Fast Half-Loop Maneuvers for the F/A-18 Fighter Aircraft

Using a Singular Perturbation Feedback Control Law

by

Frederick Earl Garrett, Jr.

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Aerospace Engineering

APPROVED:

Dr. Harold L. Stalford, Chairman

Dr. Eugene M. Cliff

Dr. Frederick H. Lutze

September, 1988
Blacksburg, Virginia
c.2

LD
5655
V855
1988
G377
c.2
Fast Half-Loop Maneuvers for the F/A-18 Fighter Aircraft

Using a Singular Perturbation Feedback Control Law

by

Frederick Earl Garrett, Jr.

Dr. Harold L. Stalford, Chairman

Aerospace Engineering

(ABSTRACT)

The primary purpose of this study is to develop a nonlinear feedback control law for the F/A-18 fighter aircraft that performs a fast half-loop maneuver. This feedback law is developed using a singular perturbation approach. A secondary purpose of this study is to establish a baseline for time optimal half-loop maneuvers. The singular perturbation approach makes it possible to develop a state feedback control law which rotates the velocity vector through one hundred and eighty degrees at a maximum equilibrium pitch rate with a nearly constant angle of attack. The response of the aircraft to the control law is compared to simulations of half-loop maneuvers generated at NASA Langley Research Center.

For this study, only the longitudinal portion of the unconstrained half-loop maneuver is considered. That is, the one hundred and eighty degree roll executed at the end of the maneuver is not accounted for. The maneuver is unconstrained in the sense that pilot load factor limitations are not enforced in the development of the feedback control law. Pilot load factors are calculated to determine the regions in which pilot limitations are met or violated.
Our nonlinear feedback control law for agile half-loop maneuvers consists of three parts. A feedback subroutine of table look-up values is used to generate our feedback control law for simulation results. The middle portion is an outer layer solution in which the stabilator angle is a nonlinear function of Mach number, flight path angle and altitude. The angle of attack is nearly constant at the stall value (approximately 36 to 38 degrees). The pitch rate, which is a function of Mach number, flight path angle and altitude, is the state that maximizes the rate of change of the flight path angle while providing equilibrium to the angle of attack and the pitch rate differential equations of motion. This portion of the control law is defined by surfaces (i.e. table look-up values) for the outer layer values of angle of attack, pitch rate and stabilator angle. A data file, consisting of look-up tables for linear interpolation, is used by the feedback subroutine to determine the outer layer values for a given Mach number, flight path angle and altitude. The interpolation nodes are given in Tables 2, 3 and 4 in Appendix A. This data is valid for angles of attack from zero to ninety (0 to 90) degrees; Mach numbers from 0.05 to 0.90; flight path angles from zero to one hundred eighty (0 to 180, using redefined flight path angle) degrees; and altitudes from zero to sixty thousand (0 to 60,000) feet. The feedback subroutine uses these values to determine the outer layer values of the feedback control law (i.e. the stabilator angle of the outer layer). This subroutine also returns the proper control setting for the first and third portions of the feedback control law.

The first and third portions of the feedback control law are transition layer solutions which minimize the time to bring the aircraft from the initial trim values to the outer layer and from the outer layer to the final specified values, respectively. In the transition layers, the stabilator angle feedback control is bang-bang. We emphasize that our feedback control law is not a pure singular perturbation solution since we use the tran-
sition layers to minimize the time in transition to and from the outer layer. But, we do use singular perturbation analysis for all three parts of the control law.

In comparison to simulations of half-loop maneuvers done at NASA Langley Research Center, the feedback control law developed does provide a fast half-loop maneuver. Our maneuver requires about nine (9) seconds to execute the longitudinal portion of the half-loop maneuver for an initial Mach number of 0.9 and an initial altitude of 15,000 feet. The NASA maneuver takes about fifteen (15) seconds to execute the half-loop with the roll included. For initial conditions of Mach 0.6 and 15,000 feet, the time required is approximately thirteen (13) seconds. The NASA simulation of the maneuver (including the roll) took twenty-two (22) seconds for the same initial conditions. In both cases, the aircraft could not execute the half-loop maneuver for an initial Mach number of 0.3 at an altitude of 15,000 feet. This is due to the lack of energy available for this maneuver at 0.3 Mach number. The thrust-to-weight ratio for the F/A-18 aircraft at an altitude of 15,000 feet varies from 0.6 at Mach 0.3 to 0.8 at Mach 0.9.

However, the pilot load factors encountered in some cases exceed those allowable in practice. In the Mach 0.9 case, our maneuver exceeded the pilot load factor limitation of 7.5 g's in the first second. This limitation was not exceeded in the other two cases (i.e. Mach 0.3 and 0.6), but the onset of high load factors are of concern in these maneuvers. For example, the pilot load factor increased from zero (0) g's to approximately six (6) g's in less than one (1) second. In all cases the load factor decreased to low levels after the first second of the maneuver. In order to avoid the high load factors encountered, the first portion of the maneuver could be executed at lower pitch rates over a longer period of time. Therefore, in the regions where load factor limitations are not exceeded,
our feedback control law does produce a relatively fast half-loop maneuver that is useful in a practical sense and as a good baseline for optimal time studies.
Acknowledgements

I would like to thank the following people for their help: Dr. Harold L. Stalford for his tremendous help on this project and for his constant encouragement, even when the going got tough, I will always be indebted to him for his help and support; Dr. E.M.Cliff and Dr. F.H.Lutze for their suggestions and for serving on my committee; and Doug Arbuckle of NASA for making the F/A-18 model available to me for use in this project. This work was partially supported by NASA Langley Research Center under the NASA Contract number NAG-1-873.

I would also like to thank my wife, Beth, for putting up with me during the work on this project and for all the help she gave me. She spent many lonely evenings and many cold dinners by herself. But through it all, she gave a lot of love to me when I most needed it. My friends have also been very helpful and encouraging. Also, thanks to my parents, Mr. and Mrs. Frederick Garrett, Sr. for their unwavering confidence in me.

Finally, my biggest thanks go to the two people who made it possible for my wife and I to attend graduate school. They are my wife's grandparents, Mr. and Mrs. Hugh
Ussery. They gave us the use of their home and made sure that we had everything we needed. This country needs more people who believe in making an education possible and who back it up with all their heart. It is to them that this work is dedicated.
Table of Contents

1.0  Introduction ................................................................. 1
1.1  Unconstrained Half-Loop Maneuver .................................... 2
1.2  F/A-18 Fighter Escort Configuration ................................. 3

2.0  Problem Definition .......................................................... 4

3.0  Singular Perturbation Approach ......................................... 6

4.0  F/A - 18 Model ............................................................... 10

5.0  Outer Layer ................................................................. 23
5.1  Analysis ................................................................. 23
5.2  Results ................................................................. 29
5.3  Pole Placement ........................................................... 53

6.0  Transition Region Analysis ............................................... 59
6.1  First Part of the Transition Region Analysis ....................... 60
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Outer Layer Solution Nodes</td>
<td>30</td>
</tr>
<tr>
<td>2.</td>
<td>Stabilator Nodes (deg)</td>
<td>169</td>
</tr>
<tr>
<td>3.</td>
<td>Mach Number Nodes</td>
<td>170</td>
</tr>
<tr>
<td>4.</td>
<td>Angle of Attack Nodes (deg)</td>
<td>171</td>
</tr>
<tr>
<td>5.</td>
<td>Model Limits</td>
<td>172</td>
</tr>
</tbody>
</table>
List of Illustrations

Figure 1. Unconstrained Half-Loop Maneuver ................................. 5
Figure 2. Lift Coefficient (h = 15000 ft.) ................................. 14
Figure 3. Drag Coefficient (h = 15000 ft.) ................................. 15
Figure 4. Pitching Moment Coefficient (h = 15000 ft.) ..................... 16
Figure 5. Pitch Rate Derivative of Lift Coefficient (h = 15000 ft.) .......... 17
Figure 6. Pitch Rate Derivative of Pitching Moment Coefficient (h = 15000 ft.) 18
Figure 7. Angle of Attack Rate Derivative of Lift Coefficient (h = 15000 ft.) ... 19
Figure 8. Angle of Attack Rate Derivative of Pitching Moment Coefficient (h = 15000 ft.) ...................................................... 20
Figure 9. Thrust per Engine (h = 15000 ft.) ................................... 21
Figure 10. Thrust in x-body Direction (h = 15000 ft.) ......................... 22
Figure 11. Outer Layer Angle of Attack (h = 5000 ft) ......................... 32
Figure 12. Outer Layer Angle of Attack (h = 15000 ft) ......................... 33
Figure 13. Outer Layer Angle of Attack (h = 25000 ft) ......................... 34
Figure 14. Outer Layer Angle of Attack (h = 35000 ft) ......................... 35
Figure 15. Outer Layer Angle of Attack (h = 45000 ft) ......................... 36
Figure 16. Outer Layer Angle of Attack (h = 55000 ft) ......................... 37
Figure 17. Outer Layer Pitch Rate (h = 5000 ft) ................................ 38
Figure 18. Outer Layer Pitch Rate (h = 15000 ft) ............................... 39
Figure 19. Outer Layer Pitch Rate (h = 25000 ft) ............................... 40
Figure 45. Half-Loop Simulation: Pole Placement Gain (M(t = 0) = 0.9, h = 15000 ft) 90
Figure 46. Half-Loop Simulation: Pole Placement Gain (M(t = 0) = 0.9, h = 15000 ft) 91
Figure 47. Half-Loop Simulation: Angle of Attack (M(t = 0) = 0.6, h = 15000 ft) ... 92
Figure 48. Half-Loop Simulation: Pitch Rate (M(t = 0) = 0.6, h = 15000 ft) ........ 93
Figure 49. Half-Loop Simulation: Airspeed (M(t = 0) = 0.6, h = 15000 ft) ........ 94
Figure 50. Half-Loop Simulation: Mach Number (M(t = 0) = 0.6, h = 15000 ft) ... 95
Figure 51. Half-Loop Simulation: Flight Path Angle (M(t = 0) = 0.6, h = 15000 ft) 96
Figure 52. Half-Loop Simulation: Stabilator Angle (M(t = 0) = 0.6, h = 15000 ft) .. 97
Figure 53. Half-Loop Simulation: Altitude vs Range (M(t = 0) = 0.6, h = 15000 ft) 98
Figure 54. Half-Loop Simulation: Load Factor (M(t = 0) = 0.6, h = 15000 ft) .... 99
Figure 55. Half-Loop Simulation: Load Factor (M(t = 0) = 0.6, h = 15000 ft) ... 100
Figure 56. Half-Loop Simulation: Dynamic Pressure (M(t = 0) = 0.6, h = 15000 ft) 101
Figure 57. Half-Loop Simulation: Fast System Eigenvalues (M(t = 0) = 0.6, h = 15000 ft) ........................................ 102
Figure 58. Half-Loop Simulation: Pole Placement Gain (M(t = 0) = 0.6, h = 15000 ft) ........................................ 103
Figure 59. Half-Loop Simulation: Pole Placement Gain (M(t = 0) = 0.6, h = 15000 ft) ........................................ 104
Figure 60. Half-Loop Simulation: Angle of Attack (M(t = 0) = 0.3, h = 15000 ft) .. 105
Figure 61. Half-Loop Simulation: Pitch Rate (M(t = 0) = 0.3, h = 15000 ft) .... 106
Figure 62. Half-Loop Simulation: Airspeed (M(t = 0) = 0.3, h = 15000 ft) .... 107
Figure 63. Half-Loop Simulation: Mach Number (M(t = 0) = 0.3, h = 15000 ft) .. 108
Figure 64. Half-Loop Simulation: Flight Path Angle (M(t = 0) = 0.3, h = 15000 ft) 109
Figure 65. Half-Loop Simulation: Stabilator Angle (M(t = 0) = 0.3, h = 15000 ft) .. 110
Figure 66. Half-Loop Simulation: Altitude vs Range (M(t = 0) = 0.3, h = 15000 ft) 111
Figure 67. Half-Loop Simulation: Load Factor (M(t = 0) = 0.3, h = 15000 ft) ... 112
Figure 68. Half-Loop Simulation: Load Factor (M(t = 0) = 0.3, h = 15000 ft) ... 113
Figure 69. Half-Loop Simulation: Dynamic Pressure \((M(t=0)=0.3, h=15000 \text{ ft})\) 114

Figure 70. Half-Loop Simulation: Fast System Eigenvalues \((M(t=0)=0.3, h=15000 \text{ ft})\) .......................... 115

Figure 71. Half-Loop Simulation: Pole Placement Gain \((M(t=0)=0.3, h=15000 \text{ ft})\) ................................. 116

Figure 72. Half-Loop Simulation: Pole Placement Gain \((M(t=0)=0.3, h=15000 \text{ ft})\) ................................. 117

Figure 73. Comparison with NASA: Angle of Attack \((M(t=0)=0.9, h=15000 \text{ ft})\) 122

Figure 74. Comparison with NASA: Pitch Rate \((M(t=0)=0.9, h=15000 \text{ ft})\) .......................... 123

Figure 75. Comparison with NASA: Airspeed \((M(t=0)=0.9, h=15000 \text{ ft})\) .......................... 124

Figure 76. Comparison with NASA: Mach Number \((M(t=0)=0.9, h=15000 \text{ ft})\) 125

Figure 77. Comparison with NASA: Pitch Angle \((M(t=0)=0.9, h=15000 \text{ ft})\) .......................... 126

Figure 78. Comparison with NASA: Stabilator Angle \((M(t=0)=0.9, h=15000 \text{ ft})\) 127

Figure 79. Comparison with NASA: Altitude \((M(t=0)=0.9, h=15000 \text{ ft})\) .......................... 128

Figure 80. Comparison with NASA: Load Factor \((M(t=0)=0.9, h=15000 \text{ ft})\) .......................... 129

Figure 81. Comparison with NASA: Angle of Attack \((M(t=0)=0.6, h=15000 \text{ ft})\) 130

Figure 82. Comparison with NASA: Pitch Rate \((M(t=0)=0.6, h=15000 \text{ ft})\) .......................... 131

Figure 83. Comparison with NASA: Airspeed \((M(t=0)=0.6, h=15000 \text{ ft})\) .......................... 132

Figure 84. Comparison with NASA: Mach Number \((M(t=0)=0.6, h=15000 \text{ ft})\) .......................... 133

Figure 85. Comparison with NASA: Pitch Angle \((M(t=0)=0.6, h=15000 \text{ ft})\) .......................... 134

Figure 86. Comparison with NASA: Stabilator Angle \((M(t=0)=0.6, h=15000 \text{ ft})\) 135

Figure 87. Comparison with NASA: Altitude \((M(t=0)=0.6, h=15000 \text{ ft})\) .......................... 136

Figure 88. Comparison with NASA: Load Factor \((M(t=0)=0.6, h=15000 \text{ ft})\) .......................... 137

Figure 89. Comparison with NASA: Angle of Attack \((M(t=0)=0.3, h=15000 \text{ ft})\) 138

Figure 90. Comparison with NASA: Pitch Rate \((M(t=0)=0.3, h=15000 \text{ ft})\) .......................... 139

Figure 91. Comparison with NASA: Airspeed \((M(t=0)=0.3, h=15000 \text{ ft})\) .......................... 140

Figure 92. Comparison with NASA: Mach Number \((M(t=0)=0.3, h=15000 \text{ ft})\) 141
Figure 93. Comparison with NASA: Pitch Angle (M(t = 0) = 0.3, h = 15000 ft) . . . 142
Figure 94. Comparison with NASA: Stabilator Angle (M(t = 0) = 0.3, h = 15000 ft) 143
Figure 95. Comparison with NASA: Altitude (M(t = 0) = 0.3, h = 15000 ft) . . . . 144
Figure 96. Comparison with NASA: Load Factor (M(t = 0) = 0.3, h = 15000 ft) . . 145
Figure 97. Thrust-to-Weight Ratio (M(t = 0) = 0.9, h = 15000 ft) .................. 147
Figure 98. Thrust-to-Weight Ratio (M(t = 0) = 0.6, h = 15000 ft) .................. 148
Figure 99. Thrust-to-Weight Ratio (M(t = 0) = 0.3, h = 15000 ft) ................. 149
List of Symbols

\( \alpha \)  
Angle of Attack

\( q \)  
Pitch Rate

\( g \)  
Gravitational Acceleration

\( v \)  
Total Velocity

\( \gamma \)  
Flight Path Angle

\( \bar{q} \)  
Dynamic Pressure

\( S \)  
Wing Area

\( m \)  
Mass

\( C_L \)  
Lift Coefficient

\( T_x \)  
Thrust in x-body direction

\( T_s \)  
Total Thrust

\( C_D \)  
Drag Coefficient

\( I_y \)  
Moment of Inertia about y-axis

\( l_s \)  
Thrust Offset from x-axis

\( l_x \)  
Distance along x-axis from center of gravity to the aerodynamic center

\( l_z \)  
Distance along z-axis from center of gravity to the aerodynamic center
$\theta$ Pitch angle
$h$ Altitude
$d_L$ Lift Offset
$d_D$ Drag Offset
$W$ Weight
$\bar{c}$ Mean Aerodynamic Chord
$b$ Wing Span
$\bar{x}$ Distance along x-axis from center of gravity to the pilot
$\bar{z}$ Distance along z-axis from center of gravity to the pilot
$\delta_h$ Stabilator Deflection
$\delta_{lf}$ Leading Edge Flap Deflection
$\delta_{tf}$ Trailing Edge Flap Deflection
$\delta_T$ Throttle Setting
$n_x$ Load Factor (x-direction)
$n_z$ Load Factor (z-direction)
$n_{xp}$ Pilot Load Factor (x-direction)
$n_{zp}$ Pilot Load Factor (z-direction)
$\varepsilon$ Perturbation Parameter
$J_{ol}$ Outer Layer Performance Index
$\Delta J_x$ First Inner Layer Performance Index
$\Delta J_z$ Second Inner Layer Performance Index
$t$ Time
$M$ Mach Number
$\mathcal{H}$ Hamiltonian
$f_{\alpha}$ Differential Equation for Angle of Attack
$f_{\dot{\theta}}$ Differential Equation for Pitch Rate
\( A \) System Matrix for the Fast Outer Layer Variables
\( B \) Input Matrix for the Fast Outer Layer Variables
\( p_{OL} \) Open-Loop Characteristic Equation
\( p_{CL} \) Closed-Loop Characteristic Equation
\( U \) Controllability Matrix
\( a_1 \) First Order Coefficient of the Open-Loop Characteristic Equation
\( a_2 \) Zeroth Order Coefficient of the Open-Loop Characteristic Equation
\( a_1 \) First Order Coefficient of the Closed-Loop Characteristic Equation
\( a_2 \) Zeroth Order Coefficient of the Closed-Loop Characteristic Equation
\( \lambda \) Open-Loop Eigenvalues
\( \tilde{\lambda} \) Closed-Loop Eigenvalues
\( \omega \) Circular Frequency of the Butterworth Poles
\( W \) Lower Toeplitz Matrix
\( K \) Gain Matrix
\( \Delta \alpha \) Change in Angle of Attack
\( \Delta \gamma \) Change in Flight Path Angle
\( \Delta t \) Time required to bring Angle of Attack and Pitch Rate to desired values
1.0 Introduction

We desire to develop a nonlinear state feedback control law that is time optimal for the half-loop maneuver of the F/A-18 aircraft. However, there is a trade-off between constructing a nonlinear feedback control law and constructing one that is optimal. Time optimal control laws are usually open loop. Therefore, to construct an optimal feedback control law, we generally have to synthesize it from a family of optimal open loop control histories. The construction of an optimal feedback law for high angle of attack flight can be extremely difficult, if not impossible, and is therefore beyond the scope of this work. In order to establish a good baseline for a time optimal study, a suboptimal (i.e. fast) feedback control law for the longitudinal portion of a fast half-loop maneuver is developed. The response of the aircraft to the fast feedback control law can then be compared with the results of any time optimal solution obtained in the future. The tradeoff referred to here is thus the relative ease in which a suboptimal feedback control law can be constructed as compared with that obtained by synthesizing optimal open loop controls. The approach used to develop our fast feedback control law is singular perturbation theory. This method is similar to a study done for the T-2C aircraft by
Stalford and Hoffman (Ref. 12). The motivation and details of this approach are discussed in chapter 3.

\section{1.1 Unconstrained Half-Loop Maneuver}

The half-loop maneuver consists of a pitch-up in which the aircraft reverses its direction in the vertical plane (see Figure 1). This maneuver is almost entirely executed in the plane of symmetry of the aircraft except for the one hundred and eighty degree roll executed at the end of the maneuver. The roll is considered to be outside the scope of our work. The time to perform the roll, which can be executed rather quickly, is not considered herein.

This maneuver is particularly useful in head-on pass engagements at high Mach numbers when the opponent is at a higher altitude. Major limitations to the maneuver are the high load factors encountered at high Mach numbers and the large amount of thrust required to perform it. For high Mach numbers and low altitudes, the load factor may be more than that allowable for a pilot. A useable feedback law would necessarily have to take pilot limitations into account and constrain the aircraft response within allowable load factor limits.

In this study, a load factor constraint is not used in determining the feedback law. The maneuver being considered is therefore called the "unconstrained half-loop maneuver". However, the load factor is computed in order to expose the regions in which load factor limits are violated. In those regions where the load factor limit is not violated, the un-
constrained feedback law will be identical to the constrained one. The load factor resulting from using the feedback law is a function of Mach number and altitude. This means that there will be a region of Mach numbers and altitudes in which the unconstrained solution will satisfy pilot load factor limits.

1.2 F/A-18 Fighter Escort Configuration

The version of the F/A-18 used for this study is a single-seat aircraft used as an escort fighter. It is meant primarily as a replacement for the F-4 fighter, A-4 and A-7 attack aircrafts (Ref. 7). The escort configuration is assumed to have 60 percent fuel with tip missiles attached. The flight control system consists of horizontal stabilators, ailerons, two twin fin rudders, full span leading-edge flaps and single slot trailing-edge flaps (Ref. 7). For this study, the ailerons and rudders are not used. The leading-edge and trailing-edge flaps are used according to a flap schedule supplied by NASA Langley Research Center. Actuator dynamics and differential control surface deflections are not considered.
2.0 Problem Definition

We desire to design a nonlinear feedback law which will execute, relatively fast, the unconstrained half-loop maneuver. Our work will also establish a baseline for future research work on time optimal controls for this maneuver. The study will only encompass the longitudinal portion of the maneuver for subsonic initial Mach numbers.

A graphical representation of the half-loop maneuver is presented in Figure 1. The aircraft begins the maneuver in straight and level flight. The velocity vector is then rotated through one hundred and eighty degrees flight path angle. It is then necessary to bring the aircraft back to straight and level flight as shown. A one hundred and eighty degree roll can then be executed, at the end, to complete the maneuver.

Stalford and Hoffman (Ref. 12) worked on a similar problem for the T-2C trainer aircraft, but only for the special constant speed case. Their research involved developing a feedback control law that would point the aircraft according to a commanded pitch angle. Their maneuver is very similar in nature to our half-loop maneuver.
Figure 1. Unconstrained Half-Loop Maneuver
3.0 Singular Perturbation Approach

A singular perturbation approach is used to derive a nonlinear feedback law for the half-loop maneuver of the F/A-18 fighter aircraft. The approach is similar to that used in Stalford and Hoffman (Ref. 12). The problem is solved using several time scales or layers. First, in the outer layer solution the differential equations of angle of attack and pitch rate are reduced to algebraic equations making it possible to hold the states of angle of attack and pitch rate nearly constant on a particular time scale. The algebraic equations obtained are used to derive a feedback law which is valid under these conditions. The resulting stabilator feedback control law is a function of the remaining non-constant states of Mach number, altitude and flight path angle. This approach makes it possible to develop a nonlinear state feedback control law which rotates the velocity vector through one hundred and eighty degrees with nearly constant angle of attack and maximum equilibrium pitch rate. The resulting feedback control law is a function of flight path angle, Mach number and altitude.

The nonlinear feedback law of the half-loop maneuver is derived using three parts; an outer layer and two transition regions. On the outer layer, the angle of attack and pitch
rate are assumed to be fast compared to velocity, flight path angle and altitude. Angle of attack and pitch rate are assumed to be fast because a change in control has a direct effect on the corresponding differential equations of motion. Similarly, a change in control has little direct effect on the flight path angle, Mach number or altitude equations of motion. The changes in these slow states are a consequence of changes in the values of the fast states, angle of attack and pitch rate. In the transition region analysis, to be discussed later, a similar process is applied to determine that the pitch rate is fast compared to the angle of attack.

In order for the outer layer of this maneuver to be executed in "minimum time", we observe that the integrand of the performance criterion is minimized by maximizing the pitch rate. This follows since the angle of attack is being held constant. The right hand sides of the two differential equations governing the angle of attack and the pitch rate variables are set to zero, producing two algebraic equations. That is, the angle of attack and the pitch rate variables are assumed constant. The feedback law for the outer layer is found by solving these two algebraic equations for the values of the angle of attack and the stabilator angle which maximize the pitch rate. The resulting nonlinear control law consists of a set of surfaces for each altitude. These surfaces represent the variables angle of attack, pitch rate and stabilator deflection as a function of flight path angle and Mach number. The surface representing the stabilator deflection is the feedback control law for the outer layer. The surfaces which represent the angle of attack and the pitch rate define the values that these states must necessarily possess in order that the aircraft be in the outer layer. The outer layer solution is thus defined.

The next part in deriving a nonlinear feedback control law is to get the states from the initial conditions to the outer layer in minimum time (i.e. the first transition region).
The last part is the transition from the outer layer to a prescribed terminal state (i.e. the second transition region. This is also to be performed in minimum time. The control of these transitions, together with that of the outer layer, form our feedback control law. In the first transition region, the angle of attack and the pitch rate are again taken to be the fast variables as compared to Mach number, altitude and flight path angle. First, the pitch rate is considered to be fast compared to the angle of attack. The control law which maximizes the time rate of change of angle of attack is found first; then, the time rate of change of pitch rate is maximized to complete the analysis of the first transition region.

The overall feedback law is therefore determined by dividing the maneuver into the three portions discussed above. First there is the outer layer region, where the outer layer stabilator surface is used as a feedback law. Then there are the two transition regions at the beginning and end of the outer layer. The transition regions comprise the transfer of the states from the initial conditions to the outer layer and the transfer from the outer layer to the terminal state at the end of the maneuver. The control laws for the transition regions are derived from singular perturbation analysis.

We emphasize that our entire feedback control law is not a pure singular perturbation solution since in the transition regions, we are using singular perturbation analysis to minimize the time to transfer the aircraft to and from the outer layer. A pure singular perturbation solution would be more complex in the transition layers. A control law obtained by developing inner layer solutions and asymptotically matching these solutions to the derived outer layer (Ref. 3) would be very complicated and the result would be a degradation of time optimality in performing the transition regions. Our feedback control for the transition regions is bang-bang and appears to be nearly time optimal.
We do use singular perturbation analysis to derive the bang-bang control laws. The work done by Stalford and Hoffman (Ref. 12 and Ref. 15) for the T-2C aircraft gives us reason to suspect that our bang-bang control laws for the transition regions are nearly time optimal. That is, the results obtained in Stalford and Hoffman (Ref. 15) using the necessary conditions of optimal control (i.e. the maximum principle) are identical with those obtained in Stalford and Hoffman (Ref. 12) using the singular perturbation analysis in the transition regions.
4.0 F/A - 18 Model

The F/A - 18 Model is based on the NASA Langley Research Center Model described in Appendix A. Modifications made to the NASA Model are given in Appendix B. These modifications are necessary in order to create a smooth and continuous model. This was necessary because the NASA model is based on two different sets of data. One set of data is from wind tunnel tests for angles of attack less than forty degrees and the other set is for angles of attack greater than forty degrees. These two tests were done using different flap settings and thus the model has a discontinuity at forty degrees angle of attack. It is therefore necessary to include a flap schedule which does away with this discontinuity. Unfortunately, the flap schedule does not alleviate the discontinuity for all Mach numbers and altitudes. Therefore, it is necessary to add a correction factor for high Mach numbers and low altitudes which gives us a smooth model. The equations of motion for longitudinal flight are presented in Appendix C and are given as follows:

\[ \dot{q} = q + \frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} C_L - \frac{T_x}{mV} \sin \alpha \quad (4.1) \]

\[ \dot{V} = -g \sin \gamma - \frac{\bar{q}S}{m} C_D + \frac{T_x}{m} \cos \alpha \quad (4.2) \]
\[ \dot{q} = \frac{1}{I_y} \left[ l_x T_x + \bar{q} S (\bar{c} C_m + d_L C_L + d_D C_D) \right] \quad (4.3) \]

\[ \dot{\theta} = q \quad (4.4a) \]

\[ \dot{\gamma} = q - \dot{\alpha} \quad (4.4b) \]

\[ \dot{h} = V \sin \gamma \quad (4.5) \]

where:

\[ d_L(\alpha) = l_x \cos \alpha + l_z \sin \alpha \quad (4.6) \]

\[ d_D(\alpha) = l_x \sin \alpha - l_z \cos \alpha \quad (4.7) \]

\[ \gamma = \theta - \alpha \quad (4.8) \]

The state variables are angle of attack, pitch rate, Mach number, flight path angle and altitude. That is, for analysis purposes, we use (4.4b) instead of (4.4a) since pitch angle, \( \theta \), and flight path angle, \( \gamma \) are related by equation (4.8). Importantly, we use (4.4b) instead of (4.4a) since angle of attack, \( \alpha \) is fast with respect to \( \gamma \) but not with respect to \( \theta \). For the F/A - 18 fighter escort configuration, the constants in the above equations are:

- \( I_y = 151,293 \text{ slug ft}^2 \)
- \( W = 33,310 \text{ lb} \)
- \( m = 1034.47 \text{ slugs} \)
- \( \bar{c} = 11.52 \text{ ft} \)
b = 37.42 ft
S = 400.0 ft²
l_s = 2.8 in
l_ι = -3.56 in
l_t = 2.8 in
g = 32.174 ft/s²
x̄ = 18 ft
z̄ = 6 ft

The control surface limits are:
-24° ≤ δ_h ≤ 10.5°
-3° ≤ δ_wf ≤ 34°
-8° ≤ δ_wf ≤ 45°

The actuator no load rate limits are:
Stabilator, δ_h 40°/s
Leading edge flap, δ_wf 18°/s
Trailing edge flap, δ_wf 18°/s

Note: Actuator dynamics are not used in this study, these limits are given as a reference.

Thrust is a function of altitude, Mach number, and throttle setting. The model is made up of look-up tables which are used by a subroutine to calculate the thrust per engine, T_s. The total thrust in the x-body direction is found by doubling the thrust per engine and taking into account the angle of the engines with respect to the x-body axis.

\[ T_x = 2T_s \cos(1.98°) \]
\[ T_s = f(M, h, \delta_T) \]
where:

\[
\delta_T = \begin{bmatrix}
30^\circ & \text{idle} \\
106.5^\circ & \text{military} \\
130^\circ & \text{full afterburner}
\end{bmatrix}
\]

The thrust in the y-body direction \( T_y \) is zero because the engines are mounted in such a way that the y-body components of thrust for each engine cancel each other out.

The pilot load factors are calculated as follows

\[
n_{x_p} = \frac{1}{g} \left[ \frac{\bar{q}S}{m} \left( (C_{L_0} + C_{L_\alpha}q + C_{Lq}) \sin \alpha - C_{D_0} \cos \alpha \right) - \bar{x}q^2 + \ddot{x} \dot{q} \right]
\]

(4.9)

\[
n_{z_p} = \frac{1}{g} \left[ \frac{\bar{q}S}{m} \left( (C_{L_0} + C_{L_\alpha}q + C_{Lq}) \cos \alpha + C_{D_0} \sin \alpha \right) \right] + \bar{x} \dot{q} + \ddot{x} q^2
\]

(4.10)

Figures 2-8 show the aerodynamic coefficients for an altitude of 15,000 feet. Figures 9 and 10 show the thrust per engine and the total thrust in the x-body direction respectively for an altitude of 15,000 feet.
Figure 2. Lift Coefficient (h = 15000 ft.)
Figure 3. Drag Coefficient (h = 15000 ft.)
Figure 4. Pitching Moment Coefficient (h = 15000 ft.)
Figure 5. Pitch Rate Derivative of Lift Coefficient (h = 15000 ft.)
Figure 6. Pitch Rate Derivative of Pitching Moment Coefficient (h = 15000 ft.)
Figure 7. Angle of Attack Rate Derivative of Lift Coefficient (h = 15000 ft.)
Figure 8. Angle of Attack Rate Derivative of Pitching Moment Coefficient (h = 15000 ft.)
Figure 9. Thrust per Engine (h = 15000 ft.)
Figure 10. Thrust in x-body Direction (h = 15000 ft.)
5.0 Outer Layer

5.1 Analysis

The outer layer portion of the maneuver consists of the rotation of the velocity vector through $180^\circ$ in flight path angle, assuming the angle of attack and the pitch rate are fast variables compared to Mach number, flight path angle, and altitude. By this assumption, the equations of motion become Eq. (5.1) through Eq. (5.5) for small values of $\varepsilon$.

\begin{equation}
\varepsilon \dot{a} = q + \frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} c_L - \frac{T_x}{mV} \sin \alpha \tag{5.1}
\end{equation}

\begin{equation}
\dot{V} = -g \sin \gamma - \frac{\bar{q}S}{m} C_D + \frac{T_x}{m} \cos \alpha \tag{5.2}
\end{equation}

\begin{equation}
\varepsilon \dot{q} = \frac{1}{I_y} [I_x \dot{T}_x + \bar{q}S(\bar{\epsilon} C_m + d_L C_L + d_D C_D)] \tag{5.3}
\end{equation}

\begin{equation}
\dot{\theta} = q \tag{5.4a}
\end{equation}
\dot{y} = q - \dot{\alpha} \quad (5.4b)

\dot{h} = V \sin \gamma \quad (5.5)

We seek to minimize the time required to execute the outer layer. The outer layer performance index \( J_{OL} \) is given by

\[ J_{OL} = {t_f}^{OL} - {t_0}^{OL} = \int_{t_0}^{t_f} \dot{y} \, dt \] \quad (5.6)

\[ y = \theta - \alpha \] \quad (5.7)

\[ \dot{y} = \dot{\theta} - \dot{\alpha} \] \quad (5.8)

Therefore, for nearly constant angle of attack, \( \alpha = 0 \)

\[ \dot{y} = \frac{dy}{dt} = \dot{\theta} = q \] \quad (5.9)

Therefore,

\[ dt = \frac{dy}{q} \] \quad (5.10)

\[ J_{OL} = \int_{\gamma(t_0)}^{\gamma(t_f)} \frac{dy}{q} \] \quad (5.11)
This equation indicates that the integrand of the outer layer performance index is minimized when the instantaneous pitch rate is maximized. Therefore, the values of the angle of attack and the stabilator deflection for the outer layer are those which maximize pitch rate subject to Eq. (5.1) - (5.5) as $\varepsilon$ approaches zero.

\[
0 = q + \frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} - \frac{T_x}{mV} \sin \alpha \quad (5.12)
\]

\[
\dot{V} = -g \sin \gamma - \frac{\bar{q}S}{m} C_D + \frac{T_x}{m} \cos \alpha \quad (5.13)
\]

\[
0 = \frac{1}{I_y} \left[ I_x T_x + \bar{q}S(\bar{C}_m + d_L C_L + d_D C_D) \right] \quad (5.14)
\]

\[
\dot{\theta} = q \quad (5.15)
\]

\[
\dot{h} = V \sin \gamma \quad (5.16)
\]

Therefore by maximizing the pitch rate in Eq. (5.12) and Eq. (5.14), a feedback control law is derived as a function of the slow states which are flight path angle, Mach number and altitude. For a given altitude, this control law consists of a surface defined by the stabilator deflection for each flight path angle and Mach number. Surfaces for the fast variables that maximize pitch rate are determined as well.

\[
q^*(\gamma, M, h) = \max_{\alpha, \delta_h} \ q(\alpha, \delta_h, \gamma, M, h) \quad (5.17)
\]

$\alpha'(\gamma, M, h)$ and $\delta_h^*(\gamma, M, h)$ are those arguments in Eq. (5.17) which maximize pitch rate subject to equations (5.12) and (5.14). Our numerical procedure generates unique maximizing arguments.
In order to find these surfaces, parameter optimization is used. This is done using equations (5.12), (5.14) and (5.17). First, the aerodynamic coefficients are rewritten in a form which explicitly shows their dependence on pitch rate.

\[ C_L(\alpha, \delta_h, M, q) = C_{L_0}(\alpha, \delta_h, M) + C_{L_0}(\alpha, M)q \] (5.18)

\[ C_m(\alpha, \delta_h, M, q) = C_{m_0}(\alpha, \delta_h, M) + C_{m_0}(\alpha, M)q \] (5.19)

\[ C_D(\alpha, \delta_h, M) = C_{D_0}(\alpha, \delta_h, M) \] (5.20)

Equation (5.12) becomes

\[ 0 = q + \frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} (C_{L_0} + C_{L_0}q) - \frac{T_x}{mV} \sin \alpha \] (5.21)

and equation (5.14) becomes

\[ 0 = \frac{1}{I_p} \left[ l_z T_x + \bar{q}S \left( C_{m_0} + C_{m_0}q \right) + d_L \left( C_{L_0} + C_{L_0}q \right) + d_D C_{D_0} \right] \]

\[ = \phi (\alpha, \delta_h, \gamma, M, h, q) \] (5.22)

Then, solving equation (5.21) for \( q \) yields

\[ q(\alpha, \delta_h, \gamma, M, h) = \frac{\frac{g}{V} \cos \gamma + \frac{\bar{q}S}{mV} C_{L_0} + \frac{T_x}{mV} \sin \alpha}{1 - \frac{\bar{q}S}{mV} C_{L_0}} \] (5.23)

For a given flight path angle, Mach number and altitude, the problem is now to maximize equation (5.23) subject to equation (5.22) over the domain \((\alpha, \delta_h)\).
subject to \( f(\alpha, \delta, \gamma, M, h, q) = 0 \)

or

\[
\min_{\alpha, \delta, \gamma, M, h} -q(\alpha, \delta, \gamma, M, h)
\]

subject to \( f(\alpha, \delta, \gamma, M, h, q) = 0 \)

This problem is solved by defining a Hamiltonian

\[
\mathcal{H}(\alpha, \delta, \gamma, M, h) = -q(\alpha, \delta, \gamma, M, h) + \lambda f(\alpha, \delta, \gamma, M, h, q) \tag{5.24}
\]

And applying the necessary conditions for a stationary point on \(-q\):

\[
f(\alpha^*, \delta^*, q^*, \gamma, M, h) = 0 \tag{5.25}
\]

\[
\frac{d\mathcal{H}}{d\alpha}(\alpha^*, \delta^*, q^*, \gamma, M, h) = 0 \tag{5.26}
\]

\[
\frac{d\mathcal{H}}{d\delta h}(\alpha^*, \delta^*, q^*, \gamma, M, h) = 0 \tag{5.27}
\]

Substituting equation (5.24) into equations (5.26) and (5.27) it follows that:

\[
f(\alpha^*, \delta^*, q^*, \gamma, M, h) = 0 \tag{5.28}
\]

\[
-\frac{dq}{d\alpha}(\alpha^*, \delta^*, q^*, \gamma, M, h) + \lambda \frac{df}{d\alpha}(\alpha^*, \delta^*, q^*, \gamma, M, h) = 0 \tag{5.29}
\]

\[
-\frac{dq}{d\delta h}(\alpha^*, \delta^*, q^*, \gamma, M, h) + \lambda \frac{df}{d\delta h}(\alpha^*, \delta^*, q^*, \gamma, M, h) = 0 \tag{5.30}
\]

where:

Outer Layer
\[ \frac{dq}{d\alpha} = \frac{\bar{q} S}{m V} \frac{dC_{L_0}}{d\alpha} + \frac{T_x}{m V} \cos \alpha \nonumber \]

\[ + \frac{1}{1 - \frac{\bar{q} S}{m V} C_{L_q}} \left( -\frac{g}{V} \cos \gamma + \frac{\bar{q} S}{m V} C_{L_0} + \frac{T_x}{m V} \sin \alpha \right) \left( \frac{\bar{q} S}{m V} \frac{dC_{L_q}}{d\alpha} \right) \]  \tag{5.31} 

\[ \frac{dq}{d\delta_h} = \frac{\bar{q} S}{m V} \frac{dC_{L_0}}{d\delta_h} \nonumber \]

\[ \frac{1}{1 - \frac{\bar{q} S}{m V} C_{L_q}} \]  \tag{5.32} 

\[ \frac{df}{d\alpha} = \frac{\bar{q} S}{I_y} \left[ \bar{e} C_{m_0} + d_L \left( C_{d_0} + \frac{dC_{L_0}}{d\alpha} \right) + d_p \left( \frac{dc_{D_0}}{d\alpha} - C_{L_0} \right) \right. \nonumber \]

\[ \left. + \left( \bar{e} \frac{dc_{m_0}}{d\alpha} + d_L \frac{dc_{L_0}}{d\alpha} - d_D C_{L_q} \right) q + (\bar{e} C_{m_q} + d_L C_{L_q}) \frac{dq}{d\alpha} \right] \]  \tag{5.33} 

\[ \frac{df}{d\delta_h} = \frac{\bar{q} S}{I_y} \left[ \bar{e} \frac{dc_{m_0}}{d\delta_h} + d_L \frac{dc_{L_0}}{d\delta_h} + d_D \frac{dc_{D_q}}{d\delta_h} \right] \]  \tag{5.34} 

The solution to this problem is a set of three surfaces for each altitude considered. These surfaces define the control setting (\(\delta_h\)), the angle of attack (\(\alpha\)) and the pitch rate (\(q\)), for a given flight path angle and Mach number, which minimize the outer layer performance index \((J_{ol})\).
5.2 Results

The problem, represented by equation (5.17) in Section 1, is solved numerically using the computer program "SLLSQP" (Sequential Linear Least SQuares Programming) written by Dieter Kraft and Klaus Schittkowski (Ref. 10). We checked this solution for uniqueness and optimality using a global grid method. We selected an initial grid spanning angles of attack from zero (0) to ninety (90) degrees and stabilator angles from negative twenty-four (-24) to ten and one-half (10.5) degrees using one (1) degree intervals for angle of attack and one half (0.5) degree intervals for the stabilator. For a selection of flight path angles, Mach numbers and altitudes corresponding to Table 1, this grid of angles of attack and stabilator angles in the domain of the F/A-18 model was processed. The pitch rate corresponding to each node on this grid was calculated using equation (5.23). The node corresponding to the maximum pitch rate on this grid was found by sorting the pitch rates found at each node on the grid. This node became the center node on a finer grid which was generated as before. The node corresponding to the maximum pitch rate was again found by sorting. This process was repeated until the grid size was suitable for finding the solution to an acceptable number of significant digits. The angle of attack and stabilator angle corresponding to the node yielding the maximum pitch rate was compared to the solution using SLLSQP. We found that the result obtained using SLLSQP was indeed unique and provided the optimal pitch rate. The outer layer surfaces are generated for the Mach numbers, flight path angles and altitudes given in Table 1. We use nineteen (19) nodes for flight path angle, six (6) nodes for Mach number and six (6) nodes for altitude.
<table>
<thead>
<tr>
<th>$\gamma$ (deg)</th>
<th>$M$</th>
<th>$h$ (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.2</td>
<td>5000</td>
</tr>
<tr>
<td>10</td>
<td>0.3</td>
<td>15000</td>
</tr>
<tr>
<td>20</td>
<td>0.4</td>
<td>25000</td>
</tr>
<tr>
<td>30</td>
<td>0.5</td>
<td>35000</td>
</tr>
<tr>
<td>40</td>
<td>0.6</td>
<td>45000</td>
</tr>
<tr>
<td>50</td>
<td>0.7</td>
<td>55000</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td></td>
<td></td>
</tr>
<tr>
<td>180</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The values corresponding to these nodes are stored in a file named "HALFLOOP_OL.DAT". The feedback subroutine reads these data into memory upon initialization. A three-dimensional linear interpolation is performed in order to calculate values not located on the nodes.

If initialization is done with outer layer values for the angle of attack and pitch rate, the outer layer portion of the maneuver can be simulated. Figures 29-31 show the results of the simulation for an initial Mach number of 0.6 and an altitude of 15,000 feet.
Figure 11. Outer Layer Angle of Attack (h = 5000 ft)
Figure 12. Outer Layer Angle of Attack ($h = 15000$ ft)
Figure 13. Outer Layer Angle of Attack ($h = 25000$ ft)
Figure 14. Outer Layer Angle of Attack (h = 35000 ft)
Figure 15. Outer Layer Angle of Attack ($h = 45000$ ft)
Figure 16. Outer Layer Angle of Attack (h = 55000 ft)
Figure 17. Outer Layer Pitch Rate (h = 5000 ft)
Figure 18. Outer Layer Pitch Rate (h = 15000 ft)
Figure 19. Outer Layer Pitch Rate (h = 25000 ft)
Figure 20. Outer Layer Pitch Rate (h = 35000 ft)
Figure 21. Outer Layer Pitch Rate (h = 45000 ft)
Figure 22. Outer Layer Pitch Rate (h = 55000 ft)
Figure 23. Outer Layer Stabilator Deflection (h = 5000 ft)
Figure 24. Outer Layer Stabilator Deflection (h = 15000 ft)
Figure 25. Outer Layer Stabilator Deflection (h = 25000 ft)
Figure 26. Outer Layer Stabilator Deflection (h = 35000 ft)
Figure 27. Outer Layer Stabilator Deflection (h = 45000 ft)
Figure 28.  Outer Layer Stabilator Deflection (h = 55000 ft)
Figure 29. Simulation of Outer Layer ($M = 0.6, h = 15000 \text{ R}$)
Figure 30. Simulation of Outer Layer ($M = 0.6, h = 15000$ ft)
Figure 31. Simulation of Outer Layer (M = 0.6, h = 15000 ft)
5.3 Pole Placement

We observe, from Figures 29 and 30, that the actual states in the simulation of the feedback law deviate from the outer layer values of angle of attack and pitch rate. We are not sure why this is happening. It may be due to the relaxed stability of the F/A-18 aircraft and/or to the time varying Mach number which was assumed constant in deriving the outer layer solution. Consequently, we attempt the use of pole placement in order to bring the actual response closer to the outer layer solution. By placing the poles of the fast variable system at more negative values, the stability of the outer layer solution is improved. This pole placement is achieved by placing another loop around the system during the outer layer portion of the maneuver. The effect is to alter the stabilator feedback by an amount based on how far the system is from the desired outer layer angle of attack and pitch rate (fast variables). This decreases the time required to execute the outer layer portion of the trajectory. The system used in the pole placement is a linearized model of the outer layer fast variables. Pole placement is done using the Bass-Gura pole placement formula. The gains calculated using the Bass-Gura formula are multiplied by the corresponding divergences of the angle of attack and the pitch rate from the outer layer solution and these values are added to the outer layer stabilator feedback.

The linearized equations of motion for the fast variables are defined by:

\[
\begin{bmatrix}
\Delta \dot{x} \\
\Delta \dot{q}
\end{bmatrix} =
\begin{bmatrix}
\frac{df_a}{d\alpha} & \frac{df_a}{dq} \\
\frac{df_q}{d\alpha} & \frac{df_q}{dq}
\end{bmatrix}
\begin{bmatrix}
\Delta \alpha \\
\Delta q
\end{bmatrix} +
\begin{bmatrix}
\frac{df_a}{d\delta_h} \\
\frac{df_q}{d\delta_h}
\end{bmatrix} \delta_h
\]  

(5.32)
where:

\[ f_x = q + \frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} C_L - \frac{T_x}{mV} \sin \alpha \]  

(5.33)

\[ f_q = \frac{1}{I_y} \left[ l_x t_x + \bar{q}S(\bar{c} C_m + d_L C_L + d_D C_D) \right] \]  

(5.34)

\[ \frac{df_x}{da} = - \frac{\bar{q}S}{mV} \frac{dC_L}{da} - \frac{T_x}{mV} \cos \alpha \]  

(5.35)

\[ \frac{df_q}{da} = 1 - \frac{\bar{q}S}{mV} C_{Iq} \]  

(5.36)

\[ \frac{df_x}{dq} = \frac{\bar{q}S}{I_y} \left[ \bar{c} \frac{dC_m}{da} + d_L \left( \frac{dC_L}{da} + C_D \right) + d_D \left( \frac{dC_D}{da} - C_L \right) \right] \]  

(5.37)

\[ \frac{df_q}{dq} = \frac{\bar{q}S}{I_y} \left[ \bar{c} \frac{dC_m}{dq} + d_L \frac{dC_L}{dq} \right] \]  

(5.38)

\[ \frac{df_x}{d\delta_h} = - \frac{\bar{q}S}{mV} \frac{dC_L}{d\delta_h} \]  

(5.39)

\[ \frac{df_q}{d\delta_h} = \frac{\bar{q}S}{I_y} \left[ \bar{c} \frac{dC_m}{d\delta_h} + d_L \frac{dC_L}{d\delta_h} + d_D \frac{dC_D}{d\delta_h} \right] \]  

(5.40)

Letting

\[ A = \begin{bmatrix} \frac{df_x}{dx} & \frac{df_x}{dq} \\ \frac{df_q}{dx} & \frac{df_q}{dq} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \frac{df_x}{d\delta_h} \\ \frac{df_q}{d\delta_h} \end{bmatrix} \]  

(5.41)
The open-loop characteristic equation is

\[ p_{O\lambda}(\bar{\lambda}) = \det(\bar{\lambda}I - A) = \bar{\lambda}^2 + \bar{a}_1 \bar{\lambda} + \bar{a}_2 \] (5.42)

where \( \bar{\lambda} \) = open-looped eigenvalues

\[ a_1 = -\frac{df_\alpha}{d\alpha} - \frac{df_q}{dq} \] (5.43)

\[ a_2 = \frac{df_\alpha}{d\alpha} \frac{df_q}{dq} - \frac{df_\alpha}{dq} \frac{df_q}{d\alpha} \] (5.44)

And the controllability matrix \( U \) is given as:

\[ U = [B, AB] \]

\[ = \begin{bmatrix}
\frac{df_\alpha}{d\delta_h} & \frac{df_\alpha}{d\alpha} & \frac{df_q}{d\delta_h} & \frac{df_q}{dq}
\frac{df_\alpha}{d\alpha} & \frac{df_\alpha}{d\delta_h} & \frac{df_q}{d\alpha} & \frac{df_q}{dq}
\frac{df_q}{d\delta_h} & \frac{df_q}{d\alpha} & \frac{df_q}{d\delta_h} & \frac{df_q}{dq}
\end{bmatrix} \] (5.45)

It is desired to have Butterworth Poles. Therefore the desired closed-loop characteristic equation is:

\[ p_{C\lambda}(\tilde{\lambda}) = \tilde{\lambda}^2 + \tilde{a}_1 \tilde{\lambda} + \tilde{a}_2 \] (5.46)

where \( \tilde{\lambda} \) = closed-loop poles

\[ \tilde{a}_1 = \sqrt{2\omega} \] (5.47)

\[ \tilde{a}_2 = \omega^2 \] (5.48)
\( \omega = 1.5 \) \hspace{1cm} (5.49)

The Bass-Gura Formula determines the feedback gains as follows:

\[ K = (\tilde{a}_1 - a_1, \tilde{a}_2 - a_2)W^{-TU^{-1}} \] \hspace{1cm} (5.50)

Where the matrix \( W \) is a lower Toeplitz matrix defined by

\[ W = \begin{bmatrix} 1 & 0 \\ a_1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{df_{\alpha}}{d\alpha} - \frac{df_{q}}{dq} & 1 \end{bmatrix} \] \hspace{1cm} (5.51)

The gains \( K \) are used in the outer layer feedback law to help stabilize the aircraft onto the outer layer. This is done in the following manner.

Define

\[ \Delta \alpha = \alpha - \alpha^* \] \hspace{1cm} (5.52)

\[ \Delta q = q - q^* \] \hspace{1cm} (5.53)

The total outer layer feedback law is then defined as:

\[ \delta_h = \delta_h^* - K \begin{bmatrix} \Delta \alpha \\ \Delta q \end{bmatrix} \quad -24^\circ \leq \delta_h \leq 10.5^\circ \]

\[ = \delta_h^* - K_1 \Delta \alpha - K_2 \Delta q \] \hspace{1cm} (5.54)
Figure 32. Eigenvalues of Fast System for Outer Layer Simulation
Figure 33. Eigenvalues of Fast System for Outer Layer Simulation
6.0 Transition Region Analysis

The transition region analysis involves the transition from the initial states to their outer layer values at the beginning of the maneuver and the transition from the outer layer to some final state at the end of the maneuver. The pitch rate and the angle of attack are assumed to be fast variables compared to Mach number, flight path angle and altitude. The analysis can be broken up into two portions. First, we consider the transition of the angle of attack from its initial state to its outer layer value. The pitch rate is assumed to be the fast variable here. Next, we consider the transition from the initial pitch rate to the outer layer pitch rate. We emphasize again that we are not obtaining a pure singular perturbation solution. Our singular perturbation treatment of these transition regions is separate from the singular perturbation treatment of the outer layer. We apply singular perturbation analysis to a new problem. This is the problem of bringing the states from a set of initial conditions to the outer layer values in minimum time, ignoring the fact that the outer layer values were obtained using singular perturbation analysis. That is, we use singular perturbation analysis to solve a new two-point boundary-value problem.
6.1 First Part of the Transition Region Analysis

The performance index $\Delta J_\alpha$ is the time required to transfer the angle of attack from its initial state to its outer layer value.

\[ \Delta J_\alpha = \int_{t_0}^{\tau} dt \]

(6.1)

The pitch rate is assumed to be fast compared to the angle of attack. Therefore, for a given flight path angle, Mach number and altitude, the equations of motion are:

\[ \dot{\alpha} = q + \frac{g}{V} \cos \gamma - \frac{\bar{q} S}{mV} C_L - \frac{T_x}{mV} \sin \alpha \]

(6.2)

\[ \epsilon \dot{q} = \frac{1}{I_y} \left[ I_y T_x + \bar{q} S (\epsilon C_m + d_L C_L + d_D C_D) \right] \]

(6.3)

From equation (6.1)

\[ \Delta J_\alpha = \int_{t_0}^{\tau} dt = \int_{\alpha(t_0)}^{\alpha(\tau)} \frac{d\alpha}{\dot{\alpha}} \]

(6.4)

Equation (6.4) indicates that the integrand of the performance index is minimized when the rate of change of angle of attack is maximized. As $\epsilon$ approaches zero the problem can be considered in the following manner.

For a given $\gamma, M, h$ and $\alpha$,
\[
\max_{\delta_h, \delta} \dot{\alpha}(\alpha, \delta_h, \gamma, M, h, q)
\]

Subject to equation (6.3) with \( \varepsilon \to 0 \)

The values of the pitch rate and the stabilator deflection which satisfy these conditions are designated \( \hat{\delta}_h(\alpha, \gamma, M, h) \) and \( \hat{\delta}_q(\alpha, \gamma, M, h) \).

Let

\[
\hat{\dot{\alpha}}(\alpha, \gamma, M, h) = \max_{\delta_h, \delta} \dot{\alpha}(\alpha, \delta_h, \gamma, M, h, q)
\]  

(6.5)

Subject to

\[
0 = \frac{1}{I_y} \left[ I_x T_x + \bar{q} s(eC_m + d_L C_L + d_D C_D) \right] = g(\alpha, \delta_h, \gamma, M, h, q)
\]  

(6.6)

Define the Hamiltonian as:

\[
H(\alpha, \delta_h, \gamma, M, h, q) = -\dot{\alpha}(\alpha, \delta_h, \gamma, M, h, q) + \lambda g(\alpha, \delta_h, \gamma, M, h, q)
\]  

(6.7)

The necessary conditions for a stationary point on \( -\dot{\alpha} \) are:

\[
g(\alpha, \delta_h, \gamma, M, h, q) = 0
\]  

(6.8)

\[
\frac{\partial H}{\partial \delta_q}(\alpha, \delta_h, \gamma, M, h, q) = 0
\]  

(6.9)

\[
\frac{\partial H}{\partial \delta_h}(\alpha, \delta_h, \gamma, M, h, q) = 0
\]  

(6.10)

Transition Region Analysis
Substituting equation (6.7) into equations (6.8) - (6.10)

\[
g(\alpha, \delta, \gamma, M, h, \hat{q}) = 0 \quad (6.11)
\]

\[
- \frac{d\hat{\alpha}}{dq} (\alpha, \delta, \gamma, M, h, \hat{q}) + \lambda \frac{dg}{dq} (\alpha, \delta, \gamma, M, h, \hat{q}) = 0 \quad (6.12)
\]

\[
- \frac{d\hat{\alpha}}{d\delta} (\alpha, \delta, \gamma, M, h, \hat{q}) + \lambda \frac{dg}{d\delta} (\alpha, \delta, \gamma, M, h, \hat{q}) = 0 \quad (6.13)
\]

If

\[
C_L(\alpha, \delta, M) = C_{L_0}(\alpha, \delta, M) + C_{L_\alpha}(\alpha, \delta, M)q + C_{L_\alpha}(\alpha, \delta, M)\hat{\alpha} \quad (6.14)
\]

and

\[
C_m(\alpha, \delta, M) = C_{m_0}(\alpha, \delta, M) + C_{m_\alpha}(\alpha, \delta, M)q + C_{m_\alpha}(\alpha, \delta, M)\hat{\alpha} \quad (6.15)
\]

Then

\[
\hat{\alpha} = \frac{q + \frac{g}{V} \cos \gamma - \frac{\bar{\alpha}S}{mV} (C_{L_0} + C_{L_\alpha}q + C_{L_\alpha} \hat{\alpha}) - \frac{T_x}{mV} \sin \alpha}{1 + \frac{\bar{\alpha}S}{mV} C_{L_\alpha}} \quad (6.16)
\]

and

\[
k = \frac{1}{l_y} \left[ l_x T_x + \bar{\alpha} S [\bar{C}(C_{m_0} + C_{m_\alpha}q + C_{m_\alpha} \hat{\alpha}) + d_D C_D + d_L (C_{L_0} + C_{L_\alpha}q + C_{L_\alpha} \hat{\alpha})] \right] \quad (6.17)
\]

Therefore,
\[
\frac{d\dot{V}}{dq} = \frac{1 - \frac{\bar{q}S}{mV} C_{L_v}}{1 + \frac{\bar{q}S}{mV} C_{L_\alpha}} \\
\frac{d\dot{\alpha}}{d\delta_h} = -\frac{\bar{q}S}{mV} \frac{dC_{L_\alpha}}{d\delta_h} \\
\frac{dk}{dq} = \frac{\bar{q}S}{I_y} \left[ \bar{\alpha} C_{m_\alpha} + d_L C_{L_\alpha} \right] \\
\frac{dk}{d\delta_h} = \frac{\bar{q}S}{I_y} \left[ \bar{\alpha} \frac{dC_{m_\alpha}}{d\delta_h} + d_D \frac{dC_D}{d\delta_h} + d_L \frac{dC_{L_\alpha}}{d\delta_h} \right]
\]

Equations (6.18)-(6.21) are used in equations (6.12)-(6.13) to satisfy the necessary conditions for a stationary point on \(-\dot{\alpha}\).

### 6.2 Results: First Part of the Transition Region Analysis

The problem is solved numerically using "SLLSQP" (Ref. 10). The feedback control law involves saturation of the control in order to minimize \(\Delta I_\alpha\). Consequently, this control law is not a function of flight path angle, Mach number or altitude. It is therefore only a function of angle of attack. This result is due the physical constraints of the stabilator angle deflection and indicates that the unconstrained minimum time control setting is outside the physical limits of the aircraft.
For $\alpha < \alpha'(\gamma, M, h)$

$$\lambda \delta_h(\alpha) = \begin{cases} 
-24^\circ & \alpha < 45^\circ \\
10.5^\circ & 45^\circ \leq \alpha \leq 55^\circ \\
-24^\circ & \alpha > 55^\circ 
\end{cases}$$

(6.22)

For $\alpha > \alpha'(\gamma, M, h)$

$$\lambda \delta_h(\alpha) = \begin{cases} 
10.5^\circ & \alpha < 45^\circ \\
-24^\circ & 45^\circ \leq \alpha \leq 55^\circ \\
10.5^\circ & \alpha > 55^\circ 
\end{cases}$$

(6.23)

Equation (6.23) results from analyzing the inverse problem where the initial angle of attack is greater than the outer layer angle of attack.

Given the control which maximizes the rate of change of angle of attack for all pitch rates, we now must calculate the pitch rate which gives us $\dot{q} = 0$. This is done by solving equation (6.6) for $q$ and using the control defined by equations (6.22) and (6.23).

$$\dot{\tilde{q}}(\alpha, M, h) = \frac{-l_z T_x - \bar{q} \tilde{S}(\tilde{C}_{m_0} + d_{L} C_{L_0} + d_{D} C_{D_0})}{\bar{q} \tilde{S}(\tilde{C}_{m_q} + d_{L} C_{L_q})}$$

(6.24)

Therefore equations (6.22)-(6.24) define the solution to the first part of the transition region analysis. In order to minimize $\Delta J_x$, we use the control law $\delta_h(\alpha)$ with the pitch rate equal to $\dot{\tilde{q}}$. 

Transition Region Analysis
6.3 Second Part of the Transition Region Analysis

For a given angle of attack, the transition from the initial pitch rate to $\dot{\gamma}(M, h)$ is considered. The equations of motion are

$$\dot{\gamma} = \frac{1}{I_y} [I_x T_x + \bar{q} S \bar{c} C_m + d_L C_L + d_D C_D]$$  \hspace{1cm} (6.25)

$$\dot{V} = -g \sin \gamma - \frac{\bar{q} S}{m} C_D + \frac{T_x}{m} \cos \alpha$$  \hspace{1cm} (6.26)

$$\dot{h} = V \sin \gamma$$  \hspace{1cm} (6.27)

The performance index to be minimized is

$$\Delta J_q = \int_{t_0}^{t_1} dt$$  \hspace{1cm} (6.28)

$$\dot{q} = \frac{dq}{dt}$$  \hspace{1cm} (6.29)

Therefore,

$$\Delta J_q = \int_{q(t_0)}^{q(t_1)} \frac{dq}{\dot{q}}$$  \hspace{1cm} (6.30)
In order to minimize the integrand of the performance index, we desire to maximize the rate of change of pitch rate for all \( q \) on the interval from the initial pitch rate to \( \hat{q} \). From equation (6.25)

\[
\dot{q} = \frac{1}{I_y} \left[ I_x T_x + \bar{q} S(\bar{C}_{m_0}(\alpha, \delta_h) + C_{m_q}(\alpha)q) + d_L(C_{L_0}(\alpha, \delta_h) + C_{L_q}(\alpha)q) \\
+ d_D C_D(\alpha, \delta_h) \right]
\]

(6.31)

we can see that we choose \( \delta_h \) such that the quantity

\[
\bar{C}_{m_0}(\alpha, \delta_h) + d_L C_{L_0}(\alpha, \delta_h) + d_D C_D(\alpha, \delta_h)
\]

is maximized. Numerical results indicate that this quantity is maximized by the following control law.

\[
\tilde{\delta}_h = \begin{cases} 
-24^\circ & q < \hat{q}(\alpha, M, h) \\
10.5^\circ & q > \hat{q}(\alpha, M, h)
\end{cases}
\]

(6.32)

The stabilator angle which minimizes the integrand of the performance index is outside the allowable region for the F/A-18. Therefore, the control law which satisfies the aircraft constraints calls for the stabilator to be saturated.
7.0 Transition Regions

The entire maneuver is considered using three distinct portions. These portions include two transition regions and the outer layer region. The transition regions are necessary because the fast variables cannot be instantaneously changed from the initial conditions to the outer layer and from the outer layer to the final conditions.

We desire to execute the transition regions in minimum time. We treat the transition regions as a two-point boundary-value problem. Therefore, switching times are calculated by considering approximate solutions to the nonlinear differential equations of motion. The approximate solutions are used to find the switching times necessary to bring the aircraft state to the outer layer or the final state of the half-loop maneuver.
7.1 First Transition Region

The switching time for the first transition region is defined as the time at which a switch in control will bring the angle of attack and the pitch rate to their outer layer values simultaneously.

The time rate of change of the pitch rate is given by

\[
\dot{q} = \frac{1}{I_y} \left[ l_x T_x + \bar{q} S \left[ \bar{C} C_m_\alpha + d_L C_{L\alpha} + d_D C_D + (\bar{C} C_m_\alpha + d_L C_{L\alpha}) \dot{\alpha} + (\bar{C} C_m_\alpha + d_L C_{L\alpha}) \dot{q} \right] \right]
\]

(7.1)

\[
\dot{q} = [f_2(\delta_h) + q] f_1
\]

(7.2)

Where

\[
f_1 = \frac{\bar{q} S}{I_y} (\bar{C} C_m_\alpha + d_L C_{L\alpha})
\]

(7.3)

\[
f_2(\delta_h) = \frac{l_x T_x + \bar{q} S \left[ \bar{C} C_m_\alpha + d_L C_{L\alpha} + d_D C_D + (\bar{C} C_m_\alpha + d_L C_{L\alpha}) \dot{\alpha} \right]}{\bar{q} S (\bar{C} C_m_\alpha + d_L C_{L\alpha})}
\]

(7.4)

The solution to equation (7.2) is

\[
q(t) = [f_2(\delta_h) + q(t_0)] e^{f(t - t_0)} - f_2(\delta_h)
\]

(7.5)

Using the control law

\[
\mathbf{\delta}^{(1)}_h = \begin{cases} 
-24^\circ & \text{if } q < q^* (\gamma, M, h) \\
10.5^\circ & \text{if } q > q^* (\gamma, M, h)
\end{cases}
\]

(7.6)
in equation (7.5), it follows that

\[ q^* = \left[ f_2(\delta_h^{(1)}) + q(t_0) \right] e^{f_1 \Delta t} - f_2(\delta_h^{(1)}) \]  

(7.7)

Where \( \Delta t \) is the time required to change the pitch rate from \( q(t_0) \) to \( q^* \) using \( \delta_h^{(1)} \).

\[ \Delta t = \frac{1}{f_1} \ln \left[ \frac{q^* + f_2(\delta_h^{(1)})}{q(t_0) + f_2(\delta_h^{(1)})} \right] \]  

(7.8)

The time rate of change of the angle of attack is given by

\[ \dot{\alpha} = q + \frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} (C_{tq} + C_{t\dot{q}} q + C_{t\dot{\alpha}} \dot{\alpha}) - \frac{T_x}{mV} \sin \alpha \]  

(7.9)

\[ \dot{\alpha} = f_3 q + f_4(\delta_h) \]  

(7.10)

where

\[ f_3 = \frac{1 - \frac{\bar{q}S}{mV} C_{tq}}{1 + \frac{\bar{q}S}{mV} C_{t\dot{\alpha}}} \]  

(7.11)

\[ f_4(\delta_h) = \frac{\frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} C_{tq} - \frac{T_x}{mV} \sin \alpha}{1 + \frac{\bar{q}S}{mV} C_{t\dot{\alpha}}} \]  

(7.12)

The solution to equation (7.10) is

\[ \alpha(t) = \left[ f_4(\delta_h^{(1)}) + f_3 q \right] \Delta t \]  

(7.13)

Transition Regions

69
We calculate the change in angle of attack for a given $\Delta t$ and $\delta_h^{(i)}$:

$$
\Delta \alpha = \left[ f_3(\delta_h^{(i)}) - f_3 f_4(\delta_h^{(i)}) \right] \Delta t + \frac{f_3}{f_1} \left[ f_2(\delta_h^{(i)}) + q(t_0) \right] (e^{f_1 \Delta t} - 1) \tag{7.14}
$$

Therefore, to bring the angle of attack and the pitch rate to the outer layer values at the same time, we apply $\delta_h^{(i)}$ when $|\alpha'(\gamma, M, h) - \alpha| \leq \Delta \alpha$.

### 7.2 Second Transition Region

In order to change the angle of attack and the pitch rate from the outer layer to the final conditions of zero pitch rate and a specified flight path angle, the switching time for the second transition region must be determined. The switching time is defined as the time in which the pitch rate is changed from its outer layer value to zero at the same time that the flight path angle reaches the desired value ($\gamma_f = 180^\circ$ for the half-loop maneuver).

Assuming that the flight path angle is near the desired value and nearly constant aero-dynamic coefficients, the time rate of change of flight path angle is given by

$$
\dot{\gamma} = -\frac{g}{V} \cos \gamma + \frac{\bar{q}S}{mV} (C_{L_0} + C_{Lq}q + C_{L\dot{\alpha}}) + \frac{T_x}{mV} \sin \alpha \tag{7.15}
$$

$$
\dot{\gamma} = f_3 q + f_3(\delta_h) \tag{7.16}
$$

where
\begin{align*}
  f_s &= \frac{\bar{q} S}{m V} C_{L_\alpha} \\
  f_6(\delta_h) &= -\frac{g}{V} \cos(\theta - \alpha) + \frac{\bar{q} S}{m V} (C_{L_\alpha} + C_{L_h} \dot{\alpha}) + \frac{T_x}{m V} \sin \alpha
\end{align*}

Using \( \delta_h^0 = 10.5^\circ \) in equation (7.5) with final pitch rate equal to zero

\begin{equation}
  0 = \left[ f_2(\delta_h^{(2)}) + q^* \right] e^{f_1 \Delta t} - f_2(\delta_h^{(2)})
\end{equation}

The time required to reach this condition is

\begin{equation}
  \Delta t = \frac{1}{f_1} \ln \left[ \frac{f_2(\delta_h^{(2)})}{q^* + f_2(\delta_h^{(2)})} \right]
\end{equation}

Equation (7.16) is solved to find the change in flight path angle for \( \Delta t \) and \( \delta_h^0 \)

\begin{equation}
  \Delta \gamma = \left[ f_6(\delta_h^{(2)}) - f_5 f_2(\delta_h^{(2)}) \right] \Delta t + \frac{f_5}{f_1} \left[ f_2(\delta_h^{(2)}) + q^* \right] (e^{f_1 \Delta t} - 1)
\end{equation}

Therefore, to bring the pitch rate to zero and the flight path angle to a specified value simultaneously, we apply \( \delta_h^0 \) when \( \gamma_r - \gamma \leq \Delta \gamma \).
8.0 Nonlinear Feedback Law

The feedback laws for each portion of the maneuver are combined to define the overall feedback law. A computer code in the form of a subroutine provides the feedback control law. It generates a value of the stabilator control setting for any given value of the F/A-18 aircraft state vector. The feedback law is divided into three regions corresponding to the first transition region, the outer layer and the second transition region.

8.1 First Transition Region

In this region, we are concerned with bringing the aircraft states from their initial values to the outer layer values simultaneously. This process involves at least two and possibly three different combinations of states which require a particular control law strategy.
8.1.1 Part I

In this part of the first transition region, we wish to bring the angle of attack up to a prescribed neighborhood of the outer layer angle of attack. The pitch rate is not equal to the value which gives us a constant pitch rate as prescribed in the analysis of the first inner layer.

\[ q \neq \hat{q}(\alpha, M, h) \]

and the angle of attack is not within the acceptable neighborhood of the outer layer angle of attack required for the first transition region.

\[ |\alpha^{*}(\gamma, M, h) - \alpha| > \Delta \alpha \]

The control law for this is given by:

For \( q < \hat{q}(\alpha, M, h) \)

\[
\delta_h^I = \begin{cases} 
-24^\circ & \alpha \leq 45^\circ \text{ or } \alpha \geq 55^\circ \\
10.5^\circ & 45^\circ < \alpha < 55^\circ 
\end{cases}
\] (8.1)

For \( q > \hat{q}(\alpha, M, h) \)

\[
\delta_h^I = \begin{cases} 
10.5^\circ & \alpha \leq 45^\circ \text{ or } \alpha \geq 55^\circ \\
-24^\circ & 45^\circ < \alpha < 55^\circ 
\end{cases}
\] (8.2)

We apply \( \delta_h^I \) until \( q = \hat{q}(\alpha, M, h) \) or \( |\alpha^{*}(\gamma, M, h) - \alpha| \leq \Delta \alpha \).
8.1.2 Part II

This part of the first transition region is used to bring the angle of attack up to the prescribed neighborhood for the case where the pitch rate is equal to the value required for a constant pitch rate in the first part of the transition region analysis.

\[ q = \hat{q}(\alpha, M, h) \]

and the angle of attack is not within the required neighborhood of the outer layer value desired.

\[ |\alpha^*(\theta, M, h) - \alpha| > \Delta \alpha \]

The feedback law for this part of the transition is

For \( \alpha < \alpha^*(\gamma, M, h) \)

\[ \delta_h^{II} = \begin{cases} -24^\circ & \alpha \leq 45^\circ \text{ or } \alpha \geq 55^\circ \\ 10.5^\circ & 45^\circ < \alpha < 55^\circ \end{cases} \tag{8.3} \]

For \( \alpha > \alpha^*(\gamma, M, h) \)

\[ \delta_h^{II} = \begin{cases} 10.5^\circ & \alpha \leq 45^\circ \text{ or } \alpha \geq 55^\circ \\ -24^\circ & 45^\circ < \alpha < 55^\circ \end{cases} \tag{8.4} \]

We apply \( \delta_h^{II} \) until \( |\alpha^*(\gamma, M, h) - \alpha| \leq \Delta \alpha \).
Often, this portion of the transition region is not used because the angle of attack is within the required neighborhood before the pitch rate reaches \( \hat{q} \). If this occurs it is no longer desirable to have the pitch rate equal to \( \hat{q} \).

### 8.1.3 Part III

In this portion of the transition region, we are concerned with applying a control which will bring the pitch rate to the required outer layer pitch rate. The pitch rate is less than or equal to the \( \hat{q} \):

\[
q \leq \hat{q}(\alpha, M, h)
\]

and the angle of attack is within the necessary neighborhood to bring it to the outer layer angle of attack

\[
| \alpha^*(y, M, h) - \alpha | \leq \Delta \alpha
\]

The control which accomplishes the objective of this portion of the transition is given as

\[
\delta_{h}^{III} = \begin{cases} 
-24^\circ & q < q^*(y, M, h) \\
10.5^\circ & q > q^*(y, M, h)
\end{cases}
\]

We apply \( \delta_{h}^{III} \) until the angle of attack and the pitch rate are equal to the desired outer layer values: \( \alpha = \alpha^*(y, M, h) \) and \( q = q^*(y, M, h) \).
8.2 Outer Layer

In the outer layer region, the velocity vector is rotated through a flight path angle of one hundred and eighty degrees. The values of the angle of attack, pitch rate and stabilator deflection are determined by the surfaces of the outer layer solution. Therefore, the values of angle of attack and pitch rate are given by

\[ q = q^* (\gamma, M, h) \]

\[ \alpha = \alpha^* (\gamma, M, h) \]

and the flight path angle is not within the neighborhood of the desired final flight path angle.

\[ \gamma < \gamma_f - \Delta \gamma \]

The control law is defined by the surface represented by the outer layer solution for stabilator deflection

\[ \delta_h^{OL} = \delta_h^* (\gamma, M, h) \]

We apply \( \delta_h^{OL} \) until \( \gamma \geq \gamma_f - \Delta \gamma \).

The size of the neighborhood defined by \( \Delta \gamma \) is calculated throughout the outer layer portion of the maneuver in order to determine when to switch to the control law of the second transition region. This time is defined as when the flight path angle equals the
value at which a switch in the control will bring the flight path angle to its final value
at the same time that the pitch rate is zero.

8.3 Second Transition Region

This final region of the control law is used to bring the flight path angle to the desired
final value and the pitch rate to zero simultaneously. The angle of attack and pitch rate
are equal to their outer layer values at the beginning of the region

\[ q = q^*(\gamma, M, h) \]
\[ \alpha = \alpha^*(\gamma, M, h) \] initially

and the flight path angle is within the neighborhood required for the second transition
region.

\[ \gamma \geq \gamma_f - \Delta \gamma \]

The control used to bring the pitch rate down to zero is

\[ \delta_h^F = 10.5^\circ \] (8.7)

and is applied until \( q = 0 \) and \( \gamma = \gamma_f \)

Equations (8.1)-(8.7) define the entire feedback control law necessary to bring the air-
craft from an initial state through the half-loop maneuver to the final flight path angle
with a zero pitch rate.
9.0 Results

The following figures show the results of simulations of the half-loop maneuver using the feedback control law developed in this study. The initial altitude in all cases was 15,000 ft. These simulations were generated using a fourth-order Runge-Kutta integrator on the VAX 8800 computer at Virginia Polytechnic Institute and State University. Figures 34 - 46 show the simulation results for an initial Mach number of 0.9. The simulation results for an initial Mach number of 0.6 are shown in figures 47 - 59. In figures 60 - 72, the simulation results are given for an initial Mach number of 0.3. For each simulation, the circular frequency of the Butterworth Poles is 1.5 rad/sec.

When the initial Mach number is 0.9, the maneuver takes approximately nine (9) seconds. In figures 34 and 35, it is possible to see the large change in the angle of attack and pitch rate associated with the first transition region. This region of the maneuver takes about one (1) second to complete. The angle of attack is increased from approximately six degrees to the outer layer value of about thirty-four (34) degrees. In the same time period, the pitch rate is increased from zero up to approximately one hundred and ten (110) degrees per second and back down to the outer layer pitch rate which is about
Figure 34. Half-Loop Simulation: Angle of Attack ($M(t = 0) = 0.9$, $h = 15000$ ft)
Figure 35. Half-Loop Simulation: Pitch Rate ($M(t = 0) = 0.9$, $h = 15000$ ft)
Figure 36. Half-Loop Simulation: Airspeed (M(t = 0) = 0.9, h = 15000 ft)
Figure 37. Half-Loop Simulation: Mach Number ($M(t=0)=0.9$, $h=15000$ ft)
Figure 38. Half-Loop Simulation: Flight Path Angle \( M(t = 0) = 0.9, \ h = 15000 \, \text{ft} \)
Figure 39.  Half-Loop Simulation: Stabilator Angle ($M(t=0)=0.9$, $h=15800$ ft)
Figure 40. Half-Loop Simulation: Altitude vs Range (M(t=0) = 0.9, h = 15000 ft)
Figure 41. Half-Loop Simulation: Load Factor (M(t = 0) = 0.9, h = 15000 ft)
Figure 42. Half-Loop Simulation: Load Factor (M(t = 0) = 0.9, h = 15000 ft)
Figure 43. Half-Loop Simulation: Dynamic Pressure ($M(t=0)=0.9$, $h=15000$ ft)
Figure 44. Half-Loop Simulation: Fast System Eigenvalues ($M(t=0) = 0.9$, $h = 15000$ ft)
Figure 45. Half-Loop Simulation: Pole Placement Gain ($M(t=0)=0.9$, $h=15000$ ft)
Figure 46. Half-Loop Simulation: Pole Placement Gain ($M(t = 0) = 0.9, h = 15900$ ft)
Figure 47. Half-Loop Simulation: Angle of Attack ($M(t=0)=0.6$, $h=15000$ ft)
Figure 48. Half-Loop Simulation: Pitch Rate ($M(t=0)=0.6$, $h=15000$ ft)
Figure 49.  Half-Loop Simulation: Airspeed ($M(t = 0) = 0.5$, $h = 15000$ ft)
Figure 50. Half-Loop Simulation: Mach Number (M(t = 0) = 0.6, h = 15000 ft)
Figure 51. Half-Loop Simulation: Flight Path Angle ($M(t=0)=0.6$, $h=15000$ ft)
Figure 52. Half-Loop Simulation: Stabilator Angle (M(t = 0) = 0.6, h = 15000 ft)
Figure 53. Half-Loop Simulation: Altitude vs Range (M(t = 0) = 0.6, h = 15000 ft)
Figure 54. Half-Loop Simulation: Load Factor ($M(t=0)=0.6$, $h=15000$ ft)
Figure 55. Half-Loop Simulation: Load Factor ($M(t=0)=0.6, h=15000$ ft)
Figure 56. Half-Loop Simulation: Dynamic Pressure ($M(t=0)=0.6$, $h=15000$ ft)
Figure 57. Half-Loop Simulation: Fast System Eigenvalues (M(t = 0) = 0.6, h = 15000 ft)
Figure 58. Half-Loop Simulation: Pole Placement Gain (M(t = 0) = 0.6, h = 15000 ft)
Figure 59. Half-Loop Simulation: Pole Placement Gain \((M(t=0)=0.6, h=15000 \text{ ft})\)
Figure 60. Half-Loop Simulation: Angle of Attack ($M(t = 0) = 0.3, h = 15000$ ft)
Figure 61. Half-Loop Simulation: Pitch Rate (M(t = 0) = 0.3, h = 15000 ft)
Figure 62. Half-Loop Simulation: Airspeed (M(t = 0) = 0.3, h = 15000 ft)
Figure 63. Half-Loop Simulation: Mach Number \( M(t=0) = 0.3, h = 15000 \text{ ft} \)
Figure 64. Half-Loop Simulation: Flight Path Angle ($M(t=0) = 0.3, h = 15000 \text{ ft}$)
Figure 65. Half-Loop Simulation: Stabilator Angle ($M(t = 0) = 0.3, h = 15000 \text{ ft}$)
Figure 66.  Half-Loop Simulation: Altitude vs Range (M(t = 0) = 0.3, h = 15000 ft)
Figure 67. Half-Loop Simulation: Load Factor ($M(t=0)=0.3$, $h=15000$ ft)
Figure 68. Half-Loop Simulation: Load Factor ($M(t=0)=0.3$, $h=15000$ ft)
Figure 69. Half-Loop Simulation: Dynamic Pressure \((M(t = 0) = 0.3, h = 15000 \text{ ft})\)
Figure 79. Half-Loop Simulation: Fast System Eigenvalues ($M(\tau = 0) = 0.3$, $h = 15000$ ft)
Figure 71. Half-Loop Simulation: Pole Placement Gain ($M(t=0)=0.3$, $h=15000$ ft)
Figure 72. Half-Loop Simulation: Pole Placement Gain ($M(t=0) = 0.3, h = 15000 \text{ ft}$)
twenty-five (25) degrees per second. The outer layer portion of the solution requires
around seven and one half (7.5) seconds. We observe that the simulation does not fol-
low exactly the outer layer values for the angle of attack and the pitch rate. Figure 39
gives the time history of the stabilator deflection. The bang-bang nature of the first
transition region is evident. The difference between the solid and dotted lines show the
effect of the pole placement on the stabilator feedback. The end of the maneuver con-
sists of a final bang control which can be seen here. The second transition region only
takes about one half (0.5) second. Figure 40 gives a representation of the path of the
maneuver in x-y space. This figure is not to the same scale in both the x and y directions
and therefore the relative altitude and range changes are skewed. However, it is possible
to see that the conclusion of the maneuver is downrange from the beginning. The evol-
ution of the flight path angle from zero (0) degrees to one hundred and eighty (180)
degrees is given in figure 38.

Two significant effects involved in this maneuver can be seen in figures 36 - 37 and 41 -
42. Figures 36 and 37 show the large drain in energy involved in this maneuver. The
airspeed and Mach numbers are significantly decreased over the course of the maneuver.
The dynamic pressure in figure 43 also shows this effect. This condition is one reason
why pilots will often avoid the half-loop maneuver in a combat situation. The small
increase in Mach number and airspeed at the beginning of the maneuver is due to the
fact that the thrust is not at an equilibrium value at the beginning of the maneuver. The
second effect of interest is the large load factors encountered at the beginning of the
maneuver. As can be seen in figure 41, the load factor in the z-body direction is over
twice the allowable limit of seven and one half (7.5) g's. Another effect of interest is the
large onset of the load factor in in the first transition region. The load factor does fall
below the allowable limit for most of the maneuver primarily because of the large
amount of energy being lost. The pilot load factor, after the switch in the first transition region, is not as large as that of the center of gravity because the pitch rate is dropping off rapidly at that time. From this figure, we can see that the effect of constraining the maneuver below a prescribed load factor would primarily change the feedback law in the first transition region and at the beginning of the outer layer region. The load factor in the x-body direction, figure 42, does not show large effects which would be dangerous for the aircraft or the pilot.

The eigenvalues of the fast system are plotted in figure 44 in order to see how the pole placement has effected them. Figures 45 and 46 give the pole placement gains corresponding to the fast variables of angle of attack (alpha) and pitch rate (q). These gains where multiplied by the amount of divergence of the corresponding states from the outer layer values desired. They are equal to zero during the transition regions because pole placement is not used during these regions of the maneuver.

For the initial Mach number of 0.6, the maneuver requires approximately thirteen (13) seconds. In figures 47 and 48, we see that the angle of attack and the pitch rate follow the outer layer values better than for the Mach 0.9 case. The outer layer angle of attack is slightly larger as is the outer layer pitch rate. However, the pitch rate only reaches around ninety (90) degrees per second at its maximum point. The time required to execute the first transition region is about the same as in the previous case but the second transition region takes considerably longer. In figures 49 and 50, the same loss of energy associated with this maneuver is visible with the final Mach number going as low as 0.13.

The flight path angle in figure 51 exhibits results very similar to that of the Mach 0.9 case. This is true of the stabilator deflection in figure 52 and the flight trajectory plot in figure 53. The load factor in the z-body direction does not violate the pilot load factor.
limitation but the rate at which the load factor increases during the first transition region is still a point of concern. Again, the load factor in the x-body direction is not of any great concern. Figures 56 - 59 indicate trends very similar to those discussed earlier for the initial Mach number of 0.9.

As can be seen in figures 60 - 72, the aircraft was unable to execute the half-loop maneuver for an initial Mach number of 0.3. This is not unexpected because the thrust-to-weight ratio (approximately 0.6) is much less than unity for this low Mach number. At Mach 0.3, the aircraft lacks the energy required to execute the half-loop maneuver. The NASA simulations that will be discussed later also indicate that the maneuver is not possible at this low initial Mach number. For this simulation, the results are plotted out to the point where it is obvious that the aircraft cannot execute the maneuver. The angle of attack and the pitch rate states both follow the outer layer values to a good extent. This indicates a trend that the fast variable system is more stable as initial Mach number decreases. The first transition region takes slightly longer to reach the outer layer at this Mach number.

The remaining figures indicate results very similar to those expected for a failed maneuver attempt. The flight path angle is only able to reach around thirty (30) degrees. The corresponding altitude change is about one thousand (1000) feet over a range of three thousand (3000) feet. The z-body direction load factor is not large and as expected the x-body direction load factor is very small. In general, the result for the initial Mach number of 0.3 are of little importance because the maneuver cannot be executed at this slow airspeed.

The next set of figures are a comparison with half-loop maneuver simulations done at NASA Langley Research Center. Figures 73 - 80 are for an initial Mach number of 0.9.
The simulation comparisons for an initial Mach number of 0.6 are given in figures 81 - 88. A comparison of the Mach 0.3 case is also included in figures 89 - 96.

In general, the aircraft response to our feedback control law is similar to the response in the NASA simulations. The major differences are that our control law uses the capabilities of the aircraft more efficiently. This efficiency produces a half-loop maneuver which requires about half the time as the NASA maneuver. The tradeoff comes in the large load factors encountered and the high rate at which these load factors increase. The energy lost in the maneuver is about the same for our control law as with the NASA maneuver. This is seen in the fact that the final Mach numbers and airspeeds are approximately equivalent. However, our maneuvers do not gain as much altitude as those of NASA indicating that the amount of energy lost is a little larger for our feedback control law. However, our half-loop maneuver is tighter. That is, it is performed with less altitude change.

It is important to note that the NASA maneuvers were not simulated in an attempt to do a half-loop in a fast time. For this reason, it is a bit unfair to make too many conclusions based on this comparison. However, these comparisons do indicate that our feedback control law does produce a fast half-loop maneuver which is similar to those done at NASA. Another important factor to keep in mind when making this comparison is that the roll at the end of the maneuver is included in the NASA simulation but not in ours. This roll can be executed in very little time but it is important to note that the final times of the NASA simulations and our simulations do not correspond to the exact same final aircraft state.

Figures 97 - 99 show the time histories for the thrust-to-weight ratios for each initial Mach number. The simulation does not include the change in weight of the aircraft due
Figure 73. Comparison with NASA: Angle of Attack ($M(t = 0) = 0.9$, $h = 15000$ ft)
Figure 74. Comparison with NASA: Pitch Rate \((M(t=0) = 0.9, h = 15000 \text{ ft})\)
Figure 75. Comparison with NASA: Airspeed ($M(t=0)=0.9$, $h=15000$ ft)
Figure 76. Comparison with NASA: Mach Number ($M(t=0) = 0.9$, $h = 15000$ ft)
Figure 77. Comparison with NASA: Pitch Angle ($M(t=0)=0.9$, $h=15000$ ft)
Figure 78. Comparison with NASA: Stabilator Angle ($M(t=0) = 0.9$, $h = 15000$ ft)
Figure 79. Comparison with NASA: Altitude ($M(t = 0) = 0.9, h = 15000$ ft)
Figure 80. Comparison with NASA: Load Factor (M(t=0)=0.9, h=15000 ft)
Figure 81. Comparison with NASA: Angle of Attack ($M(t=0)=0.6$, $h=15000$ ft)
Figure 82. Comparison with NASA: Pitch Rate ($M(t=0)=0.6$, $h=15000$ ft)
Figure 83. Comparison with NASA: Airspeed (M(t = 0) = 0.6, h = 15900 ft)
Figure 84. Comparison with NASA: Mach Number ($M(t = 0) = 0.6$, $h = 15000$ ft)
Figure 85. Comparison with NASA: Pitch Angle ($M(t=0) = 0.6, h = 15000$ ft)
Figure 86. Comparison with NASA: Stabilator Angle (M(t = 0) = 0.6, h = 15000 ft)
Figure 87. Comparison with NASA: Altitude ($M(t=0)=0.6$, $h=15000$ ft)
Figure 88. Comparison with NASA: Load Factor ($M(t = 0) = 0.6, h = 15000 \text{ ft}$)
Figure 89. Comparison with NASA: Angle of Attack ($M(t = 0) = 0.3, h = 15000$ ft)
Figure 90. Comparison with NASA: Pitch Rate (M(t=0)=0.3, h=15000 ft)
Figure 91. Comparison with NASA: Airspeed ($M(\alpha = 0) = 0.3$, $h = 15000$ ft)
Figure 92. Comparison with NASA: Mach Number ($M(t=0) = 0.3$, $h = 15000$ ft)
Figure 93. Comparison with NASA: Pitch Angle ($M(t = 0) = 0.3, h = 15000$ ft)
Figure 94. Comparison with NASA: Stabilator Angle ($M(t = 0) = 0.3$, $h = 15000$ ft)
Figure 95. Comparison with NASA: Altitude (M(t = 0) = 0.3, h = 15000 ft)
Figure 96. Comparison with NASA: Load Factor \( M(t = 0) = 0.3, \ h = 15000 \) ft
to fuel usage. Thus these thrust-to-weight time histories only show the effect of airspeed and altitude changes in the course of the maneuver. Mach number appears to be the predominant effect on these values. These plots also indicate a reason why pilots dread the high load factors of this maneuver.
Figure 97. Thrust-to-Weight Ratio ($M(t=0)=0.9, h = 15000$ ft)
Figure 98. Thrust-to-Weight Ratio ($M(t=0) = 0.6$, $h = 15000$ ft)
Figure 99. Thrust-to-Weight Ratio \((M(t = 0) = 0.3, h = 15000 \text{ ft})\)
10.0 Summary

The nonlinear feedback control law constructed in this study has been shown to produce a fast half-loop maneuver for the F/A-18 aircraft for any given initial state. The control law is relatively easy to generate using the singular perturbation approach. The feedback law approach indicates that the aircraft's capabilities can be more efficiently utilized than with a pilot supplying the input. This is visible in the comparisons with the NASA maneuvers which were generated by input from the stick by a pilot. However, the high load factors and the rate of onset of the load factor are factors to be examined in the use of the feedback control law at high Mach numbers. One way to soften the load factors at high Mach numbers is to generate a constrained feedback control law which would take the load factor limitation into account. Results from such a study would only differ in the first transition region at high initial Mach numbers. Another important point is that the aircraft is shown to be unable to execute the half-loop maneuver at an initial Mach number of 0.3. This is true for both our simulations and the NASA simulation. The low thrust-to-weight ratio of the F/A-18 (approx. 0.6) makes it impossible to perform the half-loop at low Mach numbers.
The small amount of computer code required for the feedback control law to be implemented is a benefit for this approach. The entire feedback control law is set up in about one hundred lines of FORTRAN code. The outer layer surfaces required about nine hundred (900) lines of data each in separate input files which are read at initialization. This small amount of code is able to handle all aircraft initial states possible in the region defined by altitudes up to sixty thousand (60,000) feet and Mach numbers up to 0.9. The feedback control subroutine returns the stabilator deflection, outer layer angle of attack, outer layer pitch rate and the region of the maneuver for any given state vector of the aircraft within the allowable region of Mach number and altitude. This makes this method very useful in practical implementation of the feedback control law.

The efficient use of the aircraft capabilities is a good selling point for the singular perturbations approach to this problem. The feedback control law for the outer layer is generated so that the majority of the energy used in the maneuver is used in accomplishing the rotation of the velocity vector through the required flight path angle. This is done by holding the states, which are not required to rotate the velocity vector, constant. The result is an efficient use of the aircraft state to produce the desired response. The transition regions required to get the aircraft in a position to use the outer layer control law are fast and efficient. Therefore, the singular perturbation approach is a good way of developing an efficient nonlinear feedback control law for the half-loop maneuver.

In comparison with the NASA simulations, our simulations show that our feedback control law executes the half-loop maneuver relatively fast. We notice that, in some cases, the NASA simulations appear to follow our trajectories or oscillate about them. This is an indication that our feedback control law is following a more time efficient
trajectory than that of the NASA simulation. However, it is important to remember that the NASA simulations include the roll at the end of the maneuver, which is ignored in our study, and that the terminal states of the aircraft are slightly different from those in our simulations. Also, it is important to note that the pilot in the NASA simulations was not necessarily trying to execute the maneuver in the fastest possible time. Therefore, the main reason for showing these comparisons is to show that our feedback control law produces a response which is fast in comparison with a similar maneuver. Also, these comparisons indicate that the tremendous loss in energy we experienced with our simulations are inherent in the maneuver because the NASA simulations show similar effects. Finally, the trajectories produced by our feedback control law are more efficient versions of those in the NASA simulations.

Major limitations are that the load factors produced in the first second of the maneuver for high Mach numbers are large, the onset of these large load factors is significant and the maneuver cannot be completed at low Mach numbers. A solution to the load factor problem is to apply a load factor constraint on the analysis. The first transition region would then take a longer time to complete. After the first second of the maneuver (i.e., when the load factor has fallen below 7.5 g's), the solution would be identical to that of this study. A similar approach could be used to limit the onset of the load factor in the first transition region. The inability to complete the maneuver at low Mach numbers is a problem associated with the engines of the F/A-18. If the thrust-to-weight ratio were improved, our feedback control law would be able to execute the maneuver just as well as with higher initial Mach numbers. Therefore, there is no modification to our feedback law which would improve this problem.
In conclusion, the singular perturbation approach used in this study does give us a relatively fast nonlinear feedback control law for the half-loop maneuver. A similar approach could be used to develop a nonlinear feedback control law for the very similar split-s maneuver. In each case, our feedback control law could be implemented in a practical setting onboard the F/A-18 aircraft. Also, the next step would be to use these sub-optimal control laws as a comparison or baseline for the development of both open-loop and closed-loop time optimal control laws for these maneuvers. A study of each of these aspects of the problem is planned for the future.
11.0 References


2. Arbuckle, P.D. and Buttrill, C.S., "NASA F/A-18 Fighter Escort Model" computer code


Appendix A. F/A - 18 Fighter Escort Model

The NASA F/A - 18 Fighter Escort Model consists of an aerodynamic model and a thrust model. Both models use linear interpolation based on nodes contained in a set of look-up tables. This is accomplished using two subroutines for each model. One subroutine sets up the tables in memory while the other does the interpolation.

A.1 Aerodynamic Model

Nondimensional force and moment coefficients are calculated in the aerodynamic model. The model was developed from two wind tunnel tests. Results from these were then combined to create the model. This approach resulted in a discontinuity in the model at the angle of attack where the two sets of test data were joined ($\alpha = 40^\circ$). A flap schedule is used to compensate for this discontinuity. However, for Mach numbers in the region (0.6 to 0.9) and low altitudes a discontinuity still exists. The remedy for this problem is discussed in Appendix B.
The present study deals with flight in the plane of symmetry. Therefore, the coefficients used in longitudinal flight equations of motions are discussed here.

### A.1.1 Lift Coefficients

\[
C_L = f(\alpha, \beta, q, M, h, \dot{\alpha}, \delta_h, \delta_{a}, \delta_{r}, \delta_{\text{lef}}, \delta_{\text{lef}}, \delta_{SB}, \Delta l g) \\
= C_{L0}(\alpha, M) + (\Delta C_{L0})_f (\alpha, M, h) + \Delta C_{Lh}(\alpha, \delta_h, M) \\
+ \left[ C_{L_{\text{lef}}}(\alpha, M) + (\Delta C_{L_{\text{lef}}})_f (\alpha, M, h) \right] \delta_{\text{lef}} \\
+ \left[ C_{L_{\text{lef}}}(\alpha, M) + (\Delta C_{L_{\text{lef}}})_f (\alpha, M, h) \right] \delta_{\text{def}} \\
+ \Delta C_{L_{\beta}}(\alpha, \delta_{a}, M) + \Delta C_{L_{\beta}}(\alpha, \beta, M) + \Delta C_{L_{\epsilon}}(\alpha, \delta_{r}, M) \\
+ \Delta C_{L_{SS}} + C_{L_{SS}}(\alpha, M) \delta_{SB} + \Delta C_{L_{a}}(\alpha) \Delta l g \\
+ \frac{\bar{e}}{2V} \left[ C_{L_{q}}(\alpha, M) + (\Delta C_{L_{q}})_f (M, h) \right] q \\
+ \frac{\bar{e}}{2V} \left[ C_{L_{q}}(\alpha, M) + (\Delta C_{L_{q}})_f (M, h) \right] \dot{\alpha}
\]

(A.1)

For flight in the plane of symmetry:

\[
\beta = \delta_{a} = \delta_{r} = 0
\]

(A.2)

And assuming landing gear up ($\Delta l g = 0$), no speed brake deflection ($\delta_{SB} = 0$) and no extra external stores ($\Delta C_{L_{SS}} = 0$):

\[
C_L = C_{L_{q}}(\alpha, M) + (\Delta C_{L_{q}})_f (\alpha, M, h) + \Delta C_{L_{h}}(\alpha, \delta_h, M) \\
+ \left[ C_{L_{\text{lef}}}(\alpha, M) + (\Delta C_{L_{\text{lef}}})_f (\alpha, M, h) \right] \delta_{\text{lef}} \\
+ \left[ C_{L_{\text{lef}}}(\alpha, M) + (\Delta C_{L_{\text{lef}}})_f (\alpha, M, h) \right] \delta_{\text{def}} \\
+ \frac{\bar{e}}{2V} \left[ C_{L_{q}}(\alpha, M) + (\Delta C_{L_{q}})_f (M, h) \right] q \\
+ \frac{\bar{e}}{2V} \left[ C_{L_{q}}(\alpha, M) + (\Delta C_{L_{q}})_f (M, h) \right] \dot{\alpha}
\]

(A.3)
A.1.2 Drag Coefficient

\[ C_D = f(C_L, \alpha, M, \delta_h, \delta_r, \delta_{\text{lef}}, \delta_{\text{lef}}, \delta_{SB}, \Delta l g) \]

\[ = C_{D_b}(\alpha, M) + (\Delta C_D)_{C_l}(C_L, M) + \Delta C_{Dh}(\alpha, \delta_h, M) \]

\[ + C_{D_{\text{lef}}}(\alpha, M)\delta_{\text{lef}} + C_{D_{\text{ref}}}(\alpha, M)\delta_{\text{ref}} \]

\[ + \Delta C_{Dh}(\alpha, \delta_r, M) + C_{D_{SB}}(\alpha, M)\delta_{SB} \]

\[ + \Delta C_{D_b}(\alpha)\Delta l g \]  

(A.4)

For flight in the plane of symmetry:

\[ \delta_r = 0 \]  

(A.5)

And assuming landing gear up (\(\Delta lg = 0\)), no speed brake deflection (\(\delta_{SB} = 0\)) and no extra external stores (\(\Delta C_{DS} = 0\)):

\[ C_D = C_{D_b}(\alpha, M) + (\Delta C_D)_{C_l}(C_L, M) + \Delta C_{Dh}(\alpha, \delta_h, M) \]

\[ + C_{D_{\text{lef}}}(\alpha, M)\delta_{\text{lef}} + C_{D_{\text{ref}}}(\alpha, M)\delta_{\text{ref}} \]  

(A.6)
A.1.3 Pitching Moment Coefficient

\[ C_m = f(\alpha, \beta, q, M, h, \dot{\alpha}, \delta_h, \delta_a, \delta_r, \delta_{lef}, \delta_{ref}, \delta_{SB}, \Delta lg) \]

\[ = C_{m_0}(\alpha, M) + (\Delta C_{m_0})(\alpha, M, h) + \Delta C_{m_{ue}}(\alpha, M) \]

\[ + \Delta C_{m_{uh}}(\alpha, \delta_h, M) + \left[ C_{m_{uf}}(\alpha, M) + (\Delta C_{m_{uf}})(\alpha, M, h) \right] \delta_{lef} \]

\[ + \left[ C_{m_{uf}}(\alpha, M) + (\Delta C_{m_{uf}})(\alpha, M, h) \right] \delta_{ref} + \delta C_{m_p}(\alpha, \beta, M) \]

\[ + \Delta C_{m_{se}}(\alpha, \delta_a, M) + \Delta C_{m_{se}}(\alpha, \delta_r, M) + \Delta C_{m_{ee}} \]

\[ + C_{m_{se}}(\alpha, M) \delta_{SB} + \Delta C_{m_{se}}(\alpha) \Delta lg \]

\[ + \frac{\bar{c}}{2V} \left[ C_m(\alpha, M) + (\Delta C_m)(M, h) \right] q \]

\[ + \frac{\bar{c}}{2V} \left[ C_m(\alpha, M) + (\Delta C_m)(M, h) \right] \dot{\alpha} \quad (A.7) \]

For flight in the plane of symmetry:

\[ \beta = \delta_a = \delta_r = 0 \quad (A.8) \]

And assuming landing gear up (\( \Delta lg = 0 \)), no speed brake deflection (\( \delta_{SB} = 0 \)) and no extra external stores (\( \Delta C_{m_{ee}} = 0 \)):

\[ C_m = C_{m_0}(\alpha, M) + (\Delta C_{m_0})(\alpha, M, h) + \Delta C_{m_{se}}(\alpha, M) \]

\[ + \Delta C_{m_{uh}}(\alpha, \delta_h, M) + \left[ C_{m_{uf}}(\alpha, M) + (\Delta C_{m_{uf}})(\alpha, M, h) \right] \delta_{lef} \]

\[ + \left[ C_{m_{uf}}(\alpha, M) + (\Delta C_{m_{uf}})(\alpha, M, h) \right] \delta_{ref} \quad (A.9) \]

\[ + \frac{\bar{c}}{2V} \left[ C_m(\alpha, M) + (\Delta C_m)(M, h) \right] q \]

\[ + \frac{\bar{c}}{2V} \left[ C_m(\alpha, M) + (\Delta C_m)(M, h) \right] \dot{\alpha} \]

Appendix A. F/A - 18 Fighter Escort Model
A.1.4 Flap Schedule

The deflection of the leading edge flaps ($\delta_{le}$) and the trailing edge flaps ($\delta_{tr}$) is determined by a set of subroutines based on Mach number, altitude and angle of attack.

\[ \delta_{le} = f(\alpha, M, h) \quad (A.10) \]

\[ \delta_{tr} = f(\alpha, M, h) \quad (A.11) \]

The flap schedule is used to alleviate a discontinuity in the aerodynamic model.

A.2 Thrust Model

The F/A - 18 engine model calculates the thrust produced by each engine as a function of altitude, Mach number and throttle setting. Fuel flow and other dynamic engine parameters are calculated as well, but are not used in the present study.

\[ T = f(h, M, \delta_{\gamma}) \quad (A.12) \]

The engines on the F/A - 18 are positioned at an angle of 1.98 degrees from the center line of the aircraft. Therefore it is necessary to consider the reduction of thrust due to this situation. Of interest in the present study is the total thrust in the x - direction $T_x$.

\[ T_x = 2T \cos(1.98^\circ) \quad (A.13) \]
A.3 F/A - 18 Constants

The F/A - 18 fighter escort configuration is characterized by:

- 60% Fuel
- Wing tip missiles
- Total weight \( W = 33,310 \text{ lb.} \)
- Mass \( m = 1034.47 \text{ slugs} \)
- Wing area \( S = 400 \text{ ft}^2 \)
- Wing span \( b = 37.42 \text{ ft.} \)
- Mean aerodynamic chord \( \bar{c} = 11.52 \text{ ft.} \)

Aerodynamic Center:
- x-location wrt cg \( l_x = -0.297 \text{ ft.} \)
- z-location wrt cg \( l_z = 0.23 \text{ ft.} \)

Thrust Centerline:
- z-location wrt cg \( l_w = 0.23 \text{ ft.} \)

Control surface limits are:
- Stabilator \(-24^\circ \leq \delta_h \leq 10.5^\circ \) (+ for trailing edge down)
- Trailing edge flap \(-8^\circ \leq \delta_{\text{w}} \leq 45^\circ \) (+ for trailing edge down)
- Leading edge flap \(-3^\circ \leq \delta_{\text{w}} \leq 34^\circ \) (+ for trailing edge down)

Throttle limits are:
- \( 30^\circ \leq \delta_T \leq 106.5^\circ \)
- Idle: Maximum Military Thrust
- \( 106.5^\circ \leq \delta_T \leq 131^\circ \)
- Minimum: Maximum Afterburner
Appendix B. Modified F/A - 18 Model

The NASA Aerodynamic Model is not well suited for use with a nonlinear programming routine. This is primarily due to the linear interpolation used in the model. Therefore, a B-spline interpolation is created using the NASA model as a basis. Several modifications to the NASA model are necessary to make the B-spline useful and to correct the discontinuity in the NASA model at transonic Mach numbers and low altitudes.

### B.1 Modifications

Assuming:

\[
C_L = (C_{L})_{q=0} + \frac{\bar{c}}{2V} \left[ C_{Lq} + (\Delta C_{Lq})_{j} \right] q + \frac{\bar{c}}{2V} \left[ C_{Lq} + (\Delta C_{Lq})_{j} \right] \dot{\alpha}
\]

\[
= (C_{L})_{q=0} + (C_{Lq})_{m} \; q + (C_{Lq})_{M} \; \dot{\alpha}
\]  

\[(B.1)\]
\[ C_m = (C_m)_{q=0} + \frac{\bar{e}}{2V} [C_m + (\Delta C_m)_f]_q + \frac{\bar{e}}{2V} [C_m + (\Delta C_m)_f]_a \]

\[ = (C_m)_{q=0} + (C_m)_m + (C_m)_M \dot{\alpha} \]  

(B.2)

where:

\[ (C_L)_{q=0} = f(\alpha, \delta_h, M, \delta_{lef}, \delta_{rlef}, h) \]

\[ = C_{L_q}(\alpha, M) + (\Delta C_{L_q})_f(\alpha, M, h) + \Delta C_{L_{q-h}}(\alpha, \delta_h, M) \]

\[ + [C_{L_{q-h}}(\alpha, M) + (\Delta C_{L_{q-h}})_f(\alpha, M, h)]\delta_{lef} \]

\[ + [C_{L_{q-h}}(\alpha, M) + (\Delta C_{L_{q-h}})_f(\alpha, M, h)]\delta_{rlef} \]  

(B.3)

\[ (C_m)_{q=0} = f(\alpha, \delta_h, M, \delta_{lef}, \delta_{rlef}, h) \]

\[ = C_{m_q}(\alpha, M) + (\Delta C_{m_q})_f(\alpha, M, h) + \Delta C_{m_{q-h}}(\alpha, M) \]

\[ + \Delta C_{m_{q-h}}(\alpha, \delta_h, M) \]

\[ + [C_{m_{q-h}}(\alpha, M) + (\Delta C_{m_{q-h}})_f(\alpha, M, h)]\delta_{lef} \]

\[ + [C_{m_{q-h}}(\alpha, M) + (\Delta C_{m_{q-h}})_f(\alpha, M, h)]\delta_{rlef} \]  

(B.4)

\[ (C_{L_q})_M = f(\alpha, M, h) \]

\[ = \frac{\bar{e}}{2V} [C_{L_q}(\alpha, M) + (\Delta C_{L_q})_f(M, h)] \]  

(B.5)

\[ (C_{m_q})_M = f(\alpha, M, h) \]

\[ = \frac{\bar{e}}{2V} [C_{m_q}(\alpha, M) + (\Delta C_{m_q})_f(M, h)] \]  

(B.6)

\[ (C_{L_q})_M = f(\alpha, M, h) \]

\[ = \frac{\bar{e}}{2V} [C_{L_q}(\alpha, M) + (\Delta C_{L_q})_f(M, h)] \]  

(B.7)
\[(C_m)_M = f(x, M, h)\]
\[= \frac{\delta}{2V} \left[ C_{m_\alpha}(x, M) + (\Delta C_{m_\alpha})_f (M, h) \right] \quad (B.8)\]

The utility of having the lift and pitching moment coefficient in this form will be apparent in the description of the B-spline.

In order to obtain the data in this form, it is necessary to make following modifications to the outputs of NASA subroutine “REFAERO”:

1) The subroutine is called with pitch rate equal to zero. The lift coefficient \((C_L)\) returned is equal to \((C_L)_{\alpha=0}\) and the pitching moment coefficient \((C_m)\) returned is equal to \((C_m)_{\alpha=0}\).

2) Coefficients \(C_{L_\alpha}, C_{m_\alpha}, (\Delta C_{L_\alpha})_f\) and \((\Delta C_{m_\alpha})_f\) are added as outputs.

3) Coefficients \((C_{L_{\alpha}})_M\) and \((C_{m_{\alpha}})_M\) are calculated as defined above.

4) Coefficients \((C_{L_{\alpha}})_M\) and \((C_{m_{\alpha}})_M\) are used as they are output by “REFAERO”.

These coefficients are calculated, as defined above, inside the subroutine “REFAERO”.

5) A correction factor is added to the lift, drag, and pitching moment coefficients for Mach numbers above 0.6 and angles of attack below forty degrees \((40^\circ)\). This is necessary to correct a discontinuity which occurs in this region.

\[(C_L)_{\alpha=0} = (C_L)_{\alpha=0}(x, \delta_h, M, \delta_{left}, \delta_{left}, h) + \Delta C_{L_{CORR}}(\delta_h, M, \delta_{left}, \delta_{left}, h) \quad (B.9)\]

\[C_D = C_D(x, \delta_h, M, C_L, \delta_{left}, \delta_{left}) + \Delta C_{D_{CORR}}(\delta_h, M, C_L, \delta_{left}, \delta_{left}) \quad (B.10)\]

\[(C_m)_{\alpha=0} = (C_m)_{\alpha=0}(x, \delta_h, M, \delta_{left}, \delta_{left}, h) + \Delta C_{m_{CORR}}(\delta_h, M, \delta_{left}, \delta_{left}, h) \quad (B.11)\]

where:

Appendix B. Modified F/A - 18 Model 164
\[\Delta C_{L\text{CORR}} = \begin{cases} 0 & M \leq 0.5 \\ (C_L)_{\alpha=40^\circ} - (C_L)_{\alpha=39^\circ} & M > 0.5, \alpha < 40^\circ \\ 0 & M > 0.5, \alpha \geq 40^\circ \end{cases} \]  
(B.12)

\[\Delta C_{D\text{CORR}} = \begin{cases} 0 & M \leq 0.5 \\ C_D(\alpha = 40^\circ) - C_D(\alpha = 39^\circ) & M > 0.5, \alpha > 40^\circ \\ 0 & M > 0.5, \alpha \geq 40^\circ \end{cases} \]  
(B.13)

\[\Delta C_{m\text{CORR}} = \begin{cases} 0 & M \leq 0.5 \\ (C_m)_{\alpha=40^\circ} - (C_m)_{\alpha=39^\circ} & M > 0.5, \alpha < 40^\circ \\ 0 & M > 0.5, \alpha \leq 40^\circ \end{cases} \]  
(B.14)

The correction factors are calculated in a separate subroutine.

**B.2 B - Spline Interpolation**

The B-spline interpolation routines used were developed by the International Mathematical and Statistical Library (IMSL, Ref. 6). This process consists of the use of four subroutines to create and evaluate the B-spline. The algorithm is based on one by De Boor (Ref. 6). The subroutines listed in there order of use are:

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. &quot;BSNAK&quot;</td>
<td>Computes the required knot sequence</td>
</tr>
<tr>
<td>2. &quot;BS3IN&quot;</td>
<td>Computes 3 dimensional spline coefficients</td>
</tr>
<tr>
<td>or &quot;BS2IN&quot;</td>
<td>Computes 2 dimensional spline coefficients</td>
</tr>
</tbody>
</table>
3. "BS3VL" Evaluates spline interpolant (3 dimensional)
or "BS2VL" Evaluates spline interpolant (2 dimensional)
4. "BS3DR" Evaluates the derivative of the spline interpolant (3 dimensional)
or "BS2DR" Evaluates the derivative of the spline interpolant (2 dimensional)

The first two subroutines are used in subroutine "BCSMOD" to set up the spline coefficients which are passed to the subroutine "COEFF" for evaluation. The coefficients are passed using common block "SPLINE". "COEFF" is the subroutine which is called by all programs to get the aerodynamic coefficients.

The subroutine "BCSMOD" is used to implement the modifications discussed in part A) and is only called once in order to set up the spline. Subroutine "COEFF" can then be used to evaluate the spline using the last two subroutines above.

The following is a discussion of the details concerning the choice of knots and the limits imposed on the modified model.

**B.2.1 Lift Coefficients**

\[
C_L = (C_{L})_{q=0} + (C_{L})_{Mq} + (C_{L})_{Ma}
= f(\alpha, \delta_h, M, \delta_{tef}, \delta_{tef}, h)
\]  \hspace{1cm} (B.15)

The leading edge and trailing edge flaps are scheduled as a function of angle of attack, Mach number, and altitude. Therefore \((C_{L})_{q=0}\) is actually a function of four variables: angle of attack, stabilator deflection, Mach number, and altitude. A three dimensional B-spline is used to model \((C_{L})_{q=0}\) for a given altitude.
The coefficients \((C_{L_q})_M\) and \((C_{L_q})_N\) are modelled using a two dimensional B-spline for each altitude. This is possible because these coefficients are not a function of stabilator deflection. The nodes chosen for each coefficient are given in section B.2.4.

### B.2.2 Drag Coefficient

The drag coefficient is modelled using a three dimensional B-spline for a given altitude. This coefficient is not a function of pitch rate or angle of attack derivative and therefore it is unnecessary to model the corresponding coefficients. The node sequence is given in section B.2.4.

### B.2.3 Pitching Moment Coefficients

The pitching moment coefficient is modelled using the same scheme as with the lift coefficient.

\[
C_m = (C_m)_{q=0} + (C_{m_q})_M \dot{q} + (C_{L_q})_M \ddot{q} \\
= f (\alpha, \delta_h, M, \delta_{lef}, \delta_{tef}, h)
\]

As with the lift coefficient, \((C_m)_{q=0}\) is modelled using a three dimensional B-spline for each altitude. The coefficients \((C_{m_q})_M\) and \((C_{L_q})_M\) are modelled using two dimensional B-spline. See section B.2.4 for the node sequence used.
B.2.4 Node Sequences

The following tables contain the node sequences used to create the modified F/A - 18 Model.
Table 2. Stabilator Nodes (deg)

<table>
<thead>
<tr>
<th>$(C_L)_{en0}$</th>
<th>$C_D$</th>
<th>$(C_m)_{en0}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-24</td>
<td>-24</td>
<td>-24</td>
</tr>
<tr>
<td>-12</td>
<td>-12</td>
<td>-12</td>
</tr>
<tr>
<td>-6</td>
<td>-6</td>
<td>-11</td>
</tr>
<tr>
<td>-3</td>
<td>-3</td>
<td>-10</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-9</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>-8</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>-7</td>
</tr>
<tr>
<td>10.5</td>
<td>10.5</td>
<td>-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.5</td>
</tr>
</tbody>
</table>
Table 3. Mach Number Nodes

<table>
<thead>
<tr>
<th>$(C_L)_{q=0}$</th>
<th>$C_{D_1}$</th>
<th>$(C_m)_{q=0}$</th>
<th>$(C_L)_M^*$</th>
<th>$(C_m)_M^*$</th>
<th>$(C_{Lz})_M^*$</th>
<th>$(C_{mz})_M^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.10</td>
<td>0.20</td>
<td>0.40</td>
<td>0.60</td>
<td>0.80</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.90</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4. Angle of Attack Nodes (deg)

<table>
<thead>
<tr>
<th>(C_L)_{\alpha=0}</th>
<th>C_D</th>
<th>(C_m)_{\alpha=0}</th>
<th>(C_{Lq})_M</th>
<th>(C_{nq})_M</th>
<th>(C_{Lq})_M</th>
<th>(C_{nq})_M</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
</tr>
<tr>
<td>-8</td>
<td>-8</td>
<td>-8</td>
<td>-5</td>
<td>-5</td>
<td>-8</td>
<td>-8</td>
</tr>
<tr>
<td>-6</td>
<td>-6</td>
<td>-6</td>
<td>2</td>
<td>2</td>
<td>-6</td>
<td>-6</td>
</tr>
<tr>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>4</td>
<td>4</td>
<td>-4</td>
<td>-4</td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>6</td>
<td>6</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>11</td>
<td>12</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>13</td>
<td>14</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>16</td>
<td>16</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>18</td>
<td>18</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>19</td>
<td>20</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>21</td>
<td>22</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>14</td>
<td>22</td>
<td>24</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>26</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>18</td>
<td>28</td>
<td>28</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
<td>32</td>
<td>30</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>22</td>
<td>36</td>
<td>32</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>24</td>
<td>38</td>
<td>34</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>26</td>
<td>44</td>
<td>36</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>28</td>
<td>46</td>
<td>38</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>30</td>
<td>48</td>
<td>39</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>32</td>
<td>50</td>
<td>41</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>34</td>
<td>34</td>
<td>34</td>
<td>55</td>
<td>42</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>36</td>
<td>36</td>
<td>35</td>
<td>60</td>
<td>45</td>
<td>36</td>
<td>35</td>
</tr>
<tr>
<td>38</td>
<td>38</td>
<td>36</td>
<td>65</td>
<td>47</td>
<td>38</td>
<td>36</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>37</td>
<td>70</td>
<td>48</td>
<td>50</td>
<td>37</td>
</tr>
<tr>
<td>55</td>
<td>55</td>
<td>38</td>
<td>75</td>
<td>49</td>
<td>55</td>
<td>38</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>39</td>
<td>80</td>
<td>51</td>
<td>60</td>
<td>39</td>
</tr>
<tr>
<td>65</td>
<td>65</td>
<td>40</td>
<td>85</td>
<td>55</td>
<td>65</td>
<td>40</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>41</td>
<td>90</td>
<td>60</td>
<td>70</td>
<td>41</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>42</td>
<td>64</td>
<td>75</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>80</td>
<td>43</td>
<td>66</td>
<td>80</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>85</td>
<td>44</td>
<td>70</td>
<td>85</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>90</td>
<td>45</td>
<td>75</td>
<td>90</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table 5. Model Limits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$-10^\circ \leq \alpha \leq 90^\circ$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$-24^\circ \leq \delta \leq 10.5^\circ$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0.05 \leq M \leq 0.90$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix C. Equations of Motion

The equations of motion for the longitudinal flight of the aircraft are developed using Stalford (Ref. 14). Assuming flight in the symmetric plane and that the center of gravity is the origin of the body axes, the equations of motion are given as:

\[ \ddot{u} = -qw - g \sin \theta + \frac{T_x}{m} + \frac{\rho V^2 S}{2m} C_x \]  
(\text{C.1})

\[ \dot{w} = qu + g \cos \theta + \frac{\rho V^2 S}{2m} C_z \]  
(\text{C.2})

\[ \dot{q} = \frac{1}{l_{ij}} \left[ l_{xe} T_x + \frac{\rho V^2 S}{2} (\bar{c} C_m + l_z C_x - l_x c_z) \right] \]  
(\text{C.3})

\[ \dot{\theta} = q \]  
(\text{C.4})

\[ \dot{h} = u \sin \theta - w \cos \theta \]  
(\text{C.5})

The transformation to wind axes is used as follows:
\[ \alpha = \tan^{-1}\frac{w}{u} \quad \text{(C.6)} \]

\[ V^2 = u^2 + w^2 \quad \text{(C.7)} \]

\[ \bar{q} = \frac{1}{2} \rho V^2 \quad \text{(C.8)} \]

\[ C_x = C_L \sin \alpha - C_d \cos \alpha \quad \text{(C.9)} \]

\[ C_z = -C_L \cos \alpha - C_D \sin \alpha \quad \text{(C.10)} \]

Using this transformation, the equations of motion become:

\[ mV\ddot{\alpha} = mVq + mg \cos(\theta - \alpha) - \bar{q}SC_L - T_x \sin \alpha \quad \text{(C.11)} \]

\[ m\ddot{V} = -mg \sin(\theta - \alpha) - \bar{q}SC_D + T_x \cos \alpha \quad \text{(C.12)} \]

\[ I_\theta \ddot{\theta} = \bar{q}S[\bar{C}_m + (l_x \cos \alpha + l_z \sin \alpha)C_L + (l_x \sin \alpha - l_z \cos \alpha)C_D] \quad \text{(C.13)} \]

\[ \dot{\theta} = q \quad \text{(C.14)} \]

\[ \dot{h} = V \sin(\theta - \alpha) \quad \text{(C.15)} \]

letting

\[ d_L(\alpha) = l_x \cos \alpha + l_z \sin \alpha \quad \text{(C.16)} \]

\[ d_P(\alpha) = l_z \sin \alpha - l_x \cos \alpha \quad \text{(C.17)} \]

\[ \gamma = \theta - \alpha \quad \text{(C.18)} \]
it follows that:

\[
\dot{\alpha} = q + \frac{g}{V} \cos \gamma - \frac{\bar{q}S}{mV} C_L - \frac{T_x}{mV} \sin \alpha \tag{C.19}
\]

\[
\dot{V} = -g \sin \gamma - \frac{\bar{q}S}{m} C_D + \frac{T_x}{m} \cos \alpha \tag{C.20}
\]

\[
\dot{q} = \frac{1}{I_y} \left[ l_e T_x + \bar{q} s(\bar{c} C_m + d_L C_L + d_D C_D) \right] \tag{C.21}
\]

\[
\dot{\theta} = q \tag{C.22}
\]

\[
\dot{\gamma} = q - \dot{\alpha} \tag{C.23}
\]

\[
\dot{h} = V \sin \gamma \tag{C.24}
\]

We remark that the F/A-18 aerodynamic coefficients $C_m$, $C_L$ and $C_D$ are defined with respect to the aerodynamic center. Consequently, we need the last two terms in equation (C.21) to account for the moment about the center of gravity rather than the aerodynamic center.

In the half-loop maneuver, a special definition for the pitch angle ($\theta$) is used. It is allowed to have values from $0^\circ$ to $360^\circ$. Therefore, for values greater than $90^\circ$, it no longer is defined in the conventional manner. This definition makes it necessary to define flight path angle ($\gamma$) in the same manner.
Appendix D. Computer Programs

```
F/A-18 FEEDBACK FOR F/A-18 FEEDBACK SUBROUTINES

THIS FILE CONTAINS FEEDBACK SUBROUTINES FOR THE F/A-18 AIRCRAFT

F/A-18 FEEDBACK SUBROUTINE

THIS SUBROUTINE RETURNS THE FEEDBACK CONTROL FOR HALF-LOOP AND SPLIT-S MANEUVER.

INPUTS - ALPDEG  ANGLE OF ATTACK  (deg)
         GAMDEG  FLIGHT PATH ANGLE  (deg)
         GAMMAF  FINAL FLIGHT PATH ANGLE (deg)
         MACH    MACH NUMBER
         Q       PITCH RATE        (rad/s)
         IPORT   FEEDBACK INDICATOR

= 0  INITIALIZATION
= 1  FIRST TRANSITION : PART I
= 2  FIRST TRANSITION : PART II
= 3  FIRST TRANSITION : PART III
= 4  OUTER LAYER
= 5  FINAL TRANSITION
= 0  END OF MANEUVER

ALT  ALTITUDE  (ft)
AKM  MANEUVER INDICATOR
     = 1. FOR HALF-LOOP
```
SUBROUTINE FEEDBACK(DHT, ASTAR, QDSTAR, MACH, ALPDEG, GAMDEG, GAMMAF, ALT, IPORT, AKM)

REAL MACH
DTR = ACOS(-1.) / 180.

IF(IPORT .EQ. 0) THEN
IERF = 0
IPORT = 1
END IF

GAMREF = GAMDEG * AKM
CALL OLAYER(DHSTAR, ASTAR, QDSTAR, GAMREF, MACH, ALT, IERF)

CALL FINLAY(QHAT, ALPDEG, GAMDEG, MACH, IPORT)
QSTAR = QDSTAR * DTR
GOTO(100, 200, 300, 400, 500, IPORT)

100 CONTINUE
CALL FTRANS(DELALP, ALPDEG, GAMDEG, Q, QSTAR, MACH)
QCHK = ABS(Q - QHAT)
ACHK = ABS(ASTAR - ALPDEG)
IF(QCHK .LE. 1.E-4 .OR. ACHK .LE. DELALP) THEN
IPORT = 2
GOTO 200
ELSE
IF(Q .LT. QHAT) THEN
IF(ALPDEG .LT. 45.0 .OR. ALPDEG .GT. 55.0) THEN
DHT = -24.0
ELSE
  DHT = 10.5
END IF
ELSE
  IF(ALPDEG .LT. 45.0 .OR. ALPDEG .GT. 55.0) THEN
    DHT = 10.5
  ELSE
    DHT = -24.0
  END IF
END IF
END IF
RETURN

C
----------------------------------------
C FIRST TRANSITION: PART II
C

200 CONTINUE
CALL FTRANS(DELABP ,
  1 ALPDEG ,GAMDEG ,Q ,QSTAR ,MACH )
ACHK = ABS(ASTAR - ALPDEG)
IF(ACHK .LT. DELALP) THEN
  IPORT = 3
  GOTO 300
ELSE
  IF(ALPDEG .LT. ASTAR) THEN
    IF(ALPDEG .LT. 45.0 .OR. ALPDEG .GT. 55.0) THEN
      DHT = -24.0
    ELSE
      DHT = 10.5
    END IF
  ELSE
    IF(ALPDEG .LT. 45.0 .OR. ALPDEG .GT. 55.0) THEN
      DHT = 10.5
    ELSE
      DHT = -24.0
    END IF
  END IF
END IF
END IF
RETURN

C
----------------------------------------
C FIRST TRANSITION: PART III
C

300 CONTINUE
QCHK = ABS(Q - QSTAR)
ACHK = ABS(ALPDEG - ASTAR)
IF(QCHK .LE. 1.E-3 .AND. ACHK .LE. 3.E-1) THEN
  IPORT = 4
  GOTO 400
ELSE
  IF(Q .LT. QSTAR) THEN
    DHT = -24.0
  END IF
ELSE
    DHT = 10.5
END IF
END IF
RETURN
C
-------------------------------------
C   OUTER LAYER
C
-------------------------------------
400 CONTINUE
    CALL STRANS(DELGAM ,
    1   ALPDEG,GAMDEG,Q   ,MACH)
    GAMCHK = GAMMAF - DELGAM
    IF(GAMREF .GE. GAMCHK) THEN
        IPORT = 5
        GOTO 500
    ELSE
        DHT = DHSTAR
    END IF
    RETURN
C
-------------------------------------
C   FINAL TRANSITION
C
-------------------------------------
500 CONTINUE
    IF(ABS(Q) .LT. 1.E-3) THEN
        IPORT = 0
    ELSE
        DHT = 10.5
    END IF
    RETURN
END

-------------------------------------
C   F/A-18 FIRST INNER LAYER SUBROUTINE
C
C   THIS SUBROUTINE CALCULATES THE PITCH RATE FOR
C   MAXIMUM d(alpha)/dt IN THE FIRST INNER LAYER.
C
C   INPUTS - ALPDEG  ANGLE OF ATTACK (deg)
C             GAMDEG  FLIGHT PATH ANGLE (deg)
C             MACH    MACH NUMBER
C             IPORT   FEEDBACK INDICATOR
C
C   OUTPUTS - QHAT   PITCH RATE (rad/s)
C             FOR MAXIMUM d(alpha)/dt
C
C   AUTHOR - FREDERICK E. GARRETT, JR.
C            GRADUATE RESEARCH ASSISTANT
C            AEROSPACE ENGINEERING
C            VPI & SU, BLACKSBURG, VA

Appendix D. Computer Programs 179
SUBROUTINE FINLAY(QHAT,)
    =f(
        1 ALPDEG, GAMDEG, MACH, IPORT )
REAL MACH
REAL MASS
COMMON/FCONT /
  1 BETDEG, ALT, SPSD, RHO, P, R ,
  2 HGCL, DAR, DAL, DRR, DRL, DSBK ,
  3 DLG, DEBUG, QSE
COMMON/CONST /
  1 G, DUMMY, DTR
COMMON/F18FC /
  1 CBAR, B, S, E1Y, ELZ, ALZ ,
  2 ALX, MASS
COMMON/ANISC /
  1 ARM, TX, QBAR, KD ER

DETERMINE CONTROL

IF (I PORT .GT. 3) THEN
   DHT = 10.5
ELSE
   DHT = -24.0
END IF

CALL COEFF(CL, CD, CH, CLQ, CMQ ,
  1 CLO, CDO, CHO, CLAD, CMAD ,
  2 CLA, CDA, CMA, CLDH, CDDH ,
  3 CMDH, CLOA, CM0A, CLODH, CMODH ,
  4 CLQA, CMQA ,
  5 ALPDEG, DHT, MACH, 0. )

ALP = ALPDEG * DTR
GAMMA= GAMDEG * DTR
DL = ALX * COS(ALP) + ALZ * SIN(ALP)
DD = ALX * SIN(ALP) - ALZ * COS(ALP)
VEL = MACH * SPSD
ADOT = (1. - QBAR * S * CLQ / (MASS * VEL)) * Q
  1 + G / VEL * COS(GAMMA)
  2 - QBAR * S * CLO / (MASS * VEL)
  3 - TX * SIN(ALP) / (MASS * VEL)
ADOT = ADOT / (1. + QBAR * S * CLAD / (MASS * VEL))
QHAT = -(ELZ * TX + QBAR * S

Appendix D. Computer Programs
1       *(CBAR *(CMO + CMAD * ADOT)  
2          + DL *(CL0 + CMAD * ADOT)  
3         + DD * CDO))               
4       / (QBAR * S *(CBAR * CMQ + DL * CLQ))
RETURN
END

C-------------------------------------------------------------
C  F/A-18 FIRST TRANSITION SUBROUTINE
C-------------------------------------------------------------
C  THIS SUBROUTINE CALCULATES THE VALUES USED TO
C  DETERMINE WHEN THE END OF THE FIRST TRANSITION
C  HAS BEEN REACHED.
C
C  INPUTS - ALPDEG     ANGLE OF ATTACK      (deg)
C           GAMDEG     FLIGHT PATH ANGLE     (deg)
C           Q         PITCH RATE            (rad/s)
C           QSTAR     OUTER LAYER PITCH RATE (rad/s)
C           MACH      MACH NUMBER
C
C  OUTPUTS - DELALP    DELTA ALPHA          (deg)
C
C  AUTHOR - FREDERICK E. GARRETT, JR.
C          GRADUATE RESEARCH ASSISTANT
C          AEROSPACE ENGINEERING
C          VPI & SU, BLACKSBURG, VA
C
C  DATE - APRIL 1988
C-------------------------------------------------------------
SUBROUTINE FTRANS(DELALP ,
  =f(
    1       ALPDEG ,GAMDEG ,Q ,QSTAR ,
    2      MACH   )
REAL MACH  
REAL MASS
COMMON/FCONT /  
   1       BETDEG ,ALT  ,SPSD ,RHO  ,P   ,R   ,
   2       HGCL  ,BAR  ,DAL  ,DRR  ,DRL  ,DSBK ,
   3       DLG   ,DEBUG ,QSE
COMMON/CONST /  
   1      G      ,DUMMY  ,DTR
COMMON/F18FC /  
   1       CBAR  ,B   ,S   ,EIY  ,ELZ  ,AL2  ,
   2       ALX   ,MASS
COMMON/AMISC /  
   1      AKM   ,TX   ,QBAR  ,KDER
C-------------------------------------------------------------
C DETERMINE CONTROL SETTING
C-------------------------------------------------------------
IF(Q .LT. QSTAR) THEN
  DHT = -24.0
ELSE
  DHT = 10.5
END IF

CALL COEFF(CL, CD, CM, CLQ, CMQ, 
1 CL0, CD0, CM0, CLAD, CMAD, 
2 CLA, CDA, CMA, CLDH, CDDH, 
3 CMDH, CLOA, CMQA, CLODH, CMODH, 
4 CLQA, CMQA, 
5 ALPDEG, DHT, MACH, 0.)

CALCULATE F1 AND F2

ALP = ALPDEG * DTR
GAMMA = GAMDEG * DTR
DL = ALX * COS(ALP) + ALZ * SIN(ALP)
DD = ALX * SIN(ALP) - ALZ * COS(ALP)
VEL = MACH * PSD
ADOT = (1. - QBAR * S * CLQ / (MASS * VEL)) * Q
1 + G / VEL * COS(GAMMA)
2 - QBAR * S * CLO / (MASS * VEL)
3 - TX * SIN(ALP) / (MASS * VEL)
ADOT = ADOT / (1. + QBAR * S * CLAD / (MASS * VEL))
F1 = QBAR * S * (CBAR * CMQ + DL * CLQ) / E1Y
F2 = (ELZ * TX + QBAR * S
1 * (CBAR * CM0 + DL * CLO + DD * CD0
2 + (CBAR * CMAD + DL * CLAD) * ADOT))
3 / (F1 * E1Y)

DELT

DELT = LOG(Abs((QSTAR + F2) / (Q + F2))) / F1

DELP

F3 = 1. - QBAR * S * CLQ / (MASS * VEL)
F3 = F3 / (1. + QBAR * S * CLAD / (MASS * VEL))
F4 = G / VEL * COS(GAMMA)
1 - QBAR * S * CLO / (MASS * VEL)
2 - TX * SIN(ALP) / (MASS * VEL)
F4 = F4 / (1. + QBAR * S * CLAD / (MASS * VEL))
DELP = (F4 - F3 * F2) * DELT
1 + F3 * (F2 + Q) * (EXP(F1 * DELT) - 1.) / F1
DELP = DELP / DTR
RETURN
END

C

F/A-18 SECOND TRANSITION SUBROUTINE

C

THIS SUBROUTINE CALCULATES THE VALUES USED TO
DETERMINE WHEN THE END OF THE SECOND TRANSITION
HAS BEEN REACHED.

C

INPUTS - ALPDEG   ANGLE OF ATTACK (deg)
          GAMDEG   FLIGHT PATH ANGLE (deg)
          Q        PITCH RATE (rad/s)
          MACH     MACH NUMBER

C

OUTPUTS - DELGAM   DELTA GAMDEG (deg)

C

AUTHOR - FREDERICK E. GARRETT, JR.
GRADUATE RESEARCH ASSISTANT
AEROSPACE ENGINEERING
VPI & SU, BLACKSBURG, VA

C

DATE - APRIL 1988

C

SUBROUTINE STRANS(DELGAM ,
                  =f( 
   1 ALPDEG,GAMDEG,Q,MACH )
REAL MACH
REAL MASS
COMMON/FCONT /  
1 BETDEG,ALT,SPSD,RHO,P,R ,
2 HGCL,DAR,DAL,DRR,DRL,DSBK ,
3 DLG,DEBUG,QSE
COMMON/CONST /  
1 G ,DUMMY,DTR
COMMON/F18FC /  
1 CBAR,B,S,EIY,ELZ,ALZ ,
2 ALX,MASS
COMMON/AMISC /  
1 AKM,TX,QBAR,KDER

C

DETERMINE CONTROL SETTING

C

DHT = 10.5

C

CALCULATE MODEL

C

CALL COEFF(CL ,CD ,CM ,CLQ ,CMQ ,
  1 CL0 ,CD0 ,CM0 ,CLAD ,CMAD ,
  2 CLA ,CDA ,CMA ,CLDH ,CDDH ,

Appendix D. Computer Programs 183
```c
3   CMDH ,CLOA ,CMQA ,CLODH ,CMODH ,
4   CLOA ,CMQA ,
5   ALPDEG ,DHT ,MACH ,0. )

C CALCULATE F1 AND F2
     -----------------------------------
C ALP = ALPDEG * DTR
C GAMMA= GAMDEG * DTR
C DL = ALX * COS(ALP) + ALZ * SIN(ALP)
C DD = ALX * SIN(ALP) - ALZ * COS(ALP)
C VEL = MACH * SPD
C ADOT = (1. - QBAR * S * CLO / (MASS * VEL)) * Q
C         + G / VEL * COS(GAMMA)
1 2 - QBAR * S * CLO / (MASS * VEL)
3    - TX * SIN(ALP) / (MASS * VEL)
C ADOT = ADOT / (1. + QBAR * S * CLO / (MASS * VEL))
C F1 = QBAR * S * (CBAR + CMO + DL * CLO) / E1Y
C F2 = (ELZ + TX + QBAR * S
1 2 + (CBAR + CMAD + DL * CLAD) * ADOT))
    / (F1 * E1Y)

C DELT
     -----------------------------------
C DELT = LOG( F2 / (Q + F2)) / F1

C DELGAM
     -----------------------------------
C F5 = QBAR * S * CLO / (MASS * VEL)
C F6 = - G * COS(GAMMA) / VEL
C 1 + QBAR * S * (CLO + CLAD + ADOT) / (MASS * VEL)
C 2 + TX * SIN(ALP) / (MASS * VEL)
C DELGAM = (F6 - F5 * F2) * DELT
C 1 + F5 * (F2 + Q) * (EXP(F1 * DELT) - 1) / F1
C DELGAM = DELGAM / DTR
RETURN
END

C--------------------------------------

C F/A-18 OUTER LAYER SUBROUTINE
C
C THIS SUBROUTINE CALCULATES THE FEEDBACK CONTROL
C FOR THE OUTER LAYER
C
C INPUTS - GAMDEG          FLIGHT PATH ANGLE (deg)
C           MACH            MACH NUMBER
C           ALT            ALTITUDE (ft)
C           IERF  = 0 READ INTERPOLATION DATA
C                       = 1 DO INTERPOLATION

Appendix D. Computer Programs

184
SUBROUTINE OILAYER(DHSTAR,ASTAR,QSTAR,
                  GAMDEG,MACH,ALT,IREF)

COMMON/OLTBL/
 1 ALPHA,Q,DHT
REAL MACH
REAL ALPHA ( 19, 7, 6)
REAL Q ( 19, 7, 6)
REAL DHT ( 19, 7, 6)
REAL AS ( 2)
REAL QS ( 2)
REAL DHS ( 2)
CHARACTER*80 HEADER
IF(IREF.EQ.1) GOTO 100

READ INTERPOLATION DATA

DO 50 K = 1,6
   DO 40 J = 1,7
      READ(30,5) HEADER
5      FORMAT(80A)
      DO 30 I = 1,19
         READ(30,10) ALPHA(I,J,K),Q(I,J,K),DHT(I,J,K)
30      CONTINUE
40      CONTINUE
50      CONTINUE
IREF = 1

DETERMINE INTERPOLATION NODES

100 I = INT(GAMDEG / 10.) + 1
   IF(I.LT. 1) I = 1
   IF(I.GE. 19) I = 18
   GA = FLOAT(I-1) * 10.0
   GA1 = GA + 10.0
   J = INT(MACH * 10.) - 2
AMACH = MACH
IF(J .LT. 1) THEN
  J = 1
  AMACH = 0.3
END IF
IF(J .GE. 7) THEN
  J = 6
  AMACH = 0.9
END IF
AM = FLOAT(J+2) / 10.
AM1 = AM + 0.1
K = INT((ALT / 1000.) - 5.) / 10.) + 1
IF(K .LT. 1) K = 1
IF(K .GE. 6) K = 5
AL = (FLOAT(K-1) * 10. + 5.) * 1000.
AL1 = AL + 10000.
DO 200 I = 1, 2
   M = K + L - 1
  C-----------------------------------------------
C INTERPOLATE FOR OUTER LAYER ANGLE OF ATTACK (ASTAR)
C-----------------------------------------------
ALJ = ATERP(GA,GA1,
1                  ALPHA(I,J,M),ALPHA(I+1,J,M),
2                  GAMDEG )
ALJ1 = ATERP(GA,GA1,
1                  ALPHA(I,J+1,M),ALPHA(I+1,J+1,M),
2                  GAMDEG )
AS(L) = ATERP(AM,AM1,
1                  ALJ,ALJ1,
2                  AMACH )
C-----------------------------------------------
C INTERPOLATE FOR OUTER LAYER Q (QSTAR)
C-----------------------------------------------
QJ = ATERP(GA,GA1,
1                  Q(I,J,M),Q(I+1,J,M),
2                  GAMDEG )
QJ1 = ATERP(GA,GA1,
1                  Q(I,J+1,M),Q(I+1,J+1,M),
2                  GAMDEG )
QS(L) = ATERP(AM,AM1,
1                  QJ,QJ1,
2                  AMACH )
C-----------------------------------------------
C INTERPOLATE FOR OUTER LAYER DHT (DHSTAR)
C-----------------------------------------------
DHJ = ATERP(GA,GA1,
1                  DHT(I,J,M),DHT(I+1,J,M),
2                  GAMDEG )
DHJ1 = ATERP(GA,GA1,
1                  DHT(I,J+1,M),DHT(I+1,J+1,M),
2                  GAMDEG )

Appendix D. Computer Programs 186
2      GAMDEG
      DHS(L) = ATERP(AM
1      DHJ
2      AMACH

200 CONTINUE
      ASTAR = ATERP(AL
1      AS(1)
2      ALT
QSTAR = ATERP(AL
1      QS(1)
2      ALT
      DHSTAR = ATERP(AL
1      DHS(1)
2      ALT
      RETURN
END

C---------------------------------------------
C  LINEAR INTERPOLATION FUNCTION <FEG>
C---------------------------------------------
FUNCTION ATERP(X1 ,X2 ,Y1 ,Y2 ,X )
      ATERP = Y2 + (X - X2) * (Y1 - Y2) / (X1 - X2)
      RETURN
END

C---------------------------------------------
F/A-18 BASS-GURA FORMULA POLE PLACEMENT SUBROUTINE
C
   THIS SUBROUTINE CALCULATES THE FEEDBACK GAINS
NECESSARY TO PLACE THE POLES OF THE SYSTEM EQUAL
TO THE SECOND ORDER BUTTERWORTH POLES.
C
   i.e. CHARACTERISTIC EQUATION GIVEN BY:
      2 + sqrt[lambda*omega + lambda + omega] 2 = 0
C
   SUBROUTINE COEFF MUST BE CALLED BEFORE CALLING
   THIS SUBROUTINE
C
INPUTS - ALPDEG     ANGLE OF ATTACK (deg)
         DHT       STABILATOR DEFLECTION (deg)
         MACH      MACH NUMBER
         OMEGA     SPEED OF THE BUTTERWORTH POLES
            thalf    ]
            ] = ln(2) / [sqrt(2) * omega]
            tdouble ]
            period  = 1 / [sqrt(2) * omega]
C
OUTPUTS - K           GAIN ROW VECTOR OF DIMENSION 2
         K( 1 , 1 ) => GAIN FOR delta ALPHA
         K( 1 , 2 ) => GAIN FOR delta Q
SUBROUTINE BASGURA(K ,IERR ,
                   ALPDEG ,MACH ,OMEGA )

COMMON/FCONT /
  1 BETDEG ,ALT ,SPSD ,RHO ,P ,R ,
  2 HGCL ,DAR ,DAL ,DRR ,DRL ,DSBK ,
  3 DLG ,DEBUG ,QSE

COMMON/CONST /
  1 G ,DUMMY ,DTR

COMMON/F18FC /
  1 CBAR ,B ,S ,EIY ,ELZ ,ALZ ,
  2 ALX ,MASS

COMMON/AOFF /
  1 CL ,CD ,CM ,CLQ ,CMQ ,CLO ,
  2 CDO ,CMO ,CLAD ,CMAD ,CLA ,CDA ,
  3 CMA ,CLDH ,CDDH ,CMDH ,CLOA ,CMOA ,
  4 CLDHDH ,CMODH ,CLQA ,CMQA

COMMON/AMISC /
  1 AKM ,TX ,QBAR ,KDER

REAL MACH
REAL MASS
REAL A ( 2 , 2 )
REAL AB ( 2 , 1 )
REAL BB ( 2 , 1 )
REAL KD ( 1 , 2 )
REAL O ( 1 , 2 )
REAL U ( 2 , 2 )
REAL UW ( 2 , 2 )
REAL W ( 2 , 2 )
REAL CA ( 2 )
REAL OA ( 2 )
REAL K ( 1 , 2 )
LOGICAL QSE
LOGICAL DEBUG

C-------------------------------

Appendix D. Computer Programs
C INITIALIZE

V = MACH * SPSD
ALP = ALPDEG * DTR
PL1 = ALX * COS(ALP) + ALZ * SIN(ALP)
PL2 = ALX * SIN(ALP) - ALZ * COS(ALP)
ITRR = 0
AMV = MASS * V

C CALCULATE SYSTEM MATRIX A

C

C d(d(alpha)/dt)/d(alpha)
C
A(1, 1) = - QBAR * S * CLA / AMV
2 = TX * COS(ALP) / AMV

C d(d(alpha)/dt)/d(q)
C
A(1, 2) = 1 - QBAR * S * CLQ / AMV

C d(d(q)/dt)/d(alpha)
C
A(2, 1) = QBAR * S / E1Y
1 = (CBAR * CMA
2 + PL1 * CLA
3 + PL2 * CDL
4 - (ALX * CL - ALZ * CD) * SIN(ALP)
5 + (ALX * CD + ALZ * CL) * COS(ALP))

C d(d(q)/dt)/d(q)
C
A(2, 2) = QBAR * S / E1Y
1 = (CBAR * CMQ
2 + PL1 * CLQ
3 + PL2 * CDQ)

C CALCULATE CONTROL MATRIX BB

C d(d(alpha)/dt)/d(dht)
C
BB(1, 1) = - QBAR * S * CLDH / AMV

C d(d(q)/dt)/d(dht)
C
BB(2, 1) = QBAR * S / E1Y
1 = (CBAR * CMDH
2 + PL1 * CLDH

Appendix D. Computer Programs
3 + PL2 * CDDH)

C ----------------------------------------------
C CALCULATE COEFFICIENT VECTORS
C ----------------------------------------------
C OPEN LOOP COEFFICIENT VECTOR OA
C
OA( 1 ) = -(A( 1, 1 ) + A( 2, 2 ))
OA( 2 ) = A( 1, 1 ) * A( 2, 2 )
1 - A( 1, 2 ) * A( 2, 1 )
C
C CLOSED LOOP COEFFICIENT VECTOR CA
C
CA( 1 ) = SQRT( 2. ) * OMEGA
CA( 2 ) = OMEGA * OMEGA
C
C CALCULATE THE TOTAL COEFFICIENT MATRIX O
C
O ( 1 , 1 ) = CA( 1 ) - OA( 1 )
O ( 1 , 2 ) = CA( 2 ) - OA( 2 )
C
C TEST FOR CONTROLLABILITY
C
C
C CALCULATE CONTROLLABILITY MATRIX U
C
U( 1 , 1 ) = BB( 1 , 1 )
U( 2 , 1 ) = BB( 2 , 1 )
CALL MULT( AB , 2 , 2 , 1 , A , BB )
U( 1 , 2 ) = AB( 1 , 1 )
U( 2 , 2 ) = AB( 2 , 1 )
C
C DETERMINE RANK OF U
C
DETU = U( 1 , 1 ) * U( 2 , 2 )
1 - U( 1 , 2 ) * U( 2 , 1 )
IF( DETU .EQ. 0.) THEN
  IERR = 100
  RETURN
END IF
C
C INVERT CONTROLLABILITY MATRIX U
C
U11 = U( 2, 2 ) / DETU
U12 = - U( 2, 1 ) / DETU
U21 = - U( 1, 2 ) / DETU
U22 = U( 1, 1 ) / DETU
U( 1 , 1 ) = U11
U( 1 , 2 ) = U12
U( 2 , 1 ) = U21
U( 2, 2 ) = u22

C U IS NOW THE INVERSE OF THE
C CONTROLLABILITY MATRIX
C
C CREATE BASS-GURA MATRIX W
C NOTE:
C THIS MATRIX IS ACTUALLY THE TRANSPOSE
C OF THE MATRIX W IN THE ALGORITHM
C
C W( 1, 1 ) = 1.
W( 1, 2 ) = OA( 1 )
W( 2, 1 ) = 0.
W( 2, 2 ) = 1.

C INVERT W
C
C DETW = W( 1, 1 ) * W( 2, 2 )
1 - W( 1, 2 ) * W( 2, 1 )
W11 = W( 2, 2 ) / DETW
W12 = - W( 2, 1 ) / DETW
W21 = - W( 1, 2 ) / DETW
W22 = W( 1, 1 ) / DETW

C W IS NOW EQUAL TO THE INVERSE OF THE
C TRANSPOSE OF W IN THE ALGORITHM
C
C CALCULATE THE GAIN MATRIX K
C
C CALL MULT(WU , 2 , 2 , 2 , W , u )
CALL MULT(K , 1 , 2 , 2 , 0 , WU )

10 CONTINUE
RETURN
END

C
C MATRIX MULTIPLICATION SUBROUTINE <FEG>
C
C INPUTS:
C
C N ROW DIMENSION OF MATRIX A
C K COLUMN DIMENSION OF MATRIX A
C M COLUMN DIMENSION OF MATRIX B
C A n x k MATRIX
C B k x m MATRIX

OUTPUTS:

AB AB(n x m) = A(n x k) * B(k x m)

SUBROUTINE MULT(AB ,
C =f( 1
     N , K , M , A , B )
REAL A ( N , K )
REAL B ( K , M )
REAL AB ( N , M )
DO 30 I = 1 , N
   DO 20 J = 1 , M
      AB( I , J ) = 0.
      DO 10 L = 1 , K
         AB( I , J ) = AB( I , J )
         + A( I , L ) * B( L , J )
   10 CONTINUE
20 CONTINUE
30 CONTINUE
RETURN
END

F/A-18 OUTER LAYER STABILITY SUBROUTINE

THIS SUBROUTINE CALCULATES THE EIGENVALUES OF
LINEARIZED EQUATIONS FOR THE FAST VARIABLES
IN THE OUTER LAYER

INPUTS - ALPDEG ANGLE OF ATTACK (deg)
          MACH MACH NUMBER

OUTPUTS - A SYSTEM MATRIX
          A( 1 , 1 ) = d( d(alpha)/dt ) / d(alpha)
          A( 1 , 2 ) = d( d(alpha)/dt ) / d(q)
          A( 2 , 1 ) = d( d(q)/dt ) / d(alpha)
          A( 2 , 2 ) = d( d(q)/dt ) / d(q)
          ALAMR1 REAL PART OF FIRST EIGENVALUE
          ALAMI1 IMAGINARY PART OF FIRST EIGENVALUE
          ALAMR2 REAL PART OF SECOND EIGENVALUE
          ALAMI2 IMAGINARY PART OF SECOND EIGENVALUE

AUTHOR - FREDERICK E. GARRET, JR.
GRADUATE RESEARCH ASSISTANT
AEROSPACE ENGINEERING
VPI & SU, BLACKSBURG, VA
DATE - MARCH 1988

SUBROUTINE STBLTY (ALPDEG, MACH)

COMMON/FCONT/
1 BETDEG , ALT , SPSD , RHO , P , R ,
2 HGCL , DAR , DAL , DRR , DRL , DSBK ,
3 DLG , DEBUG , QSE

COMMON/CONST/
1 G , DUMMY , DTR

COMMON/F18FC/
1 CBAR , B , S , EIY , ELZ , ALZ ,
2 ALX , MASS

COMMON/ACOFF/
1 CL , CD , CM , CLQ , CMQ , CL0 ,
2 CDO , CMO , CLAB , CMAD , CLA , CDA ,
3 CMA , CLDH , CDDH , CMDH , CL0A , CM0A ,
4 CL0DH , CMODH , CLQA , CMQA

COMMON/AMISC/
1 AKM , TX , QBAR , KDER

REAL MACH
REAL MASS
REAL A(2,2)

INITIALIZATION

V = MACH * SPSD
ALP = ALPDEG * DTR

CALCULATE SYSTEM MATRIX

d( d(alpha)/dt ) / d(alpha)

A(1,1) = - QBAR * S * CLA / (MASS * V)
2  = TX * COS(ALP) / (MASS * V)

A(1,2) = 1 - QBAR * S * CLQ / (MASS * V)

A(2,1) = QBAR * S / EIY
1 * (CBAR * CMA
2 + (ALX * COS(ALP) + ALZ * SIN(ALP)) * CLA
3 + (ALX * SIN(ALP) - ALZ * COS(ALP)) * CDA

Appendix D. Computer Programs 193
4       - (ALX * CL - ALZ * CD) * SIN(ALP)
5       + (ALX * CD + ALZ * CL) * COS(ALP))
--------------------
C       d( d(q)/dt ) / d(q)
--------------------
A(2,2) = QBAR * S / EIY
1       * (CBAR * CMQ
2       + (ALX * COS(ALP) + ALZ * SIN(ALP)) * CLQ)
--------------------
C       CALCULATE EIGENVALUES
C       det(LAMDA * I - A) = 0.
C
C       or
C       2 LAMQUAD * LAMDA + QUADC = 0.
--------------------
QUADA  = 1.
QUADB  = - (A(1,1) + A(2,2))
QUADC  = A(1,1) * A(2,2) - A(1,2) * A(2,1)
RAD    = QUADB * QUADB - 4 * QUADA * QUADC
IF(RAD .LT. 0.) THEN
   ALAMR1 = - QUADB / (2 * QUADA)
   ALAMR2 = ALAMR1
   ALAMI1 = SQRT(-RAD) / (2 * QUADA)
   ALAMI2 = - ALAMI1
ELSE
   ALAMR1 = -( QUADB + SQRT(RAD) ) / (2 * QUADA)
   ALAMR2 = -( QUADB - SQRT(RAD) ) / (2 * QUADA)
   ALAMI1 = 0.
   ALAMI2 = 0..
END IF
RETURN
END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++  
C+++  F18_SIMULATION.FOR  F/A-18 SIMULATION PROGRAM  
C+++  +++
C+++  PURPOSE - TO SIMULATE THE LONGITUDINAL  
C+++  +++
C+++  FLIGHT OF THE F/A-18 FIGHTER  
C+++  +++
C+++  ESCORT CONFIGURATION  
C+++  +++
C+++  PRECISION - SINGLE  
C+++  +++
C+++  REQUIRED -  
C+++  +++
C+++  F18_SUB.FOR  
C+++  +++
C+++  F18_AERO_TABLES.DAT  
C+++  +++
C+++  F18_ENG_TABLES.DAT  
C+++  +++
C+++  IMSL MATH/LIBRARY  
C+++  +++
C+++  AUTHOR - FREDERICK E. GARRETT, JR.  
C+++  +++
C+++  GRADUATE RESEARCH ASSISTANT  
C+++  +++
C+++  AEROSPACE ENGINEERING DEPT.  
C+++  +++
COMMON/FCONT /
  1  BETDEG, ALT, SPSD, RHO, P, R, 
  2  HGCL, DAR, DAL, DRR, DRL, DSBK, 
  3  DLG, DEBUG, QSE
COMMON/CONST /
  1  G, DUMMY, DTR
COMMON/F18FC /
  1  CBAR, B, S, E1Y, ELZ, ALZ, 
  2  ALX, MASS
COMMON/AOFFC /
  1  CL, CD, CM, CLQ, CMQ, CLO, 
  2  CDO, CMO, CLAD, CMAD, CLA, CDA, 
  3  CMA, CLDH, CDDH, CMDH, CLOA, CM0A, 
  4  CLODH, CMODH, CLQA, CMQA
COMMON/AMISC /
  1  AKM, TX, QBAR, KDUR
COMMON/WORKSP /
  1  RWKSP
REAL MACH
REAL MASS
REAL A ( 2, 2 )
REAL K ( 1, 2 )
REAL RWKSP ( 5137 )
REAL YPRIME ( 6 )
REAL Y ( 6 )
LOGICAL DEBUG
LOGICAL QSE
CHARACTER*80 HEADER
CHARACTER*10 MANVER
CALL IWKIN ( 5137 )

C-------------------------------
C _INITIALIZATION
C-------------------------------
C
C  GENERAL CONSTANTS
C-------------------------------
G   = 32.174
DUMMY = 1.
DTR   = ACOS(-1.) / 180.
KINV  = 0
KDER  = 1

C-------------------------------
C  FLIGHT CONDITIONS
C-------------------------------
BETDEG = 0.
ALT = 15000.
CALL ATMOS(RHO,SPSD,ALT)
P = 0.
R = 0.
HGCL = ALT

---

C
ENGINE PARAMETERS
---

WDIF = DUMMY
GK1 = DUMMY
P1 = DUMMY
P2 = DUMMY
TAU = DUMMY
TESTV = DUMMY
VAE = DUMMY

---

C
CHOOSE MANEUVER
---

AKM = 1. => HALF-LOOP
AKM = -1. => SPLIT-S

---

AKM = 1.
IF( AKM .EQ. 1. ) THEN
   MANVER = 'HALF-LOOP'
ELSE
   MANVER = 'SPLIT-S'
END IF

---

C
F/A-18 CONSTANTS FIGHTER ESCORT CONFIGURATION
---

CBAR = 11.52
XBAR = 18.0
ZBAR = 6.0
B = 37.42
S = 400.0
E11 = 151293.0
E12 = 2.8 / 12.0
E22 = 2.8 / 12.0
ALX = -3.56 / 12.0
MASS = 1034.47

---

C
IGNORABLE CONTROL SETTINGS
---

DAR = 0.
DAL = 0.
DRR = 0.
DRL = 0.
DSBK = 0.
DL2 = 0.
DTH = 130.
LOGICAL DECLARATIONS FOR SUBROUTINE REFAERO

DEZUG = .FALSE.
QSE  = .TRUE.
CALL IAERO (HEADER)
CALL IENG (HEADER)

INITIAL STATES AND CONTROLS

MACH  = 0.6
IF(MACH .EQ. 0.3) THEN
   ALPDEG = 11.68
   DHT   = -1.675
END IF
IF(MACH .EQ. 0.6) THEN
   ALPDEG = 2.884
   DHT   = -0.1561
END IF
IF(MACH .EQ. 0.9) THEN
   ALPDEG = 1.314
   DHT   = 0.5223
END IF
Y(1) = ALPDEG * DTR
Y(2) = MACH * SPSD
QD   = 0.0
Y(3) = QD * DTR
THETAD = ALPDEG
THOUT = THETAD
GAMDEG = 0.0
Y(4) = THETAD * DTR
Y(5) = ALT
Y(6) = 0.0
T    = 0.0
TRIM  = 0.0

INITIALIZE INTEGRATION SUBROUTINE

DT   = 1.E-2
KSTEP = 4 * INT(1.E-2 / DT + 1.E-6)
IORT  = 0
IERF  = 0
NOPOLE = 1
GAMMAF = 180.
OMEGA = 1.5
N     = 6
INTG  = 1
DO 10 I = 1,N
   YPRIME(I) = 0.
10 CONTINUE
CALCULATE B-SPLINE COEFFICIENTS

USING SUBROUTINE BCSMOD

CALL BCSMOD

INITIALIZE OUTPUT FILES

WRITE(10,11) MANVER ,ALT
11 FORMAT( 'F/A-18 SIMULATION (STATES)' , / ,
       1 ' FOR THE ',A10,' MANEUVER' , / ,
       2 ' AT ',F9.2,' ft' , / ,
       3 ' t(sec) alpha(deg) v(ft/s) q(deg/s)' ,
       4 ' theta(deg) MACH' , / )

WRITE(11,12) MANVER ,ALT
12 FORMAT( 'F/A-18 SIMULATION (STATE DERIVATIVES)' , / ,
       1 ' FOR THE ',A10,' MANEUVER' , / ,
       2 ' AT ',F9.2,' ft' , / ,
       3 ' t(sec) d(alpha)/dt d(v)/dt(dt/f/s/s)' ,
       4 ' d(q)/dt d(theta)/dt d(z)/dt d(x)/dt' , / )

WRITE(12,13) MANVER ,ALT
13 FORMAT( 'F/A-18 SIMULATION (FEEDBACK DATA)' , / ,
       1 ' FOR THE ',A10,' MANEUVER' , / ,
       2 ' AT ',F9.2,' ft' , / ,
       3 ' t(sec) dh*(deg) alpha*(deg) q*(deg/s)' , / )

WRITE(13,14) MANVER ,ALT
14 FORMAT( 'F/A-18 SIMULATION (LOAD FACTORS)' , / ,
       1 ' FOR THE ',A10,' MANEUVER' , / ,
       2 ' AT ',F9.2,' ft' , / ,
       32 )
       3 ' t(sec) anz(g)' s' anx(g)' s' dynamic pressure(lb/ft
       4 ' altitude(ft) range(ft) theta(deg)' , / )

WRITE(15,51) MANVER ,ALT
51 FORMAT( 'F/A-18 SIMULATION (GAINS AND CONTROLS)' , / ,
       1 ' FOR THE ',A10,' MANEUVER' , / ,
       2 ' AT ',F9.2,' ft' , / ,
       3 ' t(sec) Kalpha Kq dh(deg) IPORT' , / )

SIMULATION TIME LOOP

Y( 1 ) => ANGLE OF ATTACK (rad)
Y( 2 ) => VELOCITY (ft/s)
Y( 3 ) => PITCH RATE (rad/s)
Y( 4 ) => PITCH ANGLE (rad)
Y( 5 ) => ALTITUDE (ft)
Y( 6 ) => RANGE (ft)

DO 20 I = 1,100000

UPDATE VELOCITY TERMS
QBAR = 0.5 * RHO * Y(2) * Y(2)
CO2VT = CBAR / (2 * Y(2))
BO2VT = B / (2 * Y(2))

C CALCULATE THRUST

CALL ENG1 (TH ,WDTF ,GK1 ,P1 ,P2 ,
1 TAU ,TESTV ,TMIL ,TAUGMN ,DTH ,
2 MACH ,ALT ,VAE )

TX = 2 * TH * COS(1.98 * DTR)

C CALL FEEDBACK SUBROUTINE

IF(T .GE. TTRIM) THEN
QOF = Y(3)
CALL FEDBCK(DHSTAR ,ASTAR ,QSTAR ,
1 ALPDEG ,GAMDEG ,GAMMAF ,MACH ,
2 QOF ,ALT ,IPORT ,AKM )
IF(NOPOLE .EQ. 0 .AND. IPORT .EQ. 4) THEN

C POLE PLACEMENT

DA = Y(1) - ASTAR * DTR
DQ = Y(3) - QSTAR * DTR
CALL BASGURA(K ,IERR ,ALPDEG ,
1 MACH ,OMEGA )
DHT = DHSTAR - ( K(1,1) * DA
1 + K(1,2) * DQ ) / DTR
IF( DHT .GT. 10.5 ) DHT = 10.5
IF( DHT .LT. -24.0 ) DHT = -24.0
ELSE
DHT = DHSTAR
END IF
END IF

C CALCULATE AERODYNAMIC COEFFICIENTS

CALL COEFF (CL ,CD ,CM ,CLQ ,CMQ ,
1 CL0 ,CDO ,CMO ,CLA ,CMAD ,
2 CLA ,CDA ,CMA ,CLD ,CDDH ,
3 CMDH ,CLOA ,CMOA ,CLDH ,CMODH ,
4 CLIQ ,CMQA ,
5 ALPDEG ,DHT ,MACH ,0. )

C CALL DIFFERENTIAL EQUATIONS

CALL RHS(N ,T ,Y ,YPRIME )

C CALCULATE FAST EQUATION EIGENVALUES
IF( IPORT .EQ. 4 ) THEN
  CALL STBLTY(A , ALAMR1 , ALAMI1 , ALAMR2 , ALAMI2 ,
             ALPDIC , MACH )
END IF

C -----------------------------------------------
C CALCULATE LOAD FACTORS
C -----------------------------------------------
ANZG = QBAR * S * (CD0 * SIN(Y1))
  + (CLO + CLQ * Y3 + CLAD * YPRIME(1) * COS(Y1))
  / (MASS * G)
ANZP = (XBAR * YPRIME(3) + ZBAR * Y(3) * Y(3)) / G
ANZ  = ANZG + ANZP
ANXG = QBAR * S * (- CD0 * COS(Y1))
  + (CLO + CLQ * Y3 + CLAD * YPRIME(1) * SIN(Y1))
  / (MASS * G)
ANXP = (- XBAR * Y(3) * Y(3) + ZBAR * YPRIME(3)) / G
ANX  = ANXG + ANXP

C -----------------------------------------------
C OUTPUT EVERY 10th STEP
C -----------------------------------------------
IF( MOD((I-1),KSTEP) .EQ. 0 ) THEN
  WRITE(10,15) T, ALPDIC , Y(2) , QD , THETAD ,
           MACH
15    FORMAT( F7.3 , 4F12.4 , F9.3 )
  WRITE(11,16) T , YPRIME(1), YPRIME(2), YPRIME(3), YPRIME(4),
           YPRIME(5), YPRIME(6)
16    FORMAT( F7.3 , 6F14.7 )
  TTOW = TX / 33310.
  WRITE(11,16) T , TX , TTOW
17    FORMAT( F7.3 , 2F14.7 )
  WRITE(12,17) T , DHSTAR , ASTAR , QSTAR
18    FORMAT( F7.3 , 6F11.3 )
  WRITE(14,55) T , ANZG , ANXG , ANZP , ANXP
55    FORMAT( F7.3 , 4F12.3 )
  WRITE(15,56) T , K(1,1) , K(1,2) , DHT , GAMDEG ,
1     IPORT
56    FORMAT( F7.3 , 4F14.7 , I3)
  IF( IPORT .EQ. 4 ) THEN
    WRITE(16,57) T , ALAMR1 , ALAMR2
57    FORMAT( F7.3 , 2F14.7 )
  WRITE(17,58) T , ALAMI1 , ALAMI2
58    FORMAT( F7.3 , 2F14.7 )
END IF
END IF

IF( IPORT .EQ. 3 ) THEN
  DT = 1.E-3
  KSTEP = 4 * INT(1.E-2 / DT + 1.E-6)
IF(T .GT. 3.0) STOP
ELSE
    DT    = 1.E-2
    KSTEP = 4 * INT(1.E-2 / DT + 1.E-6)
END IF
IF(IPORT .EQ. 5) THEN
    K(1,1) = 0.
    K(1,2) = 0.
END IF
IF(GAMDEG .GT. 200.0) STOP
IF(IPORT .EQ. 0) STOP

CALL INTEGRATION ROUTINE

CALL XGRAT(N ,Y ,YPRIME,INTG ,T ,DT )

TRANSITION FROM STATE MATRIX TO STATE VARIABLES

ALT    = Y( 5 )
CALL ATMOS(RHO ,SPSD ,ALT )
ALPDEG = Y( 1 ) / DTR
MACH   = Y( 2 ) / SPSD
QD     = Y( 3 ) / DTR
THETAD = Y( 4 ) / DTR
GAMDEG = THETAD - ALPDEG
IF(THETAD .LE. 90.0) THEN
    THOUT = THETAD
ELSE
    THOUT = 180 - THETAD
END IF
20 CONTINUE
STOP
END

DIFFERENTIAL EQUATION SUBROUTINE

PURPOSE - TO COMPUTE THE NUMERICAL APPROXIMATION
FOR THE DIFFERENTIAL EQUATIONS FOR THE
F/A-18 LONGITUDINAL MODEL
REQUIRED FOR IMSL SUBROUTINE DVERK

INPUTS - N  NUMBER OF EQUATIONS
         T  TIME    (sec)
         Y( 1 )  ANGLE OF ATTACK  (rad)
         Y( 2 )  VELOCITY  (ft/s)
         Y( 3 )  PITCH RATE  (rad/s)
         Y( 4 )  PITCH ANGLE  (rad)
         Y( 5 )  ALTITUDE  (ft)

OUTPUTS - YPRIME( I )  d( Y( I ) ) / dt
SUBROUTINE RHS(N, T, Y, YPRIME)
COMMON/CONST / 
  1    G, DUMMY, DTR 
COMMON/F18FC / 
  1    CBAR, B, S, ELY, ELZ, ALZ, 
  2    ALX, MASS 
COMMON/ACOFF / 
  1    CL, CD, CM, CLQ, CMQ, CLO, 
  2    CD0, CM0, CLAD, CHAD, CLA, CDA, 
  3    CMA, CLDH, CDDH, CMDH, CLOA, CMOA, 
  4    CLODH, CMDDH, CLQA, CMQA 
COMMON/AMISC / 
  1    AKM, TX, QBAR, KDER 
REAL MASS 
REAL MACH 
REAL Y (6) 
REAL YPRIME (6) 

Real Mass
Real Mach
Real Y (6)
Real YPrime (6)

d(Alpha)/dt
-------------------------------------
YPrime (1) = (1 - QBAR * S * CLQ / (MASS * Y(2))) * Y(3) 
  1 + (AKM * G * COS(Y(4)) - AKM * Y(1)) / Y(2) 
  2 - (QBAR * S * CLO) / (MASS * Y(2)) 
  3 - (TY * SIN(Y(1))) / (MASS * Y(2)) 

YPrime (1) = YPrime(1) / (1 + QBAR * S * CLAD 
  1 / (MASS * Y(2))) 

-------------------------------------
d(V)/dt
-------------------------------------
YPrime (2) = - G * SIN(Y(4)) - AKM * Y(1)) 
  1 - (QBAR * S * CD0) / MASS 
  2 + TX * COS(Y(1)) / MASS 

-------------------------------------
d(q)/dt
-------------------------------------
YPrime (3) = (ELZ * TX 
  1 + QBAR * S 
  2 + (CBAR * (CMO + CMQ * Y(3) 
  3 + CMDAD * YPrime(1))) 
  4 + (ALX * COS(Y(1)) + ALZ * SIN(Y(1))) 
  5 + (CLO + CLQ * Y(3) 
  6 + CLAD * YPrime(1)) 

Appendix D. Computer Programs
7 + (ALX * SIN(Y( 1 )) - ALZ * COS(Y( 1 )))
8 * CDO
9 )
&
10 ) / EiY
C--------------------------------------------------------
C d(theta)/dt
C--------------------------------------------------------
C YPRIME ( 4 ) = AKM * Y( 3 )
C--------------------------------------------------------
C d(h)/dt
C--------------------------------------------------------
C YPRIME ( 5 ) = Y( 2 ) * SIN(Y( 4 ) - Y( 1 ))
C--------------------------------------------------------
C d(x)/dt
C--------------------------------------------------------
C YPRIME ( 6 ) = Y( 2 ) * COS(Y( 4 ) - Y( 1 ))
RETURN
END
C--------------------------------------------------------
C INTEGRATION SUBROUTINE
C--------------------------------------------------------
C FOURTH ORDER RUNGE-KUTTA INTEGRATOR
C--------------------------------------------------------
C INPUTS - N NUMBER OF DIFFERENTIAL EQUATIONS
C Y(N) INITIAL STATES
C YDOT(N) DERIVATIVES OF STATES wrt TIME
C INTG INTEGRATION STEP ( =1 TO INITIALIZE)
C T TIME
C H TIME STEP
C--------------------------------------------------------
C OUTPUT - Y(N) UPDATED STATES
C INTG INTEGRATION STEP
C--------------------------------------------------------
C AUTHOR - DR. HAROLD L. STALFORD
C ASSOCIATE PROFESSOR
C AEROSPACE ENGINEERING
C VPI & SU, BLACKSBURG, VA.
C--------------------------------------------------------
SUBROUTINE XGRAT(N ,Y ,YDCT ,INTG ,T ,H )
COMMON/XINTG /
1 XK ,X ,Q
REAL XK ( 7 )
REAL Q ( 7 )
REAL Y ( 7 )
REAL YDCT ( 7 )
GOTO(10,20,30,40),INTG
10 CONTINUE
DO 15 J = 1,N
XK(J) = H * YDOT(J)
X = 0.5 * (XK(J) - 2.0 * Q(J))
Y(J) = Y(J) + X
Q(J) = Q(J) + 3.0 * X - 0.5 * XK(J)

15 CONTINUE
T = T + 0.5 * H
INTG = 2
RETURN

20 CONTINUE
DUM = 1.0 - SQRT(0.5)
DO 25 J = 1,N
   XK(J) = H * YDOT(J)
   X = DUM * (XK(J) - Q(J))
   Y(J) = Y(J) + X
   Q(J) = Q(J) + 3.0 * X - DUM * XK(J)
25 CONTINUE
INTG = 3
RETURN

30 CONTINUE
DUM = 1.0 + SQRT(0.5)
DO 35 J = 1,N
   XK(J) = H * YDOT(J)
   X = DUM * (XK(J) - Q(J))
   Y(J) = Y(J) + X
   Q(J) = Q(J) + 3.0 * X - DUM * XK(J)
35 CONTINUE
INTG = 4
T = T + 0.5 * H
RETURN

40 CONTINUE
DUM = 1.0 / 6.0
DO 45 J = 1,N
   XK(J) = H * YDOT(J)
   X = DUM * (XK(J) - 2.0 * Q(J))
   Y(J) = Y(J) + X
   Q(J) = Q(J) + 3.0 * X - 0.5 * XK(J)
45 CONTINUE
INTG = 1
RETURN
END

C
C
C F18_SUB.FOR  F/A-18 MODEL SUBROUTINES
C
C THIS FILE CONTAINS THE MODEL SUBROUTINES
C FOR THE F/A-18 AIRCRAFT
C
C COEFFICIENT MODEL SUBROUTINE
THIS SUBROUTINE USES THE B-SPLINE COEFFICIENTS GENERATED
BY SUBROUTINE 'BCS3VL' IN THE IMSL FUNCTIONS BS3VL AND
BS3DR (STORED IN COMMON BLOCK 'SPLINE') TO DETERMINE THE
COEFFICIENTS CLO, CDO, CMQ, CLQ, CMA, CMQ, CLA, CDA, CMQ
CLA, CLDH, CDDH, CL, CD, AND CM.

SUBROUTINE BCS3MOD MUST BE CALLED PRIOR TO THIS SUBROUTINE

ACCEPTABLE RANGES ARE:
-10. <= ALPDEG <= 90.
-24. <= DHT <= 10.5
0.05 <= MACH <= 0.90

INPUTS - ALPDEG  ANGLE OF ATTACK  (deg)
          DHT  STABILATOR DEFORMATION  (deg)
          MACH  MACH NUMBER
          Q  PITCH RATE  (rad/s)
          KDER  DERIVATIVE CONDITION

KDER = 0  =>  CALCULATE alpha AND dht
DERIVATIVES
KDER = 1  =>  DO NOT CALCULATE alpha
AND dht DERIVATIVES
(SAVES TIME IF THESE DERIVATIVES ARE NOT DESIRED)

OUTPUTS - CL  LIFT COEFFICIENT
            CD  DRAG COEFFICIENT
            CM  PITCHING MOMENT COEFFICIENT
            CLQ  d(CL)/d(q)  (s/rad)
            CMQ  d(CM)/d(q)  (s/rad)
            CL6  LIFT COEFFICIENT FOR q=0, d(alpha)/dt=0
            CDO  DRAG COEFFICIENT FOR q=0, d(alpha)/dt=0
            CMO  PITCHING MOMENT COEFFICIENT FOR
            q=0, d(alpha)/dt=0
            CLAD  d(CL)/d( d(alpha)/dt )  (s/rad)
            CMAD  d(CM)/d( d(alpha)/dt )  (s/rad)
            CLA  d(CL)/d( alpha )  (/deg)
            CDA  d(CD)/d( alpha )  (/deg)
            CMA  d(CM)/d( alpha )  (/deg)
            CLDH  d(CL)/d( dht )  (/deg)
            CDDH  d(CD)/d( dht )  (/deg)
            CMDH  d(CM)/d( dht )  (/deg)
            CL0A  d(CL0)/d( dht )  (/deg)
            CMQA  d(CM)/d( dht )  (/deg)
            CL0A  d(CL0)/d( alpha )  (/deg)
            CMQA  d(CM)/d( alpha )  (/deg)
            CL0A  d(CL0)/d( alpha )  (/deg)
            CMQA  d(CM)/d( alpha )  (/deg)
            2) CLQA  d(CLQ)/d( alpha )  (/rad)
            2) CMQA  d(CMQ)/d( alpha )  (/rad)
SUBROUTINE COEFF(CL ,CD ,CM ,CLQ ,CMQ ,
5   CLO ,CDO ,CMO ,CLAD ,CMAD ,
2   CLA ,CDA ,CMA ,CLDH ,CDDH ,
3   CMDH ,CLOA ,CMOA ,CL0DH ,CM0DH ,
4   CLQA ,CMQA ,
C
5   =f(
ALPDEG ,DHT ,MACH ,Q )
EXTERNAL BS3VL
EXTERNAL BS2VL
EXTERNAL BS3DR
EXTERNAL BS2DR
COMM/FCONT /
1   BETDEG ,ALT ,SPSD ,RHO ,P ,R ,
2   HSCCL ,DAR ,DAL ,DRR ,DRL ,DSBK ,
3   DLG ,DEBUG ,QSE
COMM/F18FC /
1   CBAR ,B ,S ,E1Y ,ELZ ,ALZ ,
2   ALX ,MASS
COMM/AMISC /
1   AKM ,TX ,QBAR ,KDER
COMM/SPLINE /
1   CCL ,CCD ,CCM ,CCLAD ,CCLQ ,CCMAD ,CCMQ ,
2   X ,XM ,XLQ ,XMQ ,Y ,YM ,Z ,
2   NX ,NMX ,NXLQ ,NMXQ ,NY ,NMY ,NZ ,
3   XKN0T ,XN0TM ,XN1LQ ,XN1MQ ,YKN0T ,YN0TM ,ZKNOT ,
4   KXORD ,KYORD ,KZORD ,KYORDM
REAL CCL ( 34 , 8 , 8 )
REAL CCD ( 34 , 8 , 8 )
REAL CCM ( 43 , 13 , 8 )
REAL CCLAD( 34 , 8 )
REAL CCLQ ( 30 , 8 )
REAL CCMAD( 43 , 8 )
REAL CCMQ ( 35 , 8 )
REAL X ( 34 )
REAL XM ( 43 )
REAL XLQ ( 30 )
REAL XMQ ( 35 )
REAL Y ( 8 )
REAL YM ( 13 )
REAL Z ( 8 )
REAL XKN0T( 37 )
REAL XN0TM ( 46 )
REAL XNTLQ( 33 )
REAL XNTMQ( 38 )
REAL YKNOT( 11 )
REAL YNOTM( 16 )
REAL ZKNOT( 11 )
REAL MACH

DTR = ACOS(-1.)/180.

C CALCULATE COEFFICIENTS FOR ALPDEG AND DHT
C (USING IMSL FUNCTION BS3VL)

IF(DHT .GT. 10.5) DHT = 10.5
IF(ALPDEG .GT. 90.) ALPDEG = 90.
IF(ALPDEG .LT. -10.) ALPDEG = -10.
IF(MACH .LT. 0.05) MACH = 0.05
IF(MACH .GT. 0.90) MACH = 0.90

GLO = BS3VL(ALPDEG, DHT, MACH, KXORD, KYORD, KZORD,
1        XKNOT, YKNOT, ZKNOT, NX, NY, NZ,
2        CCL)

CDO = BS3VL(ALPDEG, DHT, MACH, KXORD, KYORD, KZORD,
1        XKNOT, YKNOT, ZKNOT, NX, NY, NZ,
2        CCD)

CMO = BS3VL(ALPDEG, DHT, MACH, KXORD, KYORDM, KZORD,
1        XNOTM, YNOTM, ZKNOT, NXM, NYM, NZ,
2        CCM)

C CALCULATE DERIVATIVE COEFFICIENTS
C (USING IMSL FUNCTION BS3VL)

CLAD = BS2VL(ALPDEG, MACH, KXORD, KZORD,
1        XKNOT, ZKNOT, NX, NZ,
2        CCLAD)

CMAD = BS2VL(ALPDEG, MACH, KXORD, KZORD,
1        XNOTM, ZKNOT, NXM, NZ,
2        CCMAD)

CLQ = BS2VL(ALPDEG, MACH, KXORD, KZORD,
1        XNTLQ, ZKNOT, NXLQ, NZ,
2        CCLQ)

CMQ = BS2VL(ALPDEG, MACH, KXORD, KZORD,
1        XNTMQ, ZKNOT, NXMQ, NZ,
2        CCMQ)

C CHECK DERIVATIVE CONDITION
C

IF(KDER .EQ. 1) THEN
    CLA = 0.
    CDA = 0.
    CMA = 0.
    CLDH = 0.
CDDH = 0.
CMDH = 0.
CLOA = 0.
CMA = 0.
CLODH = 0.
CMODH = 0.
CLQA = 0.
CMQA = 0.
GOTO 10
END IF

C
C CALCULATE DERIVATIVE COEFFICIENTS
C (USING IMSL FUNCTIONS BS3DR)
C
CLOA = BS3DR(1 ,0 ,0 ,ALPDEG ,DHT ,MACH ,
1
KXORD ,KYORD ,KZORD ,XKNOT ,YKNOT ,ZKNOT ,
2
NX ,NY ,NZ ,CCL )
CDOA = BS3DR(1 ,0 ,0 ,ALPDEG ,DHT ,MACH ,
1
KXORD ,KYORD ,KZORD ,XKNOT ,YKNOT ,ZKNOT ,
2
NX ,NY ,NZ ,CDD)
CMOA = BS3DR(1 ,0 ,0 ,ALPDEG ,DHT ,MACH ,
1
KXORD ,KYORDM ,KZORD ,XNOM ,YNOM ,ZKNOT ,
2
NXM ,NYM ,NZ ,CCM)
CLODH = BS3DR(0 ,1 ,0 ,ALPDEG ,DHT ,MACH ,
1
KXORD ,KYORD ,KZORD ,XKNOT ,YKNOT ,ZKNOT ,
2
NX ,NY ,NZ ,CCL)
CDODH = BS3DR(0 ,1 ,0 ,ALPDEG ,DHT ,MACH ,
1
KXORD ,KYORD ,KZORD ,XKNOT ,YKNOT ,ZKNOT ,
2
NX ,NY ,NZ ,CDD)
CMODH = BS3DR(0 ,1 ,0 ,ALPDEG ,DHT ,MACH ,
1
KXORD ,KYORDM ,KZORD ,XNOM ,YNOM ,ZKNOT ,
2
NXM ,NYM ,NZ ,CCM)
CLQA = BS2DR(1 ,0 ,ALPDEG ,MACH ,
1
KXORD ,KZORD ,XNTLQ ,ZKNOT ,
2
NXLQ ,NZ ,CCLQ)
CMQA = BS2DR(1 ,0 ,ALPDEG ,MACH ,
1
KXORD ,KZORD ,XNTMQ ,ZKNOT ,
2
NXMQ ,NZ ,CCMQ)

C
C CALCULATE TOTAL COEFFICIENT DERIVATIVES
C
CLA = CLOA + CLQA * Q
CDA = CDOA
CM = CMA + CMQA * Q
CDLH = CLDTH
CDDH = CDODH
CMDH = CMODH

C
C PUT DERIVATIVES IN CORRECT UNITS
C

Appendix D. Computer Programs  208
CLQA = CLQA / DTR
CMQA = CMQA / DTR

C

CALCULATE TOTAL COEFFICIENTS

C

10 CL = CL0 + CLQ * Q
CD = CD0
CM = CM0 + CMQ * Q
RETURN
END

C

B-SPLINE MODEL SUBROUTINE <FEG>

C

THIS SUBROUTINE USES IMSL SUBROUTINES BS3IN AND BSNAK TO
CALCULATE B-SPLINE COEFFICIENTS (STORED IN COMMON BLOCK
SPLINE) FOR SUBROUTINE COEFF.

C

NASA SUBROUTINE IAERO MUST BE CALLED PRIOR TO THIS SUBROUTINE

C

OUTPUTS - COMMON BLOCK SPLINE

C

AUTHOR - FREDERICK E. GARRETT, JR.
GRADUATE RESEARCH ASSISTANT
AEROSPACE ENGINEERING DEPT.
VPI & SU, BLACKSBURG, VA;

C

DATE - DECEMBER 1987

C

SUBROUTINE BCSM0D
EXTERNAL BS3IN
EXTERNAL BS2IN
EXTERNAL BSNAK
COMMON/FCONT /
1 BETDEG ,ALT ,SPSD ,RHO ,P ,R ,
2 HGL ,DAR ,DAL ,DRR ,DRL ,DSBK ,
3 DLG ,DEBUG ,QSE
COMMON/F18FC /
1 CBAR ,B ,S ,EIY ,ELZ ,ALZ ,
2 ALX ,MASS
COMMON/SPLINE /
1 CCL ,CCD ,CCM ,CCLAD ,CCLQ ,CMAD ,CMQ ,
2 X ,XM ,XLQ ,XMQ ,Y ,YM ,Z ,
3 NX ,NXLQ ,NXM ,NXLQ ,NY ,NMM ,NZ ,
3 XNORT ,XNOTM ,XNXLQ ,XNMQ ,YNOT ,YNOTM ,ZKNOT ,
4 KXORD ,KYORD ,KZORD ,KORDM
REAL CCL ( 34 , 8 , 8 )
REAL CCD ( 34 , 8 , 8 )
REAL CCM ( 43 , 13 , 8 )
REAL FCL ( 34 , 8 , 8 )
<table>
<thead>
<tr>
<th>REAL  FCD</th>
<th>(34, 8, 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL  FCM</td>
<td>(43, 13, 8)</td>
</tr>
<tr>
<td>REAL  CCLAD</td>
<td>(34, 8)</td>
</tr>
<tr>
<td>REAL  CCLQ</td>
<td>(30, 8)</td>
</tr>
<tr>
<td>REAL  CCMAD</td>
<td>(43, 8)</td>
</tr>
<tr>
<td>REAL  CCMQ</td>
<td>(35, 8)</td>
</tr>
<tr>
<td>REAL  FCLA</td>
<td>(34, 8)</td>
</tr>
<tr>
<td>REAL  FCLQ</td>
<td>(30, 8)</td>
</tr>
<tr>
<td>REAL  FCMA</td>
<td>(43, 8)</td>
</tr>
<tr>
<td>REAL  FCMQ</td>
<td>(35, 8)</td>
</tr>
<tr>
<td>REAL  X</td>
<td>(34)</td>
</tr>
<tr>
<td>REAL  XM</td>
<td>(43)</td>
</tr>
<tr>
<td>REAL  XLQ</td>
<td>(30)</td>
</tr>
<tr>
<td>REAL  XMQ</td>
<td>(35)</td>
</tr>
<tr>
<td>REAL  Y</td>
<td>(8)</td>
</tr>
<tr>
<td>REAL  YM</td>
<td>(13)</td>
</tr>
<tr>
<td>REAL  Z</td>
<td>(8)</td>
</tr>
<tr>
<td>REAL  XKNOT</td>
<td>(37)</td>
</tr>
<tr>
<td>REAL  XNOTM</td>
<td>(46)</td>
</tr>
<tr>
<td>REAL  XNTLQ</td>
<td>(33)</td>
</tr>
<tr>
<td>REAL  XNTMQ</td>
<td>(38)</td>
</tr>
<tr>
<td>REAL  YKNOT</td>
<td>(11)</td>
</tr>
<tr>
<td>REAL  YNOTM</td>
<td>(16)</td>
</tr>
<tr>
<td>REAL  ZNOT</td>
<td>(11)</td>
</tr>
<tr>
<td>REAL  MASS</td>
<td></td>
</tr>
</tbody>
</table>

LOGICAL DEBUG
LOGICAL QSE

<table>
<thead>
<tr>
<th>DATA  X</th>
<th>-10., -8., -6., -4., -2., 0., 2., 4.,</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6., 8., 10., 12., 14., 16., 18., 20.,</td>
</tr>
<tr>
<td></td>
<td>22., 24., 26., 28., 30., 32., 34., 36.,</td>
</tr>
<tr>
<td></td>
<td>38., 50., 55., 60., 65., 70., 75., 80.,</td>
</tr>
<tr>
<td></td>
<td>85., 90./,</td>
</tr>
<tr>
<td></td>
<td>5 XM</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 XLQ</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 XMQ</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp; Y</td>
<td>-24., -12., -6., -3., 0., 3., 6., 10.5./,</td>
</tr>
<tr>
<td>1 YH</td>
<td>-24., -12., -11., -10., -9., -8., -7., -6.,</td>
</tr>
<tr>
<td>2</td>
<td>-3., 0., 3., 6., 10.5./,</td>
</tr>
</tbody>
</table>
3   Z/ 0.05, 0.10, 0.20, 0.40, 0.60, 0.80, 0.85, 0.90/,
4   NX/ 34 /,
5   NXM/ 43 /,
6   NXLQ/ 30 /,
7   NXMQ/ 35 /,
8   NY/ 8 /,
9   NYM/ 13 /,
&   NZ/ 8 /,
1   Q/ 0. /,
2   KXORD/ 3 /, KYORD/ 3 /, KZORD/ 3 /, KYORDM/ 3 /
C -----------------------------
C SET UP INTERPOLATION POINTS
C (USING IMSL SUBROUTINE BSNAK)
C -----------------------------
C CALL BSNAK (NX ,X ,KXORD ,XKNOT )
C CALL BSNAK (NXM ,XM ,KXORD ,XNOMT )
C CALL BSNAK (NXLQ ,XLQ ,KXORD ,XNTLQ )
C CALL BSNAK (NXMQ ,XMQ ,KXORD ,XNTMQ )
C CALL BSNAK (NY ,Y ,KYORD ,YKNOT )
C CALL BSNAK (NYM ,YM ,KYORDM ,YNOTM )
C CALL BSNAK (NZ ,Z ,KZORD ,ZKNOT )
C -----------------------------
C GET COEFFICIENT ARRAYS
C -----------------------------
DO 70 K = 1,NZ
VEL = Z(K) * SPSD
QBAR = 0.5 * RHO * VEL**VEL
CO2VT = CBAR / (2 * VEL)
BO2VT = B / (2 * VEL)
DO 20 I = 1,NX
CALL FLPSCK(Z(K) ,ALT ,X(I) ,DLFR ,DTFR )
DLFL = DLFR
DTFL = DTFR
DO 10 J = 1,NY
CALL REFAERO(FCD(I,J,K) ,CY ,FCL(I,J,K) ,C1 ,
1   CM ,CN ,CLAD ,CMAD ,
2   CLQ ,DLQF ,CMQ ,DCMQF ,
3   X(I) ,BETDEG ,Z(K) ,P ,Q ,
4   R ,ALT ,HGCL ,QBAR ,CO2VT ,
5   BO2VT ,Y(J) ,Y(J) ,DAR ,DAL ,
6   DRR ,DRL ,DTFR ,DTFL ,DLFR ,
7   DLFL ,DSBK ,DLG ,DEBUG ,QSE )
IF(Z(K) .GE. 0.6 .AND. X(I) .LT. 40.0) THEN
 CALL CORREC(CORL ,CORD ,CORM ,
1   Y(J) ,Z(K) ,QBAR ,CO2VT ,BO2VT )
 FCL(I,J,K) = FCL(I,J,K) + CORL
 FCD(I,J,K) = FCD(I,J,K) + CORD
END IF
IF(Y(J) .EQ. 0.) FCLA(I,K) = CO2VT * CLAD
10 CONTINUE
CONTINUE
DO 40 I = 1,NXM
   CALL FLPSCH(Z(K),ALT,XM(I),DLFR,DTFR)
   DLFL = DLFR
   DTFL = DTFR
DO 30 J = 1,NYM
   CALL REFAERO(CD,CY,CL,C1)
   FCM(I,J,K),CN,CLAD,CMAD
   CLQ,DCLQF,CMQ,DCMQF
   XM(I),BETDEG,Z(K),P,Q
   R,ALT,HGCL,QBAR,CO2VT
   BO2VT,YM(J),YM(J),DAR,DAL
   DRR,DRL,DTFR,DTFL,DLFR
   DLFL,DSBK,DLG,DEBUG,QSE
   IF(Z(K) .GE. 0.6 .AND. XM(I) .LT. 40.0) THEN
      CALL CORREC(CORL,CORD,CORM)
      YM(J),Z(K),QBAR,CO2VT,BO2VT
      FCM(I,J,K) = FCM(I,J,K) + CORM
   END IF
   IF(YM(J) .EQ. 0.) FCMA(I,K) = CO2VT * CMAD
30
CONTINUE
DO 50 I = 1,NXLQ
   CALL FLPSCH(Z(K),ALT,XLQ(I),DLFR,DTFR)
   DLFL = DLFR
   DTFL = DTFR
   CALL REFAERO(CD,CY,CL,C1)
   CM,CN,CLAD,CMAD
   CLQ,DCLQF,CMQ,DCMQF
   XLQ(I),BETDEG,Z(K),P,Q
   R,ALT,HGCL,QBAR,CO2VT
   BO2VT,0.,0.,DAR,DAL
   DRR,DRL,DTFR,DTFL,DLFR
   DLFL,DSBK,DLG,DEBUG,QSE
   FCLQ(I,K) = CO2VT * (CLQ + DCLQF)
50
CONTINUE
DO 60 I = 1,NXMQ
   CALL FLPSCH(Z(K),ALT,XMQ(I),DLFR,DTFR)
   DLFL = DLFR
   DTFL = DTFR
   CALL REFAERO(CD,CY,CL,C1)
   CM,CN,CLAD,CMAD
   CLQ,DCLQF,CMQ,DCMQF
   XMQ(I),BETDEG,Z(K),P,Q
   R,ALT,HGCL,QBAR,CO2VT
   BO2VT,0.,0.,DAR,DAL
   DRR,DRL,DTFR,DTFL,DLFR
   DLFL,DSBK,DLG,DEBUG,QSE
   FCMQ(I,K) = CO2VT * (CMQ + DCMQF)
60
CONTINUE
70 CONTINUE

C ------------------------------
C CALCULATE B-SPLINE COEFFICIENTS
C (USING IMSL SUBROUTINE BS3IN)
C ------------------------------

CALL BS3IN (NX , X , NY , Y , NZ , Z ,
1 FCL , NX , NY , KYORD , KXORD , KZORD ,
2 XKNOT , YKNOT , ZKNOT , CCL )
CALL BS3IN (NX , X , NY , Y , NZ , Z ,
1 FCD , NX , NY , KYORD , KXORD , KZORD ,
2 XKNOT , YKNOT , ZKNOT , CCD )
CALL BS3IN (NXM , XM , NYM , YM , NZ , Z ,
1 FCM , NXM , NYM , KYORD , KXORD , KYORDM , KZORD ,
2 XNOMT , YNOMT , ZKNOT , CCM )
CALL BS2IN (NX , X , NZ , Z ,
1 FCLA , NX , XKORD , KZORD ,
2 XKNOT , ZKNOT , CCLAD )
CALL BS2IN (NXM , XM , NXM , Z ,
1 FCMA , NXM , XKORD , KZORD ,
2 XNOMT , ZKNOT , CCMAD )
CALL BS2IN (NXLQ , XNLQ , NZ , Z ,
1 FCLQ , NXLQ , XKORD , KZORD ,
2 XNTLQ , ZKNOT , CCLQ )
CALL BS2IN (NXMQ , XMQ , NZ , Z ,
1 FCMQ , NXMQ , XKORD , KZORD ,
2 XNMQ , ZKNOT , CCMQ )
RETURN
END

C---------------------------------------
C MODEL CORRECTION SUBROUTINE
C---------------------------------------

THIS SUBROUTINE CORRECTS THE NASA MODEL
FOR HIGH ANGLES OF ATTACK

INPUTS - DH STABILATOR ANGLE (deg)
          MACH MACH NUMBER
          QBAR DYNAMIC PRESSURE (lb/ft**2)
          CO2VT CBAR / (2 * V)
          BO2VT B / (2 * V)

OUTPUTS - CORL CORRECTION IN CL
           FOR ANGLES OF ATTACK > 40 deg
           CORD CORRECTION IN CD
           FOR ANGLES OF ATTACK > 40 deg
           CORM CORRECTION IN CM
           FOR ANGLES OF ATTACK > 40 deg

AUTHOR - FREDERICK E. GARRETT, JR.
GRADUATE RESEARCH ASSISTANT
AEROSPACE ENGINEERING
C VPI & SU, BLACKSBURG, VA
C
C DATE - MARCH 1988
C
SUBROUTINE CORREC(CORL,CORD,CORM, C
  =f( DH ,MACH ,QBAR ,CO2VT ,B02VT)
   COMMON/FCONT /
  1 BETDEG ,ALT ,SPSD ,RKO ,P ,R ,
  2 HGCL ,DAR ,DAL ,DRR ,DRL ,DSBK ,
  3 DLG ,DEBUG ,QSE
LOGICAL DEBUG
LOGICAL QSE
REAL MACH
Q = 0.
C
C CALCULATE CL AT ALPDEG = 39.99999 deg
C
ALPDEG = 39.99999
CALL FLPSC(MACH ,ALT ,ALPDEG,DLFR ,DTFR )
DLFL = DLFR
DTFL = DTFR
CALL REFAERO(CD39 ,CY ,CL39 ,C1 ,
  1 CM39 ,CN ,CLAD ,CMAD ,
  2 CLQ ,DCLQF ,CMQ ,DCMQF ,
  3 ALPDEG,BETDEG ,MACH ,P ,Q ,
  4 R ,ALT ,HGCL ,QBAR ,CO2VT ,
  5 B02VT ,DH ,DH ,DAR ,DAL ,
  6 DRR ,DRL ,DTFR ,DTFL ,DLFR ,
  7 DLFL ,DSBK ,DLG ,DEBUG ,QSE )

C
C CALCULATE CL AT ALPDEG = 40.00000 deg
C
ALPDEG = 40.0
CALL FLPSC(MACH ,ALT ,ALPDEG,DLFR ,DTFR )
DLFL = DLFR
DTFL = DTFR
CALL REFAERO(CD40 ,CY ,CL40 ,C1 ,
  1 CM40 ,CN ,CLAD ,CMAD ,
  2 CLQ ,DCLQF ,CMQ ,DCMQF ,
  3 ALPDEG,BETDEG ,MACH ,P ,Q ,
  4 R ,ALT ,HGCL ,QBAR ,CO2VT ,
  5 B02VT ,DH ,DH ,DAR ,DAL ,
  6 DRR ,DRL ,DTFR ,DTFL ,DLFR ,
  7 DLFL ,DSBK ,DLG ,DEBUG ,QSE )

C
C CALCULATE CORRECTION
C
CORL = CL40 - CL39

Appendix D. Computer Programs
CORD = CD40 - CD39
CORD = CM40 - CM39
RETURN
END

C------------------------------------------------------------------------
C ATMOSPHERIC MODEL
C------------------------------------------------------------------------
SUBROUTINE ATMOS(RHO ,SPSD ,ALT )
C------------------------------------------------------------------------
C SPEED OF SOUND
C------------------------------------------------------------------------
H = ALT / 1000.
H = H / 3.28084
TEMP = 3.72E-3 * H + 0.193315
TEMP = TEMP * H - 8.87743
TEMP = TEMP * H + 292.1
SPSD = 20.0468 * SQRT(TEMP)
SPSD = SPSD / 0.3048
C------------------------------------------------------------------------
C DENSITY
C------------------------------------------------------------------------
ARG = 1.15219733E-6 * H - 8.3300053E-5
ARG = ARG * H + 3.50991865D-3
ARG = ARG * H - 3.48643241E-2
ARG = ARG * H
PHI = 1.0228055 * EXP(-ARG)
RHO = PHI - 1.0228055 - 0.12122693 * H
RHO = EXP(RHO)
RHO = RHO * 1.225
RHO = RHO * 0.02832
RHO = RHO * 0.06852
RETURN
END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C+++ F18_OUTER_LAYER.FOR  F/A-18 OUTER LAYER PROGRAM +++
C+++ +++
C+++ PURPOSE - TO CALCULATE THE OUTER LAYER +++
C+++ SOLUTION USING PARAMETER +++
C+++ OPTIMIZATION +++
C+++ +++
C+++ +++
C+++ REQUIRED - F18_SUB.FOR +++
C+++ F18_AERO_TABLES.DAT +++
C+++ F18_ENG_TABLES.DAT +++
C+++ IMSL MATH/LIBRARY +++
C+++ SLLSQP +++
C+++ DBLAS +++
C+++ +++
C+++ AUTHOR - FREDERICK E. GARRET, JR. +++
C+++ +++

Appendix D. Computer Programs
COMMON/FCONT / 
  1  BETDEG , ALT , SPSD , RHO , P , R ,
  2  HGCL , DAR , DAL , DRR , DRL , DSBK ,
  3  DLG , DEBUG , QSE
COMMON/CONST /
  1  G , DUMMY , DTR
COMMON/F18FC /
  1  CBAR , B , S , EIY , ELZ , ALZ ,
  2  ALX , MASS
COMMON/AMISC /
  1  AKM , TX , QBAR , KDER
COMMON/WORKSP /
  1  RWKSP
REAL*8 DG( 1, 2 )
REAL*8 GG( 1 )
REAL*8 X ( 2 )
REAL*8 XL( 2 )
REAL*8 XU( 2 )
REAL*8 DF( 2 )
REAL*8 W ( 150 )
REAL*8 ACC
REAL*8 ALPHA
REAL*8 DHT
REAL*8 F
REAL*8 AMV
REAL*8 GV
REAL*8 QBSM
REAL*8 QBI
REAL*8 TXI
REAL*8 TMV
REAL*8 QT
REAL*8 QB
REAL*8 QQ
REAL*8 DL
REAL*8 DD
REAL*8 QTA
REAL*8 QBA
REAL*8 QA
REAL*8 QDH
REAL*8 GAMMA
REAL*8 DMASS
REAL*8 FMASS
REAL*8 GCOST
REAL*8 GCOST
REAL*8 DBG
REAL*8 DDTR
REAL*8 DCBAR
REAL*8 DB
REAL*8 DS
REAL*8 DEIY
REAL*8 DELZ
REAL*8 DALZ
REAL*8 DALX
REAL*8 DTX
REAL*8 DQBAR
REAL*8 DV
REAL*8 DAKM
REAL MACH
REAL MASS
REAL RWKSP ( 5137 )
INTEGER INDEX ( 10 )
LOGICAL DEBUG
LOGICAL QSE
CHARACTER*80 HEADER
CHARACTER*10 MANVER
CALL IWKIN ( 5137 )

C-----------------------------------------------
C     INITIALIZATION
C-----------------------------------------------
C
C     GENERAL CONSTANTS
C-----------------------------------------------
G      = 32.174
DUMMY = 1.
DTR    = ACOS(-1.) / 180.
DBG    = DBLE ( G )
DDTR   = DBLE ( DTR )

C-----------------------------------------------
C     FLIGHT CONDITIONS
C-----------------------------------------------
BETDEG = 0.
P      = 0.
R      = 0.

C
C     CHOOSE MANEUVER
C
C     AKM = 1. => HALF-LOOP
C
C     AKM = -1. => SPLIT-S
C-----------------------------------------------
AKM    = -1.
DAKM   = DBLE ( AKM )
IF( AKM .EQ. 1. ) THEN
    MANVER = 'HALF-LOOP'
ELSE
    MANVER = 'SPLIT-S'

C

Appendix D. Computer Programs
END IF

F/A-18 CONSTANTS FIGHTER ESCORT CONFIGURATION

CBAR = 11.52
B  = 37.42
S  = 400.0
EiY = 151293.
ELZ = 0.23
ALZ = 0.23
ALX = -0.297
MASS = 1034.47
DCBAR = DBLE ( CBAR )
DB = DBLE ( B )
DS = DBLE ( S )
DEiY = DBLE ( EiY )
DELZ = DBLE ( ELZ )
DALZ = DBLE ( ALZ )
DALX = DBLE ( ALX )
DMASS = DBLE ( MASS )

IGNOREABLE CONTROL SETTINGS

DAR = 0.
DAL = 0.
DRR = 0.
DRL = 0.
DSBK = 0.
DLG = 0.

LOGICAL DECLARATIONS FOR SUBROUTINE REFAERO

DEBUG = .FALSE.
QSE = .TRUE.
CALL IAERO (HEADER)

ENGINE PARAMETERS

WDTF = DUMMY
GK1 = DUMMY
P1 = DUMMY
P2 = DUMMY
TAU = DUMMY
TSTV = DUMMY
V Ae = DUMMY
CALL IENG (HEADER)

CONSTANT STATES AND CONTROLS

DTH = 130.0
KDER = 0

---

C INITIALIZE SLSQP

---

ACC = 1.D-05
MAXIT = 1 E+09
N = 2
M = 1
ME = 1
LDG = 1
LW = 150
KW = 10
ALOW = 30.0
DLow = -24.0
AUP = 50.0
DUP = 10.5
XL(1) = DBLE (ALOW * DTR)
XL(2) = DBLE (DLow * DTR)
XU(1) = DBLE (AUP * DTR)
XU(2) = DBLE (DUP * DTR)

---

C ALTITUDE LOOP

---

DO 200 IH = 1, 6
ALT = (FLOAT(IH - 1) * 10. + 5.) * 1000.
CALL ATMOS(RHO, SPSD, ALT)
HGCL = ALT

---

C CALCULATE B-SPLINE COEFFICIENTS

---

USING SUBROUTINE BCSMOD

---

CALL BCSMOD

---

C MACH NUMBER LOOP

---

DO 140 K = 3, 9
MACH = FLOAT(K) / 10.
V = MACH * SPSD
DV = DBLE (V)
QBAR = 0.5 * RHO * V * V
DQBAR = DBLE (QBAR)
WRITE(10, 20) ALT, MACH
20 FORMAT(' ALTITUDE = ', F14.2, ' ft MACH = ', F10.2)

---

C CALCULATE THRUST

---

CALL ENGL (TH, WDTF, GK1, P1, P2, 1
TAU, TESTV, TMI, TAUUMN, DTH,
2 MACH, ALT, VAE)
TX = 2.0 * TH * COS(1.98 * DTR)
DTX = DBLE ( TX )
ANV = D MASS * DV
GV = DBG / DV
QBSM = DQBAR * DS / AMV
QBI = DQBAR * DS / DEIY
TXI = DELZ * DTX / DEIY
TNV = DTX / AMV

C---------------------------------------------
C FLIGHT PATH ANGLE LOOP
C---------------------------------------------
DO 130 J = 1,19
   GAMDEG = AKM * FLOAT(J-1) * 10.
   GAMMA = DBLE(GAMDEG) * DDTR
   KZERO = 0
   FCOST = 1.0D0
   GCOST = 1.0D0
   ALPHA = 36.0D0
   X( 1 )= ALPHA * DDTR
   DHT = -13.0D0
   X( 2 )= DHT * DDTR

C---------------------------------------------
C ITERATION LOOP
C---------------------------------------------
DO 110 I = 1,2
   MODE = 0
   DO 100 II = 1,250

C---------------------------------------------
C CALCULATE AERODYNAMIC COEFFICIENTS
C---------------------------------------------
SALPHA = SNGL ( ALPHA )
SDHT = SNGL ( DHT )
CALL COEFF(SCL, SCD, SCM, SCLQ, SCMQ, 1
        SCL0, SCDO, SCM0, SCLAD, SCMAD, 2
        SCLA, SCDA, SCM0, SCLDH, SCDDH, 3
        SCLDH, SCLQA, SCMQA, SCLODH, SCMODH, 4
        SCLQA, SCMQA, 5
        SALPHA, SDHT, MACH, 0. )

C---------------------------------------------
C CONVERT COEFFICIENTS TO DOUBLE PRECISION
C---------------------------------------------
CL = DBLE ( SCL )
CD = DBLE ( SCD )
CM = DBLE ( SCM )
CLQ = DBLE ( SCLQ )
CMQ = DBLE ( SCMQ )
CL0 = DBLE ( SCL0 )
CD0 = DBLE ( SCDO )
CM0 = DBLE ( SCM0 )
CLAD = DBLE ( SCLAD )
CMAD = DBLE ( SCMAD )
CLA = DBLE ( SCLA )
CDA = DBLE ( SCDA )
CMA = DBLE ( SCMA )
CLDH = DBLE ( SCLDH )
CDDH = DBLE ( SCDDH )
CMDH = DBLE ( SCMDH )
CL0A = DBLE ( SCLOA )
CM0A = DBLE ( SCM0A )
CL0DH = DBLE ( SCLODH )
CM0DH = DBLE ( SCM0DH )
CLQ0A = DBLE ( SCLQ0A )
CMQ0A = DBLE ( SCMQ0A )

C

CONVERT DERIVATIVES TO RADIANS

CLA = CLA / DDTR
CDA = CDA / DDTR
CMA = CMA / DDTR
CLDH = CLDH / DDTR
CDDH = CDDH / DDTR
CMDH = CMDH / DDTR
CL0A = CL0A / DDTR
CM0A = CM0A / DDTR
CL0DH = CL0DH / DDTR
CM0DH = CM0DH / DDTR
DL = DALX * DCOS(X(1)) + DALZ * DSIN(X(1))
DD = DALX * DSIN(X(1)) - DALZ * DCOS(X(1))
IF(MODE .EQ. -1) GOTO 40

C

OBJECTIVE FUNCTION d(alpha)/dt = 0

QT = - GV * DCOS(GAMMA) * DAKM
  + QBSM * CL0
2
  + TMV * DSIN(X(1))
QB = 1 - QBSM * CLQ
QQ = QT / QB
F = - FCOST * QQ

C

EQUALITY CONSTRAINT d(q)/dt = 0

GG(1) = TXI
  + QBI * (DCBAR * CM0 + DL * CL0 + DD * CDO
1
  + (DCBAR * CMQ + DL * CLQ) * QQ)
2
GG(1) = GCOST * GG(1)
IF(MODE .EQ. 1) GOTO 50

C

DERIVATIVES OF OBJECTIVE FUNCTION

40 QT0 = QBSM * CL0A
  + TMV * DCOS(X(1))

Appendix D. Computer Programs 221
QBA = QBSM * CLQA
QA = (QTA * QB - QT * QBA) / (QB * QB)
DF(1) = FCOST * QA
QDK = QBSM * CLODH / QB
DF(2) = FCOST * QDH

C NORMALS TO THE CONSTRAINT

DG(1,1) = QBI * (DCBAR * CMQA + DL * (CLOA + CDO)
+ DD * (CDA - CLQ)
+ (DCBAR * CMQA + DL * CLQ - DD * CLQA)
+ QQ
DG(1,1) = GCOST * DG(1,1)
DG(1,2) = QBI * (DCBAR * CMODH + DL * CLODH + DD * CDDH
+ (DCBAR * CMQ + DL * CLQ) * QDH
DG(1,2) = GCOST * DG(1,2)

C CALL SLLSQP

50 IF(KZERO .EQ. 1) GOTO 120
CALL SLLSQP(M ,ME ,LDG ,N ,X ,XL ,XU ,
, F ,GG ,DF ,DG ,ACC ,MAXIT ,MODE ,
, W ,LW ,INDEX)
ALPHA = X(1) / DDTR
DHT = X(2) / DDTR
IF(MODE .EQ. 0) KZERO = 1
100 CONTINUE
FCOST = FCOST * 1.D1
GCOST = GCOST * 1.D-1
110 CONTINUE
WRITE(10,116)
116 FORMAT( ' SLLSQP DID NOT CONVERGE')
120 AMAX = SNGL ( ALPHA )
DHMAX = SNGL ( DHT )
QMAX = SNGL ( QQ )
QDMAX = QMAX / DTR
WRITE(10,125) AMAX, QDMAX, DHMAX
125 FORMAT(3F14.7)
130 CONTINUE
140 CONTINUE
200 CONTINUE
STOP
END
Vita

The author was born on February 5, 1961 in South Boston, VA. He graduated from Prince Edward Academy in Farmville, VA. in June, 1979. He received a B.S. in Aerospace and Ocean Engineering in 1986 from Virginia Polytechnic Institute and State University, Blacksburg, Va. and began graduate study there in August, 1986.