https://ntrs.nasa.gov/search.jsp?R=20030013626 2019-08-14T13:51:56+00:00Z

NASA/TM-2003-212145

A Collection of Nonlinear Aircraft Simulations in MATLAB

Frederico R. Garza
George Washington University
Joint Institute for the Advancement of Flight Sciences
Langley Research Center, Hampton, Virginia

Eugene A. Morelli
Langley Research Center, Hampton, Virginia

January 2003

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2003-212145

A Collection of Nonlinear Aircraft Simulations in MATLAB

Frederico R. Garza
George Washington University
Joint Institute for the Advancement of Flight Sciences
Langley Research Center, Hampton, Virginia

Eugene A. Morelli
Langley Research Center, Hampton, Virginia

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

January 2003

The use of trademarks or names of manufacturers in the report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Table of Contents

TABLES iv

FIGURES v

ABSTRACT vi

NOMENCLATURE vii

SUPERSCRIPTS: x

SUBSCRIPTS: x

NON-DIMENSIONAL DERIVATIVES: xi

Longitudinal xi

Lateral / Directional xi

I. INTRODUCTION 1

II. EQUATIONS OF MOTION 2

ASSUMPTIONS 2

COLLECTED EQUATIONS OF MOTION 2

ENGINE MODEL 6

AERODYNAMIC MODEL 7

ATMOSPHERIC PROPERTIES 7

AIRCRAFT STATES AND CONTROLS 8

III. GENERAL SIMULATION TOOLS 9

TRIM 9

LINEARIZATION 11

Detailed Linearization Example *12*

NUMERICAL INTEGRATION 15

Euler's Method *16*

Runge-Kutta Method of Order Two *16*

Runge-Kutta Method of Order Four *17*

Adams-Bashforth Method of Order Three *17*

Detailed Numerical Integration Example *17*

IV. AVDS / MATLAB INTERFACE 21
EQUIPMENT / REQUIREMENTS 21
SOFTWARE INSTALLATION 21
INTERACTIVE SIMULATION 22
Joystick Control Input *23*
Initial States for Interactive Simulation *23*
AVDS / MATLAB Piloted Simulation Software *24*
Stick Gain Modification *26*
Control Deflection Exaggeration Factor *27*
Integration Routine *27*
Starting the AVDS / MATLAB Simulation *27*

V. SPECIFIC AIRCRAFT SIMULATION DETAILS 29
F-16 SIMULATION 29
Aerodynamic Model and Database *30*
Engine Model and Database *31*
Mass Properties *33*
F-106B SIMULATION 33
Aerodynamic Model and Database *34*
Engine Model and Database *35*
Mass Properties *35*
F-14 SIMULATION 36
Aerodynamic Model and Database *37*
Engine Model and Database *39*
Mass Properties *39*
F-4 SIMULATION 39
Aerodynamic Model and Database *40*
Engine Model and Database *45*
Mass Properties *45*
FASER SIMULATION 45
Aerodynamic Model and Database *46*
Engine Model and Database *48*
Mass Properties *49*
HL-20 SIMULATION 49
Aerodynamic Model and Database *50*
Dynamic Equations *53*
Mass Properties *53*

X-31 SIMULATION 54
Aerodynamic Model and Database *55*
Engine Model and Database *57*
Mass Properties *57*

A-7 SIMULATION 58
Aerodynamic Model and Database *59*
Engine Model and Database *60*
Mass Properties *60*

VI. GENERATING 3-D MODELS FOR AVDS 62

MODEL CONSTRUCTION REQUIREMENTS 62

MODELING EXAMPLE OF A SIMPLE 3-D AIRCRAFT 62

IMPORTING A COMPLETED 3-D CAD MODEL INTO AVDS 66

VII. CONCLUDING REMARKS 68

VIII. ACKNOWLEDGEMENTS 69

IX. APPENDIX 70

X. REFERENCES 75

Tables

TABLE 1. MASS PROPERTIES OF THE SIMULATED F-16 33
TABLE 2. MASS PROPERTIES OF THE SIMULATED F-106B 36
TABLE 3. MASS PROPERTIES OF THE SIMULATED F-14 39
TABLE 4. MASS PROPERTIES OF THE SIMULATED F-4 45
TABLE 5. MASS PROPERTIES OF THE SIMULATED FASER 49
TABLE 6. MASS PROPERTIES OF THE SIMULATED HL-20 54
TABLE 7. MASS PROPERTIES OF THE SIMULATED X-31 58
TABLE 8. MASS PROPERTIES OF THE SIMULATED A-7 61
TABLE A-1. F-16 SIMULATION PACKAGE (DIRECTORY NAME: F-16) 70
TABLE A-2. F-106B SIMULATION PACKAGE (DIRECTORY NAME: F-106B) 71
TABLE A-3. F-14 SIMULATION PACKAGE (DIRECTORY NAME: F-14) 72
TABLE A-4. F-4 SIMULATION PACKAGE (DIRECTORY NAME: F-4) 72
TABLE A-5. FASER SIMULATION PACKAGE (DIRECTORY NAME: FASER) 73
TABLE A-6. HL-20 SIMULATION PACKAGE (DIRECTORY NAME: HL-20) 73
TABLE A-7. X-31 SIMULATION PACKAGE (DIRECTORY NAME: X-31) 74
TABLE A-8. A-7 SIMULATION PACKAGE (DIRECTORY NAME: A-7) 74

Figures

FIGURE 1. FLOWCHART FOR NUMERICAL LINEARIZATION 12
FIGURE 2. FLOWCHART FOR NONLINEAR SIMULATION 16
FIGURE 3. ELEVATOR PERTURBATION..... 18
FIGURE 4. AVDS AIRCRAFT STATE INITIALIZATION WINDOW 24
FIGURE 5. INTERACTIVE SIMULATION FLOWCHART 25
FIGURE 6. SCREEN SHOT OF THE F-16 AVDS / MATLAB INTERACTIVE SIMULATION .. 26
FIGURE 7. THREE-VIEW OF THE F-16 30
FIGURE 8. THREE-VIEW OF THE F-106B 34
FIGURE 9. THREE-VIEW OF THE F-14 37
FIGURE 10. THREE-VIEW OF THE F-4 40
FIGURE 11. THREE-VIEW OF THE FASER 46
FIGURE 12. THREE-VIEW OF THE HL-20 50
FIGURE 13. THREE-VIEW OF THE X-31 55
FIGURE 14. THREE-VIEW OF THE A-7 59
FIGURE 15. THREE-VIEW OF THE MIG-15 AIRCRAFT..... 63
FIGURE 16. FRONT AND REAR FUSELAGE OPENINGS 64
FIGURE 17. MODELED MIG-15 FUSELAGE SECTION 64
FIGURE 18. 3-D MODEL WITH WING, TAIL, AND CONTROL SURFACES..... 65
FIGURE 19. MODEL WITH WIRE FRAME CANOPY 65
FIGURE 20. COMPLETED 3-D MIG-15 MODEL..... 66
FIGURE 21. INPUT SCREEN FOR *ACNAME.SIM.INI* 67

ABSTRACT

Nonlinear six degree-of-freedom simulations for a variety of aircraft were created using MATLAB. Data for aircraft geometry, aerodynamic characteristics, mass / inertia properties, and engine characteristics were obtained from open literature publications documenting wind tunnel experiments and flight tests. Each nonlinear simulation was implemented within a common framework in MATLAB, and includes an interface with another commercially-available program to read pilot inputs and produce a three-dimensional (3-D) display of the simulated airplane motion. Aircraft simulations include the General Dynamics F-16 Fighting Falcon, Convair F-106B Delta Dart, Grumman F-14 Tomcat, McDonnell Douglas F-4 Phantom, NASA Langley Free-Flying Aircraft for Sub-scale Experimental Research (FASER), NASA HL-20 Lifting Body, NASA / DARPA X-31 Enhanced Fighter Maneuverability Demonstrator, and the Vought A-7 Corsair II. All nonlinear simulations and 3-D displays run in real time in response to pilot inputs, using contemporary desktop personal computer hardware. The simulations can also be run in batch mode. Each nonlinear simulation includes the full nonlinear dynamics of the bare airframe, with a scaled direct connection from pilot inputs to control surface deflections to provide adequate pilot control. Since all the nonlinear simulations are implemented entirely in MATLAB, user-defined control laws can be added in a straightforward fashion, and the simulations are portable across various computing platforms. Routines for trim, linearization, and numerical integration are included. The general nonlinear simulation framework and the specifics for each particular aircraft are documented.

NOMENCLATURE

α	angle of attack, deg or rad
β	angle of sideslip, deg or rad
Δt	numerical integration time step, s
$\delta_{a,\,left}$	left aileron control surface deflection, deg
$\delta_{a,\,right}$	right aileron control surface deflection, deg
δ_a	aileron control surface deflection $=\left(\delta_{a,\,right}-\delta_{a,\,left}\right)/2$, deg
δ_c	canard control surface deflection, deg
δ_{df}	differential flap deflection, deg
δ_e	elevator control surface deflection, deg
δ_{lf}	leading edge flap deflection, deg
δ_{lg}	landing gear deflection, deg
δ_{nf}	negative flap deflection, deg
δ_{pf}	positive flap deflection, deg
δ_r	rudder control surface deflection, deg
δ_{sb}	speed brake deflection, deg
δ_{sp}	symmetric wing spoiler deflection, deg
δ_{tf}	trailing edge flap deflection, deg
δ_{th}	throttle deflection
$\delta_{th,left}$	left engine throttle deflection
$\delta_{th,right}$	right engine throttle deflection
δ_{tv}	outboard thrust-vectoring paddle deflection, deg
Φ	Euler roll angle, rad

γ	flight path angle, rad
Θ	Euler pitch angle, rad
ρ	atmospheric density, slug/ft^3
τ_{eng}	engine power level time constant, s
Ψ	Euler yaw angle, rad
ω_{eng}	angular velocity of the engine rotating mass about the X body-axis, rad/s
a_X	linear acceleration along X body-axis, *g*
a_Y	linear acceleration along Y body-axis, *g*
a_Z	linear acceleration along Z body-axis, *g*
$\boldsymbol{A}$	system matrix
b	wing span, ft
$\boldsymbol{B}$	input matrix
$\bar{c}$	mean aerodynamic chord, ft
$\boldsymbol{C}$	output matrix
$\boldsymbol{c}$	vector of mass / inertia properties
$c_1 \ldots c_9$	inertia constants
C_A	non-dimensional axial force coefficient, $C_A = -C_X$
C_D	drag coefficient
$C_{D_{min}}$	minimum drag coefficient
C_l	non-dimensional rolling moment coefficient
C_L	lift coefficient
C_{L_α}	lift-curve slope
C_m	non-dimensional pitching moment coefficient
C_N	non-dimensional normal force coefficient, $C_N = -C_Z$
C_n	non-dimensional yawing moment coefficient
C_X	non-dimensional X body-axis force coefficient
C_Y	non-dimensional Y body-axis force coefficient

C_Z	non-dimensional Z body-axis force coefficient
$\boldsymbol{D}$	feed-through matrix
g	earth gravitational acceleration = 32.174 ft/s^2
h	altitude above mean sea level, ft
h_{eng}	engine angular momentum about X body-axis, slug-ft^2/s
I_{eng}	X body-axis moment of inertia for the engine rotating mass, slug-ft^2
I_X	X body-axis moment of inertia, slug-ft^2
I_{XZ}	X-Z body-axis product of inertia, slug-ft^2
I_Y	Y body-axis moment of inertia, slug-ft^2
I_Z	Z body-axis moment of inertia, slug-ft^2
m	aircraft mass, slugs
M	Mach number
$O(\ )$	order
p	X body-axis angular velocity component, rad/s
P_c	commanded engine power level, percent
P_a	actual engine power level, percent
q	Y body-axis angular velocity component, rad/s
$\bar{q}$	dynamic pressure, lb/ft^2
r	Z body-axis angular velocity component, rad/s
S	wing reference area, ft^2
T	engine thrust, lb
T^*	temperature factor
T_{idle}	idle engine thrust, lb
T_l	rolling moment about X body-axis generated by thrust, ft-lb
T_m	pitching moment about Y body-axis generated by thrust, ft-lb
T_{max}	maximum engine thrust, lb
$T_{max,ab}$	engine thrust with maximum afterburner, lb

T_{mil}	military engine thrust, lb
$T_{min,ab}$	engine thrust with minimum afterburner, lb
T_n	yawing moment about Z body-axis generated by thrust, ft-lb
T_X	X body-axis thrust component, lb
T_Y	Y body-axis thrust component, lb
T_Z	Z body-axis thrust component, lb
u	X body-axis translational velocity component, ft/s
$\boldsymbol{u}$	control vector
v	Y body-axis translational velocity component, ft/s
V_t	airspeed, ft/s
w	Z body-axis translational velocity component, ft/s
$\boldsymbol{x}$	state vector
$x_{c.g.}$	longitudinal center of gravity location, fraction of $\bar{c}$
$x_{c.g.,ref}$	reference longitudinal center of gravity location, fraction of $\bar{c}$
x_E	X earth-axis position, ft
$\boldsymbol{y}$	output vector
y_E	Y earth-axis position, ft

SUPERSCRIPTS:

$\dot{}$	time derivative

SUBSCRIPTS:

0	basic or trim value
left	left
right	right
S	stability axes
E	earth axes

NON-DIMENSIONAL DERIVATIVES:

Longitudinal

$$C_{L_i} = \frac{\partial C_X}{\partial i} \qquad C_{L_q} = \frac{\partial C_X}{\partial \frac{q\bar{c}}{2V_t}}$$

$$C_{X_i} = \frac{\partial C_X}{\partial i} \qquad C_{X_q} = \frac{\partial C_X}{\partial \frac{q\bar{c}}{2V_t}}$$

$$C_{Z_i} = \frac{\partial C_Z}{\partial i} \qquad C_{Z_q} = \frac{\partial C_Z}{\partial \frac{q\bar{c}}{2V_t}}$$

$$C_{m_i} = \frac{\partial C_m}{\partial i} \qquad C_{m_q} = \frac{\partial C_m}{\partial \frac{q\bar{c}}{2V_t}}$$

where $i = V_t, \alpha, \delta_e,$ and/or other controls

Lateral / Directional

$$C_{Y_j} = \frac{\partial C_Y}{\partial j} \qquad C_{Y_k} = \frac{\partial C_Y}{\partial \frac{kb}{2V_t}}$$

$$C_{l_j} = \frac{\partial C_l}{\partial j} \qquad C_{l_k} = \frac{\partial C_l}{\partial \frac{kb}{2V_t}}$$

$$C_{n_j} = \frac{\partial C_n}{\partial j} \qquad C_{n_k} = \frac{\partial C_n}{\partial \frac{kb}{2V_t}}$$

where $j = \beta, \delta_a, \delta_r,$ and/or other controls and $k = p, r, p_S, r_S$

I. INTRODUCTION

Nonlinear aircraft simulations are useful for dynamic analysis, control law design and validation, guidance and trajectory studies, air combat investigations, pilot training, and many other tasks. Most real-time nonlinear aircraft simulations are implemented in programming languages such as FORTRAN or C++, for historical reasons or to achieve the required execution speed. Advances in desktop computing capabilities have made it possible to implement real-time nonlinear aircraft simulations in an advanced computing environment called MATLAB[1], using relatively inexpensive desktop personal computer hardware. MATLAB is easy to learn and program, is portable across many computing platforms without modification, and is a popular tool for scientific and engineering computations. The speed of current desktop computing hardware is sufficient for real-time execution of nonlinear aircraft simulations programmed in MATLAB, including numerical integration of the nonlinear equations of motion, general nonlinear aerodynamic and engine model computations, and interface with software to read pilot inputs and display of a pilot view or a 3-D view of the aircraft in motion from an arbitrary vantage point outside the aircraft.

In addition to these computational hardware and software capabilities, reports available in the open literature contain aircraft geometry, mass / inertia properties, and extensive wind tunnel and flight test aerodynamic data for a variety of different aircraft, many of which are still in service. Data on engine performance is less common in the open literature, but available nonetheless.

This report describes realistic and easy-to-use nonlinear aircraft simulations implemented in MATLAB for a variety of different aircraft whose particular characteristics were available in the open literature. The nonlinear simulations include a general framework for the nonlinear equations of motion and a set of utility programs that apply to all the aircraft. For each individual aircraft, routines that implement the specific aerodynamic, propulsion, and mass / inertia characteristics of the aircraft were developed and documented.

Aircraft simulations included in this report are the General Dynamics F-16 Fighting Falcon, Convair F-106B Delta Dart, Grumman F-14 Tomcat, McDonnell Douglas F-4 Phantom, NASA Langley Free-Flying Aircraft for Sub-scale Experimental Research (FASER), NASA HL-20 Lifting Body, NASA / DARPA X-31 Enhanced Fighter Maneuverability Demonstrator, and the Vought A-7 Corsair II. Complete MATLAB simulations for each individual aircraft were developed in independent software packages. Each software package includes the capability to calculate aircraft trim conditions, linearize the equations of motion, run batch simulations of linear and nonlinear aircraft dynamics, and perform real-time interactive flight simulation. A commercially-available program called Aviator Visual Design Simulator (AVDS)[2] is used to interface each MATLAB nonlinear aircraft simulation with a human pilot in real time. Software to run each MATLAB simulation and exchange data with AVDS in real time is included in the simulation packages, along with 3-D model definition files for

each aircraft. The 3-D model definition files are used by AVDS to create the 3-D display of the aircraft in flight.

Full documentation of each nonlinear aircraft simulation is included in this work, along with instructions for implementing the simulation software. Examples illustrate the steps required to execute the trim, linearization, and nonlinear simulation routines, and run the MATLAB nonlinear aircraft simulations in real time using AVDS. A short tutorial detailing the creation and importation of a 3-D Computer-Aided Design (CAD) aircraft model into AVDS is also provided.

II. EQUATIONS OF MOTION

Aircraft nonlinear equations of motion were implemented in MATLAB. Nonlinear aerodynamic and propulsion models were used to simulate the applied forces and moments for each aircraft. Usually, the aerodynamic and propulsion models were implemented as linear interpolations using look-up tables in a database. In some cases, analytic polynomial models were used to model the aerodynamic and propulsion dependencies on state and control variables. Some simplifying assumptions were made in the development of the nonlinear aircraft simulations, as described next.

ASSUMPTIONS

Each modeled aircraft was assumed rigid with constant mass density and symmetry about the X-Z plane in body axes. Forces and moments acting on the aircraft were from aerodynamics, propulsion, and gravity. Thrust was assumed to act along the X body-axis and through the center of gravity. A stationary atmosphere was assumed, with aircraft flight limited to altitudes less than 50,000 feet and subsonic speeds. The earth's curvature was ignored and the earth was assumed to be fixed in inertial space, so that earth axes were considered inertial axes. The gravity field was assumed to be uniform, so that the center of mass and the center of gravity were coincident, and there were no gravity moments or changes in gravity with altitude.

Full nonlinear six-degree-of-freedom rigid body translational and rotational aircraft motion was modeled, neglecting relative motion of the aircraft internal components, structural distortion, and sloshing of liquid fuel. The gyroscopic effects caused by rotating engine turbomachinery were included for some aircraft. Airplane rotation conventions follow the usual right-hand rule. Control surface deflections follow a standard right-hand rule sign convention, in which the fingers curl in the direction of positive control surface deflections when the thumb is pointed along the control surface hinge line in the direction of a positive body axis.

COLLECTED EQUATIONS OF MOTION

The six nonlinear rigid-body equations of motion were used to model the nonlinear aircraft dynamics in translational and rotational motion. These equations are[3-7]:

$$\dot{u} = rv - qw - g\,sin\,\Theta + \frac{\bar{q}SC_X + T}{m}$$

$$\dot{v} = pw - ru + g\,cos\,\Theta\,sin\,\Phi + \frac{\bar{q}SC_Y}{m} \tag{1}$$

$$\dot{w} = qu - pv + g\,cos\,\Theta\,cos\,\Phi + \frac{\bar{q}SC_Z}{m}$$

$$\dot{p}I_X - \dot{r}I_{XZ} = \bar{q}SbC_l - qr\left(I_Z - I_Y\right) + qpI_{xz}$$

$$\dot{q}I_Y = \bar{q}S\bar{c}C_m - pr\left(I_X - I_Z\right) - \left(p^2 - r^2\right)I_{XZ} - rh_{eng} \tag{2}$$

$$\dot{r}I_Z - \dot{p}I_{XZ} = \bar{q}SbC_n - pq\left(I_Y - I_X\right) - qrI_{XZ} + qh_{eng}$$

where h_{eng} is the magnitude of the angular momentum vector for the rotating mass of the engine, assumed to act along the positive X body-axis,

$$\boldsymbol{h}_{eng} = \begin{bmatrix} h_{eng} & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} I_{eng}\omega_{eng} & 0 & 0 \end{bmatrix}^T$$

It is preferable to write the translational equations in terms of $V_t, \alpha,$ and β instead of $u, v,$ and w, because $V_t, \alpha,$ and β can be measured directly on real aircraft, and have a more direct relationship to piloting and the aerodynamic forces and moments. The relationships between V_t, α, β and u, v, w are:

$$V_t = \sqrt{u^2 + v^2 + w^2}$$

$$\alpha = tan^{-1}\left(\frac{w}{u}\right) \tag{3}$$

$$\beta = sin^{-1}\left(\frac{v}{V_t}\right)$$

and

$$u = V_t\,cos\,\alpha\,cos\,\beta$$

$$v = V_t\,sin\,\beta \tag{4}$$

$$w = V_t\,sin\,\alpha\,cos\,\beta$$

Differentiating the equations (3) with respect to time gives

$$\dot{V}_t = \frac{u\dot{u} + v\dot{v} + w\dot{w}}{V_t}$$

$$\dot{\alpha} = \frac{u\dot{w} - w\dot{u}}{u^2 + w^2} \tag{5}$$

$$\dot{\beta} = \frac{V_t\dot{v} - v\dot{V}_t}{V_t^2\sqrt{1 - \left(\frac{v}{V_t}\right)^2}}$$

The translational states for the nonlinear aircraft simulations are $V_t, \alpha,$ and β. The time derivatives of these translational states are found by computing $\dot{u}, \dot{v},$ and $\dot{w}$ from (1), then computing $u, v,$ and w from (4) using current values of $V_t, \alpha,$ and β, then substituting these $u, v,$ and w values and the current value of V_t into (5) to obtain $\dot{V}_t, \dot{\alpha},$ and $\dot{\beta}$ for use in the numerical integration routines. This approach avoids the lengthy nonlinear calculations associated with the equations of motion written in wind axes, and allows aircraft non-dimensional aerodynamic coefficient data to be implemented in the body-axis coordinate system, with no need for converting the data to wind axes components.

For the rotational motion, straightforward algebraic manipulation transforms the equations (2) into a form suitable for numerical integration, with only one time derivative on the left side of each equation,

$$\dot{p} = \left(c_1 r + c_2 p + c_4 h_{eng}\right) q + \bar{q} S b \left(c_3 C_l + c_4 C_n\right)$$

$$\dot{q} = \left(c_5 p - c_7 h_{eng}\right) r - c_6 \left(p^2 - r^2\right) + \bar{q} S \bar{c} c_7 C_m \tag{6}$$

$$\dot{r} = \left(c_8 p - c_2 r + c_9 h_{eng}\right) q + \bar{q} S b \left(c_4 C_l + c_9 C_n\right)$$

where

$$c_1 = \frac{(I_Y - I_Z)I_Z - I_{XZ}^2}{I_X I_Z - I_{XZ}^2} \qquad c_2 = \frac{(I_X - I_Y + I_Z)I_{XZ}}{I_X I_Z - I_{XZ}^2} \qquad c_3 = \frac{I_Z}{I_X I_Z - I_{XZ}^2}$$

$$c_4 = \frac{I_{XZ}}{I_X I_Z - I_{XZ}^2} \qquad c_5 = \frac{I_Z - I_X}{I_Y} \qquad c_6 = \frac{I_{XZ}}{I_Y} \tag{7}$$

$$c_7 = \frac{1}{I_Y} \qquad c_8 = \frac{(I_X - I_Y)I_X - I_{XZ}^2}{I_X I_Z - I_{XZ}^2} \qquad c_9 = \frac{I_X}{I_X I_Z - I_{XZ}^2}$$

The nonlinear aircraft simulation also includes rotational kinematic equations and navigation equations. The rotational kinematic equations, which relate Euler angular rates to body-axis angular rates, are given by:

$$\begin{aligned} \dot{\Phi} &= p + \tan\Theta\left(q \sin\Phi + r \cos\Phi\right) \\ \dot{\Theta} &= q \cos\Phi - r \sin\Phi \\ \dot{\Psi} &= \frac{q \sin\Phi + r \cos\Phi}{\cos\Theta} \end{aligned} \tag{8}$$

Equations (8) are nonlinear state equations for the Euler angles $\Phi, \Theta,$ and Ψ, in a form suitable for numerical integration. The rotational kinematic equations describe the time evolution of the aircraft attitude angles, which are required to properly resolve the gravity force along the aircraft body axes in (1). A singularity exists for the rotational kinematic state equations at $\Theta = \pm 90^\circ$. This is eliminated within the simulation code by limiting Θ to the ranges 0° to ±89.99° and ±90.01° to ±180°. This limits $\left|(\cos\Theta)^{-1}\right|$ to a maximum value of approximately 5730, which allows sufficiently accurate calculations of the kinematic state derivatives near the singularity. In the nonlinear simulation, Θ automatically skips from ±89.99° to ±90.01° and vice-versa depending on the direction of aircraft pitch angle motion near $\Theta = \pm 90^\circ$.

The navigation equations relate aircraft translational velocity components in body axes to earth-axis components, neglecting wind effects. These differential equations describe the time evolution of the position of the aircraft c.g. relative to earth axes:

$$
\begin{aligned}
\dot{x}_E &= u\cos\Psi\cos\Theta + v\left(\cos\Psi\sin\Theta\sin\Phi - \sin\Psi\cos\Phi\right) \\
&\quad + w\left(\cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi\right) \\
\dot{y}_E &= u\sin\Psi\cos\Theta + v\left(\sin\Psi\sin\Theta\sin\Phi + \cos\Psi\cos\Phi\right) \\
&\quad + w\left(\sin\Psi\sin\Theta\cos\Phi - \cos\Psi\sin\Phi\right) \\
\dot{h} &= u\sin\Theta - v\cos\Theta\sin\Phi - w\cos\Theta\cos\Phi
\end{aligned}
\tag{9}
$$

Assuming thrust acts along the X body-axis, body-axis accelerations a_X, a_Y, and a_Z are calculated from

$$
\begin{aligned}
a_X &= \frac{\bar{q}SC_X + T}{mg} \\
a_Y &= \frac{\bar{q}SC_Y}{mg} \\
a_Z &= \frac{\bar{q}SC_Z}{mg}
\end{aligned}
\tag{10}
$$

Equations (1), (4), (5), (6), (7), (8), and (9) are the nonlinear aircraft equations of motion for each aircraft simulation. These nonlinear dynamic equations are implemented for each simulated aircraft in functions called *ACname_deq.m*, where *ACname* is the aircraft name (e.g., *F16_deq.m*, *F106B_deq.m*, etc.). Body-axis accelerations a_X, a_Y, and a_Z are also calculated in each *ACname_deq.m* function, using equations (10).

ENGINE MODEL

The engine power dynamic response was modeled with an additional state equation as a simple first order lag in the actual power level response to commanded power level:

$$
\dot{P}_a = \frac{1}{\tau_{eng}}\left(P_c - P_a\right) \tag{11}
$$

Commanded power level was computed as a function of throttle position,

$$
P_c = P_c\left(\delta_{th}\right)
$$

and the engine power level time constant τ_{eng} was chosen at or near 1 sec. Engine thrust force was typically computed by linear interpolation of an engine thrust database, as a function of actual power level, altitude, and Mach number,

$$T = T\left(P_a, h, M\right)$$

For each aircraft simulation, calculation of the engine thrust occurs in a function called *ACname_engine.m*, where *ACname* is the aircraft name (e.g., *F16_engine.m*, *F106B_engine.m*, etc.). The routine *tgear.m* implements the throttle gearing, which translates the throttle deflection in the interval [0,1] to commanded power level in the interval [0,100], but not always in a linear fashion. The routine *rtau.m* computes the engine power level time constant τ_{eng}, and *pdot.m* implements the engine thrust dynamics in equation (11).

For many of the simulated aircraft, engine data was not available. In these cases, the engine model for the F-16 aircraft was used, and the output thrust was scaled to match the engine thrust of the specific aircraft.

AERODYNAMIC MODEL

Aerodynamic forces and moments acting on each aircraft were characterized in terms of non-dimensional coefficients in body axes. Typically, the non-dimensional force and moment coefficients were calculated from an expression that included separate build-up functions for each non-dimensional coefficient. The value for each build-up function was usually found by linear interpolation of aerodynamic data, using current values of the states and controls. For values of states and controls outside the range of the available data, the interpolation routines extrapolated linearly using the nearest data points. In a few cases, build-up functions or complete non-dimensional coefficients were computed using analytical polynomial expressions involving state and control variables. Once the non-dimensional aerodynamic force and moment coefficients were calculated, these values were used in equations (1) and (6) with thrust, mass / inertia properties, geometry, and current states and controls to compute state time derivatives for the aircraft dynamics. Subsequent sections describing the individual aircraft simulations contain details of the aerodynamic model for each nonlinear aircraft simulation.

ATMOSPHERIC PROPERTIES

Air density and the speed of sound were calculated using relations modeling the U.S. Standard Atmosphere[3]. Required quantities that depend on these atmospheric properties, namely Mach number *M*, and dynamic pressure $\bar{q}$, were also calculated. The relationships are:

$$T^* = 1 - 0.703 \times 10^{-5}\, h$$

$$\rho = 0.002377\left(T^*\right)^{4.14}$$

$$M = \begin{cases} \dfrac{V_t}{\sqrt{1.4(1716.3)390}} & h \geq 35{,}000 \text{ ft} \\ \dfrac{V_t}{\sqrt{1.4(1716.3)\left(519T^*\right)}} & h < 35{,}000 \text{ ft} \end{cases} \tag{12}$$

$$\bar{q} = \frac{1}{2}\rho V_t^2$$

The function *atm.m* was used to compute all the quantities from equations (12), given h and V_t.

AIRCRAFT STATES AND CONTROLS

For a typical aircraft, the state vector was:

$$\boldsymbol{x} = \left[V_t \quad \alpha \quad \beta \quad \Phi \quad \Theta \quad \Psi \quad p \quad q \quad r \quad x_E \quad y_E \quad h \quad P_a\right]^T \tag{13}$$

The elements of the state vector are associated with equations (1), (4), (5), (6), (7), (8), (9), and (11). A typical control vector was:

$$\boldsymbol{u} = \left[\delta_{th} \quad \delta_e \quad \delta_a \quad \delta_r\right]^T \tag{14}$$

The number of control inputs and engine power states varied depending on the number of control surfaces and aircraft engines. The set of coupled, nonlinear, first-order ordinary differential equations that comprise the simulation model can be represented by the vector differential equation:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \tag{15}$$

The output equations can be represented by the vector equation:

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})$$

III. GENERAL SIMULATION TOOLS

Approximating the solution to the nonlinear equations of motion requires a suitable numerical integration technique. Linearization of the nonlinear equations of motion requires appropriate numerical gradient techniques, which are necessary to calculate elements of the linearized system matrices. Numerical algorithms for both these tasks are included in the software. Reference [8] provides background information on the numerical techniques applied. Runge-Kutta and Adams-Bashforth numerical integration routines are used to approximate solutions to the nonlinear equations of motion. Finite difference numerical gradient procedures are used to determine the elements of the linearized system matrices. The scripts called *gen_ACname_model.m*, where *ACname* is the simulated aircraft name (e.g., *gen_F16_model.m*, *gen_F106B_model.m*, etc.), demonstrate the use of the trim and linearization tools for each simulated aircraft.

TRIM

A linearized form of the state-space model for the nonlinear equations of motion is obtained using a first-order multivariable Taylor series expansion about an equilibrium flight condition known as trim. At the trim condition, all translational and rotational accelerations are equal to zero. This condition is described by the equation:

$$\dot{\boldsymbol{x}}_0 = 0 = \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0) \tag{16}$$

where only the state equations for V_t, α, β, p, q, and r are included in equation (16). The six force and moment equations are augmented with a rate of climb constraint, and a coordinated turn constraint, for a total of eight trim equations[3,4]:

$$0 = \dot{V}_t = \frac{u\dot{u} + v\dot{v} + w\dot{w}}{V_t}$$

$$0 = \dot{\alpha} = \frac{u\dot{w} - w\dot{u}}{u^2 + w^2}$$

$$0 = \dot{\beta} = \frac{V_t\dot{v} - v\dot{V}_t}{V_t^2\sqrt{1 - \left(\frac{v}{V_t}\right)^2}}$$

$$0 = \dot{p} = \left(c_1 r + c_2 p + c_4 h_{eng}\right) q + \bar{q} S b \left(c_3 C_l + c_4 C_n\right)$$

$$0 = \dot{q} = \left(c_5 p - c_7 h_{eng}\right) r - c_6 \left(p^2 - r^2\right) + \bar{q} S \bar{c} c_7 C_m \tag{17}$$

$$0 = \dot{r} = \left(c_8 p - c_2 r + c_9 h_{eng}\right) q + \bar{q} S b \left(c_4 C_l + c_9 C_n\right)$$

$$0 = tan\,\Theta - \frac{ab + sin\,\gamma\sqrt{a^2 - sin^2\,\gamma + b^2}}{a^2 - sin^2\,\gamma} \quad ; \quad \Theta \neq 90^\circ \qquad \left[\text{rate of climb constraint}\right]$$

$$0 = G\,cos\,\beta\left(sin\,\alpha\,tan\,\Theta + cos\,\alpha\,cos\,\Phi\right) \quad ; \quad \Theta \neq 90^\circ \qquad \left[\text{coordinated turn constraint}\right]$$

where

$$a = cos\,\alpha\,cos\,\beta$$

$$b = sin\,\Phi\,sin\,\beta + cos\,\Phi\,sin\,\alpha\,cos\,\beta$$

$$G = \frac{\dot{\Psi} V_t}{g}$$

With eight trim equations to find the trim condition, any combination of eight state or control surface deflection variables can be free to vary so that the trim equations can be satisfied.

User-defined values for fixed states, control surface deflections, and trim values for γ and turn rate $\dot{\Psi}$ are set in the function *ACname_trm.m*, where *ACname* is the simulated aircraft name (e.g., *F16_trm.m*, *F106B_trm.m*, etc.), via code modification. The *ACname_trm.m* function calls the *ACname_deq.m* function to implement the state derivative equations included in the trim equations. Pitch angle Θ is limited to the ranges 0° to ±89.99° and ±90.01° to ±180° in the function *ACname_deq.m*, as described above. This eliminates the singularity at $\Theta = \pm 90^\circ$ within the rate of climb and coordinated turn constraint equations. In addition, the engine power level dynamics in equation (11) are bypassed for the trimming operation, so P_a is set equal to P_c inside *ACname_trm.m*.

Trim solutions are obtained by solving the eight nonlinear trim equations above, using the function *solve.m*. Function *solve.m* uses a Newton-Raphson iterative solution technique, where gradients are calculated in *grad.m* by central finite differences. Trim solution convergence is determined using *cnvrg.m*. Convergence is achieved when the absolute value of every trim equation is less than the solution tolerance, and the absolute value of every change in the free states and controls for successive iterations has reduced to less than the solution tolerance. The solution tolerance is specified as 1×10^{-8} within the code. When the trim search is successful, the words `CONVERGENCE CRITERIA SATISFIED` are output to the MATLAB command window.

LINEARIZATION

A linear, time-invariant (LTI) system representing aircraft dynamics for small perturbations about a reference trim condition is given by the state and output equations:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{aligned} \tag{18}$$

Elements of the constant-coefficient matrices in the linearized dynamic equations are gradients evaluated at the trim values $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$:

$$\begin{aligned} \boldsymbol{A} &\equiv \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_0,\boldsymbol{u}_0} = n \times n \text{ matrix} \qquad & \boldsymbol{B} &\equiv \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}_0,\boldsymbol{u}_0} = n \times m \text{ matrix} \\ \boldsymbol{C} &\equiv \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_0,\boldsymbol{u}_0} = l \times n \text{ matrix} \qquad & \boldsymbol{D} &\equiv \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}_0,\boldsymbol{u}_0} = l \times m \text{ matrix} \end{aligned} \tag{19}$$

where n is the number of elements in the state vector, m is the number of elements in the control vector, and l is the number of elements in the output vector. The gradients are obtained numerically by perturbing each state and control input independently and recording the changes in the trimmed state and output equations. This is done using the numerical technique of central finite differences:

$$\begin{aligned} \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_0,\boldsymbol{u}_0} &= \frac{\boldsymbol{f}(\boldsymbol{x}_0 + \frac{\Delta \boldsymbol{x}}{2}, \boldsymbol{u}_0) - \boldsymbol{f}(\boldsymbol{x}_0 - \frac{\Delta \boldsymbol{x}}{2}, \boldsymbol{u}_0)}{\Delta \boldsymbol{x}} \\ \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}_0,\boldsymbol{u}_0} &= \frac{\boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0 + \frac{\Delta \boldsymbol{u}}{2}) - \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0 - \frac{\Delta \boldsymbol{u}}{2})}{\Delta \boldsymbol{u}} \end{aligned} \tag{20}$$

where $\Delta \boldsymbol{x}$ and $\Delta \boldsymbol{u}$ are the perturbations of the state and control variables, respectively. In the simulations, the output vector is the state vector $\left(\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{x}\right)$, so $\boldsymbol{C}$ is an identity matrix, and $\boldsymbol{D}$ is a null matrix. A flowchart describing the numerical linearization scheme is shown in figure 1.

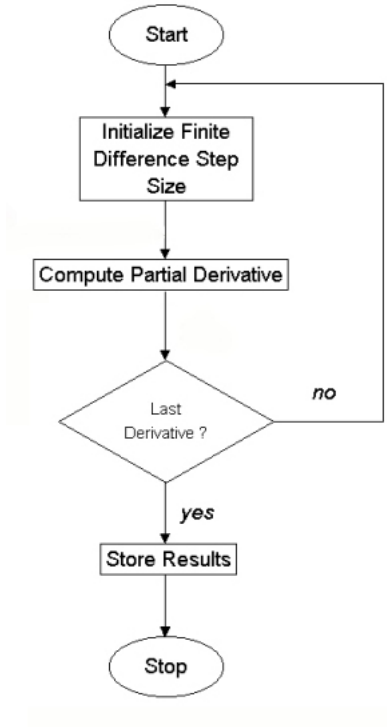

Figure 1. Flowchart for Numerical Linearization

For a reference trim condition of straight and level flight, linearization results in two decoupled sets of linear, constant-coefficient differential equations for longitudinal and lateral / directional motion. The linearization is valid for small perturbations about the reference trim condition.

Detailed Linearization Example

The trim state and control vectors ($\boldsymbol{x}_0$ and $\boldsymbol{u}_0$) are assembled in the function *ic_ftrm.m,* for use in *lnze.m*, which computes the linear system matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$, using central finite differences. Perturbations equal to 1% of the trim reference values are used for each state and control variable during gradient calculation for the elements of the system matrices. If the trim state or control is zero, the perturbation size is set to 0.01. Setting the variable `lonflg` to 1 within the MATLAB workspace before executing *gen_ACname_model.m*, where *ACname* is the simulated aircraft name (e.g., *gen_F16_model.m*, *gen_F106B_model.m*, etc.), produces the longitudinal system matrices. Setting `lonflg` to 0 provides the lateral / directional system matrices. Typical constant-coefficient longitudinal or lateral / directional system matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$, for an aircraft with controls for throttle, elevator, aileron, and rudder deflection, are created in the form:

For `lonflg` = 1:

$$\begin{bmatrix} \Delta\dot{V}_t \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\Theta} \\ \Delta\dot{P}_a \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} \Delta V_t \\ \Delta\alpha \\ \Delta q \\ \Delta\Theta \\ \Delta P_a \end{bmatrix} + \boldsymbol{B}\begin{bmatrix} \Delta\delta_{th} \\ \Delta\delta_e \end{bmatrix} \qquad \boldsymbol{y} = \boldsymbol{C}\begin{bmatrix} \Delta V_t \\ \Delta\alpha \\ \Delta q \\ \Delta\Theta \\ \Delta P_a \end{bmatrix} + \boldsymbol{D}\begin{bmatrix} \Delta\delta_{th} \\ \Delta\delta_e \end{bmatrix}$$

For `lonflg` ≠ 1:

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\Phi} \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\Phi \end{bmatrix} + \boldsymbol{B}\begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \qquad \boldsymbol{y} = \boldsymbol{C}\begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\Phi \end{bmatrix} + \boldsymbol{D}\begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix}$$

The following example illustrates the procedure used to trim the F-16 simulation and calculate the longitudinal and lateral / directional linearized system matrices.

Fixed Quantities		Free States / Controls
$V_t = 500$ ft/s	$x_E = 0$ ft	
$p = 0$ rad/s	$y_E = 0$ ft	
$q = 0$ rad/s	$h = 10{,}000$ ft	α, β, Φ, Θ, δ_{th}, δ_e, δ_a, and δ_r
$r = 0$ rad/s	$\gamma = 0.349$	
$\Psi = 0$ rad	$\dot{\Psi} = 0.052$ rad/s	
$x_{c.g.} = 0.3$		

Step 1. Modify *F16_trm.m* to account for the desired trim condition, indicating the fixed and free state and control variables. Within this file, the variables `xfree` and `xinit` control how the states are handled for trim. The definitions of `xfree` and `xinit` are located on lines 117 and 118, respectively. Line 117 and 118 must be modified to read:

```
xfree = [0,1,1,1,1,0,0,0,0,0,0,0,0]';
xinit = [500,0,0,0,0,0,0,0,0,0,0,10000,0]';
```

In the vector `xfree`, a 0 element indicates that the corresponding state is fixed, while 1 specifies that the corresponding state is free to be chosen by the trim solver. Values for the fixed states are input using the vector `xinit`. If the state is free, the corresponding element of `xinit` contains the starting value

used to determine trim solution. If the state is fixed, the corresponding element of `xinit` contains the fixed value.

The variables `ufree` and `uinit` contain information for fixed and free control variables. They are located on lines 122 and 123, respectively. All controls are free variables for this particular example. Lines 122 and 123 should be changed to:

```
ufree = [1,1,1,1];
uinit = [0,0,0,0];
```

Other starting values for the free state and control variables may need to be tried in order to converge to the trim solution.

Flight path angle γ is input in radians on line 127:

```
gam=0.349;
```

which corresponds to an angle of 20 deg.

Turn rate is input in rad/s on line 131:

```
trate =0.052;
```

This corresponds to a turn rate of 3 deg/s. This completes the necessary modifications to *F16_trm.m*. Save the modified file.

<u>Step 2.</u> Modify the file *F16_massprop.m* to account for an $x_{c.g.}$ location at 30% of the mean aerodynamic chord. Percentages must be normalized to values between 0 and 1. Change line 45 of the file to read:

```
xcg=0.3;
```

<u>Step 3.</u> Set the variable `lonflg` in the MATLAB workspace to either 1 or 0. Setting `lonflg` to 1 will provide the longitudinal system matrices, while setting the variable to 0 will provide lateral / directional matrices.

<u>Step 4.</u> Run the file *gen_F16_model.m* by typing the following on the MATLAB command line:

```
gen_F16_model;
```

This function computes the trim solution as well as the linear system matrices at the desired flight condition and longitudinal center of gravity location. After program execution, $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ are stored in the MATLAB workspace as variables `x0` and `u0`, respectively. The linear system matrices are stored in the MATLAB workspace as matrix variables `A, B, C,` and `D`.

The preceding procedure can be applied to the other supplied aircraft simulations with little modification (e.g., replace *F16* with *F106B*, *F14*, *F4*, etc. in the called functions).

NUMERICAL INTEGRATION

The equations of motion for a rigid aircraft are a set of coupled first-order nonlinear ordinary differential equations. Experimental aerodynamic and thrust data are used to model the applied aerodynamic and propulsion forces and moments for arbitrary states and controls. There is no closed form solution to such problems, so the equations must be solved using numerical integration. Techniques for solving this initial value problem for ordinary differential equations are employed to obtain approximate solutions at discrete points along the aircraft state trajectory.

The equations of motion have already been arranged into the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \tag{21}$$

where

$$\begin{aligned} \boldsymbol{x} &= \begin{bmatrix} V_t & \alpha & \beta & \Phi & \Theta & \Psi & p & q & r & x_E & y_E & h & P_a \end{bmatrix}^T \\ \boldsymbol{u} &= \begin{bmatrix} \delta_{th} & \delta_e & \delta_a & \delta_r \end{bmatrix}^T \end{aligned} \tag{22}$$

Control vector elements can be added or removed depending on the modeled aircraft.

A number of numerical integration techniques are available to approximate the solutions of the well-posed equations of motion. The choice of which technique to use depends on the desired accuracy of the solution and the computational effort to be expended. Three types of numerical integration techniques are employed within the software to approximate solutions to the equations of motion – Euler's method, Runge-Kutta method of orders two and four, and Adams-Bashforth method of order three. A flowchart illustrating the procedure used during the nonlinear simulation is given in figure 2.

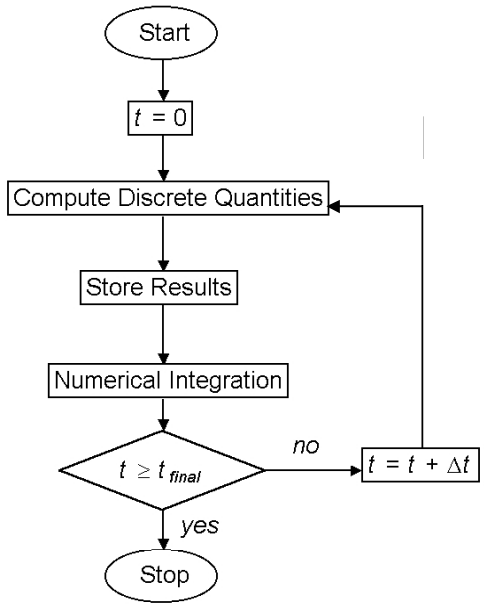

Figure 2. Flowchart for Nonlinear Simulation

Euler's Method

Euler's method is a one-step integration routine that approximates solutions to the initial value problem at discrete time steps Δt. Solutions to the state equations are approximated starting from initial states given at $t = 0$ with arbitrary control inputs provided for $t \geq 0$. The local truncation error for this method is $O(\Delta t)$. A single state derivative calculation is required for each time step, making this the least computationally intensive of the three approximation methods used in the simulation. However, it is also the least accurate. The algorithm is given in difference equation form as:

$$\boldsymbol{x}(t+\Delta t) = \boldsymbol{x}(t) + \Delta t\, \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) \tag{23}$$

Runge-Kutta Method of Order Two

The second-order Runge-Kutta method is also a one-step integration routine but provides for a more accurate solution approximation as compared to Euler's method. However, two state derivative calculations per time step are required making it comparatively slower than Euler's method. The local truncation error for this routine is $O(\Delta t^2)$. The difference equation is given by:

$$\boldsymbol{x}(t+\Delta t) = \boldsymbol{x}(t) + \Delta t\, \boldsymbol{f}\left(\left[\boldsymbol{x}(t) + \frac{\Delta t}{2}\boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right)\right], \boldsymbol{u}\left(t + \frac{\Delta t}{2}\right)\right) \tag{24}$$

Runge-Kutta Method of Order Four

The fourth-order Runge-Kutta method is a one-step routine with a local truncation error of $O(\Delta t^{4})$. Solution approximations are more accurate than either the Euler or second-order Runge-Kutta method, but the method runs one-fourth and one-half as fast, respectively. The difference equation for this scheme is given by:

$$\boldsymbol{x}(t+\Delta t)=\boldsymbol{x}(t)+\Delta t\left(\frac{k_1}{6}+\frac{k_2}{3}+\frac{k_3}{3}+\frac{k_4}{6}\right) \tag{25}$$

where

$$k_1=\boldsymbol{f}\left(\boldsymbol{x}(t),\boldsymbol{u}(t)\right)$$

$$k_2=\boldsymbol{f}\left(\boldsymbol{x}(t)+\frac{\Delta t}{2}k_1,\boldsymbol{u}\left(t+\frac{\Delta t}{2}\right)\right)$$

$$k_3=\boldsymbol{f}\left(\boldsymbol{x}(t)+\frac{\Delta t}{2}k_2,\boldsymbol{u}\left(t+\frac{\Delta t}{2}\right)\right)$$

$$k_4=\boldsymbol{f}\left(\boldsymbol{x}(t)+\Delta t k_3,\boldsymbol{u}\left(t+\Delta t\right)\right)$$

Adams-Bashforth Method of Order Three

The third-order Adams-Bashforth method is a multi-step routine that approximates the solution of the initial value problem using three previous step calculations. The approximation has a local truncation error of $O(\Delta t^{3})$. Second-order Runge-Kutta is used for the initial time steps before switching to the Adams-Bashforth integration method. The difference equation for the third-order Adams-Bashforth method is given by

$$\begin{aligned}\boldsymbol{x}\left(t+3\Delta t\right)=\boldsymbol{x}\left(t+2\Delta t\right)+\frac{\Delta t}{12}\Big[&23\boldsymbol{f}\left(\boldsymbol{x}(t+2\Delta t),\boldsymbol{u}\left(t+3\Delta t\right)\right)\\&-16\boldsymbol{f}\left(\boldsymbol{x}(t+\Delta t),\boldsymbol{u}\left(t+2\Delta t\right)\right)+5\boldsymbol{f}\left(\boldsymbol{x}(t),\boldsymbol{u}\left(t+\Delta t\right)\right)\Big]\end{aligned} \tag{26}$$

Detailed Numerical Integration Example

The state vector derivative $\dot{\boldsymbol{x}}$ is integrated numerically in the function *ACname.m*, where *ACname* is the simulated aircraft name (e.g., *F16.m*, *F106B.m, F4.m*, etc.) using arbitrary control inputs for $t \geq 0$ with initial states specified at $t = 0$. This produces a set of state time histories $\boldsymbol{x}(t)$ and output time histories $\boldsymbol{y}(t)$ for $t \geq 0$. Inputs to *ACname.m* are the control input time history $\boldsymbol{u}(t)$, time vector $\boldsymbol{t}$, initial state vector $\boldsymbol{x}$ at $t = 0$, and the vector of mass / inertia properties $\boldsymbol{c}$. The function *mksqw.m* is provided to allow creation of a $\boldsymbol{u}(t)$ comprised of arbitrary square waves with user-specified amplitudes, widths, delay, step size, and total length in seconds. However, any $\boldsymbol{u}(t)$ can be used with the

nonlinear simulation. A user-specified integration routine, such as the Adams-Bashforth method of order three (*ab3.m*), Runge-Kutta method of order two (*rk2.m*), or Runge-Kutta method of order four (*rk4.m*), is called within the *ACname.m* function.

A sample procedure for calculating state and output trajectories is provided below for the F-16 simulation. Procedures for the other aircraft simulations are similar.

Step 1. Create $\boldsymbol{u}(t)$ along with $\boldsymbol{t}$, which will be inputs to the nonlinear simulation function. The control vector must be stored in the MATLAB workspace as a matrix with the individual controls making up the columns and the deflection of each control over time comprising the rows. The time vector contains the discrete time points for the control vector. An example using the supplied function *mksqw.m* to create $\boldsymbol{u}(t)$ is shown below in figure 3. The elevator control surface δ_e will be deflected using a doublet with an amplitude of 1 degree that will be applied after a 3 second delay, while keeping the remaining control inputs (δ_{th}, δ_a, and δ_r) unperturbed from their trim values. Each pulse of the doublet will have a width of 2 seconds. The entire length of the maneuver will be 10 seconds with a sampling interval 0.025 seconds. The elevator perturbation time history is shown in figure 3.

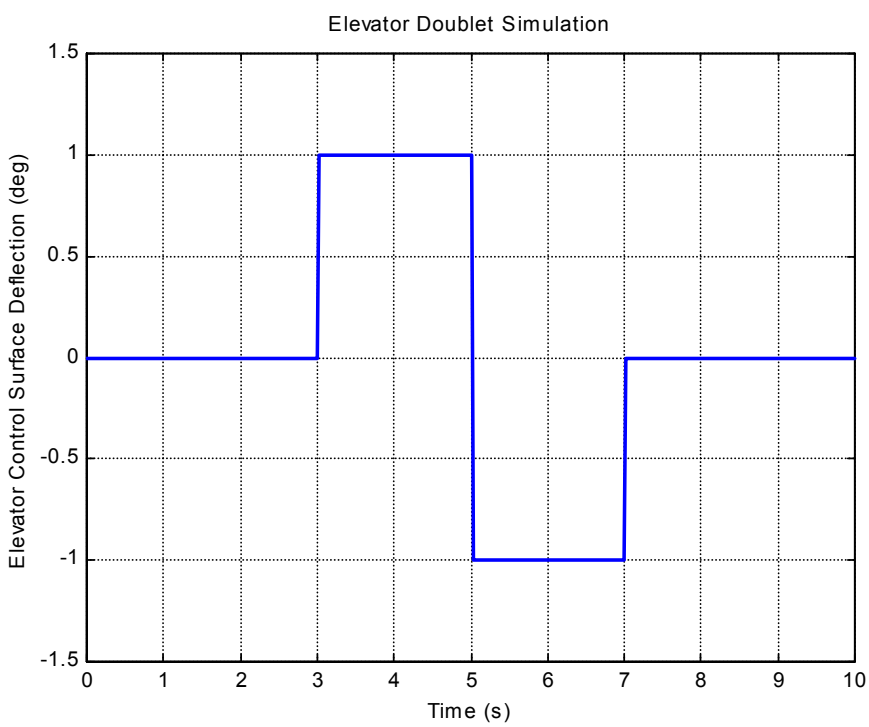

Figure 3. Elevator Perturbation

a. Type the following command in MATLAB:

```
[usw,t]=mksqw(1,2,[1 1], 3,0.025,10);
```

The first input to the function gives pulse amplitude, the second gives the width of a single pulse in seconds, the third gives the vector of

integer pulse widths, the fourth gives the delay in seconds, the fifth gives the sampling time, and the final input gives the length of the entire maneuver in seconds. The output of the function is stored in the MATLAB workspace under the variable names `usw` and `t`. The variable `usw` is a column vector composed of 401 elements describing the deflection history. The variable `t` is also a column vector composed of 401 elements recording the sample times for `usw`.

b. Create the control matrix `u` within the MATLAB workspace by adding `usw` to the second column of a 401 × 4 dimensional matrix, where the first column corresponds to values of δ_{th}, the second to δ_e, the third to δ_a, and the last column to δ_r. If the elements in the columns of `u` are initially set to the trim values of the controls, then adding `usw` to column 2 creates $\boldsymbol{u}(t)$ for an elevator doublet perturbation deflection, leaving other control inputs unperturbed from trim values during the maneuver.

Step 2. Define the initial state vector `x0` by creating a 13 × 1 vector composed of the user-defined state initial conditions. This vector contains the same state variables as the state vector $\boldsymbol{x}$ of the simulation model. To start from trim, use the trim state vector output from *ic_ftrm.m* or *gen_ACname_model.m*.

Step 3. Generate the mass property vector $\boldsymbol{c}$ by running the function *F16_massprop.m*:

```
c=F16_massprop;
```

The vector `c` will be available in the MATLAB workspace after execution of this function. Center of gravity position $x_{c.g.}$ can be adjusted within this file to simulate various longitudinal c.g. locations.

Step 4. Run the nonlinear F-16 simulation using the following command on the MATLAB command line:

```
[y,x]=F16(u,t,x0,c);
```

The output time history is stored in the MATLAB vector `y`, and the state output time history is stored in the vector `x`.

By default, the nonlinear simulation implements the Runge-Kutta integration method of order two. The user can modify the code to implement either the Runge-Kutta method of order four or the Adams-Bashforth method of order three. This is done by modifying the function *F16.m* following the procedure below:

To use Fourth-Order Runge-Kutta

Change line 102 to read:

```
%x=rk2('F16_deq',u,t,x0,c);
```

line 101 to read:

```
x=rk4('F16_deq',u,t,x0,c);
```

and line 103 to read:

```
%x=ab3('F16_deq',u,t,x0,c);
```

To use Third-Order Adams-Bashforth

Change line 102 to read:

```
%x=rk2('F16_deq',u,t,x0,c);
```

line 101 to read:

```
%x=rk4('F16_deq',u,t,x0,c);
```

and line 103 to read:

```
x=ab3('F16_deq',u,t,x0,c);
```

IV. AVDS / MATLAB INTERFACE

A MATLAB software interface to the software package called AVDS was developed to provide the capability to fly the MATLAB nonlinear aircraft simulations interactively in real time, using AVDS to read pilot inputs and display the aircraft response. Pilot inputs are measured by AVDS and used as inputs to the MATLAB simulations. Results of the simulation numerical integration are sent to AVDS and used to draw 3-D images of the aircraft in motion. Control deflections are also shown on the 3-D aircraft model displayed by AVDS. The following material provides instructions for AVDS / MATLAB software installation and usage.

EQUIPMENT / REQUIREMENTS

The nonlinear simulation software described here was developed and tested on IBM-compatible personal computers (PC), using MATLAB version 6.1, and AVDS version 1.2.14. System requirements for these commercially-available software tools are specified in their documentation. A graphics accelerator card capable of hardware acceleration using OpenGL™ produces smoother-running piloted simulations.

Control inputs are supplied to AVDS via a mouse, joystick, and / or keyboard. A joystick is recommended. A Microsoft® SideWinder® Precision 2 joystick has been used exclusively during software development and is recommended for use; however, any joystick can be used. This particular joystick contains a throttle control lever, hat switch, rudder control, and eight programmable buttons, which provide for somewhat realistic aircraft controls.

SOFTWARE INSTALLATION

AVDS is available from RasSimTech Ltd.© and is not part of the MATLAB aircraft simulation packages. The AVDS software allowing data communication between MATLAB and AVDS is available with version 1.2.14 of AVDS, available on the Internet at http://www.rassimtech.com.

Once the AVDS software is installed, the 3-D aircraft images provided for each simulated aircraft can be installed into AVDS. The 3-D image files are included in the aircraft simulation packages. The provided 3-D aircraft models are installed into AVDS using the following procedure:

Step 1. Copy the following aircraft 3-D files to the AVDS directory *\usr\local\AVDS\craft\:*

\usr\local\F-16\F16_hires.txt

\usr\local\F-16\F16_lores.txt

\usr\local\F-106B\F106B.txt

\usr\local\F-14\F14.txt

\usr\local\F-4\F4.txt

\usr\local\FASER\FASER_hires.txt

\usr\local\FASER\FASER_lores.txt

\usr\local\HL-20\HL20.txt

\usr\local\X-31\X31.txt

\usr\local\A-7\A7.txt

High and low resolution 3-D aircraft models are provided for the F-16 and FASER aircraft. The user has the choice to use either during interactive piloted simulations. Low-resolution models are provided for simulation on slower computers.

Step 2. Add the following lines to the file *craftcap.txt* located in the directory *\usr\local\AVDS\craft\:*

```
include "F16_lores.txt"
include "F16_hires.txt"
include "F106B.txt"
include "F14.txt"
include "F4.txt"
include "FASER_lores.txt"
include "FASER_hires.txt"
include "HL20.txt"
include "X31.txt"
include "A7.txt"
```

This file tells AVDS which 3-D aircraft models to make available for interactive simulation.

INTERACTIVE SIMULATION

The user can interactively provide control inputs through AVDS using a mouse, joystick, and/or keyboard. For control input device selection and configuration, consult section 3.4.2.9 of the AVDS User's Manual[2]. The following will discuss providing interactive control input with the Microsoft® SideWinder® Precision 2 joystick. Other joysticks should work as well, but only this joystick was used in the development and testing of the nonlinear simulations. Configuration of the initial aircraft states is also described.

Joystick Control Input

Control inputs for pitch, roll, yaw, and thrust during piloted interactive simulations can be provided by a Microsoft® SideWinder® Precision 2 joystick. For proper set-up of the joystick, set the scale factors for X, Y, Z, and R joystick axes to 2, 2, 1, and –2, respectively, within the AVDS joystick configuration screen (refer to section 3.4.2.9 of the AVDS User's Manual[2]). This normalizes AVDS inputs to the ranges 0 to1 for thrust control, and –1 to 1 for pitch, roll, and yaw. The normalized values are then converted to actual control surface deflections, measured in degrees, within the function *AVDS_Matlab_ACname.m*, where *ACname* is the simulated aircraft name (e.g., *AVDS_Matlab_F16.m*, *AVDS_Matlab_F106B.m*, *AVDS_Matlab_F4.m*, etc.). Other joystick settings, such as dead-zone, bias, trim step size, etc., can also be configured within the AVDS joystick configuration screen.

Aircraft pitch control is achieved with the fore / aft movement of the joystick, left / right movement of the joystick provides roll, and clockwise / counterclockwise twist of the joystick supplies yaw control. Throttle deflection is actuated by means of a small lever located at the base of the joystick. A hat switch located on the tip of the joystick is used to set elevator and aileron control surface trim deflections. Trim elevator deflection can be varied by pressing the top or bottom of the trim hat, while trim aileron can be set by pressing the left and right sides of the trim hat. The eight programmable buttons of the joystick have not been configured for use in AVDS / MATLAB interactive simulations.

Initial States for Interactive Simulation

AVDS allows specification of aircraft initial conditions for interactive simulations. The user can specify the aircraft starting altitude, velocity, Euler angles, and throttle position. These are set in the *Aircraft State Initialization* pop-up window, shown in figure 4, which is called within AVDS prior to simulation start-up. This is done as follows:

Step 1. Select **Initialize** from the top menu

Step 2. Select **Aircraft State**

Initial altitude, given in feet above sea level, is adjusted by moving the corresponding slider up or down. Initial body-axis velocity components u, v, and w (in ft/s), initial Euler angles Φ, Θ, and Ψ (in degrees), and initial throttle position δ_{th} (in percent) are adjusted similarly. Starting x_E and y_E positions cannot be adjusted with these sliders. The default starting positions for x_E and y_E are both set to 0 ft. Before hitting the **OK** button to complete initial state configuration, make certain that the **Apply Settings** button is depressed so that a check appears in the box. This allows the initial states set with the sliders to take effect in the interactive simulation. The button **Apply Init Lat/Long** has no effect within the AVDS / MATLAB simulation since starting x_E and y_E aircraft positions are set within the MATLAB simulation software.

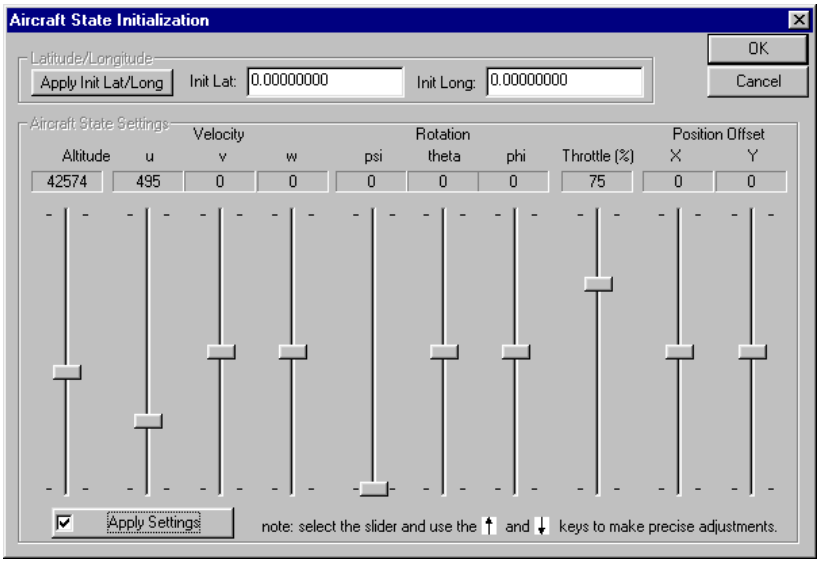

Figure 4. AVDS Aircraft State Initialization Window

AVDS / MATLAB Piloted Simulation Software

Piloted simulations are executed using the function *AVDS_Matlab_ACname.m* included in the aircraft simulation packages. This function establishes communication between AVDS and MATLAB and implements the integration routine to numerically solve the nonlinear equations of motion. Integration time steps during piloted simulations are dynamically updated within the *AVDS_Matlab_ACname.m* file, to prevent MATLAB from lagging behind AVDS simulation time. This allows synchronization of the real-time simulation of the aircraft in MATLAB with the 3-D display in AVDS. However, an upper limit (0.035 s) is imposed on the time steps to prevent loss of model fidelity. For less powerful computers (e.g., 500 MHz and slower), this can cause MATLAB to lag behind AVDS during simulation.

Control inputs at a given time step are sent from AVDS to MATLAB via a shared memory connection. Solutions of the nonlinear equations of motion are computed in MATLAB using the control inputs from AVDS. The solution is returned to AVDS by MATLAB, using the same data pathway. Control inputs and aircraft initial states, supplied by AVDS, are stored in the MATLAB workspace using the variable name `InputVector`. The output vector, stored in the variable called `OutputVector`, contains the translational and rotational states, angle of attack, sideslip angle, linear accelerations in *g*'s, airspeed, and Mach number, computed by the nonlinear simulation in MATLAB. Elements within `InputVector` and `OutputVector` are identified to AVDS via the file *ACname.sim.ini*, where *ACname* is the simulated aircraft name (e.g., *F16.sim.ini*, *F106B.sim.ini*, etc.). Refer to section 4.2 of the AVDS User's Manual[2] for modification instructions for this file. Typical elements of these vectors are shown below:

$$\texttt{InputVector} = \left[flag_{run} \;\; \delta_e \;\; \delta_a \;\; \delta_r \;\; \delta_{th} \;\; t_{AVDS} \;\; 0 \;\; 0 \;\; h \;\; \Phi \;\; \Theta \;\; \Psi \;\; V_t \;\; P_a \;\; \alpha \;\; \beta \right]^T$$

$$\texttt{OuputVector} = \left[x_E \;\; y_E \;\; h \;\; \Phi \;\; \Theta \;\; \Psi \;\; \alpha \;\; \beta \;\; g \;\; V_t \;\; M \;\; \left[\text{control surface deflections}\right] \right]^T$$

where $flag_{run}$ is a flag equal to one when AVDS is running, and zero otherwise, and t_{AVDS} is the simulation time in the AVDS program.

Figure 5 shows a flowchart illustrating the interactive simulation.

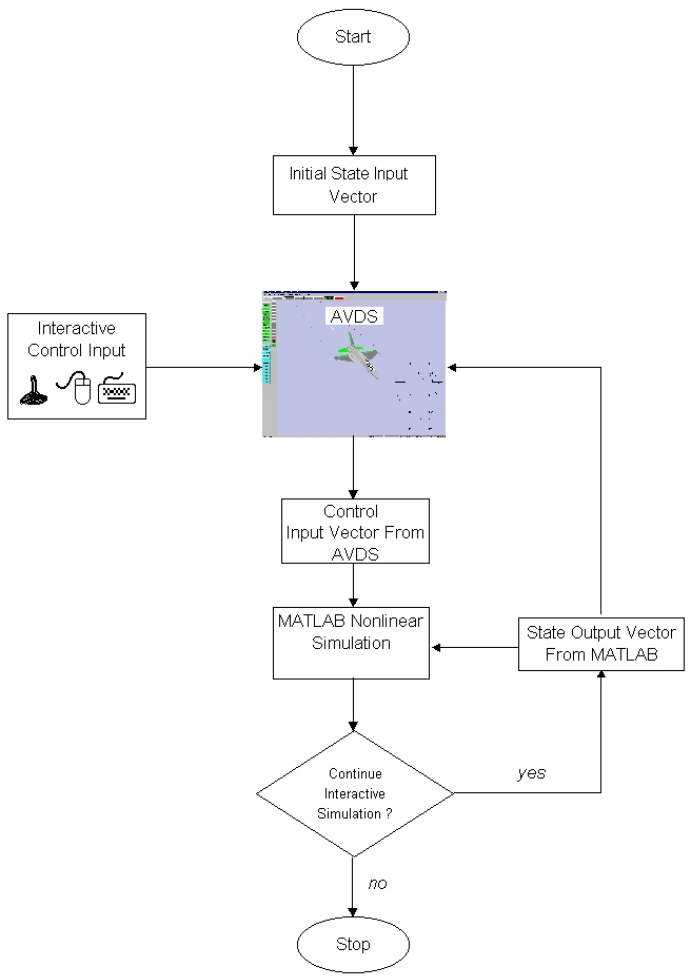

Figure 5. Interactive Simulation Flowchart

The translational and rotational aircraft states calculated within MATLAB are used by AVDS to display the 3-D aircraft model along the calculated trajectory. Quantities important to the pilot (α, β, M, V_t, h, Θ, Ψ, and a_N (g's)) are displayed to the user in the AVDS Head-Up Display (HUD). A screen shot of the AVDS screen during interactive simulation of the F-16 is shown in figure 6. Flight path of the aircraft is shown with trailing ribbons generated on each wing tip and the aircraft center tail section.

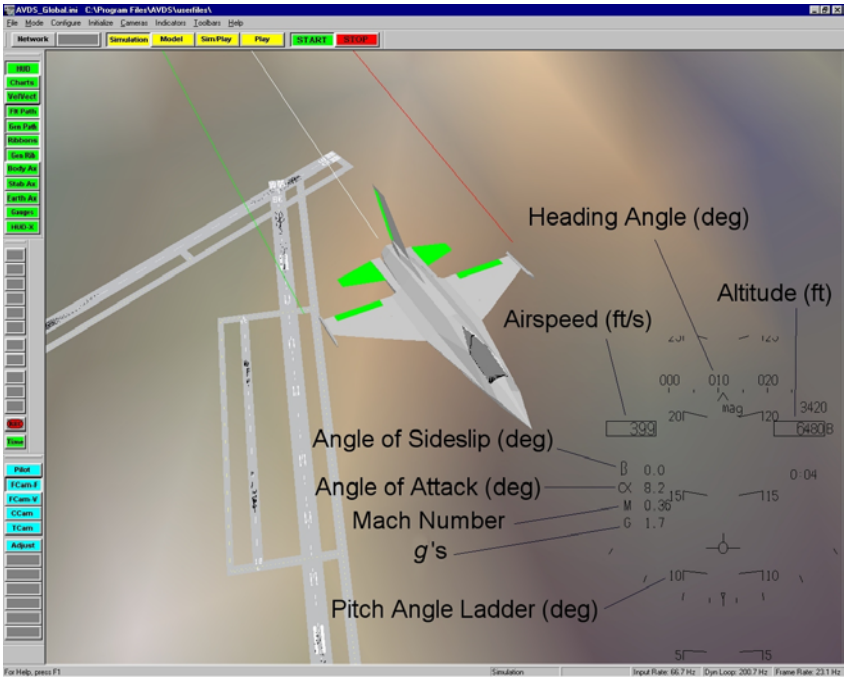

Figure 6. Screen Shot of the F-16 AVDS / MATLAB Interactive Simulation

Stick Gain Modification

Stick gains for elevator, aileron, and rudder are applied within the function *AVDS_Matlab_ACname.m* to allow for easier open-loop flying. All stick gains for the simulated aircraft are set to 0.5 by default. The procedure used to modify the stick gains for the F-16 simulation are given below (other simulation modifications are similar):

To change elevator stick gain

Modify line 123 within *AVDS_Matlab_F16.m* to read:

```
gain_el = xxx;
```

where `xxx` is the user-defined numerical value for the elevator stick gain.

To change aileron stick gain

Modify line 124 within *AVDS_Matlab_F16.m* to read:

```
gain_ail = xxx;
```

where `xxx` is the user-defined numerical value for the aileron stick gain.

To change rudder stick gain

Modify line 125 within *AVDS_Matlab_F16.m* to read:

```
gain_rdr = xxx;
```

where `xxx` is the user-defined numerical value for the rudder stick gain.

Control Deflection Exaggeration Factor

Deflections of aircraft control surfaces, shown in green on the AVDS aircraft 3-D models, can be visually exaggerated for easier viewing. By default, all deflections shown in AVDS are exaggerated by a factor of two. The following provides an example of modifying the exaggeration factor for the F-16. Other simulation modifications are similar.

To adjust control surface deflection exaggeration factor

Modify line 130 within *AVDS_Matlab_F16.m* to read:

```
cdfac = xxx;
```

where `xxx` is the user-defined numerical value for the deflection exaggeration factor. If `cdfac` is set to 2, the displayed control surface deflections will be 2 times the actual control surface deflections being used in the MATLAB nonlinear aircraft simulation.

Integration Routine

Integration routines used to solve the equations of motion for the AVDS / MATLAB interactive simulations are implemented within *AVDS_Matlab_ACname.m* as opposed to *ab3.m*, *rk2.m*, and *rk4.m* for the batch nonlinear simulations. The *AVDS_Matlab_ACname.m* function can implement either the Adams-Bashforth integration method of order three, the Runge-Kutta method of order two, or Euler's method. The following gives an example for implementing particular integration routines for the F-16 simulation (*AVDS_Matlab_F16.m)*. Modifications for other simulations are similar.

To implement Adams-Bashforth method of order three

Remove the `%` character from the beginning of lines 250-266 and add the `%` character to the beginning of lines 270-274 and lines 278-280. MATLAB ignores lines of code beginning with the `%` character during function execution.

To implement Runge-Kutta method of order two

Remove the `%` character from the beginning of lines 270-274 and add the `%` character to the beginning of lines 250-266 and lines 278-280.

To implement Euler's method

Remove the `%` character from the beginning of lines 278-280 and add the `%` character to the beginning of lines 250-266 and lines 270-274.

Starting the AVDS / MATLAB Simulation

The following provides an example of running the F-16 piloted simulation. A similar procedure is used for the other provided simulations.

To start the piloted F-16 simulation

<u>Step 1.</u> Start AVDS

Step 2. Load the simulation initialization file F16.sim.ini

a. Select File from the top menu

b. Select Simulation Init, then Open

c. Choose the file F16.sim.ini located in the F-16 directory of the simulation software package

Step 3. Load the 3-D aircraft image F-16 (Lo-Res) or F-16 (Hi-Res)

a. Select Initialize from the top menu

b. Select Aircraft Image

c. Select either F-16 (Lo-Res) or F-16 (Hi-Res)

Step 4. In AVDS, switch to simulation mode and start the simulation

a. Press SIMULATION button on main toolbar

b. Press START button on main toolbar

Step 5. Run AVDS_Matlab_F16 in MATLAB

a. Set the current directory to the F-16 directory of the simulation software package

b. Type AVDS_Matlab_F16 on the MATLAB command line followed by ENTER

To end the simulation

Press the **STOP** button located on main toolbar in AVDS

V. SPECIFIC AIRCRAFT SIMULATION DETAILS

Details specific to each aircraft simulation package are given below. Unique simulation characteristics such as aerodynamic and engine models, aircraft controls, and valid ranges for the states and controls are given. Angle of attack, sideslip angle, and control surface deflection inputs to the aerodynamic models are in degrees, unless specified otherwise. Aircraft angular velocities are in radians per second, unless specified otherwise. For a list of the files included in each package, refer to tables A-1 through A-8 in the Appendix.

F-16 SIMULATION

The General Dynamics F-16 is a single-seat, multi-role fighter with a blended wing / body and a cropped delta wing planform with leading edge sweep of 40º. The wing is fitted with leading edge flaps and trailing edge flaperons (flaps / ailerons). Tail surfaces are swept and cantilevered. The horizontal stabilator is composed of two all-moving tail plane halves, while the vertical tail is fitted with a trailing edge rudder. Thrust is provided by one General Electric F110-GE-100 or Pratt & Whitney F100-PW-220 afterburning turbofan engine mounted in the rear fuselage.

The aircraft was modeled with controls for δ_{th}, δ_e, δ_a, and δ_r. Aerodynamic force and moment data were derived from low-speed static and dynamic forced oscillation wind tunnel tests conducted on a 16% scale model of the F-16 flown out of ground effect, with landing gear retracted, and no external stores[3,9]. Static aerodynamic data are in tabular form as a function of angle of attack and sideslip over the ranges $-10° \le \alpha \le 45°$ and $-30° \le \beta \le 30°$. Dynamic data is provided in tabular form at zero sideslip angle over the angle of attack range $-10° \le \alpha \le 45°$. Each non-dimensional aerodynamic force and moment coefficient is built up from component functions. Aerodynamic coefficients are referenced to a center of gravity location at $0.35\bar{c}$. Corrections to the flight center of gravity position are made in the coefficient build-up equations.

The aerodynamic database was simplified slightly by dropping second order dependencies (e.g., dependence of longitudinal aerodynamic forces on sideslip angle). Throttle deflection is limited to the range $0 \le \delta_{th} \le 1$, elevator deflection is limited to $-25° \le \delta_e \le 25°$, aileron deflection is limited to $-21.5° \le \delta_a \le 21.5°$, and rudder deflection is limited to $-30° \le \delta_r \le 30°$.

Engine thrust data is in tabular form as a function of power level, altitude, and Mach number over the ranges 0 ft $\le h \le$ 50,000 ft and $0 \le M \le 1$, for idle, military, and maximum power settings. Engine power level dynamics and gyroscopic effects[9] are included for this aircraft simulation, in the manner described previously. Figure 7 shows a three-view of the F-16.

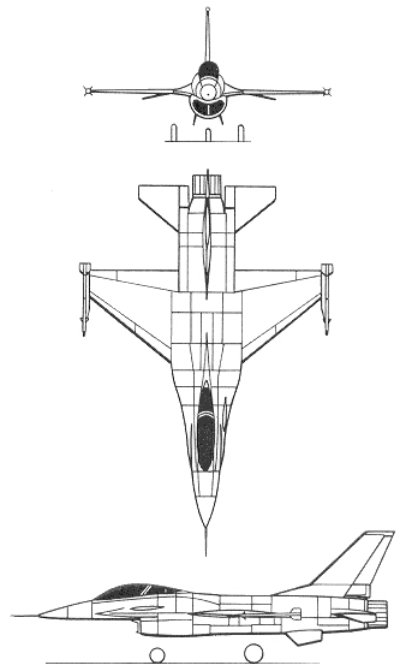

Figure 7. Three-View of the F-16

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the F-16 vary nonlinearly with flow angles (α, β), aircraft angular velocities (p, q, r), and control surface deflections (δ_e, δ_a, δ_r). Moment coefficients C_n and C_m include a correction for the center of gravity position. The coefficients are computed as follows:

$$C_X = C_{X_0}(\alpha, \delta_e) + \left(\frac{q\bar{c}}{2V_t}\right) C_{X_q}(\alpha)$$

$$C_Y = -0.02\beta + 0.021\left(\frac{\delta_a}{20}\right) + 0.086\left(\frac{\delta_r}{30}\right) + \left(\frac{b}{2V_t}\right)\left(C_{Y_p}(\alpha)\, p + C_{Y_r}(\alpha)\, r\right)$$

$$C_Z = C_{Z_0}(\alpha)\left[1 - \left(\frac{\beta\pi}{180}\right)^2\right] - 0.19\left(\frac{\delta_e}{25}\right) + \left(\frac{q\bar{c}}{2V_t}\right) C_{Z_q}(\alpha)$$

$$C_l = C_{l_0}(\alpha,\beta) + \Delta C_{l,\delta_a=20^\circ}\left(\frac{\delta_a}{20}\right) + \Delta C_{l,\delta_r=30^\circ}\left(\frac{\delta_r}{30}\right) + \left(\frac{b}{2V_t}\right)\left(C_{l_p}(\alpha)\,p + C_{l_r}(\alpha)\,r\right)$$

$$C_m = C_{m_0}(\alpha,\delta_e) + \left(\frac{q\bar{c}}{2V_t}\right)C_{m_q}(\alpha) + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = C_{n_0}(\alpha,\beta) + \Delta C_{n,\delta_a=20^\circ}\left(\frac{\delta_a}{20}\right) + \Delta C_{n,\delta_r=30^\circ}\left(\frac{\delta_r}{30}\right) + \left(\frac{b}{2V_t}\right)\left(C_{n_p}(\alpha)\,p + C_{n_r}(\alpha)\,r\right)$$

$$-\left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

where

$$\Delta C_{l,\delta_a=20^\circ} = C_{l,\delta_a=20^\circ}(\alpha,\beta) - C_{l_0}(\alpha,\beta)$$

$$\Delta C_{l,\delta_r=30^\circ} = C_{l,\delta_r=30^\circ}(\alpha,\beta) - C_{l_0}(\alpha,\beta)$$

$$\Delta C_{n,\delta_a=20^\circ} = C_{n,\delta_a=20^\circ}(\alpha,\beta) - C_{n_0}(\alpha,\beta)$$

$$\Delta C_{n,\delta_r=30^\circ} = C_{n,\delta_r=30^\circ}(\alpha,\beta) - C_{n_0}(\alpha,\beta)$$

Data tables for the component functions of the non-dimensional coefficients are defined in the file *F16_aero_setup.m*. The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *F16_aero.m,* using linear interpolation in the functions *cxo.m*, *czo.m*, *cmo.m*, *clo.m*, *cno.m*, *dlda.m*, *dldr.m*, *dnda.m*, *dndr.m*, and *dampder.m*.

Engine Model and Database

The F-16 is powered by a single afterburning turbofan jet engine, which is modeled taking into account throttle gearing and engine power level lag. The engine angular momentum h_{eng} is assumed to act along the aircraft X body-axis with a fixed value of 160 slug-ft^2/s. Throttle gearing is implemented in the file *tgear.m*, which outputs commanded engine power in percent of full power [0,100] using throttle setting input [0,1]. Commanded power P_c is a function of δ_{th}, given by:

$$P_c(\delta_{th}) = \begin{cases} 64.94\delta_{th} & \text{if } \delta_{th} \le 0.77 \\ 217.38\delta_{th} - 117.38 & \text{if } \delta_{th} > 0.77 \end{cases}$$

Engine power level dynamic response is modeled using a first order lag with the time constant computed by the function *rtau.m*. The module *pdot.m* calculates the rate of

change of power level with time using the difference between P_c and P_a. The actual power level derivative $\dot{P}_a$ is given by:

$$\dot{P}_a = \frac{1}{\tau_{eng}}\left(P_c - P_a\right)$$

where

$$P_c = \begin{cases} P_c & \text{if } P_c \geq 50.0 \text{ and } P_a \geq 50.0 \\ 60 & \text{if } P_c \geq 50.0 \text{ and } P_a < 50.0 \\ 40 & \text{if } P_c < 50.0 \text{ and } P_a \geq 50.0 \\ P_c & \text{if } P_c < 50.0 \text{ and } P_a < 50.0 \end{cases}$$

$$\frac{1}{\tau_{eng}} = \begin{cases} 5.0 & \text{if } P_c \geq 50.0 \text{ and } P_a \geq 50.0 \\ \frac{1}{\tau_{eng}}^{*} & \text{if } P_c \geq 50.0 \text{ and } P_a < 50.0 \\ 5.0 & \text{if } P_c < 50.0 \text{ and } P_a \geq 50.0 \\ \frac{1}{\tau_{eng}}^{*} & \text{if } P_c < 50.0 \text{ and } P_a < 50.0 \end{cases}$$

$$\frac{1}{\tau_{eng}}^{*} = \begin{cases} 1.0 & \text{if } \left(P_c - P_a\right) \leq 25.0 \\ 0.1 & \text{if } \left(P_c - P_a\right) \geq 50.0 \\ 1.9 - 0.036\left(P_c - P_a\right) & \text{if } 25.0 < \left(P_c - P_a\right) < 50.0 \end{cases}$$

Engine thrust is modeled as a function of h, M, and P_a. Data tables for idle, military, and maximum power as a function of h and M are defined in the file *F16_engine_setup.m*. Linear interpolation and thrust calculation is performed in the module *F16_engine.m*. Thrust is computed from:

$$T = \begin{cases} T_{idle} + \left(T_{mil} - T_{idle}\right)\left(\dfrac{P_a}{50}\right) & \text{if } P_a < 50 \\ T_{mil} + \left(T_{max} - T_{mil}\right)\left(\dfrac{P_a - 50}{50}\right) & \text{if } P_a \geq 50 \end{cases}$$

Mass Properties

The mass properties of the F-16 are given in the file *F16_massprop.m*. These properties include the weight of the aircraft, $x_{c.g.}$, I_X, I_Y, I_Z, and I_{XZ}. This module also calculates the constants $c_1, c_2, \ldots, c_9$ for use in the body-axis moment equations. The longitudinal center of gravity location $x_{c.g.}$ is given in fraction of mean aerodynamic chord, and can be adjusted within this module to account for different longitudinal center of gravity locations. Table 1 lists the mass properties used in the F-16 simulation.

Table 1. Mass Properties of the Simulated F-16

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	20,500	9,496	55,814	63,100	982

F-106B SIMULATION

The Convair F-106B is a two-seat all-weather interceptor incorporating a delta wing planform with a single swept vertical tail surface. Elevon control surfaces, which perform both elevator and aileron control functions, are located at the trailing edges of the wing. The rudder control surface is located at the trailing edge of the vertical tail. Thrust is provided by a single Pratt & Whitney J75-P-17 afterburning jet engine mounted in the rear fuselage.

The aircraft simulation has controls for δ_{th}, δ_e, δ_a, and δ_r. Low-speed static and dynamic forced oscillation wind tunnel tests on a 15% scale model of the aircraft[7] provided aerodynamic force and moment data. Tests were conducted in the NASA Langley 30- by 60-foot wind tunnel, with the model flown out of ground effect, with landing gear retracted, and no external stores[10,11]. Static and dynamic aerodynamic data are in tabular form as a function of angle of attack over the range $-5° \leq \alpha \leq 50°$. Each non-dimensional aerodynamic force and moment coefficient is built up from component functions. Aerodynamic coefficients are referenced to a center of gravity location at $0.275\bar{c}$, and corrected to the flight center of gravity position in the coefficient build-up equations.

Throttle deflection is limited to the range $0 \leq \delta_{th} \leq 1$, elevator deflection is limited to $-28° \leq \delta_e \leq 28°$, aileron deflection is limited to $-7° \leq \delta_a \leq 7°$, and rudder deflection is limited to $-25° \leq \delta_r \leq 25°$.

The F-106B engine has maximum thrust values of 24,500 lb and 17,200 lb with and without afterburner, respectively. A scaled version of the F-16 engine model was used for this simulation. Idle, military, and maximum power settings were scaled using the maximum thrust for the F-106B engine. Throttle gearing, gyroscopic effects, and engine power level dynamics were carried over from the F-16 engine model. Figure 8 shows a three-view of the F-106B aircraft.

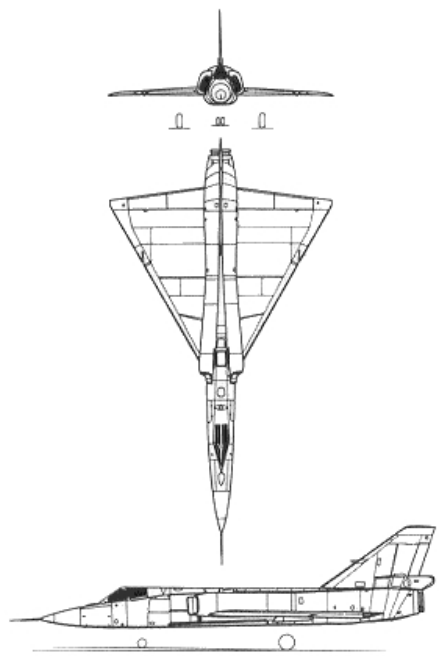

Figure 8. Three-View of the F-106B

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the F-106B vary nonlinearly with flow angles (α, β), aircraft angular velocities (p, q, r), and control surface deflections (δ_e, δ_a, δ_r). Moment coefficients C_n and C_m include a correction for the center of gravity position. The coefficients are computed as follows:

$$C_X = C_{L_0}(\alpha,\delta_e)\,sin\left(\frac{\alpha\pi}{180}\right) - C_{D_0}(\alpha,\delta_e)\,cos\left(\frac{\alpha\pi}{180}\right)$$

$$C_Y = C_{Y_\beta}(\alpha)\,\beta + \left(\frac{b}{2V_t}\right)\left(C_{Y_p}(\alpha)\,p + C_{Y_r}(\alpha)\,r\right)$$

$$C_Z = -C_{L_0}(\alpha,\delta_e)\,cos\left(\frac{\alpha\pi}{180}\right) - C_{D_0}(\alpha,\delta_e)\,sin\left(\frac{\alpha\pi}{180}\right) - \left(\frac{q\bar{c}}{2V_t}\right)C_{N_q}(\alpha)$$

$$C_l = C_{l_\beta}(\alpha)\beta + \Delta C_{l,\delta_a=7^\circ}\left(\frac{\delta_a}{7}\right) + \Delta C_{l,\delta_r=25^\circ}\left(\frac{\delta_r}{25}\right) + \left(\frac{b}{2V_t}\right)\left(C_{l_p}(\alpha)p + C_{l_r}(\alpha)r\right)$$

$$C_m = C_{m_0}(\alpha,\delta_e) + \left(\frac{q\bar{c}}{2V_t}\right)C_{m_q}(\alpha) + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = C_{n_\beta}(\alpha)\beta + \Delta C_{n,\delta_a=7^\circ}\left(\frac{\delta_a}{7}\right) + \Delta C_{n,\delta_r=25^\circ}\left(\frac{\delta_r}{25}\right) + \left(\frac{b}{2V_t}\right)\left(C_{n_p}(\alpha)p + C_{n_r}(\alpha)r\right)$$

$$-\left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

where

$$\Delta C_{l,\delta_a=7^\circ} = C_{l,\delta_a=7^\circ}(\alpha) - C_{l_0}(\alpha)$$

$$\Delta C_{l,\delta_r=25^\circ} = C_{l,\delta_r=25^\circ}(\alpha) - C_{l_0}(\alpha)$$

$$\Delta C_{n,\delta_a=7^\circ} = C_{n,\delta_a=7^\circ}(\alpha) - C_{n_0}(\alpha)$$

$$\Delta C_{n,\delta_r=25^\circ} = C_{n,\delta_r=25^\circ}(\alpha) - C_{n_0}(\alpha)$$

Data tables for the component functions of the non-dimensional coefficients are defined in the file *F106B_aero_setup.m*. The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *F106B_aero.m*, using linear interpolation in the functions *cdo.m*, *clo.m*, *cmo.m*, *dldb.m*, *dlda.m*, *dldr.m*, *dndb.m*, *dnda.m*, *dndr.m*, *dydb.m*, *dydr.m*, and *dampder.m*.

Engine Model and Database

Thrust scaling for the F-106B engine is applied in the function *F106B_engine_setup.m*. Linear interpolation and thrust calculation is performed in the module *F106B_engine.m*. Functions *tgear.m*, *pdot.m*, and *rtau.m* are unaltered from the F-16 engine simulation.

Mass Properties

Aircraft mass properties are given in the module *F106B_massprop.m*. Table 2 lists the mass properties used in the F-106B simulation.

Table 2. Mass Properties of the Simulated F-106B

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	29,776	18,634	177,858	191,236	5,539

F-14 SIMULATION

The Grumman F-14 is a two-seat, carrier-based, multi-role fighter with a variable geometry wing and leading edge sweep varying between 20º and 68º. Small canard surfaces extend from the leading edges of the fixed portion of the wing to provide extra longitudinal stability. Spoilers are located on the upper surfaces of the wing and provide aileron control during low speed flight. Trailing edge flaps are located at the rear portion of the wing and extend over most of the span, while slats are fitted to the wing leading edges. Two all-moving horizontal tail surfaces are located at the rear of the fuselage and provide aileron and elevator control using differential and symmetric deflections, respectively. Twin vertical tails, also located at the rear of the fuselage, are fitted with trailing edge rudder control surfaces. Speed brakes deflect symmetrically from a location at the rear fuselage between the bases of the vertical tails. Two Pratt & Whitney TF30-P-412A afterburning turbofan engines, located at the rear of the fuselage, provide thrust for the aircraft.

The aircraft was modeled in the unswept wing configuration, with controls for δ_{th}, δ_e, δ_a, δ_r, δ_{sp}, and δ_{sb}. Aircraft aerodynamic data was taken with the vehicle flown out of ground effect, with landing gear retracted, and no external stores[12,13]. Static aerodynamic data are in tabular form as a function of angle of attack, sideslip angle, and Mach number over the ranges $-5° \leq \alpha \leq 50°$, $-20° \leq \beta \leq 20°$, and $0 \leq M \leq 1$. The aerodynamic effects of angular rates are also in tabular form over the same angle of attack range. Each non-dimensional aerodynamic force and moment coefficient is built up from component functions. Aerodynamic coefficients are referenced to a center of gravity location at $0.16\bar{c}$, and corrected to the flight center of gravity position in the coefficient build-up equations.

Throttle deflection is limited to the range $0 \leq \delta_{th} \leq 1$, elevator deflection is limited to $-30° \leq \delta_e \leq 12°$, aileron deflection is limited to $-21° \leq \delta_a \leq 21°$, rudder deflection is limited to $-30° \leq \delta_r \leq 30°$, wing spoiler deflection is limited to $0° \leq \delta_{sp} \leq 55°$, and speed brake deflection is limited to $0° \leq \delta_{sb} \leq 60°$.

Tabular data for idle, military, and maximum thrust were used to model the thrust produced by the F-14 engine. Engine dynamics and throttle gearing were carried over from the F-16 engine model; however, engine angular momentum was neglected. Figure 9 shows a three-view of the F-14.

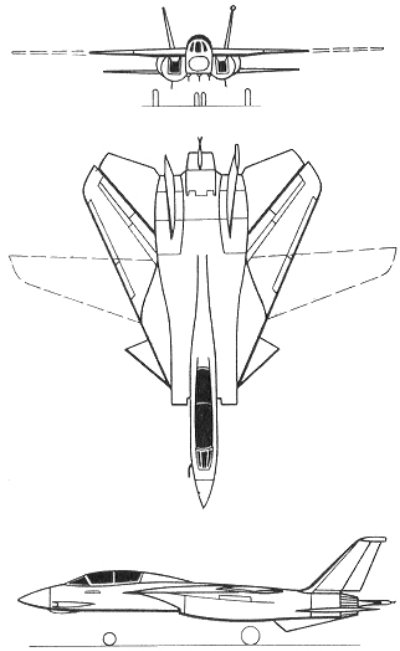

Figure 9. Three-View of the F-14

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the F-14 model vary nonlinearly with relative airflow quantities (α, β, M), aircraft angular velocities (p, q, r), and control surface deflections (δ_e, δ_a, δ_r, δ_{sp}, δ_{sb}). Moment coefficients C_n and C_m include a correction for the center of gravity position. The coefficients are computed as follows:

$$C_X = C_{X_0}(\alpha,\beta) + \Delta C_{X,\delta_e} + \Delta C_{X,\delta_{sb}=60^\circ}\left(\frac{\delta_{sb}}{60}\right) + \Delta C_{X,\delta_{sp}=55^\circ}\left(\frac{\delta_{sp}}{55}\right)$$

$$C_Y = C_{Y_0}(\alpha,\beta) + C_{Y_{\delta_a}}(\alpha,\beta)\delta_a + C_{Y_{\delta_r}}(\alpha,\beta)\delta_r + C_{Y_{\delta_{sp}}}(\alpha)\delta_{sp}$$

$$+\left(\frac{b}{2V_t}\right)\left(C_{Y_p}(\alpha)p + C_{Y_r}(\alpha)r\right)$$

$$C_Z = C_{Z_0}(\alpha,\beta) + \Delta C_{Z,\delta_e} + \Delta C_{Z,\delta_{sb}=60^\circ}\left(\frac{\delta_{sb}}{60}\right) + \Delta C_{Z,\delta_{sp}=55^\circ}\left(\frac{\delta_{sp}}{55}\right) + \left(\frac{q\bar{c}}{2V_t}\right)C_{Z_q}(\alpha)$$

$$C_l = C_{l_0}(\alpha,\beta) + C_{l_{\delta_a}}(\alpha,\beta)\delta_a + C_{l_{\delta_r}}(\alpha,\beta)\delta_r + C_{l_{\delta_{sp}}}(\alpha,\beta)\eta_{sp,l}(M)\delta_{sp}$$

$$+\left(\frac{b}{2V_t}\right)\left(C_{l_p}(\alpha)p + C_{l_r}(\alpha)r\right)$$

$$C_m = C_{m_0}(\alpha,\beta) + \Delta C_{m,\delta_e} + \Delta C_{m,\delta_{sb}=60^\circ}\left(\frac{\delta_{sb}}{60}\right) + \Delta C_{m,\delta_{sp}=55^\circ}\left(\frac{\delta_{sp}}{55}\right)\eta_{sp,m}(M) + \left(\frac{q\bar{c}}{2V_t}\right)C_{m_q}(\alpha)$$

$$+\left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = C_{n_0}(\alpha,\beta) + C_{n_{\delta_a}}(\alpha,\beta)\delta_a + C_{n_{\delta_r}}(\alpha,\beta)\delta_r + C_{n_{\delta_{sp}}}(\alpha)\delta_{sp} + \left(\frac{b}{2V_t}\right)\left(C_{n_p}(\alpha)p + C_{n_r}(\alpha)r\right)$$

$$-\left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

where

$$\Delta C_{X,\delta_e} = C_{X,\delta_e}(\alpha,\beta,\delta_e) - C_{X_0}(\alpha,\beta)$$

$$\Delta C_{X,\delta_{sb}=60^\circ} = C_{X,\delta_{sb}=60^\circ}(\alpha) - C_{X_0}(\alpha)$$

$$\Delta C_{X,\delta_{sp}=55^\circ} = C_{X,\delta_{sp}=55^\circ}(\alpha) - C_{X_0}(\alpha)$$

$$\Delta C_{Z,\delta_e} = C_{Z,\delta_e}(\alpha,\beta,\delta_e) - C_{Z_0}(\alpha,\beta)$$

$$\Delta C_{Z,\delta_{sb}=60^\circ} = C_{Z,\delta_{sb}=60^\circ}(\alpha) - C_{Z_0}(\alpha)$$

$$\Delta C_{Z,\delta_{sp}=55^\circ} = C_{Z,\delta_{sp}=55^\circ}(\alpha) - C_{Z_0}(\alpha)$$

$$\Delta C_{m,\delta_e} = C_{m,\delta_e}(\alpha,\beta,\delta_e) - C_{m_0}(\alpha,\beta)$$

$$\Delta C_{m,\delta_{sb}=60^\circ} = C_{m,\delta_{sb}=60^\circ}(\alpha) - C_{m_0}(\alpha)$$

$$\Delta C_{m,\delta_{sp}=55^\circ} = C_{m,\delta_{sp}=55^\circ}(\alpha) - C_{m_0}(\alpha)$$

$\eta_{sp,l}(M) =$ wing spoiler rolling moment efficiency factor given as a function of Mach number

$\eta_{sp,m}(M) =$ wing spoiler pitching moment efficiency factor given as a function of Mach number

Data tables for the component functions of the non-dimensional coefficients are defined in the file *F14_aero_setup.m*. The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *F14_aero.m*, using linear interpolation in the functions *etamsp.m*, *etalsp.m*, *dndr.m*, *dnda.m*, *dlda.m*, *dldr.m*, *dczsp.m*, *dczsb.m*, *dczds.m*, *dcxsp.m*, *dcxsb.m*, *dcxds.m*, *dcmsp.m*, *dcmsb.m*, *dcmds.m*, *czo.m*, *cyo.m*, *cydsp.m*, *cydr.m*, *cyda.m*, *cxo.m*, *cno.m*, *cndsp.m*, *cmo.m*, *clo.m*, *cldsp.m*, and *dampder.m*.

Engine Model and Database

Data tables for the F-14 engine are defined in the function *F14_engine_setup.m*. Linear interpolation and thrust calculation is performed in the module *F14_engine.m*. Functions *tgear.m*, *pdot.m*, and *rtau.m* are unaltered from the F-16 engine simulation.

Mass Properties

Mass properties for the aircraft are given in the module *F14_massprop.m*. Table 3 lists the mass properties used in the F-14 simulation.

Table 3. Mass Properties of the Simulated F-14

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	48,669	58,500	227,000	276,000	-2,820

F-4 SIMULATION

The McDonnell Douglas F-4 is a two-seat all-weather fighter / bomber. The aircraft is fitted with a low-mounted swept wing with wingtip dihedral. The tail section consists of an all-moving horizontal stabilator placed in a cathedral configuration and a single vertical tail. The trailing edge of the main wing houses control surfaces for the aileron and flaps. The trailing edge of the vertical tail accommodates a rudder control surface. Thrust is provided by two General Electric GE-J19-17 afterburning jet engines mounted at the rear of the fuselage.

The simulation has controls for δ_{th}, δ_e, δ_a, and δ_r. Aerodynamic data were acquired with the model flown out of ground effect, with landing gear retracted, and no external stores[14]. Static and dynamic aerodynamics are modeled in terms of polynomial functions, which limit control surface effects to first order dependencies, angle of attack static dependencies to second order, and sideslip angle static dependencies to third order. Polynomial fits for each non-dimensional aerodynamic force and moment coefficient are valid over an angle of attack range of $-15^{\circ} \le \alpha \le 55^{\circ}$. Aerodynamic coefficients are referenced to a center of gravity location at $0.289\bar{c}$, and corrected to the flight center of gravity position in the coefficient build-up equations.

Throttle deflection is limited to the range $0 \leq \delta_{th} \leq 1$, elevator deflection is limited to $-21° \leq \delta_e \leq 7°$, aileron deflection is limited to $-15.5° \leq \delta_a \leq 15.5°$, and rudder deflection is limited to $-30° \leq \delta_r \leq 30°$. A scaled version of the F-16 engine model is used with this simulation. Thrust for idle, military, and maximum power settings were scaled according to the maximum thrust values of the F-4 power plant. Throttle gearing and engine power level dynamics were carried over from the F-16 engine model, neglecting engine angular momentum effects. Figure 10 shows a three-view of the F-4 aircraft.

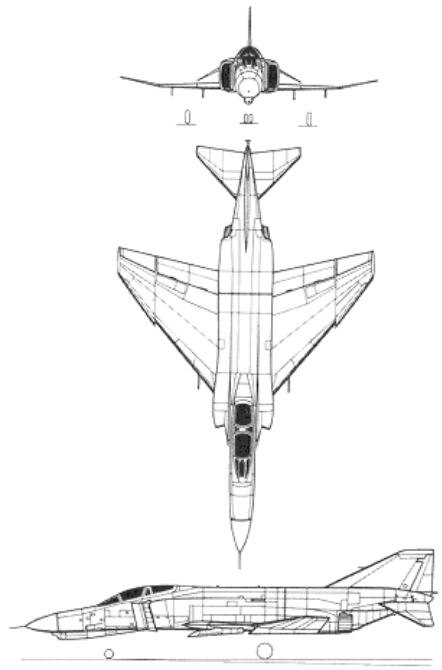

Figure 10. Three-View of the F-4

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the F-4 model vary nonlinearly with flow angles (α, β), aircraft angular velocities (p, q, r), and control surface deflections (δ_e, δ_a, δ_r). Moment coefficients C_n and C_m include a correction for the center of gravity position. The coefficients are computed as follows:

For $\alpha \leq 15°$

$$C_X = -0.0434 + 2.39\times10^{-3}\alpha + 2.53\times10^{-5}\beta^2 - 1.07\times10^{-6}\alpha\beta^2 + 9.5\times10^{-4}\delta_e$$

$$-8.5\times10^{-7}\delta_e\beta^2 + \left(\frac{180q\bar{c}}{\pi 2V_t}\right)\left(8.73\times10^{-3} + 0.001\alpha - 1.75\times10^{-4}\alpha^2\right)$$

$$C_Y = -0.012\beta + 1.55\times10^{-3}\delta_r - 8\times10^{-6}\delta_r\alpha$$

$$+\left(\frac{180b}{\pi 2V_t}\right)\left(2.25\times10^{-3}p + 0.0117r - 3.67\times10^{-4}r\alpha \ + 1.75\times10^{-4}r\delta_e\right)$$

$$C_Z = -0.131 - 0.0538\alpha - 4.76\times10^{-3}\delta_e - 3.3\times10^{-5}\delta_e\alpha - 7.5\times10^{-5}\delta_a{}^2$$

$$+\left(\frac{180q\bar{c}}{\pi 2V_t}\right)(-0.111 + 5.17\times10^{-3}\alpha - 1.1\times10^{-3}\alpha^2)$$

$$C_l = -5.98\times10^{-4}\beta - 2.83\times10^{-4}\alpha\beta + 1.51\times10^{-5}\alpha^2\beta$$

$$-\delta_a\,(6.1\times10^{-4} + 2.5\times10^{-5}\alpha - 2.6\times10^{-6}\alpha^2)$$

$$-\delta_r\,(-2.3\times10^{-4} + 4.5\times10^{-6}\alpha)$$

$$+\left(\frac{180b}{\pi 2V_t}\right)(-4.12\times10^{-3}p - 5.24\times10^{-4}p\alpha + 4.36\times10^{-5}p\alpha^2$$

$$+4.36\times10^{-4}r + 1.05\times10^{-4}r\alpha + 5.24\times10^{-5}r\delta_e)$$

$$C_m = -6.61\times10^{-3} - 2.67\times10^{-3}\alpha - 6.48\times10^{-5}\beta^2$$

$$-2.65\times10^{-6}\alpha\beta^2 - 6.54\times10^{-3}\delta_e - 8.49\times10^{-5}\delta_e\alpha$$

$$+3.74\times10^{-6}\delta_e\beta^2 - 3.5\times10^{-5}\delta_a{}^2$$

$$+\left(\frac{180q\bar{c}}{\pi 2V_t}\right)(-0.0473 - 1.57\times10^{-3}\alpha) + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = 2.28\times10^{-3}\beta + 1.79\times10^{-6}\beta^3 + 1.4\times10^{-5}\delta_a$$

$$+7.0\times10^{-6}\delta_a\alpha - 9.0\times10^{-4}\delta_r + 4.0\times10^{-6}\delta_r\alpha$$

$$+\left(\frac{180b}{\pi 2V_t}\right)\Big(-6.63\times10^{-5}p - 1.92\times10^{-5}p\alpha + 5.06\times10^{-6}p\alpha^2$$

$$-6.06\times10^{-3}r - 8.73\times10^{-5}r\delta_e + 8.7\times10^{-6}r\delta_e\alpha\Big)$$

$$-\left(\frac{\overline{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

<u>For 15° < $\alpha \leq$ 30°</u>

$$C_X = 0.141 - 0.0154\alpha + 2.96\times10^{-4}\alpha^2 - 3.72\times10^{-4}\beta^2$$

$$+4.14\times10^{-5}\alpha\beta^2 - 9.12\times10^{-7}\alpha^2\beta^2 + 1.82\times10^{-3}\delta_{\mathrm{e}}$$

$$-7.3\times10^{-5}\delta_{\mathrm{e}}\alpha + \left(\frac{180q\overline{c}}{\pi 2V_t}\right)(-0.0602 + 2.04\times10^{-3}\alpha)$$

$$C_Y = -2.08\times10^{-2}\beta + 6.07\times10^{-4}\alpha\beta + 2.37\times10^{-6}\beta^3$$

$$-3.64\times10^{-7}\alpha\beta^3 + 2.3\times10^{-3}\delta_r - 5.9\times10^{-5}\delta_r\alpha$$

$$+\left(\frac{180b}{\pi 2V_t}\right)(-1.62\times10^{-3}p + 3.32\times10^{-4}p\alpha$$

$$+0.0311r - 1.4\times10^{-3}r\alpha + 1.75\times10^{-4}r\delta_e)$$

$$C_Z = -0.608 - 0.022\alpha - 6.77\times10^{-3}\delta_{\mathrm{e}} + 9.7\times10^{-5}\delta_{\mathrm{e}}\alpha - 7.5\times10^{-5}\delta_a{}^2$$

$$+\left(\frac{180q\overline{c}}{\pi 2V_t}\right)(1.136 - 0.1418\alpha + 3.11\times10^{-3}\alpha^2)$$

$$
\begin{aligned}
C_l = \; & -1.29\times10^{-2}\beta + 1.04\times10^{-3}\alpha\beta - 2.02\times10^{-5}\alpha^2\beta \\
& + 1.36\times10^{-5}\beta^3 - 1.13\times10^{-6}\alpha\beta^3 + 2.01\times10^{-8}\alpha^2\beta^3 \\
& - \delta_a(7.74\times10^{-4} - 1.9\times10^{-5}\alpha) - \delta_r\,(-2.0\times10^{-4} + 5.0\times10^{-6}\alpha) \\
& + \left(\frac{180b}{\pi 2V_t}\right)(2.78\times10^{-3}p - 2.79\times10^{-4}p\alpha - 6.81\times10^{-3}r \\
& + 6.46\times10^{-4}r\alpha + 5.24\times10^{-5}r\delta_e)
\end{aligned}
$$

$$
\begin{aligned}
C_m = \; & 0.0549 - 6.08\times10^{-3}\alpha - 1.69\times10^{-4}\beta^2 \; + \; 5.64\times10^{-7}\alpha\beta^2 \\
& - \; 8.14\times10^{-3}\delta_e + 1.1\times10^{-4}\delta_e\alpha - 3.5\times10^{-5}{\delta_a}^2 \\
& + \left(\frac{180q\bar{c}}{\pi 2V_t}\right)(-0.0951 \; + \; 1.4\times10^{-3}\alpha) + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z
\end{aligned}
$$

$$
\begin{aligned}
C_n = \; & 1.02\times10^{-2}\beta - 5.12\times10^{-4}\alpha\beta - 5.27\times10^{-6}\beta^3 \\
& + 3.79\times10^{-7}\alpha\beta^3 + 9.1\times10^{-5}\delta_a + 3.0\times10^{-6}\delta_a\alpha \\
& - 1.37\times10^{-3}\delta_r + 3.8\times10^{-5}\delta_r\alpha \\
& + \left(\frac{180b}{\pi 2V_t}\right)(0.0236p - 2.5\times10^{-3}p\alpha + 6.25\times10^{-5}p\alpha^2 \\
& + 6.2\times10^{-4}r - 4.89\times10^{-4}r\alpha - 8.73\times10^{-5}r\delta_e + 8.7\times10^{-6}r\delta_e\alpha) \\
& - \left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y
\end{aligned}
$$

For $\alpha > 30°$

$$C_X = -0.0326 - 2.16\times10^{-3}\alpha + 4.89\times10^{-5}\alpha^2 - 1.24\times10^{-4}\beta^2$$
$$+ 1.076\times10^{-5}\alpha\beta^2 - 1.54\times10^{-7}\alpha^2\beta^2 + 7.5\times10^{-4}\delta_e$$
$$-3.9\times10^{-5}\delta_e\alpha + \left(\frac{180q\bar{c}}{\pi 2V_t}\right)(-0.026 + 8.73\times10^{-4}\alpha)$$

$$C_Y = -2.095\times10^{-3}\beta - 6.36\times10^{-5}\alpha\beta - 2.15\times10^{-5}\beta^3$$
$$+5.42\times10^{-7}\alpha\beta^3 + 1.4\times10^{-3}\delta_r - 2.6\times10^{-5}\delta_r\alpha$$
$$+\left(\frac{180b}{\pi 2V_t}\right)(0.196p - 9.27\times10^{-3}p\alpha + 1.01\times10^{-4}p\alpha^2$$
$$+0.0326r - 2.55\times10^{-3}r\alpha + 3.26\times10^{-5}r\alpha^2 + 1.75\times10^{-4}r\delta_e)$$

$$C_Z = -0.891 - 0.01146\alpha + 6.2\times10^{-3}\delta_e - 5.4\times10^{-3}\delta_e\alpha$$
$$+ 6.2\times10^{-6}\delta_e\alpha^2 - 7.5\times10^{-5}\delta_a{}^2$$
$$+ \left(\frac{180q\bar{c}}{\pi 2V_t}\right)(0.589 - 0.0494\alpha + 6.11\times10^{-4}\alpha^2)$$

$$C_l = 1.18\times10^{-2}\beta - 5.29\times10^{-4}\alpha\beta + 4.88\times10^{-6}\alpha^2\beta$$
$$-2.2\times10^{-5}\beta^3 + 9.05\times10^{-7}\alpha\beta^3 - 9.08\times10^{-9}\alpha^2\beta^3$$
$$-5.0\times10^{-5}\delta_a - \delta_r\,(-9.0\times10^{-5} + 1.8\times10^{-6}\alpha)$$
$$+\left(\frac{180b}{\pi 2V_t}\right)(-0.0428p + 1.82\times10^{-3}p\alpha - 1.94\times10^{-5}p\alpha^2$$
$$+0.073r - 3.02\times10^{-3}r\alpha + 3.14\times10^{-5}r\alpha^2 + 5.24\times10^{-5}r\delta_e)$$

$$C_m = 7.3\times10^{-3} - 5.5\times10^{-3}\alpha - 7.93\times10^{-3}\delta_e + 8.23\times10^{-5}\delta_e\alpha$$
$$-3.5\times10^{-5}\delta_a{}^2 + \left(\frac{180q\bar{c}}{\pi 2V_t}\right)(0.16 - 0.0101\alpha + 1.05\times10^{-4}\alpha^2)$$
$$+\left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$
\begin{aligned}
C_n = & -9.23\times10^{-3}\beta + 1.52\times10^{-4}\alpha\beta + 1.62\times10^{-5}\beta^3 \\
& -3.46\times10^{-7}\alpha\beta^3 + 1.5\times10^{-4}\delta_a - 6.8\times10^{-4}\delta_r \\
& +1.2\times10^{-5}\delta_r\alpha - \delta_e\,(1.67\times10^{-4}\beta - 5.56\times10^{-6}\alpha\beta \\
& -3.81\times10^{-7}\beta^3 + 1.27\times10^{-8}\alpha\beta^3) \\
& +\left(\frac{180b}{\pi 2V_t}\right)(0.0385p - 1.73\times10^{-3}p\alpha + 1.92\times10^{-5}p\alpha^2 \\
& -0.0202r + 3.67\times10^{-4}r\alpha + 2.58\times10^{-3}r\delta_e \\
& -1.31\times10^{-4}r\delta_e\alpha + 1.69\times10^{-6}r\delta_e\alpha^2) - \left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y
\end{aligned}
$$

The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *F4_aero.m*.

Engine Model and Database

Scaled thrust values for the F-4 power plant are applied in the function *F4_engine_setup.m*. Linear interpolation and thrust calculation are performed in the module *F4_engine.m*. Functions *tgear.m*, *pdot.m*, and *rtau.m* are unaltered from the F-16 engine simulation.

Mass Properties

Aircraft mass properties are given in the module *F4_massprop.m*. Table 4 lists the mass properties used in the F-4 simulation.

Table 4. Mass Properties of the Simulated F-4

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	38,924	24,970	122,190	139,800	1,175

FASER SIMULATION

A radio-controlled aircraft, called the NASA Langley FASER, has the form of a high-wing general aviation aircraft. The aircraft was modeled with controls for δ_{th}, δ_e, δ_a, and δ_r. Non-dimensional static and dynamic aerodynamic data were estimated[15] using airfoil data and approximations based on aircraft geometry, for the aircraft out of ground effect. The aerodynamic data is in tabular form as a function of angle of attack, Mach

number, lift coefficient, and thrust over the ranges $-10° \leq \alpha \leq 30°$, $0.05 \leq M \leq 0.3$, $-1 \leq C_L \leq 1.6$, and 0 lb $\leq T \leq$ 10 lb. Each non-dimensional aerodynamic force and moment coefficient is built up from component functions, which are provided in the stability-axis frame. Aerodynamic coefficients are referenced to a center of gravity location at $0.273\,\bar{c}$, and corrected to the flight center of gravity position in the coefficient build-up equations.

Throttle deflection is limited to the range $0 \leq \delta_{th} \leq 1$, elevator deflection is limited to $-25° \leq \delta_e \leq 25°$, aileron deflection is limited to $-25° \leq \delta_a \leq 25°$, and rudder deflection is limited to $-30° \leq \delta_r \leq 30°$. FASER is fitted with a single electric motor that drives a propeller to generate thrust. Figure 11 shows a three-view of the FASER.

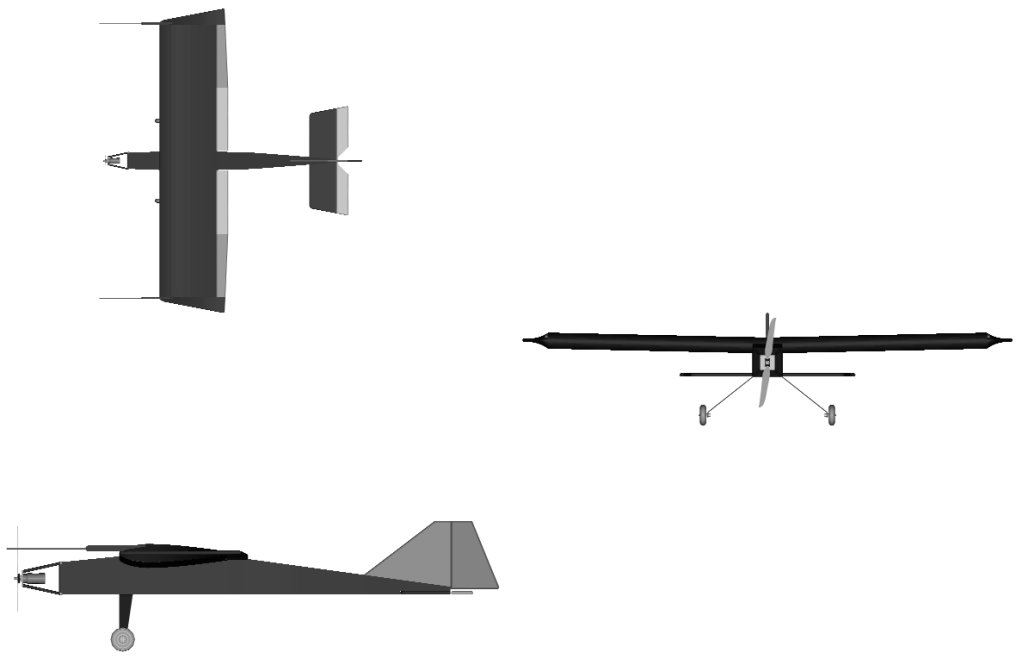

Figure 11. Three-View of the FASER

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the FASER vary nonlinearly with relative airflow quantities (α, β, M), lift coefficient, (C_L), thrust (T), aircraft angular velocities (p, q, r), and control surface deflections (δ_e, δ_a, δ_r). Moment coefficients C_n and C_m include a correction for the center of gravity position. The coefficients are computed as follows:

$$C_X = C_L \, sin\left(\frac{\alpha\pi}{180}\right) - C_D \, cos\left(\frac{\alpha\pi}{180}\right)$$

$$C_Y = C_{Y_\beta}(M)\left(\frac{\beta\pi}{180}\right) + C_{Y_{\delta_r}}(M)\left(\frac{\delta_r\pi}{180}\right)$$

$$+\left(\frac{b}{2V_t}\right)\left(C_{Y_{p_S}}(\alpha,M)\,p_S + C_{Y_{r_S}}(\alpha,M)\,r_S\right)$$

$$C_Z = -C_L\,cos\left(\frac{\alpha\pi}{180}\right) - C_D\,sin\left(\frac{\alpha\pi}{180}\right)$$

$$C_l = C_{l,S}\,cos\left(\frac{\alpha\pi}{180}\right) - C_{n,S}\,sin\left(\frac{\alpha\pi}{180}\right)$$

$$C_m = C_{m_0}(\alpha,M) + \left(\frac{q\bar{c}}{2V_t}\right)C_{m_q}(\alpha,M) + C_{m_{\delta_e}}(\alpha,M)\left(\frac{\delta_e\pi}{180}\right)$$

$$+\Delta C_{m,_T} + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = C_{l,S}\,sin\left(\frac{\alpha\pi}{180}\right) + C_{n,S}\,cos\left(\frac{\alpha\pi}{180}\right) - \left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

where

$$C_L = C_{L_0}(\alpha,M) + C_{L_{\delta_e}}(\alpha,M)\left(\frac{\delta_e\pi}{180}\right) + \Delta C_{L,T} + \left(\frac{q\bar{c}}{2V_t}\right)C_{L_q}(\alpha,M)$$

$$C_D = C_{D_0}(\alpha,M) + \Delta C_{D,T}$$

$$C_{l,S} = C_{l_\beta}(M,C_L)\left(\frac{\beta\pi}{180}\right) + C_{l_{\delta_a}}(M)\left(\frac{\delta_a\pi}{180}\right) + C_{l_{\delta_r}}(\alpha,M)\left(\frac{\delta_r\pi}{180}\right)$$

$$+\left(\frac{b}{2V_t}\right)\left(C_{l_{p_S}}(M)\,p_S + C_{l_{r_S}}(\alpha,M,C_L)\,r_S\right)$$

$$C_{n,S} = C_{n_\beta}(\alpha,M)\left(\frac{\beta\pi}{180}\right) + C_{n_{\delta_a}}(M,C_L)\left(\frac{\delta_a\pi}{180}\right) + C_{n_{\delta_r}}(\alpha,M)\left(\frac{\delta_r\pi}{180}\right)$$

$$+\left(\frac{b}{2V_t}\right)\left(C_{n_{p_S}}(\alpha,M,C_L)\,p_S + C_{n_{r_S}}(\alpha,M,C_L)\,r_S\right)$$

$$\Delta C_{L,T} = C_{L,T}(\alpha,M,T) - C_{L_0}(\alpha,M)$$

$$\Delta C_{D,T} = C_{D,T}(\alpha,M,T) - C_{D_0}(\alpha,M)$$

$$\Delta C_{m,T} = C_{m,T}(\alpha,M,T) - C_{m_0}(\alpha,M)$$

$$p_S = p\,cos\left(\frac{\alpha\pi}{180}\right) + r\,sin\left(\frac{\alpha\pi}{180}\right)$$

$$r_S = -p\,sin\left(\frac{\alpha\pi}{180}\right) + r\,cos\left(\frac{\alpha\pi}{180}\right)$$

Data tables for the component functions of the non-dimensional coefficients are defined in the file *FASER_aero_setup.m*. The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *FASER_aero.m*, using linear interpolation in one-, two-, and three-dimensional interpolation functions *int1.m*, *int2.m*, and *int3.m*.

Engine Model and Database

Propulsion for the FASER is modeled neglecting engine angular momentum. Throttle gearing is implemented in the file *tgear.m*, which outputs commanded engine power in percent of full power [0,100] using throttle setting input [0,1]. Commanded power P_c is a linear function of δ_{th}, given by:

$$P_c\left(\delta_{th}\right) = 100\delta_{th}$$

Engine power level dynamic response is modeled using a first order lag with time constant computed by the function *rtau.m*. The module *pdot.m* calculates the rate of change of power level with time using the difference between P_c and P_a. The actual power level derivative $\dot{P}_a$ is given by:

$$\dot{P}_a = \frac{1}{\tau_{eng}}\left(P_c - P_a\right)$$

where

$$\frac{1}{\tau_{eng}} = \begin{cases} 1.0 & \text{if } \left(P_c - P_a\right) \le 25.0 \\ 0.2 & \text{if } \left(P_c - P_a\right) \ge 50.0 \\ 1.0 - \left(\frac{0.8}{25}\right)\left[\left(P_c - P_a\right) - 25.0\right] & \text{if } 25.0 < \left(P_c - P_a\right) < 50.0 \end{cases}$$

Engine thrust is modeled as a linear function of P_a. Thrust calculation is performed in the module *FASER_engine.m* using the equation:

$$T = 0.1P_a$$

Mass Properties

Mass properties of the aircraft are given in the module *FASER_massprop.m*. Table 5 lists the mass properties used in the FASER simulation.

Table 5. Mass Properties of the Simulated FASER

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	16.5	1.324	1.946	0.627	0.317

HL-20 SIMULATION

The NASA HL-20 is a manned lifting-body spacecraft. It is designed to be placed into earth orbit by an expendable launch vehicle and re-enter the atmosphere for an unpowered approach and landing. Aircraft control surfaces consist of an all moving vertical tail (rudder), two upper and lower body flaps, and two trailing edge wing flaps. Elevator control is provided by symmetric deflection of the wing flaps, aileron control by differential deflection of the wing flaps, positive flap control by symmetric deflection of the lower body flaps, negative flap control by symmetric deflection of the upper body flaps, and differential flap control by differential deflection of the upper and lower body flaps.

The HL-20 was modeled with controls for δ_e, δ_a, δ_r, δ_{pf}, δ_{nf}, δ_{df}, and δ_{lg}. Low-speed (M <0.6) wind tunnel data for a scale model of the aircraft flown out of ground effect with landing gear retracted and no external stores were taken in the Langley 30- by 60-foot tunnel and Calspan 8-foot transonic tunnel[16,17]. Landing gear effects were based on scaled Space Shuttle data, and dynamic derivative data were found using estimation techniques. Static and dynamic aerodynamic data are modeled with polynomial fits as a function of angle of attack and sideslip angle over the ranges $-10^{\circ} \le \alpha \le 30^{\circ}$ and $-10° \le \beta \le 10°$. Each non-dimensional aerodynamic force and moment coefficient is built up from component functions. Aerodynamic coefficients are referenced to a center of gravity location at $0.54\bar{c}$, and corrected to the flight center of gravity position in the coefficient build-up equations.

Elevator deflection is limited to $-30° \le \delta_e \le 30°$, aileron deflection is limited to $-30^{\circ} \le \delta_a \le 30^{\circ}$, rudder deflection is limited to $-30° \le \delta_r \le 30°$, positive flap deflection is limited to $0° \le \delta_{pf} \le 60°$, negative flap deflection is limited to $-60° \le \delta_{nf} \le 0°$, differential flap deflection is limited to $-60° \le \delta_{df} \le 60°$, and landing gear deflection is limited to $0° \le \delta_{lg} \le 98°$. An engine model is not included in this simulation since the HL-20 uses no propulsion during atmospheric flight after re-entry. Figure 12 shows a three-view of the HL-20.

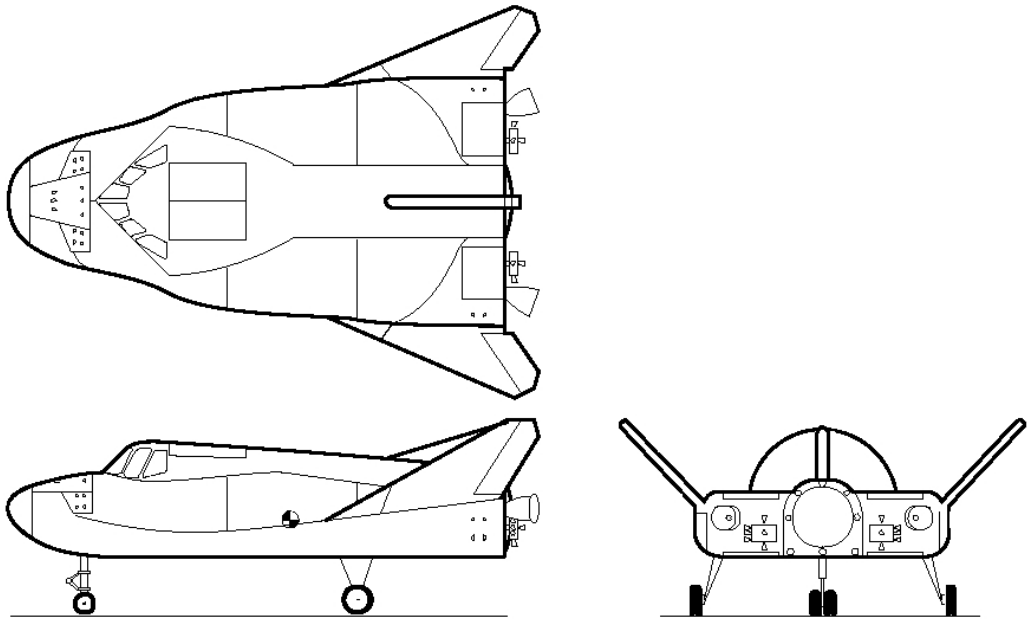

Figure 12. Three-View of the HL-20

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the HL-20 model vary nonlinearly with flow angles (α, β), aircraft angular velocities (p, q, r), and control deflections (δ_e, δ_a, δ_r, δ_{pf}, δ_{nf}, δ_{df}, δ_{lg}). Moment coefficients C_n and C_m include a correction for the center of gravity position. The coefficients are computed as follows:

$$C_X = C_{X_0} + C_{X_{\delta_e}} \delta_e + C_{X_{\delta_a}} |\delta_a| + C_{X_{\delta_r}} |\delta_r| + C_{X_{\delta_{pf}}} \delta_{pf} + C_{X_{\delta_{nf}}} \delta_{nf} + C_{X_{\delta_{df}}} |\delta_{df}|$$

$$+ \Delta C_{X,\delta_{lg}}$$

$$C_Y = -0.01242\beta + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r + C_{Y_{\delta_{df}}} \delta_{df}$$

$$C_Z = C_{Z_0} + C_{Z_{\delta_e}} \delta_e + C_{Z_{\delta_{pf}}} \delta_{pf} + C_{Z_{\delta_{nf}}} \delta_{nf} + \Delta C_{Z,\delta_{lg}}$$

$$C_l = -0.00787\beta + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r + C_{l_{\delta_{df}}}\delta_{df} + \left(\frac{b}{2V_t}\right)\left(C_{l_p}p + C_{l_r}r\right)$$

$$C_m = C_{m_0} + C_{m_{\delta_e}}\delta_e + C_{m_{\delta_{pf}}}\delta_{pf} + C_{m_{\delta_{nf}}}\delta_{nf} + \Delta C_{m,\delta_{lg}} + \left(\frac{\bar{c}q}{2V_t}\right)C_{m_q} + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = C_{n_0}(\alpha,\beta) + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r + C_{n_{\delta_{df}}}\delta_{df} + \left(\frac{b}{2V_t}\right)\left(C_{n_p}p + C_{n_r}r\right) -$$

$$\left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

where

$$C_{X_0} = -(7.362\times10^{-2} - 2.56\times10^{-4}\alpha - 2.208\times10^{-4}\alpha^2 - 2.262\times10^{-6}\alpha^3 + 2.966\times10^{-7}\alpha^4$$

$$-3.64\times10^{-9}\alpha^5 + 9.388\times10^{-12}\alpha^6 - 5.299\times10^{-4}|\beta| - 4.709\times10^{-4}\beta^2 + 8.572\times10^{-5}|\beta^3|$$

$$-4.199\times10^{-6}\beta^4 + 1.295\times10^{-4}\alpha|\beta|)$$

$$C_{X_{\delta_e}} = -(-1.854\times10^{-4} + 2.83\times10^{-6}\alpha - 6.966\times10^{-7}\alpha^2 + 1.323\times10^{-7}\alpha^3 - 2.758\times10^{-9}\alpha^4)$$

$$C_{X_{\delta_a}} = -(9.776\times10^{-4} - 2.703\times10^{-5}\alpha - 8.303\times10^{-6}\alpha^2 + 6.645\times10^{-7}\alpha^3 - 1.273\times10^{-8}\alpha^4)$$

$$C_{X_{\delta_r}} = -(5.812\times10^{-4} + 1.41\times10^{-5}\alpha - 2.585\times10^{-6}\alpha^2 + 3.051\times10^{-7}\alpha^3 - 8.161\times10^{-9}\alpha^4)$$

$$C_{Y_{\delta_a}} = 3.357\times10^{-3} - 1.661\times10^{-5}\alpha - 3.28\times10^{-6}\alpha^2 + 5.526\times10^{-8}\alpha^3 - 3.269\times10^{-10}\alpha^4$$

$$C_{Y_{\delta_r}} = 1.855\times10^{-3} + 1.128\times10^{-5}\alpha + 6.069\times10^{-6}\alpha^2 - 1.78\times10^{-7}\alpha^3 - 1.886\times10^{-12}\alpha^4$$

$$C_{Y_{\delta_{df}}} = 2.672\times10^{-5} - 3.849\times10^{-5}\alpha + 4.564\times10^{-7}\alpha^2 + 1.798\times10^{-8}\alpha^3 - 4.099\times10^{-10}\alpha^4$$

$$C_{Z_0} = -(-9.025\times10^{-2} + 4.07\times10^{-2}\alpha + 3.094\times10^{-5}\alpha^2 + 1.564\times10^{-5}\alpha^3 - 1.386\times10^{-6}\alpha^4$$

$$+2.545\times10^{-8}\alpha^5 - 1.189\times10^{-10}\alpha^6 + 2.564\times10^{-3}|\beta| + 8.501\times10^{-4}\beta^2$$

$$-1.156\times10^{-4}|\beta^3| + 3.416\times10^{-6}\beta^4 - 4.862\times10^{-4}\alpha|\beta|)$$

$$C_{Z_{\delta_e}} = -(5.140\times10^{-3} + 3.683\times10^{-5}\alpha - 6.092\times10^{-6}\alpha^2 + 2.818\times10^{-9}\alpha^3 - 2.459\times10^{-9}\alpha^4)$$

$$C_{X_{\delta_{pf}}} = -(1.31\times10^{-4} + 1.565\times10^{-6}\alpha^2 - 1.542\times10^{-9}\alpha^4)$$

$$C_{X_{\delta_{nf}}} = -(-4.415\times10^{-4} - 4.056\times10^{-6}\alpha - 4.657\times10^{-7}\alpha^2)$$

$$C_{X_{\delta_{df}}} = -(-6.043\times10^{-4} - 1.858\times10^{-5}\alpha + 8\times10^{-7}\alpha^2 - 4.848\times10^{-8}\alpha^3 + 1.36\times10^{-9}\alpha^4)$$

$$\Delta C_{X,\delta_{lg}} = C_{X,\delta_{lg}}\left(\alpha,\delta_{lg}\right) - C_{X_0}\left(\alpha\right)$$

$$C_{Z_{\delta_{pf}}} = -(3.7779\times10^{-3} - 7.017\times10^{-7}\alpha^2 + 1.4\times10^{-10}\alpha^4)$$

$$C_{Z_{\delta_{nf}}} = -(3.711\times10^{-3} - 3.547\times10^{-5}\alpha - 2.706\times10^{-6}\alpha^2 + 2.938\times10^{-7}\alpha^3$$

$$-5.552\times10^{-9}\alpha^4)$$

$$\Delta C_{Z,\delta_{lg}} = C_{Z,\delta_{lg}}\left(\alpha,\delta_{lg}\right) - C_{Z_0}\left(\alpha\right)$$

$$C_{l_{\delta_a}} = 2.538\times10^{-3} + 1.963\times10^{-5}\alpha - 3.725\times10^{-6}\alpha^2 + 3.539\times10^{-8}\alpha^3 - 1.778\times10^{-10}\alpha^4$$

$$C_{l_{\delta_r}} = 2.26\times10^{-4} - 1.299\times10^{-5}\alpha + 5.565\times10^{-6}\alpha^2 - 3.382\times10^{-7}\alpha^3 + 6.461\times10^{-9}\alpha^4$$

$$C_{l_{\delta_{df}}} = -\left(7.453\times10^{-4} - 1.811\times10^{-5}\alpha - 1.264\times10^{-7}\alpha^2 + 9.972\times10^{-8}\alpha^3 - 2.684\times10^{-9}\alpha^4\right)$$

$$C_{l_p} = -0.5261357 + 0.029749438\alpha - 0.006960559\alpha^2 + 0.00053564\alpha^3 - 1.35043\times10^{-5}\alpha^4$$

$$C_{l_r} = 0.497793859 - 0.028742272\alpha + 0.016480243\alpha^2 - 0.001439044\alpha^3 + 3.56179\times10^{-5}\alpha^4$$

$$C_{m_0} = 2.632\times10^{-2} - 2.226\times10^{-3}\alpha - 1.8594\times10^{-5}\alpha^2 + 6.001\times10^{-7}\alpha^3 + 1.828\times10^{-7}\alpha^4$$

$$-9.733\times10^{-9}\alpha^5 + 1.71\times10^{-10}\alpha^6 - 5.233\times10^{-4}|\beta| + 6.795\times10^{-5}\beta^2$$

$$-1.993\times10^{-5}\left|\beta^3\right| + 1.341\times10^{-6}\beta^4 + 6.061\times10^{-5}\alpha|\beta|$$

$$C_{m_{\delta_e}} = -1.903\times10^{-3} - 1.593\times10^{-5}\alpha + 2.611\times10^{-6}\alpha^2 + 5.116\times10^{-8}\alpha^3 - 1.626\times10^{-9}\alpha^4$$

$$C_{m_{\delta_{pf}}} = -9.896\times10^{-4} - 1.494\times10^{-9}\alpha^2 + 6.303\times10^{-11}\alpha^4$$

$$C_{m_{\delta_{nf}}} = -1.086\times10^{-3} + 1.57\times10^{-5}\alpha + 4.174\times10^{-7}\alpha^2 - 1.133\times10^{-7}\alpha^3 + 2.723\times10^{-9}\alpha^4$$

$$\Delta C_{m,\delta_{lg}} = C_{m,\delta_{lg}}\left(\alpha,\delta_{lg}\right) - C_{m_0}\left(\alpha\right)$$

$$C_{m_q} = -0.195552417 - 0.006776550\alpha + .00285193\alpha^2 - 0.000184146\alpha^3$$

$$+2.45653\times10^{-6}\alpha^4$$

$$C_{n_{\delta_a}} = -2.769\times10^{-3} - 4.377\times10^{-5}\alpha + 9.952\times10^{-6}\alpha^2 - 3.642\times10^{-7}\alpha^3 + 4.692\times10^{-9}\alpha^4$$

$$C_{n_{\delta_r}} = -1.278\times10^{-3} + 1.32\times10^{-5}\alpha - 4.72\times10^{-6}\alpha^2 + 2.371\times10^{-7}\alpha^3 - 3.340\times10^{-9}\alpha^4$$

$$C_{n_{\delta_{df}}} = -5.107\times10^{-5} + 1.108\times10^{-5}\alpha - 1.547\times10^{-8}\alpha^2 - 1.552\times10^{-8}\alpha^3 + 1.413\times10^{-10}\alpha^4$$

$$C_{n_p} = 0.385147809 - 0.001356061\alpha - 0.00209489\alpha^2 + 8.52445\times10^{-5}\alpha^3 + 1.02309\times10^{-7}\alpha^4$$

$$C_{n_r} = -0.790855552 - 0.010770449\alpha - 0.00095452\alpha^2 + 0.000151569\alpha^3 - 5.13541\times10^{-6}\alpha^4$$

Data tables required for the component functions of the non-dimensional coefficients are defined in the file *HL20_aero_setup.m*. The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *HL20_aero.m*. Linear interpolation is done in the functions *cno.m*, *cmlgear.m*, *cxlgear.m*, and *czlgear.m*.

Dynamic Equations

Thrust effects and the power level state derivative were removed from the dynamic equations for the HL-20 simulation.

Mass Properties

Aircraft mass properties are given in the module *HL20_massprop.m*. Table 6 lists the mass properties used in the HL-20 simulation.

Table 6. Mass Properties of the Simulated HL-20

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	19,100	7,512	33,594	35,644	0

X-31 SIMULATION

The NASA / DARPA X-31 is a thrust-vectored, single-seat research aircraft designed to explore highly maneuverable flight. The aircraft has a cranked delta-wing planform, all-moving canard surfaces mounted near the nose, and a single vertical tail. The wing houses leading and trailing edge flaps. Trailing edge flaps act as elevon control surfaces, which function as separate elevator and aileron control surfaces within the simulation. Speed brakes are attached to the rear section of the fuselage and deflect symmetrically outward when actuated. A single General Electric F404-GE-400 non-afterburning turbofan engine provides thrust for the aircraft. Three paddles are symmetrically attached at the rear exit of the engine nozzle to provide thrust-vectoring capabilities. However, thrust vectoring is not implemented in this simulation. The effect of symmetric outboard deflection of the three thrust-vectoring paddles is modeled as an additional speed-braking device.

The aircraft was modeled with controls for δ_{th}, δ_e, δ_a, δ_r, δ_c, δ_{sb}, and δ_{tv}. Low-speed static wind tunnel tests provided force and moment aerodynamic data for a 19% scale model of the aircraft[18]. Tests were conducted in the Langley 14- by 22-foot subsonic tunnel with the model flown out of ground effect, with landing gear retracted, and no external stores. Dynamic data were obtained from forced oscillation wind tunnel testing[19]. Static aerodynamic data are in tabular form as a function of angle of attack and sideslip angle over the ranges $-5° \le \alpha \le 55°$ and $-18° \le \beta \le 18°$. Each non-dimensional aerodynamic force and moment coefficient is built up from component functions. Aerodynamic coefficients are referenced to a center of gravity location at $0.5286\bar{c}$, and corrected to the flight center of gravity position in the coefficient build-up equations.

Throttle deflection is limited to the range $0 \le \delta_{th} \le 1$, elevator deflection is limited to $-40° \le \delta_e \le 30°$, aileron deflection is limited to $-30° \le \delta_a \le 30°$, rudder deflection is limited to $-30° \le \delta_r \le 30°$, canard deflection is limited to $-20° \le \delta_c \le 60°$, speed brake deflection is limited to $0° \le \delta_{sb} \le 50°$, and thrust vectoring paddle deflection is limited to $0° \le \delta_{tv} \le 50°$. Data from the F-16 engine model was scaled using the maximum thrust values for the X-31 engine. Throttle gearing and engine power level dynamics were carried over from the F-16 engine model; however, engine angular momentum was ignored. Figure 13 shows a three-view of the X-31.

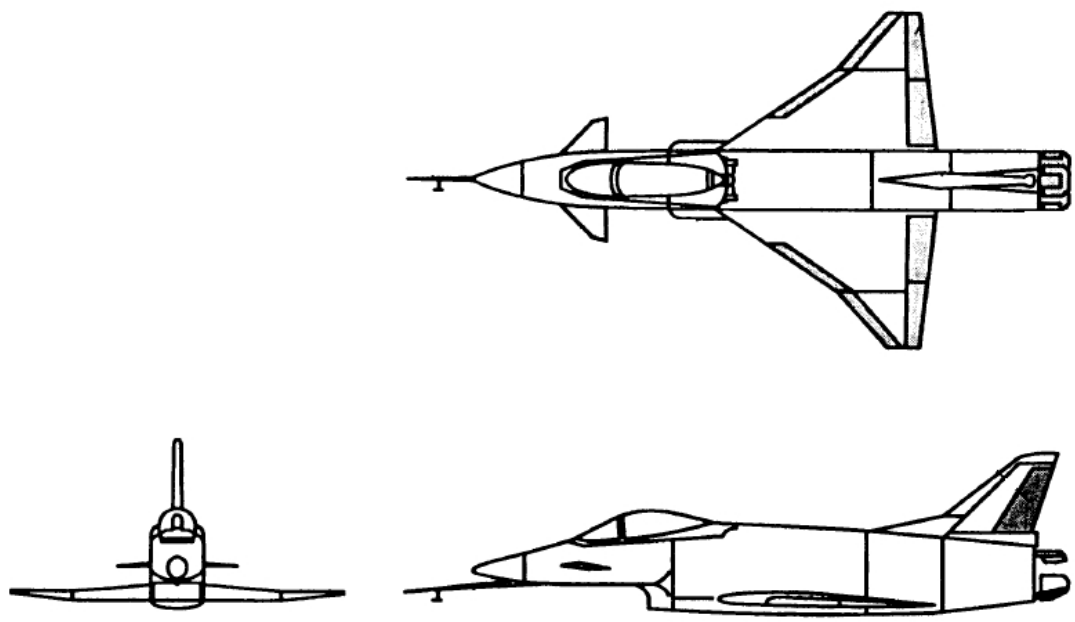

Figure 13. Three-View of the X-31

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the X-31 model vary nonlinearly with flow angles (α, β), aircraft angular velocities (p, q, r), and control surface deflections (δ_e, δ_a, δ_r, δ_c, δ_{sb}, δ_{tv}). Moment coefficients C_n and C_m include a correction for center of gravity position. The coefficients are computed as follows:

$$C_X = -\left[C_{A_0}(\alpha,\delta_c) + C_{A_{\delta_e}}(\alpha,\delta_e)\delta_e + \Delta C_{A,\delta_{sb}=50^\circ}\left(\frac{\delta_{sb}}{50}\right) + \Delta C_{A,\delta_{tv}=50^\circ}\left(\frac{\delta_{tv}}{50}\right)\right]$$

$$C_Y = C_{Y_0}(\beta) + \Delta C_{Y,\delta_r=30^\circ}\left(\frac{\delta_r}{30}\right) + \Delta C_{Y,\delta_a=30^\circ}\left(\frac{\delta_a}{30}\right) + \Delta C_{Y,\delta_{sb}=50^\circ}\left(\frac{\delta_{sb}}{50}\right)$$

$$+\Delta C_{Y,\delta_{tv}=50^\circ}\left(\frac{\delta_{tv}}{50}\right)$$

$$C_Z = -\left[C_{N_0}(\alpha,\delta_c) + C_{N_{\delta_e}}(\alpha,\delta_e)\delta_e + \Delta C_{N,\delta_{sb}=50^\circ}\left(\frac{\delta_{sb}}{50}\right) + \Delta C_{N,\delta_{tv}=50^\circ}\left(\frac{\delta_{tv}}{50}\right)\right]$$

$$-\left(\frac{q\bar{c}}{2V_t}\right)C_{N_q}(\alpha)$$

$$C_l = C_{l_0}(\beta) + \Delta C_{l,\delta_r=30^\circ}\left(\frac{\delta_r}{30}\right) + \Delta C_{l,\delta_a=30^\circ}\left(\frac{\delta_a}{30}\right) + \Delta C_{l,\delta_{sb}=50^\circ}\left(\frac{\delta_{sb}}{50}\right) + \Delta C_{l,\delta_{tv}=50^\circ}\left(\frac{\delta_{tv}}{50}\right)$$

$$+\left(\frac{b}{2V_t}\right)\left(C_{l_p}(\alpha)\,p + C_{l_r}(\alpha)\,r\right)$$

$$C_m = C_{m_0}(\alpha,\delta_c) + C_{m_{\delta_e}}(\alpha,\delta_e)\,\delta_e + \Delta C_{m,\delta_{sb}=50^\circ}\left(\frac{\delta_{sb}}{50}\right) + \Delta C_{m,\delta_{tv}=50^\circ}\left(\frac{\delta_{tv}}{50}\right)$$

$$+\left(\frac{q\bar{c}}{2V_t}\right)C_{m_q}(\alpha) + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = C_{n_0}(\beta) + \Delta C_{n,\delta_r=30^\circ}\left(\frac{\delta_r}{30}\right) + \Delta C_{n,\delta_a=30^\circ}\left(\frac{\delta_a}{30}\right) + \Delta C_{n,\delta_{sb}=50^\circ}\left(\frac{\delta_{sb}}{50}\right) + \Delta C_{n,\delta_{tv}=50^\circ}\left(\frac{\delta_{tv}}{50}\right)$$

$$+\left(\frac{b}{2V_t}\right)\left(C_{n_p}(\alpha)\,p + C_{n_r}(\alpha)\,r\right) - \left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

where

$$\Delta C_{A,\delta_{sb}=50^\circ} = C_{A,\delta_{sb}=50^\circ}(\alpha) - C_{A_0}(\alpha,\delta_c)$$

$$\Delta C_{A,\delta_{tv}=50^\circ} = C_{A,\delta_{tv}=50^\circ}(\alpha) - C_{A_0}(\alpha,\delta_c)$$

$$\Delta C_{N,\delta_{sb}=50^\circ} = C_{N,\delta_{sb}=50^\circ}(\alpha) - C_{N_0}(\alpha,\delta_c)$$

$$\Delta C_{N,\delta_{tv}=50^\circ} = C_{N,\delta_{tv}=50^\circ}(\alpha) - C_{N_0}(\alpha,\delta_c)$$

$$\Delta C_{Y,\delta_r=30^\circ} = C_{Y,\delta_r=30^\circ}(\beta) - C_{Y_0}(\beta)$$

$$\Delta C_{Y,\delta_a=30^\circ} = C_{Y,\delta_a=30^\circ}(\beta) - C_{Y_0}(\beta)$$

$$\Delta C_{Y,\delta_{sb}=50^\circ} = C_{Y,\delta_{sb}=50^\circ}(\alpha) - C_{Y_0}(\beta)$$

$$\Delta C_{Y,\delta_{tv}=50^\circ} = C_{Y,\delta_{tv}=50^\circ}(\alpha) - C_{Y_0}(\beta)$$

$$\Delta C_{l,\delta_r=30^\circ} = C_{l,\delta_r=30^\circ}(\beta) - C_{l_0}(\beta)$$

$$\Delta C_{l,\delta_a=30^\circ} = C_{l,\delta_a=30^\circ}(\beta) - C_{l_0}(\beta)$$

$$\Delta C_{l,\delta_{sb}=50^\circ} = C_{l,\delta_{sb}=50^\circ}(\alpha) - C_{l_0}(\beta)$$

$$\Delta C_{l,\delta_{tv}=50^\circ} = C_{l,\delta_{tv}=50^\circ}(\alpha) - C_{l_0}(\beta)$$

$$\Delta C_{n,\delta_r=30^\circ} = C_{n,\delta_r=30^\circ}(\beta) - C_{n_0}(\beta)$$

$$\Delta C_{n,\delta_a=30^\circ} = C_{n,\delta_a=30^\circ}(\beta) - C_{n_0}(\beta)$$

$$\Delta C_{n,\delta_{sb}=50^\circ} = C_{n,\delta_{sb}=50^\circ}(\alpha) - C_{n_0}(\beta)$$

$$\Delta C_{n,\delta_{tv}=50^\circ} = C_{n,\delta_{tv}=50^\circ}(\alpha) - C_{n_0}(\beta)$$

$$\Delta C_{m,\delta_{sb}=50^\circ} = C_{m,\delta_{sb}=50^\circ}(\alpha) - C_{m_0}(\alpha,\delta_c)$$

$$\Delta C_{m,\delta_{tv}=50^\circ} = C_{m,\delta_{tv}=50^\circ}(\alpha) - C_{m_0}(\alpha,\delta_c)$$

Data tables for the component functions of the non-dimensional coefficients are defined in the file *X31_aero_setup.m*. The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *X31_aero.m*, using linear interpolation in the functions *ca_can.m*, *dca_dele.m*, *dca_dsb.m*, *dca_dtv.m*, *cnorm_can.m*, *dcnorm_dele.m*, *dcnorm_dsb.m*, *dcnorm_dtv.m*, *cy_beta.m*, *dcy_drdr.m*, *dcy_dail.m*, *dcy_dsb.m*, *dcy_dtv.m*, *cm_can.m*, *dcm_dele.m*, *dcm_dsb.m*, *dcm_dtv.m*, *cl_beta.m*, *dcl_drdr.m*, *dcl_dail.m*, *dcl_dsb.m*, *dcl_dtv.m*, *cn_beta.m*, *dcn_drdr.m*, *dcn_dail.m*, *dcn_dsb.m*, *dcn_dtv.m*, *caq.m*, *cyr.m*, *cyp.m*, *cnormq.m*, *clr.m*, *clp.m*, *cmq.m*, *cnr.m*, and *cnp.m*.

Engine Model and Database

Thrust scaling for the X-31 engine is applied in the function *X31_engine_setup.m*. Linear interpolation and thrust calculation is performed in the module *X31_engine.m*. Functions *tgear.m*, *pdot.m*, and *rtau.m* are unaltered from the F-16 engine simulation.

Mass Properties

Aircraft mass properties are given in the module *X31_massprop.m*. Table 7 lists the mass properties used in the X-31 simulation.

Table 7. Mass Properties of the Simulated X-31

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	16,000	3,553	50,645	49,367	-156

A-7 SIMULATION

The Vought A-7 Corsair II is a single-seat tactical attack fighter based on the F-8 Crusader platform. The aircraft is controlled with ailerons located on the trailing edge of the main wing, a rudder control surface located at the trailing edge of the vertical tail, and an all-moving horizontal stabilator located at the rear of the fuselage. Thrust is provided by a single non-afterburning Pratt & Whitney TF30-P-408 turbofan engine mounted in the rear fuselage.

Aircraft controls are provided for δ_{th}, δ_e, δ_a, and δ_r. Low-speed static wind tunnel tests and dynamic forced oscillation wind tunnel tests provided force and moment data for a scale model of the aircraft flown out of ground effect, with landing gear retracted, and no external stores[20]. Static and dynamic aerodynamic data are in tabular form as a function of angle of attack and sideslip over the ranges $0° \le \alpha \le 90°$ and $0° \le \beta \le 90°$. Each non-dimensional aerodynamic force and moment coefficient is built up from component functions. Aerodynamic coefficients are referenced to a center of gravity location at $0.35\bar{c}$, and corrected to the flight center of gravity position in the coefficient build-up equations.

Throttle deflection is limited to the range $0 \le \delta_{th} \le 1$, elevator deflection is limited to $-25° \le \delta_e \le 15°$, aileron deflection is limited to $-25° \le \delta_a \le 25°$, and rudder deflection is limited to $-30° \le \delta_r \le 30°$. The A-7 engine has a maximum thrust of 13,400 lb. A scaled version of the F-16 engine model was used for this simulation. Idle and military power thrust data were scaled according to the maximum thrust for the A-7 engine. Throttle gearing, engine gyroscopic effects, and engine power level dynamics were carried over from the F-16 engine model. Figure 14 shows a three-view of the A-7 aircraft.

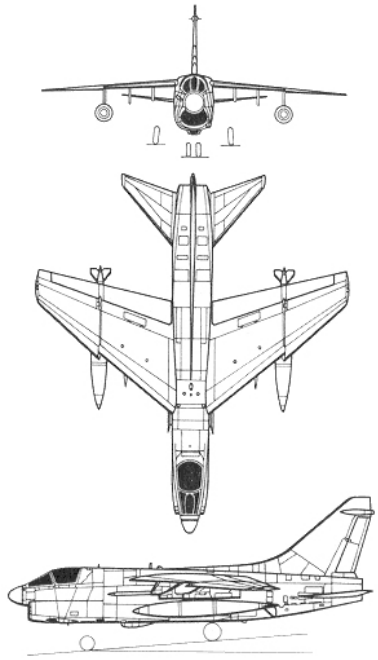

Figure 14. Three-View of the A-7

Aerodynamic Model and Database

The non-dimensional aerodynamic force and moment coefficients for the A-7 vary nonlinearly with flow angles (α, β), aircraft angular velocities (p, q, r), and control surface deflections (δ_e, δ_a, δ_r). Moment coefficients C_n and C_m include a correction for the center of gravity position. The coefficients are computed as follows:

$$C_X = C_L(\alpha, \delta_e) sin\left(\frac{\alpha\pi}{180}\right) - C_D(\alpha, \delta_e) cos\left(\frac{\alpha\pi}{180}\right)$$

$$C_Y = C_{Y_0}(\alpha, |\beta|) sign(\beta) + C_{Y_{\delta_r}}(\alpha)\delta_r + C_{Y_{\delta_{a,right}}}(\alpha, \delta_{a,right})\delta_{a,right} - C_{Y_{\delta_{a,left}}}(\alpha, \delta_{a,left})\delta_{a,left}$$

$$C_Z = -C_L cos\left(\frac{\alpha\pi}{180}\right) - C_D sin\left(\frac{\alpha\pi}{180}\right)$$

$$C_l = C_{l_0}\left(\alpha,|\beta|\right) sign\left(\beta\right) + C_{l,\delta_{a,right}}\left(\alpha,\delta_{a,right}\right) - C_{l,\delta_{a,left}}\left(\alpha,\delta_{a,left}\right) + C_{l_{\delta_r}}\left(\alpha\right)\delta_r$$
$$+\left(\frac{b}{2V_t}\right)\left(C_{l_p}\left(\alpha\right)p + C_{l_r}\left(\alpha\right)r\right)$$

$$C_m = C_{m_0}\left(\alpha,\delta_e\right) + \left(\frac{q\bar{c}}{2V_t}\right)\left[C_{m_q}\left(\alpha,\delta_e\right) + C_{m_{\dot{\alpha}}}\left(\alpha,\delta_e\right)\right] + C_{m,\delta_{a,right}}\left(\alpha,\delta_{a,right}\right)$$
$$+C_{m,\delta_{a,left}}\left(\alpha,\delta_{a,left}\right) + \left(x_{c.g.,ref} - x_{c.g.}\right)C_Z$$

$$C_n = C_{n_0}\left(\alpha,|\beta|\right) sign\left(\beta\right) + C_{n,\delta_{a,right}}\left(\alpha,\delta_{a,right}\right) - C_{n,\delta_{a,left}}\left(\alpha,\delta_{a,left}\right) + C_{n_{\delta_r}}\left(\alpha\right)\delta_r$$
$$+\left(\frac{b}{2V_t}\right)\left(C_{n_p}\left(\alpha\right)p + C_{n_r}\left(\alpha\right)r\right) - \left(\frac{\bar{c}}{b}\right)\left(x_{c.g.,ref} - x_{c.g.}\right)C_Y$$

Data tables for the component functions of the non-dimensional coefficients are defined in the file *A7_aero_setup.m*. The non-dimensional aerodynamic force and moment coefficients are computed for this simulation in *A7_aero.m*, using linear interpolation in one- and two-dimensional interpolation functions *int1.m* and *int2.m*.

Engine Model and Database

Thrust scaling for the A-7 engine is applied in the function *A7_engine_setup.m* using the F-16 engine data, omitting afterburner data. Throttle gearing, implemented in *tgear.m*, outputs commanded engine power in percent of full power [0,100] using throttle setting input [0,1]. Commanded power P_c is a linear function of δ_{th}, given by

$$P_c\left(\delta_{th}\right) = 100\delta_{th}$$

Functions *pdot.m* and *rtau.m* are unaltered from the F-16 engine simulation.

Linear interpolation and thrust calculation is performed in the module *A7_engine.m*, scaling F-16 engine data to the maximum thrust of the A-7 engine. Scaled thrust is computed from

$$T = \left(\frac{13,400}{12,680}\right)\left(T_{idle} + 0.01\left(T_{mil} - T_{idle}\right)P_a\right)$$

Mass Properties

Aircraft mass properties are given in the module *A7_massprop.m*. Table 8 lists the mass properties used in the A-7 simulation.

Table 8. Mass Properties of the Simulated A-7

Parameter	Weight (lb)	I_X (slug-ft^2)	I_Y (slug-ft^2)	I_Z (slug-ft^2)	I_{XZ} (slug-ft^2)
Value	22,699	16,970	65,430	76,130	4,030

VI. GENERATING 3-D MODELS FOR AVDS

Three-dimensional aircraft models used within AVDS are constructed using a separate CAD drafting program. Models created with CAD are converted to an input format suitable for AVDS. AVDS supplies a utility that allows conversion of Drawing Interchange Format (DXF) 3-D aircraft models into AVDS format. The following sections provide a tutorial for creating and adding a 3-D aircraft model into AVDS for use in AVDS / MATLAB piloted simulations.

MODEL CONSTRUCTION REQUIREMENTS

Three-dimensional aircraft models must be constructed in AutoCAD® DXF format using surface modeling techniques. The AVDS conversion program does not recognize models created using solid modeling techniques. Surface models have exteriors composed of meshes made of three and four sided polygons whose vertices are located in three-dimensional space. The total number of polygons in a surface mesh determines model complexity. AVDS / MATLAB interactive simulations are computationally intensive; therefore, it is necessary to create 3-D models that are as simple as possible in order to minimize graphics rendering requirements. Complex models with dense surface meshes can slow AVDS / MATLAB interactive simulations considerably on less powerful computers.

Many 3-D CAD programs support both surface modeling and the DXF format. Ruled surfaces, tabulated surfaces, and edge surfaces are used extensively during surface model creation. A surface created using any of these tools is recognized by the AVDS conversion utility. However, revolved surfaces and two-dimensional lines are not recognized during conversion.

Aircraft 3-D models should be created using two-dimensional surfaces for the wings and horizontal / vertical tails. This reduces the complexity of the aircraft surface mesh while not excessively detracting from aircraft realism. The fuselage, canopy, and other significant aircraft structures should be created as simply as possible using low-density surface meshes. Following these guidelines will simplify model making as well as prevent any unnecessary slow-downs during piloted simulations run on less powerful computers.

MODELING EXAMPLE OF A SIMPLE 3-D AIRCRAFT

An example is given to illustrate the surface model creation of a simple 3-D aircraft, shown in figure 15. It is assumed that the reader has familiarity with using and drawing in AutoCAD Release 14 (or higher) or an equivalent 3-D CAD program. For more information on CAD design or surface modeling consult any AutoCAD Release 14 (or higher) reference book and surface-modeling tutorial such as Reference [21]. Aileron, rudder, and elevator control surfaces will be included in the model, while wheels and any external stores are ignored.

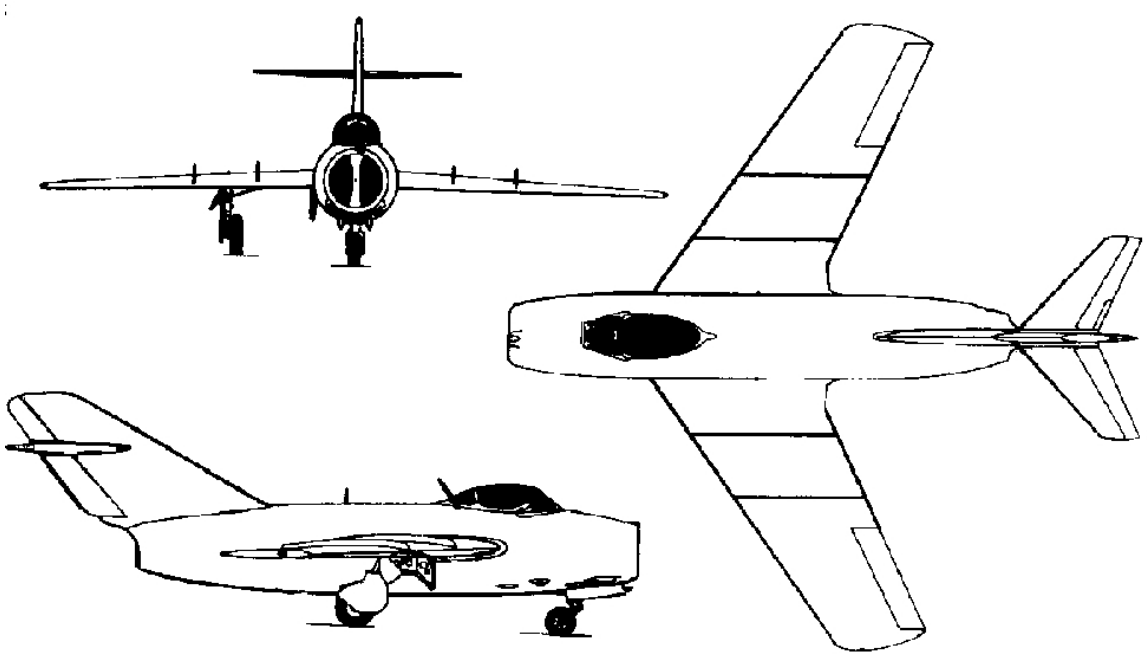

Figure 15. Three-View of the MiG-15 Aircraft

It is convenient to model aircraft using available three-views such as the one shown in figure 15. These drawings may be found on various internet sites, and in references such as [22]. Dimensions used during initial modeling of the aircraft should be those of the drawing rather than the actual airplane. This simplifies model-making by eliminating the need to scale the dimension of every point on the model. Once modeling is complete, the finished 3-D aircraft must be scaled to the dimensions of the actual aircraft. Dimensions must be in feet for proper implementation into AVDS. The scaling feature is available in most CAD programs. The scaled model must also be aligned along the world coordinate system (WCS), prior to conversion into the AVDS format, with the nose of the model pointing along the positive X-axis and positive Z-axis pointing out the top of the model. The origin of the WCS must be positioned at the location of the aircraft center of gravity. AVDS uses this origin location as placement for the body, earth, and stability axes on the model during interactive simulations.

The fuselage for this particular aircraft is a relatively simple shape and can be modeled as a cylinder with the front and back openings used as the intake and exhaust. Figure 16 shows the preliminary front and rear openings of the aircraft.

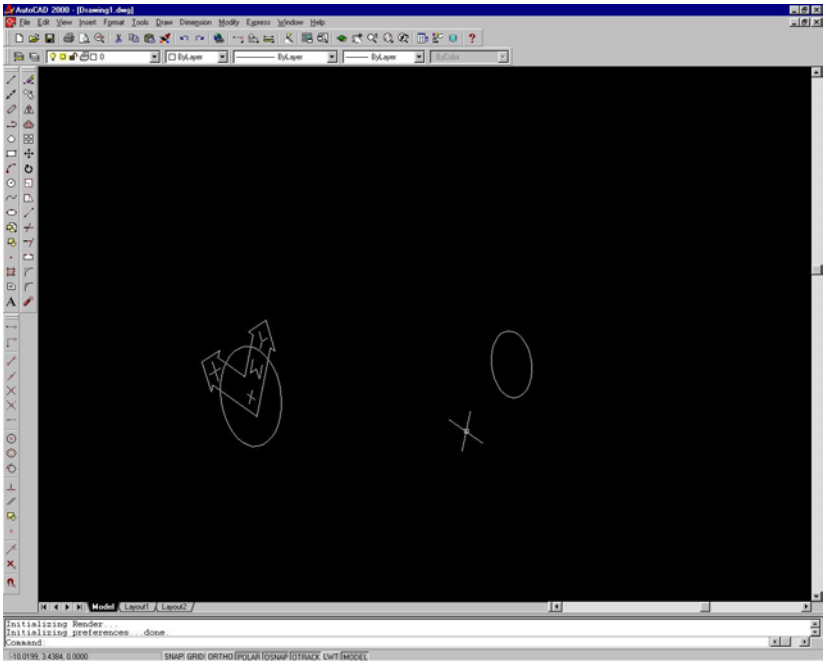

Figure 16. Front and Rear Fuselage Openings

The AutoCAD command RULESURF is used to create a surface mesh between the intake and exhaust. The completed fuselage is shown in figure 17.

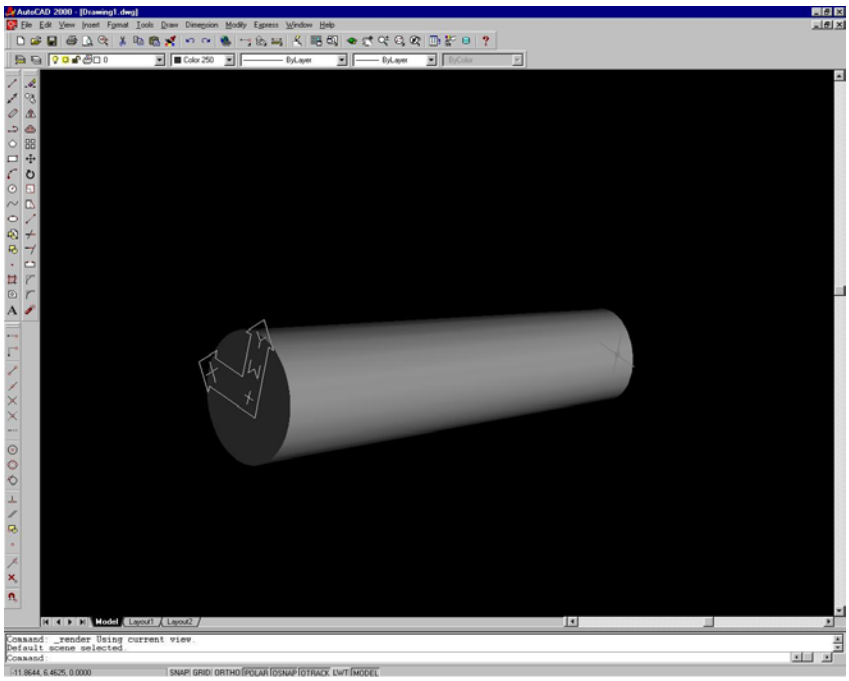

Figure 17. Modeled MiG-15 Fuselage Section

The wing, tail, and control surfaces of the aircraft are modeled using planar surfaces. This is accomplished using the AutoCAD command 3DFACE. Figure 18 shows the

aircraft with the various surfaces added. RULESURF was used to create surface meshes for the modeled intake and exhaust.

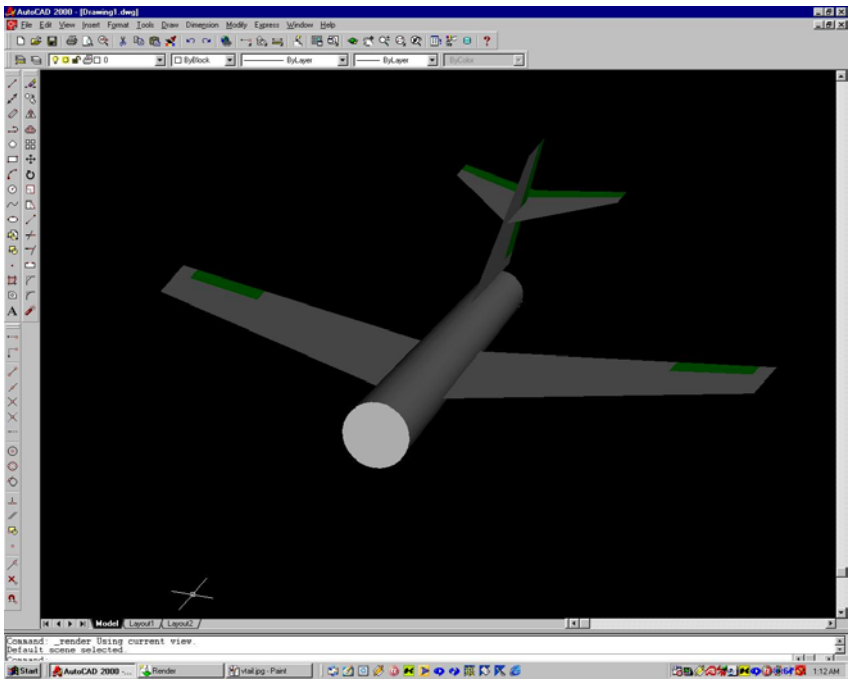

Figure 18. 3-D Model with Wing, Tail, and Control Surfaces

The canopy can be modeled as an elongated dome. A wire frame is constructed outlining the general shape of the canopy, shown in figure 19. Surface meshes are drawn over the wire frame using the RULESURF command.

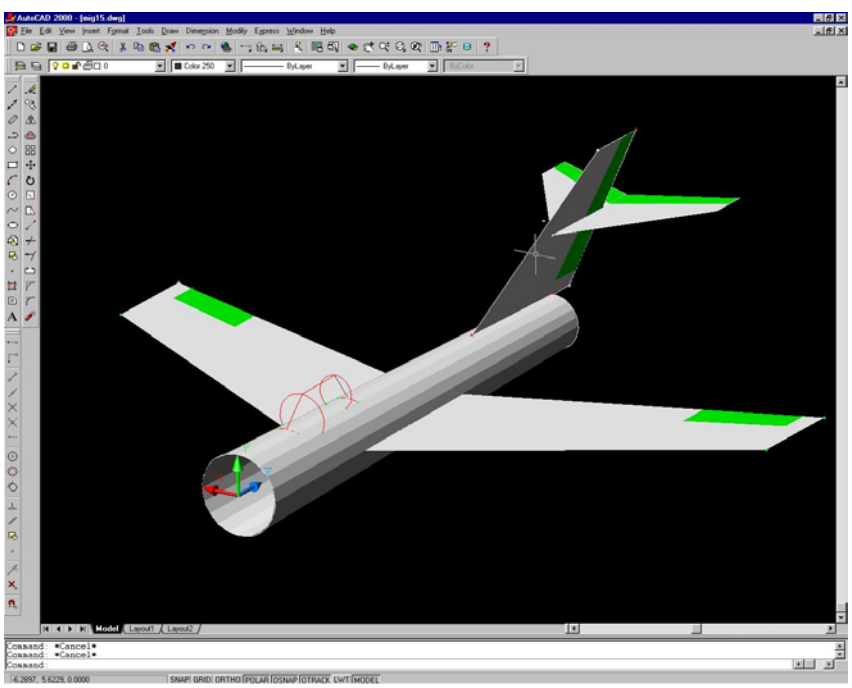

Figure 19. Model with Wire Frame Canopy

The finished canopy is shown in Figure 20. The final drawing must be scaled (in feet) to the dimensions of the actual aircraft and aligned accordingly with the WCS specified previously.

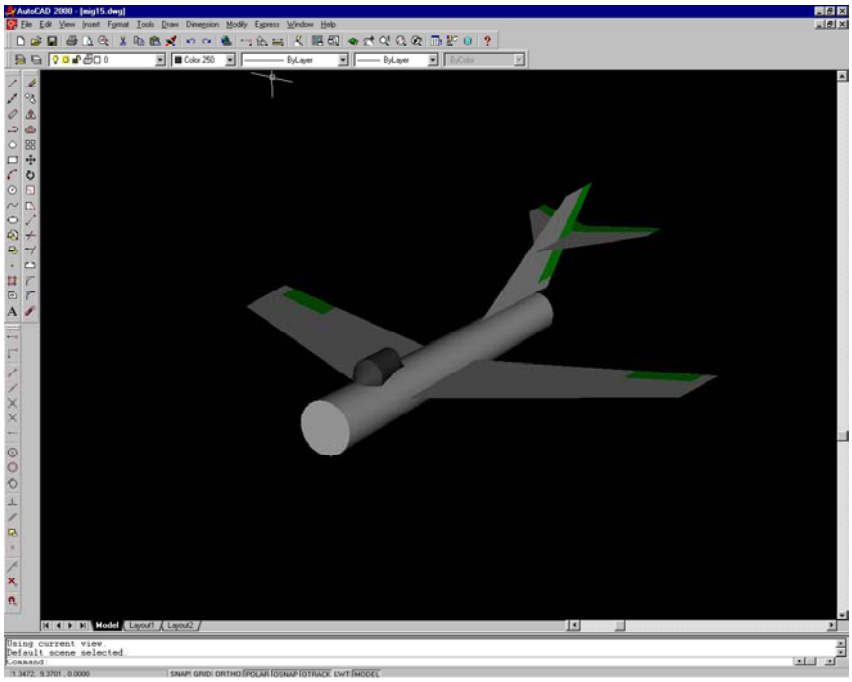

Figure 20. Completed 3-D MiG-15 Model

IMPORTING A COMPLETED 3-D CAD MODEL INTO AVDS

A completed model saved in DXF format is converted to the AVDS format with the DXF to AVDS conversion utility. For usage instructions, refer to Appendix E of the AVDS User's Manual[2]. The box marked *Two sided polygons* should be checked in the interface window of the conversion utility prior to conversion, in order to create surface meshes composed of front and back sided polygons. If this option is not chosen, meshes composed of either front or back sided polygons will be shown in AVDS. This would cause surfaces to disappear unexpectedly when viewing the aircraft from certain angles.

The DXF to AVDS conversion utility has difficulty implementing entire aircraft models into the AVDS format. Therefore, it is necessary to separate each major model element (e.g., elements that will share the same color, control surfaces elements, canopy, etc.) into individually saved DXF files. These individual files are converted separately and assembled into a single AVDS craft file by the user. The user must identify section color, wing and tail ribbon location, pilot view, control surfaces, and control surface rotation axes for each newly compiled AVDS craft file, following the format given in Appendix A of the AVDS User's Manual[2].

Control surfaces for new aircraft models must also be identified within the files *ACname.sim.ini* and *AVDS_Matlab_ACname.m*, where *ACname* is the aircraft name (e.g. *F16.sim.ini*, *AVDS_Matlab_F16*, etc.), using the same control surface category

designations as were used in the AVDS craft file. Refer to Section 4.2 in the AVDS User's Manual[2] for instructions on creating / modifying *ACname.sim.ini* files. Figure 21 shows the input screen for files of this type. Within this configuration window, **FCS** must be set to *Basic FCS*, **AIRCRAFT** should be set to *MATLAB AC*, and **Simulation Input Type** should be set to *Positions*. Elements within the MATLAB vectors *InputVector* and *OutputVector*, which have been declared by the user in *AVDS_Matlab_ACname.m*, must also be declared to AVDS via the *ACname.sim.ini* file. The user must declare *InputVector* elements under the **MATLAB AC** sub-heading section of **INPUTS**, while *OutputVector* elements must be declared under the **Environment** sub-heading section under **INPUTS**. Control surfaces are identified in AVDS as *sfc01* to *sfc15* under the **Environment** sub-heading section under **INPUTS.**

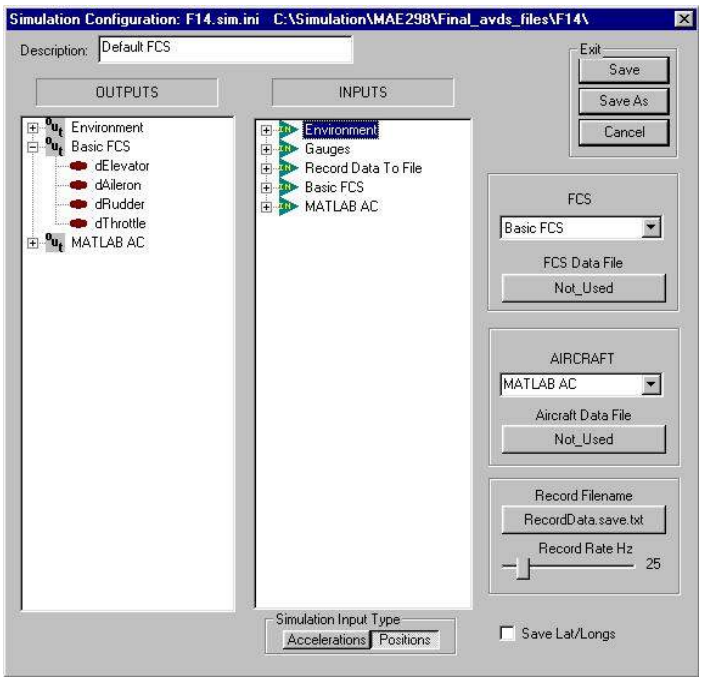

Figure 21. Input Screen for *ACname.sim.ini*

VII. CONCLUDING REMARKS

Software has been developed in MATLAB for six-degree-of-freedom nonlinear simulations of various aircraft. Trim, linearization, numerical integration, batch simulation, 3-D visualization, and real-time piloted simulation software are included for all simulations.

Aerodynamic models were developed using data gathered from open literature reports documenting wind tunnel tests, flight tests, and analytic estimation techniques. Most of the engine models contained in the aircraft simulation packages were based on the F-16 engine model, using scaled values for thrust. Other models were used for aircraft engines when suitable data was available or when the jet engine model did not apply (e.g., FASER and the HL-20 aircraft).

Interactive piloted simulations can be run using the nonlinear aircraft simulation packages and visual flight simulator software. The nonlinear simulations described in this report can be used for education, control law development, dynamic analysis, and many other purposes.

The structure of the simulations was designed for easy use and to allow modifications with relatively little effort. The collection of aircraft simulations gives the user a range of aircraft dynamics to explore. This compilation is also a testament of the applicability of the basic simulation structure to a variety of aircraft.

Future improvements to this work, outside of adding more aircraft, could include such changes as adding landing gear dynamics, allowing for more general aircraft c.g. locations, including a more realistic gravitational model that accounts for earth geodesy, including a non-stationary atmosphere (e.g., turbulence, micro-bursts, winds, etc.), and adding aeroelastic effects. Since AVDS currently allows only four control inputs, an improvement that allows the user to command a larger number of controls would be desirable for interactive simulation. It is anticipated that network capability will be built into future editions of the simulations, allowing multiple users to fly interactively with (or against) one another in real time. Finally, an obvious improvement for nearly all the aircraft simulations would be the addition of a flight control system. This was intentionally omitted in the current work, to encourage the use of these aircraft nonlinear simulations for control system design and analysis.

VIII. ACKNOWLEDGEMENTS

The original versions of the aerodynamic and engine models for some of the aircraft simulations described in this report were developed by students in the George Washington University graduate course entitled *Stability and Control of Aircraft*, taught by the second author. The authors gratefully acknowledge the work of Javier Velez on the F-14, Bo Trieu on the F-4, Byron Monzón on the FASER, Pat Hinchy on the HL-20, Mike Lensi on the X-31, and Brian Stewart and Tony Tyler on the A-7. Thanks also to Steve Rasmussen for technical work on the MATLAB/AVDS interface software.

IX. APPENDIX

Table A-1. F-16 Simulation Package (Directory name: F-16)

F-16 Directory	
ab3.m	F16_deq.m
atm.m	F16_engine.m
AVDS_Matlab_F16.m	F16_engine_setup.m
clo.m	F16_hires.txt
cmo.m	F16_lores.txt
cno.m	F16_massprop.m
cnvrg.m	F16_trm.m
cxo.m	gen_F16_model.m
czo.m	grad.m
dampder.m	ic_ftrm.m
dlda.m	lnze.m
dldr.m	mksqw.m
dnda.m	pdot.m
dndr.m	rk2.m
F16.m	rk4.m
F16.sim.ini	rtau.m
F16_aero.m	solve.m
F16_aero_setup.m	tgear.m

Table A-2. F-106B Simulation Package (Directory name: F-106B)

F-106B Directory	
ab3.m	F106B_aero.m
atm.m	F106B_aero_setup.m
AVDS_Matlab_F106B.m	F106B_deq.m
cdo.m	F106B_engine.m
clo.m	F106B_engine_setup.m
cmo.m	F106B_massprop.m
cnvrg.m	F106B_trm.m
dampder.m	gen_F106B_model.m
dlda.m	grad.m
dldb.m	ic_ftrm.m
dldr.m	lnze.m
dnda.m	mksqw.m
dndb.m	pdot.m
dndr.m	rk2.m
dydb.m	rk4.m
dydr.m	rtau.m
F106B.m	solve.m
F106B.sim.ini	tgear.m
F106B.txt	

Table A-3. F-14 Simulation Package (Directory name: F-14)

F-14 Directory		
ab3.m	dcmsp.m	F14_deq.m
atm.m	dcxds.m	F14_engine.m
AVDS_Matlab_F14.m	dcxsb.m	F14_engine_setup.m
cldsp.m	dcxsp.m	F14_massprop.m
clo.m	dczds.m	F14_trm.m
cmo.m	dczsb.m	gen_F14_model.m
cndsp.m	dczsp.m	grad.m
cno.m	dlda.m	ic_ftrm.m
cnvrg.m	dldr.m	lnze.m
cxo.m	dnda.m	mksqw.m
cyda.m	dndr.m	pdot.m
cydr.m	etalsp.m	rk2.m
cydsp.m	etamsp.m	rk4.m
cyo.m	F14.m	rtau.m
czo.m	F14.sim.ini	solve.m
dampder.m	F14.txt	tgear.m
dcmds.m	F14_aero.m	
dcmsb.m	F14_aero_setup.m	

Table A-4. F-4 Simulation Package (Directory name: F-4)

F-4 Directory	
ab3.m	F4_trm.M
atm.m	gen_F4_model.m
AVDS_Matlab_F4.m	grad.m
cnvrg.m	ic_ftrm.m
F4.m	lnze.m
F4.sim.ini	mksqw.m
F4.txt	pdot.m
F4_aero.m	rk2.m
F4_deq.m	rk4.m
F4_engine.m	rtau.m
F4_engine_setup.m	solve.m
F4_massprop.m	tgear.m

Table A-5. FASER Simulation Package (Directory name: FASER)

FASER Directory	
ab3.m	grad.m
atm.m	ic_ftrm.m
AVDS_Matlab_FASER.m	int1.m
cnvrg.m	int2.m
FASER.m	int3.m
FASER.sim.ini	lat_aero_tables.mat
FASER_aero.m	lnze.m
FASER_aero_setup.m	lon_aero_tables.mat
FASER_deq.m	mksqw.m
FASER_engine.m	pdot.m
FASER_hires.txt	rk2.m
FASER_lores.txt	rk4.m
FASER_massprop.m	rtau.m
FASER_trm.m	solve.m
gen_FASER_model.m	tgear.m

Table A-6. HL-20 Simulation Package (Directory name: HL-20)

HL-20 Directory	
ab3.m	HL20.txt
atm.m	HL20_aero.m
AVDS_Matlab_HL20.m	HL20_aero_setup.m
cmlgear.m	HL20_deq.m
cno.m	HL20_massprop.m
cnvrg.m	HL20_trm.m
cxlgear.m	ic_ftrm.m
czlgear.m	lnze.m
gen_HL20_model.m	mksqw.m
grad.m	rk2.m
HL20.m	rk4.m
HL20.sim.ini	solve.m

Table A-7. X-31 Simulation Package (Directory name: X-31)

X-31 Directory		
ab3.m	dca_dtv.m	ic_ftrm.m
atm.m	dcl_dail.m	lnze.m
AVDS_Matlab_X31.m	dcl_drdr.m	mksqw.m
caq.m	dcl_dsb.m	pdot.m
ca_can.m	dcl_dtv.m	rk2.m
clp.m	dcm_dele.m	rk4.m
clr.m	dcm_dsb.m	rtau.m
cl_beta.m	dcm_dtv.m	solve.m
cmq.m	dcnorm_dele.m	tgear.m
cm_can.m	dcnorm_dsb.m	X31.m
cnormq.m	dcnorm_dtv.m	X31.sim.ini
cnorm_can.m	dcn_dail.m	X31.txt
cnp.m	dcn_drdr.m	X31_aero.m
cnr.m	dcn_dsb.m	X31_aero_setup.m
cnvrg.m	dcn_dtv.m	X31_deq.m
cn_beta.m	dcy_dail.m	X31_engine.m
cyp.m	dcy_drdr.m	X31_engine_setup.m
cyr.m	dcy_dsb.m	X31_massprop.m
cy_beta.m	dcy_dtv.m	X31_trm.m
dca_dele.m	gen_X31_model.m	
dca_dsb.m	grad.m	

Table A-8. A-7 Simulation Package (Directory name: A-7)

A-7 Directory	
ab3.m	gen_A7_model.m
atm.m	grad.m
AVDS_Matlab_A7.m	ic_ftrm.m
A7.txt	int1.m
A7.m	int2.m
A7.sim.ini	lnze.m
A7_aero.m	mksqw.m
A7_aero_setup.m	pdot.m
A7_deq.m	rk2.m
A7_engine.m	rk4.m
A7_engine_setup.m	rtau.m
A7_massprop.m	solve.m
A7_trm.m	tgear.m
cnvrg.m	

X. REFERENCES

1. *Using MATLAB, Version 6*, The MathWorks, Inc., Natick, MA, November 2000.
2. *AVDS User's Manual*, RasSim Tech Ltd., Columbus, OH, October 2000.
3. Stevens, B.L. and Lewis, F.L., *Aircraft Control and Simulation*, John Wiley & Sons, Inc., New York, NY, 1992.
4. Morelli, E.A., *Airplane Flight Mechanics - Stability and Control*, Class Notes, December 2000.
5. Schmidt, L.V. *Introduction to Aircraft Flight Dynamics*, American Institute of Aeronautics and Astronautics, Inc., Reston, VA, 1998.
6. Cook, M.V. *Flight Dynamics Principles*, John Wiley & Sons Inc., New York, NY, 1997.
7. Etkin, B. and Reid, L.D. *Dynamics of Flight, Stability and Control*, 3rd Ed., John Wiley & Sons, Inc., New York, NY, 1996.
8. Burden, R.L. and Faires, J.D. *Numerical Analysis* 6th Ed., Brooks / Cole Publishing Company, Pacific Grove, CA. 1997.

F-16

9. Nguyen, L.T., Ogburn, M.E., Gilbert, W.P., Kibler, K.S., Brown, P.W., and Deal, P.L., "Simulator Study of Stall / Post-Stall Characteristics of a Fighter Airplane With Relaxed Longitudinal Static Stability", NASA TP 1538, December 1979.

F-106B

10. Teper, G.L., "Aircraft Stability and Control Data", NASA CR 96008, April 1969.
11. Yip, L.P., "Wind-Tunnel Free-Flight Investigation of a 0.15-Scale Model of the F-106B Airplane With Vortex Flaps", NASA TP 2700, May 1987.

F-14

12. Freudinger, L.C., and Kehoe, M.W., "Flutter Clearance of the F-14A Variable-Sweep Transition Flight Experiment Airplane-Phase 2", NASA TM 101717, July 1990.
13. Gilbert, W.P., Nguyen, L.T., and Van Gunst, R.W. "Simulator Study of Applications of Automatic Departure and Spin Prevention Concepts to a Variable Sweep Fighter Airplane," NASA TM X-2928, November 1973.

F-4

14. Eulrich, B.J., and Weingrarten, N.C., "Identification and Correlation of the F-4E Stall / Post-Stall Aerodynamic Stability and Control Characteristics from Existing Test Data", AFFDL-TR-73-125, May 1974.

FASER

15. Monzón, B.R., "Non-Linear Simulation Development For a Sub-Scale Research Airplane", M.S. Thesis, The George Washington University, January 2002.

HL-20

16. Dutton, K.E., "Optimal Control Theory Determination of Feasible Return-to-Launch-Site Aborts for the HL-20 Personnel Launch System Vehicle", NASA TP 3449, July 1994.

17. Jackson, E.B., and Cruz, C.I., "Preliminary Subsonic Aerodynamic Model for Simulation Studies of the HL-20 Lifting Body", NASA TM 4302, August 1992.

X-31

18. Banks, D.W., Gatlin, G.M., and Paulson Jr., J.W., "Low-Speed Longitudinal and Lateral-Directional Aerodynamic Characteristics of the X-31 Configuration", NASA TM 4351, October 1992.

19. Smith, M.S., "Analysis of Wind Tunnel Oscillatory Data of the X-31A Aircraft", NASA CR-1999-208725, February 1999.

A-7

20. Johnston, D.E., Hogge, J.R., and Teper, G.L. "Investigation of Flying Qualities of Military Aircraft at High Angles of Attack, Vol. II. Appendices", AFFDL-TR-74-61, June 1974.

General

21. McFarlane, B. *Modelling with AutoCAD R14 for Windows NT and Windows 95*, John Wiley & Sons Inc., New York, New York. 1999.

22. Taylor, J.W.R. ed. Jane's All the World's Aircraft 1989-90, Jane's Information Group Ltd., Coulsdon, Surrey, UK. 1989.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
January 2003

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
A Collection of Nonlinear Aircraft Simulations in MATLAB

5. FUNDING NUMBERS
WU 728-30-30-03

6. AUTHOR(S)
Garza, Frederico R.; and Morelli, Eugene A.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-18259

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-2003-212145

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 05 Distribution: NonStandard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Nonlinear six degree-of-freedom simulations for a variety of aircraft were created using MATLAB. Data for aircraft geometry, aerodynamic characteristics, mass / inertia properties, and engine characteristics were obtained from open literature publications documenting wind tunnel experiments and flight tests. Each nonlinear simulation was implemented within a common framework in MATLAB, and includes an interface with another commercially-available program to read pilot inputs and produce a three-dimensional (3-D) display of the simulated airplane motion. Aircraft simulations include the General Dynamics F-16 Fighting Falcon, Convair F-106B Delta Dart, Grumman F-14 Tomcat, McDonnell Douglas F-4 Phantom, NASA Langley Free-Flying Aircraft for Sub-scale Experimental Research (FASER), NASA HL-20 Lifting Body, NASA / DARPA X-31 Enhanced Fighter Maneuverability Demonstrator, and the Vought A-7 Corsair II. All nonlinear simulations and 3-D displays run in real time in response to pilot inputs, using contemporary desktop personal computer hardware. The simulations can also be run in batch mode. Each nonlinear simulation includes the full nonlinear dynamics of the bare airframe, with a scaled direct connection from pilot inputs to control surface deflections to provide adequate pilot control. Since all the nonlinear simulations are implemented entirely in MATLAB, user-defined control laws can be added in a straightforward fashion, and the simulations are portable across various computing platforms. Routines for trim, linearization, and numerical integration are included. The general nonlinear simulation framework and the specifics for each particular aircraft are documented.

14. SUBJECT TERMS
Real-time desktop nonlinear aircraft simulations

15. NUMBER OF PAGES
92

16. PRICE CODE

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