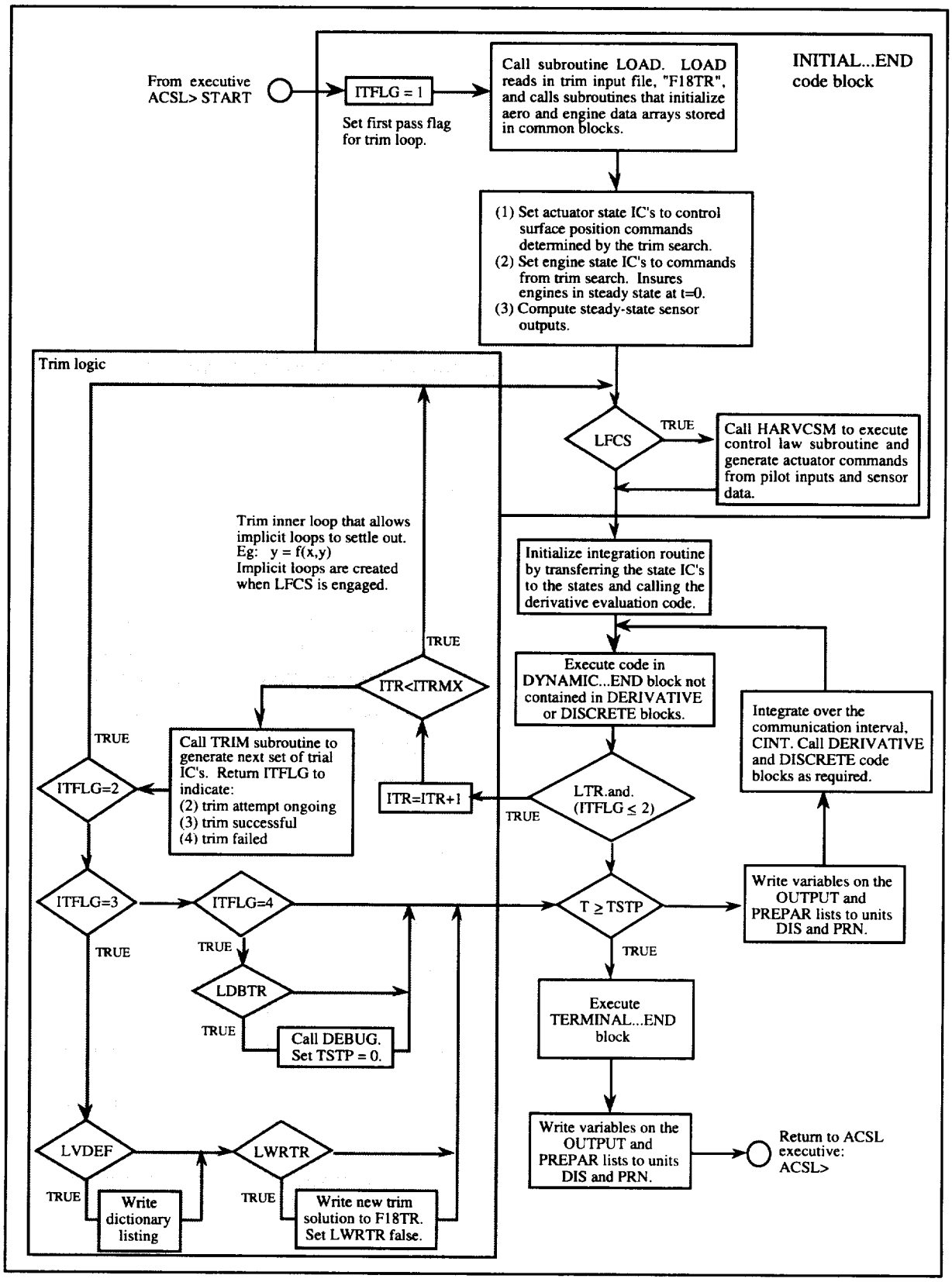


Figure 2.4. Trim Loop Expanded.

XTRIM and YTRIM are shown in tables 2.1 and 2.2, respectively. All trim subroutines in the *f18harv* simulation were changed to read input from and write output to the trim file in degrees rather than in radians. The XTRIM and YTRIM arrays in the trim file were changed to reflect degrees. The input/output units were converted and trim computations using XTRIM and YTRIM are still computed in radians. Also, the XTRIM and YTRIM arrays were modified in the simulation to reflect the updated engine model and M/P and to provide additional trim parameters associated with the inclusion of the nose-strake aero model into *f18harv*. The XTRIM array was modified so that PJETTR and YJETTR replaced the parameters DTVL and DTVR in *f18bas*, while the parameters RJETTR, DTASTR, and DTSYTR were added. The array YTRIM was modified by adding the parameter ERRORS to trim RFCS. The parameters ALFDDG and BETDDG were added to YTRIM for new trim capabilities.

Table 2.1. Selectable members of XTRIM array

1	MACHTR	Mach number, n.d.
2	BETTRDG	angle of sideslip, β, degrees
3	ALFTRDG	angle of attack, α, degrees
4	PICDG	body-axis roll rate, deg/sec
5	QICDG	body-axis pitch rate, deg/sec
6	RICDG	body-axis yaw rate, deg/sec
7	THETRDG	Euler pitch angle, degrees
8	PHITRDG	Euler roll angle, degrees
9	PSITRDG	Euler heading angle, degrees
10	GAMTRDG	longitudinal flight path angle, γ, degrees
11	PJETTR	total pitch-jet-angle deflection, degrees
12	YJETTR	total yaw-jet-angle 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, degrees
17	DPASTR	throttle position, antisymmetric, degrees
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	DELSTM	sum of stabilator and pitch-jet commands, degrees
31	RJETTR	total roll-jet-angle deflection, degrees
32	DTSYTR	nose-strake position, symmetric, degrees
33	DTASTR	nose-strake position, antisymmetric, degrees

Table 2.2. Selectable members of YTRIM array

1	UD	d/dt (x-body component of inertial velocity), ft/sec^2
2	VD	d/dt (y-body component of inertial velocity), ft/sec^2
3	WD	d/dt (z-body component of inertial velocity), ft/sec^2
4	PDDG	d/dt (body-axis roll rate), degrees/sec^2
5	QDDG	d/dt (body-axis pitch rate), degrees/sec^2
6	RDDG	d/dt (body-axis yaw rate), degrees/sec^2
7	GAMZRDG	= GAMDG - GAMTRDG = $\gamma - \gamma_{TR}$, degrees
8	PHIZRDG	= PHIDG - PHITRDG = $\phi - \phi_{TR}$, degrees
9	THEDG	Euler pitch angle, degrees
10	LAMBADG	lateral flight path angle, λ, degrees
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	ERRORS	error signal out of proportional gain in RFCS
15	ALFDDG	d/dt (angle of attack at the c.g.), degrees/sec^2
16	BETDDG	d/dt (angle of sideslip at the c.g.), degrees/sec^2

The elements of XTRIM and YTRIM are selected by setting the first NXTR and NYTR elements of the IXSEL and IYSEL arrays, respectively. A typical 5×5 longitudinal trim at a specified low angle of attack, flight path angle, and altitude could be set up as indicated in figure 2.5. In this example, altitude, angle of attack, and flight path angle are selected a priori, and the variables in the set {Mach number, θ, power setting, $\delta_{H_{symm}} + \delta_{Pitch\_jet\_cmd}$, longitudinal stick} are varied until the variables in the set $\{\dot{u}, \dot{w}, \dot{q}, \gamma - \gamma_{TR}$, integrator error$\}$ are driven to zero.

```
ACSL>    LTHVEC=.T., LFCS=.T.,
ACSL>    ALFTRDG=5., GAMTRDG=5., HIC=25000.
ACSL>    NXTR = 5
ACSL>    IXSEL=1,7,16,30,14
ACSL>    NYTR = 5
ACSL>    IYSEL=1,3,5,7,14
ACSL>    START
```

Figure 2.5. Trim example 1

Trim example 2 in figure 2.6 is illustrative when a high-α trim is sought. The throttle setting command and flight path angle are deleted from the list of parameters to be varied.

```
ACSL>    LTHVEC=.T., LFCS=.T.,
ACSL>    ALFTRDG=45., HIC=25000.
ACSL>    NXTR = 4
ACSL>    IXSEL=1,7,14,30
ACSL>    NYTR = 4
ACSL>    IYSEL=1,3,5,14
ACSL>    START
```

Figure 2.6. Trim example 2

Trim example 3 in figure 2.7 is slightly different in that trim at a specified Mach number

and altitude is sought. This example is like that of Figure 2.5 with Mach number replacing α in the list of parameters to be varied.

```
ACSL>    LTHVEC=.T., LFCS=.T.,
ACSL>    MACHTR=.4, HIC=25000.
ACSL>    NXTR = 5
ACSL>    IXSEL=3,7,14,16,30
ACSL>    NYTR = 5
ACSL>    IYSEL=1,3,5,7,14
ACSL>    START
```

Figure 2.7. Trim example 3

Trim example 4 in figure 2.8 is a combination of trim examples 1 and 3. For this example, trim at a specified angle of attack, Mach number, and altitude is sought, and throttle setting replaces both α and Mach number in the list of parameters to be varied.

```
ACSL>    LTHVEC=.T., LFCS=.T.,
ACSL>    ALFTRDG=20., MACHTR=.4, HIC=25000.
ACSL>    NXTR = 5
ACSL>    IXSEL=5,7,14,16,30
ACSL>    NYTR = 5
ACSL>    IYSEL=1,3,5,7,14
ACSL>    START
```

Figure 2.8. Trim example 4

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 associated 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. 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 also on past values of {YTRIM}. The degradation of the functional relationship will impede or prevent convergence. The somewhat ad hoc solution adopted 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 six. 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.

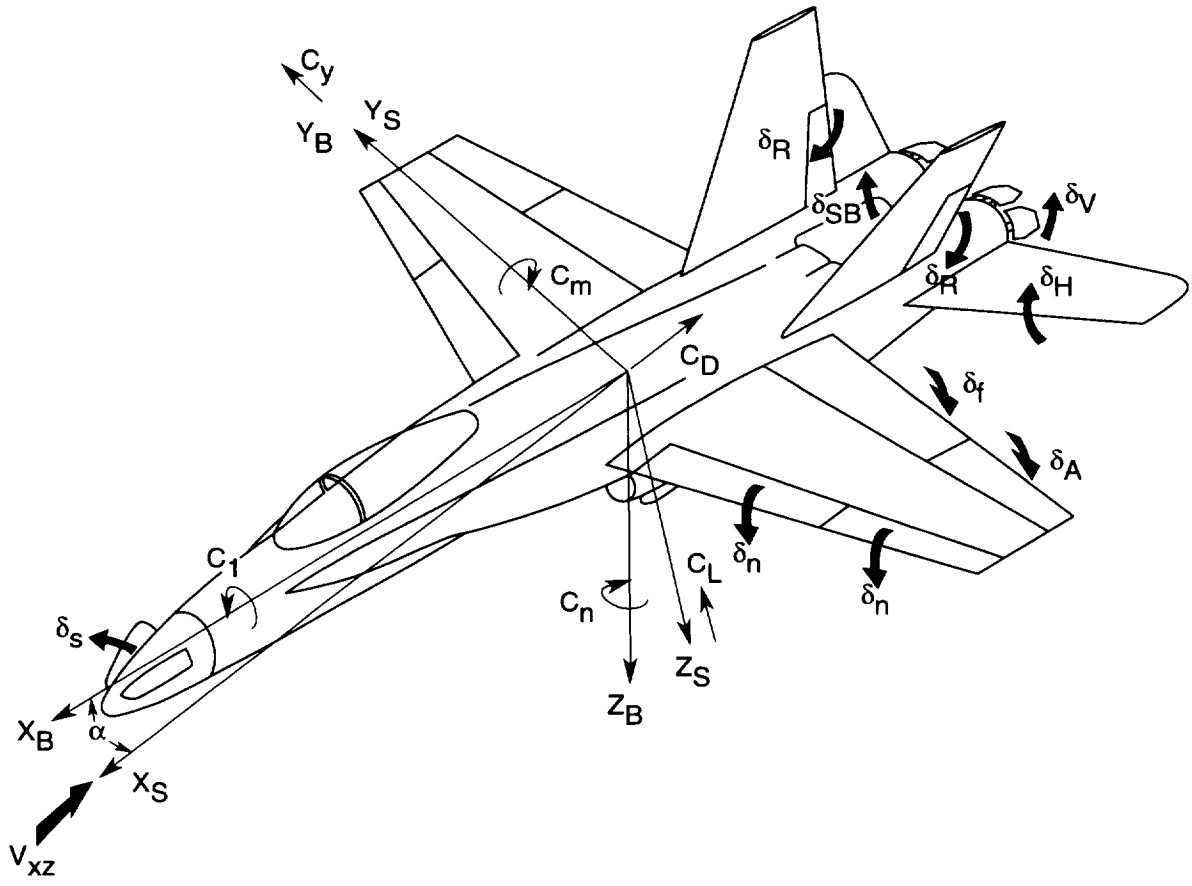
2.3 HARV Dimensions, Weight, and Inertia

The axis system and sign convention presented in section 3.2 of reference 1.0 remains entirely the same. Figure 2.9 shows the axis system and sign convention with the addition of the nose strakes. When the strake conforms to the aircraft forebody, the deflection is zero degrees; when the strake is fully deployed, the deflection is +90 degrees. Major dimensions and key features of the F/A-18 HARV thrust vectoring system is shown in figure 2.10. The thrust-vectoring capability was added to the basic F/A-18 by removing the secondary (divergent) nozzles and adding three thrust-vectoring vanes per engine. The spin chute was mounted on top of the tail section aft of the vertical tails. A detailed sketch of the modified tail section is shown in figure 2.10.

The weight, center-of-gravity location, and inertia data for the *f18harv* simulation is shown in table 2.3. The Fighter Escort configuration, which represents the F/A-18 HARV aircraft with 60 percent internal fuel, served as the baseline configuration for the RFCS design. Linear models were produced using the light (minimum fuel) and heavy (maximum fuel) configurations to test effects of weight changes on stability margins of the RFCS design. Mass properties do not change significantly when adding the actuated forebody strakes because ballast is removed from the forebody to offset the additional weight of the strakes and associated actuators. The light, heavy, and empty weight configurations are listed for reference in table 2.3.

Table 2.3. Summary of Simulation Weight, Center-of-Gravity, and Inertia

	Weight (lbs)	c.g. Locations FS (inches)	WL	Moments and Products of Inertia I_{xx}	I_{yy} (slug - ft^2)	I_{zz}	I_{xz}
Fighter Escort (60% internal fuel)	35764.6	456.3	105.4	22632.6	174246.3	189336.4	-2131.8
Light Weight	31617.7	460.89	103.36	22163.0	172237.5	186823.1	-2043.1
Heavy Weight	37618.7	456.31	105.91	22938.3	179130.1	194003.0	-2507.3
Empty Weight	29615.0	463.75	102.31	21643.3	171184.9	185408.8	-1967.4



Note: Direction of arrows indicate + values

Figure 2.9. F/A-18 HARV Axis System and Sign Convention

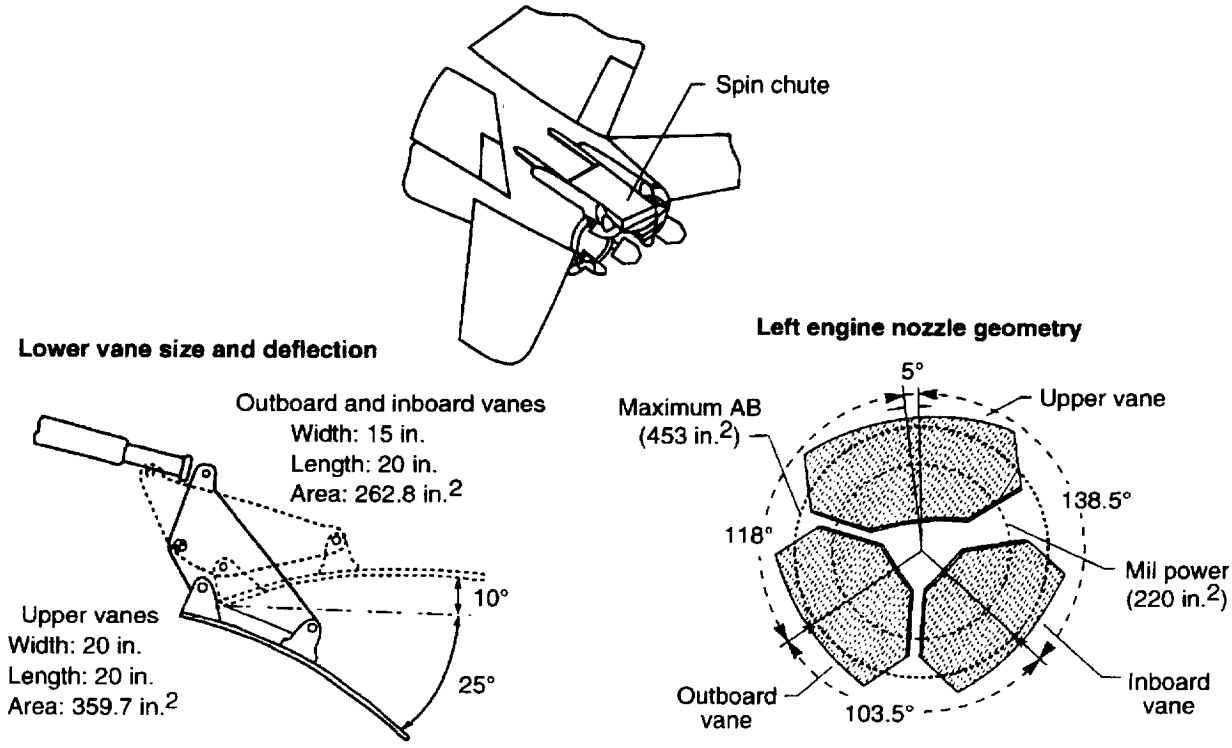


Figure 2.10. Detail sketch of F/A-18 HARV Thrust Vectoring and Spin Chute Implementation.

2.4 HARV Simulation File Structure

The *f18harv* simulation has been re-hosted from a VAX to a Sun SPARCStation Unix platform. The *f18bas* simulation was run entirely on the VAX platform as were early versions of the *f18harv* simulations. The file *f18bas.for* contained source code from the original *f18bas* simulation that was not changed in *f18harv*. To take advantage of the compilation features in Unix, all FORTRAN subroutines and functions were removed from the files *f18harv.csl* and *f18bas.for* separating the *f18harv* simulation into two parts. The first portion consists of only the ACSL source code and is contained in the file *f18harv.csl*. The second portion of the simulation contains nine files of FORTRAN source code, which have been grouped by modeling subjects. This separation of files enabled the simulation user to compile modifications made to a specific portion of the source code without having to recompile the entire simulation. The $MATRIX_X$® FORTRAN Autocode source code is all contained in one directory named *Rfcs*. The nine files of FORTRAN source code are listed in table 2.4 by name, and a list of all subroutines and functions in each file is included for reference.

Table 2.4. FORTRAN Source Code Description

FILE	SUBROUTINE	FUNCTION
f18act.f	actdef	hng2ca
	actdyn	hng2cb
	actmod	hng2cc
	actrli	hng2cd
	actset	hng2ce
	actsin	hng2cf
	hinge	hng2cg
	hngin	hng3ea
	hngindx	hng3eb
	hngtlu	hng3ec
	hyd18	hng3ed
	shmknx	hng3ee
	spedbrk	hng3ef
	sth	hng3eg
	sthngi	hng3eh
	stkhng	hng3ei
	stknlg	hng4ga
	strkact	hng4gb
	s2bcoef	hng4gc
	s2coef	hng4gd
		hng4ge
		hng4gf
		hng4gg
		hng4gh
		sth1aa
		sth2ca
		sth2cb
		sth2cc
		sth2cd
		sth2ce
		sth2cf
		sth2cg
		sth2ch
		sth2ci
		sth2cj

Table 2.4. Continued

FILE	SUBROUTINE	FUNCTION
f18aero.f	frwgc1	xlim
	aeroinc	
	afusch	
	iaero	
	in3060	
	tlu3060	
	usaerrf	
	sfaerrf	
	fsdfo	
	nstat	
	tvcoef	
	actrml	
f18bas.f	info	
	load	
	marrc	
	mativx	
	rdwrtr	
	spare	
	ssdeb	
	ssdebd	
	timeck	
	trim	
	unew	
	w2body	
	xform	
	xrbvdc	
	yform	
f18eng.f	inteng	
	harvin	
	vanein	
	engdin	
	engmod	
	tvact	
	tvhinge	
	engmdl	
	tluhrv	
	tvangl	
	tvceng	
f18ftd.f	ftdcdt	
	ftdfcm	
	ftdrdf	
	ffdwrf	

Table 2.4. Concluded

FILE	SUBROUTINE	FUNCTION
f18misc.f	nozero	climit
	search	deadband
		f1
		f2
		f3
		funcx1
		funcx2
		oned
		pwtus1
		search2
		search3
		sotust
		srch
f18mp.f	thrcal	f_tl
	mixpre	flimit
	mixer_4	pitch_priority
	ntrp0h	rlimit
	ntrp1h	vane_limit
	mixer_init	yaw_priority
	tv_vanes	
f18output.f	cmpfm	
	matsav	
	mtxfm	
	mtxwrf	
f18sensor.f	pstat	alteqv
	atmat62	
	insmdl	
	sensor	
	aoains	
	accel	

3.0 AERODYNAMIC MODEL CHANGES

The force and moment coefficients, which are the outputs from subroutine SFAERRF (see reference 1.0, section 5), consist of the aerodynamic coefficients for the F/A-18 aircraft configuration only. Aerodynamic model changes were implemented to account for HARV specific geometry and jet induced effects resulting from installation and use of thrust vectoring. The modeled geometry changes from a "basic F/A-18" included removal of the secondary (divergent) nozzles, addition of vectoring vanes and associated actuator housings, and addition of spin chute canister. Spin-chute deployment is not modeled. The jet induced effects model accounts for the impact of turning a high velocity jet directly aft of the aircraft on the aerodynamics upstream. This is akin to the effect of a blown flap. The thrust-vectoring-induced effects change the fundamental F/A-18 configuration only; these increments do not include any thrust-vectoring engine forces. The Actuated Nose Strakes for Enhanced Rolling (ANSER) system results in additional aerodynamic increments which are summed with the F/A-18 aircraft and HARV-specific increments to produce the total aerodynamic coefficients used in *f18harv*.

3.1 HARV-Specific Aerodynamics

The following equations and text which describe the HARV aerodynamic model in section 3.1 are taken from Reference 3.0. The aerodynamic model was constructed from data collected using the 16% scale F/A-18 in the 30 by 60-Foot Tunnel (LaRC), January-February 1989.

An exhaust-nozzle geometry factor used to compensate for the large A8 exhaust nozzles on the 30 x 60 foot LaRC test of the 16% F/A-18 model is listed in equation 3.1.

$$Geom_{fac} = 0.7 \tag{3.1}$$

The full scale nozzles exhaust area were 520 in^2, whereas the cold jet data was taken at 348 in^2 corresponding to a more representative maximum afterburner nozzle area. No attempt was made to approximate the military power nozzle area of 220 in^2.

The pitch and yaw factors which represent the maximum deflection in each jet deflection direction, up, down, left, and right, are defined as F_p and F_y respectively.

$$F_p = -19.48° \tag{3.2}$$

$$F_y = -11.04° \tag{3.3}$$

$$\delta_p = \frac{\delta_{p_l} T_l + \delta_{p_r} T_r}{T_l + T_r} \tag{3.4}$$

where δ_{p_l} = thrust-vectored pitch jet-turning-angle for the left engine in the x-y engine-axis plane, degrees

δ_{p_r} = thrust-vectored pitch jet-turning-angle for the left engine in the x-y engine-axis plane, degrees

T_l = gross thrust for left engine including effect of T56 bias only, lbs

T_r = gross thrust for right engine including effect of T56 bias only, lbs

Likewise the average jet deflection in yaw δ_y, positive left, is defined by the following equation:

$$\delta_y = \frac{\delta_{y_l} T_l - \delta_{y_r} T_r}{T_l + T_r} \tag{3.5}$$

where δ_{y_l} = thrust-vectored yaw jet-turning-angle for the left engine in the x-z engine-axis plane, degrees

δ_{y_r} = thrust-vectored yaw jet-turning-angle for the right engine in the x-z engine-axis plane, degrees

Finally, the average jet deflection in roll δ_r, positive right wing down, is defined by the following equation:

$$\delta_r = \frac{\delta_{p_l} T_l - \delta_{p_r} T_r}{T_l + T_r} \tag{3.6}$$

The next set of equations are used for the configuration changes on the F/A-18 aircraft, that is, F/A-18, F/A-18 plus spin chute plus HARV TVCS. These increments are found directly for the longitudinal set, but the lateral directional are

$$\left.\begin{aligned} \Delta C_{Y\beta} &= \left(\Delta C_{Y\beta_{TVCS}} + \Delta C_{Y\beta_{vector}} \frac{\delta_y}{F_y} C_{T_{fac}} Geom_{fac}\right) \\ \Delta C_{l\beta} &= \left(\Delta C_{l\beta_{TVCS}} + \Delta C_{l\beta_{vector}} \frac{\delta_y}{F_y} C_{T_{fac}} Geom_{fac}\right) \\ \Delta C_{n\beta} &= \left(\Delta C_{n\beta_{TVCS}} + \Delta C_{n\beta_{vector}} \frac{\delta_y}{F_y} C_{T_{fac}} Geom_{fac}\right) \end{aligned}\right\} \tag{3.7}$$

where $\Delta C_{Y\beta}$ = increment in side force due to sideslip

$\Delta C_{Y\beta_{tvcs}}$ = increment in side force due to sideslip for tvcs

$\Delta C_{Y\beta_{vector}}$ = increment in side force due to vector

$\Delta C_{l\beta}$ = increment in roll moment due to sideslip

$\Delta C_{l\beta_{tvcs}}$ = increment in roll moment due to sideslip for tvcs

$\Delta C_{l\beta_{vector}}$ = increment in roll moment due to vector

$\Delta C_{n\beta}$ = increment in yaw moment due to sideslip

$\Delta C_{n\beta_{tvcs}}$ = increment in yaw moment due to sideslip for tvcs

$\Delta C_{n\beta_{vector}}$ = increment in yaw moment due to vector

$C_{T_{fac}}$ = thrust coefficient factor

The sideslip and vector increments in equations 3.7 are computed with table look ups in the

subroutine TLU3060 and are a function of angle of attack only.

The total vector increments are computed as follows:

$$\left.\begin{aligned}
\Delta C_{L_{vector}} &= \Delta C_{L_{\delta_H}} C_{T_{fac}} Geom_{fac} \sqrt{(\frac{\delta p}{Fp})^2 + (\frac{\delta y}{Fy})^2} \\
\Delta C_{D_{vector}} &= \Delta C_{D_{\delta_H}} C_{T_{fac}} Geom_{fac} \sqrt{(\frac{\delta p}{Fp})^2 + (\frac{\delta y}{Fy})^2} \\
\Delta C_{m_{vector}} &= \Delta C_{m_{\delta_H}} C_{T_{fac}} Geom_{fac} \sqrt{(\frac{\delta p}{Fp})^2 + (\frac{\delta y}{Fy})^2}
\end{aligned}\right\} \qquad (3.8)$$

where
$\Delta C_{L_{vector}}$ = increment in lift due to vector
$\Delta C_{L_{\delta_H}}$ = change in lift due to stabilator
$\Delta C_{D_{vector}}$ = increment in drag due to vector
$\Delta C_{D_{\delta_H}}$ = change in drag due to stabilator
$\Delta C_{m_{vector}}$ = increment in pitch moment due to vector
$\Delta C_{m_{\delta_H}}$ = change in pitch moment due to stabilator

The longitudinal coefficients $\Delta C_{L_{\delta_H}}$, $\Delta C_{D_{\delta_H}}$, and $\Delta C_{m_{\delta_H}}$ in equations 3.8 are a function of angle of attack, average horizontal tail (stabilator) deflection δ_H, and jet deflection direction. Again these coefficients are computed with table look ups in the subroutine TLU3060.

The total vector lateral/directional equations (3.9) use $\frac{\delta_y}{F_y}$ with the vectoring to the right as the positive direction. For this reason the left and right tables are identical and the sign is taken care of in δ_y. The lateral/directional coefficients $\Delta C_{Y_{vector}}$, $\Delta C_{l_{vector}}$, and $\Delta C_{n_{vector}}$ are a function of angle of attack only and computed in subroutine TLU3060. The lateral/directional vector coefficients due to rudder $\Delta C_{Y_{\delta_R}}$, $\Delta C_{l_{\delta_R}}$, and $\Delta C_{n_{\delta_R}}$ are a function of angle of attack and jet deflection direction.

$$\left.\begin{aligned}
\Delta C_{Y_{vector_{tot}}} &= (\Delta C_{Y_{vector}} \frac{\delta_y}{F_y} + \Delta C_{Y_{\delta_R}} \sqrt{(\frac{\delta p}{Fp})^2 + (\frac{\delta y}{Fy})^2}\,) C_{T_{fac}} Geom_{fac} \\
\Delta C_{l_{vector_{tot}}} &= (\Delta C_{l_{vector}} \frac{\delta_y}{F_y} + \Delta C_{l_{\delta_R}} \sqrt{(\frac{\delta p}{Fp})^2 + (\frac{\delta y}{Fy})^2}\,) C_{T_{fac}} Geom_{fac} \\
\Delta C_{n_{vector_{tot}}} &= (\Delta C_{n_{vector}} \frac{\delta_y}{F_y} + \Delta C_{n_{\delta_R}} \sqrt{(\frac{\delta p}{Fp})^2 + (\frac{\delta y}{Fy})^2}\,) C_{T_{fac}} Geom_{fac}
\end{aligned}\right\} \qquad (3.9)$$

where
$\Delta C_{Y_{vector_{tot}}}$ = total increment in side force due to vector
$\Delta C_{Y_{vector}}$ = increment in side force due to vector

$\Delta C_{Y_{\delta_R}}$ = change in side force due to rudder

$\Delta C_{l_{vector_{tot}}}$ = total increment in roll moment due to vector

$\Delta C_{l_{vector}}$ = increment in roll moment due to vector

$\Delta C_{l_{\delta_R}}$ = change in roll moment due to rudder

$\Delta C_{n_{vector_{tot}}}$ = total increment in yaw moment due to vector

$\Delta C_{n_{vector}}$ = increment in yaw moment due to vector

$\Delta C_{n_{\delta_R}}$ = change in yaw moment due to rudder

To get the total increments for thrust vectoring, equations 3.10, the F/A-18 HARV TVCS numbers must be summed with the vector increments and, in the lateral/directional cases, with the β increments. Prandtl-Glauert is also applied at this time.

$$\left.\begin{aligned}
\Delta C_{L_{HARV}} &= (\Delta C_{L_{vector}} + \Delta C_{L_{tvcs}}) / \text{P.G.} \\
\Delta C_{D_{HARV}} &= (\Delta C_{D_{vector}} + \Delta C_{D_{tvcs}}) / \text{P.G.} \\
\Delta C_{m_{HARV}} &= (\Delta C_{m_{vector}} + \Delta C_{m_{tvcs}}) / \text{P.G.} \\
\Delta C_{Y_{HARV}} &= (\Delta C_{Y_{vector_{tot}}} + \Delta C_{Y_\beta}\beta) / \text{P.G.} \\
\Delta C_{l_{HARV}} &= (\Delta C_{l_{vector_{tot}}} + \Delta C_{l_\beta}\beta) / \text{P.G.} \\
\Delta C_{n_{HARV}} &= (\Delta C_{n_{vector_{tot}}} + \Delta C_{n_\beta}\beta) / \text{P.G.}
\end{aligned}\right\} \qquad (3.10)$$

where $\Delta C_{L_{tvcs}}$ = increment in lift due to tvcs

$\Delta C_{D_{tvcs}}$ = increment in drag due to tvcs

$\Delta C_{m_{tvcs}}$ = increment in pitch moment due to tvcs

and where P.G. is the Prandtl-Glauert compressibility correction and is

$$\text{P.G.} = \sqrt{\text{abs}(1 - M^2)}$$

for Mach numbers below 0.66 and above 1.25. Between these Mach numbers P.G. is assumed to be equal to 0.75.

In all cases in the aerodynamic thrust-vectoring interaction model, the sign of the vectoring increments to the basic forces and moments is taken care of in the δ_y sign; that is, the increments are linearized with respect to yaw vectoring. This is not the case in the pitch vectoring increments where separate tables are provided for up and down vectoring. All lateral vectoring increments are tabulated for right vectoring. For left vectoring the sign of δ_y is negative compared to right vectoring and will reverse the value of the coefficients in the lateral/directional derivatives.

3.2 Introduction to Actuated Nose Strakes for Enhanced Rolling (ANSER) System and Geometry

The Actuated Nose Strakes for Enhanced Rolling (ANSER) system consists of actuated strakes located on the forebody of the F/A-18 HARV. The purpose of the strakes is to augment the level of directional control of the F/A-18 HARV at high angle of attack. The ANSER system does this by increasing suction pressure on the side of the radome opposite a deflected strake. The increased suction results in a sideforce applied near the nose of the aircraft, thereby producing a yawing moment. Because the effect is significant only for relatively large angles-of-attack, the ANSER system complements the existing directional controllers (rudders) on the F/A-18, which are most effective at relatively low angles of attack. References (3.1-3.4) provide detailed information on the development of the forebody strake concept.

Strake geometry is shown in Figure 3.1. When deflection for a strake is zero degrees, the surface conforms to the aircraft forebody and produces no effect. Full strake deployment is 90 degrees.

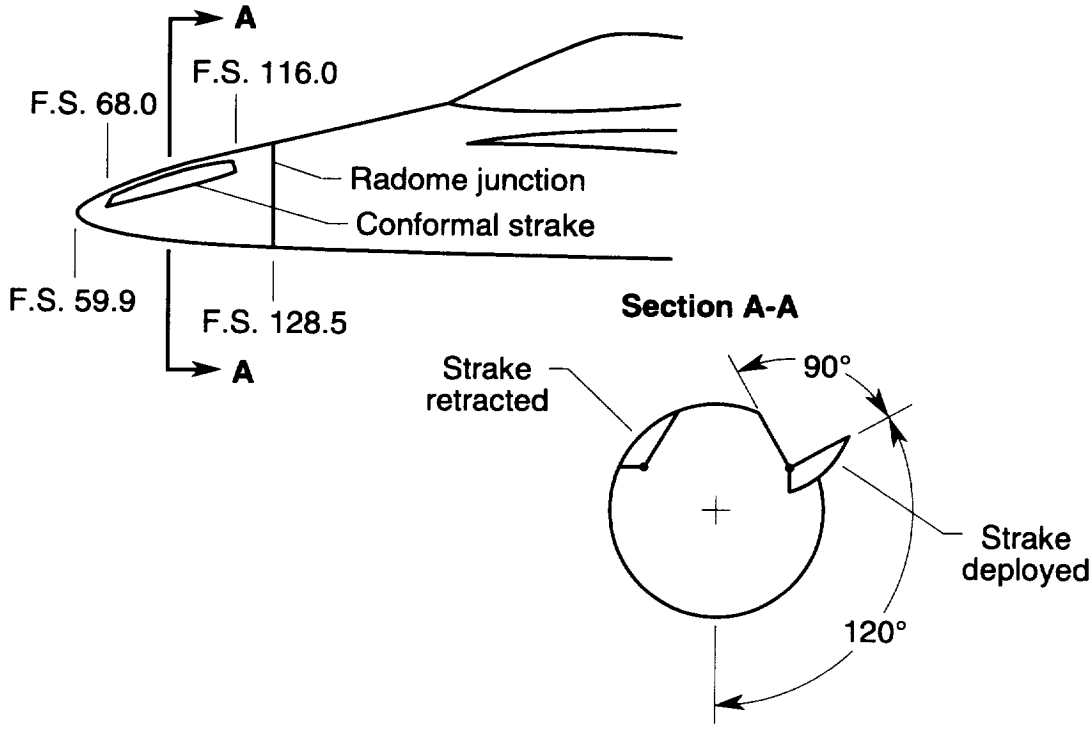


Figure 3.1. ANSER System Geometry

Tunnel data for the strakes demonstrate their utility as strong yawing-moment effectors. However an adverse yawing moment nonlinearity occurs at high angle of attack for small strake deflections resulting in the need to deflect the strakes symmetrically as angle of attack increases above thirty degrees. A symmetric strake deflection schedule was developed as a function of angle of attack, requiring the deflections shown in Figure 3.2 in the absence of directional control requirements. If directional control is desired at the higher angles of attack, differential strake deflections are commanded causing one strake to be retracted while the opposite strake is extended by an equal amount. Refer to section 3.4 on strake sign convention for more information.

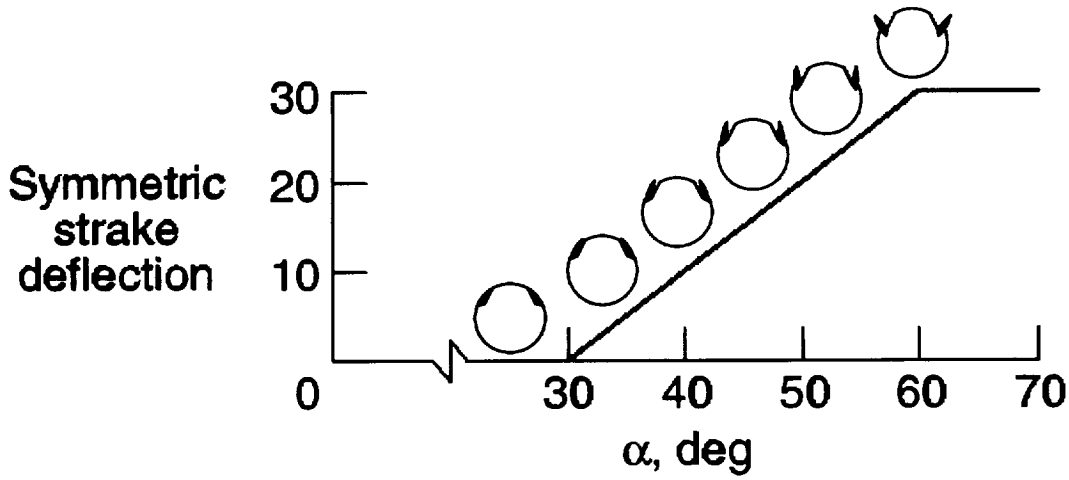


Figure 3.2. ANSER Symmetric Reference Deflection Schedule

3.3 Aerodynamic Data Sources

Following the convention of reference 1.0, the aerodynamic model for ANSER was developed as increments to the existing aerodynamic force and moment buildup equations.

The ANSER aerodynamic model was constructed from data obtained through an extensive series of wind-tunnel experiments . These experiments were conducted at the NASA Langley (LaRC) and Ames Research Center (ARC) and the David Taylor Research Center (DTRC). The three major portions of the ANSER model were tested with the following tunnel experiments.

3.3.1 Static testing

a) 6% F/A-18 in the 7 by 10-Foot Transonic Tunnel (DTRC), Nov. 89.
b) 16% F/A-18 in the 30 by 60-Foot Tunnel (LaRC), July 90.
c) Full-Scale F/A-18 in the 80 by 120-Foot Tunnel (ARC), Aug. 91.
d) Full-Scale F/A-18 Forebody in the 30 by 60-Foot Tunnel (LaRC), Sept.91.
e) 16% F/A-18 in the 30 by 60-Foot Tunnel (LaRC), Nov. 91.

3.3.2 Dynamic testing

a) 16% F/A-18 in the 30 by 60-Foot Tunnel (LaRC) with the forced oscillation rig, Nov. 90.
b) 10% F/A-18 in the 20-Foot Vertical Spin Tunnel (LaRC) with the rotary balance rig, Mar. 91.

3.4 ANSER Surface Sign Convention

The strakes follow the standard sign convention dictating that a positive differential deflection imparts a negative resulting moment to the aircraft. For this to be true, the maximum positive strake deflection equates to, for individual surface deflections, right strake at its maximum (ninety degrees) and left strake at its minimum (zero degrees). The maximum negative strake deflection gives left strake at maximum (ninety degrees) and right strake at minimum (zero degrees). The differential strake deflection is defined as

$$\delta_{S_{diff}} = \delta_{S_r} - \delta_{S_l} \tag{3.11}$$

Additionally, the symmetric strake deflection was defined as

$$\delta_{S_{symm}} = \text{MIN}(\delta_{S_l}, \delta_{S_r}) \tag{3.12}$$

where the output of the MIN function is the lesser of its two arguments. Since both left and right strake deflections are limited to positive values between zero and ninety degrees, the symmetric strake deflection will never be a negative value. This definition of symmetric deflection simplifies the strake configuration by identifying the deflection value of at least one of the surfaces. When used in conjunction with the differential deflection value, the deflections of both surfaces can be obtained.

Any control law that includes the ANSER system must contain symmetric/differential manipulations in order to command desired deflections through the left and right strake channels. However, the limitations of the strake data in this aerodynamic model require that the symmetric schedule outlined above be followed. This requirement implies that angle-of-attack feedback into the control system is necessary to keep the strakes on the symmetric schedule. The following logic is incorporated into any ANSER control system:

a) Use current angle of attack to determine the symmetric strake reference deflection, $\delta_{S_{symm,ref}}$

b) Use current yaw control requirements to define the desired strake differential deflection, $\delta_{S_{diff}}$

c)

If [abs($\delta_{S_{diff}}$) is less than 2.0*$\delta_{S_{symm,ref}}$] then

$\delta_{S_l} = \delta_{S_{symm,ref}} - \delta_{S_{diff}}/2.0$

$\delta_{S_r} = \delta_{S_{symm,ref}} + \delta_{S_{diff}}/2.0$

Else if [$\delta_{S_{diff}}$ is greater than zero] then

$\delta_{S_r} = \delta_{S_{diff}}$

$\delta_{S_l} = 0.0$

Else

$\delta_{S_l} = -\delta_{S_{diff}}$

$\delta_{S_r} = 0.0$

Endif

The symmetric strake deflection can then be calculated from (3.12) above. Symmetric deflection is equal to the reference symmetric deflection only when differential deflection is zero. If differential deflection is nonzero, symmetric deflection must be determined from the above logic. The aerodynamic model expects differential and symmetric deflection values as inputs.

3.5 ANSER Aerodynamic Coefficient Buildup

The aerodynamic coefficient buildup equations are constructed in a manner similar to the "reconstruction equations" of reference 1.0. The strake increments are modeled as additions to the steady flow increments from that document. The actual equation of interest can be constructed by replacing "X" with "L" ,"D", "m", "Y", "l", or "n" as appropriate in the equation that follows.

$$C_{X_{sf}} = C_{X_{sf_o}} + \Delta C_{X_{strake}} \tag{3.13}$$

$C_{Xsf,o}$ represents the combination of all existing steady aerodynamic increments for that axis. The new steady increment is then equal to the old steady increment added to the ANSER system increment. The ANSER, or strake, increment is formed from static and dynamic segments:

$$\Delta C_{X_{strake}} = \Delta C_{X_{static}} + \Delta C_{X_{dynamic}} \tag{3.14}$$

The difference between axes manifests itself in the details of the static and dynamic calculations. See the sections below for more information on each axis.

3.6 Lift and Drag

Since lift and drag increments due to the ANSER system are calculated in an identical fashion, they are combined here for brevity. Only in the specifics of the dynamic forced oscillation calculations do the buildup for these axes differ.

3.6.1 Strake static coefficient buildup (L,D)

The static portion of lift and drag is equal to the low-speed static increment plus the high-speed contribution. The low-speed static increment is constructed by combining symmetric and differential strake effects. The end result is equal to the symmetric strake increment at small absolute values of differential strake deflection (0 degrees) and fades into the differential strake increment at larger values (10 degrees and greater). The high-speed static increment varies linearly with differential strake deflection as described in the following equations.

$$\Delta C_{X_{static}} = \Delta C_{X_{static_{symm}}} + (\Delta C_{X_{static_{diff}}} - \Delta C_{X_{static_{symm}}}) \cdot k_1 + k_2 \cdot \Delta C_{X_{static_{high-speed}}} \tag{3.15}$$

where

$$k_1 = \left[\frac{ABS(\delta_{S_{diff}})}{10.0}\right]_0^{1.0} \text{ and } k_2 = \left[\frac{ABS(\delta_{S_{diff}})}{90.0}\right]_0^{1.0} \tag{3.16}$$

The 'k' factors above are used to linearly extend the data into regions where insufficient data exists from wind tunnel experiments. The other coefficients are:

$$\left.\begin{aligned}
\Delta C_{X_{static_{symm}}} &= \left[F_{CXSSS}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{symm}}\left[{}^{30}_{0}\right)\right)\right]\right] \\
\Delta C_{X_{static_{diff}}} &= \left[F_{CXSSD}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{diff}}\left[{}^{-10}_{-90}\right)\right)\right]\right] \\
\Delta C_{X_{static_{high-speed}}} &= \left[F_{CXSFH}\left(\alpha\left[{}^{55}_{10}, M\left[{}^{0.8}_{0.2}\right)\right)\right]
\end{aligned}\right\} \tag{3.17}$$

The notation above is defined as follows.

$$F_{CXSSS}(\beta[{}^{30}_{-30}, \alpha[{}^{70}_{10}, \delta_{S_{symm}}[{}^{30}_{0}) \equiv \left\{\begin{array}{l}\text{the function } F_{SSS} \text{ as defined by data tables} \\ \text{evaluated at an angle - of - sideslip,} \\ \text{angle - of - attack, and symmetric strake} \\ \text{deflection that are constrained according to} \\ \text{the bracketed values}\end{array}\right\} \tag{3.18}$$

3.6.2 Strake dynamic coefficient buildup (L,D)

The segment of the ANSER aerodynamic buildup that takes into account the effect of dynamic motions of the aircraft is formed from two sources: forced-oscillation data and rotary-balance data. Forced-oscillation data is supplied in the form of an increment to the body-axis coefficient derivative with respect to the non-dimensional rotation rate for that axis. Rotary-balance data is supplied in the form of force and moment coefficient increments that are a direct function of non-dimensional rotation rate.

$$\Delta C_{X_{dynamic}} = \Delta C_{X_{forced\ oscillation}} + \Delta C_{X_{rotary\ balance}} \tag{3.19}$$

The forced-oscillation data was collected in terms of axial and normal force coefficient derivatives and must be resolved into components of lift and drag, as described below.

$$\left.\begin{aligned}
\Delta C_{L_{forced\ oscillation}} &= \left(\Delta C_{N_{q_{symm}}} + \Delta C_{N_{q_{diff}}}\right) \cdot \frac{q\bar{c}}{2V} \cdot \cos(\alpha) \\
&\quad - \left(\Delta C_{A_{q_{symm}}} + \Delta C_{A_{q_{diff}}}\right) \cdot \frac{q\bar{c}}{2V} \cdot \sin(\alpha) \\
\Delta C_{D_{forced\ oscillation}} &= \left(\Delta C_{N_{q_{symm}}} + \Delta C_{N_{q_{diff}}}\right) \cdot \frac{q\bar{c}}{2V} \cdot \sin(\alpha) \\
&\quad + \left(\Delta C_{A_{q_{symm}}} + \Delta C_{A_{q_{diff}}}\right) \cdot \frac{q\bar{c}}{2V} \cdot \cos(\alpha)
\end{aligned}\right\} \tag{3.20}$$

where $\frac{q\bar{c}}{2V}$ is the non-dimensionalized pitch-axis rotation rate. The non-dimensional coefficient derivatives are:

$$\left.\begin{aligned}
\Delta C_{N_{q_{symm}}} &= \left[F_{CNQFSS}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{symm}}\left[{}^{45}_{0}\right)\right)\right] \right. \\
\Delta C_{N_{q_{diff}}} &= \left[F_{CNQFSD}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{diff}}\left[{}^{90}_{0}\right)\right)\right] \right. \\
\Delta C_{A_{q_{symm}}} &= \left[F_{CAQFSS}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{symm}}\left[{}^{45}_{0}\right)\right)\right] \right. \\
\Delta C_{A_{q_{diff}}} &= \left[F_{CAQFSD}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{diff}}\left[{}^{90}_{0}\right)\right)\right] \right.
\end{aligned}\right\} \qquad (3.21)$$

Rotary-balance data was available for only three ANSER configurations: (i) left strake fully extended/right retracted, (ii) left retracted/right fully extended, and (iii) both strakes retracted. Due to this limitation, the effects of symmetric strake deflections were not modeled. This constraint is not likely to be a practical limitation, due to the method of blending the two types of dynamic data.

The rotary-balance data represents a steady-state rotation condition and its contribution to the aerodynamic buildup should not be used while aircraft angular rates are not fully established. Until the angular rates are fully established, forced oscillation data is used to provide the dynamic contribution. The transition from forced oscillation to rotary-balance data is implemented as a time delay of the wind-axis roll rate, described below.

$$\begin{aligned}
&\frac{\Omega_{SS}(s)}{p_{wind}(s)} = \frac{1}{\tau s + 1} \\
&p_{wind} = p \cdot \cos(\alpha) \cdot \cos(\beta) + q \cdot \sin(\beta) + r \cdot \sin(\alpha) \cdot \cos(\beta) \\
&\tau = \{\text{time constant governing onset of steady rotation}\} \\
&k = \frac{\Omega_{SS} \cdot b}{2V} = \{\text{non - dimensional rotation rate input to rotary balance data tables}\}
\end{aligned} \qquad (3.22)$$

The 'k' parameter is one of the independent variable inputs to the rotary-balance data tables. In addition, the angular rates used to calculate the forced oscillation increments are redefined to be the residual oscillatory rotation rates about the steady-state rotation rate. The residual rotation rates are defined below.

$$\begin{aligned}
p[\text{forced osc.}] &= p_{osc} = p - \Omega_{ss} \cdot \cos(\beta) \cdot \cos(\alpha) \\
q[\text{forced osc.}] &= q_{osc} = q - \Omega_{ss} \cdot \sin(\beta) \\
r[\text{forced osc.}] &= r_{osc} = r - \Omega_{ss} \cdot \cos(\beta) \cdot \sin(\alpha)
\end{aligned} \qquad (3.23)$$

where p, q, r are the body-axis roll, pitch, and yaw rotation rates of the aircraft, respectively. By defining the angular rates for the forced oscillation calculations in terms of the steady state rotation rate, the delay simultaneously fades in the rotary-balance increments and fades out the forced oscillation increments.

Referring to the lack of rotary-balance data at symmetric strake deflections, it becomes apparent that this data is unnecessary. For a steady rotation condition, the strakes will most likely be commanded to a fully asymmetric configuration. Whether the strakes are commanded in an attempt to induce or arrest the steady rotation, the strakes will likely be at a differential extreme at any angle of attack. Therefore the symmetric contribution will be minimal for these

conditions.

The rotary-balance increments and their independent parameters are:

$$\Delta C_{X_{\text{rotary balance}}} = \left\{ F_{CXFSR}\left(k\left[{}^{0.2}_{-0.2}, \alpha\left[{}^{90}_{0}, \beta\left[{}^{10}_{-10}, \delta_{S_{\text{diff}}}\left[{}^{90}_{-90} \right) \right\} \right. \right. \right. \tag{3.24}$$

3.7 Pitch (m)

The ANSER pitch-axis static parameters found from data tables are:

$$\left.\begin{aligned} \Delta C_{m_{\text{static}_{\text{symm}}}} &= \left[F_{CMSSS}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{\text{symm}}}\left[{}^{30}_{0} \right) \right] \right] \right. \right. \right. \\ \Delta C_{m_{\text{static}_{\text{diff}}}} &= \left[F_{CMSSD}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{\text{diff}}}\left[{}^{-10}_{-90} \right) \right] \right] \right. \right. \right. \\ \Delta C_{m_{\text{static}_{\text{high-speed}}}} &= \left[F_{CMSFH}\left(\alpha\left[{}^{55}_{10}, M\left[{}^{0.8}_{0.2} \right) \right] \right. \right. \end{aligned}\right\} \tag{3.25}$$

which are combined in a different manner than the lift/drag example. The high speed dependent parameter acts as a multiplicative factor instead of an additive increment, demonstrated below:

$$\Delta C_{m_{\text{static}}} = \Delta C_{m_{\text{static}_{\text{symm}}}} + \left(\Delta C_{m_{\text{static}_{\text{diff}}}} - \Delta C_{m_{\text{static}_{\text{symm}}}} \right) \cdot k_1 \cdot \Delta C_{m_{\text{static}_{\text{high-speed}}}} \tag{3.26}$$

where k_1 is defined in equation (3.16) above.

The forced oscillation coefficient derivatives for pitch are:

$$\left.\begin{aligned} \Delta C_{m_{q_{\text{symm}}}} &= \left[F_{CMQFSS}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{\text{symm}}}\left[{}^{45}_{0} \right) \right] \right] \right. \\ \Delta C_{m_{q_{\text{diff}}}} &= \left[F_{CMQFSD}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{\text{diff}}}\left[{}^{90}_{0} \right) \right] \right. \end{aligned}\right\} \tag{3.27}$$

The rotary-balance increments in pitch are:

$$\Delta C_{m_{\text{rotary balance}}} = \left\{ F_{CMFSR}\left(k\left[{}^{0.2}_{-0.2}, \alpha\left[{}^{90}_{0}, \beta\left[{}^{10}_{-10}, \delta_{S_{\text{diff}}}\left[{}^{90}_{-90} \right) \right\} \right. \right. \right. \tag{3.28}$$

The dynamic parameters from the data tables are combined to form the dynamic pitch increment:

$$\Delta C_{m_{\text{dynamic}}} = \left(\Delta C_{m_{q_{\text{symm}}}} + \Delta C_{m_{q_{\text{diff}}}} \right) \cdot \frac{q\bar{c}}{2V} + \Delta C_{m_{\text{rotary balance}}} \tag{3.29}$$

The ANSER pitch increment is then the combination of static and dynamic increments.

3.8 Sideforce (Y)

The static parameters for sideforce are:

$$\left.\begin{aligned}
\Delta C_{Y_{static_{symm}}} &= \left[F_{CYSSS}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{symm}}\left[{}^{30}_{0}\right)\right)\right]\right] \right. \\
\Delta C_{Y_{static_{diff}}} &= \left[F_{CYSSD}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{diff}}\left[{}^{-10}_{-90}\right)\right)\right]\right] \\
\left. \Delta C_{Y_{static_{high-speed}}} &= \left[F_{CYSFH}\left(\alpha\left[{}^{55}_{10}, M\left[{}^{0.8}_{0.2}\right)\right)\right]\right]
\end{aligned}\right\} \tag{3.30}$$

The high-speed static contribution to sideforce due to the strakes is implemented as an increment, much like lift and drag, but in this case both differential strake-deflection sign and magnitude are necessary to extend the available data for intermediate strake-deflection values. The effect of Mach number on aerodynamic forces is modeled as an increment, but the effect on aerodynamic moments is modeled as a multiplicative factor. See the roll and yaw section for further information. The static ANSER increment is calculated as:

$$\left.\begin{aligned}
\Delta C_{Y_{static}} &= \Delta C_{Y_{static_{symm}}} + \left(\Delta C_{Y_{static_{diff}}} - \Delta C_{Y_{static_{symm}}}\right) \cdot k_1 \\
&\quad + k_3 \cdot \Delta C_{Y_{static_{high-speed}}} \\
k_3 &= \left[\frac{\delta_{S_{diff}}}{-90.0}\right]_{-1.0}^{1.0}
\end{aligned}\right\} \tag{3.31}$$

The forced-oscillation calculations for sideforce require data from both lateral/directional tests (roll and yaw) on the forced oscillation rig. The sideforce dynamic increment is modeled as a function of both body-axis roll and yaw rates.

$$\left.\begin{aligned}
\Delta C_{Y_{r_{symm}}} &= \left[F_{CYRFSS}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{symm}}\left[{}^{45}_{0}\right)\right)\right]\right] \\
\Delta C_{Y_{r_{diff}}} &= \left[F_{CYRFSD}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{diff}}\left[{}^{90}_{0}\right)\right)\right] \\
\Delta C_{Y_{p_{symm}}} &= \left[F_{CYPFSS}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{symm}}\left[{}^{45}_{0}\right)\right)\right]\right] \\
\Delta C_{Y_{p_{diff}}} &= \left[F_{CYPFSD}\left(\alpha\left[{}^{80}_{0}, \delta_{S_{diff}}\left[{}^{90}_{0}\right)\right)\right]\right]
\end{aligned}\right\} \tag{3.32}$$

The rotary-balance increment for sideforce is given as

$$\Delta C_{Y_{\text{rotary balance}}} = \left\{ F_{CYFSR}\left(k\left[{}^{0.2}_{-0.2}, \alpha\left[{}^{90}_{0}, \beta\left[{}^{10}_{-10}, \delta_{S_{diff}}\left[{}^{90}_{-90} \right) \right) \right) \right\} \tag{3.33}$$

The sideforce dynamic increment is then given below.

$$\begin{aligned} \Delta C_{Y_{dynamic}} = & \left(\Delta C_{Y_{r_{symm}}} + \Delta C_{Y_{r_{diff}}} \right) \cdot \frac{rb}{2V} \\ & + \left(\Delta C_{Y_{p_{symm}}} + \Delta C_{Y_{p_{diff}}} \right) \cdot \frac{pb}{2V} + \Delta C_{Y_{\text{rotary balance}}} \end{aligned} \tag{3.34}$$

3.9 Roll and Yaw (n,l)

The static-data tables for roll and yaw are given as:

$$\left.\begin{aligned} \Delta C_{l_{static_{symm}}} &= \left[F_{CRMSSS}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{symm}}\left[{}^{30}_{0} \right) \right) \right) \right] \\ \Delta C_{l_{static_{diff}}} &= \left[F_{CRMSSD}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{diff}}\left[{}^{-10}_{-90} \right) \right) \right) \right] \\ \Delta C_{l_{static_{high-speed}}} &= \left[F_{CLLSFH}\left(\alpha\left[{}^{55}_{10}, M\left[{}^{0.8}_{0.2} \right) \right) \right] \end{aligned}\right\} \tag{3.35}$$

$$\left.\begin{aligned} \Delta C_{n_{static_{symm}}} &= \left[F_{CNSSS}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{symm}}\left[{}^{30}_{0} \right) \right) \right) \right] \\ \Delta C_{n_{static_{diff}}} &= \left[F_{CNSSD}\left(\beta\left[{}^{30}_{-30}, \alpha\left[{}^{70}_{10}, \delta_{S_{diff}}\left[{}^{-10}_{-90} \right) \right) \right) \right] \\ \Delta C_{n_{static_{high-speed}}} &= \left[F_{CNSFH}\left(\alpha\left[{}^{55}_{10}, M\left[{}^{0.8}_{0.2} \right) \right) \right] \end{aligned}\right\} \tag{3.36}$$

Here, the Mach effects are modeled as multiplicative factors, much like the pitch axis.

$$\left.\begin{aligned} \Delta C_{l_{static}} &= \Delta C_{l_{static_{symm}}} + \left(\Delta C_{l_{static_{diff}}} - \Delta C_{l_{static_{symm}}} \right) \cdot k_1 \cdot \Delta C_{l_{static_{high-speed}}} \\ \Delta C_{n_{static}} &= \Delta C_{n_{static_{symm}}} + \left(\Delta C_{n_{static_{diff}}} - \Delta C_{n_{static_{symm}}} \right) \cdot k_1 \cdot \Delta C_{n_{static_{high-speed}}} \end{aligned}\right\} \tag{3.37}$$

The dynamic increments are once again formed from an expression involving forced-oscillation and rotary-balance increments. The naming convention for roll and yaw forced-oscillation coefficient derivatives is similar enough to use a generic notation below. The coefficient/function of interest can be found by replacing the "x" below with "l" or "n" for roll or yaw, respectively.

$$\left.\begin{aligned}
\Delta C_{x_{r_{symm}}} &= \left[F_{CXRFSS}\left(\alpha\left[{}^{80}_{0}, \delta_{symm}\left[{}^{45}_{0}\right)\right)\right]\right. \\
\Delta C_{x_{r_{diff}}} &= \left[F_{CXRFSD}\left(\alpha\left[{}^{80}_{0}, \delta_{diff}\left[{}^{90}_{0}\right)\right)\right] \right. \\
\Delta C_{x_{p_{symm}}} &= \left[F_{CXPFSS}\left(\alpha\left[{}^{80}_{0}, \delta_{symm}\left[{}^{45}_{0}\right)\right)\right]\right. \\
\Delta C_{x_{p_{diff}}} &= \left[F_{CXPFSD}\left(\alpha\left[{}^{80}_{0}, \delta_{diff}\left[{}^{90}_{0}\right)\right)\right] \right.
\end{aligned}\right\} \tag{3.38}$$

The rotary-balance increments for roll and yaw are:

$$\left.\begin{aligned}
\Delta C_{l_{rotary\,balance}} &= \left[F_{DCRMFSR}\left(k\left[{}^{0.2}_{-0.2}, \alpha\left[{}^{90}_{0}, \beta\left[{}^{10}_{-10}, \delta_{diff}\left[{}^{90}_{-90}\right)\right)\right)\right]\right] \\
\Delta C_{n_{rotary\,balance}} &= \left[F_{DCNFSR}\left(k\left[{}^{0.2}_{-0.2}, \alpha\left[{}^{90}_{0}, \beta\left[{}^{10}_{-10}, \delta_{diff}\left[{}^{90}_{-90}\right)\right)\right)\right]\right]
\end{aligned}\right\} \tag{3.39}$$

The dynamic contributions to roll and yaw are:

$$\left.\begin{aligned}
\Delta C_{l_{dynamic}} &= \left(\Delta C_{l_{r_{symm}}} + \Delta C_{l_{r_{diff}}}\right)\cdot\frac{rb}{2V} \\
&\quad + \left(\Delta C_{l_{p_{symm}}} + \Delta C_{l_{p_{diff}}}\right)\cdot\frac{pb}{2V} + \Delta C_{l_{rotary\,balance}} \\
\Delta C_{n_{dynamic}} &= \left(\Delta C_{n_{r_{symm}}} + \Delta C_{n_{r_{diff}}}\right)\cdot\frac{rb}{2V} \\
&\quad + \left(\Delta C_{n_{p_{symm}}} + \Delta C_{n_{p_{diff}}}\right)\cdot\frac{pb}{2V} + \Delta C_{n_{rotary\,balance}}
\end{aligned}\right\} \tag{3.40}$$

3.10 Force and Moment Calculations

The calculation of the total aerodynamic forces and moments is accomplished in the same manner as outlined in reference 1.0. After combining ANSER static and dynamic segments in equation 3.14, the strake increments simply augment the steady-flow coefficient increments for the aircraft. The HARV increments in equations 3.10 are summed with the strake increments and basic F/A-18 increments in subroutine AEROINC. Calculations related to the equations of motion of the aircraft then proceed as in Section 5.6 of Reference 1.0, which documents the basic F/A-18 aero increments in sections 5.1 - 5.5. The following equations list the total aerodynamic increments and variable names used in subroutine AEROINC.

$$C_{D_{total}} = C_{D_{total}} + \Delta C_{D_{HARV}} + \Delta C_{D_{strake}}$$

CDRFSF = CDRFSF + DELCD + DCDFS

$$C_{L_{total}} = C_{L_{total}} + \Delta C_{L_{HARV}} + \Delta C_{L_{strake}}$$

CLRFSF = CLRFSF + DELCLF + DCLFS

$$C_{m_{total}} = C_{m_{total}} + \Delta C_{m_{HARV}} + \Delta C_{m_{strake}} \tag{3.41}$$

CMRFSF = CMRFSF + DELCM + DCMFS

$$C_{Y_{total}} = C_{Y_{total}} + \Delta C_{Y_{HARV}} + \Delta C_{Y_{strake}}$$

CYRFSF = CYRFSF + DELCY + DCYFS

$$C_{l_{total}} = C_{l_{total}} + \Delta C_{l_{HARV}} + \Delta C_{l_{strake}}$$

CIRFSF = CIRFSF + DELCL + DCRMFS

$$C_{n_{total}} = C_{n_{total}} + \Delta C_{n_{HARV}} + \Delta C_{n_{strake}}$$

CNRFSF = CNRFSF + DELCN + DCNFS

3.11 Implementation

The subroutine SFAERRF produces the total force and moment coefficients for the basic F/A-18 configuration about the aerodynamic reference center for steady-flow conditions (constant angle of attack and sideslip). Tables 3.1 and 3.2 lists the names of each input/output parameter as they appear in both the subroutine SFAERRF and in the *f18harv.csl* code from which SFAERRF is called. The density altitude used in *f18bas* as an input to the flex/rigid multipliers and increments has been changed to altitude to match the implementations in the DMS and Dryden batch simulation. Also, the left rudder deflection had a sign change to match the HARV control-deflection sign convention.

Table 3.1. Input parameters for subroutine SFAERRF

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
α	ALDGRF	ALPDEG	CSL	angle of attack, degrees
β	BEDGRF	BETDEG	CSL	sideslip angle, degrees
M	MACHRF	MACH	CSL	Mach number, n.d.
p	P	P	CSL	Body-frame roll rate, rad/sec
q	Q	Q	CSL	Body-frame pitch rate, rad/sec
r	R	R	CSL	Body-frame yaw rate, rad/sec
H	H	ALT	CSL	Altitude, feet
H_{gcl}	HGC	HGCL	CSL	Ground clearance, feet
$\bar{q}$	QBAR	QBAR	CSL	Dynamic pressure, lbs/ft^2

Table 3.1. Concluded

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
– c	CWRF	CREF	CSL	aerodynamic reference wing chord, feet
b	BWRF	BREF	CSL	aerodynamic reference wing span, feet
V	VTRF	VTRF	CSL	Total airspeed at the aero reference center, ft/sec
δ_{H_r}	DSR	DHTR	ACTDYN	Right stabilator (horizontal tail), positive trailing edge down (+ t.e.d.), degrees
δ_{H_l}	DSL	DHTL	ACTDYN	Left stabilator (horizontal tail), + t.e.d., degrees
δ_{A_r}	DAR	DAR	ACTDYN	Right aileron, + t.e.d., degrees
δ_{A_l}	DAL	DAL	ACTDYN	Left aileron, + t.e.d., degrees
δ_{R_r}	DRR	DRR	ACTDYN	Right rudder, + t.e. left, degrees
δ_{R_l}	DRL	DRL	ACTDYN	Left rudder, + t.e. left, degrees
δ_{f_r}	DFR	DTFR	ACTDYN	Right trailing-edge flap, + t.e.d., degrees
δ_{f_l}	DFL	DTFL	ACTDYN	Left trailing-edge flap, + t.e.d., degrees
δ_{n_r}	DNR	DLFR	ACTDYN	Right leading-edge flap, + l.e.d., degrees
δ_{n_l}	DNL	DLFL	ACTDYN	Left leading-edge flap, + l.e.d., degrees
δ_{SB}	DSB	DSBK	ACTDYN	Speed brakes, 0 - 60, degrees
δ_{LG}	DLG	DLG	CSL	Normalized landing gear, 0 - 1, full down
logical (.T.)	LDEBUG	LDEBUG	CSL	Detailed print of intermediate variables on unit=20.
(.F.)				No action
logical (.T.)	LQSE	LQSE	CSL	Flex/rigid multipliers and increments calculated
(.F.)				Aero for rigid airplane, flex/rigid -> 1.
logical (.T.)	LRTE	LRTE	CSL	Real time equivalent aero
(.F.)				Enhanced aerodynamic model.

Table 3.2. Output parameters for subroutine SFAERRF

Symbol	*f18harv* Name	Local Name	Definition
$C_{D_{sf}}$	CDRFSF	CDREFO	Total drag coefficient at a.r.c. evaluated at (= 0)
$C_{L_{sf}}$	CLRFSF	CLREFO	Total lift coefficient at a.r.c. evaluated at (= 0)
$C_{m_{sf}}$	CMRFSF	CMREFO	Total pitch coefficient at a.r.c. evaluated at (= 0)
$C_{Y_{sf}}$	CYRFSF	CYREFO	Total side-force coefficient at a.r.c. evaluated at (= 0)
$C_{l_{sf}}$	C1RFSF	C1REFO	Total roll coefficient at a.r.c. evaluated at (= 0)
$C_{n_{sf}}$	CNRFSF	CNREFO	Total yaw coefficient at a.r.c. evaluated at (= 0)

The input parameters to subroutine AEROINC are listed in table 3.3.

Table 3.3. Input parameters for subroutine AEROINC

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
α	ALDGRF	ADEG	CSL	angle of attack, degrees
β	BEDGRF	BDEG	CSL	sideslip angle, degrees
M	MACHRF	AMCH	CSL	Mach number, n.d.
$\bar{q}$	QBAR	QBAR	CSL	Dynamic pressure, lbs/ft^2
S	SWRF	S	CSL	aerodynamic reference wing area, feet^2
T_l T_r	FG(2)	FGAERO(2)	TLUHRV	gross thrust for left & right engine including effect of T56 bias only, lbs
δ_{p_l} δ_{p_r}	TVJANP(2)	TVJANP(2)	TVANGL	thrust-vectored pitch jet-turning-angle for the left & right engine in the x-y engine axis plane, degrees
δ_{y_l} δ_{y_r}	TVJANY(2)	TVJANY(2)	TVANGL	thrust-vectored yaw jet-turning-angle for the left & right engine in the x-z engine axis plane, degrees
δ_{H_l}	DSL	DSL	ACTDYN	Left stabilator (horizontal tail), + t.e.d., degrees
δ_{H_r}	DSR	DSR	ACTDYN	Right stabilator (horizontal tail), + t.e.d., degrees
δ_{R_r}	DRR	DRR	ACTDYN	Right rudder, + t.e. left, degrees
δ_{R_l}	DRL	DRL	ACTDYN	Left rudder, + t.e. left, degrees
p	P	P	CSL	Body-frame roll rate, rad/sec
q	Q	Q	CSL	Body-frame pitch rate, rad/sec
r	R	R	CSL	Body-frame yaw rate, rad/sec
$\bar{c}$	CWRF	CWRF	CSL	aerodynamic reference wing chord, feet
b	BWRF	BWRF	CSL	aerodynamic reference wing span, feet
V	VTRF	VTRF	CSL	Total airspeed at the aero reference center, ft/sec
-	XSTRAKE	XSTRAKE	CSL	switch to activate nose strake control law
-	DG2RA	DG2RA	CSL	degrees to radians conversion
δ_{S_l}	DNSL	DNSL	CSL	left strake deflection, degrees
δ_{S_r}	DNSR	DNSR	CSL	right strake deflection, degrees

Table 3.4 contains the outputs from subroutine AEROINC which represent the total force and moment aerodynamic increments.

Table 3.4. Output parameters for subroutine AEROINC

Symbol	*f18harv* Name	Local Name	Definition
$C_{D_{sf}}$	CDRFSF	CDRFSF	Total drag coefficient at a.r.c. evaluated at (= 0)
$C_{L_{sf}}$	CLRFSF	CLRFSF	Total lift coefficient at a.r.c. evaluated at (= 0)
$C_{m_{sf}}$	CMRFSF	CMRFSF	Total pitch coefficient at a.r.c. evaluated at (= 0)

Table 3.4 Concluded

Symbol	*f18harv* Name	Local Name	Definition
$C_{Y_{sf}}$	CYRFSF	CYRFSF	Total side-force coefficient at a.r.c. evaluated at (= 0)
$C_{l_{sf}}$	C1RFSF	C1RFSF	Total roll coefficient at a.r.c. evaluated at (= 0)
$C_{n_{sf}}$	CNRFSF	CNRFSF	Total yaw coefficient at a.r.c. evaluated at (= 0)
δ_p	TVJPAVG	TVJPAVG	average of TVJANP(1) & TVJANP(2), degrees
δ_y	TVJYAVG	TVJYAVG	average of TVJANY(1) & TVJANY(2), degrees
δ_r	TVJRAVG	TVJRAVG	difference of TVJANP(1) & TVJANP(2), degrees

The HARV increments are computed in subroutine TVCOEF which calls two other subroutines. The first, IN3060, contains the aerodynamic data in data tables, and the second, TLU3060, does the interpolation in the tables. After the calculations in TLU3060, TVCOEF builds the HARV-specific increments which are then passed to subroutine AEROINC through common blocks. The HARV increments are added to the basic F/A-18 aerodynamic force and moment coefficients for use in the equations of motion in *f18harv.csl*. The input parameters to subroutine TVCOEF are listed in table 3.5 below.

Table 3.5. Input parameters for subroutine TVCOEF

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
α	ALDGRF	ADEG	CSL	angle of attack, degrees
M	MACHRF	AMCH	CSL	Mach number, n.d.
β	BEDGRF	BDEG	CSL	sideslip angle, degrees
δ_H	-	DHTD	AEROINC	average stabilator deflection, degrees
δ_R	-	DRO2F	AEROINC	average rudder deflection, degrees
T_l T_r	FG(2)	FG(2)	TLUHRV	gross thrust for left & right engine including effect of T56 bias only, lbs
$\bar{q}$	QBAR	QBAR	CSL	Dynamic pressure, lbs/ft^2
δ_{p_l} δ_{p_r}	TVJANP(2)	TVJANP(2)	TVANGL	thrust-vectored pitch-jet-turning angle for the left & right engine in the x-y engine axis plane, degrees
δ_{y_l} δ_{y_r}	TVJANY(2)	TVJANY(2)	TVANGL	thrust-vectored yaw-jet-turning angle for the left & right engine in the x-z engine axis plane, degrees
δ_p	TVJPAVG	TVJPAVG	AEROINC	average of TVJANP(1) & TVJANP(2), degrees
δ_y	TVJYAVG	TVJYAVG	AEROINC	average of TVJANY(1) & TVJANY(2), degrees
δ_r	TVJRAVG	TVJRAVG	AEROINC	difference of TVJANR(1) & TVJANR(2), degrees

Outputs from subroutine TVCOEF are passed to AEROINC thru a common block called OUTPUTS.

Input parameters to subroutine TLU3060 are listed in table 3.6.

Table 3.6. Input parameters for subroutine TLU3060

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
α	ALDGRF	ADEG	CSL	angle of attack, degrees
δ_H	-	DHTD	AEROINC	average stabilator deflection, degrees
T_l	FG(2)	FG(2)	TLUHRV	gross thrust for left & right engine
T_r				engine including effect of T56 bias only, lbs
$\bar{q}$	QBAR	QBAR	CSL	Dynamic pressure, lbs/ft^2
S	S	S	CSL	aerodynamic reference wing area, feet^2
δ_{p_l}	TVJANP(2)	TVJANP(2)	TVANGL	thrust-vectored pitch-jet-turning angle
δ_{p_r}				for the left & right engine in the x-y engine axis plane, degrees
δ_{y_l}	TVJANY(2)	TVJANY(2)	TVANGL	thrust-vectored yaw-jet-turning angle for
δ_{y_r}				the left & right engine in the x-z engine axis plane, degrees

The aerodynamic force and moment calculations for the ANSER system are performed in a set of three strake-specific subroutines: NSTAT, FSDFO, and FSROB. These routines were created using the DATAFIX program. The primary ANSER subroutine NSTAT has the inputs and outputs listed in Tables 3.7 and 3.8 and is called from subroutine AEROINC.

Table 3.7. Input parameters for subroutine NSTAT

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
M	MACHRF	AMCH	CSL	Mach number
α	ALDGRF	ADEG	CSL	angle of attack, deg
β	BEDGRF	BDEG	CSL	angle of sideslip, deg
k	-	WNONDIM	AEROINC	N.D. rotation rate
pnd	-	PND	AEROINC	N.D. body-axis roll rate
qnd	-	QND	AEROINC	N.D. body-axis pitch rate
rnd	-	RND	AEROINC	N.D. body-axis yaw rate
$\delta_{S_{diff}}$	-	DFSDIF	AEROINC	differential nose strake deflection, deg
$\delta_{S_{symm}}$	-	DFSSYM	AEROINC	symmetric nose strake deflection, deg
-	DG2RA	DG2RA	INITIAL	degrees-to-radians conversion factor, acos(-1.0)/180.0

Table 3.8. Output parameters for subroutine NSTAT

Symbol	*f18harv* Name	Local Name	Definition
$\Delta C_{D_{strake}}$	-	DCDFS	ANSER drag-coefficient increment
$\Delta C_{L_{strake}}$	-	DCLFS	ANSER lift-coefficient increment
$\Delta C_{m_{strake}}$	-	DCMFS	ANSER pitching-moment-coefficient increment
$\Delta C_{Y_{strake}}$	-	DCYFS	ANSER sideforce-coefficient increment
$\Delta C_{l_{strake}}$	-	DCRMFS	ANSER rolling-moment-coefficient increment
$\Delta C_{n_{strake}}$	-	DCNFS	ANSER yawing-moment-coefficient increment

The subroutines FSDFO and FSROB contain the data table lookup and increment calculations for the forced-oscillation and rotary-balance contributions, respectively. The subroutine NSTAT contains calls to FSDFO and FSROB, and the full ANSER aerodynamic increments are formed there. Inputs to subroutines FSDFO and FSROB are listed in tables 3.9 and 3.10, respectively. All outputs from FSDFO and FSROB are passed through common block STRAKE.

Table 3.9. Input parameters for subroutine FSDFO

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
α	ALDGRF	ADEG	CSL	angle of attack, deg
qnd	-	QND	AEROINC	N.D. body-axis pitch rate
pnd	-	PND	AEROINC	N.D. body-axis roll rate
rnd	-	RND	AEROINC	N.D. body-axis yaw rate
-	DG2RA	DG2RA	INITIAL	degrees-to-radians conversion factor, acos(-1.0)/180.0

Table 3.10. Input parameters for subroutine FSROB

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
k	-	WNONDIM	AEROINC	N.D. rotation rate
α	ALDGRF	ADEG	CSL	angle of attack, deg
β	BEDGRF	BDEG	CSL	angle of sideslip, deg
$\delta_{S_{diff}}$	-	DFSDIF	AEROINC	differential nose strake deflection, deg

All ANSER aerodynamic model calculations are performed at 80 Hz update rate. Currently in the *f18harv* simulation the rotary-balance terms for the strakes are computed but not used as part of the dynamic contribution of the strake aerodynamic increments. Future simulation users may want to add this model depending on the research being performed.

4.0 THRUST VECTORING SYSTEM

The engine model originally included with the *f18bas* simulation was replaced with a representation of the F/A-18 HARV engine received from DFRC in the form of FORTRAN and ADA source code subroutines. The thrust-vectoring model in the *f18bas* simulation was replaced in the *f18harv* simulation with a model which represented the actual HARV Thrust-Vectoring System (TVS). The *f18bas* simulation employed a TVS which consisted of two paddles per engine nozzle while the F/A-18 HARV uses three vanes per engine nozzle.

4.1 Engine Thrust Model

A block diagram of the engine model implemented in *f18harv* is shown in figure 4.1. The steady-state engine performance of the General Electric GE-404 engine without thrust vectoring is determined by table look-up in the subroutine TLUHRV. Input parameters are lagged power-lever-angle, Mach number, and altitude. The table is composed of 29 arrays of data. Four arrays contain the values at flight idle, military power (mil pwr), minimum afterburner (min A/B), and maximum afterburner (max A/B) for each of seven dependent variables, which are nozzle area, drag, gross thrust, nozzle pressure ratio, turbine discharge pressure, ram drag, and fuel flow. The calculations for fuel-flow sequence used to compute fuel usage, which in turn adjusts the weight, center of gravity, and moments of inertia during a simulation run, were neglected. The last array corresponds to windmill drag. Each array consists of 12 rows by 7 columns. The rows correspond to Mach numbers at 0.0, 0.2, 0.4, 0.6, 0.8, 0.9, 1.0, 1.1, 1.2, 1.4, 1.6, and 1.8. The columns correspond to altitudes at 0, 10, 20, 30, 35, 40, and 50 kft. At each sample time (0.0125 seconds), TLUHRV linearly interpolates over Mach number and altitude to find the flight idle, mil power, min A/B, and max A/B for each of the seven variables. Effects on gross thrust of the T56 bias for stall-margin enhancement and of angle of attack (AOA) are then incorporated as multiplying factors.

Engine dynamics are approximated by incorporating first-order-lag dynamics on the throttle, or PLA, prior to the table look-up. These dynamics also include rate limiting, which is applied only when the throttle is increasing. The time constants are 0.625 sec and 0.55 sec for non-A/B and A/B, respectively, and similarly the rate limits are 14 deg/sec and 22 deg/sec.

The effectiveness of the vanes in vectoring the thrust is computed by table look-up in subroutine TVANGL using data taken by LaRC in the cold-jet facility. The dependent variable is the effective thrust vectoring, or effective turning angle, as a function of vane deflection, nozzle area, and nozzle pressure ratio (NPR) for two vanes deflected, assuming that the third is stowed at -10 degrees. The table is composed of 60 arrays in groups of 20, where each of the three groups contains data for the case of a different stowed vane (a different non-participating vane). Within a group of 20 there are 10 arrays containing the effective turning angles in the pitch plane and 10 arrays for the yaw plane. Within the subgroup of 10 there are five arrays of data for a nozzle area of 220 in^2 representing mil power for NPR's of 2, 3, 4, 5, and 6, and five arrays for an area of 343 in^2 representing max A/B. Each array is 11 rows by 11 columns and contains effective turning-angle data for all combinations of vane deflection angles of -10, -5, 0, 5, 10, 15, 17.5, 20, 22.5, 25, and 30 degrees for two vanes. The thrust-vectored pitch and yaw turning angles in the x-z and x-y engine axis planes, respectively, are limited to $\pm$ 30 degrees. Effective turning angles in pitch and yaw are determined by doing a 4-dimensional linear interpolation of the tabular data.

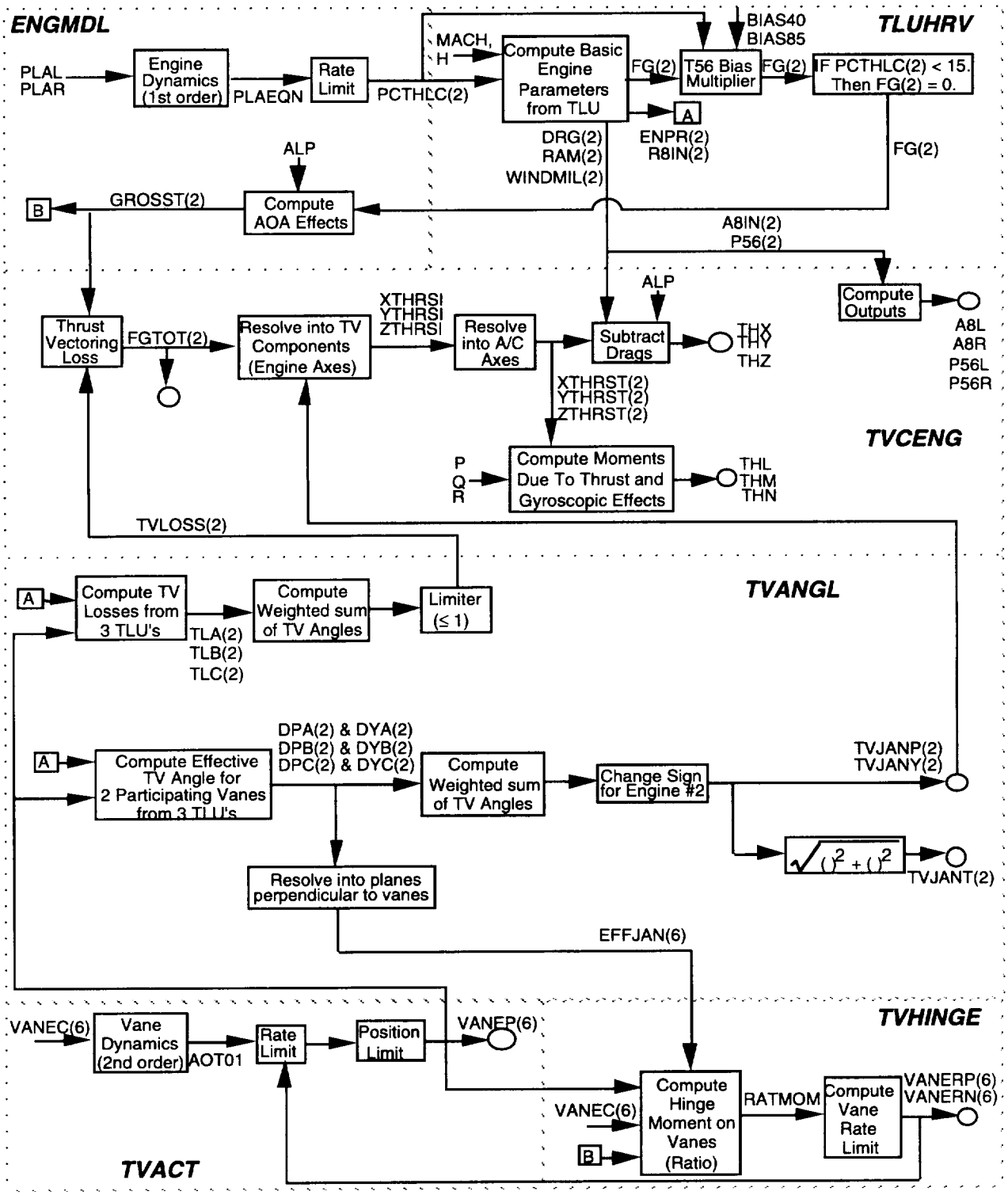


Figure 4.1. Engine Model Block Diagram

Vane-heating code in subroutine TVANGL computes vane temperatures for each vane in units of degrees Fahrenheit. The vane temperature outputs are stored in array VANTMP to help the user determine if excessive vane heating occurs as a result of vane deflections.

Since the tables contain data describing the effective jet-turning angles for two active vanes assuming that the third vane is stowed at -10 degrees, the first step in the look-up procedure is to determine which vane is non-participating, that is, the vane with the smallest deflection. The tables are then used to compute the turning angles produced by the two participating vanes as discussed in the previous paragraph. The two participating vanes are then assumed to be stowed at -10 degrees, and the tables are used to compute the turning angles due to the non-participating vane at its actual, rather than stowed, position. Outputs are the effective jet-turning angles in the pitch and yaw planes for each engine.

The gross-thrust loss due to vectoring is likewise computed via table look-up using cold-jet data. The table, which is grouped like the effective turning-angle table, is composed of 30 arrays arranged according to the non-participating vane, nozzle area, and NPR. Each 11-by-11 array contains the TV-loss, or reduction factor, as a function of the nozzle area and NPR. The appropriate TV-loss for each engine is found by 4-dimensional linear interpolation and is used by subroutine TVCENG.

The F/A-18 HARV aircraft vane actuators are the same as the aileron actuators, and they are modeled in subroutine TVACT as discrete second-order systems with rate and position limiting. The actuator-model transfer function is defined as

$$H(s) = \frac{75^2}{s^2 + 2(0.59)75s + 75^2} \tag{4.1}$$

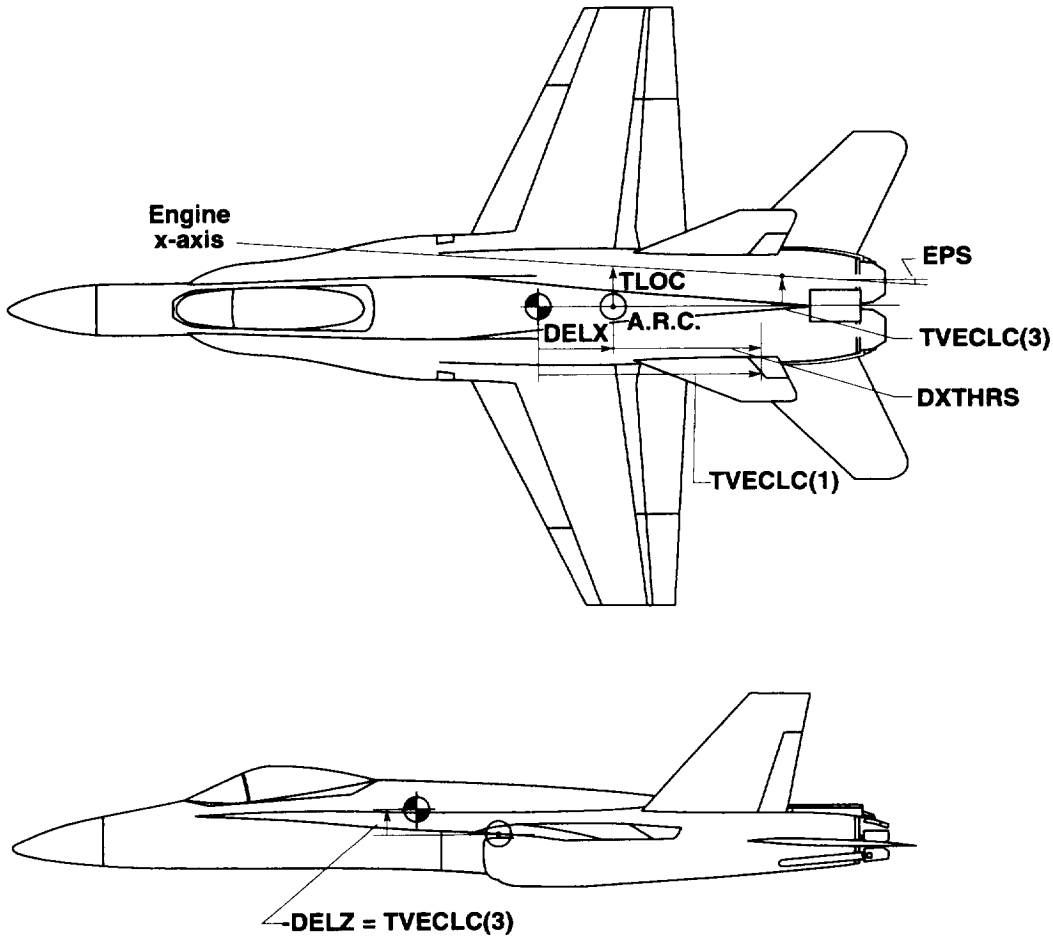
The rate limits are computed in subroutine TVHINGE as a function of the hinge moment produced on the vane by the thrust. The vane-actuator position limits are +25 and -10 deg, and the no-load rate limits are ± 80 deg/sec.

The gross thrust for each engine is modified in subroutine TVCENG to account for thrust-vectoring losses and is then resolved into engine axes using the effective pitch and yaw turning angles. Figure 4.2 lists the engine axis locations that can be found in subroutine TVCENG. Gross thrust is further resolved into aircraft body axes for the subtraction of ram, inlet/aftbody, and windmill drags along with the computation of the total engine forces and moments for use in the equations of motion.

4.2 McAir Mixer/Predictor

Two Mixer/Predictor (M/P) designs have been incorporated in the *f18harv* simulation. The first is referred to as the McAir M/P as this design originated at McDonnell-Douglas and was the first M/P to be implemented on the F/A-18 HARV aircraft and in the simulations. A brief description of the McAir M/P is included in this section. The second M/P design is presented in section 4.3. A new subroutine called MIX was developed to interface between ACSL and the Thrust Estimator and either of the M/P designs available in the *f18harv* simulation.

The function of the Mixer/Predictor is to accept pitch and yaw thrust-vectoring commands from the Research Flight Control System (RFCS) and to convert them into individual commands to each of the six thrust-vectoring vanes. The Mixer/Predictor is implemented in the HARV research flight computer (1750A PACE Computer) along with the RFCS.



EPS = 2°
TLOC(1) = -2.33 ft
TLOC(2) = +2.33 ft
DXTHRS = 20.1 ft
TVECLC(1) = -20.2803 ft
TVECLC(2) = +0.449 ft

Figure 4.2. - Engine Axis Locations

The McAir M/P is contained in subroutine MIXPRE. Inputs to the subroutine are pitch and yaw thrust-vectoring-angle commands (PJETAC, YJETAC), nozzle pressure ratios (LNPR, RNPR), nozzle areas (A8L, A8R), estimated gross thrusts (LGTHR, RGTHR), sensor Mach number (MACH), and sensor-model static pressure (STPRES). Outputs are the six vane deflection commands (VANEC(6)). A block diagram of this M/P is shown in figure 4.3. The vane nomenclature, or numbering scheme, and vane locations are shown in figure 4.4, and the sign conventions are shown in figure 4.5.

The RFCS computes the required thrust-vectoring angles assuming that the gross thrust is constant at the reference level of 15000 lbs. The first task of the M/P is to convert these angle commands to equivalent commands for the actual engine thrust level, that is, to vectoring angles producing the required moments in the pitch and yaw directions. Since the actual thrust is not measured, the M/P uses estimates of the thrust from the Thrust Estimator to be discussed later. The commands are then hard-limited to +16 and -20 degrees in pitch and to ± 10 degrees in yaw. Commands are further limited as functions of Mach number and altitude (static pres-

sure) to reduce loads on the vanes and angular accelerations of the aircraft.

The pitch and yaw vectoring commands are next transformed into polar coordinates, and these commands are the average amount of thrust vectoring required from the left and right engines combined. If these commands were used for both engines, the deflections of the corresponding vanes on the two engines (e.g., vanes 2 and 6) could be considerably different because the vanes on the two engines are anti-symmetrical. Therefore, if the magnitude of the vectoring command is greater than six degrees, the pitch and yaw commands (in polar coordinates) for the left and right engines are adjusted such that the deflections of the corresponding vanes are approximately equal. This adjustment prevents the vane on one engine from reaching a position limit significantly before the vane on the other engine does, and assures that corresponding vanes are commanded to move similar distances, thus reaching their commanded deflections at approximately the same time.

The angular polar coordinate of the commands are examined to determine which of the vanes is the non-participating vane for each engine. The commands are then resolved into effective thrust-vectoring angles for each of the four participating vanes. For an individual vane the transformation of the effective thrust-vectoring-angle command for that vane into a commanded deflection angle is based on the same LaRC cold-jet data used in the simulation of the HARV TVS. However, instead of using the entire data table in a table look-up, the data is represented by a piece-wise linear model of the effective thrust-vectoring angle as a function of angle of attack, as shown in figure 4.6 (a). The angle of attack is the actual vane deflection plus a contribution, defined as vanewash, from the other participating vane, modeled as a linear function (fig 4.6 (b)). There are different curves for four nozzle pressure ratios (3.0, 4.0, 5.0, and 6.0) and for two nozzle areas (max A/B and mil power) for a total of eight curves for a nominal-size vane and eight for a large vane. Model parameters stored in subroutine MIXPRE are thrust deadband, low slope, high slope, angle-of-attack breakpoint, effective vectoring breakpoint, vanewash deadbands, and vanewash slopes. Linear-model vane-deflection-command parameters are determined from linear interpolation as functions of nozzle pressure ratio and nozzle radius. Since nozzle pressure is not measured, an estimate of NPR is supplied by the Thrust Estimator. From the required effective vectoring for each of the two participating vanes on a particular engine, the vanewash and the required vane angle of attack are computed from the linear models. These are then subtracted to obtain the required vane deflection.

To prevent the non-participating vane from producing unwanted forces, the non-participating vane is position limited to -10 degrees when the plume is being deflected towards it with a vectoring magnitude of at least 6 degrees. "Directed towards it" means that the polar coordinate of the vectoring direction is at least 30 degrees from the edge of the region where the non-participating vane becomes participating. (See figure 4.7) When the plume is being deflected within 20 degrees of the edge or when the magnitude of the vectoring is 4 degrees or less, the vane is placed on the plume boundary to reduce the reaction time should deflection of the vane into the plume be required. Between these regions the non-participating vane command is found by linearly interpolating between the two above solutions.

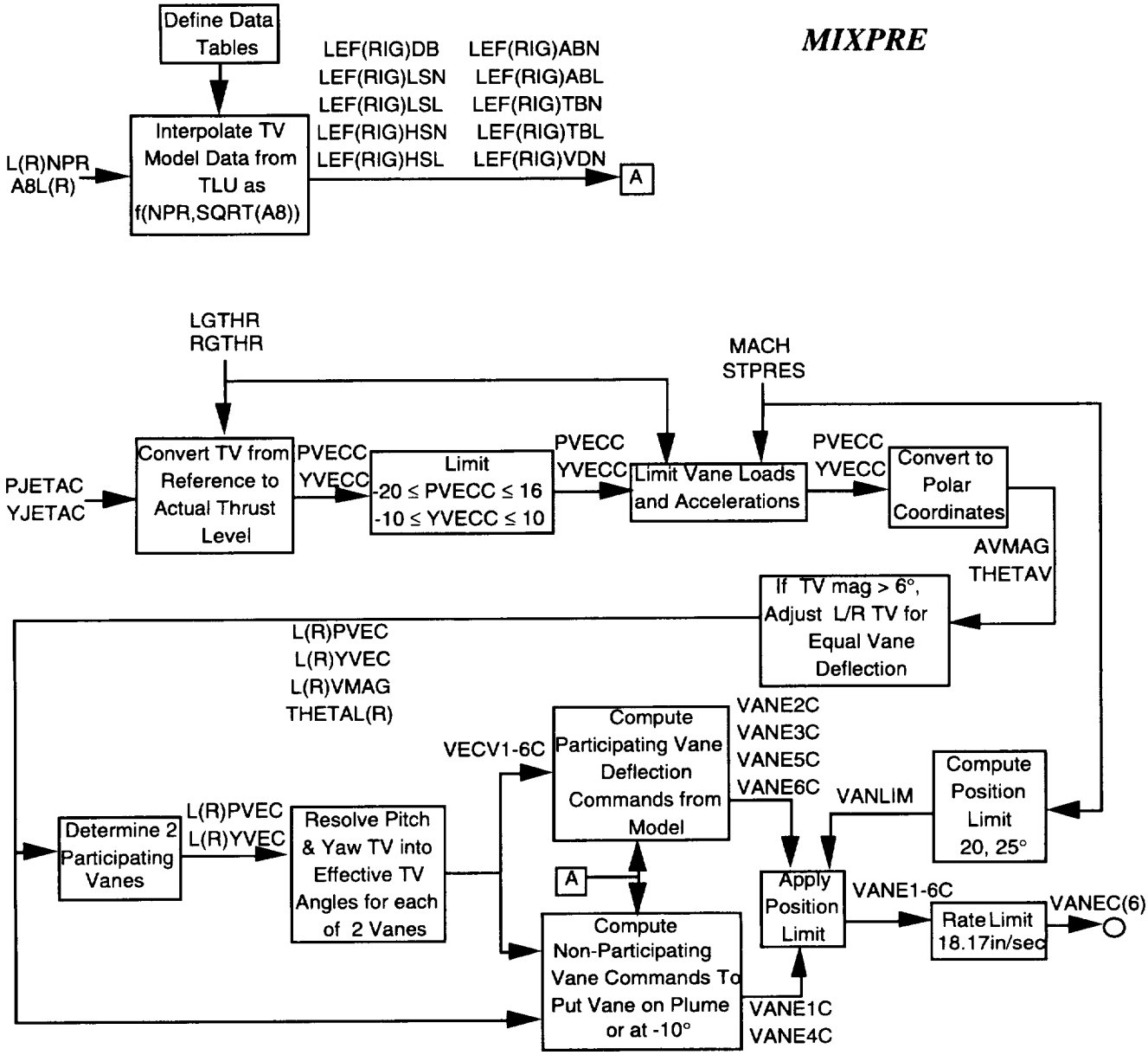


Figure 4.3. - Block diagram for the McAir Mixer/Predictor

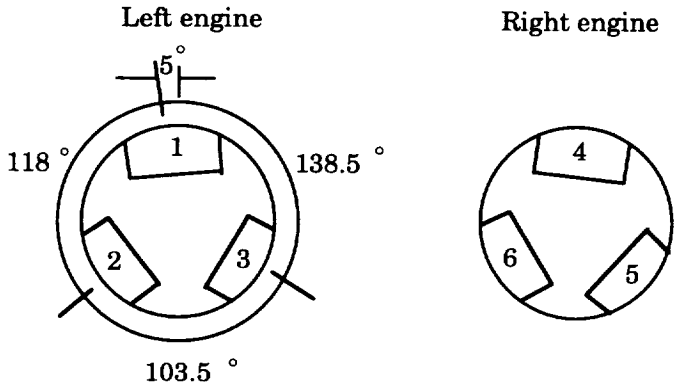


Figure 4.4. - Vane numbering scheme.

The above procedure computes the required deflections for the six vanes. These deflections are then position limited on the positive side using a limit value between 20 and 25 degrees computed as a function of Mach number and static pressure. Rate limits are then applied to obtain the final vane commands.

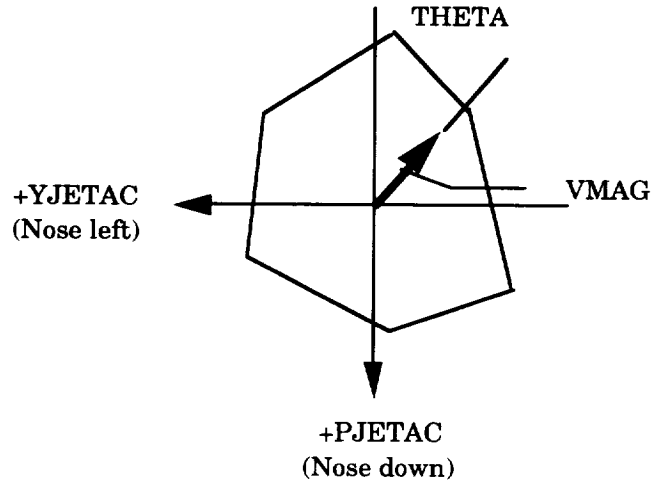


Figure 4.5. - Thrust vectoring sign convention.

4.3 LaRC/DFRC Mixer/Predictor

The second Mixer/Predictor design available in the *f18harv* simulation is referred to as the LaRC/DFRC M/P since this design originated at LaRC and DFRC . This new M/P design was initiated to enhance performance of new RFCS designs, to enhance understanding of the M/P operation, provide an-easier-to-modify design, and to take advantage of lessons learned from flight test of the McAir M/P.

The LaRC/DFRC M/P implemented in the *f18harv* simulation is version 4.5. A detailed discussion of this M/P design is presented in reference 4.0. New features incorporated into the LaRC/DFRC M/P design are listed below.

- The ability to apply a roll thrust-vectoring command.

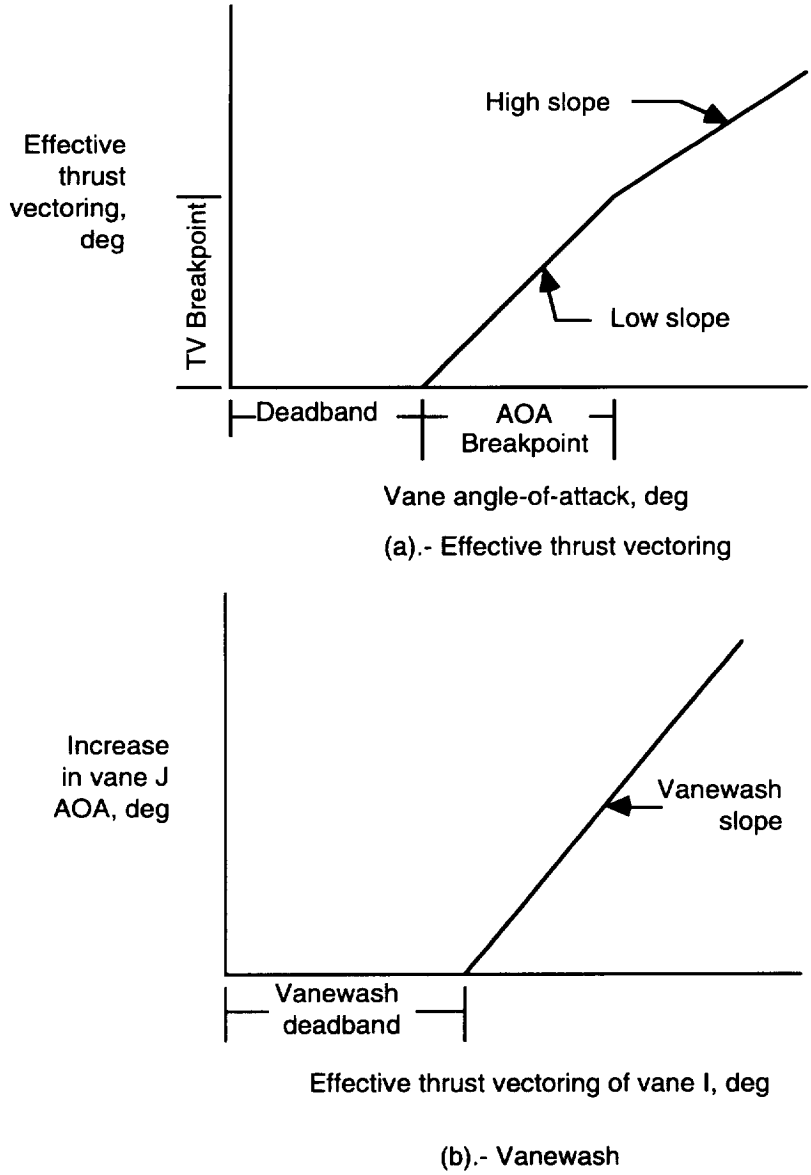


Figure 4.6. - Linear models of LaRC cold-jet data.

- Priority logic to give priority first to pitch, then yaw, and then roll thrust-vectoring commands when all of these commands are not simultaneously achievable.
- New thrust adjustment factor used in the computation of the vane deflections. This factor was decreased at low thrust levels to reduce small vane deflections (also referred to as vane rumble in flight).
- An algorithm was added to estimate the thrust loss due to thrust vectoring. This estimate is used to adjust the vane-deflection commands.
- Vane limiting logic as a function of mach, altitude, and power-lever-angle was added for structural load considerations.

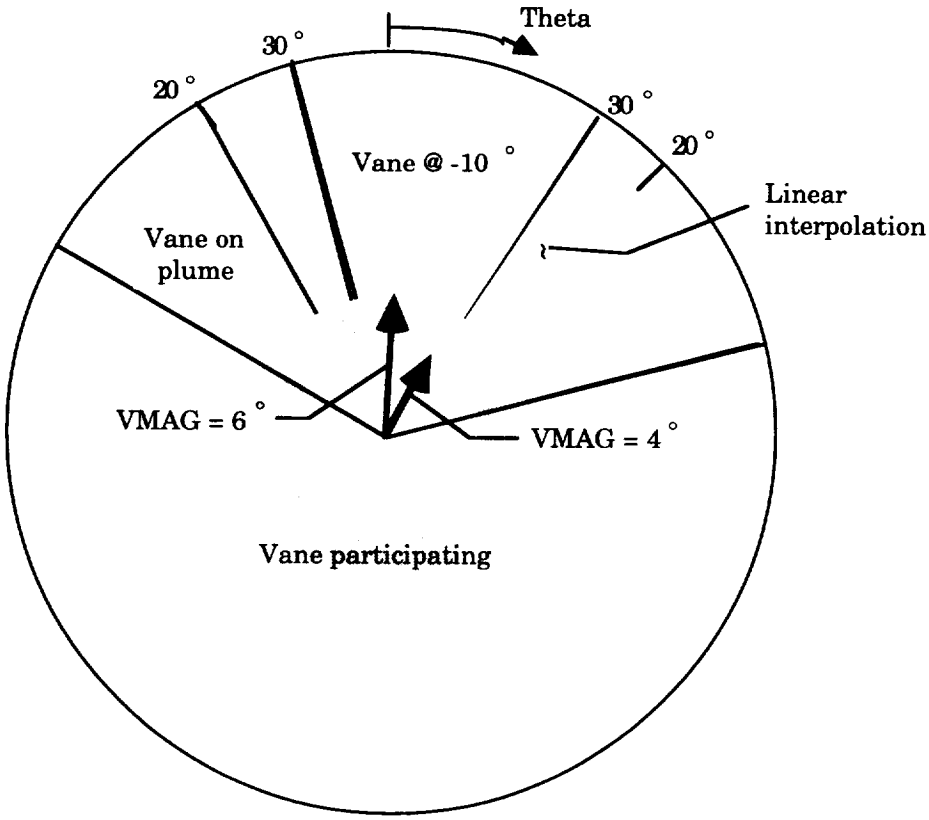


Figure 4.7. - Positioning of non-participating vane.

- A deadband was added to the thrust-vectoring command to reduce vane movement due to measurement noise.

The *f18harv* simulation uses a LOGICAL switch named LMIX4 (default = .TRUE.) to determine if the LaRC/DFRC or McAir M/P design will be simulated. The McAir M/P will be simulated if this LOGICAL is FALSE.

Figure 4.8 presents a flow chart of the LaRC/DFRC M/P design. Table 4.1 contains the FORTRAN subroutine and function names which are CALLed in subroutine MIXER_4. A brief description of the calculations performed is included.

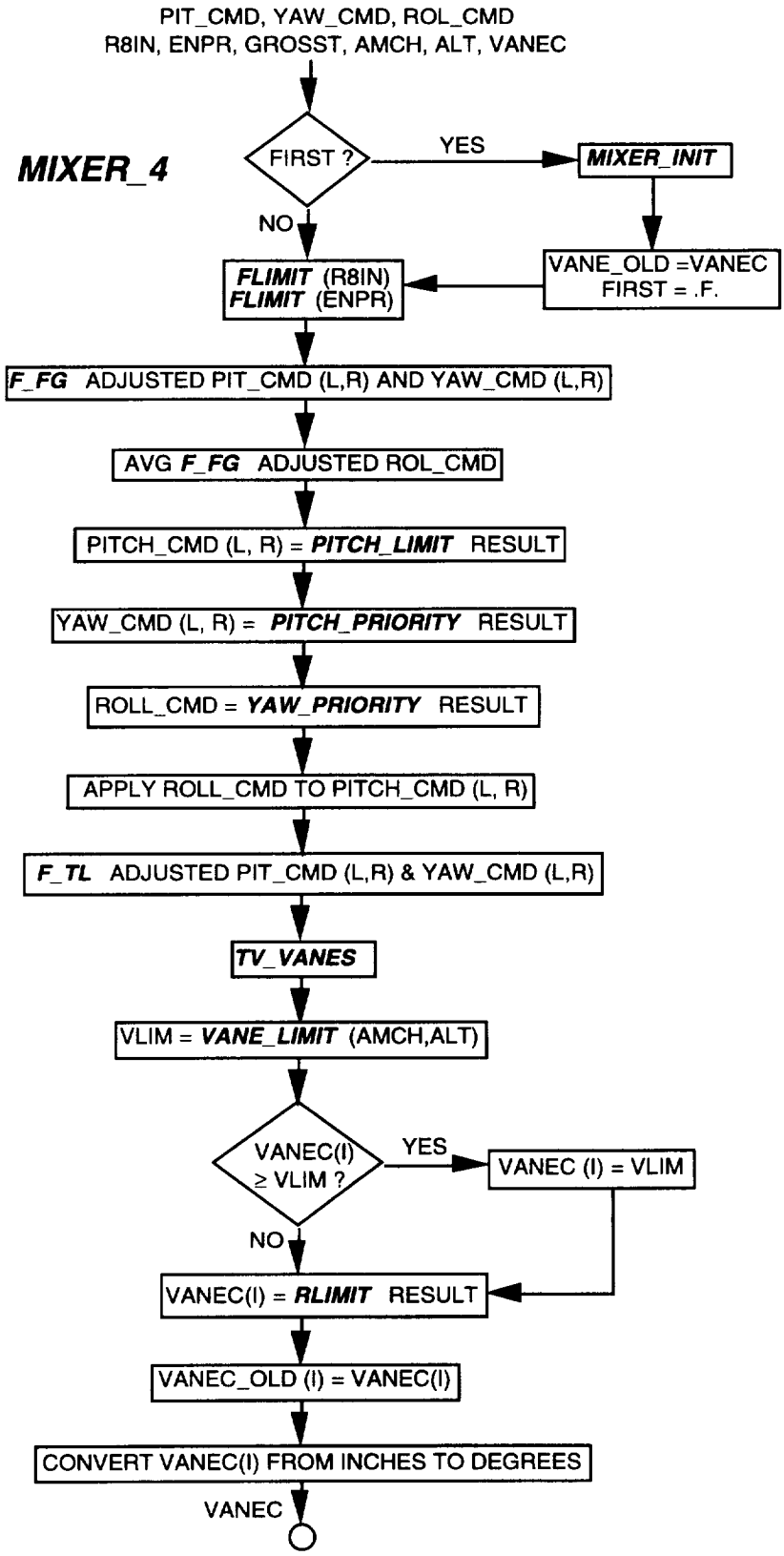


Figure 4.9. - LaRC/DFRC Mixer/Predictor flow chart

Table 4.1. Subroutine CALLs in MIXER_4

Subroutine Name	Description
MIXER_INIT	initialization
FLIMIT	limits input parameter
F_FG	scales pitch and yaw commands with the thrust available
PITCH_LIMIT	limits the pitch vectoring commands
PITCH_PRIORITY	limits the yaw vectoring command as a function of the pitch vectoring command
YAW_PRIORITY	limits the roll vectoring command as a function of the average limited yaw and pitch vectoring command
F_TL	approximates thrust loss due to RMS vectoring magnitude as 1%/deg of total vectoring
TV_VANES	computes vane commands from table schedules
VANE_LIMIT	limits the maximum vanes commands in the lower left and right hand corner of the F/A-18 HARV flight envelope
RLIMIT	rate limit vane commands to 120 deg/sec
ASIN*	arcsine function

* FORTRAN Functions

4.4 Thrust Estimator

The thrust estimation algorithms are contained in two subroutines, THREST and THRCAL. Inputs to THREST are throttle setting, nozzle area, turbine discharge pressure, and static pressure. Outputs are estimates of gross thrust and nozzle pressure ratio. THRCAL is CALLed by THREST. A flow chart of the thrust estimator (TE) is shown in figure 4.9.

Subroutine THREST first performs reasonableness checks, such as out-of-range checks and left-right comparisons, on the input data. If the checks fail, estimates are not computed on that interation. The reasonableness checks are a carry over from the flight ADA code which this subroutine was written from. The flight ADA code check inputs to ensure data is within reasonable bounds. Subroutine THRCAL is CALLed to compute the estimates of NPR and thrust using empirical formulae. THREST then performs reasonableness checks on these estimates, and if the checks fail, the new estimates are discarded and the previous estimates are used.

4.5 Implementation

The first task in integrating the engine and the TVS models into the *f18harv* simulation was the removal of the code implementing the previous TVS and engine models in the *f18hbas* simulation. Included in the removal were CONSTANT statements related to the engine, CALLs to the thrust-vectoring-vane actuator macros, the throttle boost actuator, CALLs to subroutines ENG1, TVENG1, ENG2, THRVEC, assignment statements resolving forces and moments into body-axis components, and DISCRETE blocks ABL and ABR. The optional DMS throttle was not removed.

Since the engine model is a collection of subroutines, a new FORTRAN subroutine ENG-MOD was written to CALL the other subroutines in the proper sequence and to communicate with ACSL via a parameter list. Dynamic simulation of the engine, M/P, and thrust estimator

is performed by CALLing THREST, MIX, and ENGMOD with the appropriate parameters at an 80 Hz sample rate. These subroutine CALLs are placed in the ACSL code DISCRETE Block ENG within the DYNAMIC Block with sample interval DTENG = 0.0125. The DFRC simulations were implemented so that the engine subroutines were CALLed every sample period, but only one engine was updated during each CALL to reduce computation time. Even though the LaRC sample rate is different, this structure was maintained such that each engine is updated on alternate executions of the DISCRETE Block ENG resulting in a 40 Hz update rate. Appropriate parameters in the discrete models of the dynamics have been adjusted accordingly.

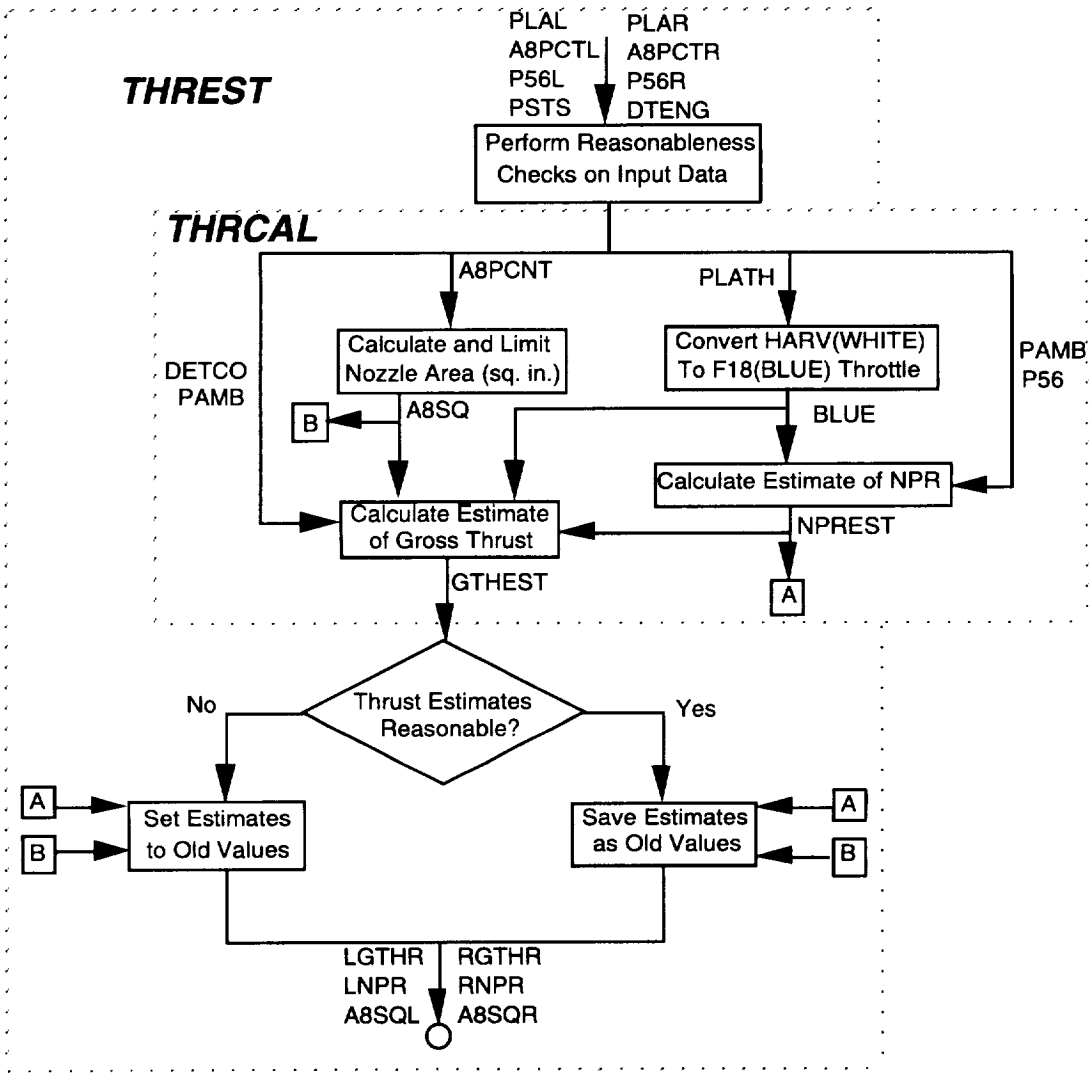


Figure 4.9. - Thrust Estimator flow chart

Initialization of parameters in the engine model is accomplished in the INITIAL Block by a number of CONSTANT statements and by a CALL to the subroutine INTENG. This is a new subroutine which CALLs subroutines HARVIN and VANEIN to read in three data files and then CALLs subroutine ENGDIN to calculate engine-model constants. This is followed by a CALL to ENGMOD using trim values or initial condition values as actual parameters to compute initial values for P56 and A8 for use in the first CALL to THREST. To initialize the engine and vane actuator dynamics, CALLs are then made to MIX and ENGMOD, again using

trim values as the actual parameters for the input variables. These three CALLs are made after the CALL to the flight control system so that thrust-vectoring commands in the pitch, roll, and yaw planes are available from the RFCS. This sequence is important not only for initialization but also for the trim process to be discussed later. Flags have been incorporated into subroutines ENGMDL and TVACT to ensure that the engine and actuator dynamics for both engines are initialized during the first CALL to ENGMOD. The CALLing sequence for the engine, M/P, and Thrust Estimator subroutines is shown in figure 4.10.

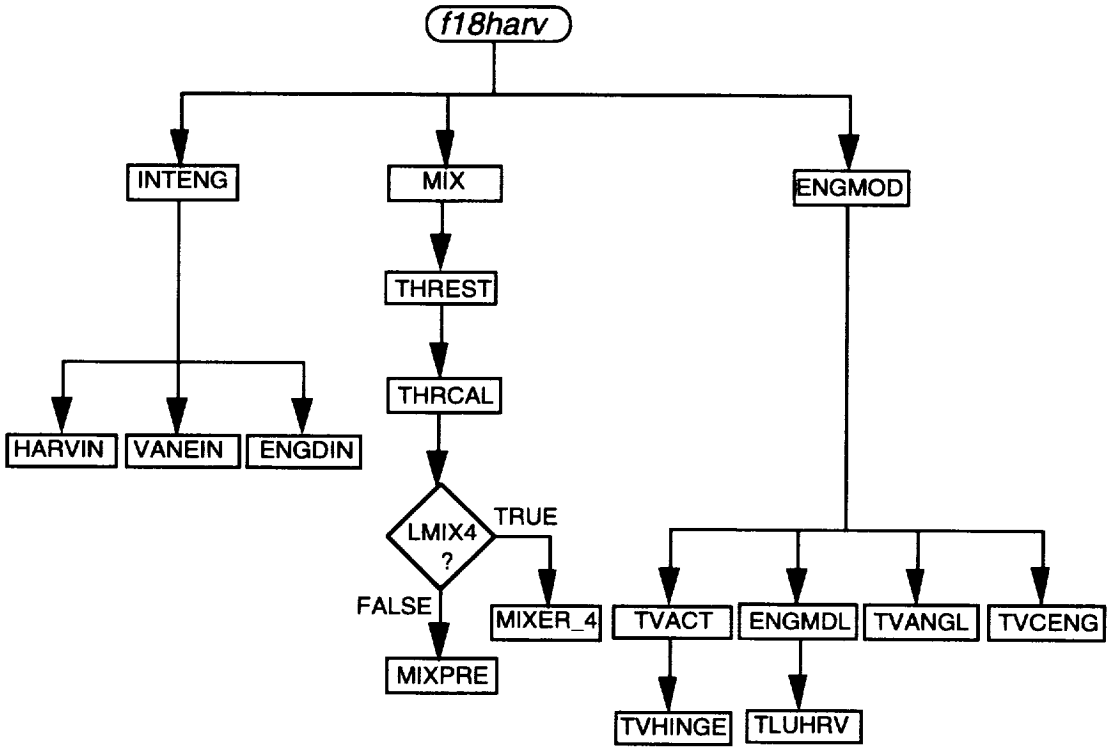


Figure 4.10.- Subroutine CALLing sequence.

Three input data files containing the basic engine data table, the thrust-vectoring effective-turning-angle data table, and the thrust-vectoring-loss data table are required by the *f18harv* simulation engine model and the Mixer/Predictor/Thrust Estimator during execution of the simulation. The engine-model data read into *f18harv* is stored in subroutines HARVIN and VANEIN. Before execution these files should be assigned to the proper units as listed in Table 4.2. No output files are produced by the engine model, M/P, or Thrust Estimator.

Table 4.2. Input Files for the Engine Model and M/P/Thrust Estimator

Description	File name	Unit number
Basic engine data	F18_HARV_ENG.DAT;1	FORTRAN21
Thrust vectoring effectiveness	F18_COLDJET4.DAT;1	FORTRAN19
Thrust vectoring losses	F18_THRLOS4.DAT;1	FORTRAN22

Data is assigned using COMMON blocks and passed between various engine model, M/P,

and TE subroutines. Figure 4.11 presents a diagram of the parameters stored in these COMMON blocks.

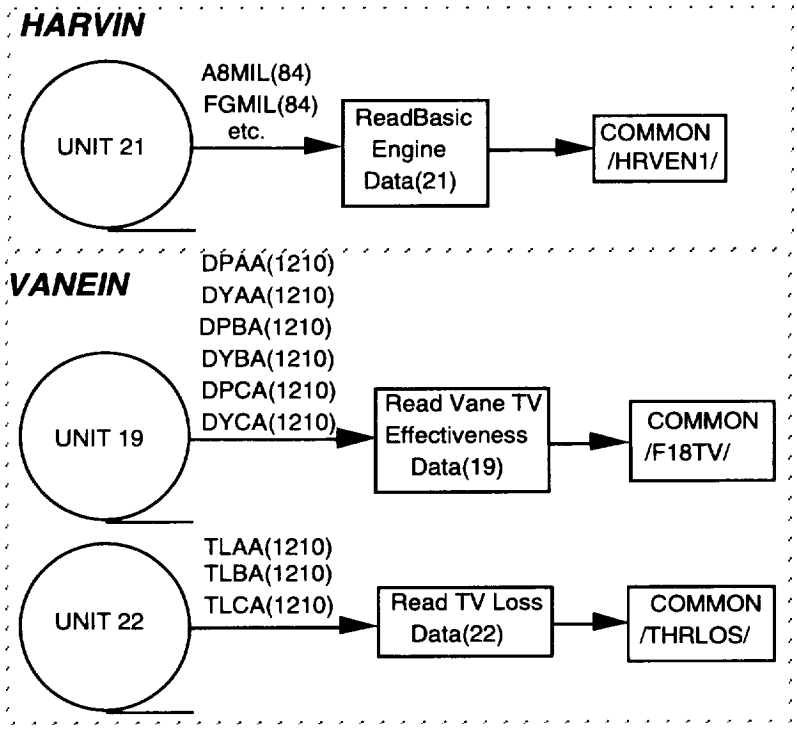


Figure 4.11. - Engine-model data read into the *f18harv* simulation

Subroutines INTENG, MIX, and ENGMOD use parameters from the ACSL code for both the initial CALLs in the INITIAL Block and the CALLs in the DYNAMIC Block. Aircraft parameters required by the engine model are angle of attack, Mach number, altitude, and attitude rates. Commands required are the throttle, or power-lever-angle, logical values for the stall-margin-enhancement flags, and M/P vane deflections. Logicals LLN and LTRFLG will be discussed later, but note that LTRFLG is set automatically by the simulation, not by the user. Aircraft measurements required by the M/P and Thrust Estimator are sensor-model Mach number, nozzle area, turbine discharge pressure, and sensor-model static pressure. Nozzle area in inches squared from the engine simulation (TLUHRV via ENGMOD) is converted to percent in subroutine MIX for input into the Thrust Estimator. The Thrust Estimator converts the nozzle area from percent to inches squared for use by the M/P. Since gross thrust and NPR are not measured, estimates of these produced by the Thrust Estimator are used as inputs to the M/P. Commands required by the M/P are the effective pitch, roll, and yaw jet-turning angles produced by the RFCS. Commands required by the Thrust Estimator are the throttle positions.

Subroutines have been modified to enable multiple CALLs to be made to accommodate simulating multiple airplanes for the DMS. These modifications involved adding to the subroutine parameter lists all of the parameters that must be saved for each airplane from one iteration to the next. The subroutine MIX is CALLed directly from the ACSL code in the *f18harv* simulation. The first CALL to MIX for each airplane in any run should be made with NTHRER = 0 and ICALLM = 1 so that the rate limiters in the vane command computation will be initialized correctly (by setting the previous command to the current one). During the first CALL the subroutine sets ICALLM = 2, and this value should be saved externally for each airplane to reuse in subsequent CALLs. NTHRER accumulates the number of iterations

that the thrust estimator fails its data tests, and NTHRER should be saved externally for each airplane to reuse in subsequent CALLs. Logical variables LLN and LTRFLG are used when computing Jacobians and when computing trim, respectively, in ACSL. Setting either of these equal to TRUE bypasses the rate limiters in the vane command computations. Both LLN and LTRFLG should be set to FALSE during a normal run. Outputs to the simulation from the M/P and Thrust Estimator are commanded vane positions for input into the vane actuators and estimates of the gross thrust and the nozzle pressure ratio for information and data reduction purposes.

Tables 4.3 through 4.13 list the input and output parameters for the various subroutines used to implement the M/P and Thrust Estimator. Variable names external (*f18harv* name) and internal (local name) to the subroutine are included. For the input parameters the tables also define where the parameter s are calculated (simulation source).

Table 4.3. Input parameters for subroutine MIX

f18harv Name	Local Name	Simulation Source	Definition
LGTHR	LGTHR	THREST	estimated left engine gross thrust, lbs
RGTHR	RGTHR	THREST	estimated right engine gross thrust, lbs
LNPR	LNPR	THREST	estimated left engine nozzle pressure ratio, n.d.
RNPR	RNPR	THREST	estimated right engine nozzle pressure ratio, n.d.
PLAL	PLAL	ENGMDL	left engine throttle position, degrees
PLAR	PLAR	ENGMDL	right engine throttle position, degrees
PJETAC	PJETAC	RFCS	thrust-vectoring pitch-jet-angle command, degree
YJETAC	YJETAC	RFCS	thrust-vectoring yaw-jet-angle command, degrees
A8LS	A8L	THREST	estimated left engine nozzle throat area, inches^2
A8RS	A8R	THREST	estimated right engine nozzle throat area, inches^2
P56LS	P56L	TVCENG	left turbine discharge pressure, lbs/inches^2
P56RS	P56R	TVCENG	right turbine discharge pressure, lbs/inches^2
MACHS	MACH	SENSOR	Mission Computer Mach number, n.d.
PSTS	PSTS	SENSOR	sensor-model static atmospheric pressure, psf
NTHRER	NTHRER	CSL	counter which accumulates number of times the thrust estimator fails its data compatibility tests
ICALLM	ICALLM	CSL	integer switch to multiple CALL status for DMS
DTENG	MIXDT	INITIAL	sample time interval at which discrete engine model is CALLed (0.0125 seconds).
LLN	LLN	CSL	logical switch for Jacobian calculations
LTRFLG	LTRFLG	THREST	logical switch to indicate trim status
RJETAC	RJETAC	RFCS	thrust-vectoring roll jet angle command, degrees
H	H	CSL	altitude, ft
LMIX4	LMIX4	CSL	logical switch to indicate MIXER design
MIX4INT	MIX4INT	CSL	integer switch used for initialization
VANECO(6)	VANECO(6)	CSL	previous frame vane commands, degrees
DGTHLC(2)	DGTHLC(2)	ENGMDL	lagged left right throttle position, degrees

Table 4.4. Output parameters for subroutine MIX

f18harv Name	Local Name	Definition
A8LEST	A8SQL	estimated left engine nozzle throat area, inches^2
A8REST	A8SQR	estimated right engine nozzle throat area, inches^2
A8PCTL	A8PCTL	left engine nozzle throat area, %
A8PCTR	A8PCTR	right engine nozzle throat area, %
LTOPVC	LTOPVC	left top vane command, inches
LOUTVC	LOUTVC	left outboard vane command, inches
LINBVC	LINBVC	left inboard vane command, inches
RTOPVC	RTOPVC	right top vane command, inches
ROUTVC	ROUTVC	right outboard vane command, inches
RINBVC	RINBVC	right inboard vane command, inches
PLTVC	PLTVC	left top vane command from previous iteration, inches
PLIVC	PLIVC	left inboard vane command from previous iteration, inches
PLOVC	PLOVC	left outboard vane command from previous iteration, inches
PRTVC	PRTVC	right top vane command from previous iteration, inches
PRIVC	PRIVC	right inboard vane command from previous iteration, inches
PROVC	PROVC	right outboard vane command from previous iteration, inches
LGLGT	LGLGT	last good estimate of left gross thrust, lbs
LGA8L	LGA8L	last good estimate of left nozzle throat area, inches^2
LGLNPR	LGLNPR	last good estimate of left nozzle pressure ratio, n.d.
LGRGT	LGRGT	last good estimate of right gross thrust, lbs
LGA8R	LGA8R	last good estimate of right nozzle throat area, inches^2
LGRNPR	LGRNPR	last good estimate of right nozzle pressure ratio, n.d.
VANEC(6)	VANEC(6)	vane commands, degrees
PVECC	PVECC	thrust adjusted pitch thrust-vector command, degrees
YVECC	YVECC	thrust adjusted yaw thrust-vector command, degrees
RVECC	RVECC	thrust adjusted roll thrust-vector command, degrees
PJETC	PJETC	final pitch thrust-vector command, degrees
YJETC	YJETC	final yaw thrsut-vector command, degrees

Table 4.5. Input parameters for subroutine THREST

f18harv Name	Local Name	Simulation Source	Definition
DGTHLC(1)	PLAL	ENGMDL	left engine lagged throttle position, degrees
DGTHLC(2)	PLAR	ENGMDL	right engine lagged throttle position, degrees
A8PCTL	A8PCTL	TVCENG	left engine nozzle throat area, %
A8PCTR	A8PCTR	TVCENG	right engine nozzle throat area, %
P56L	P56L	TVCENG	left turbine discharge pressure, lbs/inches^2
P56R	P56R	TVCENG	right turbine discharge pressure, lbs/inches^2
PSTS	STPRES	SENSOR	sensor-model static atmospheric pressure, psf
MIXDT	DTENG	INITIAL	sample time interval at which discrete engine model is CALLed (0.0125 seconds).

Table 4.6. Output parameters for subroutine THREST

f18harv Name	Local Name	Definition
LGLGT	LGLGT	last good estimate of left gross thrust, lbs
LGA8L	LGA8L	last good estimate of left nozzle throat area, inches^2
LGLNPR	LGLNPR	last good estimate of left nozzle pressure ratio, n.d.
LGRGT	LGRGT	last good estimate of right gross thrust, lbs
LGA8R	LGA8R	last good estimate of right nozzle throat area, inches^2
LGRNPR	LGRNPR	last good estimate of right nozzle pressure ratio, n.d.
NTHRER	NTHRER	counter which accumulates number of times the thrust estimator fails its data compatibility tests.
A8LEST	A8SQL	estimated left engine nozzle throat area, inches^2
A8REST	A8SQR	estimated right engine nozzle throat area, inches^2
LGTHR	LGTHR	estimated left engine gross thrust, lbs
RGTHR	RGTHR	estimated right engine gross thrust, lbs
LNPR	LNPR	estimated left engine nozzle pressure ratio
RNPR	RNPR	estimated right engine nozzle pressure ratio

Table 4.7. Input parameters for subroutine THRCAL

f18harv Name	Local Name	Simulation Source	Definition
A8PCTL	A8PCTL	TVCENG	left engine nozzle throat area, %
A8PCTR	A8PCTR	TVCENG	right engine nozzle throat area, %
DETCOL	DETCO	THREST	left engine thrust deterioration coefficient (=0.96), n.d.
DETCOR	DETCO	THREST	right engine thrust deterioration coefficient (=0.96), n.d.
PAMB	PAMB	THREST	sensor-model static atmospheric pressure, lb/inches^2
PLAL	PLATH	ENGMDL	left engine lagged throttle position, degrees
PLAR	PLATH	ENGMDL	right engine lagged throttle position, degrees
P56L	P56L	TVCENG	left turbine discharge pressure, lbs/inches^2
P56R	P56R	TVCENG	right turbine discharge pressure, lbs/inches^2

Table 4.8. Output parameters for subroutine THRCAL

f18harv Name	Local Name	Definition
A8SQL	A8SQ	estimated left engine nozzle throat area, inches^2
A8SQR	A8SQ	estimated right engine nozzle throat area, inches^2
LNPR	NPREST	estimated left engine nozzle pressure ratio, n.d.
RNPR	NPREST	estimated right engine nozzle pressure ratio, n.d.
LGTHR	GTHEST	estimated left engine gross thrust, lbs
RGTHR	GTHEST	estimated right engine gross thrust, lbs

Table 4.9. Input parameters for subroutine MIXPRE

f18harv Name	Local Name	Simulation Source	Definition
ICALLM	ICALLM	CSL	integer switch to multiple CALL status for DMS
PJETAC	PJETAC	RFCS	thrust-vectoring pitch-jet angle command, degree
YJETAC	YJETAC	RFCS	thrust-vectoring yaw-jet angle command, degrees
LNPR	LNPR	THREST	estimated left engine nozzle pressure ratio, n.d.
RNPR	RNPR	THREST	estimated right engine nozzle pressure ratio, n.d.
A8LEST	A8L	THREST	estimated left engine nozzle throat area, inches^2
A8REST	A8R	THREST	estimated right engine nozzle throat area, inches^2
LGTHR	LGTHR	THREST	estimated left engine gross thrust, lbs
RGTHR	RGTHR	THREST	estimated right engine gross thrust, lbs
MACHS	MACH	SENSOR	Mission Computer Mach number, n.d.
PSTS	STPRES	SENSOR	sensor-model static atmospheric pressure, psf
LLN	LLN	CSL	logical switch for Jacobian calculations
LTRFLG	LTRFLG	THREST	logical switch to indicate trim status
DGTHLC(2)	DGTHLC(2)	ENGMDL	lagged left right engine throttle position, degrees
MIX4INT	FIRST	CSL	integer switch used for initialization

Table 4.10. Output parameters for subroutine MIXPRE

f18harv Name	Local Name	Definition
VANEC(6)	VANEC(6)	vane commands, degrees
LTOPVC	LTOPVC	left top vane command, inches
LOUTVC	LOUTVC	left outboard vane command, inches
LINBVC	LINBVC	left inboard vane command, inches
RTOPVC	RTOPVC	right top vane command, inches
ROUTVC	ROUTVC	right outboard vane command, inches
RINBVC	RINBVC	right inboard vane command, inches
PLTVC	PLTVC	left top vane command from previous iteration, inches
PLIVC	PLIVC	left inboard vane command from previous iteration, inches
PLOVC	PLOVC	left outboard vane command from previous iteration, inches
PRTVC	PRTVC	right top vane command from previous iteration, inches
PRIVC	PRIVC	right inboard vane command from previous iteration, inches
PROVC	PROVC	right outboard vane command from previous iteration, inches
PVECC	PVECC	thrust adjusted pitch thrust vector command, degrees
YVECC	YVECC	thrust adjusted yaw thrust vector command, degrees

Table 4.11. Input parameters for subroutine MIXER_4

f18harv Name	Local Name	Simulation Source	Definition
PJETAC	PIT_CMD	RFCS	thrust-vectoring pitch-jet-angle command, degrees
YJETAC	YAW_CMD	RFCS	thrust-vectoring yaw-jet-angle command, degrees
RJETAC	ROL_CMD	RFCS	thrust-vectoring roll-jet-angle command, degrees
R8EST	R8IN	THREST	estimated engine radius of nozzle exit, inches
ENPR	ENPR	THREST	estimated nozzle pressure ratio
GRTHEST	GROSST	THREST	estimated gross thrust, lbs
MACHS	AMCH	SENSOR	Mission Computer Mach number, n.d.
H	ALT	CSL	altitude, ft
DGTHLC	DGTHLC	ENGMDL	lagged throttle position, degrees
LLN	LLN	CSL	logical switch for Jacobian calculations
LTRFLG	LTRFLG	THREST	logical switch to indicate trim status
ICALLM	ICALLM	CSL	integer switch to multiple CALL status for DMS
MIX4INT	FIRST	INITIAL	integer switch used for initialization

Table 4.12. Output parameters for subroutine MIXER_4

f18harv Name	Local Name	Definition
VANEC(6)	VANE(6), VANEC(6)	vane commands, degrees
PVECC	PVECC	thrust adjusted pitch thrust vector command, degrees
YVECC	YVECC	thrust adjusted yaw thrust vector command, degrees
RVECC	RVECC	thrust adjusted roll thrust vector command, degrees
PJETC	PJETC	final pitch thrust vector command, degrees
YJETC	YJETC	final yaw thrsut vector command, degrees
VANECO(6)	VANEC_OLD(6)	previous frame vane commands, degrees

Table 4.13 lists the subroutines or FORTRAN functions CALLed by MIXER_4, their input and output parameters, and other subroutines/functions that they CALL.

The engine model contains subroutines INTENG and ENGMOD that interface with the ACSL code directly. One CALL to INTENG must be made before any CALLs to ENGMOD in order to initialize the engine parameters. One CALL is sufficient to initialize the engine parameters for all aircraft. The input variables for INTENG are defined in Table 4.14. The input variables for ENGDIN are the same as INTENG and are also defined in this table.

Subroutine ENGMOD receives thrustvectoring-vane commands VANEC from the M/P and TE and throttle commands PLAL and PLAR in degrees (30.73° to 130.04°). ENGMOD CALLs the appropriate engine, vane actuator, and thrust-vectoring subroutines. ENGMOD then outputs net engine forces (THX, THY, and THZ) and moments (THL, THM, and THN) in body axes for use in the equations of motion and vane-actuator-output positions (VANEP) to the *f18harv* simulation. Nozzle area and turbine discharge pressure are output as (uncorrupted) engine measurements. Gross thrust, vane positions, and effective jet-turning angles are output for information and data reduction purposes. The input/output parameters for the engine model subroutines are defined in Tables 4.15 - 4.27. ENGMOD should be CALLed once for each airplane in any run before CALLing MIX to calculate values of A8 and P56 for use by MIX. The first CALL to ENGMOD for each airplane should be made with ICALLA = 1 and with ICALLE = 1 so that the engine and actuator dynamics will be initialized correctly

(by setting the lagged throttle positions to the commanded positions, by setting the vane output positions to the commanded positions, etc.). After MIX has been CALLed the first time for each airplane, ENGMOD should be CALLed a again with ICALLA = 1 and with ICALLE = 1 to compute the initial thrusts, forces, moments, and vane positions. For subsequent CALLs to ENGMOD, ICALLA and ICALLE should be set to an integer greater than 1. Logical variables LLN and LTRFLG are used when computing Jacobians and when computing trim, respectively, in the ACSL program. Setting either of these equal to TRUE bypasses the engine and vane actuator dynamics, so both should be set to FALSE during a normal run.

Table 4.13. Input/Output and Subroutine/Function CALLs in MIXER_4

Subroutine	Input	Outputs	Calls
MIXER_INIT	-	-	-
FLIMIT	-	-	MIN*
	-	-	MAX*
F_FG	GROSST	-	NTRP1H
PITCH_LIMIT	PIT_CMD	-	FLIMIT
	YAW_CMD	-	ABS*
PITCH_PRIORITY	PIT_CMD	-	NTRP1H
	YAW_CMD	-	-
YAW_PRIORITY	PIT_CMD	-	NTRP1H
	YAW_CMD	-	ABS*
	ROL_CMD	-	MAX*
	-	-	SIGN*
F_TL	PITCH	-	SQRT*
	YAW	-	-
TV_VANES	PIT_CMD	VANE	-
	YAW_CMD	-	-
	R8IN	-	-
	ENPR	-	-
VANE_LIMIT	AMCH	-	-
	ALT	-	-
RLIMIT	VANEC_OLD	VANEC	-
	0.12857	-	-
NTRP1H	-	-	NTRP0H

* FORTRAN Functions

Table 4.14. Input parameters for subroutines INTENG and ENGDIN

f18harv Name	Local Name	Simulation Source	Definition
DTENG	DTENG	INITIAL	sample time interval at which discrete engine model is CALLed (0.0125 seconds).
PLAL	PLAL	ENGMDL	left engine total throttle position, degrees
PLAR	PLAR	TVCENG	right engine total throttle position, degrees
FSCG	CGFS	TRIM FILE	fuselage station coordinate of the c.g. inches
BLCG	CGBL	TRIM FILE	buttock line coordinate of the c.g., inches
WLCG	CGWL	TRIM FILE	water line coordinate of the c.g., inches
FSRF	FSRF	INITIAL	fuselage station coordinate of the a.r.c., inches
BLRF	BLRF	INITIAL	buttock line coordinate of the a.r.c., inches
WLRF	WLRF	INITIAL	water line coordinate of the a.r.c., inches

Table 4.15. Input parameters for subroutine ENGMOD

f18harv Name	Local Name	Simulation Source	Definition
ALF	ALP	CSL	angle of attack at the c.g., radians
MACH	AMCH	CSL	Mach number, n.d.
BIAS40	BIAS40	INITIAL	inlet stall margin increase, medium
BIAS85	BIAS85	INITIAL	inlet stall margin increase, maximum
H	H	CSL	altitude, feet
P	P	CSL	body axis roll rate, radians/sec
PLAL	PLAL	CSL	left engine commanded throttle position, degrees
PLAR	PLAR	CSL	right engine commanded throttle position, degrees
Q	Q	CSL	body axis pitch rate, radians/sec
R	R	CSL	body axis yaw rate, radians/sec
VANEC(6)	VANEC(6)	MIX	vane commands, degrees
DTENG	DTENG	CSL	sample interval at which discrete engine model is CALLed (0.0125 seconds)
LTRFLG	LTRFLG	CSL	logical switch to indicate trim status
LLN	LLN	CSL	logical switch for Jacobian calculations
ICALLA	ICALLA	CSL	integer switch to multiple CALL status for DMS
ICALLE	ICALLE	CSL	integer switch to multiple CALL status for DMS
LGYRO	LGYRO	INITIAL	logical switch to zero engine gyroscopic effects
FIRST2	FIRST2	INITIAL	integer switch used for initialization
DGTHLC(2)	DGTHLC(2)	ENGMDL	lagged left and right throttle position, degrees

Table 4.16. Output parameters for subroutine ENGMOD

f18harv Name	Local Name	Definition
A8L	A8L	estimated left engine nozzle throat area, inches^2
A8R	A8R	estimated right engine nozzle throat area, inches^2
P56L	P56L	left engine turbine discharge pressure, lbs/inches^2
P56R	P56R	right engine turbine discharge pressure, lbs/inches^2
FGTOT(2)	FGTOT(2)	gross thrust for left & right engine including T56, AOA, and nozzle effects and vane deflection losses
TVJANP(2)	TVJANP(2)	thrust vectored pitch-jet turning angle for the left & right engine in the x-y engine axis plane, degrees
TVJANY(2)	TVJANY(2)	thrust vectored yaw-jet turning angle for the left engine in the x-z engine axis plane, degrees
VANEP(6)	VANEP(6)	actual vane positions, degrees
THL	THL	total engine roll moment, lbs
THM	THM	total engine pitch moment, lbs
THN	THN	total engine yaw moment, lbs
THX	THX	total net thrust in the x-body axis, lbs
THY	THY	total net thrust in the y-body axis, lbs
THZ	THZ	total net thrust in the z-body axis, lbs
VNOT3(6)	VNOT3(6)	vane actuator lagged position from 2 iterations ago, degrees
VNOT2(6)	VNOT2(6)	vane actuator lagged position from previous iteration, degrees
VNIN3(6)	VNIN3(6)	vane actuator lagged command from 2 iterations ago, degrees
VNIN2(6)	VNIN2(6)	vane actuator lagged command from previous iteration, degrees
EFFJAN(6)	EFFJAN(6)	effective jet angle turning, degrees
GROSST(2)	GROSST(2)	engine gross thrusts including angle-of-attack effects, lbs
FGAERO(2)	FGAERO(2)	engine gross thrusts after T56 bias multiplier losses, lbs
PLAOLD(2)	PLAOLD(2)	left and right engine commanded throttle position, %
PCTHLC(2)	PCTHLC(2)	left and right engine lagged throttle position, %
RPMOLD(2)	RPMOLD(2)	left and right engine commanded rpm from previous iteration, rpm
ERPM(2)	ERPM(2)	left and right engine rpm, rpm
VANTMP(6)	VANTMP(6)	vane temperatures, degrees Fahrenheit

Table 4.17. Input parameters for subroutine TVACT

f18harv Name	Local Name	Simulation Source	Definition
VANEC(6)	VANEC(6)	MIX	vane commands, degrees
GROSST(2)	GROSST(2)	ENGMDL	left, right engine gross thrust including angle-of-attack effects, lbs
EFFJAN(6)	EFFJAN(6)	TVANGL	effective jet angle turning, degrees
DTENG	DTENG	CSL	sample interval at which discrete engine model is CALLed (0.0125 seconds)
LTRFLG	LTRFLG	CSL	logical switch to indicate trim status
LLN	LLN	CSL	logical switch for Jacobian calculations
ICALLA	ICALLA	CSL	integer switch to multiple CALL status for DMS

Table 4.18. Output parameters for subroutine TVACT

f18harv Name	Local Name	Definition
VANEP(6)	VANEP(6)	actual vane positions, degrees
VNOT3(6)	VNOT3(6)	vane actuator lagged position from 2 iterations ago, degrees
VNOT2(6)	VNOT2(6)	vane actuator lagged position from previous iteration, degrees
VNIN3(6)	VNIN3(6)	vane actuator lagged command from 2 iterations ago, degrees
VNIN2(6)	VNIN2(6)	vane actuator lagged command from previous iteration, degrees

Table 4.19. Input parameters for subroutine TVHINGE

f18harv Name	Local Name	Simulation Source	Definition
VANEC(6)	VANEC(6)	MIX	vane commands, degrees
EFFJAN(6)	EFFJAN(6)	TVANGL	effective jet angle turning, degrees
GROSST(2)	GROSST(2)	ENGMDL	left, right engine gross thrust including angle-of-attack effects, lbs

Table 4.20. Output parameters for subroutine TVHINGE

f18harv Name	Local Name	Definition
VANEP(6)	VANEP(6)	actual vane positions, degrees

Table 4.21. Input parameters for subroutine ENGMDL

f18harv Name	Local Name	Simulation Source	Definition
ALP	ALP	CSL	angle of attack at the c.g., radians
AMCH	AMCH	CSL	Mach number, n.d.
BIAS40	BIAS40	INITIAL	inlet stall margin increase, medium
BIAS85	BIAS85	INITIAL	inlet stall margin increase, maximum
H	H	CSL	altitude, feet
PLAL	PLAL	CSL	left engine commanded throttle position, degrees
PLAR	PLAR	CSL	right engine commanded throttle position, degrees
LGTHR	LGTHR	THREST	estimated left engine gross thrust, lbs
LTRFLG	LTRFLG	CSL	logical switch to indicate trim status
LLN	LLN	CSL	logical switch for Jacobian calculations
ICALLE	ICALLE	CSL	integer switch to multiple CALL status for DMS

Table 4.22. Output parameters for subroutine ENGMDL

f18harv Name	Local Name	Definition
ERPM(2)	ERPM(2)	left and right engine rpm, rpm
FGAERO(2)	FGAERO(2)	engine gross thrusts after T56 bias multiplier losses, lbs
PCTHLC(2)	PCTHLC(2)	left and right engine lagged throttle position, %
PLAOLD(2)	PLAOLD(2)	left and right engine commanded throttle position, %
RPMOLD(2)	RPMOLD(2)	left and right engine commanded rpm, rpm
GROSST(2)	GROSST(2)	engine gross thrusts including angle-of-attack effects, lbs

Table 4.23. Input parameters for subroutine TLUHRV

f18harv Name	Local Name	Simulation Source	Definition
AMCH	AMCH	DERIVATIVE	Mach number, n.d.
BIAS40	BIAS40	INITIAL	inlet stall margin increase, medium
BIAS85	BIAS85	INITIAL	inlet stall margin increase, maximum
H	H	DERIVATIVE	altitude, feet
PCTHLC(2)	PCTHLC(2)	ENGMDL	left and right engine lagged throttle position, %

Table 4.24. Input parameters for subroutine TVANGL

f18harv Name	Local Name	Simulation Source	Definition
VANEP(6)	VANEP(6)	TVHINGE	actual vane positions, degrees
PLAOLD(6)	PLAOLD(6)	INITIAL	left and right engine commanded throttle position, %
LTRFLG	TRMMING	INITIAL	logical switch to indicate trim status
FIRST2	FIRST2	INITIAL	integer switch used for initialization

Table 4.25. Output parameters for subroutine TVANGL

f18harv Name	Local Name	Definition
TVJANP(2)	TVJANP(2)	thrust vectored pitch-jet turning angle for the left & right engine in the x-y engine axis plane, degrees
TVJANY(2)	TVJANY(2)	thrust vectored yaw-jet turning angle for the left engine in the x-z engine axis plane, degrees
EFFJAN(6)	EFFJAN(6)	effective jet-angle turning, degrees
VANTMP(6)	VANTMP(6)	vane temperatures, degrees Fahrenheit

Table 4.26. Input parameters for subroutine TVCENG

f18harv Name	Local Name	Simulation Source	Definition
GROSST(2)	GROSST(2)	TVHINGE	engine gross thrusts including angle-of-attack effects, lbs
ALP	ALP	CSL	angle of attack at the c.g., radians
P	P	CSL	body axis roll rate, radians/sec
Q	Q	CSL	body axis pitch rate, radians/sec
R	R	CSL	body axis yaw rate, radians/sec
TVJANP	TVJANP	TVANGL	thrust vectored pitch-jet turning angle for the left & right engine in the x-y engine axis plane, degrees
TVJANY	TVJANY	TVANGL	thrust vectored yaw-jet turning angle for the left engine in the x-z engine axis plane, degrees
LGYRO	LGYRO	INITIAL	logical switch to zero engine gyroscopic effects

Table 4.27. Output parameters for subroutine TVCENG

f18harv Name	Local Name	Definition
A8L	A8L	estimated left engine nozzle throat area, inches^2
A8R	A8R	estimated right engine nozzle throat area, inches^2
P56L	P56L	left engine turbine discharge pressure, lbs/inches^2
P56R	P56R	right engine turbine discharge pressure, lbs/inches^2
THL	THL	total engine roll moment, lbs
THM	THM	total engine pitch moment, lbs
THN	THN	total engine yaw moment, lbs
THX	THX	total net thrust in the x-body axis, lbs
THY	THY	total net thrust in the y-body axis, lbs
THZ	THZ	total net thrust in the z-body axis, lbs

4.6 Trim

The simulation will seek a new trim solution when the logical LTR is set to TRUE. As mentioned previously the initialization CALLs to the engine, M/P, and Thrust Estimator subroutines in the INITIAL Block were placed in the trim loop following the CALL to the RFCS. Trim values for the input variables are used as actual parameters in the CALLs. Modifications to the simulation and to the engine model, M/P, and Thrust Estimator were required to enable the trim function to execute properly. When the simulation is in the trim process actively seeking a new solution, the logical LTRFLG is automatically set TRUE. This flag is then used by subroutine ENGMDL to cause both engines to be updated on every iteration rather than on alternate iterations as during a normal simulation run and to cause the engine dynamics to be bypassed by setting the lagged throttle position to the commanded throttle position on every iteration. LTRFLG is also used by subroutine TVACT to bypass the vane-actuator dynamics by setting the vane position to the vane command on every iteration. Variables PJETTR, RJETTR, and YJETTR were established as the trim values for PJETAC, RJETAC, and YJETAC so the aircraft could be trimmed and flown with thrust vectoring activated while RFCS was inactive. These variables were added to the trim file in place of the previous DTVL and DTVR. Subroutines LOAD, TRIM, XFORM, and WRTRM were modified to allow PJETTR, RJETTR, and YJETTR to be selected as driver variables in the trim procedure. After a trim solution is found, the simulation overwrites the new trim values onto the trim file if the logical LWRTR = TRUE.

4.7 Jacobians

The ACSL program has an option that allows the simulation to be executed in a mode that computes Jacobian, or system **A**, **B**, **C**, and **D**, matrices. To enable this option to work with the *f18harv* simulation engine model, M/P, and Thrust Estimator, several modifications and additions to the simulation were necessary. A decision was made to use the vectoring commands PJETAC, RJETAC, and YJETAC as the vectoring control variables rather than the vane positions VANEP as is done with the aerodynamic surfaces. The rationale for this decision is that when the M/P is in use, the control system designer is interested in **B** and **D** matrices which show the effects of the control system outputs PJETAC, RJETAC, and YJETAC on system performance. Therefore, variables PJETLN, RJETLN, and YJETLN were defined to be the perturbation variables for thrust vectoring. To avoid the ACSL error message "FUNCTION EVALUATION NOT REPEATABLE" during Jacobian evaluation, a section of code was added to the DERIVATIVE Block to CALL the engine, M/P, and Thrust Estimator subroutines and calculate the engine forces and moments only during a Jacobian simulation run. This section of code is bypassed unless the logical LLN is TRUE, and similarly the CALLs to the engine, M/P, and Thrust Estimator and the computation of engine forces and moments in the DISCRETE Block ENG are bypassed when LLN is TRUE. The error message was caused by the calculation of engine forces and moments in DISCRETE Block ENG being delayed one iteration relative to the control and state perturbations. LLN is also used by subroutine ENGMDL to update both engines every iteration rather than on alternate iterations as during a normal simulation run and to bypass the engine dynamics by setting the lagged throttle position to the commanded throttle position on every iteration. LLN is used by subroutine TVACT to bypass the vane-actuator dynamics by setting the vane position to the vane command on every iteration. In subroutines MIXPRE or MIXER_4, the vane command position and rate limits are bypassed when LLN is TRUE.

4.8 Summary of Options

Several options in the simulation have been discussed previously and will be summarized here. The flight control system can be turned "on" or "off" with the logical LFCS. When the

RFCS is "off", the thrust vectoring is controlled by varying PJETAC, RJETAC, and YJETAC. Thrust vectoring can be turned "off" using the logical LTHVEC. When the thrust vectoring is "off", the vanes are stowed at a position determined by the variable VSTOW, whose default value is -10 degrees. VSTOW can be set via an ACSL run-time command. Note that any value of LTHVEC set via a run-time command in the ACSL command file is overwritten by the value of LTHVEC in the trim file when that file is read by the simulation. The DMS throttle schedule can be selected using the logical LTHDMS. The HARV engine model and the Mixer/Predictor/Thrust Estimator have been implemented in the simulation such that trim runs can be made by setting LTR = TRUE with the RFCS either "on" or "off" and with thrust vectoring either "on" or "off". Likewise, Jacobian matrices can be computed with or without thrust vectoring by setting LLN = TRUE and using the appropriate ACSL run-time commands. The logical LMIX4 is used to select the LaRC/DFRC M/P (TRUE) or the McAir M/P (FALSE). The logical LEULER is used to select either Euler angles or quaternions as system states.

5.0 SENSOR MODEL MODIFICATIONS

The sensor-model code that exists in *f18bas* was replaced with a more representative model based on DFRC simulations. The models implemented in the *f18harv* simulation are used to simulate F/A-18 HARV aircraft systems such as the Airdata Computer (ADC), accelerometers, rate gyros, Inertial Navigation System (INS), and Mission Computer (MC). The outputs from these system models are required by the subroutines that model the RFCS, Mixer/Predictor, Thrust Estimator, and Thrust-Vectoring System. Table 5.1 lists the physical aircraft system source where the sensor output originates as well as where it is used in the *f18harv* simulation.

The *f18harv* simulation currently calculates two sets of sensor outputs. The first set is from an ideal sensor-model, which was an option preserved from the *f18bas* simulation that uses parameters calculated from the equations of motion (EOM). The second set is based on the F/A18 HARV aircraft and is the sensor model used for the RFCS designs implemented in *f18harv*. RFCS and M/P can use either sensor model, and the ideal model was preserved as a capability for future use. The integer flag ISENS determines which set of sensor parameters are sent to RFCS and M/P. A detailed description of the two sensor output sets is included in the following sections. The section 5.1 describes the ideal sensor-model calculations, and section 5.2 describes the F/A-18 HARV aircraft sensor models.

5.1 Ideal Sensor Calculations (ISENS = 0)

The sensor equations for the ideal sensor model appear in both the INITIAL and DERIVATIVE sections of the ACSL program code. Sections 5.1.1 through 5.1.8 describe the calculations which are performed when ISENS = 0.

5.1.1 Angle of Attack and Angle-of-Attack Rate

The F/A-18 HARV aircraft uses two sources for angle of attack. These sources are 1) airdata probes mounted on the wings and fuselage of the aircraft and 2) calculations performed in the Mission Computer. The ideal sensor model uses "true" angle of attack α from the EOM for both the probe and MC angle-of-attack sources. The equation below defines the angle of attack from the EOM.

$$\alpha = \arctan(w/u) \cdot (180/\pi) \tag{5.1}$$

The probe angle of attack α_{probe} has both a bias and gain applied to the EOM calculated value based on air flow. The bias is determined by the position of the landing gear, and the gain is a function of Mach number. The bias is 1.9° for the *f18harv* simulation and is intended for gear-up simulation only . The gain is 0.59 throughout the F/A-18 HARV flight envelope. The angle-of-attack-probe calculation is described below.

$$\alpha_{probe} = (\alpha - 1.9)/0.59 \tag{5.2}$$

Angle-of-attack rate $\dot{\alpha}$ is defined below.

$$\dot{\alpha} = \left(\frac{u\dot{w} - w\dot{u}}{u^2 + w^2}\right)\left(\frac{180}{\pi}\right) \tag{5.3}$$

Table 5.1. Sensor Outputs

Symbol	A/C Source	Definition	Usage
α_{probe}	ADC	ADC angle of attack, degrees	RFCS
α_{MC}	MC	MC angle of attack, degrees	RFCS
α_S	RFCS	selected angle of attack, degrees	RFCS
$\dot{\alpha}_S$	MC	MC angle-of-attack rate, degrees/sec	RFCS
β_S	MC	MC angle of sideslip, degrees	Not used
$\dot{\beta}_S$	MC	MC angle-of-sideslip rate, degrees/sec	RFCS
n_{y_S}	Accelerometer	lateral acceleration, g's	RFCS
n_{z_S}	Accelerometer	vertical acceleration, g's	RFCS
p_S	Rate Gyro	body-axis roll rate, degrees/sec	RFCS
q_S	Rate Gyro	body-axis pitch rate, degrees/sec	RFCS
r_S	Rate Gyro	body-axis yaw rate, degrees/sec	RFCS
$P_{S_{ADC}}$	ADC	ADC static pressure, psf	RFCS
$P_{S_{MC}}$	MC	MC static pressure, psf	RFCS
P_{S_S}	RFCS	selected static pressure, psf	RFCS, M/P
$Q_{C_{ADC}}$	ADC	ADC impact pressure, psf	RFCS
$Q_{C_{MC}}$	MC	MC impact pressure, psf	RFCS
Q_{C_S}	RFCS	selected impact pressure, psf	RFCS
R_{i_S}	RFCS	Q_{C_S}/P_{S_S} pressure ratio, n.d.	RFCS
θ_S	MC	pitch angle, radians	Not used directly
ϕ_S	MC	roll angle, radians	Not used directly
φ_S	MC	yaw angle, radians	Not used directly
$\cos\theta_S$	MC	cosine of pitch angle, n.d.	RFCS
$\sin\theta_S$	MC	sine of pitch angle, n.d.	RFCS
$\cos\phi_S$	MC	cosine of roll angle, n.d.	RFCS
$\sin\phi_S$	MC	sine of roll angle, n.d.	RFCS
M_S	MC	Mach number, n.d.	RFCS, M/P
H_S	MC	altitude, ft	RFCS, M/P
V_S	MC	velocity, fps	RFCS
$\bar{q}_S$	MC	dynamic pressure, psf	Not used

5.1.2 Angle of Sideslip and Angle-of-Sideslip Rate;

Angle of sideslip β and angle-of-sidelip rate $\dot{\beta}$ are calculated in the ACSL DERIVATIVE section by the following equations:

$$\beta = \arcsin(v/V) \cdot (180/\pi) \tag{5.4}$$

$$\dot{\beta} = \left(\frac{V\dot{v} - v\dot{V}}{V^2 \cos\beta}\right)\left(\frac{180}{\pi}\right) \tag{5.5}$$

5.1.3 Accelerations and Rates;

The lateral and vertical accelerations n_y and n_z, respectively, are calculated in the DERIVATIVE section by the following equations:

$$n_y = \frac{\dot{v} + ru - pw - gC_{yz}}{g} \tag{5.6}$$

$$n_z = \frac{\dot{w} + pv - qu - gC_{zz}}{g} \tag{5.7}$$

The accelerations units are g's.

Pitch, roll, and yaw rates (q, p, and r, respectively) used as ideal sensor-model rates are derived from the EOM in the DERIVATIVE section. The rates are calculated with units of radians/sec and then converted to degrees/sec to be used as the sensor-model rates.

5.1.4 Static Pressure

Ideal static pressure P_S is computed by CALLing subroutine PSTAT in the INITIAL section using initial atmoshperic altitude HRFIC and in the DERIVATIVE section using altitude HRF. Subroutine PSTAT interpolates the static pressure value from a single independent variable H. Static pressure P_S is calculated in units of lbs/ft^2. The function f_{PSTAT} implemented in subroutine PSTAT is plotted in figure 5.1.

$$P_S = f_{PSTAT}(HRF) \tag{5.8}$$

5.1.5 Impact Pressure

Ideal impact pressure Q_C is computed from the ideal static pressure P_S and Mach number M. The variable MACH is used in the main loop, and the trim variable MACHTR is used in the INITIAL section. Q_C is calculated in the DERIVATIVE section by the following equations:

IF($M \leq 1.0$) THEN

$$Q_c = \left[\left(1.0 + 0.2M^2\right)^{3.5} - 1.0\right]P_s \tag{5.9}$$

ELSE

$$Q_c = \left[\frac{166.9M^2}{\left(7.0 - 1.0/M^2\right)^{2.5} - 1.0}\right]P_s$$

ENDIF

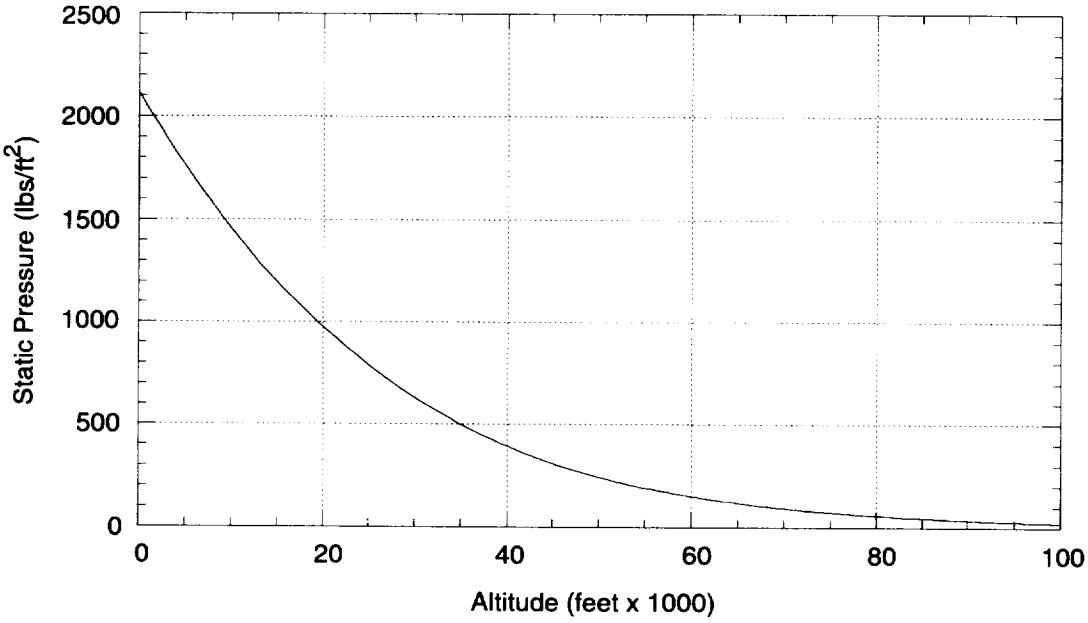


Figure 5.1. - Static Pressure as a Function of Altitude

5.1.6 Pressure Ratio

The pressure ratio R_i is defined simply as the ratio Q_c/P_s.

5.1.7 Euler Angles

The pitch θ, roll ϕ, and yaw angles φ used as ideal sensor-model outputs come directly from Euler angle integrations. The sine and cosine functions of the Euler angles are computed and used as inputs to the RFCS.

5.1.8 Mach Number, Altitude, Velocity, and Dynamic Pressure

Initial conditions for Mach number and altitude are input to the simulation through the trim file. During a simulation time history, these parameters are a result of Runge-Kutta integration. The velocity is defined by the following equation:

$$V = M \cdot V_{SOS} \tag{5.10}$$

The speed-of-sound V_{SOS} is a result of a CALL to ATMAT62 where the single independent variable H is the input. The Dynamic pressure $\bar{q}$ is calculated as follows:

$$\bar{q} = \frac{1}{2}\rho V^2 \tag{5.11}$$

The density ρ is also an output of ATMAT62 using H as the input.

Table 5.2 lists a summary of the sensor-model output definitions for ISENS = 0.

Table 5.2. Summary of Ideal Sensor Outputs Definitions (ISENS = 0)

Symbol	Definition		Units
	Symbolic	*f18harv* Names	
α_{probe}	$\alpha_{probe} = (\alpha - 1.9)/0.59$	ALFPRO = ALFDG - 1.9)/0.59	degrees
α_{MC}	$\alpha_{MC} = \alpha$	ALFINS = ALFDG	degrees
$\dot{\alpha}_s$	$\dot{\alpha}_s = \dot{\alpha}$	ADS = ALFDDG	deg/sec
β_s	$\beta_s = \beta$	BETS = BETDG	degrees
$\dot{\beta}_s$	$\dot{\beta}_s = \dot{\beta}$	BDS = BETDDG	deg/sec
n_{y_s}	$n_{y_s} = n_y$	AYS = AY	g's
n_{z_s}	$n_{z_s} = - n_z - 1$	AZS = - AZ - 1	g's
p_s	$p_s = (180/\pi)p$	PS = RA2DG·P	deg/sec
q_s	$q_s = (180/\pi)q$	QS = RA2DG·Q	deg/sec
r_s	$r_s = (180/\pi)r$	RS = RA2DG·R	deg/sec
P_{s_s}	$P_{s_s} = P_s$	PSTS = PSTATC	psf
Q_{c_s}	$Q_{c_s} = Q_c$	QCIS = QC	psf
R_{i_s}	$R_{i_s} = Q_{c_s}/P_{s_s}$	RIS = QCIS/PSTS	n.d.
θ_s	$\theta_s = \theta$	THAINS = THE	radians
ϕ_s	$\phi_s = \phi$	PHIINS = PHI	radians
φ_s	$\varphi_s = \varphi$	PSIINS = PSI	radians
$\cos\theta_s$	$\cos\theta_s = \cos\theta$	COSTHE = cos(THE)	n.d.
$\sin\theta_s$	$\sin\theta_s = \sin\theta$	SINTHE = sin(THE)	n.d.
$\cos\phi_s$	$\cos\phi_s = \cos\phi$	COSPHI = cos(PHI)	n.d.
$\sin\phi_s$	$\sin\phi_s = \sin\phi$	SINPHI = sin(PHI)	n.d.
M_s	$M_s = M$	MACHS = MACH	n.d.
H_s	$H_s = H$	HS = H	ft
V_s	$V_s = V$	VTS = VT	fps
$\bar{q}_s$	$\bar{q}_s = \bar{q}$	QBARSN = QBAR	psf

5.2 HARV Sensor Calculations (ISENS = 1)

The F/A-18 HARV aircraft sensor model implemented in the *f18harv* simulation is comprised of a collection of five subroutines CALLed in the ACSL INITIAL and DISCRETE sections. These subroutines are AIRDATA, INSMDL, ACCEL, AOAINS, and SENSOR. The following sections describe the calculations performed in these subroutines, whose outputs are passed to HARVCSM and MIX which are the control law and Mixer/Predictor interfaces, respectively.

5.2.1 Angle-of-Attack from Probe

The angle-of-attack-probe model is implemented in subroutine AIRDATA. The probe quantity is calculated first by passing angle of attack at the c.g. through the following filter representing the probe dynamics:

$$\alpha_{\text{local probe}}(s) = \left(\frac{1}{0.073s + 1}\right)\alpha(s) \tag{5.12}$$

The probe dynamics are implemented using the macro ALFDYN, which is CALLed in the DERIVATIVE section. The local probe angle of attack is then adjusted as a function of airspeed and pitch rate to correct for the upwash induced by pitch rate. The following equation defines the probe angle of attack:

$$\alpha_{\text{probe}} = \left(\alpha_{\text{local probe}} - \frac{l_p q}{V}\right)\left(\frac{180}{\pi}\right) \tag{5.13}$$

where l_p = 24 feet. The parameter l_p is defined as the distance from the center-of-gravity to the angle-of-attack probe. The final sensor output ALFPRO is then limited to the range +56 to -14 degrees. The subroutine AIRDATA is CALLed in the DISCRETE section of ACSL at a sample rate of 80 Hz. Since the actual sample rate for α_{probe} on the HARV aircraft is 40 Hz, the subroutine SENSOR updates ALFPRO at 40 Hz.

5.2.2 INS Modeling

The subroutine INSMDL is used to simulate the Inertia Navigation System (INS) and to provide inputs to the Mission Computer and sensor-model-related subroutines. The basic functions of this subroutine are defined below.

- Changes the velocities and accelerations from the reference c.g. to the INS location.
- Filters the body-axis pitch, roll, and yaw rates through a 5 Hz filter.
- Filters the accelerations at the INS location through a 2.5 Hz filter.
- Delays all output parameters to 40 Hz, which is the output rate of the INS.

The location of the INS in the body frame in feet is given by,

$$\left.\begin{aligned} x_{INS} &= \frac{-\left(FS_{INS} - FS_{rcg}\right)}{12} \\ y_{INS} &= \frac{-\left(BL_{INS} - BL_{rcg}\right)}{12} \\ z_{INS} &= \frac{-\left(WL_{INS} - WL_{rcg}\right)}{12} \end{aligned}\right\} \tag{5.14}$$

where x_{INS}, y_{INS}, z_{INS} are distances in feet forward from the reference c.g., right of the reference c.g., and below the reference c.g., respectively. The values of the terms used in equations 5.14 are defined below.

FS_{rcg}	= Reference Fuselage Station c.g.	=	458.56 inches
BL_{rcg}	= Reference Buttock Line c.g.	=	0.0 inches
WL_{rcg}	= Reference Water Line c.g.	=	100.0 inches
FS_{INS}	= FSINS, Fuselage Station INS location	=	307.95 inches
BL_{INS}	= BLINS, Buttock Line INS location	=	-13.36 inches
WL_{INS}	= WLINS, Water Line INS location	=	104.71 inches

The Body-axis velocities at the INS location are calculated as follows.

$$\left.\begin{aligned} V_{North_{INS}} &= \dot{x} + qz_{INS} - ry_{INS} \\ V_{East_{INS}} &= \dot{y} + rx_{INS} - pz_{INS} \\ V_{Vertical_{INS}} &= \dot{h} + py_{INS} - qx_{INS} \end{aligned}\right\} \tag{5.15}$$

The Body-axis velocities are converted to Earth-axis for use in the Mission Computer calculations which are described in section 5.2.3.

The accelerations at the INS location are calculated as follows in units of ft/sec^2.

$$\left.\begin{aligned} n_{x_{INS}} &= n_x g - \left(q^2 + r^2\right)x_{INS} + \left(pq - \dot{r}\right)y_{INS} + \left(pr + \dot{q}\right)z_{INS} \\ n_{y_{INS}} &= n_y g + \left(pq + \dot{r}\right)x_{INS} - \left(p^2 + r^2\right)y_{INS} + \left(qr - \dot{p}\right)z_{INS} \\ n_{z_{INS}} &= n_z g + \left(pr - \dot{q}\right)x_{INS} + \left(qr + \dot{p}\right)y_{INS} - \left(p^2 + q^2\right)z_{INS} \end{aligned}\right\} \tag{5.16}$$

The INS rates are filtered with a 2.5 Hz filter with transfer function

$$H_{p_{INS},q_{INS},r_{INS}}(s) = \frac{15.896^2}{s^2 + 2(0.707)(15.896)s + 15.896^2} \tag{5.17}$$

The INS accelerations are filtered using a 5 Hz filter with transfer function

$$H_{n_{x_{INS}}, n_{y_{INS}}, n_{z_{INS}}}(s) = \frac{34.557^2}{s^2 + 2(0.707)(34.557)s + 34.557^2} \tag{5.18}$$

The INS pitch θ_{INS}, INS roll ϕ_{INS}, and INS yaw angles φ_{INS} are set equal to θ, ϕ, and φ, respectively, from the ACSL DERIVATIVE section. The INS angles are updated at 40 Hz since the subroutine INSMDL is CALLed at 40 Hz.

5.2.3 Angle of Attack from Mission Computer

The F/A-18 HARV has an inertially derived flow-angle-measurement (angle-of-attack) algorithm implemented in the MC. The algorithm, which is implemented in subroutine AOAINS in the ACSL simulation, was developed for simulation sensitivity studies of the angle-of-attack and angle-of-sideslip-rate feedbacks at high AOA's. A flow diagram of the INS algorithm is presented in figure 5.2. The algorithm computes an angle of attack α_{MC}, angle-of-attack rate $\dot{\alpha}_{MC}$, angle of sideslip β_{MC}, and angle-of-sideslip rate $\dot{\beta}_{MC}$ from INS and ADC outputs. The MC calculates an angle of attack at all times but uses this angle of attack only when α_{probe} exceeds 30 degrees. The *f18harv* simulation uses the angle of attack from the INS algorithm when α_{probe} reaches 30 degrees. The α_{MC}, $\dot{\alpha}_{MC}$, β_{MC}, and $\dot{\beta}_{MC}$ are available as inputs to RFCS.

The inputs to subroutine AOAINS are α_{probe}, which is limited to 34 degrees, and airspeed V, an input from the EOM which is limited to 200 ft/sec. Above α_{probe} = 34 deg, MC angle of attack is computed using the integral of the $\dot{\alpha}_{MC}$, which is calculated using INS data. Therefore, the acuracy of the α_{MC} calculation is based on the accuracy of the $\dot{\alpha}_{MC}$ calculation. Body-axis velocities V_X, V_Y, and V_Z are computed from α and V assuming $\beta = 0$, and correspond to the velocities of the airmass in the x, y, and z directions ($V_{X_{airmass}}$, $V_{Y_{airmass}}$, $V_{Z_{airmass}}$). The body-axis velocities are then converted to Earth-axis velocities $V_{North_{airmass}}$, $V_{East_{airmass}}$, and $V_{Vertical_{airmass}}$ to compute INS velocities. Differences between the airmass velocities and INS velocities $V_{North_{INS}}$, $V_{East_{INS}}$, and $V_{Vertical_{INS}}$ are to the estimated winds $V_{North_{Wind}}$, $V_{East_{Wind}}$, and $V_{Vertical_{Wind}}$.

A first-order lag filter is applied to these estimated winds. If the aircraft is not maneuvering, the flow angles and hence the wind estimates are considered reliable. In this case, a filter time constant of 0.8 seconds is used, and the estimates are updated relatively quickly. On the other hand, if the aircraft is maneuvering, the estimates are not considered reliable. Then a time constant of 20 seconds is used in the filter, and the filter tends to hold the last estimate computed before the maneuvering began. The following conditions are the criteria for determining whether or not the aircraft is maneuvering:

$|p_{INS}| > 5$ deg/sec
$|q_{INS}| > 5$ deg/sec
$|r_{INS}| > 2.5$ deg/sec
$|n_{y_{INS}}| > 3.2$ ft/sec
$\alpha_{probe} > 25$ degrees
$\alpha_{probe} < -5$ degrees

The wind estimates are combined with the INS velocities to get aircraft airspeed in earth axes. After conversion of the airspeed to body-axis, angle of attack, angle of sideslip, angle-of-attack rate, and angle-of-sideslip rate are computed as follows:

$$\alpha_{MC} = \arctan\left(\frac{V_z}{V_x}\right) \tag{5.19}$$

$$\beta_{MC} = \arctan\left(\frac{V_y}{\sqrt{V_x^2 + V_z^2}}\right) \tag{5.20}$$

$$\dot{\beta}_{MC} = \frac{g\cos\theta_{INS}\sin\phi_{INS} + n_{yINS}}{V\cos\beta} + p_{INS}\sin\alpha - r_{INS}\cos\alpha \tag{5.21}$$

$$\dot{\alpha}_{MC} = \frac{-p_{INS}V\sin\beta + g\cos\theta_{INS}\cos\phi_{INS} + n_z + V\dot{\beta}\sin\alpha\sin\beta - \dot{V}\sin\alpha\cos\beta}{V\cos\alpha\cos\beta} + q_{INS} \tag{5.22}$$

The variable LUSEINS provides the user with some options in the calculation of the above equations. When LUSEINS = 1, the equations above will be used. This option simulates modeling the NASA-1A RFCS design.

When LUSEINS = 0, the INS algorithm uses α_{MC}, $\dot{\alpha}_{MC}$, β_{MC}, and $\dot{\beta}_{MC}$ directly from the EOM. This option simulates the effect of perfect angles and rates in the algorithm.

When LUSEINS = 3, α_{MC} and $\dot{\beta}_{MC}$ are modified to allow implementation of sensor-model inputs as used by the McAir/DFRC RFCS (referred to as NASA-0) and by the ANSER RFCS. The equations for α_{MC} and $\dot{\beta}_{MC}$ rate are calculated in two parts. The first part called the inertia component contains terms which are available in the MC, while the second part, called the non-inertia component, contains terms which are available in both the MC and 701E. Since the flow-angle algorithm uses INS and ADC data, the computations in the algorithm are subject to 1553 BUS delays on these inputs. To reduce the overall time delay associated with the α_{MC} and $\dot{\beta}_{MC}$ signals, the MC inertia component is calculated in the MC. The non-inertia component is calculated in RFCS using body-axis rates which are available in the 701E, effectively reducing the time delay in the computation of the non-inertia component. The rate signals available in RFCS do not have the 1553 BUS delays since these rate signals come directly from the rate gyros to the 701E. The total signal for $\dot{\alpha}_{MC}$ and $\dot{\beta}_{MC}$ is computed using these two components in RFCS . These total signals are then used as feedbacks for RFCS control laws to reduce time delay discussed above. The *f18harv* simulation does not have any models to simulate the effects of the 1553 BUS and other delays which are a function of hardware on the F/A-18 HARV. The LUSEINS = 3 option should be chosen to accurately simulate the control law when either NASA-0 or ANSER is used. The equations below define the modified $\dot{\alpha}_{MC}$ and $\dot{\beta}_{MC}$ (inertial parts) associated with this option.

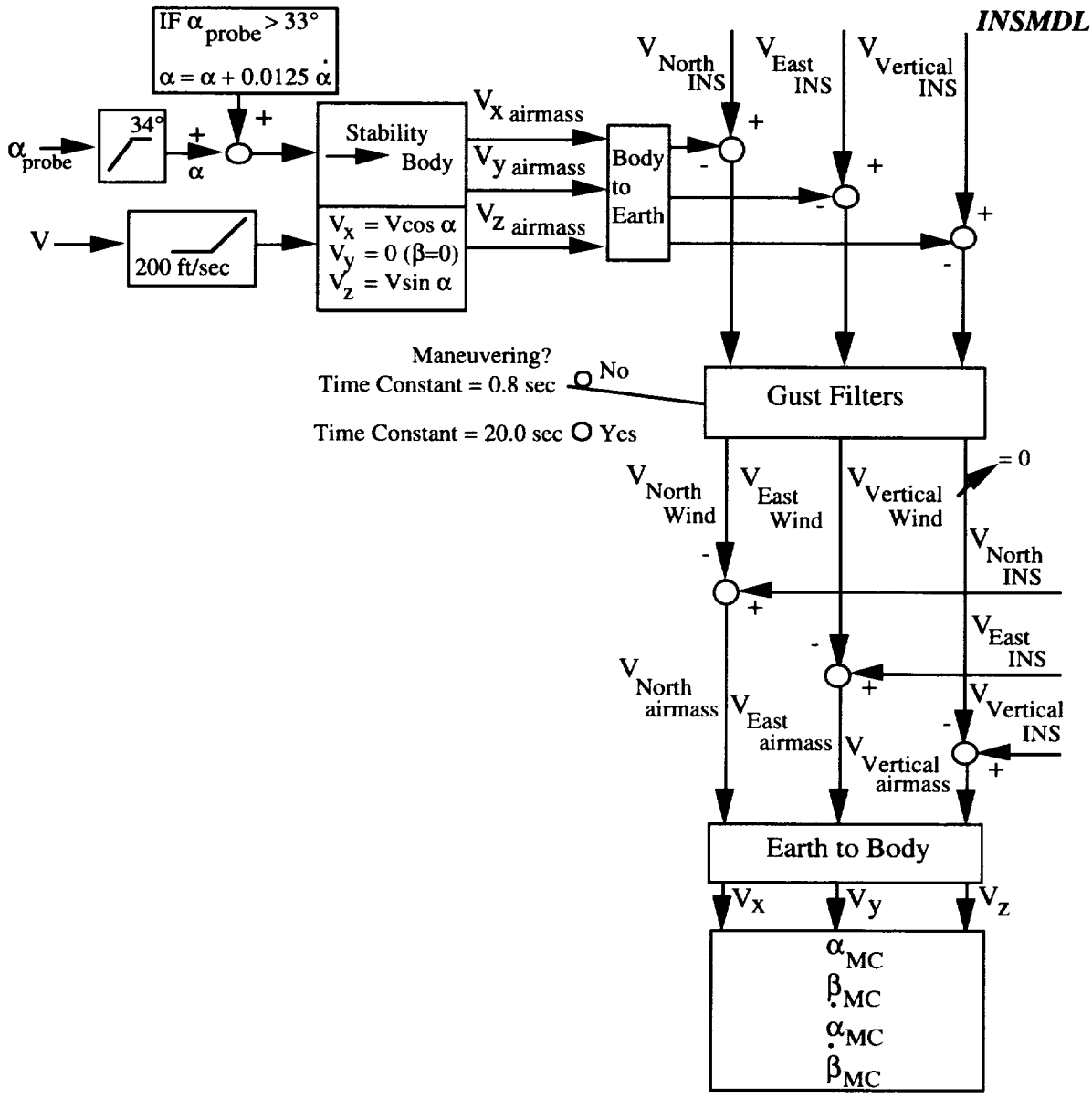


Figure 5.2. INS Algorithm Flow Diagram

$$\dot{\alpha}_{MC} = \frac{-p_{INS} V \sin\beta + g\cos\theta_{INS}\cos\phi_{INS} + n_z + V\dot{\beta}\sin\alpha\sin\beta - \dot{V}\sin\alpha\cos\beta}{V\cos\alpha\cos\beta} \tag{5.23}$$

$$\dot{\beta}_{MC} = \frac{g\cos\theta_{INS}\sin\phi_{INS} + n_{yINS}}{V\cos\beta} \tag{5.24}$$

The INS algorithm also provides the capability to perform sensitivity studies on α_{MC} and $\dot{\beta}_{MC}$ by use of a scale factor error, bias, or time delay on these calculations. The scale factor error for α_{MC} is called ALPGAIN and the bias is called ALPBIAS. The scale factor error

is multiplied by the output α_{MC} variable ADS and the bias is added to ADS. The scale factor error and bias are applied to the output $\dot{\beta}_{MC}$ variable BDS in the same manner. The scale factor error for $\dot{\beta}_{MC}$ is called BTAGAIN and the bias is called BTABIAS. Time delay can be added to both ADS and BDS through the variable NDELAY. Time frame delays on both α_{MC} and β_{MC} are added as a function of the integer value of the parameter NDELAY (default = 0). For example, if NDELAY is set to 3, three 80 Hz frames, or 37.5 ms (0.0375 seconds), will be added.

5.2.4 Selected Angle of Attack

The two sources of angle of attack, α_{probe} and α_{MC}, are sent to RFCS. RFCS makes a selection between these angle-of-attack sources based on α_{probe}. When α_{probe} exceeds 30°, the angle of attack used by RFCS is faded to the α_{MC} value. A lag filter is applied to α_{probe} before the fade to α_{MC} occurs. The output of the fade logic corresponds to the selected angle of attack α_S called ALFS in the *f18harv* simulation. The reader should review the references that present the RFCS designs for details of the angle-of-attack selection logic. The selected angle-of-attack signal is an output from RFCS and is used in the subroutine SENSOR. Selected angle of attack is used to choose either ADC or MC sensor-model impact and static pressure outputs. Since the ADC pressures are considered in error above 30 degrees angle of attack, MC pressure calculations are selected above 30 degrees AOA. The selection between these pressures occurs in RFCS on the F/A-18 HARV, while the SENSOR subroutine is used to simulate this transition in the *f18harv* simulation. Using the previous frame calculation for the selected angle of attack in SENSOR causes a one frame (0.0125 seconds) time delay associated with the pressure determination. The impact of this delay in the *f18harv* simulation time history responses is minimal.

5.2.5 Angle of Sideslip and Angle-of-Sideslip Rate

Angle of sideslip and angle-of-sideslip rate produced from the INS algorithm are discussed in detail in section 5.2.2. The angle-of-sideslip rate from this algorithm is used as a sensor input to RFCS at all times. Angle of sideslip is not used in either NASA-0 or ANSER. A 40 Hz update rate is applied in the AOAINS subroutine to model the actual sample rate of angle of sideslip on the F/A-18 HARV aircraft.

5.2.6 Accelerometers

Lateral and vertical accelerometer calculations are performed in the subroutine ACCEL. The lateral accelerometer output AYS is positive for an acceleration to the right and the vertical accelerometer output AZS is positive for an upward acceleration. Both AYS and AZS have units of g's. Lever arm effects are calculated and the 1 g bias removed from AZS. Hysteresis, bias, quantization, and misalignment are not modeled in the DFRC sensor set. The location of the accelerometers in the body frame in feet is given by

$$\left.\begin{aligned} x_a &= \frac{-\left(FS_a - FS_{cg}\right)}{12} \\ y_a &= \frac{\left(BL_a - BL_{cg}\right)}{12} \\ z_a &= \frac{-\left(WL_a - WL_{cg}\right)}{12} \end{aligned}\right\} \tag{5.25}$$

where x_a, y_a, z_a are distances in feet forward from the c.g., right of the c.g., and below the c.g., respectively. The values of the terms used in equations 5.25 are defined below.

FS_{cg}	= FSCG, Fuselage Station c.g.	= 456.3	inches
BL_{cg}	= BLCG, Buttock Line c.g.	= 0.0	inches
WL_{cg}	= WLCG, Water Line c.g.	= 105.4	inches
FS_a	= NYLOC(1), Fuselage Station accelerometer	= 306.75	inches
BL_a	= NYLOC(2), Buttock Line accelerometer	= 10.5	inches
WL_a	= NYLOC(3), Water Line accelerometer	= 91.3	inches

The c.g. locations are obtained from the trim file which contains weight and balance information in addition to initial aircraft states. The c.g. values given above correspond to a 60% internal fuel configuration for the F/A-18 HARV. The accelerometer locations can be found in the INITIAL section of the ACSL code and in subroutine ACCEL. With lever-arm effects included and 1 g bias removed the lateral and vertical accelerations are given by the equations:

$$n_{y_{ACC}} = n_y + \frac{(pq + \dot{r})x_a - \left(p^2 + r^2\right)y_a + (qr - \dot{p})z_a}{g} \tag{5.26}$$

$$n_{z_{ACC}} = -1 - n_z - \frac{(pr - \dot{q})x_a + (qr + \dot{p})y_a - \left(p^2 + q^2\right)z_a}{g} \tag{5.27}$$

. The output accelerations are bounded in the SENSOR subroutine as follows:

$$n_{y_s} = n_{y_{ACC}}\Big[^{4}_{-4} \tag{5.28}$$

$$n_{z_s} = n_{z_{ACC}}\Big[^{10}_{-10} \tag{5.29}$$

Accelerations are updated at 40 Hz to reflect the update rates on the F/A-18 HARV.

5.2.7 Rate Gyros

The outputs of the rate gyros are modeled in subroutine ACCEL. The body-axis roll, pitch, and yaw rates have aliasing filtering applied to the input signals into the 701E. The following equations represent the second-order aliasing filters applied to the body-axis rates in units of deg/sec:

$$H_{p_{ACC}}(s) = \frac{90^2}{s^2 + 2(0.8)90s + 90^2} \tag{5.30}$$

$$H_{q_{ACC}}(s) = \frac{200^2}{s^2 + 2(0.89)200s + 200^2} \tag{5.31}$$

$$H_{r_{ACC}}(s) = \frac{200^2}{s^2 + 2(0.89)200s + 200^2} \tag{5.32}$$

The subroutine ACCEL outputs are XPGY, XQGY, and XRGY which correspond to p_{ACC}, q_{ACC}, and r_{ACC}, respectively. These rates are then limited as follows in subroutine SENSOR:

$$p_s = \left(\frac{180}{\pi}\right) p_{ACC} \Big[{}^{300}_{-300} \text{ deg/sec} \tag{5.33}$$

$$\begin{aligned}
&\text{IF}\left(|q_{ACC}| < 55\right)\text{THEN} \\
&\quad q_s = \left(\frac{180}{\pi}\right) q_{ACC} \Big[{}^{60}_{-60} \text{ deg/sec} \\
&\text{ELSE} \\
&\quad q_s = \left(\frac{180}{\pi}\right) q_{ACC} \Big[{}^{100}_{-100} \text{ deg/sec} \\
&\text{ENDIF}
\end{aligned} \tag{5.34}$$

$$\begin{aligned}
&\text{IF}\left(|r_{ACC}| < 55\right)\text{THEN} \\
&\quad r_s = \left(\frac{180}{\pi}\right) r_{ACC} \Big[{}^{60}_{-60} \text{ deg/sec} \\
&\text{ELSE} \\
&\quad r_s = \left(\frac{180}{\pi}\right) r_{ACC} \Big[{}^{100}_{-100} \text{ deg/sec} \\
&\text{ENDIF}
\end{aligned} \tag{5.35}$$

All of the rate-gyro signals sent to the 701E are limited to ± 60 deg/sec. The F/A-18 HARV limits for pitch rate and yaw rate were increased to ± 100 deg/sec. The F/A-18 HARV has two separate rate-gyro-input signals to RFCS. When the absolute value of the rate-gyro signal in RFCS reaches 55 deg/sec, the rate-gyro signals are switched from the ± 60 deg/sec-limited signal to the ± 100 deg/sec-limited signal. The yaw-rate-gyro signal which corresponds to the sensor-model yaw rate in the subroutine SENSOR is updated at 40 Hz to reflect the update rate on the F/A-18 HARV.

5.2.8 Static Pressure

Two sources of static pressure determine which sensor-model value is used in the *f18harv* simulation. Static pressure is calculated from either the AIRDATA Computer (ADC) or the Mission Computer (MC) source. The value of the selected angle-of-attack signal determines which static-pressure calculation is used. This determination is made in RFCS and physically corresponds to using either the ADC or MC value on the F/A-18 HARV.

The ADC static pressure $P_{s_{ADC}}$ is calculated in the subroutine AIRDATA and is called PSTS1. The AIRDATA subroutine calculates an indicated Mach number based on true Mach number and an initial estimate of Mach error. Then pressure ratio is calculated from look-up tables as a function of only Mach number and used to calculate coefficients for the ADC calibrations, which in turn are used to compute ADC static and impact pressure.

The MC is the second source for static pressure $P_{s_{MC}}$, which is derived in the subroutine AOAINS using the following equations:

$$\begin{aligned}
&\text{IF}(H \geq 36089.)\ \text{THEN} \\
&\quad P_{s_{MC}} = \frac{6.6823e^{(36089-H)}}{20805.8} \\
&\text{ELSE} \\
&\quad P_{s_{MC}} = 29.99213\left[\left(1-\frac{H}{145442}\right)^{5.2559}\right] \\
&\text{ENDIF}
\end{aligned} \tag{5.36}$$

The output variable for static pressure from AOAINS is called XPS and has units of inches Hg. The following calculation in the subroutine SENSOR determines the sensor-model static pressure.

$$\begin{aligned}
&\text{IF}(\alpha < 30)\ \text{THEN} \\
&\quad P_{s_s} = P_{s_{ADC}} \\
&\text{ELSE} \\
&\quad P_{s_s} = 70.726912\,P_{s_{MC}} \\
&\text{ENDIF}
\end{aligned} \tag{5.37}$$

Both $P_{s_{ADC}}$ and $P_{s_{MC}}$ are computed at 80 Hz in subroutines AIRDATA and AOAINS, respectively. The variable P_{s_s} is updated at 20 Hz in the SENSOR subroutine to reflect the update rate on the F/A-18 HARV.

5.2.9 Impact Pressure

Like static pressure, impact pressure is calculated from either the ADC or MC source in either subroutine AIRDATA or AOAINS, respectively. The ADC impact pressure $Q_{c_{ADC}}$ is computed in subroutine AIRDATA from the pressure ratio used to calculate coefficients for the ADC calibrations. The output variable for impact pressure from subroutine AIRDATA is QCIS1 with units of psf.

The MC impact pressure $Q_{c_{MC}}$ represents the second source for impact pressure and is calculated in subroutine AOAINS. Also in subroutine AOAINS, an airspeed based on the x, y, and z body-axis velocities, is computed from the following equation:

$$V_{MC} = \sqrt{V_x^2 + V_y^2 + V_z^2} \tag{5.38}$$

It has a lower limit of 50 fps. Another input to subroutine AOAINS is temperature in degrees Kelvin, °K. This temperature converted to degrees Rankin, °R, has a lower limit of 408°R and is used as follows:

$$V_{SOS_{MC}} = 49.021\sqrt{|R°|} \tag{5.39}$$

to calculate the speed of sound $V_{SOS_{MC}}$. Mission Computer Mach number is airspeed divided by the speed of sound as follows:

$$M_{MC} = V_{MC}/V_{SOS_{MC}} \tag{5.40}$$

MC dynamic pressure is calculate using Mach number and static pressure in the following equations:

$$\begin{aligned}
&\text{IF}(M_{MC} \le 1.0)\ \text{THEN} \\
&\quad Q_{c_{MC}} = \left(\left(1.0 + 0.2M_{MC}^2\right)^{3.5} - 1.0\right)P_{s_{MC}} \\
&\text{ELSE} \\
&\quad Q_{c_{MC}} = \left(\frac{166.9M_{MC}^2}{\left(7 - \frac{1.0}{M_{MC}^2}\right)^{2.5} - 1.0}\right)P_{s_{MC}} \\
&\text{ENDIF}
\end{aligned} \tag{5.41}$$

$$\text{IF}\left(M_{MC} \le 0.0\right)M_{MC} = 0.08$$

The output variable for impact pressure from AOAINS is called XQC, and has units of inches Hg. The value of the selected angle of attack determines which impact pressure calculation is used. The following calculation in the subroutine SENSOR determines the sensor-model impact pressure:

$$\begin{aligned}
&\text{IF}(\alpha_s < 30)\ \text{THEN} \\
&\quad Q_{c_S} = Q_{c_{ADC}} \\
&\text{ELSE} \\
&\quad Q_{c_S} = 70.726912Q_{c_{MC}} \\
&\text{ENDIF}
\end{aligned} \tag{5.42}$$

Both $Q_{c_{ADC}}$ and $Q_{c_{MC}}$ are computed at 80 Hz in subroutines AIRDATA and AOAINS, respectively. In the SENSOR subroutine Q_{c_S} is updated at 20 Hz to reflect the update rate on the F/A-18 HARV.

The sensor-model impact pressure Q_{c_S} is filtered both with a 2.5 radian/sec first-order lag filter and a 10.0 radian/sec first-order lag filter. These filtered signals, known as QCFILT1 and QCFILT2, respectively, are used as inputs to the ANSER RFCS. The filtering of impact pressure was a result of flight testing of the NASA-1A RFCS, which showed significant noise on the impact pressure input signal.

5.2.10 Pressure Ratio

The pressure ratio R_{i_S} is defined simply as the ratio Q_{c_S}/P_{s_S}.

5.2.11 Euler Angles

The pitch, roll, and yaw Euler angles from the EOM are set equal to THAINS, PHIINS, and PSIINS, respectively, and are computed in subroutine INSMDL. Since the subroutine INSMDL is CALLed in the DISCRETE section at 40 Hz, these angles are updated and available at 40 Hz. The sine and cosine functions are approximated by a power series as follows:

$$\cos\theta_s = 1.0 - \frac{\theta_s^2}{2!} + \frac{\theta_s^4}{4!} - \frac{\theta_s^6}{6!} \qquad \text{for } -90 \le \theta_S < 90° \qquad (5.43)$$

$$\sin\theta_s = \theta_s - \frac{\theta_s^3}{3!} + \frac{\theta_s^5}{5!} - \frac{\theta_s^7}{7!} + \frac{\theta_s^9}{9!} \qquad \text{for } -90 \le \theta_S < 90° \qquad (5.44)$$

$$\cos\phi_s = 1.0 - \frac{\phi_s^2}{2!} + \frac{\phi_s^4}{4!} - \frac{\phi_s^6}{6!} + \frac{\phi_s^8}{8!} \qquad \text{for } -180 \le \phi_S < 180° \qquad (5.45)$$

$$\sin\phi_s = \phi_s - \frac{\phi_s^3}{3!} + \frac{\phi_s^5}{5!} - \frac{\phi_s^7}{7!} + \frac{\phi_s^9}{9!} \qquad \text{for } -180 \le \phi_S < 180° \qquad (5.46)$$

These calculations are computed at 20 Hz in subroutine SENSOR to model the update rate of the F/A-18 HARV aircraft.

The subroutine SENSOR contains additional calculations to compute sensor-model wind-axis Euler angles. The Euler-angle calculations were added for other advanced RFCS designs, namely, NASA-2, and are utilized in the NASA-2 RFCS DMS implementation design. These addtional equations follow:

$$\gamma_s = \arcsin\{\cos\alpha_s \cos\beta_s \sin\theta_s - \cos\theta_s \sin\beta_s \sin\phi_s + \sin\alpha_s \cos\beta_s \cos\phi_s\} \qquad (5.47)$$

$$\mu_S = \arctan\left\{\frac{\cos\alpha_S \sin\beta_S \sin\theta_S + \cos\theta_S \cos\beta_S \sin\phi_S - \sin\alpha_S \sin\beta_S \cos\phi_S}{\sin\alpha_S \sin\theta_S + \cos\alpha_S \cos\phi_S \cos\theta_S}\right\} \tag{5.48}$$

$$\chi_S = \arctan\left\{\frac{\left[\begin{array}{l}\cos\alpha_S \cos\beta_S \cos\theta_S \sin\phi_S + \sin\beta_S(\sin\phi_S \sin\theta_S \sin\psi_S + \cos\phi_S \cos\psi_S) \\ + \sin\alpha_S \cos\beta_S(\cos\phi_S \sin\theta_S \sin\psi_S - \sin\phi_S \cos\psi_S)\end{array}\right]}{\left[\begin{array}{l}\cos\alpha_S \cos\beta_S \cos\theta_S \cos\phi_S + \sin\beta_S(\sin\phi_S \sin\theta_S \cos\psi_S - \cos\phi_S \sin\psi_S) \\ + \sin\alpha_S \cos\beta_S(\cos\phi_S \sin\theta_S \cos\psi_S + \sin\phi_S \sin\psi_S)\end{array}\right]}\right\} \tag{5.49}$$

5.2.12 Mach Number, Altitude, Velocity, and Dynamic Pressure

The sensor-model Mach number M_S is computed from the pressure-ratio computation in the subroutine SENSOR. The following equations define sensor Mach number, which is updated at 20 Hz.

$$\begin{array}{l}\text{IF } (R_{i_S} \le 0.018) \text{ THEN} \\ \quad M_S = 0.16 \\ \text{ELSEIF } (R_{i_S} \le 0.278) \text{ THEN} \\ \quad M_S = 0.08510 + 4.9102R_{i_s} - 32.967R_{i_s}^2 \\ \qquad\qquad + 168.96R_{i_s}^3 - 465.41R_{i_s}^4 + 513.56R_{i_s}^5 \\ \text{ELSE} \\ \quad M_S = 0.32293 + 1.1798R_{i_s} - 0.65244R_{i_s}^2 \\ \qquad\qquad + 0.23755R_{i_s}^3 - 0.043113R_{i_s}^4 + 0.003024R_{i_s}^5 \\ \text{ENDIF}\end{array} \tag{5.50}$$

M_S is then limited as follows:

$$\text{IF } (M_S > 2.0)\ M_S = 2.0.$$

Sensor altitude used in the *f18harv* simulation corresponds to altitude H computed in the DERIVATIVE section of ACSL. An additional computation for sensor altitude is made in subroutine SENSOR. This altitude calculation is a function of static pressure P_{S_S} only and is used in the lateral/directional RFCS and M/P.

$$H_S = 70763.0 + 1.0522E-08P_{s_s}^4 - 5.1458E-05P_{s_s}^3 + 9.91E-02P_{s_s}^2 - 109.534P_{s_s} \tag{5.51}$$

The sensor-model airspeed V_S is computed in subroutine SENSOR as a function of M_S and P_{S_S} as follows:

$$V_S = M_S(897.3145 + 0.17518P_{s_s} - 0.00003404P_{s_s}^2) \tag{5.52}$$

Airspeed is updated at a rate of 20 Hz and has a lower limit of 1 fps.

The dynamic pressure $\bar{q}_S$ is also calculated in SENSOR using the following equation:

$$\bar{q}_S = \frac{Q_{c_S}}{1.0 + 0.25M_S^2 + 0.025M_S^4} \tag{5.53}$$

The dynamic pressure has a lower limit of 40 psf and an update rate of 20 Hz.

Figure 5.3 presents a block diagram summary of the sensor-model outputs with ISENS = 1. Both the symbol and *f18harv* simulation variable names are used along with the subroutine source and units.

Table 5.3 lists a summary of the sensor-model outputs for ISENS = 1.

Table 5.3. F/A-18 HARV Sensor-Model Outputs for (ISENS = 1)

Symbol	*f18harv* Name	Definition
α_{probe}	ALFPRO	ADC angle of attack, degrees
α_{MC}	ALFINS	MC angle of attack, degrees
α_S	ALFS	selected angle of attack, degrees
$\dot{\alpha}_S$	ADS	sensor angle-of-attack rate, degrees/sec
β_S	BETS	sensor angle of sideslip, degrees
$\dot{\beta}_S$	BDS	sensor angle-of-sideslip rate, degrees/sec
n_{y_S}	AYS	sensor lateral acceleration, g's
n_{z_S}	AZS	sensor vertical acceleration, g's
p_S	PS	sensor body axis roll rate, degrees/sec
q_S	QS	sensor body axis pitch rate, degrees/sec
r_S	RS	sensor body axis yaw rate, degrees/sec
$P_{s_{ADC}}$	PSTS1	ADC static pressure, psf
$P_{s_{MC}}$	XPS	MC static pressure, psf
P_{s_S}	PSTS	selected static pressure, psf
$Q_{c_{ADC}}$	QCIS1	ADC impact pressure, psf
$Q_{c_{MC}}$	XQC	MC impact pressure, psf
Q_{c_S}	QCIS	selected impact pressure, psf
R_{i_S}	RIS	sensor pressure ratio, n.d.
θ_{INS}	THAINS	INS pitch angle, radians
ϕ_{INS}	PHIINS	INS roll angle, radians
φ_{INS}	PSIINS	INS yaw angle, radians
$\cos\theta_S$	COSTHE	sensor cosine of pitch angle, n.d.
$\sin\theta_S$	SINTHE	sensor sine of pitch angle, n.d.
$\cos\phi_S$	COSPHI	sensor cosine of roll angle, n.d.

Table 5.3. Concluded

Symbol	*f18harv* Name	Definition
$\sin \phi_S$	SINPHI	sensor sine of roll angle, n.d.
M_S	MACHS	sensor Mach number, n.d.
H_S	ALTS	sensor altitude, ft
V_S	VTS	sensor velocity, ft/sec
$\bar{q}_S$	QBARSN	sensor dynamic pressure, psf

5.3 Implementation

Tables 5.4 through 5.13 list the input and output parameters for subroutines AIRDATA, INSMDL, ACCEL, AOAINS, and SENSOR.

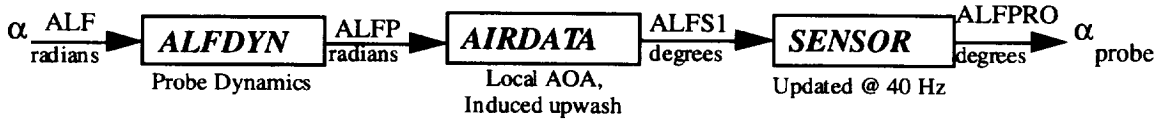
Table 5.4. Input parameters for subroutine AIRDATA

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
$\alpha_{\text{local probe}}$	ALFP	ALP	CSL	local probe angle of attack, radians
M	MACH	AMCH	CSL	Mach number, n.d.
-	DLGPCT	DNGEAR	CSL	gear position (0 = up, 1 = down)
q	Q	Q	CSL	body axis pitch rate, rad/sec
V	VT	V	CSL	aircraft total atmosphere velocity, ft/sec
-	PSTATC	PA	CSL	static atmospheric pressure, = f(HRF), psf

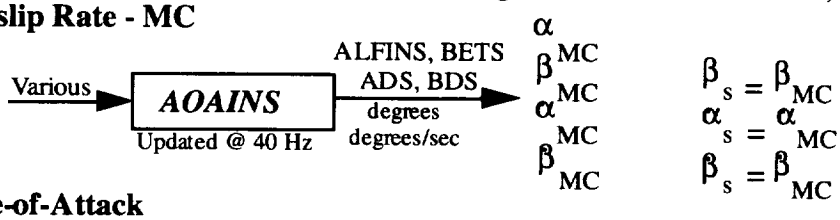
Table 5.5. Output parameters for subroutine AIRDATA

Symbol	*f18harv* Name	Local Name	Definition
α_{probe}	ALFS1	ALPHAISM	airdata probe angle of attack without bias or gain applied, degrees
$P_{S_{ADC}}$	PSTS1	PAI	airdata static pressure, psf
$Q_{C_{ADC}}$	QCIS1	QCI	airdata impact pressure, psf
-	PTIS1	PTI	intermediate variable in pressure computation, psf
$R_{i_{ADC}}$	RIS1	RI	pressure ratio, n.d.

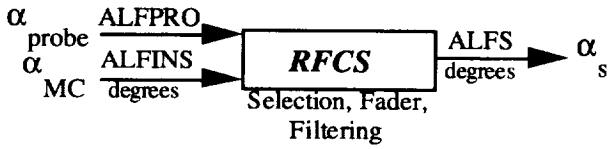
Angle-of-Attack - Probe



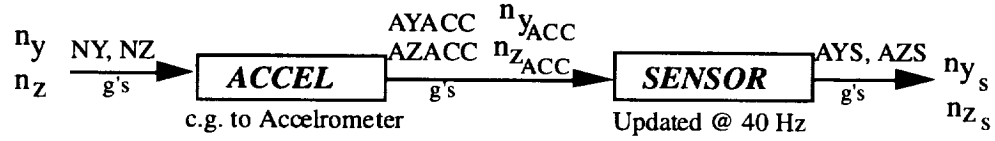
Angle-of-Attack - MC, Angle-of-Sideslip - MC, Angle-of-Attack Rate - MC, Angle-of-Sideslip Rate - MC



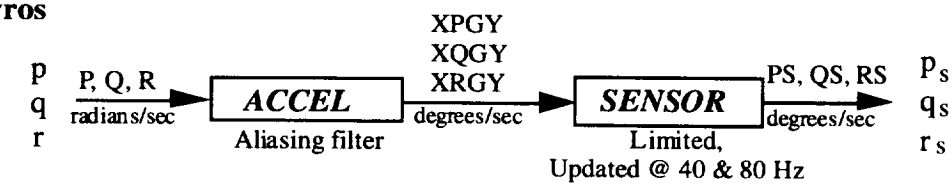
Selected Angle-of-Attack



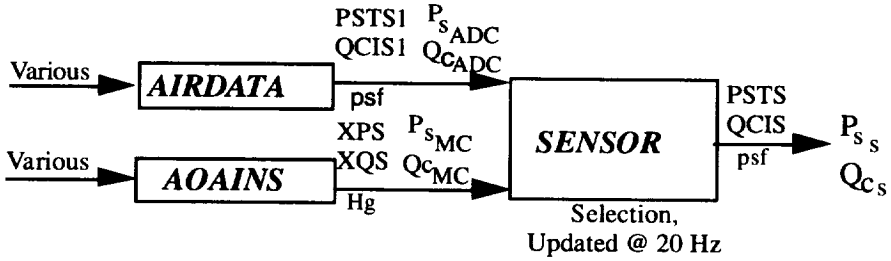
Accelerometers



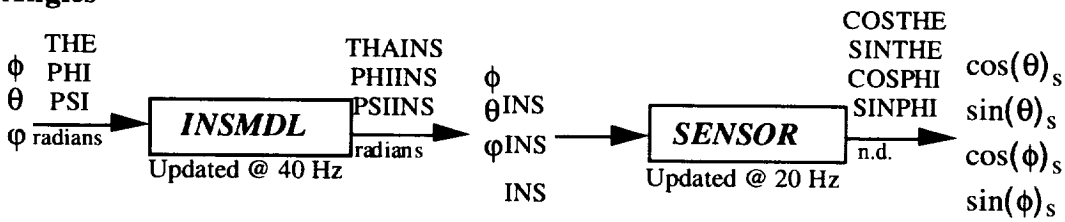
Rate Gyros



Selected Static and Impact Pressure



Euler Angles



Pressure Ratio, Mach Number, Altitude, Velocity, Dynamic Pressure

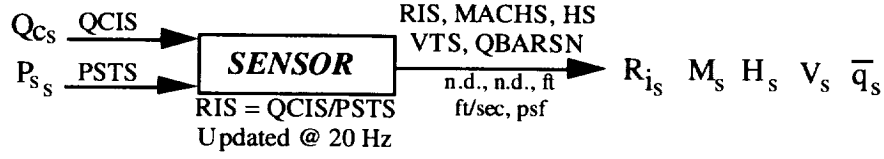


Figure 5.3. F/A-18 HARV Sensor Model Summary (ISENS = 1)

Table 5.6. Input parameters for subroutine INSMDL

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
$\dot{x}$	XD	XDOT	CSL	reference c.g. velocity (North), ft/sec
$\dot{y}$	YD	YDOT	CSL	reference c.g. velocity (East), ft/sec
$\dot{h}$	HD	HDOT	CSL	reference c.g. velocity (Vertical), ft/sec
q	Q	Q	CSL	body axis pitch rate, rad/sec
p	P	P	CSL	body axis roll rate, rad/sec
r	R	R	CSL	body axis yaw rate, rad/sec
θ	THE	THA	CSL	Euler pitch angle, radians
ϕ	PHI	PHI	CSL	Euler roll angle, radians
ψ	PSI	PSI	CSL	Euler yaw angle, radians
n_y	AY	ANY	CSL	lateral acceleration, g's
$\dot{r}$	RD	RD	CSL	body axis yaw acceleration, rad/sec^2
$\dot{p}$	PD	PD	CSL	body axis roll acceleration, rad/sec^2
BL_{cg}	BLCG	BLCG	CSL	buttock line coordinate of the center of gravity, inches
FS_{cg}	FSCG	FSCG	CSL	fuselage station coordinate of the center of gravity, inches
WL_{cg}	WLCG	WLCG	CSL	water line coordinate of the center of gravity, inches
n_x	AX	AX	CSL	axial acceleration, g's
n_z	AZ	AZ	CSL	longitudinal acceleration, g's
$\dot{q}$	QD	QD	CSL	body axis pitch acceleration, rad/sec^2

Table 5.7. Output parameters for subroutine INSMDL

Symbol	*f18harv* Name	Local Name	Definition
$V_{North\,INS}$	VNINS	VNINS	INS velocity (North), ft/sec
$V_{East\,INS}$	VEINS	VEINS	INS velocity (East), ft/sec
$V_{Vertical\,INS}$	VVINS	VVINS	INS velocity (Vertical), ft/sec
q_{INS}	QINS	QINS	INS pitch rate, rad/sec
p_{INS}	PINS	PINS	INS roll rate, rad/sec
r_{INS}	RINS	RINS	INS yaw rate, rad/sec
$n_{y_{INS}}$	RNYINS	NYINS	INS y-acceleration, ft/sec^2
θ_{INS}	THAINS	THETAINS	INS Euler pitch angle, radians
ϕ_{INS}	PHIINS	PHIINS	INS Euler roll angle, radians
ψ_{INS}	PSIINS	PSIINS	INS Euler yaw angle, radians
$n_{x_{INS}}$	RNXINS	NXINS	INS x-acceleration, ft/sec^2
$n_{z_{INS}}$	RNZINS	NZINS	INS z-acceleration, ft/sec^2

Table 5.8. Input parameters for subroutine ACCEL

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
n_y	AY	AY	CSL	lateral acceleration, g's
n_z	AZ	AZ	CSL	longitudinal acceleration, g's
p	P	P	CSL	body axis roll rate, rad/sec
q	Q	Q	CSL	body axis pitch rate, rad/sec
r	R	R	CSL	body axis yaw rate, rad/sec
$\dot{p}$	PD	PD	CSL	body axis roll acceleration, rad/sec^2
$\dot{q}$	QD	QD	CSL	body axis pitch acceleration, rad/sec^2
$\dot{r}$	RD	RD	CSL	body axis yaw acceleration, rad/sec^2
BL_{cg}	BLCG	BLCG	CSL	buttock line coordinate of the center of gravity, inches
FS_{cg}	FSCG	FSCG	CSL	fuselage station coordinate of the center of gravity, inches
WL_{cg}	WLCG	WLCG	CSL	water line coordinate of the center of gravity, inches

Table 5.9. Output parameters for subroutine ACCEL

Symbol	*f18harv* Name	Local Name	Definition
$n_{y_{ACC}}$	AYACC	AYACC	lateral acceleration in y-body direction at the accelerometer location, g's
$n_{z_{ACC}}$	AZACC	AZACC	normal acceleration in z-body direction at the acclerometer location, g's
p_{ACC}	XQGY	XQGY	body axis pitch rate after aliasing filter, degrees/sec
q_{ACC}	XPGY	XPGY	body axis roll rate after aliasing filter, degrees/sec
r_{ACC}	XRGY	XRGY	body axis yaw rate after aliasing filter, degrees/sec
-	AF2IN	AF2IN	3 element past state array of inputs for pitch rate aliasing filter, degrees/sec
-	AF2OT	AF2OT	3 element past state array of outputs for pitch rate aliasing filter, degrees/sec
-	AF5IN	AF5IN	3 element past state array of input values for roll rate aliasing filter, degrees/sec
-	AF5OT	AF5OT	3 element past state array of output values for roll rate aliasing filter, degrees/sec
-	AF6IN	AF6IN	3 element past state array of input values for yaw rate, aliasing filter degrees/sec
-	AF6OT	AF6OT	3 element past state array of output values for yaw rate, aliasing filter degrees/sec

Table 5.10. Input parameters for subroutine AOAINS

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
-	ALPBIAS	ALPBIAS	INITIAL	bias added to MC angle of attack, deg
-	ALPGAIN	ALPGAIN	INITIAL	gain multiplied to MC angle of attack, n.d.
-	BTABIAS	BTABIAS	INITIAL	bias added to MC sideslip angle, deg
-	BTAGAIN	BTAGAIN	INITIAL	gain multiplied to MC sideslip angle, n.d.
-	ND80HZ	ND80HZ	INITIAL	number of frames delay
α	ALFDG	ALFDG	CSL	angle of attack, degrees
V	VT	VT	CSL	aircraft total atmosphere velocity, ft/sec
T	TMPK	TMPK	CSL	temperature, degrees Kelvin
H	H	H	CSL	altitude, ft
p_{INS}	PINS	PINS	INSMDL	INS roll rate, rad/sec
q_{INS}	QINS	QINS	INSMDL	INS pitch rate, rad/sec
r_{INS}	RINS	RINS	INSMDL	INS yaw rate, rad/sec
ψ_{INS}	PSIINS	PSIINS	INSMDL	INS Euler yaw angle, radians
θ_{INS}	THAINS	THAINS	INSMDL	INS Euler pitch angle, radians
ϕ_{INS}	PHIINS	PHIINS	INSMDL	INS Euler roll angle, radians
$n_{y_{INS}}$	RNYINS	RNYINS	INSMDL	INS y-acceleration, ft/sec^2
$V_{North\,INS}$	VNINS	VNINS	INSMDL	INS velocity (North), ft/sec
$V_{East\,INS}$	VEINS	VEINS	INSMDL	INS velocity (East), ft/sec
$V_{Vertical\,INS}$	VVINS	VVINS	INSMDL	INS velocity (Vertical), ft/sec
_	LTRFLG	TRMMING	CSL	logical switch to indicate trim status
$\dot{\beta}$	BETDDG	BETDDG	CSL	d/dt (sideslip angle), deg/sec
$\alpha_{local\ probe}$	ALFP	ALFP	CSL	local probe angle of attack, radians
β	BETDG	BETDG	CSL	sideslip angle at the c.g., degrees
-	LUSEINS	LUSEINS	INITIAL	Integer to determine INS flow angle computations
-	FIRST	FIRST	INITIAL	integer switch used for initialization
n_x	AX	AX	CSL	axial acceleration, g's
n_z	AZ	AZ	CSL	longitudinal acceleration, g's
$\dot{\alpha}$	ALFDDG	ALFDDG	CSL	d/dt (angle of attack), degrees/sec

Table 5.11. Output parameters for subroutine AOAINS

Symbol	*f18harv* Name	Local Name	Definition
-	WNDN	WNDN	wind vector (North), ft/sec
-	WNDE	WNDE	wind vector (East), ft/sec
-	WNDV	WNDV	wind vector (Vertical), ft/sec
α_{MC}	ALFINS	ALFINS	Mission Computer calculated angle of attack, degrees
β_{MC}	BETS	BETS	Mission Computer sideslip angle, degrees
$\dot{\beta}_{MC}$	BDS	BDS	Mission Computer angle-of-sideslip rate, degrees

Table 5.11. Concluded

Symbol	*f18harv* Name	Local Name	Definition
-	IWND	IWND	20 Hz loop flag for gust filter
-	XX(20)	XX(20)	previous frame values for MC angle of attack, deg
-	YY(20)	YY(20)	previous frame values for MC sideslip angle, deg
-	LOOP20	LOOP20	20 Hz loop flag
$P_{s_{MC}}$	XPS	XPS	Mission Computer static pressure, inches HG
$Q_{c_{MC}}$	XQC	XQC	Mission Computer impact pressure, inches HG
-	LOOP40	LOOP40	40 Hz loop flag
$\dot{\alpha}_{MC}$	ADS	ADS	Mission Computer angle-of-attack rate, degrees/sec
-	XHAOA	XHAOA	limited angle of attack, degrees
-	XHTAS	XHTAS	limited aircraft total atmosphere velocity, ft/sec
-	XTMPR	XTMPR	temperature, degrees Rankin
-	XDPALT	XDPALT	Mission Computer altitude, ft

Table 5.12. Input parameters for subroutine SENSOR

Symbol	*f18harv* Name	Local Name	Simulation Source	Definition
α_{MC}	ALFINS	XALPMC	AOAINS	Mission Computer angle of attack, degrees
α_{probe}	ALFS1	ALFS1	AIRDATA	airdata probe angle of attack without bias or gain applied, degrees
$n_{y_{ACC}}$	AYACC	AYACC	ACCEL	lateral acceleration in y-body direction at the accelerometer location, g's
$n_{z_{ACC}}$	AZACC	AZACC	ACCEL	normal acceleration in z-body direction at the acclerometer location, g's
p_{ACC}	XQGY	XQGY	ACCEL	body axis pitch rate, deg/sec
q_{ACC}	XPGY	XPGY	ACCEL	body axis roll rate, deg/sec
r_{ACC}	XRGY	XRGY	ACCEL	body axis yaw rate after, deg/sec
$P_{s_{ADC}}$	PSTS1	PAI	AIRDATA	airdata static pressure, psf
$Q_{c_{ADC}}$	QCIS1	QCI	AIRDATA	airdata impact pressure, psf
$P_{s_{MC}}$	XPS	XPS	AOAINS	Mission Computer static pressure, inches Hg
$Q_{c_{MC}}$	XQC	XQC	AOAINS	Mission Computer impact pressure, inches Hg
-	ISENS	ISENS	INITIAL	integer switch for sensor model
-	LTRFLG	TRMMING	CSL	logical switch to indicate trim status
-	LLN	LLN	INITIAL	logical switch for Jacobians
-	FIRST	FIRST	INITIAL	integer switch used for initialization
θ	THEDG	XTHETADG	CSL	Euler pitch angle, degrees
ϕ	PHIDG	XPHIDG	CSL	Euler roll angle, degrees
ψ	PSIDG	XPSIDG	CSL	Euler yaw angle, degrees
α	ALFDG	ALFDG	CSL	angle of attack, degrees
β_{MC}	BETS	BETAMC	CSL	Mission Computer sideslip angle, degrees
θ	THE	THA	CSL	Euler pitch angle, radians
ϕ	PHI	PHI	CSL	Euler roll angle, radians
ψ	PSI	PSI	CSL	Euler yaw angle, radians

Table 5.13. Output parameters for subroutine SENSOR

Symbol	*f18harv* Name	Local Name	Definition
α_{probe}	ALFPRO	ALFS1	airdata probe angle of attack w/o bias or gain applied, deg
n_{y_s}	AYS	AYACC	sensor-model lateral acceleration, g's
n_{z_s}	AZS	AZACC	sensor-model normal acceleration, g's
p_s	PS	XQGY	sensor-model pitch rate, degrees/sec
q_s	QS	XPGY	sensor-model roll rate, degrees/sec
r_s	RS	XRGY	sensor-model yaw rate, degrees/sec
Q_{c_s}	QCIS	QCIS	sensor-model impact pressure, psf
P_{s_s}	PSTS	PSTS	sensor-model static atmospheric pressure, psf
R_{i_s}	RIS	RIS	pressure ratio, n.d.
-	LOP20	LOP20	20 Hz loop flag
-	LOP40	LOP40	40 Hz loop flag
M_s	MACHS	MACHS	sensor-model Mach number, n.d.
V_s	VTS	VTS	sensor-model true velocity, fps
$\bar{q}_s$	QBARSN	QBARSN	sensor-model dynamic pressure, psf
$\cos \phi_s$	COSPHI	COSPHI	cosine of sensor-model Euler roll angle, n.d.
$\sin \phi_s$	SINPHI	SINPHI	sine of sensor-model roll angle, n.d.
$\cos \theta_s$	COSTHE	COSTHE	cosine of sensor-model Euler pitch angle, n.d.
$\sin \theta_s$	SINTHE	SINTHE	sine of pitch angle, n.d.
H_s	ALTS	ALTS	sensor-model altitude, ft
γ_s	GAMS	GAM	sensor-model wind-axis Euler angle, radians
μ_s	MUS	MU	sensor-model wind-axis Euler angle, radians
χ_s	CHIS	CHI	sensor-model wind-axis Euler angle, radians
-	QCFILT1	QCFILT1	sensor-model impact pressure filtered at 2.5 rad/sec, psf
-	QCFILT2	QCFILT2	sensor-model impact pressure filtered at 10. rad/sec, psf
$\sin \gamma_s$	SINGAM	SINGAM	sine of sensor-model wind-axis Euler angle, n.d.
$\cos \gamma_s$	COSGAM	COSGAM	cosine of sensor-model wind-axis Euler angle, n.d.
$\sin \mu_s$	SINMUS	SINMUS	sine of sensor-model wind-axis Euler angle, n.d.
$\cos \mu_s$	COSMUS	COSMUS	cosine of sensor-model wind-axis Euler angle, n.d.
$\sin \chi_s$	SINCHIS	SINCHIS	sine of sensor-model wind-axis Euler angle, n.d.
$\cos \chi_s$	COSCHIS	COSCHIS	cosine of sensor-model wind-axis Euler angle, n.d.

6.0 CONTROL SURFACE ACTUATOR MODIFICATIONS

Since the original release of the *f18bas* simulation, the actuator models have undergone some significant modifications. The following list summarizes the simulation changes that were required to make the actuation system modeled in the *f18harv* simulation representative of the F/A-18 HARV.

(1) Second order actuator models based on Dryden F/A-18 HARV simulation
(2) No-load rate-limit modifications
(3) Hinge-moment model
(4) Thrust-vectoring vanes (3 per engine, 6 total)
(5) Nose-strake actuator model

A discussion of the thrust-vectoring-vane actuation system implemented in the *f18harv* simulation was presented in section 2 and will not be covered in this section.

6.1 Primary Controls

The actuation system for the primary controls described in Chapter 8 of Reference 1.0 has been replaced in the *f18bas* simulation. The *f18harv* simulation contains the same actuation model implemented in the Dryden batch F/A-18 HARV simulation. These actuator models are higher fidelity second-order models versus the simple first-order models implemented in the *f18bas* simulation. A series of subroutines and functions make up the actuator model used in *f18harv*. Figure 6.1 contains a block diagram of the CALLing sequence used for the new actuator models. A description of subroutines that form the actuator model follows.

ACTDYN	Serves as the top level actuator and hinge-moment model for the F/A-18 simulations.
ACTSET	Initializes values needed for the actuator model.
STHNGI	Initializes the strake hinge-moment model. Reads in data file called STRAKE_HINGE.DAT.
ACTSIN	Initializes the strake-actuator model.
HNGINDX	Computes indices and interpolation ratios for use in computation of dependent variables.
HNGTLU	Calls individual functions used to evaluate each dependent variable.
HINGE	Computes actuator rates, modeling power cylinders and series servo positions from mechanical and CAS control inputs.
ACTMOD	Models actuator with second-order dynamics.
STRKACT	Serves as top level subroutine for strake-actuator and hinge-moment models.
STKHNG	CALLs the strake hinge-moment model to find the external loads.
STKNLG	Converts the actuator positions to degrees (Non-linear gearing).

The transfer functions for the five primary actuators (stabilator, rudder, aileron, leading-edge flaps, and trailing-edge flaps) are presented in Table 6.1. A logical has been added to the ACSL simulation to set the surface commands equal to the deflections. This logical is called LNOACT and has a default value of FALSE. Setting LLN to TRUE will also set the surface commands equal to the deflections.

6.2 Nonlinearities - Primary Controls

The only nonlinear actuator effects modeled in the *f18bas* simulation were the no-load rate limits. A hinge-moment model which changes the no-load rate limit based on flight condition

and aerodynamic surface loads has been added to the *f18harv* simulation. Hinge-moment increments are computed from table look-ups as a function of angle of attack, angle of sideslip, Mach number, altitude, and surface positions. The aerodynamic hinge-moment capability is computed by multiplying each surface increment by the surface area and dynamic pressure. Hinge-moment ratios are then computed by dividing the aerodyanamic capability by the maximum actuator hinge-moment capability in either extension or retraction. The square root of the hinge-moment ratio added (+ rate limit) or subtracted (- rate limit) from one is then multiplied by the no-load rate limit. The resulting product from this calculation corresponds to a hinge-moment rate limit. The hinge-moment model is based on MDC reports A7813, A7247, and A8450, which are listed in References 6.0, 6.1, and 6.2, respectively. The rudder and leading-edge-flap no-load rate limits were modified from *f18bas* to reflect the Dryden F/A-18 HARV simulations. The position limits for the aileron leading-edge flaps were also modified. Table 6.2 contains the no-load rate and position limits for the primary controls.

Table 6.1. Linear Actuator Models - $\delta(s)/\delta_c(s)$

Actuator	Transfer Function
Stabilator	$\frac{30.74^2}{s^2 + 2(0.509)(30.74)s + 30.74^2}$
Rudder	$\frac{72.1^2}{s^2 + 2(0.69)(72.1)s + 72.1^2}$
Aileron	$\frac{75^2}{s^2 + 2(0.59)(75)s + 75^2}$
Trailing-Edge Flap	$\frac{35^2}{s^2 + 2(0.71)(35)s + 35^2}$
Leading-Edge Flap	$\frac{(26.9)(82.9)}{(s + 26.9)(s + 82.9)}$

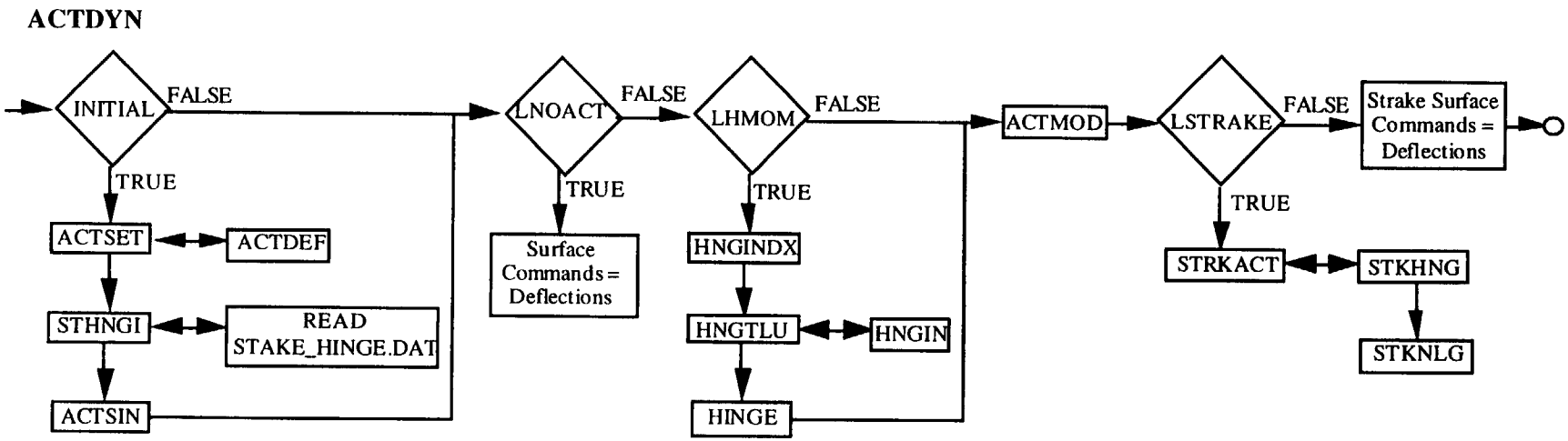


Figure 6.1. Actuator Model Flow Chart

Table 6.2. F/A-18 HARV Actuator Characteristics

Actuator	Rate Limit	Position Limit
Stabilator	+/- 40°/s	-24°,+10.5°
Rudder	+/- 82°/s	-30°,+30°
Aileron	+/- 100°/s	-25°,+42°
Trailing-Edge Flaps	+/- 18°/s	-8°,+45°
Leading-Edge Flaps	+/- 15°/s	-3°,+33°

6.3 ANSER Actuator Model Modifications

The actuators for the ANSER system are modeled in the simulation according to specifications received from the actuator manufacturer, HR Textron. These specifications describe a simple second-order no-load dynamic model of the closed-loop actuator. The model assumes that the present F/A-18 aileron-controller electronics are used in conjunction with the actuator, four-way servo-valve, and Linear Variable Differential Transformer (LVDT). The transfer function for this no-load model is given as:

$$\frac{XA}{XC} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$
$$\omega = \sqrt{\omega_1 \cdot \omega_2} \qquad (6.1)$$
$$\zeta = \frac{1}{2}\sqrt{\frac{\omega_2}{\omega_1}}$$

where XA is the actual output actuator position, and XC is the commanded actuator position. See Table 6.3 for values of ω_1 and ω_2 and a listing of relevant actuator parameters, including position and rate limits.

Table 6.3. ANSER Actuator Parameters

Symbol	Numerical Value	Comments
ω_1	67.0 rad/sec	no-load open-loop gain; ω_1 and ω_2 combined to get
ω_2	200.0 rad/sec	transfer function coefficients D,E,F in subroutine STRACT
XA(min)	-2.84 inches	negative displacement limit, equivalent to 0.0 deg
XA(max)	+2.84 inches	positive displacement limit, 90 deg
VA(min)	-11.36 inches/sec	negative rate limit, -180 deg/sec
VA(max)	+11.36 inches/sec	positive rate limit, +180 deg/sec

Figure 6.2 shows a block diagram representation of the servo loop. Bode magnitude and phase plots for this simplified system are given in Figure 6.3 This model is valid for 90 deg F hydraulic fluid temperature and 2765 psid pressure drop across the actuator.

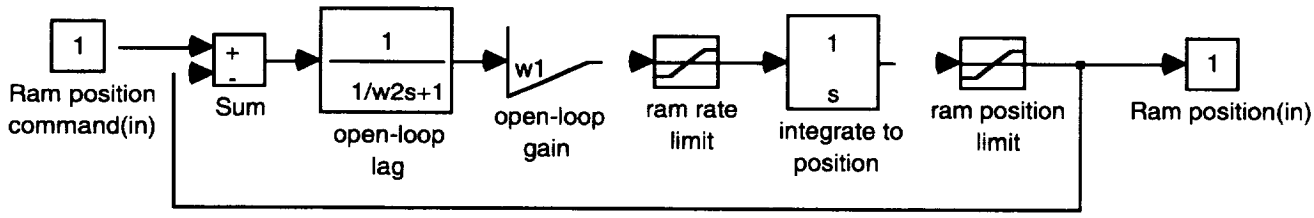


Figure 6.2. Strake-actuator servo loop block diagram

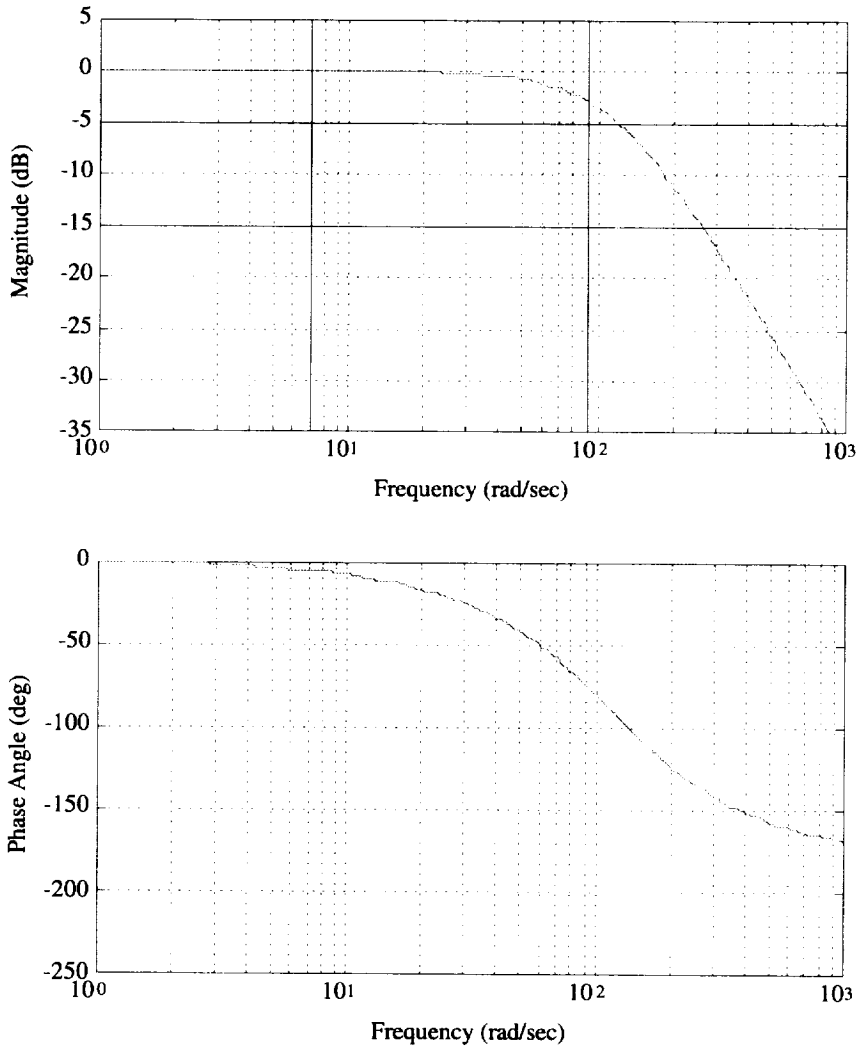


Figure 6.3. Gain and phase response for ANSER actuator no-load dynamic model

The FORTRAN implementation of the actuator was accomplished in subroutine STRKACT, which performs conversion from inches to degrees of strake deflection and interfaces with subroutine ACTDYN.

Figure 6.4 shows a flow chart of subroutine STRKACT.

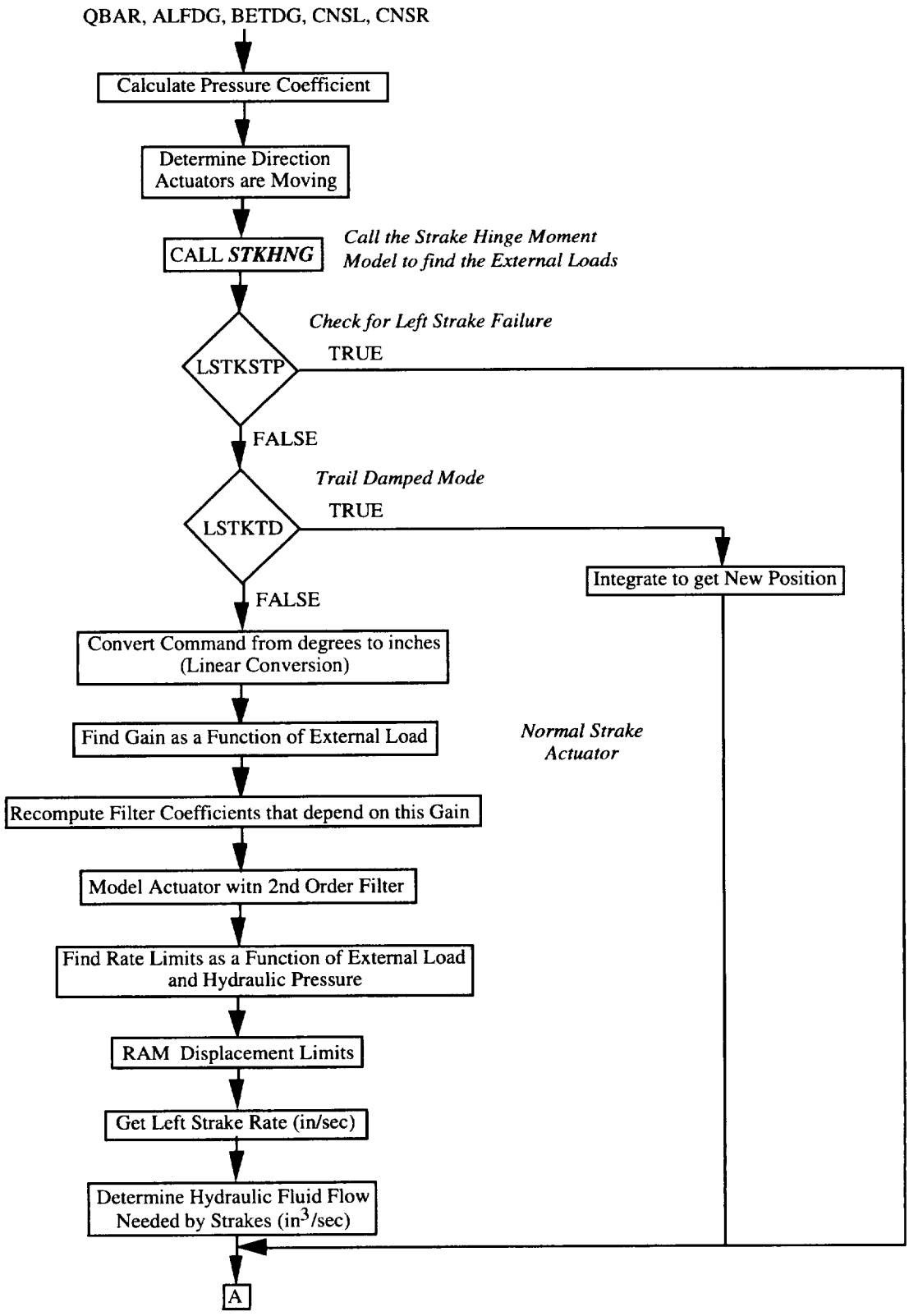


Figure 6.4. Strake-actuator model flow chart.

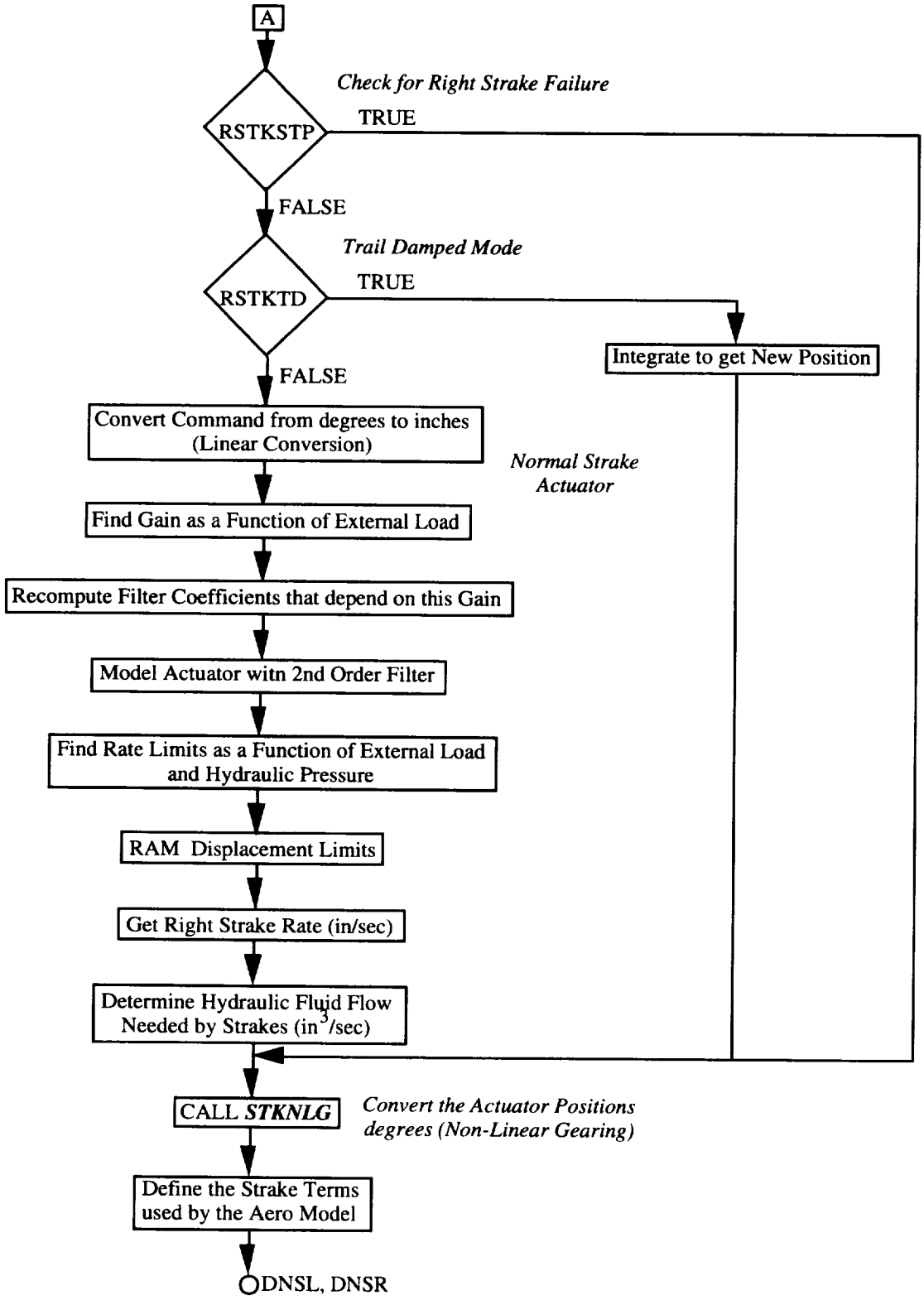


Figure 6.4. Concluded.

Table 6.4. Input Parameters for Subroutine ACTDYN

f18harv Name	Local Name	Simulation Source	Definition
ALFDG	ALFDG	CSL	angle of attack, degrees
BETDG	BETDG	CSL	angle of sideslip at the c.g., degrees
MACH	MACH	CSL	Mach number, n.d.
H	H	CSL	altitude, ft
QBAR	QBAR	CSL	aircraft dynamic pressure, lbs/inches^2
LHMOM	LHMOM	CSL	logical to allow use of hinge-moment effects,
LNOACT	LNOACT	CSL	logical flag to choose DRYDEN second-order actuator model
LTRFLG	LTRFLG	THREST	logical switch to indicate trim status
LLN	LLN	CSL	logical switch for Jacobian calculations
LSTR	LSTR		logical nose-strake actuator selection switch
LSTKTD	LSTKTD	CSL	logical to fail left nose-strake actuator in trail damp mode
RSTKTD	RSTKTD	CSL	logical to fail right nose-strake actuator in trail damp mode
LSTKSTP	LSTKSTP	CSL	logical to fail left nose-strake actuator
RSTKSTP	RSTKSTP	CSL	logical to fail right nose-strake actuator
CSL	CSL	CSL	left stabilator command, degrees
CSR	CSR	CSL	right stabilator command, degrees
CAL	CAL	CSL	left aileron command, degrees
CAR	CAR	CSL	right aileron command, degrees
CRL	CRL	CSL	left rudder command, degrees
CRR	CRR	CSL	right rudder command, degrees
CNL	CNL	CSL	left leading-edge flap command, degrees
CNR	CNR	CSL	right leading-edge flap command, degrees
CFL	CFL	CSL	left trailing-edge flap command, degrees
CFR	CFR	CSL	right trailing-edge flap command, degrees
CNSL	CNSL	CSL	left nose-strake command, degrees
CNSR	CNSR	CSL	right nose-strake command, degrees

Table 6.5. Output Parameters for Subroutine ACTDYN

f18harv Name	Local Name	Definition
DSL	DSL	total left stabilator position, includes linearization input, degrees
DSR	DSR	total right stabilator position, includes linearization input, degrees
DAL	DAL	total left aileron position, includes linearization input, degrees
DAR	DAR	total right aileron position, includes linearization input, degrees
DRL	DRL	total left rudder position, includes linearization input, degrees
DRR	DRR	total right rudder position, includes linearization input, degrees
DNL	DNL	total left leading-edge flap position, includes linearization input, degrees
DNR	DNR	total right leading-edge flap position, includes linearization input, degrees
DFL	DFL	total left trailing-edge flap position, includes linearization input, degrees
DFR	DFR	total right trailing-edge flap position, includes linearization input, degrees
DNSL	DNSL	left nose-strake deflection, degrees
DNSR	DNSR	right nose-strake deflection, degrees

7.0 RESEARCH FLIGHT CONTROL SYSTEM (RFCS)

The *f18harv* simulation implements two RFCS control laws: the McAir/Dryden designs commonly referred to as NASA-0, and the LaRC design, commonly referred to as ANSER. Since numerous versions of these control systems were tested and analyzed through batch simulation, DMS piloted evaluations, and flight tests on the F/A-18 HARV, the following table defines the specific version of the RFCS code that is implemented in the *f18harv* simulation release documented to this report.

McAir/DFRC RFCS (NASA-0)	- Version 28
LaRC RFCS (ANSER)	- Longitudinal Controls Version 151.0 - Lateral/Directional Controls Version 6.9.9N - Lateral/Directional Pseudo Controls Version 8.1

This report will not attempt to present the design or theory of the Research Flight Control Systems since numerous publications on the research associated with their development have been produced. (See, for example, refs. 7.0 - 7.2.) Implementation of the RFCS designs incorporated into the *f18harv* simulation is discussed in detail in the following sections, and documentation of the interface to the RFCS subroutines is included.

7.1 RFCS Implementation

The actual F/A-18 HARV aircraft uses a GE 701E computer which contains the basic F/A-18 control system and a PACE 1750A computer which contains the RFCS. The basic F/A-18 control system software was modified (Version 10.1) to interface with RFCS through dual-port random-access memory (DPRAM). The *f18harv* simulation operates only in RFCS. DPRAM and the basic F/A-18 control system are not modeled in the *f18harv* simulation. The RFCS communicates through DPRAM to the 701E computer and has no direct communication to the aircraft. Therefore, the aircraft flies using the basic F/A-18 control system until the pilot engages RFCS through a cockpit switch. The pilot can only engage RFCS when the aircraft is operating within the F/A-18 HARV flight envelope.

The RFCS control law is represented by a set of block diagrams implemented as a MATRIX$_X$® System Build™ model. Using the System Build model, the control law designer can perform linear control system design and analysis in MATRIX$_X$®. The MATRIX$_X$® AutoCode™ feature is invoked to automatically create FORTRAN-coded subroutines from the System Build model. These subroutines represent the RFCS control laws and are implemented in the ACSL *f18harv* simulation. Hand-coded FORTRAN changes are made to the Autocode where necessary. The FORTRAN subroutines are compiled to create an object library which is linked with the rest of the simulation. A control-law-interface subroutine HARVCSM is used to CALL the various AutoCode subroutines. HARVCSM is CALLed from the INITIAL and DISCRETE sections in the ACSL simulation with the CALL to HARVCSM occurring every 0.0125 seconds (80 Hz) in the DISCRETE section. A flow chart of HARVCSM is shown on Figure 7.1. Definitions of the various RFCS AutoCode subroutines are listed in Table 7.1. All RFCS subroutines are CALLed at 80 Hz except USR18V150P3 which is CALLed at 40 Hz.

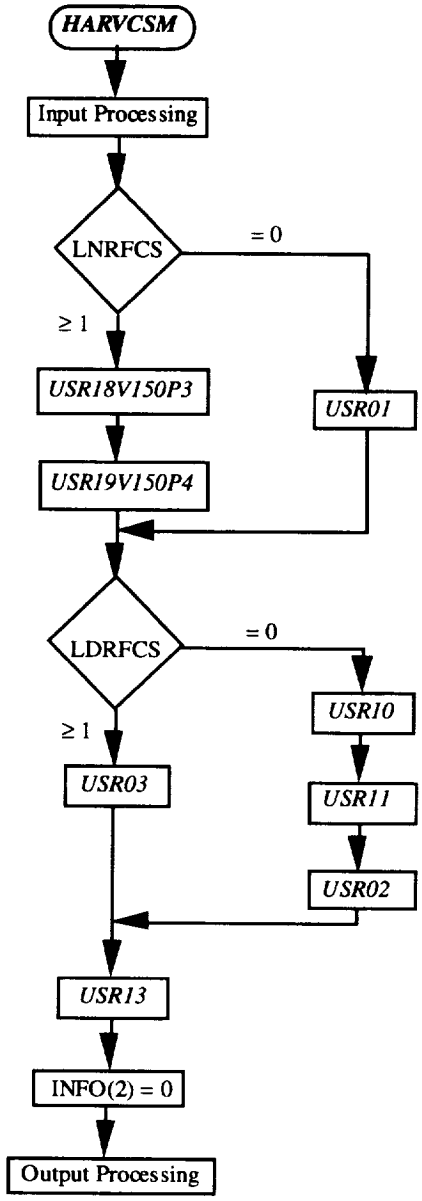


Figure 7.1. Flow chart of RFCS control law interface

The McAir/DFRC lateral/directional RFCS design uses the 701E because this RFCS design is not a full F/A-18 HARV-envelope design. Therefore, modeling of the lateral/directional 701E was needed in *f18harv* and is included as subroutines USR10 and USR11. Two integer switches have been incorporated into *f18harv* to select between the NASA-0 and ANSER RFCS. These switches are called LNRFCS and LDRFCS corresponding to the longitudinal axis and lateral/directional axis, respectively. When LNRFCS and

Table 7.1. RFCS AutoCode Subroutines

SUBROUTINE	Description
USR01	McAir/DFRC Longitudinal RFCS (80 Hz)
USR02	McAir/DFRC L/D RFCS (80 Hz)
USR10	McAir/DFRC Lateral 701E Control System (80 Hz)
USR11	McAir/DFRC Directional 701E Control System (80 Hz)
USR18V150P3	LaRC Longitudinal RFCS (40 Hz)
USR19V150P4	LaRC Longitudinal RFCS (80 Hz)
USR03	LaRC L/D Top Level Block RFCS (80 Hz)
USR13	RFCS Command Mixer (80 Hz)
USR46	LaRC L/D Feedback Block RFCS (80 Hz)
USR47	LaRC L/D Pilot Command Block RFCS (80 Hz)
USR39FJL	LaRC L/D Pseudo Controls (80 Hz)
USRCKFJL	LaRC L/D Pseudo Controls (80 Hz)
USR33FJL	LaRC L/D Pseudo Controls (20 Hz)
USR34FJL	LaRC L/D Pseudo Controls (20 Hz)
USR36FJL	LaRC L/D Pseudo Controls (20 Hz)
USR59FJL	LaRC L/D Pseudo Controls (20 Hz)
USR52FJL	LaRC L/D Pseudo Controls (40 Hz)
USR51FJL	LaRC L/D Pseudo Controls (40 Hz)
USR35FJL	LaRC L/D Pseudo Controls (80 Hz)
USR38FJL	LaRC L/D Pseudo Controls (80 Hz)
USR54FJL	LaRC L/D Pseudo Controls (40 Hz)
USR53FJL	LaRC L/D Pseudo Controls (40 Hz)
USR58FJL	LaRC L/D Pseudo Controls (80 Hz)

LDRFCS are both set to 1 (DEFAULT), ANSER is invokeed in the *f18harv* simulation. When LNRFCS and LDRFCS are both set to 0, the NASA-0 RFCS is invoked. Integers other than 0 or 1, such as 2 and 2, also select the ABSER control law. A combination of 0 and 1 flag settings is allowable but results are meaningless since HARVCSM was not developed to ensure proper input/output between these control law combinations.

7.2 Input Processing in HARVCSM

As previously stated, all inputs to RFCS on the F/A-18 HARV pass through the 701E computer and DPRAM. Since the 701E modeled in the *f18harv* simulation is used for only a portion of the NASA-0 design, additional 701E modeling of input parameters passed to RFCS exists. Significant effects in the pilot bandwidth, such as breakouts and filtering which occur in the 701E, are modeled. The first input processing performed in subroutine HARVCSM is on pilot stick and pedal inputs before they are passed to RFCS. A 7 lb mechanical breakout is applied to the pilot rudder-pedal command PCR to calculate an output variable called DRP. The second processing is application of an aliasing filter to the pilot rudder-pedal command when the RFCS lateral/directional option is not equal to 1. If LDRFCS equals 1, the rudder-pedal-command aliasing filter is not applied. The following equation represents the transfer function for the rudder-pedal aliasing filter:

$$H_{DRP}(s) = \frac{(25.5)(174.5)}{(s+25.5)(s+174.5)} \tag{7.1}$$

The aliasing filter in HARVCSM has an update rate of 40 Hz which models the sample rate that exists in the 701E.

The pilot lateral-stick input passes through a demod filter in the 701E. The pilot lateral-stick input to HARVCSM is PCA with an update rate of 80 Hz. The demod filter is applied to PCA in HARVCSM to calculate DAPRFCS. The transfer function for the demod filter is as follows:

$$H_{DAP}(s) = \frac{303}{s + 303} \tag{7.2}$$

The demod filter is not be applied if LDRFCS equals 1.

The sensor model emulates the various update rates of the input parameters passed to RFCS. The McAir/DFRC RFCS inputs which are updated at 40 Hz are averaged with the last input value. This is done to smooth the noise associated with the input signals. The averaging also has the effect of causing a one frame (0.0125 seconds) delay. All averaging for the NASA-0 RFCS occurs inside the AutoCode subroutines. None of the ANSER RFCS longitudinal 40 Hz inputs are averaged, but all of the ANSER lateral/directional 40 Hz inputs are averaged. The averaging of the ANSER lateral/directional 40 Hz inputs occurs in HARVCSM in the portion of code that sets up the input array for subroutine USR03.

7.3 AutoCode Interface

The HARVCSM interface defines input, state, and output arrays that are passed through the AutoCode CALL statements. The AutoCode outputs are assigned variable names and passed to ACSL through the HARVCSM CALL statement. The general AutoCode subroutine CALL argument list is as follows:

(INFO, T, U, NU, X, XD, NX, Y, NY, RPP, IPP)

The first argument called INFO is a one dimensional integer array of length 14 used to set input flags for the FORTRAN AutoCode. The array is initialized through a DATA statement prior to any CALLs to the FORTRAN AutoCode. The INFO array is initially set as INFO = (0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), where INFO(2) = 1 is used to initialize the FORTRAN AutoCode subroutines on the first CALL. After the last AutoCode subroutine CALL (USR13), INFO(2) is set to 0. INFO(3) and INFO(4) correspond to the state and output equation flags and are always set to 1. All other INFO parameters are not used in the FORTRAN AutoCode and remain set to 0.

Simulation time T passed as the second argument is not used in the AutoCode. The input array U is declared a REAL array and dimensioned to the number of inputs. The number of inputs NU is an integer that is set equal to the number of array elements of U. The state vector X is declared a REAL array and dimensioned to the number of states. XD, the time derivative of X, represents the value of X projected to the next sample time and is dimensioned to the number of states. The states are updated to the next sample time by setting X = XD external to the subroutines before each CALL. The number of states NX is an integer that is equal to the number of array elements of X and XD. The output vector Y is also declared a REAL array and dimensioned to the number of outputs. The number of outputs NY is an integer equal to the number of array elements of Y. The last two arguments correspond to arrays that are created by MATRIX$_X$® AutoCode but not used in the simulation. These are RPP and IPP which are defined as real and integer arrays, respectively. The input,

output, and state arrays for all AutoCode subroutines are listed in the following tables.

Table 7.2. AutoCode subroutine USR01 inputs

Array	*f18harv* Name	Local Name	Definition
U1(1)	PCS	PSTICK	Pilot longitudinal stick-position, inches
U1(2)	QS	QSEN7	Sensor-model 701E pitch rate, degrees/sec
U1(3)	QS	QSEN7	Sensor-model RFCS pitch rate, degrees/sec
U1(4)	ALFPRO	AOAP	Airdata-probe angle of attack, degrees
U1(5)	ALFINS	AOAINS	Mission-Computer-calculated angle of attack, deg
U1(6)	0.0	BLANK1	Not used
U1(7)	NZS	AZSEN	Sensor-model normal acceleration, g's
U1(8)	PS	PSEN	Sensor-model roll rate, degrees/sec
U1(9)	RS	RSEN	Sensor-model yaw rate, degrees/sec
U1(10)	PTRIM	PTRIM	Pilot pitch trim button command (-1, 0, +1)
U1(11)	PSTS	PS	Sensor-model static pressure, psf
U1(12)	QCIS	QC	Sensor-model impact pressure, psf
U1(13)	RIS	RI	Sensor-model pressure ratio, n.d.
U1(14)	SCF7	SCF7	Speedbrake compensation from 701E, deg (not connected, input = 0)
U1(15)	CSCF7	CSCF7	Collective stabilator command from 701E, degrees (not connected, input = 0)
U1(16)	RCSC	RCSC	RAV collective stabilator cmd, deg (not connected, input = 0)
U1(17)	CLEFCF7	CLEFCF7	Collective LEF command from 701E, degrees (not connected, input = 0)
U1(18)	RCLEFC	RCLEFC	RAV collective LEF command, degrees (not connected, input = 0)
U1(19)	CTEFCF7	CTEFCF7	Collective TEF command from 701E, degrees (not connected, input = 0)
U1(20)	RCTEFC	RCTEFC	RAV collective TEF command, degrees (not connected, input = 0)
U1(21)	ERME	ERME	ERMENG_RFCS_MODE_EN (REAL) (not connected, input = 0)
U1(22)	LTRFLG	TRMMING	Trim flag (REAL) - 0.0 = operate, 1.0 = trim
U1(23)	DELSTM	DELSTM	Input to trim algorithm that is the sum of collective stabilator & pitch-jet-angle command used to initialize the forward loop integrator; deg
U1(24)	ADS	AOARI	Sensor-model angle-of-attack rate, degrees/sec
U1(25)	HANGCUL	HANGCUL	HANG command upper limit, deg (default 25.5)
U1(26)	HANGRL	HANGRL	HANG rate limit, deg/sec (default 320.0)
U1(27)	DGTHLC(1)	LEFTPLA	Left-engine lagged throttle position, degrees
U1(28)	DGTHLC(2)	RIGHTPLA	Right-engine lagged throttle position, degrees
U1(29)	2.0	MODE	Reset mode (fixed to 2.0)

Table 7.3. AutoCode subroutine USR01 outputs

Array	*f18harv* Name	Local Name	Definition
Y1(1)	CSCT7	CSCT7	Collective stabilator command to 701E, degrees
Y1(2)	PJETAC	VAR_RTE_LIM_98_1	Thrust-vectoring pitch-jet-angle command, degrees
Y1(3)	CLEFCT7	CLEFCT7	Collective LEF command to 701E, degrees
Y1(4)	CTEFCT7	CTEFCT7	Collective TEF command to 701E, degrees

Table 7.3. Concluded.

Array	*f18harv* Name	Local Name	Definition
Y1(5)	ALFS	AOA	Selected angle of attack, degrees
Y1(6)	NZAF	NZF	Nz feedback, g's
Y1(7)	QSEN	QSEN	Selected pitch rate, degrees/sec
Y1(8)	FGI	FGI	Fader gain input
Y1(9)	CMDGAIN	CMDGAIN	Command gain
Y1(10)	AOAGAIN	AOAGAIN	angle-of-attack gain
Y1(11)	AOARG	AOARG	angle-of-attack rate gain
Y1(12)	QGAIN	QGAIN	Pitch-Rate-Feedback gain
Y1(13)	AOAFB	AOAFB	Open-loop-compensator input
Y1(14)	LPFIN	VAR_UP_LI M_31_1_5	Low-pass-filter input
Y1(15)	PCMD	PCMD	Stick command at input to feedback error summing junction
Y1(16)	VCMD	VCMD	Total virtual command for stabilator & pitch-jet angle (before limiter)
Y1(17)	ERRORS	ERRORS	Error signal out of proportional gain (to trim longitudinal stick)
Y1(18)	RFCSMACH	MACH	RFCS computed Mach number, n.d.
Y1(19)	AOAB	AOAB	ALFS and AOA rate inertial-feedback summing-junction output, degrees
Y1(20)	AOARIB	AOARIB	AOA inertial feedback & QSEN summing-junction output, degrees/sec
Y1(21)	CSCMDA	CSCMDA	Symmetric stabilator command to rate limiter, deg
Y1(22)	PJACMDA	PJACMDA	Pitch-jet command to rate limiter, degrees
Y1(23)	PCMDA	PCMDA	Stick command at input to lead-lag filter

Table 7.4. AutoCode subroutine USR01 States

Array	Forcing Function	Trim Equations
X1(1)	-	X1(1) = XD1(1)
X1(2)	-	X1(2) = XD1(2)
X1(3)	-	X1(3) = XD1(3)
X1(4)	-	X1(4) = XD1(4)
X1(5)	-	X1(5) = XD1(5)
X1(6)	-	X1(6) = XD1(6)
X1(7)	-	X1(7) = XD1(7)
X1(8)	-	X1(8) = XD1(8)
X1(9)	-	X1(9) = XD1(9)
X1(10)	Y1(22), PJACMDA	X1(10) = Y1(22)
X1(11)	Y1(21), CSCMDA	X1(11) = Y1(21)
X1(12)	Y1(8), FGI	X1(12) = 2.12500·Y1(8)
X1(13)	Y1(7), QSEN	X1(13) = 3.5719·Y1(7)
X1(14)	Y1(7), QSEN	X1(14) = 3.5719·Y1(7)
X1(15)	Y1(23), PCMDA	X1(15) = 2.91964·Y1(23)
X1(16)	Y1(20), AOARIB	X1(16) = 10.06179·Y1(20)
X1(17)	Y1(19), AOAB	X1(17) = 5.12495·Y1(19)
X1(18)	Y1(19), AOAB	X1(18) = 10.06279·Y1(19)
X1(19)	Y1(5), ALFS	X1(19) = 7.9623·Y1(5)
X1(20)	Y1(5), ALFS	X1(20) = 7.92494·Y1(5)
X1(21)	Y1(9), CMDGAIN	X1(21) = 2.12240·Y1(9)

Table 7.4. Concluded.

Array	Forcing Function	Trim Equations
X1(22)	Y1(10), AOAGAIN	X1(22) = 2.12240·Y1(10)
X1(23)	Y1(12), QGAIN	X1(23) = 2.12240·Y1(12)
X1(24)	Y1(11), AOARG	X1(24) = 2.12240·Y1(11)
X1(25)	Y1(13), AOAFB	X1(25) = 17.0137·Y1(13)
X1(26)	Y1(13), AOAFB	X1(26) = 17.0137·Y1(13)
X1(27)	Y1(14), LPFIN	X1(27) = 5.06257·Y1(14)

Table 7.5. AutoCode subroutine USR02 inputs

Array	*f18harv* Name	Local Name	Definition
U2(1)	PS	PSEN	Sensor-model roll rate, degrees/sec
U2(2)	BDS	BDOTSEN	Sensor-model sideslip rate, degrees/sec
U2(3)	RS	RSEN	Sensor-model yaw rate, degrees/sec
U2(4)	AYS	AYSEN	Sensor-model lateral acceleration, g's
U2(5)	LTSTICK	LTSTICK	Lateral stick, inches (from USR10)
U2(6)	RPEDAL	RPEDAL	Rudder pedal, lbs (from USR11)
U2(7)	ALFS	AOA	Selected angle of attack, degrees (from USR01)
U2(8)	QCIS	QC	Sensor-model impact pressure, psf
U2(9)	RIS	R1	Sensor-model pressure ratio, n.d.
U2(10)	PSTS	PS	Sensor-model static pressure, psf
U2(11)	NZAF	NZF	Nz feedback, g's (from USR01)
U2(12)	CSCT7	CSCMD	Collective stabilator command to 701E, deg (from USR01)
U2(13)	QS	QSEN	Sensor-model pitch rate, degrees/sec
U2(14)	YTRIM	YTRIM	Pilot yaw-trim-knob command, degrees
U2(15)	RTRIM	RTRIM	Pilot roll-trim-button command (-1, 0, +1)
U2(16)	TRMMING	TRMMING	Trim flag (REAL) - 0.0 = operate, 1.0 = trim
U2(17)	DTCF7	DTCF7	Differential tail command from 701E, deg (from USR10)
U2(18)	RDTC	RDTC	RAV differential stabilator command, degrees (not connected, input = 0)
U2(19)	DACF7	DACF7	Differential aileron cmd from 701E, degrees (from USR10)
U2(20)	RAC	RAC	RAV aileron command, deg (not connected, input = 0)
U2(21)	DLEFCF7	DLEFCF7	Differential LEF command from 701E, deg (from USR10)
U2(22)	RDLEFC	RDLEFC	RAV differential LEF cmd, deg (not connected, input = 0)
U2(23)	DTEFCF7	DTEFCF7	Differential TEF command from 701E, deg (from USR10)
U2(24)	RDTEFC	RDTEFC	RAV differential TEF cmd, deg (not connected, input = 0)
U2(25)	RCF7	RCF7	Rudder command from 701E, degrees (from USR11)
U2(26)	RRC	RRC	RAV rudder command, deg (not connected, input = 0)

Table 7.6. AutoCode subroutine USR02 outputs

Array	*f18harv* Name	Local Name	Definition
Y2(1)	RSCT7	RSCT7	Roll stabilator command to 701E, degrees
Y2(2)	ACT7	RATE_13	Aileron command to 701E, degrees
Y2(3)	RLEFCT7	RLEFCT7	Roll LEF command to 701E, degrees
Y2(4)	RTEFCT7	RTEFCT7	Roll TEF command to 701E, degrees
Y2(5)	RCT7	RATE_12	Rudder command to 701E, degrees
Y2(6)	YJETAC	YJACTM	Thrust-vector yaw-jet-angle command, degrees
Y2(7)	LAF1IN	LAF1IN	Lateral accelerometer filter 1 input
Y2(8)	RAFS	RAFS	Roll-axis feedback summation
Y2(9)	YAFS	YAFS	Yaw-axis feedback summation

Table 7.6. Concluded.

Array	*f18harv* Name	Local Name	Definition
Y2(10)	R7FID	R7FID	RFCS 701E fader gain
Y2(11)	LSG	LSG	Lateral stick gradient
Y2(12)	BDOTSCR	BDOTSCR	BDS summing-junction output, degrees
Y2(13)	ACT7A	ACT7A	Aileron command to rate limit, degrees
Y2(14)	RCT7A	RCT7A	Rudder command to rate limit, degrees

Table 7.7. subroutine USR02 states

Array	Forcing Function	Trim Equations
X2(1)	-	X2(1) = XD2(1)
X2(2)	-	X2(2) = XD2(2)
X2(3)	-	X2(3) = XD2(3)
X2(4)	U2(1), PS	X2(4)=0.8756·U2(1)
X2(5)	U2(1), PS	X2(5)=0.4378·U2(1)
X2(6)	U2(1), PS	X2(6)=4.8285·U2(1)
X2(7)	U2(1), PS	X2(7)=4.8285·U2(1)
X2(8)	U2(2), BDS	X2(8)=1.5601·U2(2)
X2(9)	U2(2), BDS	X2(9)=1.5601·U2(2)
X2(10)	Y2(11), LSG	X2(10)=4.0075·Y2(11)
X2(11)	Y2(11), LSG	X2(11)=4.0075·Y2(11)
X2(12)	Y2(12), BDOTSCR	X2(12)=1.5928·Y2(12)
X2(13)	Y2(12), BDOTSCR	X2(13)=1.5928·Y2(12)
X2(14)	U2(25), RCF7	X2(14)=1.829·U2(25)
X2(15)	U2(25), RCF7	X2(15)=1.829·U2(25)
X2(16)	Y2(13), ACT7A	X2(16)=Y2(13)
X2(17)	Y2(14), RCT7A	X2(17)=Y2(14)
X2(18)	U2(11), NZAF	X2(18)=3.6050·U2(11)
X2(19)	Y2(8), RAFS	X2(19)=6.0262·Y2(8)
X2(20)	Y2(8), RAFS	X2(20)=6.0262·Y2(8)
X2(21)	Y2(8), RAFS	X2(21)=0.98260·Y2(8)
X2(22)	Y2(9), YAFS	X2(22)=-1.097·Y2(9)
X2(23)	Y2(9), YAFS	X2(23) = -1.097·Y2(9)
X2(24)	Y2(9), YAFS	X2(24) = 0.9826·Y2(9)
X2(25)	-	X2(25) = XD2(25)
X2(26)	Y2(10), R7FID	X2(24) = 10.06279·Y2(10)

Table 7.8. AutoCode subroutine USR10 inputs

Array	*f18harv* Name	Local Name	Definition
U10(1)	DAPRFCS	LSTICK	Pilot lateral stick after 701E demod filter, inches
U10(2)	AZS	AZSEN	Sensor-model normal acceleration, g's
U10(3)	PS	PSEN	Sensor-model roll rate, degrees/sec
U10(4)	QCIS	QC	Sensor-model impact pressure, psf
U10(5)	PSTS	PS	Sensor-model static pressure, psf
U10(6)	RIS	R1	Sensor-model pressure ratio, n.d.
U10(7)	RTRI	RTRI	Rudder-to-roll interconnect (from USR11)
U10(8)	ALFS	AOA	Selected angle of attack, degrees (from USR01)
U10(9)	NZAF	NZF	Nz feedback, g's (from USR01)

Table 7.8. Concluded.

Array	*f18harv* Name	Local Name	Definition
U10(10)	CSCT7	CSCMD	Collective stabilator command, deg (from USR01)
U10(11)	TRMMING	TRMMING	Trim flag (REAL) - 0.0 = operate, 1.0 = trim

Table 7.9. AutoCode subroutine USR10 outputs

Array	*f18harv* Name	Local Name	Definition
Y10(1)	DLEFC	DLEFC	Differential LEF command, degrees
Y10(2)	DTEFC	RATEZ11	Differential TEF command, degrees
Y10(3)	DTC	RATEZ17	Differential stabilator command, degrees
Y10(4)	DAC	DAC	Differential aileron command, degrees
Y10(5)	RSRI	RSRI	Rolling surfac- to-rudder interconnect, deg
Y10(6)	LSG	STICKZ2	Lateral stick gradient
Y10(7)	LSTRFCS	LSTRFCS	Lateral stick to RFCS

Table 7.10. AutoCode subroutine USR10 states

Array	Forcing Function	Trim Equations
X10(1)	-	X10(1) = XD10(1)
X10(2)	U10(3), PS	X10(2)=0.87561728·U10(3)
X10(3)	U10(3), PS	X10(3)=0.43780864·U10(3)
X10(4)	U10(3), PS	X10(4)=4.828512·U10(3)
X10(5)	U10(3), PS	X10(5)=4.828512·U10(3)
X10(6)	-	X10(6) = XD10(6)
X10(7)	Y10(6), LSG	X10(7)=6.8685241·Y10(6)
X10(8)	Y10(6), LSG	X10(8)=6.8685241·Y10(6)
X10(9)	-	X10(9) = XD10(9)
X10(10)	-	X10(10) = XD10(10)
X10(11)	-	X10(11) = XD10(11)
X10(12)	-	X10(12) = XD10(12)
X10(13)	-	X10(13) = XD10(13)
X10(14)	-	X10(14) = XD10(14)
X10(15)	U10(9), AZS	X10(15) = 3.6050000·U10(9)

Table 7.11. AutoCode subroutine USR11 inputs

Array	*f18harv* Name	Local Name	Definition
U11(1)	DRPRFCS	RPEDAL	Pilot rudder pedal force after 701E aliasing filter, lbs
U11(2)	QCIS	QC	Sensor-model impact pressure, psf
U11(3)	PSTS	PS	Sensor-model static pressure, psf
U11(4)	RIS	R1	Sensor-model pressure ratio, n.d.
U11(5)	ALFS	AOA	Selected angle of attack, deg (from USR01)
U11(6)	RSRI	RSRI	Rolling surface to rudder interconnect, deg (from USR10)
U11(7)	PS	PSEN	Sensor-model roll rate, degrees/sec
U11(8)	AYS	AYSEN	Sensor-model lateral acceleration, g's
U11(9)	RS	RSEN	Sensor-model yaw rate, degrees/sec
U11(10)	QS	QSEN	Sensor-model pitch rate, degrees/sec
U11(11)	TRMMING	TRMMING	Trim flag (REAL) - 0.0 = operate, 1.0 = trim

Table 7.12. AutoCode subroutine USR11 outputs

Array	*f18harv* Name	Local Name	Definition
Y11(1)	RC	RC	Rudder command, degrees
Y11(2)	RTRI	RTRI	Rudder to roll interconnect
Y11(3)	FY7IN	FY7IN	Filter y7 input
Y11(4)	FY2IN	FY2IN	Filter y2 input
Y11(5)	FY3IN	FY3IN	Filter y3 input
Y11(6)	RPTRFCS	RPTRFCS	Rudder pedal to RFCS

Table 7.13. AutoCode subroutine USR11 states

Array	Forcing Function	Trim Equations
X11(1)	-	X11(1) = XD11(1)
X11(2)	-	X11(2) = XD11(2)
X11(3)	-	X11(3) = XD11(3)
X11(4)	Y11(3), FY7IN	X11(4) = 2.2060000·Y11(3)
X11(5)	-	X11(5) = XD11(5)
X11(6)	Y11(5), FY3IN	X11(6) = 10.062500·Y11(5)
X11(7)	Y11(4), FY2IN	X11(7) = 1.5021104·Y11(4)
X11(8)	Y11(4), FY2IN	X11(8) = 1.5021104·Y11(4)
X11(9)	Y11(4), FY2IN	X11(9) = 0.98260073·Y11(4)
X11(10)	-	X11(10) = XD11(10)
X11(11)	-	X11(11) = XD11(11)
X11(12)	-	X11(12) = XD11(12)
X11(13)	-	X11(13) = XD11(13)
X11(14)	U11(6), RSRI	X11(14) = 5.0625000·U11(6)
X11(15)	-	X11(15) = XD11(15)

Table 7.14. AutoCode subroutine USR18V150P3 inputs

Array	*18harv* Name	Local Name	Definition
U18(1)	ALFPRO	AOAP	Airdata probe angle of attack, degrees
U18(2)	ALFINS	AOAINS	Mission Computer angle of attack, degrees
U18(3)	QCFILT2	QCFILTER2	Sensor-model impact pressure, filtered at 10 rad/sec psf
U18(4)	PSTS	PS	Sensor-model static pressure, psf
U18(5)	RIS	RI	Sensor-model pressure ratio, n.d.
U18(6)	TRMMING	TRMMING	Trim flag (REAL) - 0.0 = operate, 1.0 = trim

Table 7.15. AutoCode subroutine USR18V150P3 outputs

Array	*f18harv* Name	Local Name	Definition
Y18(1)	TEFSC1	TEFSC1	Collective TEF command, degrees
Y18(2)	LEFSC1	LEFSC1	Collective LEF command, degrees
Y18(3)	ALFS	AOA	Selected angle of attack, degrees
Y18(4)	FGI	FGI	Fader gain input for angle of attack, n.d.
Y18(5)	GTILY1	GTILY1	Feedback gain (proportional) for angle of attack
Y18(6)	GTILY2	GTILY2	Feedback gain (proportional) for pitch rate
Y18(7)	GTILY3	GTILY3	Feedback gain (proportional) for load factor
Y18(8)	GTILU1	GTILU1	Feedback gain for filter
Y18(9)	GTILZ1	GTILZ1	Feedback gain for integrator

Table 7.15. Concluded

Array	*f18harv* Name	Local Name	Definition
Y18(10)	RHON1	RHON1	1st variable gain-schedule parameter
Y18(11)	RHON2	RHON2	2nd variable gain-schedule parameter
Y18(12)	RHON3	RHON3	3rd variable gain-schedule parameter
Y18(13)	RHON4	RHON4	4th variable gain-schedule parameter
Y18(14)	RHON5	RHON5	5th variable gain-schedule parameter
Y18(15)	RHON6	RHON6	6th variable gain-schedule parameter
Y18(16)	COSALF	COSALF	Cosine of ALFS, n.d.
Y18(17)	SINALF	SINALF	Sine of ALFS, n.d.

Table 7.16. AutoCode subroutine USR18V150P3 states

Array	Forcing Function	Trim Equations
X18(1)	-	X18(1) = XD18(1)
X18(2)	Y18(4), FGI	X18(2) = 0.8888889·Y18(4)
X18(3)	Y18(3), ALFS	X18(3) = 0.9844237·Y18(3)
X18(4)	Y18(3), ALFS	X18(4) = 0.9689441·Y18(3)

State coefficients 2 to 4 are only used when the controller is trimming and are calculated for a sampling period of 0.025 second.

Table 7.17. AutoCode subroutine USR19V150P4 inputs

Array	*f18harv* Name	Local Name	Definition
U19(1)	PSTICK	PSTICK	Pilot longitudinal-stick position, inches
U19(2)	PTRIM	PTRIM	Pilot pitch-trim-button command, inches
U19(3)	OBESPIT	OBES_LONST	OBES Longitudinal input, inches
U19(4)	ALFS	AOA	Selected angle of attack, degrees
U19(5)	QS	Q	Sensor-model pitch rate, degrees/sec
U19(6)	AZS	NZ	Sensor-model normal acceleration, g's
U19(7)	QCFILT1	QCFILTER1	Sensor-model impact pressure filtered at 2.5 rad/sec, psf
U19(8)	QCFILT2	QCFILTER2	Sensor-model impact pressure filtered at 10 rad/sec, psf
U19(9)	PSTS	PS	Sensor-model static pressure, psf
U19(10)	GTILY1	GTILY1	Feedback gain (proportional) for angle of attack
U19(11)	GTILY2	GTILY2	Feedback gain (proportional) for pitch rate
U19(12)	GTILY3	GTILY3	Feedback gain (proportional) for load factor
U19(13)	GTILU1	GTILU1	Feedback gain for filter
U19(14)	GTILZ1	GTILZ1	Feedback gain for integrator
U19(15)	VTS	VT	Sensor-model atmosphere velocity, ft/sec
U19(16)	COSALF	COSALF	Cosine of angle of attack, n.d. (from USR18V150P3)
U19(17)	SINALF	SINALF	Sensor-model sine of angle of attack, n.d. (from USR18V150P3)
U19(18)	COSTHE	COSTHE	Sensor-model cosine of the pitch angle, n.d.
U19(19)	SINTHE	SINTHE	Sensor-model sine of the pitch angle, n.d.
U19(20)	COSPHI	COSPHI	Sensor-model cosine of the roll angle, n.d.

Table 7.17. Concluded.

Array	*f18harv* Name	Local Name	Definition
U19(21)	DELSTM	DELSTM	Input to trim algorithm that is the sum of collective stabilator & pitch-jet angle command used to initialize the forward loop integrator; deg
U19(22)	TRMMING	TRMMING	Trim flag (REAL) - 0.0 = operate, 1.0 = trim

Table 7.18. AutoCode subroutine USR19V150P4 outputs

Array	*f18harv* Name	Local Name	Definition
Y19(1)	CSCT7	SBPAC1	Collective stabilator command to 701E, degrees
Y19(2)	PJETAC	TVSC	Pitch thrust-vectoring command, degrees
Y19(3)	PSGTOT	PSGTOT	Total stick command, inches
Y19(4)	YCMD	YCMD	Command to the feedback control
Y19(5)	AOATR	AOATR	Estimated angle-of-attack trim, degrees
Y19(6)	ERRORS	DY	Error in regulated variable
Y19(7)	DELY	DELY	Error in regulated variable limited to 5.0. Used to trim stk position.
Y19(8)	QCOMP	QCOMP	Compensated pitch rate, degrees/sec
Y19(9)	QCOMP1	QCOMP1	Gravity compensation in pitch rate signal
Y19(10)	VBRK1	VBRK1	Rate command for stabilator, degrees/sec
Y19(11)	UK1	UK1	Control variable for stabilator command, degrees
Y19(12)	UME11	UME11	Feedforward control variable, degrees/sec
Y19(13)	KCGT	KCGT	Feedforward gain

Table 7.19. AutoCode subroutine USR19V150P4 states

Array	Forcing Function	Trim Equations
X19(1)	-	X19(1) = XD19(1)
X19(2)	-	X19(2) = XD19(2)
X19(3)	-	X19(3) = XD19(3)
X19(4)	-	X19(4) = XD19(4)
X19(5)	-	X19(5) = XD19(5)
X19(6)	-	X19(6) = XD19(6)
X19(7)	-	X19(7) = XD19(7)
X19(8)	Y19(12), QCOMP1	X19(8) = 0.8421053·Y19(9)
X19(9)	U19(5), Q	X19(9) = 1.3459·U19(15)
X19(10)	U19(5), Q	X19(10) = 1.3459·U19(15)
X19(11)	U19(6), NZ	X19(11) = 18.656716·U19(6)
X19(12)	U19(6), NZ	X19(12) = 18.656716·U19(6)
X19(13)	U19(6), NZ	X19(13) = 0.4326·U19(6)
X19(14)	U19(6), NZ	X19(14) = 0.4326·U19(6)
X19(15)	U19(6), NZ	X19(15) = 0.4326·U19(6)
X19(16)	U19(6), NZ	X19(16) = 0.4326·U19(6)
X19(17)	Y19(1), SBPAC1	X19(17) = 0.993785·Y19(1)

States 8 to 11 are only used when the controller is trimming and are calculated for a sampling period of 0.0125 second.

Table 7.20. AutoCode subroutine USR03 inputs

Array	*f18harv* Name	Local Name	Definition
U3(1)	DAPRFCS	LATST_IN	Pilot lateral-stick position after 701E demod filter, inches
U3(2)	AYS*	NY_G	Sensor-model lateral acceleration, g's
U3(3)	PS	PS_DPS	Sensor-model roll rate, degrees/sec
U3(4)	RS*	RS_DPS	Sensor-model yaw rate, degrees/sec
U3(5)	BDS*	BDOTINERT_DPS	Sensor-model inertial sideslip rate component, degrees/sec
U3(6)	DRPRFCS*	RUDPED_LBS	Pilot rudder-pedal force after 701E aliasing filter, lbs
U3(7)	ALFS	AOA_DEG	Selected angle of attack, degrees
U3(8)	QS	QS_DPS	Sensor-model pitch rate, degrees/sec
U3(9)	OBESLAT	OBES_LATST	OBES lateral-axis frequency-sweep input, inches
U3(10)	QCFILT1	QC_PSF	Sensor-model impact pressure filtered at 2.5 rad/sec, psf
U3(11)	COSTHE	COSTHETA	Sensor-model cosine of Euler pitch angle
U3(12)	LGTHR	FGTOTL_LBS	Left-engine gross thrust estimate, lbs
U3(13)	RGTHR	FGTOTR_LBS	Right-engine gross thrust estimate, lbs
U3(14)	YTRIM	YTRIM	Pilot yaw-trim-knob command, degrees
U3(15)	RTRIM	RTRIM	Pilot roll-trim-button command, inches
U3(16)	MACHS	MACH	Sensor-model Mach number, n.d.
U3(17)	H	H_FT	Altitude (above sea level), ft
U3(18)	XSTRAKE	XSTRAKE	REAL switch to activate nose-strake control law (0., 1., or 2.)
U3(19)	AZS*	NZ_G	Sensor-model normal acceleration, g's
U3(20)	CSCT7	SYMSTAB_DEG	Collective stabilator command to 701E, degrees
U3(21)	VTS	VTRUE_FPS	Sensor-model atmosphere velocity, fps
U3(22)	SINPHI	SINPHI	Sensor-model sine of Euler roll angle, n.d.
U3(23)	OBESRUD	OBES_RUDPED	OBES directional axis frequency sweep input, lbs
U3(24)	SINALF	SINALPHA	Sine of ALFS, n.d. (from USR18V150P3)
U3(25)	COSALF	COSALPHA	Cosine of ALFS, n.d. (from USR18V150P3)
U3(26)	-	STKNLG	Strake non-linear gearing option (Set = 1.0)
U3(27)	XTVYAW	XTVYAW	REAL switch to activate yaw thrust-vectoring control law (0. or 1.)

* Note that these inputs are averaged. For example, in the equation:

AYS = (AYS + AYS_PAST)/2.

AYS is averaged with it's previous value since AYS is a 40 Hz input. This averaging occurs as part of the setup for the input array U3.

Table 7.21. AutoCode subroutine USR03 outputs

Array	*f18harv* Name	Local Name	Definition
Y3(1)	RSCT7	DD_LIM_DEG	Roll stabilator command to 701E, degrees
Y3(2)	ACT7	DA_LIM_DEG	Aileron command to 701E, degrees
Y3(3)	RLEFCT7	DL_DEG	Roll LEF command to 701E, degrees
Y3(4)	RTEFCT7	DF_DEG	Roll TEF command to 701E, degrees
Y3(5)	RCT7	DR_LIM_DEG	Rudder command to 701E, degrees
Y3(6)	YJETAC	TVYAW_MP	Thrust-vector yaw-jet-angle command, deg
Y3(7)	VROLL	VROLL	Roll pseudo control, n.d.
Y3(8)	VYAW	VYAW	Yaw pseudo control, n.d.

Table 7.21. Concluded.

Array	*f18harv* Name	Local Name	Definition
Y3(9)	LAT_CMD_RPS2	LAT_CMD_RPS2	Lateral command, rad/sec^2
Y3(10)	DIR_CMD_RPS2	DIR_CMD_RPS2	Directional command, rad/sec^2
Y3(11)	LATST_CMD	LATST_CMD	Lateral stick command, n.d.
Y3(12)	RJETAC	TVROLL_MP	Thrust-vector roll-jet-angle command, deg
Y3(13)	NY_ADJ_G	NY_ADJ_G	Adjusted lateral acceleration, g's
Y3(14)	RSTABCOR_DPS	RSTABCOR_DPS	Compensated RSTAB, degrees/sec
Y3(15)	CNSR	FSRC_LIM_DEG	Right nose-strake command, degrees
Y3(16)	CNSL	FSLC_LIM_DEG	Left nose-strake command, degrees
Y3(17)	NABYNTV	NABYNTV	Yaw moment available, n.d.
Y3(18)	LABYLTV	LABYLTV	Roll moment available, n.d.
Y3(19)	GCOMP_RPS	GCOMP_RPS	Gravity compensation, rad/sec
Y3(20)	THRUST_LBS	THRUST_LBS	Total thrust, lbs
Y3(21)	AYCORR_G	AYCORR_G	Lateral accelerometer correction, g's
Y3(22)	PDSMAX	PDSMAX	Lateral stick command gain, n.d.
Y3(23)	STVYAW	STVYAW	Yaw thrust-vectoring engage, n.d.
Y3(24)	SFSYAW	SFSYAW	Differential strake engage, n.d.
Y3(25)	TVYAW_AUTO	TVYAW_AUTO	PsC Yaw thrust-vectoring control flag (=0).
Y3(26)	FS_ON_LIM	FS_ON_LIM	PsC Differential-strake control flag (=0).
Y3(27)	FS_DEPLOY_LIM	FS_DEPLOY_LIM	PsC Strake symmetrric deployment control flag (=0).
Y3(28)	BDOTINERT_DPS	BDOTINERT_DPS	Sensor-model inertial sideslip-rate component, degrees/sec
Y3(29)	BDOT_DPS	BDOT_DPS	RFCS sideslip rate, degrees/sec

Outputs 7 through 14 and 17 through 29 are internal Lateral/Directional Control Law variables used for performance monitoring and system diagnostics.

Table 7.22. AutoCode subroutine USR03 states

Array	Forcing Function	Equation - TRIM
X3(1)	-	X3(1) = XD3(1)
X3(2)	-	X3(2) = XD3(2)
X3(3)	-	X3(3) = U3(27)
X3(4)	-	X3(4) = U3(18)
X3(5)	-	X3(5) = XD3(5)
X3(6)	-	X3(6) = XD3(6)
X3(7)	-	X3(7) = XD3(7)
X3(8)	-	X3(8) = XD3(8)
X3(9)	-	X3(9) = XD3(9)
X3(10)	-	X3(10) = XD3(10)
X3(11)	-	X3(11) = XD3(11)
X3(12)	-	X3(12) = XD3(12)
X3(13)	-	X3(13) = XD3(13)
X3(14)	-	X3(14) = XD3(14)
X3(15)	-	X3(15) = XD3(15)
X3(16)	-	X3(16) = XD3(16)
X3(17)	-	X3(17) = XD3(17)

Table 7.22. Concluded.

Array	Forcing Function	Equation - TRIM
X3(18)	-	X3(18) = XD3(18)
X3(19)	-	X3(19) = XD3(19)
X3(20)	-	X3(20) = XD3(20)
X3(21)	-	X3(21) = XD3(21)
X3(22)	-	X3(22) = XD3(22)
X3(23)	-	X3(23) = XD3(23)
X3(24)	-	X3(24) = XD3(24)
X3(25)	-	X3(25) = XD3(25)
X3(26)	-	X3(26) = XD3(26)
X3(27)	-	X3(27) = XD3(27)
X3(28)	-	X3(28) = XD3(28)
X3(29)	-	X3(29) = XD3(29)
X3(30)	-	X3(30) = XD3(30)
X3(31)	-	X3(31) = XD3(31)
X3(32)	-	X3(32) = XD3(32)

Table 7.23. AutoCode subroutine USR13 inputs

Array	*f18harv* Name	Local Name	Definition
U13(1)	CSCT7	CSCT7	Collective stabilator command to 701E, degrees (USR01 or USR19V150P4)
U13(2)	RSCT7	RSCT7	Roll stabilator command to 701E, deg (USR02 or USR03)
U13(3)	CLEFCT7	CLEFCT7	Collective LEF command to 701E, degrees (USR01 or USR19V150P4)
U13(4)	RLEFCT7	RLEFCT7	Roll LEF command to 701E, degrees (USR02 or USR03)
U13(5)	CTEFCT7	CTEFCT7	Collective TEF command to 701E, degrees (USR01 or USR19V150P4)
U13(6)	RTEFCT7	RTEFCT7	Roll TEF command to 701E, degrees (USR02 or USR03)
U13(7)	TRMMING	TRMMING	Trim flag (REAL) - 0.0 = operate, 1.0 = trim

Table 7.24. AutoCode subroutine USR13 outputs

Array	*f18harv* Name	Local Name	Definition
Y13(1)	DELTSCR	RSC	Right stabilator command, degrees
Y13(2)	DELTSCL	LSC	Left stabilator command, degrees
Y13(3)	DELTNCR	RLEFC	Right LEF command, degrees
Y13(4)	DELTNCL	LLEFC	Left LEF command, degrees
Y13(5)	DELTFCR	RTEFC	Right TEF command, degrees
Y13(6)	DELTFCL	LTEFC	Left TEF command, degrees

Table 7.25. AutoCode subroutine USR13 States

Array	Forcing Function	Trim Equations
X13(1)	-	X13(1) = XD13(1)
X13(2)	-	X13(2) = XD13(2)

7.4 Trim

The TRIM algorithm and procedure defined in reference 1.0 have remained the same in the *f18harv* simulation. When trimming the FORTRAN AutoCode implementation of RFCS, the following items need to be noted: During the simulation TRIM mode, the state array X(XN) is updated using trim equations which define the steady-state solution of the state at the given trim condition. After the simulation TRIM mode is complete, and during the simulation integration after trim, the the state array X(XN) is set equal to the time derivative of the state, X(XN) = XD(XN).

During trim for either the NASA-0 RFCS or ANSER RFCS the internal AutoCode input to the forward-loop-integrator state variable ERRORS is driven to zero. ERRORS, which determines both the collective stabilator and pitch-jet-angle command, is an output from RFCS used in the trim algorithm. During trim the algorithm varies ERRORS and the pilot longitudinal-stick position until ERRORS is zero (within a tolerance). When ERRORS is zero, the resulting pilot longitudinal-stick position is the trim position.

During trim the output of the integrator DELSTM is set equal to the sum of the collective stabilator and pitch-jet-angle command and is driven by the trim algorithm until the aircraft pitch acceleration $\dot{q}$ is zero. The parameter DELSTM is varied in the trim algorithm and used as an input to RFCS, where it is substituted for the output from the integrator during trim. The logical LTRFLG is used to determine the state of trim in ACSL. When LTRFLG is TRUE, both pilot longitudinal-stick position and DELSTM are varied until ERRORS and $\dot{q}$ are each approximately zero. When this occurs, trim is achieved, and LTRFLG changes to FALSE. The input DELSTM is then no longer used inside AutoCode and is replaced by the output from the integrator.

7.5 OBES Frequency Sweep Inputs

The F/A-18 HARV aircraft has an On-Board Excitation System (OBES) which superimposes computer-generated commands on either RFCS pilot commands or control-surface commands via separate OBES inputs. The OBES consists of a function generator with generic logic built in for generating both frequency-sweep commands and doublets. The OBES function generator logic has been implemented in subroutine HARVCSM to allow frequency-sweep inputs to be generated in the *f18harv* simulation. Frequency-sweep inputs can be generated for ANSER RFCS and applied to the longitudinal, lateral, or directional pilot command paths. The inputs can be applied to all three axes at once, individually, or any combination of the three axes. The following input parameters need to be set to define a frequency-sweep input.

STARTFQ	- start frequency, radians/sec
STOPFQ	- stop frequency, radians/sec
STARTAP	- start amplitude, inches for stick input, lbs for pedal input
STOPAP	- stop amplitude, inches for stick input, lbs for pedal input
DURAT	- frequency-sweep duration time, seconds

The LOGICALS called LOBESP, LOBESR, and LOBESY need to be set to TRUE (default = FALSE) to select a frequency-sweep input for the longitudinal, lateral, or directional axes, respectively. The user can select either a fixed frequency and amplitude input or a linear time varying frequency and amplitude input. A fixed-frequency input requires the user to set the start and stop frequencies STARTFQ and STOPFQ to the same frequency. A linear time-varying frequency and amplitude input requires the user to set STARTFQ and STOPFQ

to the desired start and stop frequencies, respectively, and to set DURAT to the desired time duration of the sweep. Similarly, the user would select a start and stop amplitude, STARTAP and STOPAP, and a duration DURAT to implement a linear time-varying amplitude. If a fixed amplitude is desired, both STARTAP and STOPAP would need to be set to the same value.

The following code represents the frequency and amplitude setup for the frequency-sweep generator in the subroutine HARVCSM:

$$\begin{aligned} \text{OMEGAR} &= (\text{STOPFQ} - \text{STARTFQ})/\text{DURAT} \\ \text{AMPLITR} &= (\text{STOPAP} - \text{STARTAP})/\text{DURAT} \\ \text{OMEGA} &= 0.5 \cdot (T \cdot \text{OMEGAR}) + \text{STARTFQ} \\ \text{AMPLIT} &= (T \cdot \text{AMPLITR}) + \text{STARTAP} \end{aligned} \tag{7.3}$$

where: T = Simulation time, seconds.

Depending on the LOGICALS LOBESP, LOBESR, and LOBESY, the frequency-sweep is generated for the pitch, roll, and yaw axes as follows.

$$\text{OBESPIT, OBESLAT, OBESRUD} = \text{AMPLIT} \cdot \sin(\text{OMEGA} \cdot T) \tag{7.4}$$

The frequency-sweep generator outputs OBESPIT, OBESLAT, and OBESRUD are used as inputs to the ANSER RFCS.

7.6 Output Processing in HARVCSM

Rudder Rate Limit

The 701E RFCS output rudder commands are calculated at 80 Hz and sent to DPRAM and then the 701E at 80 Hz. The 701E rudder-command path is updated at 40 Hz and summed with an averaged 80 Hz RFCS rudder command. The previous and current frame RFCS rudder commands are averaged. The intent of this averaging is to reduce the delay of the RFCS rudder command to one 80 Hz frame since the RFCS rudder command will only be used at 40 Hz. The *f18harv* simulation does not model a 701E control system, but the effect of the 40 to 80 Hz averaging is modeled in HARVCSM to account for the one 80 Hz frame delay. An example of the code in HARVCSM for the left-rudder command is given below:

```
IF (LOOP40 .EQ. 1) THEN
   RCLAVG = (DELTRCL - RCLLAST)/2.                    (7.5)
   RCLLAST = DELTRCL
ENDIF

DELTRCL* = DELTRCL* + RCLAVG
```

where:

DELTRCL	= RFCS Left Rudder Command (80 Hz)
RCLLAST	= Previous RFCS Left Rudder Command (40 Hz)
DELTRCL*	= Averaged RFCS Left Rudder Command (80 Hz). In the 701E, this parameter equals the 701E left rudder command (40 Hz).

The final left and right RFCS rudder commands are rate limited to 75 deg/sec in subroutine

HARVCSM before being passed to theACSL portion of the *f18harv* simulation.

Aileron Trailing Edge Down Limit

A position limit on the down deflection for the trailing-edge flap is included in HARVCSM. In the flight ADA code this limit is applied before the aileron command is separated into left and right commands and sent to DPRAM. Since this limit exists outside of the ANSER RFCS control law in the flight code, this limit was added to HARVCSM, which is the closest representation of the F/A-18 HARV system available in the *f18harv* simulation. A block diagram for this limit can be found in Figure 7.2.

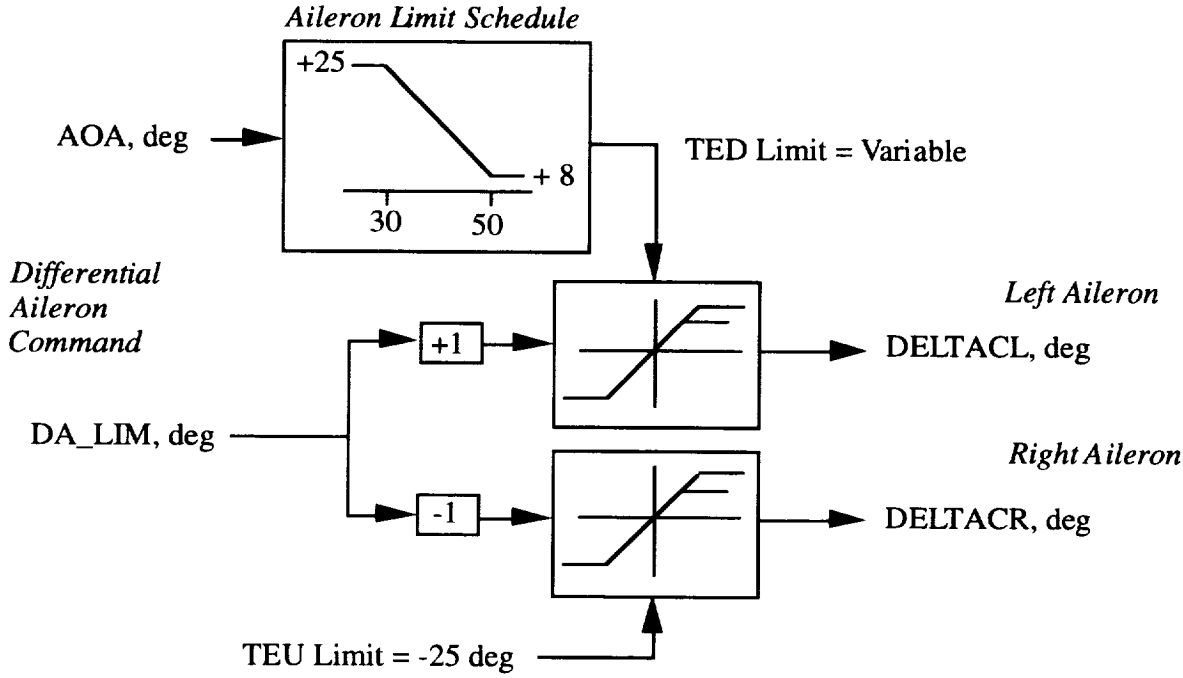


Figure 7.2. Aileron trailing-edge-down limit

8.0 OTHER MODIFICATIONS

With any simulation of this size and complexity, many modifications have been made as a result of analysis and debugging. The table below lists items that were either corrected or changed from the *f18bas* simulation and gives a brief description of the problem or reason for change.

Table 8.1. Modifications to *f18bas* simulation

Index Reference	Reason for Change	Description
8.1	New Capability	Input winds into the simulation
8.2	New Capability	Output files written for $MATRIX_X$ and Getdata

8.1 Winds

The ability to generate steady-state winds in the simulation is included in addition to corrections for computing the time rate-of-change of the body-axis components of wind. The following inputs can be assigned values in the ACSL command file to set initial wind conditions for a time-history simulation.

VWNDIC = Initial wind velocity (ft/sec)
VWNDRT = d/dh [wind velocity] × 1000 (ft/sec/kft)
HDWIC = Initial wind heading (deg)
HDWRT = d/dh [wind heading] × 1000 (deg/kft)

The wind velocity and heading are formed by the following equations in the DERIVATIVE section of ACSL, where H is defined as altitude in feet:

$$\left.\begin{aligned} \text{VWND} &= \text{VWNDIC} + \text{VWNDRT}(\text{H} - 15000)/1000 \\ \text{HEADWND} &= \text{HDWIC} + \text{HDWRT}(\text{H} - 15000)/1000 \end{aligned}\right\} \tag{8.1}$$

The wind velocity component in the x, y, and z Earth-axis-frame directions are called XDWG, YDWG, and ZDWG, respectively, in units of ft/sec. Both the velocity and accelerations are formed using VWND and HEADWND calculated above.

$$\left.\begin{aligned} \text{XDWG} &= \text{VWND}\cos\left[\text{HEADWND}\left(\frac{\pi}{180}\right)\right] \\ \text{YDWG} &= -\text{VWND}\sin\left[\text{HEADWND}\left(\frac{\pi}{180}\right)\right] \\ \text{ZDWG} &= 0.0 \end{aligned}\right\} \tag{8.2}$$

The wind acceleration components in the x, y, and z Earth-axis-frame directions are called XDDWG, YDDWG, and ZDDWG, respectively, in units of ft/sec^2.

$$
\begin{aligned}
\text{XDDWG} &= \frac{(\text{VWNDRT})(\text{HD})}{1000}\cos\left[\text{HEADWND}\left(\frac{\pi}{180}\right)\right] \\
&\quad - \text{VWND}\sin\left[\text{HEADWND}\left(\frac{\pi}{180}\right)\right]\left[\text{HDWRT}\left(\frac{\pi}{180}\right)\left(\frac{\text{HD}}{1000}\right)\right] \\
\text{YDDWG} &= -\frac{(\text{VWNDRT})(\text{HD})}{1000}\sin\left[\text{HEADWND}\left(\frac{\pi}{180}\right)\right] \\
&\quad - \text{VWND}\cos\left[\text{HEADWND}\left(\frac{\pi}{180}\right)\right]\left[\text{HDWRT}\left(\frac{\pi}{180}\right)\left(\frac{\text{HD}}{1000}\right)\right] \\
\text{ZDDWG} &= 0.0
\end{aligned}
\tag{8.3}
$$

These aerodynamic disturbances are then added to the total body-axis inertial velocity in the x, y, and z directions to form the components of airspeed.

$$
\left.
\begin{aligned}
\text{UWG} &= \text{CXX}\cdot\text{XDWG} + \text{CXY}\cdot\text{YDWG} + \text{CXZ}\cdot\text{ZDWG} \\
\text{UWG} &= \text{CYX}\cdot\text{XDWG} + \text{CYY}\cdot\text{YDWG} + \text{CYZ}\cdot\text{ZDWG} \\
\text{UWG} &= \text{CZX}\cdot\text{XDWG} + \text{CZY}\cdot\text{YDWG} + \text{CZZ}\cdot\text{ZDWG}
\end{aligned}
\right\}
\tag{8.4}
$$

The terms CXX, CXY, CXZ, CYX, CYY, CYZ, CZX, CZY, and CZZ correspond to the direction cosines in the x, y, and z component of unit x, y, and z Earth frame. Total body-axis inertial accelerations were modified in the *f18harv* simulation to include terms that were missing in the *f18bas* simulation. These terms are included in equation (8.5).

$$
\left.
\begin{aligned}
\text{UDWGP} &= \text{R}\cdot\text{VWG} - \text{Q}\cdot\text{WWG} \\
&\quad + \text{CXXD}\cdot\text{XDWG} + \text{CXYD}\cdot\text{YDWG} + \text{CXZD}\cdot\text{ZDWG} \\
\text{UDWGP} &= -\text{R}\cdot\text{UWG} + \text{P}\cdot\text{WWG} \\
&\quad \text{CYXD}\cdot\text{XDWG} + \text{CYYD}\cdot\text{YDWG} + \text{CYZD}\cdot\text{ZDWG} \\
\text{UDWGP} &= \text{Q}\cdot\text{UWG} - \text{P}\cdot\text{VWG} \\
&\quad \text{CZXD}\cdot\text{XDWG} + \text{CZYD}\cdot\text{YDWG} + \text{CZZD}\cdot\text{ZDWG}
\end{aligned}
\right\}
\tag{8.5}
$$

The terms CXXD, CXYD, CXZD, CYXD, CYYD, CYZD, CZXD, CZYD, and CZZD correspond to the rate of change of the direction cosines in the x, y, and z component of unit x, y, and z Earth frame.

8.2 Output Capability

A provision for outputing data in addition to that available with the standard ACSL output capability was added to the simulation. This new capability allows the simulation user to write an output file in compressed binary (cmp3) format which is compatible with Getdata and Xplot for post processing. Both the Getdata and Xplot programs were obtained from DFRC. Getdata is a FORTRAN utility program for time history data and is well documented in reference 8.0. Xplot and Quickplot are easy to use XY-plotting time-history applications which comes with user document, tutorial, and reference manuals (ref. 8.1 and 8.2). The LOGICAL flag LWRGD needs to be set to TRUE (Default is FALSE) in the ACSL command file (.cmd) to invoke creation of a cmp3 output file called *acsl_harv.cmp3*. The output file *acsl_harv.cmp3*

will be placed in the directory in which ACSL was run. The compressed binary file is on the order of 5 times smaller that the ASCII formats created using the standard ACSL output commands allowing the user to conserve disk space when making ACSL runs. The compressed formatted file can be read into Xplot directly and is the quickest format for reading data. The integer variable NUMOUT should be set in the ACSL command file. NUMOUT specifies the frequency with which data will be written to the output file (Default = 4). For example, if the user wants data written for every tenth sample period (0.0125x10 or 0.125 seconds), the user would need to set NUMOUT to 10, and the simulation would write data to *acsl_harv.cmp3* every 0.125 seconds. The default value of NUMOUT writes to the output files every 0.05 seconds. Data can further be thinned during post-processing using Getdata commands discussed in reference 8.0.

The output parameter list written to *acsl_harv.cmp3* contains general aircraft state parameters of interest along with either NASA-0 or ANSER RFCS specific parameters. The output files created will change by the outputs defined in section 8.3 depending on the RFCS selected with the logical LNRFCS. The subroutine that writes *acsl_harv.cmp3* is called CMPFM and can easily be modified to add additional variables. Of course, modifications necessitate recompiling the *f18harv* simulation.

A second output capability was added to ACSL to write a MATRIX$_X$®-compatible output file. This capability is invoked by setting LWRMTX to TRUE (Default is FALSE) in the ACSL command file causing creation of an output file called *MTXACSL.DAT*. This file can be read into MATRIX$_X$® and plotted or manipulated for analysis. Again the output parameter list is fixed but can be easily modified. The subroutines associated with this capability are called MTXWRF and MTXFM. The option flag NUMOUT is utilized in the same matter as described before.

For both *acsl_harv.cmp3* and *MTXACSL.ASC* the files created will be overwritten on the next ACSL run. If the user wants the output file saved, the file must be renamed before the next run is made.

9.0 CONCLUDING REMARKS

This report together with reference 1.0 describes an ACSL-based six-degree-of-freedom nonlinear batch simulation of the F/A-18 HARV. The HARV is an F/A-18 modified to incorporate pitch and yaw thrust vectoring and to incorporate actuated forebody strakes for increased yaw control at high angles of attack. Because of its flexibility and extensive flight envelope, this simulation is well suited to support advanced control law research at high angles of attack. The aero database covers the angle-of-attack range from -10° to +90°. The user can choose between different control laws and Mixer/Predictors and can turn sensor models and elastic deformation effects "on" and "off". A FORTRAN real-time version of the simulation is implemented in Langley's Differential Maneuvering Simulator for piloted simulation research.

The aerodynamic model used in this simulation is based on wind tunnel results, and the model has not been flight validated.

10.0 REFERENCES

1.0 Buttrill, C. S.; Arbuckle, P. D.; and Hoffler, K. D.: *Simulation Model of a Twin-Tail, High Performance Airplane.* NASA TM-107601, 1992.

1.1 *Advanced Continuous Simulation Language (ACSL) Reference Manual.* Edition 10.0. Mitchell and Gauthier Associates, 1991.

3.0 Bowers, A. H.; HARV Aerodynamic Model. HARV project memo NASA Dryden Flight Research Center, 22 January 1990.

3.1 Murri, D. G.; Biedron, R. T.; Erickson, G. E.; Jordan, F. L.; and Hoffler, K. D.: Development of Actuated Forebody Strake Controls for the F/A-18 High Alpha Research Vehicle. *NASA High Angle-of-Attack Technology Conf*erence. NASA CP-3149, Vol I, pp. 335-380.

3.2 Murri, D. G.; and Rao, D. M.: *Exploratory Studies of Actuated Forebody Strakes for Yaw Control at High Angles of Attack.* AIAA-87-2557-CP, August, 1987.

3.3 Erickson, G. E.; and Murri, D. G.: *Forebody Strakes for High Angle-of-Attack Vortex Flow Control - Mach Number and Strake Planform Effects.* NASA High Angle-of-Attack Technology Conference, NASA CP-3149, Vol I, pp. 380-480.

3.4 Biedron, R. T.; and Thomas, J. L.: Navier-Stokes Computations for an F/A-18 Forebody With Actuated Control Strake. NASA High Angle-of-Attack Technology Conference, NASA CP-3149, Vol I, pp. 481-506.

4.0 McDonnell Aircraft Company, "NASA High Alpha Research Vehicle Thrust Vectoring System (HARV-TVCS)", NASA Contract NAS2-12661; Theory and Implementation of the HARV/TVCS Mixer Predictor; August 1989.

4.1 Pahle, J. W.; Bundick, W. T.; Beissner, F. L.; and Yeager, J. C.: *Design of a Mixer for the Thrust-Vectoring System on the High-Alpha Research Vehicle.* NASA TM-110228, January 1996.

5.0 Memorandum 341-84-152, *F/A-18 Air Data Estimation for Analytical Simulations*, Issue date 24 Dec 1984, McDonnell Aircraft Company.

5.1 Carter, John: NASA Dryden Flight Research Center Internal Memorandum to F/A-18 HARV Project, *Proposed Improvements to F/A-18 HARV Flow Measurements*, 12 June 1992.

6.0 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

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

6.2 MDC A8450 - Volume II, Addendum I, *F/A-18 Simulation Software Support Program*, Issue date: 15 December 1983, Revision Letter A, McDonnell Aircraft Company

7.0 McDonnell Aircraft Company, *NASA High Alpha Research Vehicle Thrust Vectoring*

System (HARV-TVCS), NASA Contract NAS2-12661; Control Law Design Specification Document; 9 March 1989.

7.1 HARV Control Law Design Team: *Design Specification for a Thrust-Vectoring, Actuated-Nose-Strake Flight Control Law for the High-Alpha Research Vehicle*, NASA TM-110217, May 1996.

7.2 Ostroff, A.J.; and Proffitt, M. S.: *Longitudinal-Control Design Approach for High-Angle of-Attack Aircraft*;. NASA TP-3302, 1993.

8.0 Maine, R. E.: *Manual for GetData* , Version 3.1. NASA TM-88288, 1987.

8.1 Vernon, T. H., PRC Inc.: *Xplot.. Version 3.06 Users Guide and Command Reference.*, NASA Dryden Flight Research Center, 23 December 1992

8.2 Vernon, T. H. and Apropos Software & Associates: *Quickplot User's Guide and Command Reference*. Version 1.04, August 1994.

11.0 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)**	BLCG	CSL
BL_{rcg}	buttock line coordinate of reference c.g., + right, in. **(Default = 0.0)**		
BL_{INS}	buttock line coordinate of INS, + right, in. **(Default = -13.36)**	BLINS	INSMDL
BL_{rf}	buttock line of aerodynamic reference center, + right, in. **(Default = 0.0)**	BLRF	CSL
$\bar{c}$	Mean aerodynamic chord	CBAR	CSL
c.g.	center-of-gravity, in.		
$\cos\theta$	cosine of pitch angle	CSTHE	CSL
$\cos\theta_S$	sensor-model cosine of pitch angle	COSTHE	SENSOR
$\cos\phi$	cosine of roll angle	CSPHI	CSL
$\cos\phi_S$	sensor-model cosine of roll angle	COSPHI	SENSOR
$C_{X_{sf_0}}$	All steady aero increments except ANSER		
$C_{X_{sf}}$	Steady aero coefficient for the axis denoted by X		
C_{xx}	direction cosine, x body component of x Earth frame vector	CXX	CSL
C_{xy}	direction cosine, x body component of y Earth frame vector	CXY	CSL
C_{xz}	direction cosine, x body component of z Earth frame vector	CXZ	CSL
C_{yx}	direction cosine, y body component of x Earth frame vector	CYX	CSL
C_{yy}	direction cosine, y body component of y Earth frame vector	CYY	CSL
C_{yz}	direction cosine, y body component of z Earth frame vector	CYZ	CSL
C_{zx}	direction cosine, z body component of x Earth frame vector	CZX	CSL
C_{zy}	direction cosine, z body component of y Earth frame vector	CZY	CSL
C_{zz}	direction cosine, z body component of z Earth frame vector	CZZ	CSL
F_p	maximum deflection in pitch jet deflection	FP	TVCOEF
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 = 456.3)**	FSCG	CSL
FS_{rcg}	fuselage station coordinate of reference c.g., + aft, in.**(Default = 458.56)**		

Symbol	Definition	Variable Name	Program Unit
FS_{INS}	fuselage station coordinate of INS + aft, in. **(Default = 307.95)**	FSINS	SENSOR
FS_{rf}	fuselage station of aerodynamic reference center, + aft, in. **(Default = 458.56)**	FSRF	CSL
F_{x_E}	total x-body component of thrust-induced forces, lbs	FVRE(1)	CSL
F_y	maximum deflection in yaw jet deflection	FY	TVCOEF
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
$Geom_{fac}$	exhaust-nozzle geometry factor	GEOMFAC	TVCOEF
H	altitude, feet	H	CSL
Hg	Mercury		
H_{gcl}	Ground clearance, feet	HGCL	CSL
H_ρ	altitude appropriate for current density, ρ, in the 1962 Standard Atmosphere tables, feet	HRF	CSL
H_S	sensor-model altitude, feet	HS	SENSOR
i.c.	Initial Condition		
K	Non-dimensional rotation rate	WNONDIM	AEROINC
K°	temperature in degrees Kelvin	TMPK	CSL
l_p	distance from c.g. to angle-of-attack probe, feet	XQ	AIRDATA
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,		
M	Mach number	MACH	CSL
M_S	sensor-model Mach number	MACHS	SENSOR
n.d., N.D.	non-dimensional		
n_x	axial acceleration	AX	CSL
$n_{x_{INS}}$	INS axial acceleration, ft/sec^2	RNXINS	INSMDL
n_y	lateral acceleration, g's	AY	CSL
$n_{y_{ACC}}$	accelerometer lateral acceleration, g's	AYACC	ACCEL
$n_{y_{INS}}$	INS lateral acceleration, ft/sec^2	RNYINS	INSMDL

Symbol	Definition	Variable Name	Program Unit
n_{y_S}	sensor-model lateral acceleration, g's	AYS	SENSOR
n_z	vertical acceleration, g's	AZ	CSL
$n_{z_{ACC}}$	accelerometer vertical acceleration, g's	AZACC	ACCEL
$n_{z_{INS}}$	INS vertical acceleration, ft/sec^2	RNZINS	INSMDL
n_{z_S}	sensor-model vertical acceleration, g's	AZS	SENSOR
p	+x component of rotational velocity in the body frame, rad/sec	P	CSL
p	+x component of rotational velocity in the body frame, deg/sec	PDEG	CSL
p_{ACC}	sensor-model +x component of rotational velocity in the body frame, °/sec	XPGY	ACCEL
p_{INS}	INS +x component of rotational velocity in the body frame, rad/sec	PINSS	INSMDL
pnd	Non-dimensional body axis roll rate	PND	NSTAT
P.G.	is the Prandtl-Glauert compressibility correction	TEMP3	TVCOEF
posc	oscillatory component of body-axis roll rate	POSC	NSTAT
p_S	limited sensor-model +x component of rotational velocity in the body frame, °/sec	PS	SENSOR
pwind	Wind-axis roll rate	PSTAB	DERIV.
$\dot{p}$	+x component of rotational acceleration in the body frame, rad/sec^2	PD	CSL
P_s	static pressure (lbs/ft^2)	PSTATC	CSL
$P_{s_{ADC}}$	Airdata static pressure (lbs/ft^2)	PSTS1	AIRDATA
$P_{s_{MC}}$	Mission Computer static pressure (lbs/ft^2)	XPS	AOAINS
P_{s_S}	sensor-model static pressure (inches Hg)	PSTS	SENSOR
q	+y component of rotational velocity in the body frame, rad/sec	Q	CSL
q	+y component of rotational velocity in the body frame, deg/sec	QDEG	CSL
q_{ACC}	sensor-model +y component of rotational velocity in the body frame, °/sec	XQGY	ACCEL
q_{INS}	INS +y component of rotational velocity in the body frame, rad/sec	QINSS	INSMDL
qnd	Non-dimensional body axis pitch rate	QND	NSTAT
qosc	oscillatory component of body-axis pitch rate	QOSC	NSTAT
q_S	limited sensor-model +y component of rotational velocity in the body frame, °/sec	QS	SENSOR
$\bar{q}$	dynamic pressure, psf	QBAR	CSL
$\bar{q}_S$	sensor-model dynamic pressure, psf	QBARSN	SENSOR
$\dot{q}$	+y component of rotational acceleration in the body frame, rad/sec^2	QD	CSL
Q_c	compressible pressure, psf	QC	CSL
$Q_{c_{ADC}}$	Airdata compressible pressure, psf	QCIS1	AIRDATA
$Q_{c_{MC}}$	Mission Computer compressible pressure, (inches Hg)	XQC	AOAINS

Symbol	Definition	Variable Name	Program Unit
Q_{c_S}	sensor-model compressible pressure, psf	QCIS	SENSOR
r	+z component of rotational velocity in the body frame, rad/sec	R	CSL
r	+z component of rotational velocity in the body frame, deg/sec	RDEG	CSL
r_{ACC}	sensor-model +z component of rotational velocity in the body frame, °/sec	XRGY	ACCEL
r_{INS}	INS +z component of rotational velocity in the body frame, rad/sec	RINSS	INSMDL
rnd	Non-dimensional body-axis yaw rate	RND	NSTAT
r_{osc}	oscillatory component of body-axis yaw rate	ROSC	NSTAT
r_S	limited sensor-model +z component of rotational velocity in the body frame, °/sec	RS	SENSOR
$\dot{r}$	+z component of rotational acceleration in the body frame, rad/sec^2	RD	CSL
R_i	pressure ratio (impact/static pressure, Q_C/P_S), n.d.		
$R_{i_{ADC}}$	Airdata pressure ratio, n.d.	RI	AIRDATA
R_{i_S}	sensor-model pressure ratio, n.d.	RIS	SENSOR
$R°$	temperature in degrees Rankin	XTMPR	AOAINS
s	Laplace variable		
$\sin\theta$	sine of pitch angle	SNTHE	CSL
$\sin\theta_S$	sensor-model sine of pitch angle	SINTHE	SENSOR
$\sin\phi$	sine of roll angle	SNPHI	CSL
$\sin\phi_S$	sensor-model sine of roll angle	SINPHI	SENSOR
S	wing reference area used for force and moment coefficients, 400 ft^2	SWRF	CSL
t	time, seconds	T	CSL
T	Temperature, degrees Kelvin	TMPK	CSL
T_l	gross thrust for left engine including effect of T56 bias only, lbs	FG(1)	TLUHRV
T_r	gross thrust for right engine including effect of T56 bias only, lbs	FG(2)	TLUHRV
u	+x component of inertial translational velocity in the body frame, ft/sec	U	CSL
$\dot{u}$	+x component of inertial translational acceleration in the body frame, ft/sec^2	UD	CSL
v	+y component of inertial translational velocity in the body frame, ft/sec	V	CSL
$\dot{v}$	+y component of inertial translational acceleration in the body frame, ft/sec^2	VD	CSL
V	total atmosphere airspeed velocity, ft/sec	VT	CSL
$V_{East_{INS}}$	East component of INS velocity in the Body frame, ft/sec	VEINS	INSMDL
V_{MC}	Mission Computer total atmosphere velocity, ft/sec	XHTAS	AOAINS

Symbol	Definition	Variable Name	Program Unit
$V_{North_{INS}}$	North component of INS velocity in the Body frame, ft/sec	VNINS	INSMDL
V_S	sensor-model total atmosphere velocity, ft/sec	VTS	SENSOR
V_{SOS}	speed of sound, ft/sec	VS	CSL
$V_{SOS_{MC}}$	Mission Computer speed of sound, ft/sec	SOUND	AOAINS
$V_{Vertical_{INS}}$	Vertical component of INS velocity in the Body frame, ft/sec	VVINS	INSMDL
$V_{x_{airmass}}$	+x component of airmass velocity in the body frame, ft/sec	VVTASAC(1)	AOAINS
$V_{y_{airmass}}$	+y component of airmass velocity in the body frame, ft/sec	VVTASAC(2)	AOAINS
$V_{z_{airmass}}$	+z component of airmass velocity in the body frame, ft/sec	VVTASAC(3)	AOAINS
$\dot{V}$	total atmosphere acceleration, ft/sec^2	VTD	CSL
w	+z component of inertial translational velocity in the body frame, ft/sec	W	CSL
$\dot{w}$	+z component of inertial translational acceleration in the body frame, ft/sec^2	WD	CSL
WL_a	waterline coordinate of accelerometer, + up, in. **(Default = 91.3)**	NYLOC(3)	CSL
WL_{cg}	waterline coordinate of c.g., + up, in. **(Default = 105.4)**	WLCG	CSL
WL_{rcg}	waterline coordinate of reference c.g., + up, in. **(Default = 100.0)**		
WL_{INS}	waterline coordinate of INS, + up, in. **(Default = 104.71)**	WLINS	INSMDL
WL_{rf}	waterline coordinate of aerodynamic reference center, + up, in. **(Default = 100.0)**	WLRF	CSL
x_a	distance from c.g. to the accelerometer along x-body axis, + forward, = $-(FS_a - FS_{cg})/12$, ft	XBAR	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_{INS}	distance from c.g. to the INS along x-body axis, + forward, = $-(FS_{INS} - FS_{cg})/12$, ft	RLX	INSMDL
y_a	distance from c.g. to the accelerometer along y-body axis, + right, = $(BL_a - BL_{cg})/12$, ft	YBAR	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_{INS}	distance from c.g. to the INS along y-body axis, + right, = $(BL_{INS} - BL_{cg})/12$, ft	RLY	INSMDL
z_a	distance from c.g. to the accelerometer along z`body axis, + down, = $-(WL_a - WL_{cg})/12$, ft	ZBAR	ACCEL
z_{arf}	distance from c.g. to the a.r.c. along z-body axis, + down, = $-(WL_{rf} - WL_{cg})/12$, ft	ZARF	CSL

Symbol	Definition	Variable Name	Program Unit
z_{INS}	distance from c.g. to the INS along z-body axis, + down, = $-(WL_{INS} - WL_{cg})/12$, ft	RLZ	INSMDL
α	Angle-of-attack, degrees	ADEG, ALFDG	CSL
α_{MC}	Mission Computer angle of attack, degrees	ALFINS	AOAINS
$\alpha_{local\ probe}$	local probe angle of attack, degrees	ALOCAL	AIRDATA
α_{probe}	probe angle of attack, degrees	ALPHAISM	AIRDATA
α_S	selected angle of attack between α_{probe} and α_{MC}, degrees	ALFS	USR18V150
$\dot{\alpha}$	angle-of-attack rate, degrees/sec^2	ALFDDG	CSL
$\dot{\alpha}_{MC}$	Mission Computer angle-of-attack rate, rad/sec^2	XCAOAR	AOAINS
$\dot{\alpha}_S$	sensor-model angle-of-attack rate, degrees/sec^2	ADS	AOAINS
β	Angle of sideslip, degrees	BDEG	CSL
β_{MC}	Mission Computer angle of sideslip, radians	XCSR	AOAINS
β_S	sensor-model angle of sideslip, degrees	BETS	AOAINS
$\dot{\beta}$	angle-of-sideslip rate, degrees/sec^2	BETDDG	CSL
$\dot{\beta}_{MC}$	Mission Computer angle-of-sideslip rate, rad/sec^2	XCSSR	AOAINS
$\dot{\beta}_S$	sensor-model angle-of-sideslip rate, degrees/sec^2	BDS	AOAINS
δ_{A_l}	Left aileron, + t.e.d., degrees	DAL	CSL
δ_{A_r}	Right aileron, + t.e.d., degrees	DAR	CSL
δ_{f_l}	Left trailing-edge flap, + t.e.d., degrees	DFL	CSL
δ_{f_r}	Right trailing-edge flap, + t.e.d., degrees	DFR	CSL
δ_H	average stabilator position, degrees		
δ_{H_l}	Left stabilator (horizontal tail), + t.e.d., degrees	DSL	CSL
δ_{H_r}	Right stabilator (horizontal tail), + t.e.d., degrees	DSR	CSL
δ_{LG}	Normalized landing gear, 0 - 1, full down	DLG	CSL
δ_{n_l}	Left leading-edge flap, + l.e.d., degrees	DNL	CSL
δ_{n_r}	Right leading-edge flap, + l.e.d., degrees	DNR	CSL
δ_p	average of TVJANP(1) & TVJANP(2), degrees	TVJPAVG	AEROINC
$\delta_{Pitch\_jet\_cmd}$	RFCS pitch jet command	PJETAC	HARVCSM
δ_{p_l}	thrust-vectored pitch jet-turning-angle for the left engine in the x-y engine axis plane, degrees	TVJANP(1)	TVANGL
δ_{p_r}	thrust-vectored pitch jet-turning-angle for the right engine in the x-y engine axis plane, degrees	TVJANP(2)	TVANGL
δ_r	difference of TVJANP(1) & TVJANP(2), degrees	TVJRAVG	AEROINC
δ_{R_l}	Left rudder, + t.e. left, degrees	DRL	CSL

Symbol	Definition	Variable Name	Program Unit
δ_{R_r}	Right rudder, + t.e. left, degrees	DRR	CSL
δ_{SB}	Speed brakes, 0 - 60, degrees	DSB	CSL
$\delta_{S_{diff}}$	Differential strake deflection	DFSDIF	AEROINC
δ_{S_l}	Left strake deflection	DFSL	NSTAT
δ_{S_r}	Right strake deflection	DFSR	NSTAT
$\delta_{S_{symm}}$	Symmetric strake deflection	DFSSYM	AEROINC
$\delta_{S_{symm,ref}}$	Reference symmetric strake deflection	–	
δ_y	average of TVJANY(1) & TVJANY(2), degrees	TVJYAVG	AEROINC
δ_{y_l}	thrust-vectored pitch jet-turning-angle for the left engine in the x-z engine axis plane, degrees	TVJANY(1)	TVANGL
δ_{y_r}	thrust-vectored pitch jet-turning-angle for the right engine in the x-z engine axis plane, degrees	TVJANY(2)	TVANGL
$\Delta C_{D_{\delta_h}}$	change in drag due to stabilator	DCDDHA	IN3060
$\Delta C_{D_{tvcs}}$	increment in drag due to tvcs	DCDVECT	TVCOEF
$\Delta C_{D_{vector}}$	increment in drag due to vector	CDTVCSA	IN3060
ΔC_{l_β}	increment in roll moment due to sideslip	DCLB	TVCOEF
$\Delta C_{l_{\beta_{tvcs}}}$	increment in roll moment due to sideslip for tvcs	CLBTVCA	IN3060
$\Delta C_{l_{\delta_r}}$	change in roll moment due to rudder	CCLDRA	IN3060
$\Delta C_{l_{\beta_{vector}}}$	increment in roll moment due to sideslip for vector	CLBVCA	IN3060
$\Delta C_{l_{HARV}}$	total HARV increment in roll moment	DELCL	TVCOEF
$\Delta C_{l_{vector}}$	increment in roll moment due to vector	CLVECA	IN3060
$\Delta C_{l_{vector_{tot}}}$	total increment in roll moment due to vector	DCLVTOT	TVCOEF
$\Delta C_{L_{\delta_h}}$	change in lift due to stabilator	DCLDHA	IN3060
$\Delta C_{L_{HARV}}$	total HARV increment in lift	DELCLF	TVCOEF
$\Delta C_{L_{tvcs}}$	increment in lift due to tvcs	DCLVECT	TVCOEF
$\Delta C_{L_{vector}}$	increment in lift due to vector	CLTVCSA	IN3060
$\Delta C_{m_{\delta_h}}$	change in pitch moment due to stabilator	DCMDHA	IN3060
$\Delta C_{m_{HARV}}$	total HARV increment in pitch moment	DELCM	TVCOEF
$\Delta C_{m_{tvcs}}$	increment in pitch moment due to tvcs	DCMVECT	TVCOEF
$\Delta C_{m_{vector}}$	increment in pitch moment due to vector	CMTVCSA	IN3060
ΔC_{n_β}	increment in yaw moment due to sideslip	DCNB	TVCOEF
$\Delta C_{n_{\beta_{tvcs}}}$	increment in yaw moment due to sideslip for tvcs	CNBTVCA	IN3060
$\Delta C_{n_{\beta_{vector}}}$	increment in yaw moment due to vector	CNBVCA	IN3060
$\Delta C_{n_{\delta_r}}$	change in yaw moment due to rudder	CCNDRA	IN3060
$\Delta C_{n_{HARV}}$	total HARV increment in yaw moment	DELCN	TVCOEF

Symbol	Definition	Variable Name	Program Unit
$\Delta C_{n_{vector}}$	increment in yaw moment due to vector	CNVECA	IN3060
$\Delta C_{n_{vector_{tot}}}$	total increment in yaw moment due to vector	DCNVTOT	TVCOEF
$C_{T_{fac}}$	thrust coefficient factor	CTFACA	IN3060
$\Delta C_{X_{dynamic}}$	General ANSER dynamic aero increment	(various)	NSTAT
$\Delta C_{X_{forced\ oscillation}}$	General ANSER forced-oscillation aero increment	(various)	NSTAT
$\Delta C_{X_{strake}}$	General ANSER increment, static and dynamic	(various)	NSTAT
$\Delta C_{X_{static}}$	General ANSER static aero increment	(various)	NSTAT
$\Delta C_{X_{static_{symm}}}$	Static symmetric ANSER increment	(various)	NSTAT
$\Delta C_{X_{static_{diff}}}$	Static differential ANSER increment	(various)	NSTAT
$\Delta C_{X_{static_{high-speed}}}$	High-speed static ANSER increment	(various)	NSTAT
$\Delta C_{X_{rotary\ balance}}$	Rotary-balance ANSER increment	(various)	FSROB
$\Delta C_{X_{q_{symm}}}$	Dynamic derivative increment due to symmetric strake deflection	(various)	FSDFO
$\Delta C_{X_{q_{diff}}}$	Dynamic derivative increment due to differential strake deflection	(various)	FSDFO
ΔC_{Y_β}	increment in side force due to sideslip	DCNB	TVCOEF
$\Delta C_{Y_{\beta_{tvcs}}}$	increment in side force due to sideslip for tvcs	CYBTVCA	IN3060
$\Delta C_{y_{\beta_{vector}}}$	increment in side force due to vector	CYBVCA	IN3060
$\Delta C_{Y_{\delta_r}}$	change in side force due to rudder	CCYDRA	IN3060
$\Delta C_{Y_{HARV}}$	total HARV increment in side force	DELCY	TVCOEF
$\Delta C_{Y_{vector}}$	increment in side force due to vector	CYVECA	IN3060
$\Delta C_{Y_{vector_{tot}}}$	total increment in side force due to vector	DCYVTOT	TVCOEF
γ	longitudinal flight path angle, degrees	GAMDG	CSL
γ_s	sensor-model wind-axis Euler angle, radians	GAMS	SENSOR
λ	lateral flight path angle, degrees		
μ	bank angle, degrees	MUDGTR	CSL
μ_s	sensor-model wind-axis Euler angle, radians	MUS	SENSOR
ω	Frequency of the second-order system		
$\omega 1$	No-load open-loop gain for ANSER actuator		
$\omega 2$	Open-loop first-order lag		
Ω_{SS}	Steady-state non-dimensional rotation rate	WNONDIM	AEROINC
θ	pitch angle, radians	THE	CSL
θ_S	sensor-model pitch angle, radians	THAINS	CSL

Symbol	Definition	Variable Name	Program Unit
θ_{INS}	INS pitch angle, radians	THAINS	CSL
π	3.14159265		
ρ	density of the air, slugs/ft^3	RHO	CSL
ϕ	roll angle, radians	PHI	CSL
ϕ_s	sensor-model roll angle, radians	PHIINS	CSL
ϕ_{INS}	INS roll angle, radians	PHIINS	CSL
φ	yaw angle, radians	PSI	CSL
φ_s	sensor-model yaw angle, radians	PSIINS	INSMDL
φ_{INS}	INS yaw angle, radians	PSIINS	INSMDL
τ	Dynamic fading time constant	TAURB	AEROINC
χ_s	sensor-model wind-axis Euler angle, radians	CHIS	SENSOR
ζ	Damping of the second-order system		

Subscripts

ACC	accelerometer
ADC	airdata computer
asm	asymmetries
E,eng	engine induced
G, g	gravity
INS	inertial nagivation system
MC	Mission Computer
probe	Airdata probe
rf	quantity computed at the aerodynamic reference center (a.r.c.)
s	sensor parameters
TR	trim
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
| |	absolute value
ABS()	Absolute value of ()
$x\lceil_b^a$	the variable x restricted to the range, $b \leq x \leq a$, typically used to describe a function argument that is limited.

Acronyms

A/B	Afterburner
ACSL	Advanced Continuous Simulation Language
A/D	Analog-to-Digitial
ADC	Airdata Computer
ANSER	Actuated Nose Strakes for Enhanced Rolling
ARC	Ames Research Center
BL	Buttock line coordinate, + right
CAS	Control Augmentation System
D/A	Digital-to-Analog
DCB	Dynamics and Control Branch
DPRAM	Dual-Port Random-Access Memory
DFRC	Dryden Flight Research Facility
DMS	Differential Maneuvering Simulator twin-dome facility at NASA LaRC
DTRC	David Taylor Research Center
EOM	Equations of motion
FS	Fuselage station coordinate, + aft
f18bas	Simulation model and computer program described in Reference 1.0
f18harv	Simulation model and computer program described in this document resulting from extensive modifications to the *f18bas* program to support the HARV research program
GE	General Electric Corporation
HANG	High Angle-of-Attack Nose-Down Guidelines
HARV	High Alpha Research Vehicle
HATP	High Alpha Technology Program
HIL	Hardware-in-the-Loop
fps	feet per second
IB	Iron Bird

INS	Inertial Navigation System
LaRC	Langley Research Center
LEF	Leading Edge Flap
LVDT	Linear Variable Differential Transformer
MC	Mission Computer
MDC	McDonnell Douglas Corporation
M/P	Mixer/Predictor
NASA-0	DFRC/McDonnell Douglas Research Flight Control System
NASA-1	NASA LaRC/DFRC Research Flight Control System
NASA-2	DFRC/Honeywell Research Flight Control System
NPR	Nozzle Pressure Ratio
OBES	On Board Excitation System
PSC	Pseudo Controls
psf	pounds per square foot
QSE	Quasi-Static Elastic
RAV	Remote Augmented Vehicle
RFCS	Research Flight Control System
rpm	revolutions per minute
SOS	Speed of Sound
TE	Thrust Estimator
TEF	Trailing Edge Flap
TV	Thrust Vectoring
TVS	Thrust Vectoring System
VDB	Vehicle Dynamics Branch
WL	Waterline coordinate, + up
1750A	RFCS Computer Processor
701E	F/A-18 Flight Control Computer Processor

APPENDIX A - SIMULATION VARIABLES

Simulation Dictionary Option

The variables defined in the dictionary (unit 29) for the *f18harv* simulation has not been changed from those defined in *f18bas*. At some future date these definitions will be updated to reflect the list of simulation variables defined in the following section.

List of Simulation Variables

A8L	estimated left engine nozzle throat area, inches^2
A8LEST	estimated left engine nozzle throat area, inches^2
A8LS	estimated left engine nozzle throat area, inches^2
A8PCTL	left engine nozzle throat area, percent
A8PCTR	right engine nozzle throat area, percent
A8R	estimated right engine nozzle throat area, inches^2
A8REST	estimated right engine nozzle throat area, inches^2
A8RS	estimated right engine nozzle throat area, inches^2
A8SQ	estimated engine nozzle throat area, inches^2
A8SQL	estimated left engine nozzle throat area, inches^2
A8SQR	estimated right engine nozzle throat area, inches^2
ADS	Mission computer angle-of-attack rate, degrees/sec
AF2IN	3 element past-state array of inputs for pitch-rate aliasing filter, degrees/sec
AF2OT	3 element past-state array of outputs for pitch-rate aliasing filter, degrees/sec
AF5IN	3 element past-state array of input values for roll-rate aliasing filter, degrees/sec
AF5OT	3 element past-state array of output values for roll-rate aliasing filter, degrees/sec
AF6IN	3 element past-state array of input values for yaw-rate aliasing filter, degrees/sec
AF6OT	3 element past-state array of output values for yaw-rate aliasing filter, degrees/sec
ALDGRF	angle of attack at the aerodynamic reference center (a.r.c.), degrees
ALF	angle of attack at the c.g., radians
ALFD	d/dt of ALF, radians/sec
ALFDG	angle of attack at the c.g., degrees
ALFDDG	d/dt of angle of attack at the c.g., degrees
ALFDP	d/dt of ALFP, degrees/sec
ALFINS	Mission computer calculated angle of attack, updated at 40 Hz, degrees
ALFP	local probe angle of attack (input to AIRDATA subroutine), radians
ALFPDG	local probe angle of attack (input to AIRDATA subroutine), degrees
ALFPIC	airdata probe angle-of-attack initial condition, degrees
ALFPRO	airdata probe angle of attack, updated at 40 Hz without bias and gain correction, degrees
APRPLT	airdata probe angle of attack, updated at 40 Hz with bias and gain correction, degrees
ALFS	selected angle of attack, degrees
ALFS1	airdata angle-of-attack output without bias and gain correction from AIRDATA subroutine , degrees
ALFS2	output of AIRDATA subroutine with bias and gain correction, degrees
ALFTR	angle of attack required for trim, radians
ALFTRDG	angle of attack required for trim, degrees
ALPBIAS	bias added to Mission computer angle of attack, degrees
ALPGAIN	gain multiplied to Mission computer angle of attack, n.d.
ALPHAISM	local variable name in AIRDATA subroutine for airdata angle of attack without bias and gain applied, degrees

ALPHAS	selected angle of attack computed in SENSOR subroutine, degrees
ALT	vertical distance from sea level to aircraft c.g., ft
ALTS	sensor-model altitude, ft
AMCH	Mission computer Mach number, n.d.
AN	normal acceleration, 1 g straight and level, g's
AX	acceleration in x-body direction (referred to as axial acceleration), g's
AY	acceleration in y-body direction (referred to as lateral acceleration), g's
AYACC	sensor-model lateral acceleration in y-body direction, g's
AYS	sensor-model lateral acceleration (limited and 40 Hz), g's
AYW	wind-axis lateral acceleration, ft/sec^2
AZ	acceleration in z-body direction (referred to as normal acceleration), g's
AZACC	sensor-model normal acceleration in z-body direction, g's
AZS	sensor-model normal acceleration (limited and 40 Hz), g's
AZW	wind-axis normal acceleration, ft/sec^2
AZWG	wind-axis normal acceleration, g's
BDS	Mission computer angle-of-sideslip rate, updated at 40 Hz, degrees/sec
BEDGRF	angle-of-sideslip at the aerodynamic reference center (a.r.c.), degrees
BET	angle-of-sideslip at the c.g., radians
BETD	d/dt of BET, radians/sec
BETDDG	d/dt of angle-of-sideslip rate at the c.g., degrees/sec
BETDG	angle-of-sideslip at the c.g., degrees
BETS	Mission computer angle-of-sideslip, updated at 80 Hz, degrees
BETTR	value of BET required for trim, calculated in trim search, radians
BETTRDG	value of BET required for trim, calculated in trim search, degrees
BIAS40	logical=TRUE to activate inlet stall margin increase, medium
BIAS85	logical=TRUE to activate inlet stall margin increase, maximum
BLCG	buttock-line coordinate of the center of gravity, inches
BLRF	buttock-line coordinate of the aerodynamic reference center, **(= 0.00 inches)**
BTABIAS	bias added to Mission computer angle-of-sideslip rate, degrees/sec
BTAGAIN	gain multiplied to Mission computer angle-of-sideslip rate, n.d.
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	left aileron command, degrees
CANSL	left-nose-strake command, degrees
CANSR	right-nose-strake command, degrees
CAQFSD	differential strake CAQ increment
CAQFSS	symmetric strake CAQ increment
CAR	right aileron command, 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
CBNSL	left-nose-strake command, degrees
CBNSR	right-nose-strake command, degrees
CCNSL	dynamic-check perturbation signal to left-nose-strake command, degrees
CCNSR	dynamic-check perturbation signal to right-nose-strake command, degrees
CDRFSF	drag coefficient at aerodynamic reference center, steady flow (= 0), n.d.
CDSFH	high-speed static ANSER drag coefficient increment, n.d.
CDSFLL	low-speed left-nose-strake drag contribution, n.d.
CDSFRL	low-speed right-nose-strake drag contribution, n.d.
CDSSD	static differential ANSER drag-coefficient increment

CDSSS	static symmetric ANSER drag-coefficient increment
CFASLN	commanded antisymmetric trailing-edge flap, linearization input only, degrees
CFASP	dynamic-check perturbation signal to antisymmetric trailing-edge flap command, degrees
CFL	left trailing-edge flap command, degrees
CFR	right trailing-edge flap command, degrees
CFSYLN	commanded symmetric trailing-edge flap, linearization input only, degrees
CFSYP	dynamic-check perturbation signal to symmetric trailing-edge flap command, degrees
CHIS	sensor-model wind-axis Euler angle, radians
CINT	simulation communication interval, time interval at which data is written to the master time history file or available at the screen, seconds
CLERR	actual target lift coefficient, CLERR = CLRFSF - CLTR, n.d.
CLLSFH	high-speed static ANSER roll-coefficient increment, n.d.
CLLSFLL	low-speed left-nose-strake rolling-moment contribution, n.d.
CLLSFRL	low-speed right-nose-strake rolling-moment contribution, n.d.
CLPFSD	differential strake C_{l_p} increment
CLPFSS	symmetric strake C_{l_p} increment
CLRFSD	differential strake C_{l_r} increment
CLRFSF	lift coefficient at aerodynamic reference center, steady flow (= 0), n.d.
CLRFSS	symmetric strake C_{l_r} increment
CLSFH	high-speed static ANSER lift-coefficient increment, n.d.
CLSFLL	low-speed left-nose-strake lift contribution, n.d.
CLSFRL	low-speed right-nose-strake lift contribution, n.d.
CLSSD	static differential ANSER lift-coefficient increment
CLSSS	static symmetric ANSER lift-coefficient increment
CLTR	target trim value for CLRFSF if "12" is in active part of IYSEL array.
CMQFSD	differential strake C_{m_q} increment
CMQFSS	symmetric strake C_{m_q} increment
CMRFSF	pitch (about y-body axis) coefficient at aerodynamic reference point, steady flow (= 0), n.d.
CMSFH	high-speed static ANSER pitch-coefficient increment, n.d.
CMSFLL	low-speed left-nose-strake pitching-moment contribution, n.d.
CMSFRL	low-speed right-nose-strake pitching-moment contribution, n.d.
CMSSD	static differential ANSER pitch-coefficient increment
CNASLN	commanded antisymmetric leading-edge flap, linearization input only, degrees
CNASP	dynamic-check perturbation signal to antisymmetric leading-edge flap command, degrees
CNL	left leading-edge flap command, degrees
CNPFSD	differential strake C_{n_p} increment
CNPFSS	symmetric strake C_{n_p} increment
CNQFSD	differential strake C_{n_q} increment
CNQFSS	symmetric strake C_{n_q} increment
CNR	right leading-edge flap command, degrees
CNRFSD	differential strake C_{n_r} increment
CNRFSF	yaw (about z-body axis) coefficient at aerodynamic reference point, steady flow (= 0), n.d.
CNRFSS	symmetric strake C_{n_r} increment
CNSFH	high-speed static ANSER yaw-coefficient increment, n.d.
CNSFLL	low-speed left-nose-strake yawing-moment contribution, n.d.
CNSFRL	low-speed right-nose-strake yawing-moment contribution, n.d.
CNSL	left-nose-strake command, degrees

CNSR	right-nose-strake command, degrees
CNSSD	static differential ANSER yaw-coefficient increment
CNSSS	static symmetric ANSER yaw-coefficient increment
CNSYLN	commanded symmetric leading-edge flap, linearization input only, degrees
CNSYP	dynamic-check perturbation signal to symmetric leading-edge flap command, degrees
COSCHI	cosine of sensor-model wind-axis Euler angle, n.d.
COSGAM	cosine of sensor-model wind-axis Euler angle, n.d.
COSMU	cosine of sensor-model wind-axis Euler angle, n.d.
COSPHI	cosine of sensor-model Euler roll angle, n.d.
COSPSI	cosine of Euler yaw angle, n.d.
COSTHE	cosine of sensor-model Euler pitch angle, n.d.
CO2VT	intermediate quantity, , 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 (31° - 131°), 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 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	left rudder command, degrees
CYSSD	static differential ANSER sideforce-coefficient increment
CRMSSS	static symmetric ANSER roll-coefficient increment
CRR	right ruder command, 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., $\cos(\alpha)$, n.d.
CSALRF	cosine of angle of attack at the a.r.c., $\cos(\alpha_{ref})$, n.d.
CSASLN	commanded antisymmetric stabilator, linearization input only, degrees
CSASP	dynamic-check perturbation signal to antisymmetric stabilator command, degrees
CSB	command to speed-brake actuator, degrees
CSBET	cosine of angle-of-sideslip at the c.g., n.d.
CSL	left stabilator command, degrees
CSPHI	cosine of roll angle, n.d.
CSPHO2	cosine of (PHITR / 2), n.d.
CSPSO2	cosine of (PSITR / 2), n.d.
CSR	right stabilator command, 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, n.d.
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	cosine of (THETR / 2), n.d.
CTASP	dynamic-check perturbation signal to antisymmetric throttle, degrees
CTSYP	dynamic-check perturbation signal to symmetric throttle, degrees
CTASLN	commanded antisymmetric throttle, linearization input only, degrees

CTSYLN	commanded symmetric throttle, linearization input only, degrees
CWRF	aerodynamic wing reference chord, **(= 11.52 ft)**
CXRFSF	aerodynamic axial force (x-body) coefficient at a.r.p., steady flow (= 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
CYPFSD	differential strake C_{y_p} increment
CYPFSS	symmetric strake C_{y_p} increment
CYRFSD	differential strake C_{y_r} increment
CYRFSF	aerodynamic side-force (y-body) coefficient at a.r.p., steady flow (= 0), n.d.
CYRFSS	symmetric strake C_{y_r} increment
CYSFH	high-speed static ANSER sideforce-coefficient increment, n.d.
CYSFLL	low-speed left-nose-strake side-force contribution, n.d.
CYSFRL	low-speed right-nose-strake side-force contribution, n.d.
CYSSD	static differential ANSER sideforce-coefficient increment
CYSSS	static symmetric ANSER sideforce-coefficient increment
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 (= 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
DAL	total left aileron position, includes linearization input, degrees
DALLN	component of total left aileron deflection, linearization input only, degrees
DALLN1	= DALLN (if .not. LLNCSY) or (DASYLN - DAASLN) (if LLNCSY), degrees.
DAPRFCS	pilot lateral stick position after 701E demod filter, inches
DAR	total right aileron position, includes linearization input, degrees
DARLN	component of total right aileron deflection, linearization input only, degrees
DARLN1	= DARLN (if .not. LLNCSY) or (DASYLN + DAASLN) (if LLNCSY), degrees
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
DCDFS	total ANSER drag-coefficient increment
DCDFSF	forced oscillation ANSER drag-coefficient increment
DCDFSR	rotary balance ANSER drag-coefficient increment
DCDFSS	static ANSER drag-coefficient increment
DCNFSF	forced oscillation ANSER yaw-coefficient increment
DCDS	total nose strake drag coefficient
DCLFS	total ANSER lift-coefficient increment
DCLFSF	forced oscillation ANSER lift-coefficient increment
DCLFSR	rotary balance ANSER lift-coefficient increment
DCLFSS	static ANSER lift-coefficient increment
DCLS	total nose strake lift coefficient
DCLLS	total nose strake rolling-moment coefficient
DCMFS	total ANSER pitching-moment-coefficient increment
DCMFSF	forced oscillation ANSER pitch-coefficient increment
DCMFSR	rotary balance ANSER pitch-coefficient increment

DCMFSS	static ANSER pitch-coefficient increment
DCMS	total nose strake pitching-moment coefficient
DCNFS	total ANSER yawing-moment-coefficient increment
DCNFSF	forced oscillation ANSER yaw-coefficient increment
DCNFSR	rotary balance ANSER yaw-coefficient increment
DCNFSS	static ANSER yaw-coefficient increment
DCNS	total nose strake yawing-moment coefficient
DCRMFS	total ANSER rolling-moment-coefficient increment
DCRMFSF	forced oscillation ANSER roll-coefficient increment
DCRMFSR	rotary balance ANSER roll-coefficient increment
DCRMFSS	static ANSER roll-coefficient increment
DCYFS	total ANSER sideforce-coefficient increment
DCYFSF	forced oscillation ANSER sideforce-coefficient increment
DCYFSR	rotary balance ANSER sideforce-coefficient increment
DCYFSS	static ANSER sideforce-coefficient increment
DCYS	total nose strake side-force coefficient
DELCD	HARV total drag-coefficient increment
DELCL	HARV total rolling-moment coefficient
DELCLF	HARV total lift-coefficient increment
DELCM	HARV total pitching-moment coefficient
DELCN	HARV total yawing-moment coefficient
DELCY	HARV total side-force coefficient
DELTACL	left aileron command in subroutine HARVCSM, degrees
DELTACR	right aileron command in subroutine HARVCSM, degrees
DELTAP	equal to PCA in subroutine HARVCSM
DELTFCL	left trailing-edge flap command in subroutine HARVCSM, degrees
DELTFCR	right trailing-edge flap command in subroutine HARVCSM, degrees
DELTNCL	left leading-edge flap command in subroutine HARVCSM, degrees
DELTNCR	right leading-edge flap command in subroutine HARVCSM, degrees
DELTRP	equal to PCR in subroutine HARVCSM
DELTRCL	left rudder command in subroutine HARVCSM, degrees
DELTRCR	right rudder command in subroutine HARVCSM, degrees
DELTSCL	left stabilator command in subroutine HARVCSM, degrees
DELTSCR	right stabilator command in subroutine HARVCSM, degrees
DELTSP	equal to PCS in subroutine HARVCSM
DELSTM	stabilator and pitch-jet command position for trim, degrees
DETCO	engine thrust-deterioration coefficient (= **0.96**), n.d.
DETCOL	left-engine thrust-deterioration coefficient (= **0.96**), n.d.
DETCOR	right-engine thrust-deterioration coefficient (= **0.96**), n.d.
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
DFL	total left trailing-edge flap position, includes linearization input, 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
DFRLN	component of total right trailing-edge-flap deflection, linearization input only, degrees
DFRLN1	= DFRLN (if .not. LLNCSY) or (DFSYLN + DFASLN) (if LLNCSY).
DFSDIF	differential strake deflection, degrees
DFSL	left strake deflection, degrees
DFSR	right strake deflection, degrees
DFSSYM	symmetric strake deflection, degrees
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
DGTHLC(1)[1]	left engine lagged-throttle position (models engine dynamics), degrees (31°-131°)
DGTHLC(2)[1]	right engine lagged-throttle position (models engin dynamics), degrees (31°-131°)
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	gear position (0. - up, 1. = down) **(Default = 0.)**
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
DNGEAR	local variable for gear position (0. - up, 1. = down) in subroutine AIRDATA **(Default = 0.)**
DNL	total left leading-edge-flap position, includes linearization input, degrees
DNLLN	component of total left leading-edge-flap deflection, linearization input only, degrees
DNLLN1	= DNLLN (if .not. LLNCSY) or (DNSYLN - DNASLN) (if LLNCSY), degrees.
DNR	total right leading-edge-flap position, includes linearization input, degrees
DNRLN	component of total right leading-edge-flap deflection, linearization input only, degrees
DNRLN1	= DNRLN (if .not. LLNCSY) or (DNSYLN + DNASLN) (if LLNCSY), degrees.
DNSL	left strake deflection, degrees
DNSR	right strake deflection, degrees
DNSY	symmetric leading-edge-flap deflection, degrees
DNSYLN	component of total symmetric leading-edg- 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
DPLLIC	left throttle boost output signal initial condition, degrees
DPLLN	added to PLAL, linearization input only, degrees
DPLLN1	= DPLLN (if .not. LLNCSY) or (DPSYLN - DPASLN) (if LLNCSY), degrees.
DPLRIC	right throttle boost output signal initial condition, degrees
DPRLN	added to PLAR, linearization input only, degrees
DPRLN1	= DPRLN (if .not. LLNCSY) or (DPSYLN + DPASLN) (if LLNCSY), degrees.
DPSYLN	component of total symmetric throttle boost output signal, linearization input only, degrees
DPSYTR	symmetric power-lever-angle deflection resulting from trim calculations, degrees
DRPRFCS	pilot rudder pedal force after 701E aliasing filter, lbs
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
DRL	total left rudder position, includes linearization input, degrees
DRLLN	component of total left rudder deflection, linearization input only, degrees
DRLLN1	= DRLLN (if .not. LLNCSY) or (DRSYLN - DRASLN) (if LLNCSY), degrees.
DRP	pilot rudder-pedal force after 701E mechanical breakout, lbs
DRPRFCS	pilot rudder-pedal force after 701E aliasing filter, lbs
DRR	total right rudder position, includes linearization input, degrees
DRRLN	component of total right rudder deflection, linearization input only, degrees
DRRLN1	= DRRLN (if .not. LLNCSY) or (DRSYLN + DRASLN) (if LLNCSY), degrees.
DRSY	symmetric rudder deflection, both t.e. inward, degrees
DRSYLN	component of total symmetric rudder deflection, linearization input only, degrees
DRSYTR	symmetric rudder deflection resulting from trim calculations, 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
DSL	total left stabilator position, includes linearization input, degrees
DSLLN	component of total left stabilator deflection, linearization input only, degrees
DSLLN1	= DSLLN (if .not. LLNCSY) or (DSSYLN - DSASLN) (if LLNCSY), degrees.
DSR	total right stabilator position, includes linearization input, degrees
DSRLN	component of total right stabilator deflection, linearization input only, degrees
DSRLN1	= DSRLN (if .not. LLNCSY) or (DSSYLN + DSASLN) (if LLNCSY), degrees.
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
DTENG	sample time interval at which discrete engine model is called, **(0.0125 seconds)**
DTASLN	component of total antisymmetric throttle deflection, linearization input only, degrees
DTASTR	antisymmetric nose-strake deflection resulting from trim calculations, degrees
DTLLN	component of total left-nose-strake deflection, degrees
DTLLN1	=DTLLN (if .not. LLNCSY) or (DTSYLN + DTASLN) (if LLNCSY), degrees
DTRLN	component of total right-nose-strake deflection, degrees
DTRLN1	=DTRLN (if .not. LLNCSY) or (DTSYLN + DTASLN) (if LLNCSY), degrees
DTSYLN	component of total symmetric throttle deflection, linearization only, degrees
DTSYTR	symmetric nose-strake deflection resulting from trim calculations, degrees
DURAT	frequency sweep duration time, sec
DXTHRS	distance from the aerodynamic reference point to the c.g. of the engine in the x-axis direction, ft
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), radians/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), radians/sec
E1IC	Initial condition for E1, derived from trim Euler angles, n.d.
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), radians/sec
E2IC	Initial condition for E2, derived from trim Euler angles, n.d.
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), radians/sec
E3IC	initial condition for E3, derived from trim Euler angles, n.d.

EFFJAN(1)	effective jet-angle turning for left-top-vane command, degrees
EFFJAN(2)	effective jet-angle turning for left-outboard-vane command, degrees
EFFJAN(3)	effective jet-angle turning for left-inboard-vane command, degrees
EFFJAN(4)	effective jet-angle turning for right-top-vane command, degrees
EFFJAN(5)	effective jet-angle turning for right-outboard-vane command, degrees
EFFJAN(6)	effective jet-angle turning for right-inboard-vane command, degrees
ENGTD	sample time interval at which discrete engine model is called, **(0.0125 seconds)**
ENPR(1)	estimated left engine nozzle pressure ratio
ENPR(2)	estimated right engine nozzle pressure ratio
EPS	used to prevent divide by zero, **(= 10^{-10})**
ERPM(1)	left engine rpm, rpm
ERPM(2)	right engine rpm, rpm
ERRORS	error signal out of proportional gain in RFCS, used for trim
EVENT	time at which dynamic-check perturbation is applied, sec
FCS	marker for flight control system discrete block
FG(1)	left engine gross thrust without angle-of-attack effects, TV losses, or drags, lbs
FG(2)	right engine gross thrust without angle-of-attack effects, TV losses, or drags, lbs
FGAERO(1)	= FG(1). Gross thrust for left engine after T56 bias multiplier losses, lbs
FGAERO(2)	= FG(2). Gross thrust for right engine after T56 bias multiplier losses, lbs
FGTOT(1)	left engine gross thrust including angle-of-attack effects, lbs
FGTOT(2)	right engine gross thrust including angle-of-attack effects, lbs
FIRST	logical used for subroutine AOAINS initialization
FIRST2	logical used for subroutine TVANGL initialization
FIRST3	logical used for subroutine SENSOR initialization
FSCG	fuselage station coordinate of the center of gravity, inches
FSRF	fuselage station coordinate of the aerodynamic reference center, **(= 458.56 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)**
GAINSWITCH	switch to choose ANSER longitudinal gain set **(Default = 0.0)**
GAM	longitudinal flight path angle, radians
GAMDG	longitudinal flight path angle, degrees
GAMS	sensor-model wind-axis Euler angle, radians
GAMTR	target longitudinal flight path angle used in trim search, radians
GAMTRDG	target longitudinal flight path angle used in trim search, degrees
GAMZR	GAM - GAMTR, driven to zero by trim search, radians
GAMZRDG	GAM - GAMTR, driven to zero by trim search, degrees
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
GLOAD	desired load factor used in trim search when TCASE=3, g's
GROSST(1)	left engine gross thrust including angle-of-attack effects, lbs
GROSST(2)	right engine gross thrust including angle-of-attack effects, lbs
GRTHEST(1)	estimated left engine gross thrust, lbs
GRTHEST(2)	estimated right engine gross thrust, lbs

H	vertical distance from sea level to aircraft c.g. (referred to as altitude), ft
HANGCUL	HANG command upper limit, **(Default = 25. degrees)**
HANGRL	HANG rate limit, **(Default = 320. degrees/sec)**
HD, HDOT	d/dt of H, ft/sec
HDWIC	initial wind heading, ft/sec
HDWRT	rate of change in altitude x 1000, deg/kft
HEADWIND	Headwind disturbance used in gust model, ft/sec
HGC, HGCL	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 air density, ρ, in slugs/ft^3
HRFIC	initial condition for HRF, ft
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
ICALLA[1]	set equal to 1 for first two CALLS to subroutine for each airplane. Set actuator output to the commanded position when ICALLA = 1. Set to an integer greater than 1 on subsequent CALLS.
ICALLE[1]	set equal to 1 for first two CALLS to subroutine for each airplane. Sets lagged throttle position (models engine dynamics) to the commanded position when ICALLE = 1. Set to an integer greater than 1 on subsequent CALLS.
ICALLM[1]	set equal to 1 for first CALL to subroutine for each airplane. Subroutine sets ICALLM = 2 during 1st CALL. Should be saved and reused on subsequent CALLS.
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
IN2FT	conversion factor, **(= 1/12 ft/in)**
ISENS	Integer switch for sensor model (0) sensor effects off, ideal measurements (1) DFRC 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.)**
IWND	20 Hz loop flag for gust filter in subroutine AOAINS
IXSEL	integer pointer array of size NXTRMX used to select elements of trim-solution vector, XTRIM. The first NXTR elements of IXSEL select from 33 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 20 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
LAMDA	difference between ground track angle and heading angle, radians
LAMBADG	difference between ground track angle and heading angle, degrees
LATSLIM	pilot lateral stick limit imposed in DERIVATIVE section of ACSL, **(= +/-3 inches)**
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)**

LCTASP logical switch
(.T.) A dynamic-check perturbation is added to the antisymmetric nose-strake command.
(.F.) no action **(Default)**

LCTSYP logical switch
(.T.) A dynamic-check perturbation is added to the symmetric nose-strake 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)**

LDRFCS integer switch to choose RFCS lateral/direction control law

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

LFCS logical switch - enables research flight control system - read in from file "f18tr.dat"
(.T.) RFCS "on" **(Default)**
(.F.) RFCS "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)**

LGA8L[1] last good estimate of left nozzle throat area, inches^2

LGA8R[1] last good estimate of right nozzle throat area, inches^2

LGLGT[1] last good estimate of left gross thrust, lbs

LGLNPR[1] last good estimate of left nozzle pressure ratio

LGRGT[1] last good estimate of right gross thrust, lbs

LGRNPR[1] last good estimate of right nozzle pressure ratio

LGTHR estimated left engine gross thrust, lbs

LGYRO logical switch used to zero engine gyroscopic effects
(.T.) no action **(Default)**
(.F.) zero gyroscopic effects

LHMOM logical to allow use of hinge-moment effects, **(Default = .T.)**

LINBVC left inboard vane command for NASA-0 M/P, inches

LLDTR logical switch
(.T.) When START command is executed, initial conditions for all possible XTRIM variables are read in from file "f18tr.dat". **(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)**

LLN logical switch
(.T.) Allow Jacobian calculations
(.F.) no action **(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 _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).

LMIX4 logical flag to choose Mixer/Predictor Version 4.4 **(Default = .T.)**

LNOACT logical flag to choose DRYDEN second order actuator model **(Default = .T.)**

LNPR estimated left engine nozzle pressure ratio

LNRFCS integer switch to choose RFCS longitudinal control law

LOBESP logical switch
(.T.) add frequency-sweep input to PTRIM
(.F.) no action **(Default)**

LOBESR logical switch
(.T.) add frequency-sweep input to OBESLAT
(.F.) no action **(Default)**

LOBESY logical switch
(.T.) add frequency-sweep input to OBESRUD
(.F.) no action **(Default)**

LOOP20[1] 20 Hz loop flag for subroutine AOAINS

LOOP40[1] 40 Hz loop flag for subroutine AOAINS

LOP20[1] 20 Hz loop flag for subroutine SENSOR

LOP40[1] 40 Hz loop flag for subroutine SENSOR

LOUTVC left-outboard-vane command for NASA-0 M/P, inches

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)**

LPJETP logical switch
(.T.) A dynamic-check perturbation is added to the pitch-jet command.
(.F.) no action **(Default)**

LQDELAY logical flag to add 3 frames of delay on pitch-rate sensor signal **(Default = .F.)**

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 untime commands
(1) causes f18tr.dat trim file to be read in **(Default)**
(2) causes f18tr.dat trim file to be written on

LRJETP logical switch
(T.) A dynamic-check perturbation is added to the roll-jet command.
(.F.) no action **(Default)**

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 reference 1.0, section 5.1 - aerodynamic model

LSTR logical nose-strake actuator selection switch **(Default = .T.)**

LSTKSTP logical to fail left-nose-strake actuator **(Default = .F.)**

LSTKTD logical to fail left-nose-strake actuator in trail damp mode **(Default = .F.)**

LTHDMS logical switch - used to duplicate the DMS throttle position
(.T.) use DMS scaled throttle position
(.F.) normal MDC F-18 throttle position **(Default)**

LTHVEC logical switch - used to invoke thrust-vectored engine model
(.T.) thrust-vectored engine model **(Default)**
(.F.) non-thrust-vectored engine model, sets vanes to stowed position of -10.0 degrees

LTOPVC left-top-vane command for NASA-0 M/P, inches

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

LUSEINS integer switch to determine INS flow angle computations **(Default = 1)**

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)**
LWRGD	logical switch - used for Getdata compatible output file
	(.T.) subroutine CMPFM called - data written to file="acsl_harv.cmp3"
	(.F.) no action **(Default)**
LWRMTX	logical switch - used for $MATRIX_X$ compatible output file
	(.T.) subroutine MTXFM called - data written to file="mtxacsl.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)**
LXSTKP	logical switch to put dynamic check on XSTRAKE
LXTVYP	logical switch to put dynamic check on XTVYAW
LYJETP	logical switch
	(T.) A dynamic-check perturbation is added to the yaw jet command
	(.F.) no action **(Default)**
MACH	Mach number, n.d.
MACHS	Mission computer 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
MIXDT	sample interval for mixer/predictor. Set equal to engine model sample interval **(0.0125 seconds)**.
MIX4INT[1]	logical for inialization of subroutine MIXER_4 **(Default = .T.)**
MMRR	total rigid-rigid quadrant of a mass matrix
MUDGTR	bank angle in degrees sought in trim search if "29" included in active IXSEL list. Will produce coordinated turn if TCASE = 2.
MUS	sensor-model wind-axis Euler angle, radians
MUTRMX	In the *f18bas* trim strategy reference 1.0, MUTRMX is equivalent to variable MUMAX" in ACSL TRIM algorithm **(Default = 100)**
MXITTR	maximum number of iterations for simulation trimming scheme **(Default = 1000)**
MXSTP	maximum integration step size **(Default = 0.00625 seconds)**
ND80HZ	number of frames delay desired, each frame of delay represents 0.0125 sec, n.d.
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 reference 1.1.
NTHRER[1]	counter which accumulates number of times the thrust estimator fails its data compatibility tests. Set equal to 0 for first CALL to subroutine for each airplane. Should be saved and reused on subsequent CALLS.
NUMOUT	determines frequency of ascii data output for Getdata-compatible output file
NXPS	axial acceleration at the pilot station, g's
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
NYPS	lateral acceleration at the pilot station, g's

NYSSG	lateral-acceleration sensor signal, g's
NYTR	number of dependent variables for simulation trim algorithm (= **NXTR**)
NZPS	normal acceleration at the pilot station, g's
NZSSG	normal-acceleration sensor signal, g's
NZZR	normal acceleration 1-g steady state removed. NZZR = - AZ - 1. g's
OBESLAT	OBES lateral-axis frequency sweep input, inches
OBESPIT	OBES longitudinal-axis frequency sweep input, inches
OBESRUD	OBES directional-axis frequency sweep input, lbs
P	body-axis roll rate, radians/sec
P56L	left engine turbine discharge pressure, lbs/inches^2
P56LS	estimated left engine turbine discharge pressure, lbs/inches^2
P56R	right engine turbine discharge pressure, lbs/inches^2
P56RS	estimated right engine turbine discharge pressure, lbs/inches^2
PAGSIZ	Integer that defines page size as number of lines. **(55 for VAX)**
PAMB	sensor-model static atmospheric pressure, lbs/inches^2
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
PCTHLC(1)[1]	left engine lagged-throttle position (models engine dynamics), percent (0-100)
PCTHLC(2)[1]	right engine lagged-throttle position (models engine dynamics), percent (0-100)
PD	body-axis roll acceleration, radians/sec^2
PDG	body-axis roll rate, degrees/sec
PDDG	body-axis roll acceleration, degrees/sec^2
PGAIN	gain on dynamic-check perturbation command
PHI	Euler roll angle, radians
PHI0	Euler roll-angle intermediate state, integral of PHID, radians
PHID	Euler roll-angle rate of change, radians/sec
PHIDG	Euler roll angle, degrees
PHIINS	INS Euler roll angle, radians
PHILN	Euler roll-angle linearization dummy variable, radians
PHITR	Euler roll angle required for trim, radians
PHITRDG	Euler roll angle required for trim, degrees
PHIWND	wind-axis bank angle, degrees
PHIZR	Euler roll angle referenced to "PHITR", radians
PHIZRDG	Euler roll angle referenced to "PHITR", degrees
PIC	body-axis roll-rate initial condition, radians/sec
PICDG	body-axis roll-rate initial condition, degrees/sec
PILOTL	3 element array describing location of pilot in FS/BL/WL coordinates, inches
PINS	INS body-axis roll rate, radians/sec
PINSS	3 element past state array for INS roll rate, radians/sec
PITCH	thrust-vectoring pitch-jet-angle command after limit in M/P, degrees
PITCH_CMD_L	left engine thrust-vectoring pitch-jet-angle command limited in M/P, degrees

PITCH_CMD_R	right engine thrust-vectoring pitch-jet-angle command limited in M/P, degrees
PIT_CMD	thrust-vectoring pitch-jet-angle command from RFCS, degrees
PJETAC	thrust-vectoring pitch-jet-angle command from RFCS, degrees
PJETC(1)	final pitch thrust-vector command for left engine, degrees
PJETC(2)	final pitch thrust-vector command for right engine, degrees
PJETLN	pitch-jet-angle perturbation, linearization input, degrees
PJETL1	intermediate pitch-jet-angle perturbation variable during linearization, degrees
PJETTR	total pitch-jet-angle deflection resulting from trim calculations, degrees
PLAL	left engine commanded throttle position (power level angle), degrees
PLALTR	trim value for PLAL, degrees
PLAOLD(1)[1]	left engine commanded throttle position, percent
PLAOLD(2)[1]	right engine commanded throttle position, percent
PLAR	right engine commanded throttle position (power level angle), degrees
PLARTR	trim value for PLAR, degrees
PLIVC[1]	left-inboard-vane command from previous iteration for NASA-0 M/P , inches (Should be saved and reused in subsequent CALLS).
PLOVC[1]	left-outboard-vane command from previous iteration for NASA-0 M/P , inches (Should be saved and reused in subsequent CALLS).
PLTVC[1]	left-top-vane command from previous iteration for NASA-0 M/P , inches (Should be saved and reused in subsequent CALLS).
PRN	high volume output goes to unit=PRN. Passed to Fortran subroutines that write to an output file. (**Default = 9.**)
PRIVC[1]	right-inboard-vane command from previous iteration for NASA-0 M/P , inches (Should be saved and reused in subsequent CALLS).
PROVC[1]	right-outboard-vane command from previous iteration for NASA-0 M/P , inches (Should be saved and reused in subsequent CALLS).
PRTVC[1]	right-top-vane command from previous iteration for NASA-0 M/P , inches (Should be saved and reused in subsequent CALLS. inches
PS	sensor-model roll rate, degrees/sec
PS1	sensor-model intermediate variable in roll-rate computation, degrees/sec
PSI	Euler yaw angle, radians
PSI0	Euler yaw-angle intermediate state, integral of PSID, radians
PSID	Euler yaw-angle rate of change, radians/sec
PSIDG	Euler yaw angle, degrees
PSIINS	INS Euler yaw angle, radians
PSILN	Euler yaw -ngle rate linearization dummy variable, radians. [Constant].
PSITR	Euler yaw angle required for trim, radians
PSITRDG	Euler yaw angle required for trim, degrees
PSTAB	stability-axis roll rate, degrees/sec
PSTATC	static atmospheric pressure, = f(HRF), psf
PSTS	sensor-model static atmospheric pressure, psf
PSTS1	static pressure output from AIRDATA subroutine, psf
PTI	local variable for pressure computation in AIRDATA subroutine, psf
PTIS1	output variable for pressure computation from AIRDATA subroutine, psf
PTRIM	pilot pitch trim button command, inches
PTURB	perturbation value used for pulse or double signal if IDYNCK=1 OR 2. (**Default = 0.**)
PVECC	thrust-adjusted pitch thrust-vector command, degrees
PW	wind-axis roll rate, radians/sec
PWDG	wind-axis roll rate, degrees/sec
Q	body-axis pitch rate, radians/sec
QBAR	aircraft dynamic pressure, lbs/ft^2

QBARS	intermediate variable, = S, lbs
QBARSB	intermediate variable, = Sb, lbs-ft
QBARSC	intermediate variable, = S, lbs-ft
QBARSN	sensor-model dynamic pressure, psf
QC	impact pressure, psf
QCFILT1	sensor-model impact pressure filtered with a 2.5 radian/sec lag, psf
QCFILT2	sensor-model impact pressure filtered with a 10. radian/sec lag, psf
QCIS	sensor-model impact pressure, psf
QCIS1	impact pressure from AIRDATA subroutine, psf
QCX	intermediate value in sensor-model compressible pressure computation, psf
QD	body-axis pitch acceleration, radians/sec^2
QDG	body-axis pitch rate, degrees/sec
QDDG	body-axis pitch acceleration, degrees/sec^2
QIC	body-axis pitch-rate initial condition, radians/sec
QICDG	body-axis pitch-rate initial condition, degrees/sec
QINS	INS pitch rate, radians/sec
QINSS	3 element past--state array for INS pitch rate, radians/sec
QS	sensor-model pitch rate, degrees/sec
QS1	sensor-model intermediate variable in pitch-rate computation, degrees/sec
QW	wind-axis pitch rate, radians/sec
QWDG	wind-axis pitch rate, degrees/sec
R8EST(1)	estimated left engine radius of nozzle exit, inches
R8EST(2)	estimated right engine radius of nozzle exit, inches
R8IN(1)	left engine radius of nozzle exit, inches
R8IN(2)	right engine radius of nozzle exit, inches
R	body-axis yaw rate, radians/sec
RAC	RAV aileron command, degrees
RA2DG	radians to degrees conversion factor, degrees/radians
RCLEFC	RAV collective leading edge-flap command, degrees
RCSC	RAV collective stabilator command, degrees
RCTEFC	RAV collective trailing edge-flap command, degrees
RD	body-axis yaw acceleration, radians/sec^2
RDG	body-axis yaw rate, degrees/sec
RDDG	body-axis yaw acceleration, degrees/sec^2
RDLEFC	RAV differential leading edge-flap command, degrees
RDTC	RAV differential stabilator command, degrees
RDTEFC	RAV differential trailing edge-flap command, degrees
RGTHR	estimated right engine gross thrust, lbs
RHO	air density, slugs/ft^3
RHOSL	air density at sea level, **(= 0.00237688 slugs/ft^3)**
RIC	body-axis yaw-rate initial condition, radians/sec
RICDG	body-axis yaw-rate initial condition, degrees/sec
RINBVC	right-inboard-vane command for NASA-0 M/P, inches
RINS	INS yaw rate, radians/sec
RINSS	3 element past state array for INS yaw rate, radians/sec
RIS	sensor-model pressure ratio, n.d.
RIS1	pressure ratio in AIRDATA subroutine, n.d.
RJETAC	thrust-vectoring roll-jet-angle command from RFCS, degrees
RJETLN	roll-jet-angle perturbation, linearization input, degrees
RJETL1	intermediate roll-jet-angle perturbation variable during linearization, degrees
RJETTR	total roll-jet-angle deflection resulting from trim calculations, degrees
RNPR	estimated right engine nozzle pressure ratio

RNXINS INS x-acceleration, ft/sec^2

RNXS[1] 3 element past-state array used for 5 Hz x-acceleration filter, ft/sec^2

RNYINS INS y-acceleration, ft/sec^2

RNYS[1] 3 element past-state array used for 5 Hz y-acceleration filter, ft/sec^2

RNZINS INS z-acceleration, ft/sec^2

RNZS[1] 3 element past-state array used for 5 Hz z-acceleration filter, ft/sec^2

ROL_CMD thrust-vectoring roll-jet-angle command from RFCS, degrees

ROUTVC right-outboard-vane command for NASA-0 M/P, inches

RPMOLD(1)[1] left engine commanded rpm from previous iteration, rpm

RPMOLD(2)[1] right engine commanded rpm from previous iteration, rpm

RRC RAV rudder command, degrees

RRHO ratio of actual air density to RHOSL, n.d.

RS sensor-model yaw rate, degrees/sec

RS1 sensor-model intermediate variable in yaw-rate computation, degrees/sec

RSTAB stability-axis yaw rate, degrees/sec

RSTKSTP logical to fail right-nose-strake actuator **(Default = .F.)**

RSTKTD logical to fail right-nose-strake actuator in trail damp mode **(Default = .F.)**

RTOPVC right-top-vane command for NASA-0 M/P, inches

RTRIM pilot roll-trim button command, inches, Range (1., 0., -1)

RVECC thrustadjusted roll thrust-vector command, degrees

RW wind-axis yaw rate, radians/sec

RWDG wind-axis yaw rate, degrees/sec

SBERR error term between speed-brake position and command, degrees

SBNLRL speed-brake no-load rate limit, **(= 60/2.5 degrees/sec)**

SFXTRIM conversion factor from radians to degrees for trim variables, n.d.

SFYTRIM conversion factor from radians to degrees for trim variables, n.d.

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)

SIGXSTK value of XSTRAKE signal at current time, = XSTKSG (T)

SIGXTVY value of XTVYAW signal at current time, = XTVYSG (T)

SINCHI sine of sensor-model wind-axis Euler angle, n.d.

SINGAM sine of sensor-model wind-axis Euler angle, n.d.

SINMU sine of sensor-model wind-axis Euler angle, n.d.

SINPHI sine of sensor-model Euler roll angle, n.d.

SINPSI sine of Euler yaw angle, n.d.

SINTHE sine of sensor-model Euler pitch angle, n.d.

SNALF sine of angle of attack at c.g., n.d.

SNALRF sine of angle of attack at a.r.c., n.d.

SNBET sine of angle-of-sideslip at c.g., n.d.

SNPHI sine of roll angle, n.d.

SNPHO2 sine of (PHITR / 2), n.d.

SNPSO2 sine of (PSITR / 2), n.d.

SNTHE sine of pitch angle, n.d.

SNTHO2 sine of (THETR / 2), n.d.

STARTAP frequency-sweep input start amplitude, inches or lbs depending on desired input

STARTFQ	frequency-sweep input start frequency, radians/sec
STOPAP	frequency-sweep input stop amplitude, inches or lbs depending on desired input
STOPFQ	frequency-sweep input stop amplitude, radians/sec
STPRES	sensor-model static pressure, psf
SWRF	aerodynamic (wing) reference area, **(= 400 ft^2)**
T	simulation 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
TANTHE	tangent of pitch angle, n.d.
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
THA	Euler pitch angle, radians
THAINS	INS Euler pitch angle, radians
THE	Euler pitch angle, radians
THE0	Euler pitch-angle intermediate state, integral of THED, radians
THED	Euler pitch-angle rate of change, radians/sec
THEDG	Euler pitch angle, degrees
THELN	Euler pitch-angle linearization dummy variable, radians
THETA	direction of thrust vectoring command in polar coordinates, radians
THETR	Euler pitch angle required for trim, radians
THETRDG	Euler pitch angle required for trim, degrees
THL	total engine roll moment, lbs
THM	total engine pitch moment, lbs
THN	total engine yaw moment, lbs
THLLN	left engine thrust linearization dummy variable, lbs
THRLN	right engine thrust linearization dummy variable, lbs
THRTOT	total net thrust including angle-of-attack effects, TV losses, and drags, lbs
THX	total net thrust in the x-body axis, lbs
THY	total net thrust in the y-body axis, lbs
THZ	total net thrust in the z-body axis, lbs
TLOC(1)	distance from aerodynamic reference point in the x-y plane to the left engine x-axis, ft
TLOC(2)	distance from aerodynamic reference point in the x-y plane to the right engine x-axis, ft
TMPK	temperature, degrees Kelvin
TOTMIL	maximum military thrust, lbs
TRMMING	real trim flag based on LTRFLG =0.0 Operate mode =1.0 Trim mode
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)**
TVJANP(1)	thrust-vectored pitch jet-turning-angle for the left engine in the x-y engine-axis plane, degrees
TVJANP(2)	thrust-vectored pitch jet-turning-angle for the right engine in the x-y engine-axis plane, degrees
TVJANR(1)	thrust-vectored roll jet-turning-angle for the left engine in the y-z engine-axis plane, degrees
TVJANR(2)	thrust-vectored roll jet-turning-angle for the right engine in the y-z engine-axis plane,

	degrees
TVJANY(1)	thrust-vectored yaw jetturning-angle for the left engine in the x-z engine-axis plane, degrees
TVJANY(2)	thrust-vectored yaw jet-turning-angle for the left engine in the x-z engine-axis plane, degrees
TVJPAVG	average of TVJANP(1) and TVJANP(2) , degrees
TVJRAVG	average of TVJANR(1) and TVJANR(2) , degrees
TVJYAVG	average of TVJANY(1) and TVJANY(2) , degrees
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
VANE(1)	left-top-vane command from TV_VANES, degrees
VANE(2)	left-outboard-vane command from TV_VANES, degrees
VANE(3)	left-inboard-vane command from TV_VANES, degrees
VANE(4)	right-top-vane command from TV_VANES, degrees
VANE(5)	right-outboard-vane command from TV_VANES, degrees
VANE(6)	right-inboard-vane command from TV_VANES, degrees
VANEC(1)	left-top-vane command, degrees
VANEC(2)	left-outboard-vane command, degrees
VANEC(3)	left-inboard-vane command, degrees
VANEC(4)	right-top-vane command, degrees
VANEC(5)	right-outboard-vane command, degrees
VANEC(6)	right-inboard-vane command, degrees
VANECO(1)[1]	left-top-vane command from previous iteration, degrees
VANECO(2)[1]	left-outboard-vane command from previous iteration, degrees
VANECO(3)[1]	left-inboard-vane command from previous iteration, degrees
VANECO(4)[1]	right-top-vane command from previous iteration, degrees
VANECO(5)[1]	right-outboard-vane command from previous iteration, degrees
VANECO(6)[1]	right-inboard-vane command from previous iteration, degrees
VANEP(1)	left-top actual-vane position, degrees
VANEP(2)	left-outboard actual vane position, degrees
VANEP(3)	left-inboard actual vane position, degrees
VANEP(4)	right-top actual vane position, degrees
VANEP(5)	right-outboard actual vane position, degrees
VANEP(6)	right-inboard actual vane position, degrees
VANTMP(1)	left-top-vane temperature, degrees Fahrenheit
VANTMP(2)	left-outboard-vane temperature, degrees Fahrenheit
VANTMP(3)	left-inboard-vane temperature, degrees Fahrenheit
VANTMP(4)	right-top-vane temperature, degrees Fahrenheit
VANTMP(5)	right-outboard-vane temperature, degrees Fahrenheit
VANTMP(6)	right-inboard-vane temperature, degrees Fahrenheit
VARF	VA at a.r.c., ft/sec
VEINS	INS velocity (East), ft/sec

VD	body-axis inertial acceleration in y-direction , ft/sec^2
VDA	d/dt of VA, ft/sec^2
VDWG	d/dt of aerodynamic-disturbance component in y-body axis, ft/sec^2
VGS	aircraft ground speed, ft/sec
VIC	body-axis inertial velocity in y-direction i.c., ft/sec
VMAG	magnitude of thrust-vectoring command in polar coordinates, degrees
VNINS	INS velocity (North), ft/sec
VNIN2(1)[1]	left-top-vane actuator command from previous iteration, degrees
VNIN2(2)[1]	left-outboard-vane actuator command from previous iteration, degrees
VNIN2(3)[1]	left-inboard-vane actuator command from previous iteration, degrees
VNIN2(4)[1]	right-top-vane actuator command from previous iteration, degrees
VNIN2(5)[1]	right-outboard-vane actuator command from previous iteration, degrees
VNIN2(6)[1]	right-inboard-vane actuator command from previous iteration, degrees
VNIN3(1)[1]	left-top-vane actuator command from two iterations ago, degrees
VNIN3(2)[1]	left-outboard-vane actuator command from two iterations ago, degrees
VNIN3(3)[1]	left-inboard-vane actuator command from two iterations ago, degrees
VNIN3(4)[1]	right-top-vane actuator command from two iterations ago, degrees
VNIN3(5)[1]	right-outboard-vane actuator command from two iterations ago, degrees
VNIN3(6)[1]	right-inboard-vane actuator command from two iterations ago, degrees
VNOT2(1)[1]	left-top-vane actuator lagged position from previous iteration, degrees
VNOT2(2)[1]	left-outboard-vane actuator lagged position from previous iteration, degrees
VNOT2(3)[1]	left-inboard-vane actuator lagged position from previous iteration, degrees
VNOT2(4)[1]	right-top-vane actuator lagged position from previous iteration, degrees
VNOT2(5)[1]	right-outboard-vane actuator lagged position from previous iteration, degrees
VNOT2(6)[1]	right-inboard-vane actuator lagged position from previous iteration, degrees
VNOT3(1)[1]	left-top-vane actuator lagged position from two iterations ago, degrees
VNOT3(2)[1]	left-outboard-vane actuator lagged position from two iterations ago, degrees
VNOT3(3)[1]	left-inboard-vane actuator lagged position from two iterations ago, degrees
VNOT3(4)[1]	right-top-vane actuator lagged position from two iterations ago, degrees
VNOT3(5)[1]	right-outboard-vane actuator lagged position from two iterations ago, degrees
VNOT3(6)[1]	right-inboard-vane actuator lagged position from two iterations ago, degrees
VS	speed of sound, ft/sec
VSTOW(1)	stowed left-top-vane position, **(Default = -10.)** degrees
VSTOW(2)	stowed left-outboard-vane position, **(Default = -10.)** degrees
VSTOW(3)	stowed left-inboard-vane position, **(Default = -10.)** degrees
VSTOW(4)	stowed right-top-vane position, **(Default = -10.)** degrees
VSTOW(5)	stowed right-outboard-vane position, **(Default = -10.)** degrees
VSTOW(6)	stowed right-inboard-vane position, **(Default = -10.)** degrees
VT	aircraft total atmosphere velocity, ft/sec
VTD	aircraft total atmosphere acceleration, ft/sec^2
VTINV	coefficient to normalize by aircraft total atmosphere velocity, sec/ft
VTRF	aircraft total atmosphere velocity, at a.r.c., ft/sec
VTRINV	coefficient to normalize by aircraft total atmosphere velocity, at a.r.c., sec/ft
VTTR	aircraft total atmosphere velocity after successful trim, ft/sec
VTS	sensor-model aircraft total atmosphere velocity, ft/sec
VVINS	INS velocity (Vertical), ft/sec
VWG	y-body component of aerodynamic-disturbance velocity, ft/sec
VWNDIC	initial wind velocity, ft/sec
VWNDRT	rate of change in wind velocity x1000, ft/sec/kft

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
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
WLRF	water-line coordinate of the aerodynamic reference center, **(= 100.0 inches)**
WNDE	wind vector (East), ft/sec
WNDN	wind vector (North), ft/sec
WNDV	wind vector (Vertical), ft/sec
WT	aircraft weight, lbs
WWG	z-body component of aerodynamic disturbance velocity, ft/sec
XARF	x-direction distance from c.g. to a.r.c., ft
XBAR	normalized x-location of accelerometer from c.g., n.d.
XBARP	normalized x-location of accelerometer at pilot station, n.d.
XCGRF	engine c.g. x-offset, ft
XD	aircraft inertial ground track velocity in North direction, ft/sec
XDDWG	d/dt of XDWG, ft/sec^2
XDOT	aircraft inertial ground track velocity in North direction, ft/sec
XDPALT	internal altitude used in subroutine AOAINS, equal to H, ft
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
XHAOA	internal angle of attack used in subroutine AOAINS, limited to 34°, degrees
XHTAS	internal atmospheric velocity used in subroutine AOAINS, limited to 200 ft/sec, ft/sec
XNRF	x-direction thrust centerline distance, inches (not used but read in trim file)
XPS	Mission computer static pressure, inches hg
XPGY	body-axis roll rate after aliasing filter, computed in subroutine ACCEL, degrees/sec
XQC	Mission computer impact pressure, inches hg
XQGY	body-axis pitch rate after aliasing filter, computed in subroutine ACCEL, degrees/sec
XRGY	body-axis yaw rate after aliasing filter, computed in subroutine ACCEL, degrees/sec
XSTRAKE	switch to activate ANSER control law **(Default=0.0)**
XSTKSG	XSTRAKE position signal array for IDYNCK=3
XTMPR	temperature, degrees Rankin
XTRLM	2 by 33 array with upper and lower limits on XTRIM independent variables, radians
XTRLMDG	2 by 31 array with upper and lower limits on XTRIM independent variables, degrees
XTVYAW	switch to activate yaw thrust vectoring control law **(Default=1.0)**
XTVYSG	XSTRAKE signal array for IDYNCK=3
XX	previous-frame values for angle of attack with gain & bias, degrees
Y1	array that contains NASA-0 RFCS longitudinal outputs
Y2	array that contains NASA-0 RFCS lateral/directional outputs
Y3	array that contains ANSER RFCS lateral/directional outputs
Y10	array that contains NASA-0 RFCS lateral/directional outputs
Y11	array that contains NASA-0 RFCS lateral/directional outputs
Y13	array that contains left and right stabilator, LEF, and TEF command outputs from either NASA-0 or ANSER depending on LNRFCS and LDRFCS
Y18	array that contains ANSER RFCS longitudinal outputs
Y19	array that contains ANSER RFCS longitudinal outputs

YARF	y-direction distance from c.g. to a.r.c., ft
YAW	thrust-vectoring yaw-angle command after pitch priority logic, degrees
YAW_CMD_L	left engine thrust-vectoring yaw-angle command after pitch priority logic, degrees
YAW_CMD_R	left engine thrust-vectoring yaw-angle command after pitch priority logic, degrees
YAW_JET	thrust-vectoring yaw jet-angle command from RFCS, deg
YBAR	normalized y-location of accelerometer from c.g., n.d.
YBARP	normalized y-location of accelerometer at pilot station, n.d.
YCGRF	engine c.g. y-offset, ft
YD	aircraft inertial ground-track velocity in East direction, ft/sec
YDDWG	d/dt of YDWG, ft/sec^2
YDOT	aircraft inertial ground-track velocity in East direction, ft/sec
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
YJETAC	thrust-vectoring yaw jet-angle command from RFCS, deg
YJETC(1)	final yaw thrust-vector command for left engine, degrees
YJETC(2)	final yaw thrust-vector command for right engine, degrees
YJETLN	yaw jet-angle perturbation, linearization input, degrees
YJETL1	intermediate yaw jet-angle perturbation variable during linearization, degrees
YJETTR	total yaw jet-angle deflection resulting from trim calculations, degrees
YNRF	y-direction thrust centerline distance, inches (not used but read in trim file)
YRFCGL	'y' component of the location of left engine/airframe interface point in c.g.-centered body frame, = (BLENGL - BLCG)/12., ft.
YTRIM	pilot yaw trim knob command, n.d.
YVECC	thrust-adjusted yaw thrust-vector command, degrees
YY	previous-frame values for angle-of-sideslip rate with gain & bias, degrees
ZANSL	intermediate left strakecalculation for DTLLN1, degrees
ZANSR	intermediate right strake calculation for DTRLN1, degrees
ZARF	z-direction distance from c.g. to a.r.c., ft
ZBAR	normalized z-location of accelerometer from c.g., n.d.
ZBARP	normalized z-location of accelerometer at pilot station, n.d.
ZBNSL	intermediate left-strake calculation for DTLLN1, degrees
ZBNSR	intermediate right-strake calculation for DTRLN1, degrees
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
ZNRF	z-direction thrust-centerline distance, inches (not used but read in trim file)

[1] These variables are computed are included in the subroutine parameter list to be saved from one interation to the next to enable multiple CALLS to the subroutines for multiple airplanes in the DMS.

[2] Italicized variables are internal parameters used in a subroutine locally.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1996	Technical Memorandum

4. TITLE AND SUBTITLE

Simulation Model of the F/A-18 High Angle-of-Attack Research Vehicle Utilized for the Design of Advanced Control Laws

5. FUNDING NUMBERS

WU 505-68-30-05

6. AUTHOR(S)

Michael D. Messina, Mark E. Strickland, Keith D. Hoffler, Susan W. Carzoo, W. Thomas Bundick, Jessie C. Yeager, and Fred L. Beissner, Jr.

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

NASA Langley Research Center
Hampton, Va 23681-0001

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-110216

11. SUPPLEMENTARY NOTES

Messina, Yeager, and Beissner: Lockheed Martin Engineering & Sciences Co., Hampton, VA; Strickland and Bundick: Langley Research Center, Hampton, VA; Hoffler: ViGYAN, Inc., Hampton, VA; Carzoo: Unisys Corporation, Hampton, VA

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Unclassified-Unlimited

Subject Category - 08

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The *f18harv* six degree-of-freedom nonlinear batch simulation used to support research in advanced control laws and flight dynamics issues as part of NASA's High Alpha Technology Program is described in this report. This simulation models an F/A-18 airplane modified to incorporate a multi-axis thrust-vectoring system for augmented pitch and yaw control power and actuated forebody strakes for enhanced aerodynamic yaw control power. The modified configuration is known as the High Alpha Research Vehicle (HARV). The *f18harv* simulation was an outgrowth of the *f18bas* simulation which modeled the basic F/A-18 with a preliminary version of a thrust-vectoring system designed for the HARV. The preliminary version consisted of two thrust-vectoring vanes per engine nozzle compared with the three vanes per engine actually employed on the F/A-18 HARV. The modeled flight envelope is extensive in that the aerodynamic database covers an angle-of-attack range of -10 degrees to +90 degrees, sideslip range of -20 degrees to +20 degrees, a Mach Number range between 0.0 and 2.0, and an altitude range between 0 and 60000 feet.

14. SUBJECT TERMS

F/A-18 High-Alpha Research Vehicle (HARV), Actuated Nose Strakes for Enhanced Rolling (ANSER), Batch Simulation, aero model, engine model, sensor models, thrust-vector system model, actuator models, non-linear, six-degree-of-freeom, aroedynamic database.

15. NUMBER OF PAGES

161

16. PRICE CODE

A08

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