NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TIME-OPTIMIZATION OF HIGH PERFORMANCE COMBAT MANEUVERS

by

Benjamin R. Carter

June 2005

Thesis Advisor:	R.M. Howard
Co-Advisor:	I.M. Ross

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

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

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Time-Optimization of High Performance Combat Maneuvers	**5. FUNDING NUMBERS**
6. AUTHOR(S) LTjg Benjamin R. Carter, USN	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

Recent developments in post-stall maneuverability and thrust vectoring have opened up new possibilities in the field of air combat maneuvering. High angle of attack maneuvers like the Cobra, Herbst Reversal, and Chakra demonstrate that today's cutting edge fighters are capable of exploiting the post-stall flight regime for very dynamic and unconventional maneuvers. With the development and testing of Unmanned Combat Aerial Vehicles, even greater maneuvering ability is expected. However, little work has been done to make use of this increased ability by optimizing a wide range of combat maneuvers. The goal of this thesis was to begin that process by finding several time-optimal air combat maneuvers that could be employed by current and future high performance fighter aircraft.

14. SUBJECT TERMS ACM, air combat maneuvering, maneuver optimization, minimum time maneuvers, Cobra, Herbst, post-stall, thrust vectoring, F-18, HARV, X-31, DIDO	**15. NUMBER OF PAGES** 243
	16. PRICE CODE

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

THIS PAGE INTENTIONALLY LEFT BLANK

TIME-OPTIMIZATION OF HIGH PERFORMANCE COMBAT MANEUVERS

Benjamin R. Carter
Lieutenant Junior Grade, United States Navy
B.S., United States Naval Academy, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2005

Author: Benjamin R. Carter

Approved by: Professor R.M. Howard
Thesis Advisor

Professor I.M. Ross
Co-Advisor

Distinguished Professor Anthony J. Healey
Chairman, Department of Mechanical and Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Recent developments in post-stall maneuverability and thrust vectoring have opened up new possibilities in the field of air combat maneuvering. High angle of attack maneuvers like the Cobra, Herbst Reversal, and Chakra demonstrate that today's cutting edge fighters are capable of exploiting the post-stall flight regime for very dynamic and unconventional maneuvers. With the development and testing of Unmanned Combat Aerial Vehicles, even greater maneuvering ability is expected. However, little work has been done to make use of this increased ability by optimizing a wide range of combat maneuvers. The goal of this thesis was to begin that process by finding several time-optimal air combat maneuvers that could be employed by current and future high performance fighter aircraft.

The aircraft used for this study were the Navion, F-18 HARV and UCAV-X (a fictitious aircraft based on the X-31 EFM). Extremely detailed physical and aerodynamic models were developed for all three aircraft, including thrust vectoring data on the latter two. Different methods of coding the aerodynamic data (including look-up tables and curve-fitting) were experimented with to determine what provided the best balance of accuracy and efficiency. The optimization program used was DIDO, which is a MATLAB-based application package for solving dynamic optimization problems developed at the Naval Postgraduate School. The "aircraft code" that interfaces with DIDO (originally written by a previous thesis student) was modified to function with the HARV and UCAV-X. Further modifications were added to improve numerical stability and decrease run times. Finally, time-optimal maneuvers developed by DIDO were compared to 1) similar maneuvers performed by other test aircraft, 2) optimal maneuvers derived in other studies, and 3) current air combat maneuvers.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

NOMENCLATURE

ACRONYMS

3DOF	three-degree-of-freedom
6DOF	six-degree-of-freedom
AAM	air-to-air missile
ACM	air combat maneuvering
AGARD	Advisory Group for Aerospace Research & Development
AOA	angle of attack (deg)
BC	boundary condition
BFM	basic fighter maneuver
CF	conventional fighter
CG	center of gravity
CFD	computational fluid dynamics
DARPA	Defense Advanced Research Projects Agency
HARV	High Alpha Research Vehicle
HATP	High Angle of Attack Technology Program
HUD	heads-up display
LEF	leading-edge flap
MBB	Messerschmitt Bölkow Blohm
MDC	McDonnell Douglas Corporation
PVF	pure vectoring fighter
SF	supermaneuverable fighter
STVF	supermaneuverable thrust vectoring fighter
S&C	stability and control
TED	trailing edge down
TEF	trailing-edge flap
TEL	trailing edge left
TER	trailing edge right
TEU	trailing edge up

TIR	time-improvement ratio
TsAGI	Central Aerohydrodynamics Institute of Russia
TV	thrust vectoring
T/W	thrust-to-weight
WVU	West Virginia University

SYMBOLS

b	reference wing span (ft)
c	reference wing chord (ft)
C_A	coefficient of axial force
C_D	coefficient of drag
C_{D_0}	baseline coefficient of drag
C_{D_α}	coefficient of drag due to AOA (deg^{-1})
$C_{D_{\delta c}}$	coefficient of drag due to canard deflection (deg^{-1})
$C_{D_{\delta e}}$	coefficient of drag due to elevator deflection (deg^{-1})
$C_{D_{\delta LEF}}$	coefficient of drag due to LEF deflection (deg^{-1})
$C_{D_{\delta PV}}$	coefficient of drag due to pitch vane deflection (deg^{-1})
$C_{D_{\delta s}}$	coefficient of drag due to stabilator deflection (deg^{-1})
$C_{D_{\delta sa}}$	coefficient of drag due to symmetric aileron deflection (deg^{-1})
$C_{D_{\delta sb}}$	coefficient of drag due to speedbrake deflection (deg^{-1})
$C_{D_{\delta TEF}}$	coefficient of drag due to TEF deflection (deg^{-1})
C_L	coefficient of lift
C_{L_0}	baseline coefficient of lift
$C_{L_{\max}}$	maximum coefficient of lift
C_{L_q}	coefficient of lift due to pitch rate (deg^{-1})
C_{L_α}	coefficient of lift due to AOA (deg^{-1})

$C_{L_{\delta c}}$	coefficient of lift due to canard deflection (deg^{-1})
$C_{L_{\delta e}}$	coefficient of lift due to elevator deflection (deg^{-1})
$C_{L_{\delta LEF}}$	coefficient of lift due to LEF deflection (deg^{-1})
$C_{L_{\delta PV}}$	coefficient of lift due to pitch vane deflection (deg^{-1})
$C_{L_{\delta s}}$	coefficient of lift due to stabilator deflection (deg^{-1})
$C_{L_{\delta sa}}$	coefficient of lift due to symmetric aileron deflection (deg^{-1})
$C_{L_{\delta sb}}$	coefficient of lift due to speedbrake deflection (deg^{-1})
$C_{L_{\delta TEF}}$	coefficient of lift due to TEF deflection (deg^{-1})
C_l	coefficient of rolling moment
$C_{l_{bias}}$	rolling moment bias
C_{l_p}	coefficient of rolling moment due to roll rate (deg^{-1})
C_{l_r}	coefficient of rolling moment due to yaw rate (deg^{-1})
C_{l_β}	coefficient of rolling moment due to sideslip (deg^{-1})
$C_{l_{\delta a}}$	coefficient of rolling moment due to aileron deflection (deg^{-1})
$C_{l_{\delta ds}}$	coefficient of rolling moment due to differential stabilator deflection (deg^{-1})
$C_{l_{\delta r}}$	coefficient of rolling moment due to rudder deflection (deg^{-1})
$C_{l_{\delta YV}}$	coefficient of rolling moment due to yaw vane deflection (deg^{-1})
C_m	coefficient of pitching moment
C_{m_0}	baseline coefficient of pitching moment
C_{m_q}	coefficient of pitching moment due to pitch rate (deg^{-1})
C_{m_α}	coefficient of pitching moment due to AOA (deg^{-1})
$C_{m_{\delta c}}$	coefficient of pitching moment due to canard deflection (deg^{-1})
$C_{m_{\delta e}}$	coefficient of pitching moment due to elevator deflection (deg^{-1})
$C_{m_{\delta LEF}}$	coefficient of pitching moment due to LEF deflection (deg^{-1})
$C_{m_{\delta PV}}$	coefficient of pitching moment due to pitch vane deflection (deg^{-1})

$C_{m_{\delta s}}$	coefficient of pitching moment due to stabilator deflection (deg^{-1})
$C_{m_{\delta sa}}$	coefficient of pitching moment due to symmetric aileron deflection (deg^{-1})
$C_{m_{\delta sb}}$	coefficient of pitching moment due to speedbrake deflection (deg^{-1})
$C_{m_{\delta TEF}}$	coefficient of pitching moment due to TEF deflection (deg^{-1})
C_n	coefficient of yawing moment
$C_{n_{bias}}$	yawing moment bias
C_{n_p}	coefficient of yawing moment due to roll rate (deg^{-1})
C_{n_r}	coefficient of yawing moment due to yaw rate (deg^{-1})
C_{n_β}	coefficient of yawing moment due to sideslip (deg^{-1})
$C_{n_{\delta a}}$	coefficient of yawing moment due to aileron deflection (deg^{-1})
$C_{n_{\delta ds}}$	coefficient of yawing moment due to differential stabilator deflection (deg^{-1})
$C_{n_{\delta r}}$	coefficient of yawing moment due to rudder deflection (deg^{-1})
$C_{n_{\delta YV}}$	coefficient of yawing moment due to yaw vane deflection (deg^{-1})
C_N	coefficient of normal force
C_Y	coefficient of side force
$C_{Y_{bias}}$	side force bias
C_{Y_p}	coefficient of side force due to roll rate (deg^{-1})
C_{Y_r}	coefficient of side force due to yaw rate (deg^{-1})
C_{Y_β}	coefficient of side force due to sideslip (deg^{-1})
$C_{Y_{\delta a}}$	coefficient of side force due to aileron deflection (deg^{-1})
$C_{Y_{\delta ds}}$	coefficient of side force due to differential stabilator deflection (deg^{-1})
$C_{Y_{\delta r}}$	coefficient of side force due to rudder deflection (deg^{-1})
$C_{Y_{\delta YV}}$	coefficient of side force due to yaw vane deflection (deg^{-1})
F_X, F_Y, F_Z	applied external forces (lb)
g	gravity (32.2 ft/sec^2)

h	angular moment vector
H	altitude (ft)
I_X, I_Y, I_Z	principal moments of inertia (slug-ft^2)
I_{XY}, I_{YZ}, I_{XZ}	cross products of inertia (slug-ft^2)
m	aircraft mass (slug)
M_X, M_Y, M_Z	applied external moments (ft-lb)
n	load factor (g)
p, q, r	body rates – roll, pitch and yaw (deg/sec)
P_S	specific excess power (ft/sec)
$\bar{q}$	dynamic pressure (lb/ft^2)
s, S	reference wing area (ft^2)
T	thrust (lb)
V	velocity (ft/sec)
$\vec{V}$	velocity vector – {u v w}
W	aircraft weight (lb)
X	down range distance (ft)
Y	cross range distance (ft)
α	angle of attack (deg)
β	sideslip angle (deg)
δ	control surface deflection (deg)
Δ	delta, change in flight condition variable
ϕ, θ, ψ	Euler angles – bank, pitch and heading (deg)
ω	angular velocity vector – {p q r}

SUPERSCRIPTS

$\dot{X}$	rate of change (of “X”) with respect to time (sec^{-1})

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This thesis was a very rewarding project to work on. From preliminary research to gathering data to coding and analyzing results, each aspect of the thesis had its own trials and triumphs. Also, as a future Naval Flight Officer, it was very interesting studying the mechanics of dynamic flight, and seeing post-stall maneuvers push jets like the F-18 HARV to their limits. In fact, one of the most satisfying points of the thesis was seeing the models for the HARV and UCAV-X take shape; both aircraft required weeks of compiling figures and reconciling data from half a dozen sources, but the results were great. The only thing better was seeing video clips of the time-optimal maneuvers they were able to fly. So, while this thesis turned into a very significant volume accomplishing all of the above tasks, I think the results justify the extra “bulk.”

I owe quite a few people my thanks, both for their assistance and their patience, and I’d like to start with my advisors: Professors Richard Howard and Michael Ross. Both were extremely interested in this project and my results, and provided invaluable insight that helped me overcome obstacles, and kept me (for the most part) focused on my primary goal of creating time-optimal maneuvers. As an additional benefit, they provided assistance and expertise in completely different areas: Professor Howard with theory and aircraft mechanics, and Professor Ross with optimization, DIDO and related numerical issues. Thank you!!

Dr. Al Bowers and Dr. Brent Cobleigh at NASA Dryden Flight Research Center shared their expertise and advise on the HARV and X-31, respectively. Dr. Bowers, thank you in particular for taking the time to send answers to my exhaustive list of questions on the aerodynamics of the HARV. Dr. Cobleigh, thank you for getting me started on the X-31 with the great Rockwell Aerospace papers you sent. On a related note, thank you to the entire VECTOR Program Office at Patuxent River for the fantastic data sets on the X-31…a special thanks to Dave Sorenson, Mary Meyer, and CDR “Dutch” Rauch for their personal time and assistance. LCDR Scott Josselyn, thank you for getting this entire project started! Your initial work on the Navion, your availability

for discussing coding issues, and your continued work on this project have been a fantastic help…look forward to working with you again in the future.

A final thank you to the friends and family that have endured hearing me talk about my thesis for the last two years – thanks for the encouragement and support. Most especially to Cindy, who has been "widowed" many evenings and weekends while I hammered away at the code and thesis – thanks for being a wonderful, understanding, and inspiring companion.

To any that peruse this thesis (or are brave enough to tackle it in its entirety) – enjoy…I know I did.

I. BACKGROUND

A. AIR-TO-AIR COMBAT

Over the past eighty years, aerial combat has evolved radically to keep up with the rapidly improving technology of weapons systems, fighter aircraft, and aviation in general. The development and refining of turbojet and turbofan engines have led to the development of fighters capable of thrust-to-weight (T/W) ratios greater than one. Airfoil designs have improved dramatically with extensive wind tunnel testing, as well as computational fluid dynamics (CFD)–based airfoil modeling. These improved designs have resulted in wings with better lifting characteristics, reduced drag penalties, and increased performance over large ranges of speed and angle of attack (AOA). Digital fly-by-wire control systems have replaced purely mechanical systems, allowing a pilot to command a flight condition as opposed to a single control surface deflection. Numerous other advances have played an important role in the development of aviation and fighter aircraft, including supercritical airfoils, area ruling, canards, all-moving stabilators, thrust-vectoring, and advanced flight simulators. The result is that fighters, in general, have become faster, more maneuverable, and more deadly.

Weapons systems have undergone similar transformations: the original air-to-air weapon, a pilot's service pistol, was replaced by fixed .30-cal machine guns by the end of World War I, .50-cal machine guns by World War II, and 20mm to 40mm cannons by the early 1950's. The last major development in air-to-air guns was the M61 "Gatling-gun" cannon in 1957, capable of firing 0.22lb explosive rounds at an amazing rate of 6,000 rounds/min. [Ref. 1] The highly increased lethality of this weapon system over its predecessors is a function of the increased rate of fire, increased muzzle velocity, increased projectile weight, and use of explosive projectiles. In addition to improved air-to-air guns, the advent of guided air-to-air missiles (AAM's) following World War II has significantly changed the science of aerial combat. Weapons systems like the AIM-54 Phoenix and its AWG-9 radar can detect and engage multiple targets at ranges greater than 100nm. Modern AAM's can track targets with passive, semi-active or active homing systems, and travel at speeds greater than Mach 4. Complementing these potent weapons are very sophisticated targeting systems, using computer graphics and heads-up displays (HUD's) to quickly relay pertinent data to the pilot. In brief, weapons systems have become more lethal and more pilot-friendly.

These improved aircraft and weapons capabilities require a reassessment of air-to-air combat and air combat maneuvering (ACM). The scope of this thesis is not to address this issue in its entirety, but to examine a small portion of it. Namely, to find several time-optimal maneuvers that could be implemented by current or future fighter aircraft to obtain an advantage in short range, primarily one-versus-one air combat. The first step in this search will be to start with some characteristics of short range air combat observed and predicted by previous studies.

An excellent paper by Herbst [Ref. 2] discusses the combat and design implications of some of the afore-mentioned technological advances, in particular the development of all-aspect guns, and short and medium range AAM's. His conclusions about the first two are of primary relevance:

> [Short range] combat with rear aspect weapons was characterized by sustained turns. Conversion to a firing solution was a matter of sustained turn rate margin vs. the opponent. ...With all-aspect weapons, however, combat effectiveness proves to be significantly less sensitive to classic energy maneuverability parameters and more sensitive to attained unsteady performance. [Ref. 2: p. 594]

According to Herbst, the first variety of combat was played out in most simulated engagements by the two dueling aircraft first maneuvering into a head-on situation. If the subsequent weapons exchange was unsuccessful, both aircraft then attempted to reverse course as quickly as possible for another exchange. Hence the fighter with the better turn rate was able to maneuver for a shot before his opponent, and was more likely to win the engagement. Put in another way, the fighter that was able to reverse his velocity vector faster was more likely to win the engagement. Of the second variety of combat, Herbst notes that "The ability to aim the aircraft fuselage independently of the flight path provides a very effective way of solving the gun snap shooting problem..." [Ref. 2] Therefore, the key to winning this engagement is changing the aircraft's orientation, as opposed to the aircraft's velocity vector. The latter involves managing energy between turning, climbing and accelerating, in order to maneuver into a firing position in minimum time (without losing so much energy that escape or reengagement is impossible). The former, however, involves achieving a decisive positional advantage in minimum time, with the possible expense of some energy.

Another excellent source of air combat characteristics is Shaw's text [Ref. 1]. While his text will be referenced more extensively in Chapter II, there are some general comments that can be made for the purposes of this section. Shaw refers to "...basic fighter maneuvers (BFM's) [as] the building blocks of fighter tactics." [Ref. 1] BFM's that he further classifies as "relative" (i.e. "performed in relation to another aircraft" [Ref. 1]) are of primary importance in ACM. One such maneuver is the nose-to-nose turn. The nose-to-nose turn is similar to the air combat with rear aspect weapons that Herbst described, except that it is limited to the horizontal plane.

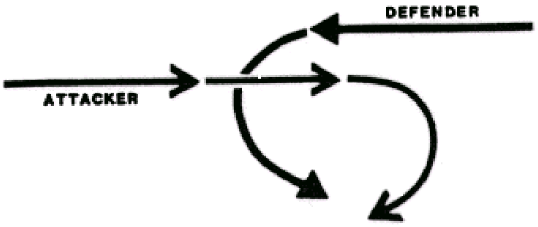


Figure 1. Nose-to-Nose Turn (From: Ref. 1, pp. 78)

For this maneuver, it can be seen from the figure above that a defender with a smaller turn radius than his opponent would be able to shoot first, as opposed to the increased turn rate needed to win in Herbst's scenario.

Another relative BFM is a pursuit curve. Pursuit curves (lead, pure and lag) are similar to the air combat with all-aspect weapons that Herbst described, except they are again limited to the horizontal plane. As with the Herbst scenario, though, the goal of the maneuver is to aim the fuselage at the target.

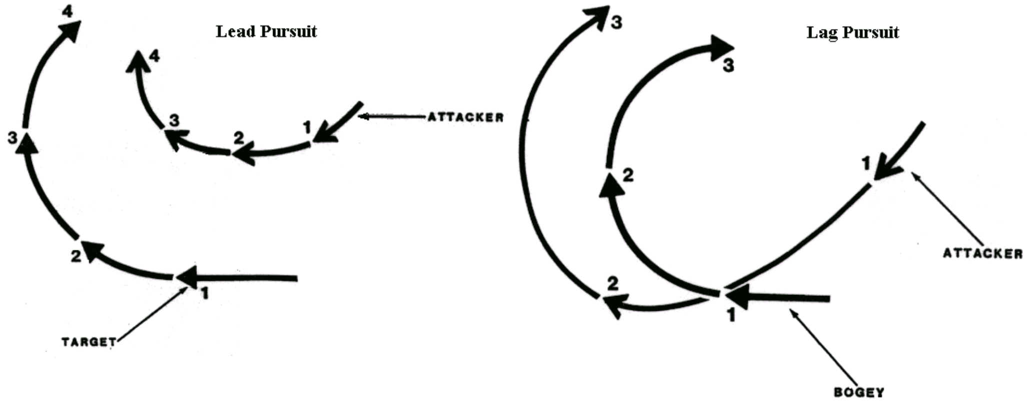


Figure 2. Lead and Lag Pursuit Curves (After: Ref. 1, pp. 63-5)

In lead pursuit, the fuselage of the attacker is actually aimed ahead of the target. The attacker is able to close on a faster target, but will require an increasingly smaller turn radius, and an increasingly faster turn rate to maintain his positional advantage. [Ref. 1] In lag pursuit, the fuselage of the attacker is aimed behind the target. The attacker is able to avoid overshooting the target, but it can be difficult to obtain the position necessary for a shot opportunity. [Ref. 1]

Based on these two references, it can be seen that success in short range air combat will likely be based on any combination of the below characteristics: turn radius, turn rate, velocity, acceleration, climb performance, energy management, and fuselage reorientation. These performance criteria will be examined more thoroughly in Chapter II; the rest of this chapter will study some other aspects of aerial combat that will play a large role in finding time-optimal maneuvers.

B. POST-STALL FLIGHT

The post-stall region has been a source of considerable interest and research in the aviation community over the past two decades. It is characterized by separated and reverse flow over the wing, loss of lift, and a steep increase in drag. As can be seen in Figure 3, stall occurs at $C_{L_{\max}}$. [Ref. 3] The AOA range past that point is the post-stall region. Notice, though, that while C_L decreases significantly after the airfoil or wing stalls, it doesn't plummet to zero. It actually begins to increase again, and for a fighter aircraft with swept wings, it may increase to a second, smaller peak before finally tapering off.

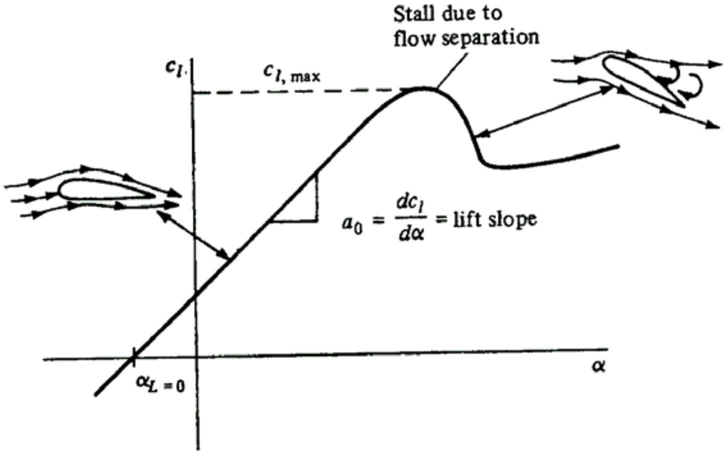


Figure 3. Maximum Lift Coefficient (From: Ref. 3, pp. 282)

What the previous two observations mean is that an aircraft can still be flyable in the post-stall region provided that several criteria are met:

1. The aircraft has enough thrust to overcome the huge drag increase.
2. The aircraft has controls that will not be rendered ineffective by separated flow over the wings and tail.
3. C_L remains great enough in post-stall to overcome the aircraft's weight.

Hence the reason that the post-stall region has only been a fairly recent area of study: T/W ratios needed to increase, $C_{L_{\max}}$ values needed to increase, and non-aerodynamic controls (such as TV) had to be developed before an aircraft would be capable of controlled flight in this very adverse aerodynamic region. Now that these advances have taken place, both simulation and flight test studies are being performed in this new flight regime.

NASA took a leading role in this testing in the mid 1980's by creating the High AOA Technology Program (HATP). The goals of the program were to "…provide flight-validated aircraft design tools and to improve the maneuverability of aircraft at high angles of attack." [Ref. 4] This was done through wind-tunnel testing, CFD modeling, and flight testing; results were shared at conferences hosted by NASA every two years between 1990 and 1996. (More detailed information concerning the HATP can be found in Ref. 5-7.) NASA was not the only organization to venture into the study of high AOA and post-stall maneuverability: the Air Force and Navy both joined with NASA for different projects, MBB in Germany conducted extensive research on the subject, and TsAGI in Russia performed extensive flight testing. The next subsection will discuss some of what has been learned about post-stall flight by these various groups, and the possible implications that operating in the post-stall regime will have on time-optimal air combat maneuvers.

1. Supermaneuverability

Herbst (of MBB) first coined the term supermaneuverability. In his words, it is

> ...a term for combined post-stall (PST) and direct force (DFM) capability. PST represents the ability of the aircraft to perform controlled tactical maneuvers beyond maximum lift angle of attack up to at least 70 deg; DFM represents the ability of the aircraft to yaw and pitch independently of the flight path, or to maneuver at constant fuselage attitude. [Ref. 8: p. 564]

Herbst made the assertion in his 1980 paper that supermaneuverability would be one of the key enabling technologies for future fighter aircraft. This claim was partly based on the results of extensive manned and unmanned flight simulations performed at MBB, which observed the following advantages of a supermaneuverable fighter (SF) over a conventional opponent:

1. "Exchange ratio of 2:1 with missiles
2. Exchange ratio of 10:1 with guns
3. Exchange ratio of 3:1 with guns and missiles
4. 5 wins out of 6 fights (hits without counter hits)
5. Withstands two conventionally equal threats" [Ref. 8: p. 566]

Based on these results, and Herbst's description of an example supermaneuver now commonly known as a Herbst Reversal [Ref. 8], it is obvious that supermaneuverability should be an essential capability of any new fighter aircraft. The dynamic maneuvers possible in the post-stall region, the freedom from purely aerodynamic control surfaces, and the ability to aim the aircraft's fuselage and weapons independent of the direction of flight combine to make a SF extremely lethal in the short range air combat arena. For these reasons, time-optimal air combat maneuvers will likely be faster, and very different in appearance when performed by a SF.

The feasibility of this supermaneuverability concept was demonstrated independently, and around the same time, by two different organizations: NASA in the US (through the HATP program), and TsAGI in Russia. Analytic research of post-stall flight was started at TsAGI in the early 1980's, shortly after the possibility of supermaneuverability was first discovered. [Ref. 9] In 1987, after several years of research, Yu. N. Zhelnin proposed that a "short duration...dynamic maneuver" [Ref. 9] could be performed by a standard Su-27 or MiG-29. TsAGI collaborated with the P.O. Sukhoi design office and the Flight Research Institute to perform the necessary testing and preparation. Su-27 test pilot Victor Pugachev first publicly performed the supermaneuver in May, 1989. Now commonly known as the "Pugachev's Cobra," the maneuver has been performed at international air shows around the world by many pilots in both the Su-27 and MiG-29.

For NASA's program, a McDonnell Douglas F-18 was selected as the test bed aircraft for supermaneuverability, and was named the High Alpha Research Vehicle (HARV). [Ref. 4] After several years of research in the mid 1980's, and extensive reassembling of the aircraft, the HARV took its first flight on April 2, 1987 [Ref. 4]; this began a nine-year career of flight testing, resulting in numerous successful demonstrations of post-stall maneuverability and controllability. Both the NASA and TsAGI programs showed the tremendous combat potential of a SF or STVF, in terms of fuselage pointing, maneuvering beyond the stall AOA, and performing supermaneuvers like the Cobra and Herbst Reversal. They also identified areas requiring more development to fully exploit the post-stall region, the most significant of which was the area of controls. Controllability is a major issue in the post-stall regime, and will be discussed more in the next section.

C. THRUST VECTORING

The goal of TV is to partially redirect the thrust of an aircraft up or down to induce pitch, left or right to induce yaw, and (for two engine aircraft) asymmetrically up and down to induce roll. There are two primary application for this technology: the first is to increase the agility of a conventional fighter aircraft by pairing TV with standard aerodynamic control surfaces to improve the aircraft's ability to rapidly change its orientation. The second, and arguably more important application, is to use TV on a SF to provide primary control of the aircraft at high AOA, where conventional control surfaces are rendered ineffective by flow separation.

This second type of TV equipped fighter is classified by Gal-Or as a Partially Vectored Aircraft [Ref. 10]. Gal-Or discusses another type of fighter (which he names a Pure Vectored Aircraft [Ref. 10]) that uses *only* TV for control of the aircraft, and has *no* conventional control surfaces (i.e. no rudder, ailerons, elevator, or flaps). For this theoretical aircraft,

> ...the flight-control forces generated by the conventional, aerodynamically-affected, external control surfaces of the aircraft, are replaced by the stronger, internal, thrust forces of the jet engine(s). These forces may be simultaneously, or separately, directed in all directions, i.e., in the yaw, pitch, roll, thrust-reversal, and forward thrust coordinates of the aircraft. [Ref. 10: p. 29]

In short, a pure vectoring fighter (PVF) would have a similar or greater advantage over a supermaneuverable, TV fighter (STVF) than a SF has (and has demonstrated) over a conventional fighter (CF). The primary reason for that advantage is very straightforward: the SF or STVF remains controllable in the post-stall region, and can therefore perform maneuvers that a CF cannot; hence the combat advantages enumerated in the previous section. Likewise, while a SF or STVF is controllable in post-stall, a PVF theoretically would have the *same* control power in the post-stall region as it does in conventional flight (since its controls are independent of the external air flow). Therefore, it would be able to outperform a SF or STVF in post-stall, and consequently would be the most capable fighter aircraft possible until a breakthrough in engine technology comes along.

For the purpose of this study, the PVF will not be used for analyzing time-optimal maneuvers, because there is no data available for such an aircraft. However, it can be expected that a PVF would be able to perform even faster and more dynamic maneuvers than are currently possible. So, concerning aircraft that are currently in existence, or at some stage of testing or development, partial TV is the key to achieving supermaneuverability. The table below lists most of the significant TV equipped aircraft that have been developed to date:

Table 1. Thrust Vectoring Aircraft

(Test flights)	**Experimental Aircraft**
1987-96	F-18 HARV (High Alpha Research Vehicle)
1990-	X-31 EFM (Enhanced Fighter Maneuverability)
1993-95	F-16 MATV (Multi-Axis Thrust Vectoring)
1996-	F-15 ACTIVE (Advanced Control Technology for Integrated VEhicle)
(First flight)	**Testing Phase Aircraft**
1996	Su-37 Super Flanker
(Enter service)	**Production Aircraft**
2005	F-22 Raptor

While very little data is available on the more recent aircraft like the Raptor and the Super Flanker, a wealth of information is available on the earlier TV projects like the

HARV and EFM. For this reason, and because both aircraft are extremely capable (but very different) STVF's, the HARV and EFM will both be used in this study for time-optimal maneuver simulations.

D. UNMANNED COMBAT AERIAL VEHICLES

Although UAV's have been in production for quite some time, even those that are designed for military combat roles (like the Predator) are typically only used as reconnaissance or strike weapon delivery systems. However, there is research and testing being done now to develop a true combat UAV. The Navy, for example, has contracted Boeing to develop a UCAV that is currently designated the X-45. [Ref. 11] Unfortunately, development of the X-45 is not at a point where it could be included in this study. Also, it is not known at this time whether or not the X-45 is being developed with air-to-air combat in mind. However, there is definitely potential that a supermaneuverable UCAV designed for short range air combat would out-fly even the most advanced air superiority fighters currently being produced.

One advantage of a UCAV over a manned fighter, of course, is that load factor (n) is only a structural consideration. However, some studies have shown that there is a relaxation of n when performing supermaneuvers in the post-stall region. [Ref. 8] (Likely due to the sharp decrease in speed that is very characteristic of a number of supermaneuvers.) Even if that is the case, there are other potential advantages that a UCAV could bring to air combat. For one, while n may not be a problem for manned aircraft in post-stall flight, $\dot{n}$ may still be a limitation. Also, the slightly smaller size of UCAV's would lead to smaller moments of inertia, and hence faster roll, pitch and yaw rates; this increased agility would be an even greater advantage when coupled with supermaneuverability, as it would allow very fast pointing of the fuselage independent of the aircraft's velocity vector. (Which, if that pointing could be done faster by a UCAV than a manned aircraft, would translate to less energy lost in a post-stall maneuver, and increased ability of the UCAV to continue the fight or engage another opponent.) Also, a

UCAV would be the ideal platform for a PVF: without the need for a cockpit, the UCAV could be an engine mounted on a flying wing, with a design that would take full advantage of pure TV.

Gal-Or discusses the potential benefits of a supermaneuverable UCAV in his text. He also lists possible roles that such an aircraft would be perfectly suited for, including helicopter killer, UAV killer, and ship protector, all of which would take advantage of the amazing post-stall maneuverability of such an aircraft. [Ref. 10]

The important fact about UCAV's for the purpose of this study is as follows: while UCAV's will most likely be remotely piloted, their air-to-air combat potential could be significantly increased if they were preprogrammed with a "library" of time-optimal maneuvers. That is, during air-to-air combat, the UCAV could detect a hostile aircraft at a certain position, and based on that position and its rate of change, a maneuver may be selected from its electronic library that corresponds to a minimum time maneuver to get the UCAV into a firing position. Or at the very least, a study of time-optimal maneuvers could be made with a simulation of the UCAV (like the ones that will be used in this study), and hundreds of time-optimal maneuvers could be calculated and used as training tools for UCAV operators. In any case, the process of categorizing and finding time-optimal maneuvers could definitely be applied to the emerging field of UCAV's.

THIS PAGE INTENTIONALLY LEFT BLANK

II. AIR COMBAT MANEUVERING

A. DISCUSSION

As previously stated, the goal of this thesis is to develop several time-optimal air combat maneuvers. The primary application for these maneuvers is in the field of air combat maneuvering (ACM), which for this study will be defined specifically as the positioning and orienting of a fighter aircraft to attack an opponent or evade an opponent's attack. These two elements of ACM (position and orientation) are related to different air combat tactics, which Shaw calls the "energy" and "angles" fights. [Ref. 1] The goal of the energy fight is to develop an energy advantage over an opponent that can be used to maneuver into a firing position. The goal of the angles fight is to achieve a positional advantage from which a shot (or snapshot) can be made. In addition to the obvious goal of giving the attacker the first firing opportunity, both tactics have a secondary goal of denying that opportunity to an opponent. However, in this study only the attacker is considered, and therefore the maneuvers developed will not be "extended" combat maneuvers that are functions of the opponent's response to the attacker. Instead, they will be "instantaneous" maneuvers that can be employed by the attacker as soon as the conditions of the engagement match those of the maneuver.

For the optimization process that will be discussed in Chapter V, these instantaneous maneuvers must be defined mathematically in terms of the change, or delta, they affect on an aircraft's flight condition. The important flight condition variables for most maneuvers will be the aircraft's position, velocity, and orientation. (These variables are discussed in more detail in Chapter III, on page 26 and following.) For example, a minimum time climb from 10,000 to 11,000 ft would be expressed as a ΔH of 1000 ft. If the goal of the maneuver was not only to climb 1000 ft, but to complete the maneuver at the aircraft's initial velocity (i.e., the velocity at the beginning of the maneuver), that additional constraint would be expressed as a ΔV of 0 fps. The maneuver could be made even more specific by adding additional constraints, or it could be made more general by "relaxing" constraints. For example, instead of saying that the velocity at the end of the maneuver must be equal to the initial velocity, the final velocity could be set equal to the initial velocity ±20 fps. Once a maneuver is completely defined in this way, the optimization code will calculate the flight path and control inputs necessary to achieve that delta in minimum time.

The majority of this chapter will be spent analyzing current air combat maneuvers, and translating them into the mathematical delta expressions as described above. Extended combat maneuvers will be broken into phases so that they can be defined in the same way. Once several maneuvers have been analyzed, similar delta expressions will be categorized into "families" of air combat maneuvers. Additional families will be included based on maneuvers described in other references. From this final list, four families of maneuvers will be selected for optimization.

B. PERFORMANCE CRITERIA

Shaw does a very thorough analysis of fighter aircraft performance in the appendix of his text. [Ref. 1] This section will briefly discuss some of the terms that he defines there, some of which have already been mentioned in the "Air-to-Air Combat" section of Chapter I. The main purpose of this section, though, is to define three general ACM metrics. These three metrics will be used to help explain why the three test aircraft in this study perform maneuvers differently, and why one aircraft may be able to perform certain maneuvers much faster than one or both of the others.

The first ACM metric is "maneuverability," which will be defined as the ability to change the aircraft's velocity vector. Some terms that are commonly related to maneuverability are turn radius and turn rate. The latter is a measure of how much time is required to complete a turn, and the former is a measure of how much distance is required to complete a turn. For non-level turns (i.e. turns that involve maneuvering in both the horizontal and vertical plane), the turn radius term can be replaced with "turn volume," which defines the volume of space required to complete the turn. All three terms are strongly influenced by velocity: in general, slower speeds minimize turn radius and turn volume, while higher subsonic speeds near an aircraft's corner velocity maximize turn rate. A balance between turn rate and turn (or maneuvering) volume will maximize an aircraft's maneuverability.

The next metric that will be used to study and compare the three test aircraft is "agility." This will be defined quite simply as the ability of an aircraft to change its orientation, or equivalently, its ability to pitch, roll and yaw. Agility is largely a function of the aircraft's control power and moments of inertia. For example, an aircraft with a large yaw moment of inertia (I_Z) will inherently have a low yaw rate due to the mass that has to be rotated about the aircraft's z-axis. However, that low yaw rate can be compensated for with sufficient rudder power ($C_{n_{\delta r}}$), or by adding an additional yaw control device like TV. In general, a TV equipped aircraft will be more agile than a similar aircraft without TV due to the extra control power for pitch, yaw, and in some cases, roll. (This increased agility will be even more pronounced in the post-stall region.) Also, an aircraft with smaller moments of inertia, like a UCAV, will typically be more agile than another aircraft of comparable technology.

The last ACM metric is "available power." This term is closely related to specific excess power (P_S), which is defined as the rate of change of energy normalized by the aircraft's weight [Ref. 12]. (See the equations below:)

$$P_S = \frac{TV - DV}{W}$$

$$P_S = \dot{H} + V\frac{\dot{V}}{g}$$

As can be seen from the second equation, P_S can either be converted into a change in altitude or a change in velocity. Available power, then, will be defined as the ability of an aircraft to climb or accelerate. This metric will be very important for comparing how well different fighters can maneuver in the vertical plane, and how well different fighters could re-engage or escape from a fight after performing a maneuver that resulted in a large loss of energy. A high T/W ratio will greatly improve an aircraft's available power. (Which can be seen by separating terms in the first equation.)

C. SAMPLE MANEUVERS

The format of this section will be to take each maneuver in turn, describe and illustrate the maneuver, and then translate the maneuver into a mathematical delta. As previously mentioned, extended maneuvers will be broken into phases that could be considered instantaneous maneuvers, and one or more of those phases will be expressed as a mathematical delta.

1. Split-S

The Split-S is a maneuver performed in the vertical plane that reverses the heading of an aircraft. The maneuver is essentially the second half of a loop: the pilot rolls the aircraft until inverted, pulls the aircraft into a dive, and then pulls the aircraft out of the dive and back to straight and level flight.

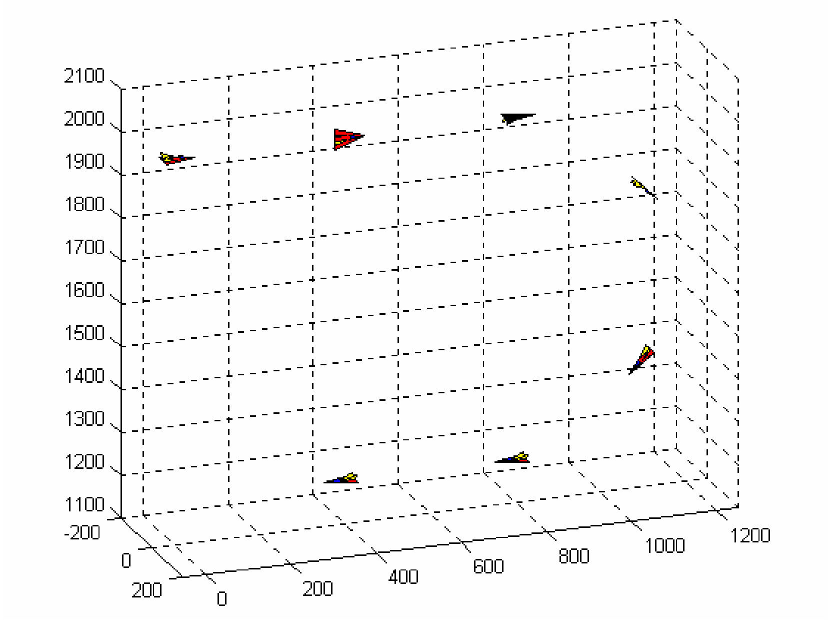


Figure 4. Split-S

This maneuver could be expressed very simply as a minimum time heading change ($\Delta\psi = 180^\circ$) that is completed with essentially no down range or cross range displacement from the starting point ($\Delta X = 0$ and $\Delta Y = 0$). There is, however, a loss in altitude (ΔH), which is traded for increased airspeed (ΔV). The primary application for this particular maneuver would be following a head-to-head pass with a lower opponent, where the loss in altitude would not be compromising.

2. Low Yo-Yo

The low yo-yo is an out-of-plane maneuver that can be used to turn inside of an opponent. The dive at the beginning of the maneuver reduces the component of velocity in the forward direction, reducing the attacker's turning radius. The climb at the end of the maneuver reduces velocity sufficiently so that the attacker doesn't overshoot the bogey while aiming for a shot. (For the high yo-yo, on the other hand, the attacker climbs first to prevent overshoot, turns inside the bogey due to his lower speed, and regains speed at the end of the maneuver by diving towards the bogey for a shot.)

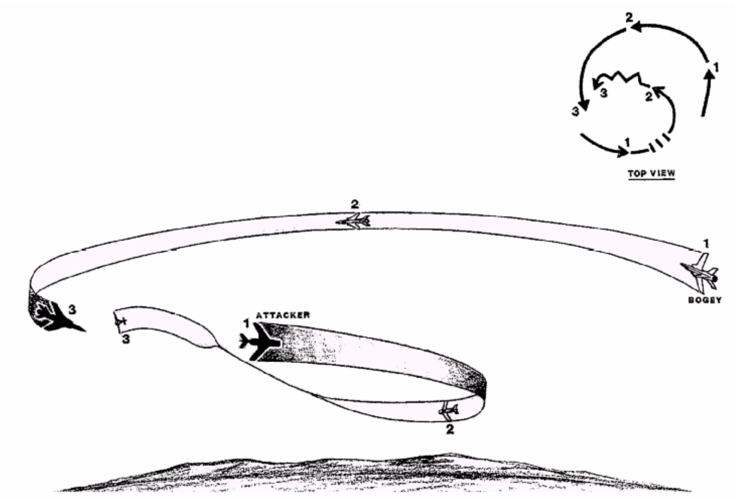


Figure 5. Low Yo-Yo (From: Ref. 1, pp. 74)

This is essentially a two-phase maneuver: the first phase (from point 1 to 2) is the dive, and the second phase (from point 2 to 3) is the climb. The reason the attacker performs this maneuver in the first place is because he is unable to turn quickly enough in the horizontal plane to maintain his lag pursuit position. Therefore, the first phase of this maneuver could be considered a minimum time 90º turn ($\Delta\psi = 90^\circ$). This can be seen by comparing the change in the bogey's heading angle from point 1 to 2, with that of the attacker: the bogey completes a 90º turn at point 2, while the attacker has turned 90º roughly two-thirds of the way between point 1 and 2. (In fact, if the attacker were still in the same plane as the bogey at the end of his 90° turn, or were able to rapidly pitch his nose up, he would have a good snapshot opportunity.)

3. Rolling Scissors

The rolling scissors is an extended maneuver characterized by the two fighter aircraft involved spiraling around each other in an attempt to maneuver into a good firing position (typically above and behind the opposing fighter). This is done by one of the fighters pulling up to climb and reduce speed, so that they cross above the flight path of the opposing fighter (see the defender's position at time 3 and 4 in the figure below). The high fighter then pulls down and towards the opposing fighter as quickly as possible to get into a firing position. If a shot is not possible for the diving fighter (or unsuccessful), then the roles of the fighters reverse. (Compare the attacker's position at times 4 and 5 to the defender's position at times 3 and 4.) This cycle would then continue until one of the fighters either has a successful shot, or disengages from the maneuver and escapes.

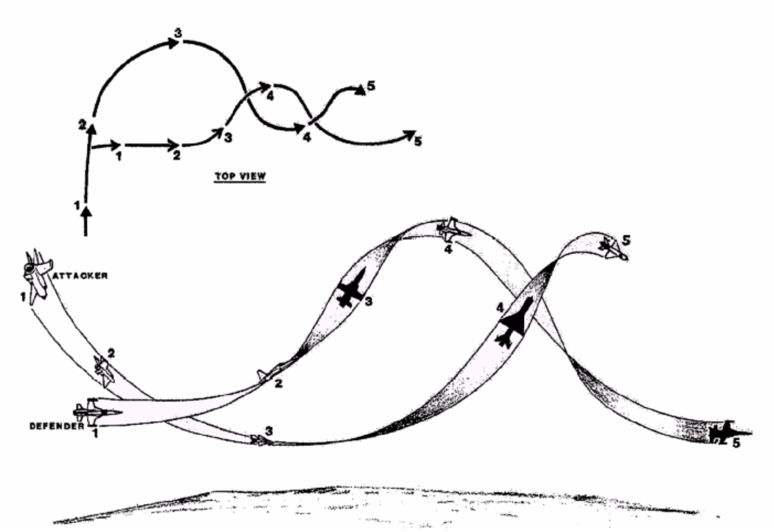


Figure 6. Rolling Scissors (From: Ref. 1, pp. 89)

The two phases of this maneuver that will be analyzed are the defender's movement from time 2 to 3 and from time 3 to 4. During this time, both aircraft are committed to the rolling scissors, and the defender is attempting to position for a shot. The goal of the defender during the first phase is essentially to reduce his velocity so that he will be positioned behind his opponent after their next pass. (If the necessary reduction was to cut his forward speed in half, for instance, the maneuver would be expressed as $\Delta V = -0.5 \cdot V_i$, where V_i is the defender's initial velocity.) The goal of the second phase is to point the aircraft at the target. In the rolling scissors, since the defender has climbed above his opponent in the first phase, this phase involves a pitch down (or up if the aircraft is inverted) and some rudder input to yaw the aircraft towards his opponent. ($\Delta\theta = -30^\circ$ and $\Delta\psi = 30^\circ$, for instance.)

4. Cobra

The Cobra is a classic post-stall maneuver that has been demonstrated in the Su-27 and MiG-29. The two primary characteristics of this maneuver are 1) a rapid pitch-up to near 90º AOA and 2) a rapid decrease in velocity by 50-75%. (The latter is due to the fact that the aircraft is flying through the air on its tail when at 90º AOA, and therefore is incurring a huge drag penalty.) The maneuver also results in an increase in altitude due to the lift generated at AOA values greater than zero.

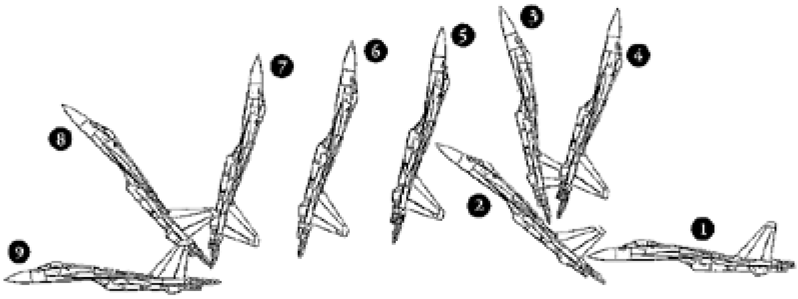


Figure 7. Cobra Maneuver

Although the Cobra is not a combat maneuver per se, there are certainly elements of the maneuver that could be applied to ACM. First, the rapid velocity decrease ($\Delta V = -0.5 \cdot V_i$) could be used to either avoid overshooting an opponent, or to evade an opponent by making *him* overshoot. Second, the rapid pitch-up ($\Delta\theta = 110^\circ$) could be paired with a roll or yaw to quickly point the aircraft at an opponent without taking the time to change the aircraft's velocity vector. (In fact, a more recently demonstrated maneuver called the Chakra has shown that an aircraft could theoretically shoot an opponent directly *behind* them by extending the Cobra past 90º to 180º. The maneuver resembles a somersault, and the aircraft actually pitches all the way from 0º to 360º AOA while flying in a nearly straight line – except for the altitude increase, as before with the Cobra. This maneuver has been performed by the latest generation Flanker, the Su-37, and is pictured below.)

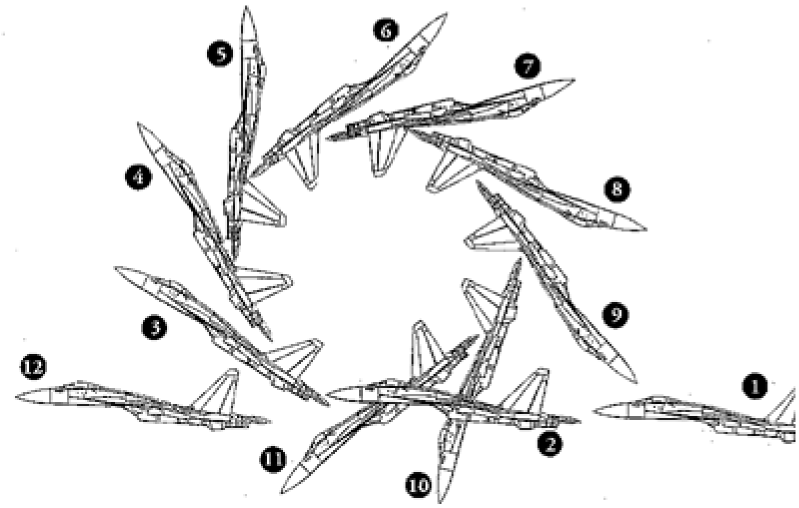


Figure 8. Chakra Maneuver

5. Herbst Reversal

The Herbst Maneuver, or Herbst Reversal, is another classic post-stall maneuver. The goal of the maneuver is quite simply to reverse the aircraft's heading angle ($\Delta\psi = 180^\circ$), and to complete the reversal at the same point and velocity that the maneuver was started from (i.e. $\Delta X = 0$, $\Delta Y = 0$, $\Delta H = 0$ and $\Delta V = 0$). The maneuver typically resembles the one illustrated in the following figure: the aircraft pitches to a high AOA to stop the forward component of its velocity, puts in rudder at the top of the climb to point the aircraft down, and then dives to regain speed as it returns to the starting point.

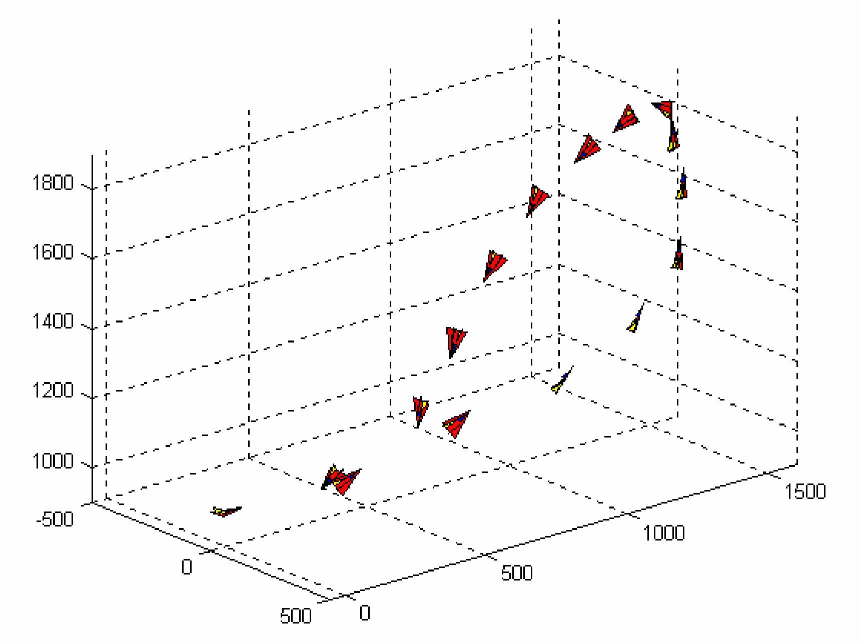


Figure 9. Herbst Reversal (max AOA = 30º)

Alternatively, if the aircraft can fly to a high enough AOA, it can fly the maneuver in a nearly straight line by pitching up until it is almost on its back, rolling so the aircraft is right-side-up again, and then diving to return to its initial velocity and position.

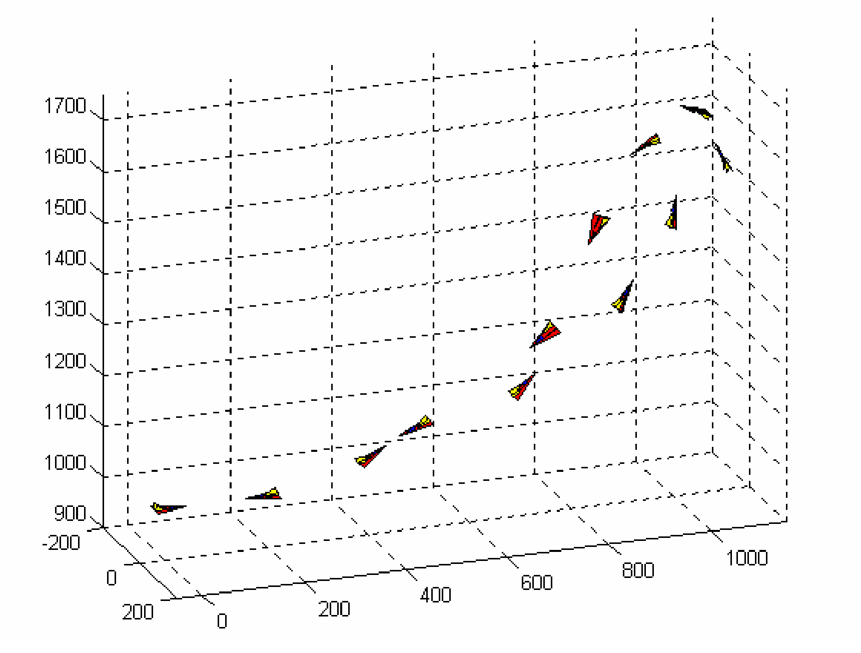


Figure 10. Herbst Reversal (max AOA = 90º)

In general, an aircraft that can operate at higher AOA can perform this maneuver with less displacement in the X and Y directions, which minimizes the maneuvering space and typically reduces the time necessary to complete the maneuver. The primary application for this maneuver is following an unsuccessful head-to-head to pass, where both aircraft need to reverse heading quickly in order to fire again.

D. FAMILIES OF MANEUVERS

After analyzing the five sample maneuvers from the previous section, it is apparent that there are several delta expressions that would be worth finding time-optimal solutions for:

- ◊ $\Delta\psi = 180^\circ$ (This delta is the primary goal of both the Split-S and the Herbst Reversal; the former has the additional constraints of $\Delta X = 0$ and $\Delta Y = 0$, and the latter has those constraints and two others: $\Delta H = 0$ and $\Delta V = 0$.)
- ◊ $\Delta\psi = 90^\circ$ (The goal of the first phase of the low yo-yo: to turn faster than the opponent.)
- ◊ $\Delta\theta$ & $\Delta\psi$ (These two deltas represent pointing the aircraft at an opponent, which is the goal of the second phase of the rolling scissors, and one of the goals of the Cobra.)
- ◊ $\Delta V = -0.5 \cdot V_i$ (Braking the aircraft is the other goal of the Cobra, and the goal of the first phase of the rolling scissors.)

These delta expressions will be generically referred to as “reversal,” “turning,” “pointing,” and “braking” maneuvers. Each expression represents an entire family of maneuvers: the reversal, for instance, when additionally constrained by $\Delta X = 0$ and $\Delta Y = 0$ (but unconstrained in altitude and velocity) may result in a time-optimal maneuver similar to the Split-S. When different constraints are added to the primary expression of $\Delta\psi = 180^\circ$, completely different maneuvers may result.

A 1982 paper by Well, Faber and Berger [Ref. 13] adds an additional two families of maneuvers to those listed above: "evasive" and "slicing" maneuvers. The goal of the evasive maneuvers was to make the attacking aircraft overshoot the defender; since this study is limited in scope to only one aircraft, this family of maneuvers is not a candidate for optimization. However, the braking maneuver in the first phase of the rolling scissors has the same goal as the evasive maneuvers, and the Cobra is generally considered an evasive maneuver designed to turn the defender into the attacker; therefore this family would be indirectly studied by optimizing the braking maneuvers. The goal of the slicing maneuvers was essentially to execute two reversals back to back, with no constraints on velocity, altitude, or final position. Again, this family would be indirectly studied by optimizing an unconstrained reversal.

Another family of maneuvers that could be studied are minimum time climbs. For a given increase in altitude, different constraints on (primarily) final position and velocity would result in multiple time-optimal maneuvers. However, the primary application for those types of maneuvers would be for missions other than air-to-air combat, so they won't be analyzed in this study. Other families of maneuvers could be identified by studying a more extensive list of combat, acrobatic and post-stall maneuvers. Ashley [Ref. 14] makes use of such a list for his study, choosing from a total of 62 combat and post-stall maneuvers a select few to analyze in detail. However, the seven families of maneuvers identified in this section are a very good representation of the types of maneuvers most commonly seen in short range air-to-air combat.

Based on the preceding paragraphs, the four families of maneuvers that will be optimized in this study are the reversal, turning, pointing and braking maneuver. However, these families will not be studied exhaustively, as there are too many possible combinations of constraints. Two common combinations that will be used on the selected families of maneuvers are an "energy" constraint and a "displacement" constraint. The energy constraint dictates that the altitude and velocity at the end of the maneuver must be the same as at the beginning of the maneuver ($\Delta H = 0$ and $\Delta V = 0$). This ensures that the aircraft has the same available energy, but hopefully a better position, as a result of the maneuver. The displacement constraint minimizes the distance the aircraft travels in the horizontal plane from its starting point. (In this case, ΔX and

ΔY can be set equal to 0ft, 100ft, 500ft, etc.) In a minimum time turn, for instance, this will tend to minimize the turn radius as well as the maneuver time. These chosen maneuvers will be discussed in more detail when the coding is discussed in Chapter V.

III. EQUATIONS OF MOTION

A. INTRODUCTION

The goal of this chapter is to develop the six-degree-of-freedom (6DOF) equations of motion for an aircraft. However, this will be a very succinct treatment of the subject, since the system of final equations are really what is most relevant to the rest of this study. For a more thorough derivation of these equations, there are excellent texts available by Etkin [Ref. 15], Schmidt [Ref. 16] and others that will fill any gaps that exist in this chapter. Also, many of these texts begin by deriving a set of equations for a point mass model, which will not be done here because this study is only concerned with the application of these equations to a (theoretically) physical aircraft.

The primary reference used for the derivations that follow was a research paper published by NATO's Advisory Group for Aerospace Research & Development (AGARD). [Ref. 17] Nearly all of the equations in this chapter come from that document, or were derived from more general expressions contained in the document. To jump ahead somewhat, the complete aircraft 6DOF equations of motion are a system of twelve coupled, nonlinear, ordinary differential equations. Those equations will be derived in essentially a two step process: first, the state variables in the aircraft problem will be identified and described; second, each of those variables will be evaluated in order to develop an equation governing how that variable changes over time. Also, the twelve state variables can be classified into four equal groups that define the aircraft's position, velocity, body rates and Euler angles; the next two sections of this chapter will follow that organization.

The final topics of this chapter will be the aircraft stability and control derivatives and aircraft controls, each of which has a very large influence on the aircraft equations of motion. Both topics will be discussed in general terms, while specific details about each topic will be addressed for each test aircraft in Chapter IV.

B. STATES

1. Position

The aircraft's position is defined by the distances X (down range), Y (cross range) and H (altitude), which are all measured from an arbitrary point on the earth's surface. These three terms are the first three state variables. If the aircraft is directly above the arbitrary point facing down range, then these three distances are aligned with the body axes of the aircraft: the x-direction going forward through the nose of the aircraft, the y-direction going out the right wing of the aircraft, and the z-direction going down. (The origin of this system is at the aircraft's CG.) Careful attention must be paid to sign convention: the z-direction is positive going down in order to create a right-handed coordinate system; however, altitude is measured in the negative z-direction, creating a left-handed system.

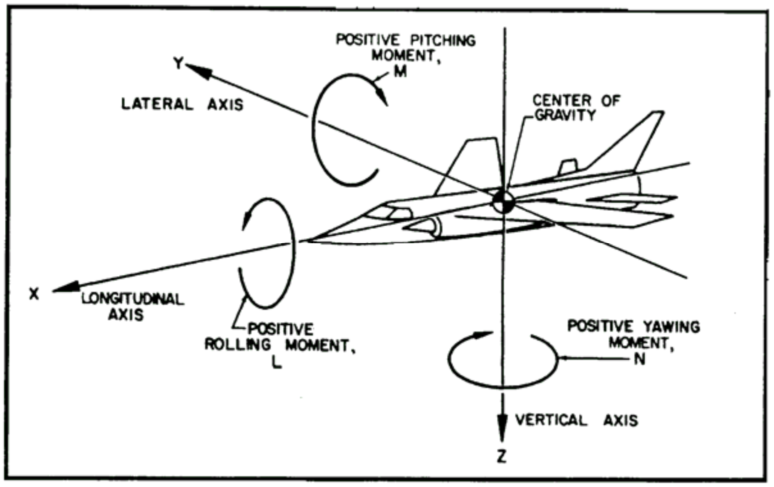

POSITIVE PITCHING MOMENT, M
Y
LATERAL AXIS
CENTER OF GRAVITY
X
LONGITUDINAL AXIS
POSITIVE ROLLING MOMENT, L
POSITIVE YAWING MOMENT, N
VERTICAL AXIS
Z

Figure 11. Body-Axis Coordinate System (From: Ref. 12, pp. 2)

2. Velocity

The velocity of the aircraft that is being evaluated in the following equations is what the AGARD text refers to as "wind-relative velocity." [Ref. 17] That is, the velocity of the aircraft relative to the surrounding air. This velocity can be expressed in two ways: using Cartesian coordinates or spherical coordinates. In Cartesian coordinates, the velocity terms are u, v and w, and they are aligned with the x, y and z-axes respectively. In spherical coordinates, the velocity terms are V, α, and β, where V is the magnitude of the velocity vector, α is the AOA, and β is the sideslip angle.

These two systems are related by Equations (1.1), (1.2), and (1.3), and depicted graphically in Figure 12.

$$V = \sqrt{u^2 + v^2 + w^2} \tag{1.1}$$

$$\begin{aligned} \alpha &= \tan^{-1}\frac{w}{u} \\ \beta &= \tan^{-1}\frac{u}{V} \end{aligned} \tag{1.2}$$

$$\begin{aligned} u &= V\cos\alpha\cos\beta \\ v &= V\sin\beta \\ w &= V\sin\alpha\cos\beta \end{aligned} \tag{1.3}$$

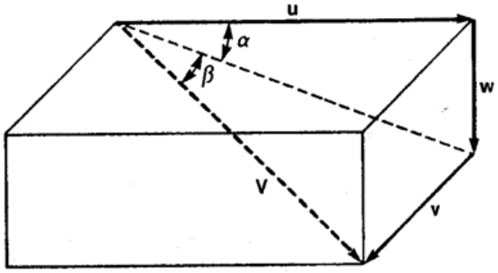


Figure 12. Wind-Relative Velocity Terms (From: Ref. 17, pp. 19)

The spherical coordinates are more relevant for this study, and therefore V, α, and β will be the second set of state variables. However, it will be useful in the next section to use the Cartesian system to simplify some of the equations, so Equations (1.3) will be used to put our final equations in terms of our velocity state variables.

3. Body Rates

The next set of state variables defines the rotation of the aircraft about its principal axes: p is the roll rate, q is the pitch rate, and r is the yaw rate. They correspond to rotations about the x, y and z-axis, respectively. Sign convention follows the right-hand rule: positive roll corresponds to right wing down, positive pitch corresponds to nose up, and positive yaw corresponds to nose right – see Figure 11.

4. Euler Angles

The familiar Euler angles of ϕ, θ, and ψ are the last three state variables for the aircraft equations of motion. They define the orientation of the aircraft with respect to earth. They are also the basis for transforming a vector from earth-fixed axes to the aircraft's body axes (where the earth-fixed axes are defined as x pointing north, y pointing east, and z pointing down). As seen in the example below (Equation (1.4)), where V_e is an arbitrary vector in earth-fixed coordinates, the vector is transformed to the aircraft's body axes by three consecutive rotations: first by the angle ψ around the z-axis, second by the angle θ around the new y_2-axis, and lastly by the angle ϕ around the new x_3-axis.

$$V_b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} V_e \qquad (1.4)$$

$$V_b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} V_e \qquad (1.5)$$

The product of the three rotation matrices, which is written out in Equation (1.5), will be referred to as the transformation matrix. This matrix and its inverse will be used in the upcoming section to transform certain state variables back and forth between the earth-fixed and body axes.

C. SIX-DEGREE-OF-FREEDOM EQUATIONS OF MOTION

Now that the state variables for the aircraft equations of motion have been identified, the four groups of variables will be analyzed in turn to develop equations for how these variables change over time.

1. Position Equations

The equations describing the change in aircraft position with respect to time are very straightforward: the velocity vector components (u, v and w) are transformed from the body axes to earth-fixed axes using the inverse of the matrix from Equation (1.5). The result of this transformation is Equations (1.6). *NOTE:* In these equations it is assumed that there is no wind. Additional terms could be added to account for components of the wind blowing from the three earth-fixed coordinate directions (north, east and up).

$$
\begin{aligned}
\dot{X} &= u(\cos\psi\cos\theta) + v(\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi) \\
&\quad + w(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\theta) \\[2ex]
\dot{Y} &= u(\sin\psi\cos\theta) + v(\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi) \\
&\quad + w(\sin\psi\sin\theta\sin\phi - \cos\psi\sin\phi) \\[2ex]
\dot{H} &= u(\sin\theta) - v(\cos\theta\sin\phi) - w(\cos\theta\cos\phi)
\end{aligned}
\tag{1.6}
$$

Since the spherical velocity terms are being used in this study, the Cartesian terms u, v and w can simply be replaced with their counterparts as listed in Equations (1.3) in order to obtain the final position state variable equations.

$$
\begin{aligned}
\dot{X} &= V\cos\alpha\cos\beta(\cos\psi\cos\theta) + V\sin\beta(\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi) \\
&\quad + V\sin\alpha\cos\beta(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\theta) \\[2ex]
\dot{Y} &= V\cos\alpha\cos\beta(\sin\psi\cos\theta) + V\sin\beta(\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi) \\
&\quad + V\sin\alpha\cos\beta(\sin\psi\sin\theta\sin\phi - \cos\psi\sin\phi) \\[2ex]
\dot{H} &= V\cos\alpha\cos\beta(\sin\theta) - V\sin\beta(\cos\theta\sin\phi) - V\sin\alpha\cos\beta(\cos\theta\cos\phi)
\end{aligned}
\tag{1.7}
$$

2. Velocity Equations

The derivation of the velocity equations starts by simply differentiating Equations (1.1), (1.2), and (1.3). This results in Equations (1.8). However, there are terms in these equations that need to be defined in order to make them useful in our complete system of equations for aircraft motion. In order to do this, the forces acting on an aircraft need to be analyzed.

$$\begin{aligned}
\dot{V} &= \frac{1}{V}(u\dot{u} + v\dot{v} + w\dot{w}) \\
\dot{\alpha} &= \frac{u\dot{w} - w\dot{u}}{u^2 + w^2} \\
\dot{\beta} &= \frac{V\dot{v} - v\dot{V}}{V^2\sqrt{1 - v^2/V^2}} = \frac{(u^2 + w^2)\dot{v} - vw\dot{w} - vu\dot{u}}{V^2\sqrt{u^2 + w^2}}
\end{aligned} \tag{1.8}$$

a. Forces

In this section two sets of equations for the forces acting on an aircraft will be derived; both will be used to further develop the equations of motion for the velocity state variables. The first set of equations is based on the principal of linear momentum. In Equation (1.9), F is the external applied force, m is the mass of the aircraft, and $\vec{V}$ is the velocity vector (with components u, v and w).

$$F = \frac{d}{dt}(m\vec{V}) \tag{1.9}$$

Transforming that equation to the rotating body-axis system results in Equation (1.10); ω in this equation is the angular velocity vector, with components p, q and r.

$$F = \frac{\delta}{\delta t}(m\vec{V}) + \omega \times (m\vec{V}) \tag{1.10}$$

Assuming that mass is constant, the above expression can be evaluated to come up with the first set of force equations - (1.11). *NOTE:* Constant mass is very reasonable for this study, because the maneuvers being analyzed should only last a few seconds, and therefore the change in mass from burning fuel will be insignificant.

$$\begin{aligned}
F_x &= m(\dot{u} + qw - rv) \\
F_y &= m(\dot{v} + ru - pw) \\
F_z &= m(\dot{w} + pv - qw)
\end{aligned} \tag{1.11}$$

The second set of force equations come from the external forces that are acting on the aircraft. These external forces include aerodynamic forces, forces due to gravity (and the weight of the aircraft), and the thrust produced by the aircraft. The aerodynamics forces are summarized in Equation (1.12), where $\overline{q}$ is the dynamic pressure, s is the reference wing area, and C_x, C_y and C_z are the coefficients of x, y and z force. (These coefficients will be discussed more in the next section.) The gravity forces are summarized in Equation (1.13).

$$\begin{aligned} F_{x(aero)} &= \overline{q}sC_x \\ F_{y(aero)} &= \overline{q}sC_y \\ F_{z(aero)} &= \overline{q}sC_z \end{aligned} \tag{1.12}$$

$$\begin{aligned} F_{x(grav)} &= -mg\sin\theta \\ F_{y(grav)} &= mg\sin\phi\cos\theta \\ F_{z(grav)} &= mg\cos\phi\cos\theta \end{aligned} \tag{1.13}$$

Adding these two equations together (plus a term for thrust) results in the second set of force equations - (1.14). *NOTE:* The terms F_X, F_Y and F_Z in these equations are equivalent to those found in Equations (1.11); i.e. in both instances the terms refer to the external forces acting on the aircraft. Therefore, the right hand side of these two sets of expressions could be set equal to each other.

$$\begin{aligned} F_x &= \overline{q}sC_x - mg\sin\theta + T \\ F_y &= \overline{q}sC_y + mg\sin\phi\cos\theta \\ F_z &= \overline{q}sC_z + mg\cos\phi\cos\theta \end{aligned} \tag{1.14}$$

Now that we have the necessary expressions for the forces acting on the aircraft, Equations (1.8) can be modified to a useable form. First, the variables $\dot{u}$, $\dot{v}$ and $\dot{w}$ will be replaced with their equivalent values from the first set of force equations - (1.11). This will replace those undefined variables with terms that are known, or can be solved for: the initial velocity vector (defined in this case by the Cartesian terms u, v and w), the initial angular velocity vector, and the initial external applied force. This substitution results in Equation (1.15):

$$\begin{aligned}
\dot{V} &= \frac{F_x}{m}\cos\alpha\cos\beta + \frac{F_y}{m}\sin\beta + \frac{F_z}{m}\sin\alpha\cos\beta \\
\dot{\alpha} &= \frac{1}{mv\cos\beta}(F_z\cos\alpha - F_x\sin\alpha) + q - \tan\beta(p\cos\alpha + r\sin\alpha) \\
\dot{\beta} &= \frac{\cos\beta}{mV}F_y + p\sin\alpha - r\cos\alpha - \frac{\sin\beta}{mV}(F_Z\sin\alpha + F_x\cos\alpha)
\end{aligned} \tag{1.15}$$

The final velocity state variable equations are found by replacing the external force terms with their equivalent expressions from the second set of force equations - (1.14).

$$\begin{aligned}
\dot{V} &= -\frac{\bar{q}s}{m}(C_D\cos\beta - C_Y\sin\beta) + g(\cos\phi\cos\theta\sin\alpha\cos\beta \\
&\quad + \sin\phi\cos\theta\sin\beta - \sin\theta\cos\alpha\cos\beta) + \frac{T}{m}\cos\alpha\cos\beta \\
\\
\dot{\alpha} &= -\frac{\bar{q}s}{mV\cos\beta}C_L + q - \tan\beta(p\cos\alpha + r\sin\alpha) \\
&\quad + \frac{g}{V\cos\beta}(\cos\phi\cos\theta\cos\alpha + \sin\theta\sin\alpha) - \frac{T\sin\alpha}{mV\cos\beta} \\
\\
\dot{\beta} &= \frac{\bar{q}s}{mV}(C_Y\cos\beta + C_D\sin\beta) + p\sin\alpha - r\cos\alpha + \frac{g}{V}\cos\beta\sin\phi\sin\theta \\
&\quad + \frac{\sin\beta}{V}(g\cos\alpha\sin\theta - g\sin\alpha\cos\phi\cos\theta + \frac{T}{m}\cos\alpha)
\end{aligned} \tag{1.16}$$

NOTE: The terms C_L and C_D that appear in this equation are related to the C_X and C_Z terms in Equation (1.14) by Equation (1.17). Instead of being aligned with the body axes of the aircraft, C_L and C_D are aligned with the stability axes of the aircraft, which is the reference frame that will be used to analyze the stability and control derivatives.

$$\begin{aligned}
C_L &= -C_Z\cos\alpha + C_X\sin\alpha \\
C_D &= -C_X\cos\alpha - C_Z\sin\alpha
\end{aligned} \tag{1.17}$$

3. Body Rate Equations

The body rates of the aircraft are highly dependent on the aircraft's principal moments of inertia, I_X, I_Y and I_Z. For example, an aircraft with a large roll moment of inertia (I_X) – perhaps an aircraft with large external stores on the tips of its wings – will not be able to roll quickly. Similar illustrations would apply to the other moments of inertia. Therefore, it is not surprising that the equations of motion that govern the change in body rates with respect to time are largely functions of the moments of inertia. In order to develop these equations, the moments acting on an aircraft will be analyzed.

a. Moments

In this section two sets of equations for the moments acting on an aircraft will be derived; both will be used to develop the equations of motion for the body rate state variables. The first set of equations is based on the principal of angular momentum. In Equation (1.18), M is the external applied moment, and h is the angular moment vector (both about the CG). The angular moment is defined in Equation (1.19). (The matrix in that equation is the inertia tensor [Ref. 17]; the diagonal elements in the matrix will be recognized as the principal moments of inertia that were discussed earlier. Not all of the terms in the inertia tensor are applicable to an aircraft, but they will all be retained for the moment.)

$$M = \frac{d}{dt}(h) \tag{1.18}$$

$$h = \begin{bmatrix} I_X & -I_{XY} & -I_{XZ} \\ -I_{XY} & I_Y & -I_{YZ} \\ -I_{XZ} & -I_{YZ} & I_Z \end{bmatrix} \omega \tag{1.19}$$

Transforming that Equation (1.18) to the rotating body-axis system results in Equation (1.20); ω is the angular velocity vector, with components p, q and r.

$$M = \frac{\delta}{\delta t}(h) + \omega \times h \tag{1.20}$$

Assuming that mass and the inertia tensor are constant, the above expression can be evaluated to come up with the first set of moment equations - (1.21).

$$\begin{aligned} M_X &= \dot{p}I_X - \dot{q}I_{XY} - \dot{r}I_{XZ} + qr(I_Z - I_Y) + (r^2 - q^2)I_{YZ} - pqI_{XZ} + rpI_{XY} \\ M_Y &= -\dot{p}I_{XY} + \dot{q}I_Y - \dot{r}I_{YZ} + rp(I_X - I_Z) + (p^2 - r^2)I_{XZ} - qrI_{XY} + pqI_{YZ} \\ M_Z &= -\dot{p}I_{XZ} - \dot{q}I_{YZ} + \dot{r}I_Z + pq(I_Y - I_X) + (q^2 - p^2)I_{XY} - rpI_{YZ} + qrI_{XZ} \end{aligned} \tag{1.21}$$

The second set of moment equations come from the external moments that are acting on the aircraft. These external moments are caused by the aerodynamic effects that tend to roll, pitch and yaw the aircraft. The aerodynamics moments are summarized in Equation (1.22), where $\bar{q}$ is the dynamic pressure, s is the wing reference area, b is the reference wingspan, c is the reference wing chord, and C_l, C_m and C_n are the coefficients of roll, pitch and yaw moment. (These coefficients will be discussed more in the next section.)

$$\begin{aligned} M_{X(aero)} &= \bar{q}sbC_l \\ M_{Y(aero)} &= \bar{q}scC_m \\ M_{Z(aero)} &= \bar{q}sbC_n \end{aligned} \tag{1.22}$$

NOTE: As before, the terms M_X, M_Y and M_Z in Equations (1.21) and (1.22) are equivalent; i.e. in both instances the terms refer to the external moments acting on the aircraft. Therefore, the right hand side of these two sets of expressions can be set equal to each other.

Now that we have the necessary expressions for the moments acting on the aircraft, we can develop an equation governing the change of the body rates over time by setting the two moment equations equal to each other. This results in Equation (1.23).

$$\begin{aligned} \dot{p}I_X - \dot{q}I_{XY} - \dot{r}I_{XZ} &= \bar{q}sbC_l + qr(I_Y - I_Z) + (q^2 - r^2)I_{YZ} + pqI_{XZ} - rpI_{XY} \\ -\dot{p}I_{XY} + \dot{q}I_Y - \dot{r}I_{YZ} &= \bar{q}scC_m + rp(I_Z - I_X) + (r^2 - p^2)I_{XZ} + qrI_{XY} - pqI_{YZ} \\ -\dot{p}I_{XZ} - \dot{q}I_{YZ} + \dot{r}I_Z &= \bar{q}sbC_n + pq(I_X - I_Y) + (p^2 - q^2)I_{XY} + rpI_{YZ} - qrI_{XZ} \end{aligned} \tag{1.23}$$

However, it would be helpful if these equations could be uncoupled so that there is a separate equation governing each body rate. This will be done by first eliminating terms from the inertia tensor that are negligible for our aircraft study: I_{XY} and I_{YZ}. (I_{XZ} will be retained, although it is usually an order of magnitude smaller than the principal moment of inertia terms for a typical aircraft shape.) This results in a separate equation for $\dot{q}$, and two equations with $\dot{p}$ and $\dot{r}$. These two equations can be solved simultaneously to find separate expressions for $\dot{p}$ and $\dot{r}$. This process results in the final body rate state variable equations:

$$
\begin{aligned}
\dot{p} &= \frac{I_Z}{I_X I_Z - I_{XZ}{}^2}(\bar{q}sbC_l + qr(I_Y - I_Z) + pqI_{XZ}) \\
&\quad + \frac{I_{XZ}}{I_X I_Z - I_{XZ}{}^2}(\bar{q}sbC_n + pq(I_X - I_Y) - qrI_{XZ})
\end{aligned}
$$

$$
\dot{q} = \frac{1}{I_Y}(\bar{q}scC_m + rp(I_Z - I_X) + (r^2 - p^2)I_{XZ}) \qquad (1.24)
$$

$$
\begin{aligned}
\dot{r} &= \frac{I_{XZ}}{I_X I_Z - I_{XZ}{}^2}(\bar{q}sbC_l + qr(I_Y - I_Z) + pqI_{XZ}) \\
&\quad + \frac{I_X}{I_X I_Z - I_{XZ}{}^2}(\bar{q}sbC_n + pq(I_X - I_Y) - qrI_{XZ})
\end{aligned}
$$

4. Euler Angle Equations

The derivation of the equations governing the change in Euler angles with respect to time is similar to the earlier derivation of the position equations. For those equations, while velocity is normally defined as the rate of change of position with respect to time, the aircraft's velocity vector had to be transformed from the body axes to the earth-fixed axes that position is measured in. Likewise, while the body rates are defined as the rate of change of Euler angles with respect to time (when bank angle and pitch angle equal zero), they must first be converted to same coordinate system as the Euler angle rates. This will be done in reverse order, and then the resulting transformation matrix will be inverted to obtain the necessary expression.

Unfortunately, the coordinate system of the Euler angle rates is not orthogonal. [Ref. 17] Recall the order of rotations in Equation (1.4): the first rotation was by the angle ψ around the earth-fixed z-axis, therefore $\dot{\psi}$ will have to be rotated by the angles θ and ϕ to be transformed to the aircraft's body axes. The second rotation was by the angle θ around the new y$_2$-axis, therefore $\dot{\theta}$ will have to be rotated by the angle ϕ. The third rotation was by the angle ϕ around the new x$_3$-axis; since this axis is identical to the body's x-axis, $\dot{\phi}$ is already in the body axis coordinate system. Equation (1.25) summarizes these rotations.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -\sin\phi \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{1.25}$$

Finally, the Euler angle state variable equations are given by inverting the previous equation.

$$\begin{aligned} \dot{\phi} &= p + q\tan\theta\sin\phi + r\tan\theta\cos\phi \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= r\cos\phi\sec\theta + q\sin\phi\sec\theta \end{aligned} \tag{1.26}$$

5. Complete Equations

The table on the following page summarizes the complete 6DOF equations of motion for an aircraft in flight. These equations assume non-relativistic mechanics, a rigid vehicle, no wind, constant aircraft mass, constant inertia tensor, negligible cross products of inertia I$_{XY}$ and I$_{YZ}$, and thrust that acts along the x-axis. The choice of spherical velocity terms creates singularities at V=0 and $\beta = \pm 90^\circ$, and there is an additional singularity at $\theta = \pm 90^\circ$. [Ref. 17]

Table 2. Complete Six-Degree-of-Freedom Equations of Motion

Position Equations
$\dot{X} = V\cos\alpha\cos\beta(\cos\psi\cos\theta) + V\sin\beta(\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi) + V\sin\alpha\cos\beta(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\theta)$ $\dot{Y} = V\cos\alpha\cos\beta(\sin\psi\cos\theta) + V\sin\beta(\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi) + V\sin\alpha\cos\beta(\sin\psi\sin\theta\sin\phi - \cos\psi\sin\phi)$ $\dot{H} = V\cos\alpha\cos\beta(\sin\theta) - V\sin\beta(\cos\theta\sin\phi) - V\sin\alpha\cos\beta(\cos\theta\cos\phi)$
Velocity Equations
$\dot{V} = -\frac{\bar{q}s}{m}(C_D\cos\beta - C_Y\sin\beta) + g(\cos\phi\cos\theta\sin\alpha\cos\beta + \sin\phi\cos\theta\sin\beta - \sin\theta\cos\alpha\cos\beta) + \frac{T}{m}\cos\alpha\cos\beta$ $\dot{\alpha} = -\frac{\bar{q}s}{mV\cos\beta}C_L + q - \tan\beta(p\cos\alpha + r\sin\alpha) + \frac{g}{V\cos\beta}(\cos\phi\cos\theta\cos\alpha + \sin\theta\sin\alpha) - \frac{T\sin\alpha}{mV\cos\beta}$ $\dot{\beta} = \frac{\bar{q}s}{mV}(C_Y\cos\beta + C_D\sin\beta) + p\sin\alpha - r\cos\alpha + \frac{g}{V}\cos\beta\sin\phi\sin\theta + \frac{\sin\beta}{V}(g\cos\alpha\sin\theta - g\sin\alpha\cos\phi\cos\theta + \frac{T}{m}\cos\alpha)$
Body Rate Equations
$\dot{p} = \frac{I_Z}{I_X I_Z - I_{XZ}^2}(\bar{q}sbC_l + qr(I_Y - I_Z) + pqI_{XZ}) + \frac{I_{XZ}}{I_X I_Z - I_{XZ}^2}(\bar{q}sbC_n + pq(I_X - I_Y) - qrI_{XZ})$ $\dot{q} = \frac{1}{I_Y}(\bar{q}scC_m + rp(I_Z - I_X) + (r^2 - p^2)I_{XZ})$ $\dot{r} = \frac{I_{XZ}}{I_X I_Z - I_{XZ}^2}(\bar{q}sbC_l + qr(I_Y - I_Z) + pqI_{XZ}) + \frac{I_X}{I_X I_Z - I_{XZ}^2}(\bar{q}sbC_n + pq(I_X - I_Y) - qrI_{XZ})$
Euler Angle Equations
$\dot{\phi} = p + q\tan\theta\sin\phi + r\tan\theta\cos\phi$ $\dot{\theta} = q\cos\phi - r\sin\phi$ $\dot{\psi} = r\cos\phi\sec\theta + q\sin\phi\sec\theta$

D. STABILITY AND CONTROL DERIVATIVES

In the 6DOF equations of motion, there were six terms that were described as force and moment coefficients: C_L, C_D, C_Y, C_l, Cm and C_n. Each of these coefficients, in turn, is a function of non-dimensional terms called stability and control (S&C) derivatives. For this study, the force and moment coefficients will be expressed as the sum of these derivatives, following the format of this equation:

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_q} q\left(\frac{c}{2V}\right) + C_{L_\delta}\delta$$

In the equation, C_{L_0}, C_{L_α}, C_{L_q} and C_{L_δ} are all S&C derivatives. C_{L_0} is the "baseline" lift coefficient, which is the lift coefficient at zero AOA with no control surfaces deflected. C_{L_α} is the change in lift coefficient due to AOA; this coefficient is multiplied by the AOA to calculate the incremental increase or decrease in lift coefficient due to AOA. C_{L_q} (the change in lift coefficient due to pitch rate) functions in the same way, with one difference: the pitch rate must be non-dimensionalized by the semi-chord over the velocity:

$$C_{L_q} q\left(\frac{c}{2V}\right) \rightarrow \frac{1}{\deg}\cdot\frac{\deg}{\sec}\left(\frac{ft}{ft/\sec}\right) = none$$

(Schmidt explains the reason for using the semi-chord instead of the chord as a convention from thin-airfoil theory. [Ref. 16]) Finally, the C_{L_δ} term represents the change in lift coefficient due to a control deflection, and can be several terms depending on how many control surfaces the aircraft has that affect lift. Assuming for this example that the aircraft has only one control surface to effect lift, C_{L_δ} is simply multiplied by that control surface's angular deflection to calculate the increment change in lift coefficient. The sum of these terms at a given instant in flight give the total lift coefficient needed in the aircraft equations of motion.

As observed in the example of lift coefficient, there are several "categories" of S&C derivatives: baseline values (C_{L_0}), control terms (C_{L_δ}), stability terms (C_{L_α}), and what are referred to as damping terms (C_{L_q}). In general, the longitudinal coefficients (C_L, C_D, and Cm) have baseline values, while the lateral-directional coefficients (C_Y, C_l and C_n) do not. Also, baseline values are not to be confused with bias values, which AGARD describes as "linear extrapolation[s] from the average α and δ of the maneuver to the zero point." [Ref. 17] (Although bias values are added to their respective coefficients in the same way that the baseline terms are.) The stability terms are derivatives of the force and moment coefficients with respect to α (for longitudinal terms) and β (for lateral-directional terms). The damping terms are derivatives of the force and moment coefficients with respect to q (for longitudinal terms) and p and r (for lateral-directional terms). Finally, the control terms are derivatives with respect to control surface deflections, which will be discussed in the next section. More detailed analyses of the S&C derivatives and the physics behind them can be found in the texts by Etkin and Schmidt referenced earlier.

An important fact about the S&C derivatives is that they are functions of the aircraft and the aircraft's state. Therefore, for a given aircraft, there is a different set of S&C derivatives for any flight condition. However, the non-dimensional S&C derivatives depend primarily on the variables of AOA, altitude and velocity. For this study, altitude change during any of the time-optimal maneuvers will not be enough to significantly affect the S&C derivatives, so that variable can be eliminated. Velocity change will definitely be a factor, especially since a sharp velocity drop is very characteristic of post-stall maneuvers; however, that data was not available for HARV, so the S&C derivatives for that aircraft will be calculated for a reference speed of Mach 0.4, which should be a very reasonable and accurate approximation due to the short duration of any low speeds. AOA change will be taken into consideration by expressing the S&C derivatives as functions of AOA. (This will also cause some of the force and moment coefficients to be nonlinear since, for example, $C_{L_\alpha}(\alpha)\cdot\alpha$ is not linear.) More specific information about the S&C derivatives will be discussed for each test aircraft in the next chapter.

E. CONTROLS

The aircraft controls are a pilot's feedback system into the equations of motion. Control inputs affect the force and moment coefficients, and hence allow a pilot to affect the state of the aircraft. Physically, nearly all control surfaces on aircraft today function by affecting the flow of air over the aircraft to induce a force or moment in the desired direction. TV is the exception to this rule. As mentioned in the last section, a control's influence on the force and moment coefficients is calculated by multiplying a control derivative by the corresponding control surface deflection. This deflection is the angular displacement of the control surface from a reference position. The sign convention followed in this study will be that trailing edge down (TED) and trailing edge left (TEL) are positive deflections. For anti-symmetric controls, the displacement will be measured as the left side position minus the right side position (i.e. with the left aileron five degrees TED and the right aileron five degrees TEU, the aileron deflection will be a positive ten degrees.)

There is a long list of control surfaces that are in use by current conventional aircraft, and the three test aircraft of this study will use most of those. For longitudinal control, the primary controls will be the elevator, stabilators, and canards. Secondary controls include pitch TV, TEF’s, LEF’s, symmetric aileron deflection, and speedbrakes. For lateral-directional control, the primary controls will be the ailerons and rudder. Secondary controls include yaw TV and differential (or anti-symmetric) stabilator deflection.

One last note on controls before describing the test aircraft: controls are limited by a physical maximum deflection (positive and negative), and a mechanical maximum deflection rate. Both of these limits will be incorporated into the aerodynamic models described in the next chapter.

IV. AERODYNAMIC MODELS

A. SELECTED TEST AIRCRAFT

Three different aircraft were used in this study. The first of these aircraft, the Navion, was used to "calibrate" the aircraft optimization code. It is a very basic airplane with conventional control surfaces, and therefore was a good candidate to use for debugging the code and running new maneuvers. The other two aircraft are of primary interest for finding time-optimized combat maneuvers: the F-18 HARV and the UCAV-X (based on the X-31 EFM). Both are STVF's, and are representative of the current capability of fighter aircraft. However, the two aircraft are different in almost every other aspect. The F/A-18 Hornet that the HARV is modified from is a two-engine multi-role strike fighter, while the UCAV-X is a single-engine fighter designed specifically for air-to-air combat. Hence, they will likely perform differently when given the same maneuver. The next sections describe each aircraft in turn, as well as how the aircraft data was coded.

B. NAVION

Figure 13. Navion

1. General Description

The NA-154 Navion was built by North American Aviation, Inc. following World War II, and was first flown in April, 1946. A prototype was created for the United States Army Air Force and designated the L-17A. Several other variants of this aircraft were produced for the US Air Force over the next twenty years, and were used for training and personnel transport, among other missions. Currently, many individuals own and fly the Navion as a personal airplane. [Ref. 18]

The physical and aerodynamic data for the Navion was found in an appendix of Schmidt's text [Ref. 16], and was coded into a MATLAB script file titled *LoadAircraft1.m*. The script file creates a single array in the MATLAB environment, and every element of that array stores a different piece of information about the Navion: wing chord, wing span, mass, etc. When the optimization code runs the script file, all of the data about the aircraft is made available for calculating the equations of motion and solving for a time-optimal maneuver. (This process will be discussed in more detail in the second section of Chapter V.) *LoadAircraft1.m* is included in Appendix A for reference.

2. Physical Parameters

The Navion is a simple propeller-driven airplane that can seat four people in most configurations. It is fairly small when compared to the other aircraft in this study, and definitely much lighter. It has low-mounted rectangular wings with some built-in dihedral and a very small amount of taper. It has the conventional control surfaces of an elevator, a pair of ailerons, and a rudder to control pitch, roll and yaw, respectively. The maximum deflection of these control surfaces was estimated (as were the AOA and load factor limits), and no deflection rate limits were included in the aerodynamic model. The lack of rate limits means that the controls can deflect instantaneously from one setting to another, which is of course unrealistic. On the other hand, this greatly simplifies the model. Since the purpose of the Navion is to validate and test the optimization code, this unfaithfulness to the reality of a Navion is acceptable. Lastly, "cruise thrust" was calculated by setting drag (at the reference altitude and velocity) equal to thrust; this was then defined as 80% of max thrust. (Some of the pertinent data for the Navion is listed in the table on the following page.)

Table 3. Summary of Navion Physical Data

Navion (general aviation aircraft)			
length:	27.5 ft	*reference altitude:*	sea level
wing span:	33.4 ft	*reference velocity:*	176 fps
weight:	2,753 lb		M = 0.16
max AOA:	20º	*T/W (at altitude):*	0.15
longitudinal controls:	Elevator		
lateral-directional " :	ailerons, rudder		

3. Aerodynamic Characteristics

The aerodynamic characteristics of an aircraft are described by that aircraft's S&C derivatives. For the Navion, these derivatives are all constant values, which can be seen by looking at the *LoadAircraft1.m* file in Appendix A. As mentioned in the previous chapter, however, S&C derivatives are actually functions of the following state variables: AOA, altitude and velocity. The flight condition these S&C derivatives are valid for, then, is when the Navion is at zero AOA, and flying at the reference altitude and velocity that are listed in the above table. (Since the Navion can't fly at sea level, a starting altitude of 1000 or 2000 ft will be used instead; the difference should not be a source of error.)

During the course of most maneuvers, though, the Navion will at some point be flying at speeds and AOA other than the reference values. Any speed difference from the reference value should not be a source of major error, for the following reasons: first, the velocity range the Navion can fly at (from stall speed to perhaps Mach 0.2 in a dive) is very small, and therefore the difference from the reference velocity will never be an extremely large value. Second, at low subsonic speeds and small AOA, a velocity difference of less than Mach 0.2 will not produce a significant change in the S&C derivatives; since the entire velocity range of the Navion is smaller than Mach 0.2, it is fair to claim that the constant values used for the S&C derivatives are representative of the Navion flying at any speed. Differences in AOA from the reference value of zero is a different story, as a 10º difference can make a noticeable change in the S&C derivatives. This is a small sacrifice in accuracy that has to made for an aerodynamic model as neat and simple as the Navion.

C. F-18 HARV

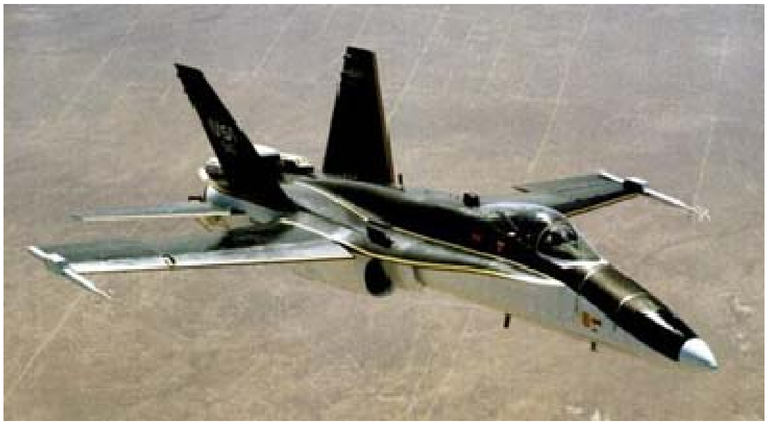


Figure 14. F-18 HARV

1. General Description

The F/A-18 Hornet was built by the McDonnell Douglas Corporation for the Navy and Marine Corps, and first flew in November 1978. As a strike fighter it can perform both air-to-air and air-to-ground missions, including escort, interdiction, close air support, etc. Until recently, two squadrons of Hornets were stationed about every US Aircraft Carrier; currently the Navy’s Hornet squadrons are in the process of being replaced with the newest variant of the aircraft, the F/A-18E and F Super Hornet.

The HARV was modified from a pre-production F-18, and hence had not yet been designated as an F/A-18. It was only the sixth full-scale Hornet produced by McDonnell Douglas (Number 160780), and was used by the Naval Air Test Center at Patuxent River, MD for high AOA and spin testing. In 1984, the Navy delivered the aircraft to the NASA Dryden Flight Research Center in Edwards, CA. After extensive reconstruction and rewiring, the HARV officially became NASA test aircraft #840, and was nicknamed the “Silk Purse.” The HARV had a distinguished nine year career with NASA from 1987 to 1996, completing 385 test flights and demonstrating controlled maneuvers at high AOA. [Ref. 4]

The primary sources of data on the HARV were two NASA Technical Papers written in 1997 and 1999 by Iliff and Wang [Ref. 19-20]. The primary goal of these papers was to determine the S&C derivatives of the HARV at high AOA. (This will be discussed more in the appropriate section below.) Those papers also included detailed physical and control data on the HARV. Three other documents were needed in order to create a complete aerodynamic model of the HARV: a West Virginia University (WVU) report sponsored by NASA [Ref. 21] and the original McDonnell Douglas Corporation (MDC) aerodynamic datasets on the F/A-18 [Ref. 22-23]. The data from these five sources was coded into the MATLAB file *LoadAircraft2.m,* which is included in Appendix A.

2. Physical Parameters

The HARV is a single-seat, twin-engine fighter easily recognized by its twin vertical stabilizers that are canted outward from the centerline. Its wings are swept and tapered, and connected to the body with leading-edge extensions. In place of elevators, the HARV has all-moving horizontal stabilators; when deflected simultaneously they affect pitch, but they can also be deflected differentially like ailerons in order to affect roll (and to a much smaller extent, yaw). The TEF’s can likewise be deflected differentially, but they were not flight-tested and analyzed in this configuration in the Iliff and Wang reports, therefore in this model they will only be used for longitudinal (pitch) control. In addition to the list of controls in the table below, the HARV is equipped with LEF’s and a speedbrake located between the vertical stabilizers; as little to no data was available on the effects of these controls in Ref. 19-23, they are not included in the aerodynamic model. (This should not affect the accuracy of the model very much as they are not primary controls in the longitudinal or lateral-directional channel.)

The TV system on the HARV is comprised of three vanes circling the edge of each engine. These vanes can be deflected in concert to achieve an equivalent “pitch vane” or “yaw vane” deflection. (The moment arm between the engines is not large enough to create a rolling moment by deflecting thrust asymmetrically.) While more detailed information on the TV system can be found in the Iliff and Wang reports, the important information necessary for the HARV aerodynamic model was the following: the control derivatives associated with the pitch and yaw vanes, the maximum and minimum pitch and yaw vane deflections, as well as the maximum deflection rate of the vanes.

Unlike the Navion, the deflection limits and deflection rate limits for *all* control surfaces of the HARV were taken directly from the Iliff and Wang reports. Therefore the HARV model should have the same control power and physical limitations as the actual aircraft. The weight used in the model was the weight of the aircraft in its Phase II and Phase III flight test configuration: 60% fuel, TV system installed, pilot and support equipment included. The maximum available thrust was calculated using a program named GASTURB: all of the information for the General Electric F404-GE-400 engine that powers the HARV was entered into the program, as well as the reference altitude and Mach number. While each engine produces 16,000 lb of force on a test stand at sea level, GASTURB calculated that they only each produce 6,173 lb of force at the specified flight condition. The resulting T/W ratio and other pertinent physical data are listed in the table below.

Table 4. Summary of F-18 HARV Physical Data

F-18 HARV (Navy fighter converted by NASA)			
length:	56 ft	*reference altitude:*	30,000 ft
wing span:	37.4 ft	*reference velocity:*	398 fps
weight:	36,099 lb		M = 0.4
max AOA:	70°	*T/W (at altitude):*	0.34
longitudinal controls:	stabilator, TEF’s, symmetric ailerons, pitch TV		
lateral-directional “ :	ailerons, rudder, differential stabilator, yaw TV		

3. Aerodynamic Characteristics

The S&C derivatives for the HARV were calculated at or near the altitude and Mach number listed in the table above. They were analyzed from 10º AOA to 60 or 70º AOA, in 10º increments. Unfortunately this is a very large interval, especially at high AOA where S&C derivative values can change noticeably in less than 5º. Nevertheless, while the data may be sparse, it is still very good, and certainly the best high AOA flight test data available for the HARV.

Originally this data was coded into a separate file named *SCTables2.m* as look-up tables for each S&C derivative; the first version of *LoadAircraft2.m* would take the current AOA, access the look-up table file, and calculate the value for each S&C derivative at that specified AOA. For AOA values between those stored in the look-up tables, MATLAB would interpolate the appropriate value using a cubic curve-fit. This curve-fit resulted in S&C derivative plots, or curves, that very closely matched Iliff and Wang's fairings in the case of every derivative. For AOA values outside the range of those in the look-up tables (0-10º, and in some cases 60-70º), MATLAB would extrapolate the appropriate value, again using a cubic curve-fit. As extrapolation can lead to notoriously bad results, every S&C derivative plot was analyzed (42 in the case of the HARV): for those that departed to unrealistic values at zero AOA, a zero AOA value was *chosen* for that derivative and inserted into the look-up table. This process resulted in a look-up table that produced very good values for S&C derivatives throughout the AOA regime of the HARV that was desired for this study, 0-70º.

The WVU report provided all of the axial force derivatives. The Iliff and Wang report on longitudinal S&C derivatives [Ref. 19] only contained normal force and pitching moment derivatives; in order to calculate the lift and drag, both normal *and* axial derivatives were needed. (Normal and axial force coefficients are related to lift and drag coefficients by the equations below.)

$$C_N = -C_Z$$
$$C_A = -C_X$$

$$C_L = -C_Z \cos\alpha + C_X \sin\alpha$$
$$C_D = -C_X \cos\alpha - C_Z \sin\alpha$$

The axial derivatives from the WVU report were on nearly the same scale as those in the Iliff and Wang reports; i.e. 10-60° AOA in 10° increments. Unfortunately, very little information was contained in the report about what flight condition the derivatives were calculated from. This was solved by comparing derivatives that appeared in both the WVU report and the Iliff and Wang reports: based on the similarities, it is likely that the two reports used exactly the same flight data. Therefore, no errors were incurred by using information from both sources in the same aerodynamic model.

The MDC documents provided the baseline S&C derivatives, which were not included in the Iliff and Wang reports: C_{D_0}, C_{L_0} and C_{m_0}. (These baseline derivatives are the same for the HARV as they are for the standard F/A-18, and were obtained through wind tunnel testing.) The derivatives were plotted for different Mach numbers, which unfortunately did not include Mach 0.4 (the reference velocity for the other reports). The values of the derivatives at Mach 0.6 were chosen as an acceptable alternative.

One final note about the aerodynamic model for the HARV: while the table look-up file worked, it had the unacceptable side effect of drastically increasing the amount of time it took to optimize a maneuver (run times will be discussed extensively in the next chapter). For that reason, the look-up tables were replaced by 4th or 5th order polynomial functions of AOA using the MATLAB curve-fitting command, “polyfit.” These curves were compared against those produced by the table look-up file, and no serious discrepancies existed. (As the table look-up plots are estimates as well, the important test of the accuracy for the curve-fitted plots was whether or not they captured most of the flight tested data points. Only five of the 42 S&C derivatives “missed” more than two of the seven test points: $C_{m_\alpha}, C_{Y_\beta}, C_{l_r}, C_{n_r}, C_{n_{\delta ds}}$. This was generally due to a “double peak” that was difficult for a 4th or 5th order polynomial to imitate [higher order polynomials triggered a “badly conditioned” warning in MATLAB]. In spite of this, even those five derivatives matched the trends predicted by Iliff and Wang and by MATLAB’s interpolation routine, and were close enough in value to retain in the aerodynamic model without modification.) Both the first (table look-up) and second (curve-fit) versions of *LoadAircraft2.m* are included in Appendix A for comparison.

D. UCAV-X

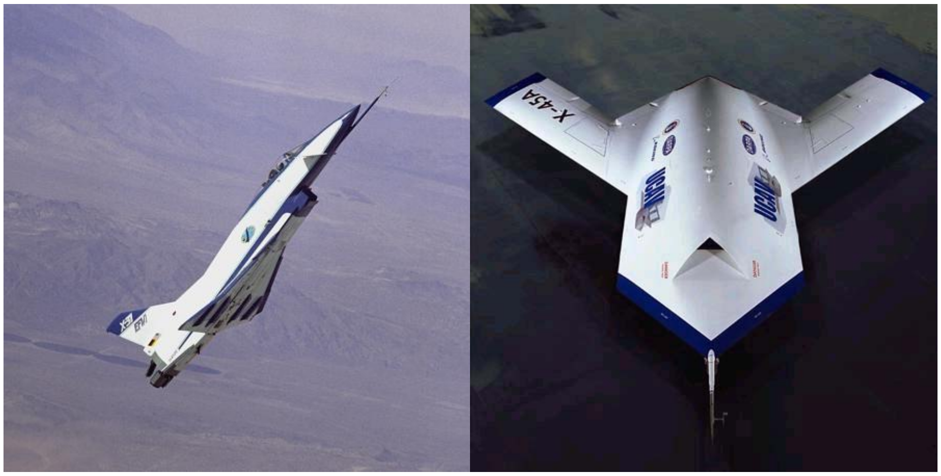


Figure 15. X-31 EFM and X-45 UCAV

1. General Description

The UCAV-X is a fictitious aircraft based primarily on the physical and aerodynamic characteristics of the X-31 Enhanced Fighter Maneuverability aircraft. The Boeing X-45 pictured next to the X-31 in the figure above is one possible design for the UCAV-X. (More information can be found on the X-45 at Boeing's website [Ref. 11] or the Navy's UAV website [Ref. 24].) The X-31 was a joint venture between the US and Germany: DARPA funded NASA's contribution to the project through NavAir, while the Germany Ministry of Defense directly funded MBB's contribution. Two aircraft were built for the program in 1990, and several hundred hours of test flights were carried out over the next several years at the Naval Air Test Center. [Ref. 25] Unlike the HARV program, which was a study of the feasibility and controllability of post-stall flight, the EFM program was a study about the military application and combat effectiveness of post-stall flight. The X-31 was designed, in essence, to be the ultimate dogfighter in both short and medium range air-to-air combat.

The primary sources of X-31 data were three technical documents published by Rockwell Aerospace (which was purchased by the Boeing Company in 1996). The first of these documents, the system manual [Ref. 26], contained general information about the EFM program, details on the construction and operating limits of the aircraft, as well as control data. The other two documents, the aerodynamic datasets [Ref. 27-28], contained a complete listing of the longitudinal and lateral-directional S&C derivatives for the X-31 (respectively). The datasets were generated by extensive wind tunnel testing, and calculated S&C derivatives at eight different Mach numbers (0, 0.2, 0.5, 0.8, 0.9, 0.95, 1.2, 1.6) and for AOA values ranging from –20º to 90º (at intervals of 2º). The density of data created an *excellent* database for the UCAV-X model, and will be discussed more in the appropriate section. Data for the UCAV-X was coded into the file *LoadAircraft3.m*, as before. (As some of the X-31 data in this file is proprietary, it is not included in Appendix A.)

2. Physical Parameters

The physical design of the X-31 was based largely on the findings of Herbst and other researches at MBB between 1972 and 1980. (See Ref. 8, as well as that article’s bibliography.) As a result, the X-31 has a distinctive shape, with a narrow fuselage, low mounted delta wings, canards, and a trio of TV vanes attached to the end of the engine. (The configuration of these vanes is very similar to those on the HARV: one larger vane at the 12 o’clock position, and two smaller vanes at the 4 and 8 o’clock positions. The vanes can be deflected in concert to achieve an equivalent “pitch vane” or “yaw vane” deflection, and a corresponding pitching or yawing moment. The deflection also creates a small lift or side force.)

The X-31 has six sets of control surfaces (in addition to its TV system), as listed in the table at the end of this section. Primary longitudinal control is provided by the canards, secondary control by a pair of inboard TEF’s and pitch TV, and tertiary control by a speedbrake and two pairs of LEF’s. (The inboard and outboard LEF’s were coded as one control surface in the UCAV-X model, as they were always deflected in concert in the longitudinal aerodynamic dataset.) Roll control was provided by a pair of ailerons (also referred to as outboard TEF’s), and yaw control was provided by a rudder and TV. Deflection limits and rate limits were included in the UCAV-X model to increase its fidelity.

The flight test weight of the X-31 was used for the UCAV-X, although the UCAV-X in reality could be 2-3,000 lb lighter (and have a correspondingly higher T/W ratio). The X-31 is powered by the same engine as the HARV (the General Electric F404-GE-400), and so the maximum available thrust for the UCAV-X was calculated with GASTURB in the same manner as before. 9321 lb of thrust are available at the selected flight condition, giving the UCAV-X at altitude twice the T/W ratio of the HARV at altitude. Significant data from the X-31 that was utilized in the UCAV-X model is listed in the table below.

Table 5. Summary of X-31 EFM Physical Data

X-31 EFM (NASA / MBB experimental aircraft)			
length:	43 ft	*reference altitude:*	20,000 ft
wing span:	22.8 ft	*reference velocity:*	518 fps
weight:	15,600 lb		M = 0.5
max AOA:	90º	*T/W (at altitude):*	0.60
longitudinal controls:	canard, LEF’s, TEF’s, speedbrake, pitch TV		
lateral-directional “ :	ailerons, rudder, yaw TV		

3. Aerodynamic Characteristics

The reference altitude and velocity of 20,000 ft and Mach 0.5 were chosen in part because they were close to the flight condition of the HARV, and in part because they represent a reasonable flight condition for dogfighting, the X-31’s raison d’être. (Also, Mach 0.5 was one of the velocities used in the aerodynamic datasets.) As already mentioned, the S&C derivatives were exhaustively listed in the two aerodynamic datasets for AOA values ranging all the way from -20º to 90º. Due to the range of the data, no extrapolation was needed for the table look-up file *SCPlots3.m*. Also, due to the density of the data, even the interpolation could have been done away with by rounding AOA at any point during a maneuver to the nearest even degree.

However, even though the X-31 data is very conducive to being used in a table look-up file, the table look-up simply slows down run times too much. So, as with the

HARV, the X-31 S&C derivatives were curve-fitted with 4th and 5th order functions of AOA. The two versions of *LoadAircraft3.m* (table look-up and curve-fit) are in exactly the same format as the load files for the HARV. For this reason, and because some of the X-31 data is proprietary, the UCAV-X load files are not included in Appendix A.

E. EXPECTED RESULTS

In general terms, the optimal maneuvers performed by the HARV and the UCAV-X should be similar in appearance, as they have comparable ability levels. The optimal maneuvers performed by the Navion, on the other hand, should be less dynamic and more similar in appearance to conventional air combat maneuvers like the “Immelman” or “wingover.” Also, for a given maneuver, the UCAV-X and HARV should complete the maneuver in less time than the Navion. (However, the Navion will have two unrealistic advantages over the other two aircraft models: first, it has no control surface deflection rates, as discussed on page 42. Second, it has constant S&C derivative values, so its control power and stability at max AOA will be somewhat better than they would in reality.)

One reason for the increased performance of the HARV and UCAV-X of course, is TV. TV allows for controlled flight in the post-stall region, effectively increasing the flyable AOA range (or flight envelop) for both these aircraft. So while the Navion only has an AOA range of 30º, the HARV has a range of 70º, and the UCAV-X has a range of 110º. These expanded flight envelops should correspond to smaller turn volumes, and hence better maneuverability. When TV is used in conjunction with either aircraft’s conventional control surfaces, more control power is available in the conventional flight regime, which increases agility. (The aircrafts’ moments of inertia are also related to agility: since the UCAV-X has moments that are each approximately an order of magnitude less than those of the HARV, it will likely be the more agile of those two. While the Navion has the smallest moments, this is offset by its extremely limited control power; i.e., only three control surfaces, no TV, no fly-by-wire system, etc.)

Another reason for the high performance levels of the HARV and UCAV-X are their T/W ratios. At sea level in their test configuration, these two aircraft have ratios of

0.89 and 1.03, respectively. Even though these ratios drop significantly at the test altitudes in this study, both aircraft still have very good “available power” levels for accelerating and maneuvering in the vertical plane. When performing conventional maneuvers, this will mean that both aircraft can climb quickly…when performing unconventional maneuvers, this will mean that following a sharp velocity drop incurred by flying at high AOA, both aircraft can quickly accelerate back to their cruising speeds. Both of these capabilities will play an important role in performing time-optimal air combat maneuvers.

Finally, the maneuvers calculated in this study will hopefully demonstrate “unconventional” (or “dynamic”) tactics by utilizing extremely high AOA to reduce maneuvering time. These maneuvers will then be compared with 1) the “conventional”, or “standard,” maneuvers performed by the Navion, and 2) the time-optimal maneuvers calculated by previous studies. (This will all be done in Chapter VI.) The results of the HARV will be focused on in particular, as it represents an actual military aircraft (albeit modified), and has been modeled by other authors for studying time-optimal maneuvers.

THIS PAGE INTENTIONALLY LEFT BLANK

V. OPTIMIZING MANEUVERS WITH DIDO

A. INTRODUCTION TO DIDO

DIDO is a MATLAB-based application package for solving dynamic optimization problems. According to the authors, “The basic idea behind the solution method is to approximate integrals by quadratures and derivatives by finite dimensional differentiation matrices that arise in pseudospectral approximation methods.” [Ref. 29] Thankfully for the user (and in particular the author), no knowledge of quadratures or pseudospectral methods is required in order to use this program. In fact, very little knowledge of optimal control theory at all is required to use this program. The most important prerequisite for using DIDO is being able to write down a mathematical formulation of the problem. This formulation must include the following three elements:

1. A system of differential equations.
2. An independent variable to minimize.
3. Boundary conditions for the problem to be solved.

For the study of time-optimal aircraft maneuvers, the problem formulation was both written down and initially coded by Scott Josselyn (who is profusely thanked in the Acknowledgments for all of his work and assistance). In this problem formulation, the system of differential equations are the 6DOF equations of motion that were derived in Chapter III…the independent variable, of course, is time, and the boundary conditions are specified by the *LoadManeuver.m* file (to be described more completely in the next section). Once the problem had been expressed in a way that DIDO could be applied to it, all that remained was to code the problem in MATLAB.

A few general comments about the format of the MATLAB script files that are necessary to run DIDO: first, extensive use is made of structure arrays. One such array that has already been mentioned is the global variable **CRAFT** created by *LoadAircraft.m*. In this array, **CRAFT.m**, **CRAFT.Ix**, and **CRAFT.b** all correspond to different data about the aircraft being used in the problem (mass, roll moment of inertia and wing span, respectively). Likewise, DIDO stores the time history of all variables during an optimization run in a single structure array called **primal**. This array is composed of **primal.states**, **primal.controls**, **primal.nodes**, and **primal.statedots**, each of which are matrices of various sizes (depending on the number of state variables and controls). The length (number of columns) of each array is defined by the number of nodes selected for that particular optimization run; a higher number of nodes corresponds to a better defined solution due to the increased density of analysis points along the flight path. (For reference, the nodes used are the Legendre-Gauss-Lobatto (LGL) points.) The width (number of rows) of **primal.states** and **primal.statedots** are always the same, as they are both defined by the number of state variables (usually twelve in this study). The width of **primal.nodes** is one, and the width of **primal.controls** varies for each aircraft; for the Navion the width is four, corresponding to the controls of elevator, aileron, rudder and thrust.

Another important format consideration is the use of vectorization. In order to make the code run smoothly and quickly in the MATLAB environment, nearly all variables are both expressed as vectors (in arrays), and operated on as vectors (instead of using loops, for instance). The primary reason for this is that MATLAB is very efficient at manipulating vectors and matrices, but very slow when it comes to running loops. (*NOTE:* Improvements in MATLAB 6.5 have started to accelerate loop performance, but only for certain data types. In general, MathWorks still recommends vectorization.)

The downside to using vectors is that sometimes variables are not stored in an intuitive fashion. For example, there are four categories of boundary conditions (BC's) used in DIDO: state variable bounds, control bounds, path bounds and event bounds. Each of these BC's include upper and lower bounds; however, instead of the upper and lower bounds being stored together in a two column matrix, they are stored as separate vectors for each category in the structure arrays **bounds.lower** and **bounds.upper** (where the structure array **bounds** is very similar to **primal**). Future versions of DIDO, though, may not require vectorization, in which case there would be more latitude in naming variables.

Understanding DIDO's use of vectorization, nodes and structure arrays (in particular **primal** and **bounds**) will greatly help in reading through the description of the aircraft maneuver optimization code. More detailed information on the functioning of DIDO in general can be found in the authors' user manual [Ref. 29]. With the preceding concepts in mind, the next section will discuss the MATLAB script files necessary to run DIDO.

B. AIRCRAFT MANEUVER OPTIMIZATION CODE

1. Format

The aircraft maneuver optimization code (hereafter referred to simply as the "code"), is the collection of MATLAB script files necessary to interface with and run DIDO. It also includes the files used to analyze the results that DIDO produces. There are a total of fifteen files, which are listed by type in the table below. (Indented files are either functions or subfunctions of the file they are listed beneath.)

Table 6. DIDO Maneuver Optimization Files

Program	**Data Files**
Main.m	*LoadUnits.m*
	LoadAircraft.m
	-- *SCTables.m*
	LoadManeuver.m
DIDO Functions	**Analysis Scripts**
AircraftCostFunction.m	*Analysis.m*
AircraftDynamicsEquations.m	*Propagator.m*
-- *EOM.m*	*Validation.m*
AircraftEventConditions.m	*FlyAircraft.m*
AircraftPathConstraints.m	-- *DrawAircraft.m*

A few comments about the different categories of files before describing the files themselves: the program file includes a basic user interface and runs the other files and functions necessary to optimize a maneuver. It calls DIDO and also saves the output of the optimization. The three load files, when run by *Main*.m, put the appropriate data on the aircraft and maneuver being studied into MATLAB's workspace. The four DIDO functions are MATLAB functions used directly by DIDO, and therefore follow the format prescribed by DIDO's authors. (The subfunction *EOM.m* was added due to the complexity of the dynamics equations used in the aircraft code, in order to keep the function *AircraftDynamicsEquations.m* clean and simple.) Lastly, the analysis files are used separately from *Main.m* to examine the results of an optimization run.

2. File Descriptions

Main.m starts out by loading the necessary data for the optimization run into the MATLAB workspace by running the three load files. It then defines the boundary event times, which the authors of DIDO call "knots." Recalling that nodes are analysis points along the flight path, knots are the simply the first and last nodes, and hence are the analysis points at the beginning and end of the maneuver. They are significant because there is always a higher density of nodes near these points than in the middle of the maneuver. *Main.m* also establishes four sets of BC's that define the "solution space" of the optimization: state variable bounds, control bounds, path bounds and event bounds. The event bounds define the maneuver to be performed, and they are coded into *LoadManeuver.m*. The path bounds are optional, but can be used to constrain a variable throughout a maneuver (load factor is a good example). Control bounds define the minimum and maximum deflections of the selected aircraft's control surfaces, and they are coded into *LoadAircraft.m*. State variable bounds minimize the range of values that DIDO will search through during the optimization routine; they also preclude DIDO from using a physically unrealistic value (like a sideslip angle of 45°) in a solution. They are listed in the table on the following page for conciseness. (*NOTE:* AOA bounds vary by aircraft; those listed in the table are for the Navion. Also, the normalized [and dimensionless] values of the BC's will be described in *LoadUnits.m*.).

Table 7. State Variable Boundary Conditions

State Variable	*Min, Max*	*Normalized*
X	-50000 ft, 50000 ft	-25, 25
Y	-50000 ft, 50000 ft	-25, 25
H	0 ft, 30000 ft	0, 15
V	10 fps, 2000 fps	0.05, 10
α	0°, 20°	0, 0.349
β	-30°, 30°	-0.524, 0.524
p	-360°/sec, 360°/sec	-31.42, 31.42
q	-360°/sec, 360°/sec	-31.42, 31.42
r	-360°/sec, 360°/sec	-31.42, 31.42
ϕ	-180°, 180°	-3.142, 3.142
θ	-175°, 175°	-3.054, 3.054
ψ	-180°, 180°	-3.142, 3.142

The user interface portion of *Main.m* asks the user if they want to "bootstrap" from a previous solution, and how many nodes they want to use. The more nodes that are used, the more refined (and in general, the more accurate) the solution will be. However, there is a significant penalty in run time that is incurred by using more nodes. (Bootstrapping is a way of minimizing that penalty, and will be discussed in detail in a subsequent section.) Following those inputs from the user, *Main.m* then calls DIDO to run the optimization. The run is timed, and the results of the run are saved in the file o*utput.mat*.

LoadUnits.m is a very short script file whose purpose is to normalize the code's state variables so that they are on the same, or nearly the same, order of magnitude. The three dimensions of primary interest are distance, velocity and time, and they are normalized by 2000 ft, 200 fps and 5 sec. For example, in a given maneuver the Navion travels 6000 ft downrange and turns 90° in a maneuver lasting 20 sec. When the maneuver is being optimized, though, those three values become 3 units of distance, 1.57 units of angle (radians), and 4 units of time. The fact that all three values are the same order of magnitude means that there will be no numerical instability during the optimization caused by dealing with decimals and numbers in the tens of thousands.

LoadAircraft.m creates the global variable **CRAFT**, which stores all of the necessary data about the aircraft being studied: moments of inertia, wing dimensions, S&C derivatives, control limits, AOA limits, etc. (For a complete list of data, look at one of the *LoadAircraft.m* files included in Appendix A.) The S&C derivatives will be in one of three formats: analytic (constant values), table look-up, or curve-fitted. With analytic derivatives, as in the case of the Navion, the values are entered directly into the *LoadAircraft.m* file. For table look-up derivatives, the function *SCTables.m* is used. The function contains rows of S&C derivatives at different AOA values, and when the function is called by *LoadAircraft.m*, it gives values for all S&C derivatives based on the current AOA. For AOA values between those loaded into *SCTables.m*, the function interpolates the appropriate S&C derivatives using MATLAB's cubic curve-fit. For AOA values outside the range of those loaded into *SCTables.m*, the function extrapolates the appropriate S&C derivatives, again using a cubic curve-fit. (More detailed information on the table look-up functions for the HARV and UCAV-X can be found in the appropriate "Aerodynamic Characteristics" subsections of Chapter IV.) Lastly, for curve-fitted derivatives, the 4th or 5th order polynomial (of AOA) was written directly into the *LoadAircraft.m* file. So, as with the table look-up derivatives, whenever a new AOA is reached during a simulation, the curve-fitted derivatives are recalculated.

LoadManeuver.m defines the initial and final conditions of the maneuver in terms of the code's state variables. (The difference between these initial and final conditions is the delta expression that was first mentioned in Chapter II.) The file also creates what are called event (or maneuver) BC's. These BC's allow for a *range* of values to be an acceptable initial or final condition. For example, the Navion performs a maneuver with an initial altitude of 1000 ft, and a final altitude of 1500 ft. If the minimum and maximum bounds for the final altitude are 1200 ft and 1800 ft, respectively, then the code will attempt to finish the maneuver at 1500 ft, but will consider any final altitude between 1200 and 1800 ft to be acceptable. For initial conditions, equality bounds are always used; i.e. for the initial altitude of 1000 ft in the previous example, the minimum and maximum bounds would both be equal to 1000 ft, forcing the Navion to start the maneuver at exactly that altitude. In fact, the initial conditions are not only set exactly, but are the same for every maneuver. (The only difference being that the three test aircraft have different initial altitudes and velocities.) Examples of *LoadManeuver.m* can be seen in Appendix B.

AircraftCostFunction.m is one of the DIDO functions, and it simply identifies the independent variable (or cost) to be minimized during the optimization. For this study, that variable was time. Given a different problem formulation, the cost could be something else entirely, like fuel consumption. (*NOTE:* The format of the cost function was a Lagrange integral.)

AircraftDynamicsEquations.m is the DIDO function that dictates how the state variables change over time. The subfunction *EOM.m* does most of this work by analyzing the equations of motion, calculating the force and moment coefficients that go into the equations of motion, etc. The new **primal.statedots** variables that are calculated by *EOM.m* are sent to *AircraftDynamicsEquations.m* and in turn sent on to DIDO.

AircraftEventConditions.m describes the end points of the maneuver: the initial and final conditions. In that respect it is similar in format to *LoadManeuver.m*.

AircraftPathConstraints.m is the last of the four MATLAB functions, and is actually not required to run DIDO correctly. However, it was kept in the code because of its usefulness: unlike event bounds, which can only constrain initial and final values, variables entered as path constraints are limited throughout the maneuver. So, as already mentioned, load factor is one good example of a path constraint, in particular for the Navion and HARV. Another example of a path constraint would be setting Y=0, which would constrain a maneuver to the vertical plane. One final example are two path constraints that had to be added for the HARV: two of the control surfaces on the HARV (ailerons and horizontal stabilator) can be deflected symmetrically or asymmetrically for longitudinal or lateral control. So, each control surface was coded as two separate controls; to ensure that those control surfaces weren't being deflected symmetrically and asymmetrically at the same time, the following path constraints were created, ensuring that at least one deflection was equal to zero.

$$\delta s \cdot \delta ds = 0$$
$$\delta a \cdot \delta sa = 0$$

Analysis.m takes the results of a DIDO solution, makes them dimensional again, and plots the time history of those results on a series of figures. The first four figures show the four different groups of state variables: position, velocity, body rates and Euler angles. Figures five and six plot the controls and control rates, respectively, while figure seven plots the load factor. The last two plots display the Hamiltonian and the costates, both of which can be used to analyze the quality of the results obtained from DIDO's optimization. In addition to the plots, *Analysis.m* summarizes some of the important information about the optimization run: number of nodes used, total number of iterations run, time for DIDO to complete the optimization, and time to fly the maneuver.

Propagator.m is used in conjunction with *Analysis.m* to test the feasibility of the solution (a concept that will be discussed more in the last section of this chapter – "Optimality"). *Propagator.m* takes the control history from a DIDO solution and propagates it forward using MATLAB's Runge-Kutta based "ode45" algorithm. The algorithm uses the differential equations from *EOM.m*, the time interval defined by the DIDO solution's node interval, and creates a new time history of the state variables. These new state variables are then plotted against those calculated by DIDO's solution.

Validation.m uses the costates from a DIDO solution to further test the optimality of that solution. The costates (and other variables) are coded into a series of equations to calculate the partial derivatives of the Hamiltonian with respect to each control. These partial derivatives are then plotted to verify that their values are approximately zero. (To be discussed in more detail in the "Optimality" section.)

FlyAircraft.m is the last tool used to analyze a DIDO solution. Once a successful optimization run has been completed, and verified by *Analysis.m*, *Propagator.m* and *Validation.m*, *FlyAircraft.m* can be used to get a better visualization of the maneuver. Instead of the bird's-eye view and side view of the maneuver provided by *Analysis.m*, *FlyAircraft.m* creates a three-dimensional view of the trajectory, with a polygon-based aircraft pictured along the flight path to display orientation. The aircraft is created by the function *DrawAircraft.m*; this very clever script file defines half a dozen triangles in the aircraft's body axes, and then renders them as two-dimensional colored solids using MATLAB's "patch" command. Each "piece" of the aircraft is then moved to the current position along the aircraft's trajectory, and rotated to the current orientation using the Direction Cosine Matrix. The scale of the aircraft can also be modified so that it is large enough to easily distinguish orientation throughout a maneuver. Furthermore, the file can continue if requested and save a movie of the trajectory as an ".avi" file.

C. NUMERICAL CONSIDERATIONS

During the course of doing optimization runs on various maneuvers, a number of numerical issues sprang up. Most of these issues had the same root cause: namely, that the dynamics of aircraft motion (governed by twelve coupled, nonlinear, ordinary differential equations) are quite complex. As a contrast between the twelve EOM that are listed on page 37, the example problem from the authors' user manual [Ref. 29] (which minimizes fuel expended for a moon-landing) consists of the following *three* equations:

$$\begin{aligned} \dot{h} &= v \\ \dot{v} &= -g + T/m \\ \dot{m} &= -T/I_{sp}g \end{aligned}$$

Obviously as an example it was designed to be straight forward, but the aircraft problem is still more complex than the typical problem to which DIDO is applied. There are two principal reasons for this: first, as alluded to by mentioning the EOM, the aircraft problem is a 6DOF problem, allowing for translation along and rotation around three different axes. (The example moon-lander problem, on the other hand, is a 1DOF problem, and 3DOF problems are more common to DIDO.) Second, the aircraft is a physically "realistic" model with dimensions, moments of inertia, aerodynamic characteristics, etc. (As opposed to something simpler like a rudimentary "point-mass" model.) All that being said, the numerical issues that came up ranged from the debilitating (locking up or crashing DIDO) to the inconvenient (finding a time-optimal maneuver that, upon closer inspection, was physically impossible.)

This chapter will identify and organize the major numerical considerations that had to be addressed in order for the aircraft code to run well. Not all of these considerations were solved to the satisfaction of the author, but the progress that *was* made will be explained and verified by numerical results. (Potential improvements in any area will be addressed in the conclusion of this report.) In general, there were four major "areas of concern" that caused or were impacted by numerical issues. Taken chronologically, those areas were:

1. Problem formulation
2. DIDO setup
3. Run time
4. Fidelity of results

Problem formulation refers to the aircraft code in general, but more specifically to issues like setting BC's on a maneuver, and refining the aircraft models. The first two subsections, "State Variable Constraints" and "Control Rate Limits," deal with these issues. DIDO setup refers to making choices about options that DIDO gives when doing an optimization run; see "Node Selection" and "Bootstrapping." Run time is an issue that really deserves its own chapter – when first doing runs on the HARV and UCAV-X, for instance, run times soared to the point that fixing them became the one and only task for well over two months. The highlights of that work will be discussed in the last two subsections, "Table Look-Up" and "Curve-Fitting."

The last area, fidelity of results, is of course the most important. Once the code was tweaked, DIDO was initialized correctly, and run times were mitigated, results could actually be obtained and analyzed on a consistent basis. While the tools for analyzing DIDO runs have already been discussed (*Analysis.m*, *Propagator.m* & *FlyAircraft.m*), how to determine the quality of results *from* those tools has not. This will be the subject of the last section of this chapter, “Optimality.”

1. State Variable Constraints

For each optimization run performed with the aircraft code, the two most important user “inputs” are to define the maneuver and to describe the aircraft. This is done using the *LoadManeuver.m* and *LoadAircraft.m* files, and the issue that will be discussed in this section refers to the first of those files. (It will be helpful to refer to the sample file for the Wingover that is in Appendix B.)

The first several lines of the file describe the maneuver to be performed by expressing it as a “delta” of two state variable (as described at the beginning of Chapter II). For the Wingover, the two delta expressions are $\Delta Y = 1000\,ft$ and $\Delta\psi = 180^\circ$. This can also be seen by comparing the maneuver’s initial and final conditions – the only difference between the two is the cross range distance (0 ft versus 1000 ft) and heading angle (0° versus 180°). The problem area with *LoadManeuver.m* is in the second half of the file, which defines the upper and lower boundaries (or the constraints) for the initial and final values of the state variables. First of all, there are two different kinds of constraints: equality and “bounded” constraints. For a generic state variable “U”, with upper and lower boundaries U_{max} and U_{min}, the following two equations demonstrate equality and “bounded” constraints:

$$U_{\min} = U = U_{\max}$$
$$U_{\min} < U < U_{\max}$$

In *LoadManeuver.m*, the BC’s for the initial state variable values are all equality constraints: this specifies an exact starting condition for the Navion that can be used for all maneuvers. (Starting conditions for the other test aircraft will be defined in a similar fashion, but with their respective reference altitudes and velocities.)

The numerical trouble lies in the BC’s for the final state variable values. On the one hand, specifying the BC’s with equality constraints would definitely produce the intended maneuver; however, this also tends to “over-constrain” the problem for DIDO, and it's unable to find a solution perfect enough. On the other hand, "relaxing" the constraints on all the final state variables will give DIDO more possible solutions, but cause the following two undesirable results, as well: first, the desired final condition for the maneuver may not be achieved, due to overly lax constraints on the state variables associated with the maneuver's delta expression. Second: although unintuitive, DIDO may actually take *longer* to reach a solution, because the solution space for the problem has been enlarged by relaxing the constraints.

Balancing the two issues listed above became quite a project. With a dozen state variables, and a wide range of distances, angles and rates to set boundaries on, the number of possible BC settings was enormous. (For a listing of the minimum and maximum unconstrained values of all state variables, see Table 7 on page 59.) This being the case, it was necessary to find an organized but efficient method of determining what BC settings maximized result quality but minimized DIDO run time. Naturally the Navion, being the simplest aircraft model, was used for this experiment, as were are all four standard maneuvers. Each maneuver was studied in turn by dozens of DIDO runs – each with slightly different BC settings. The results were recorded in a log and analyzed to see if any trends appeared from constraining or relaxing different state variables by different amounts.

The numerical problem that arose out of setting these final condition state variable constraints was that even small changes had dramatic effects on both run time and result quality. Not only that, but in some cases constraining state variables *decreased* run times, and in other cases it *increased* run times; likewise, in some cases result quality *improved*, and in other cases result quality was *degraded*. (The same four scenarios occurred when relaxing state variables, as well.) For some examples, refer to the first page of Appendix C, which contains a few excerpts from the log "DIDO Run History." In the log, all state variables that are "constrained" are limited by equality constraints, and all those that are "unconstrained" are bounded above and below their final value by the amount listed. (To review the final values for the Wingover, refer to its *LoadManeuver.m* file in Appendix B.)

First of all, notice that going from run #1 to run #2 tightened the constraint on the final values of X and Y from ±500 ft to ±100 ft. The result was an increase in result quality ("fair" to "good") and a decrease in run time – a perfect combination. (Although not listed, further constraints on the final values of X and Y yielded no benefits. Neither did more relaxed constraints, like ±200 ft. Therefore ±100 ft was kept as the "optimal" constraint on the state variables X and Y.) Changing the constraint on H from ±100 ft to an equality constraint in run #3 didn't improve result quality, but it did significantly reduce run time.

In the next two runs, state variable BC's were relaxed instead of constrained. For run #4, the final values of AOA and sideslip were allowed to vary from zero by ±2°. As in the previous run, there was no change to result quality, but in this case run time *increased* by a significant amount. In the last example run, AOA and sideslip are reset to equality constraints, while the body rate variables are relaxed by ±2°/sec. Even though this slightly increased run time (by about six seconds), result quality increased from "good" to "great," which is a good trade. (Different BC's for the body rate variables were tested to see if their was a value that improved result quality *and* decreased run time. However, values of ±5°/sec and ±10°/sec failed to improve on run time or result quality, and so the BC of ±2°/sec was retained as the optimal constraint for the body rate state variables.)

These examples from "DIDO Run History" demonstrate several things. First, they demonstrate the methodology of "tuning" DIDO results by systematically modifying final state variable BC's. The important steps in that process are listed below:

1. Start with predominately equality constraints (in particular on variables that have a final value of zero) and a few relaxed bounded constraints.
2. Tighten bounded constraints first, and one at a time. Then relax equality constraints, either one at a time or in groups (i.e. body rates, Euler angles, etc.).

3. When a positive result is achieved (i.e. better result quality or faster run time) by modifying a variable's BC, perform several more runs with slightly different BC values to determine the "optimal" setting of that BC.
4. When a variable's optimal BC setting is determined, retain it and continue on to the next variable.

Also, while not obvious from the small selection of runs listed in Appendix C, priority was given to determine the best BC's for certain "critical" state variables over other "non-critical" state variables. The critical state variables were identified in Chapter II as being those that generally "defined" a maneuver (which would be the variables used in a maneuver's delta expression): position (X, Y and H), velocity (V) and orientation (principally ψ). These are important variables to set BC's on, because different BC's will likely change what the maneuver looks like.

Unfortunately, the examples from "DIDO Run History" also demonstrated that there was no obvious correlation between changing BC's and affecting result quality or DIDO run time. Not only could constraining and relaxing BC settings affect the results and run time for better *or* for worse, but slightly different BC values could produce wildly different results. In the end, it was determined by scores of trial runs that for each maneuver, there was a relatively unique set of BC's that elicited the best results. For this reason, it could be said that the relationship between BC settings and DIDO results is, at best, extremely nonlinear. Hence the necessity for a good system to fine tune DIDO results – as described above.

NOTE: Only the standard maneuvers performed by the Navion were fine tuned to the degree described in this section. Due to longer DIDO run times for the optimal maneuvers, and significantly longer run times for maneuvers flown by the HARV and UCAV-X, it was simply not possible to perform the dozens of runs necessary to fine tune each of those 15+ aircraft/maneuver combinations. However, as a minimum, the critical state variable BC's were analyzed for each maneuver.

2. Control Rate Limits

As stated in the previous section, the two most important user "inputs" to the aircraft code are to define the maneuver and to describe the aircraft. This section will deal with the latter of those two: the aircraft model. One of the important elements of that model, of course, is to define the aircraft's control surfaces, and to set the maximum positive and negative deflection angles. The Navion's control surfaces consist of an elevator, rudder and pair of ailerons. The deflection limits for those control surfaces (which can also be found in *LoadAircraft.m*) are as follows: the elevator can deflect 15° TED or TEU, the rudder can deflect 15° TEL or TER, and the ailerons can deflect for a combined 10° (in hindsight a rather small angle).

While this was a great model to test the aircraft code with, it became evident after analyzing a number of DIDO results that it would be necessary to improve the controls for the HARV and UCAV-X models. The problem was not with the control deflections, but with the impossibly high *frequency* of control deflections. For example, one of the outputs created by *Analysis.m* is a time history of the aircraft controls; for a large number of Navion results, this plot showed controls "switching" multiple times per second. While the switching (going from maximum to minimum deflection) was a good sign, the frequency was not. On a physical airplane, even fly-by-wire controls can only deflect at a rate of about 100°/sec…the Navion model, on the other hand, was demonstrating control rates in excess of 200°/sec. The two plots in the following figure show the control history *and* the control rate history for a Reversal (the latter plot was added to *Analysis.m* to study this particular issue):

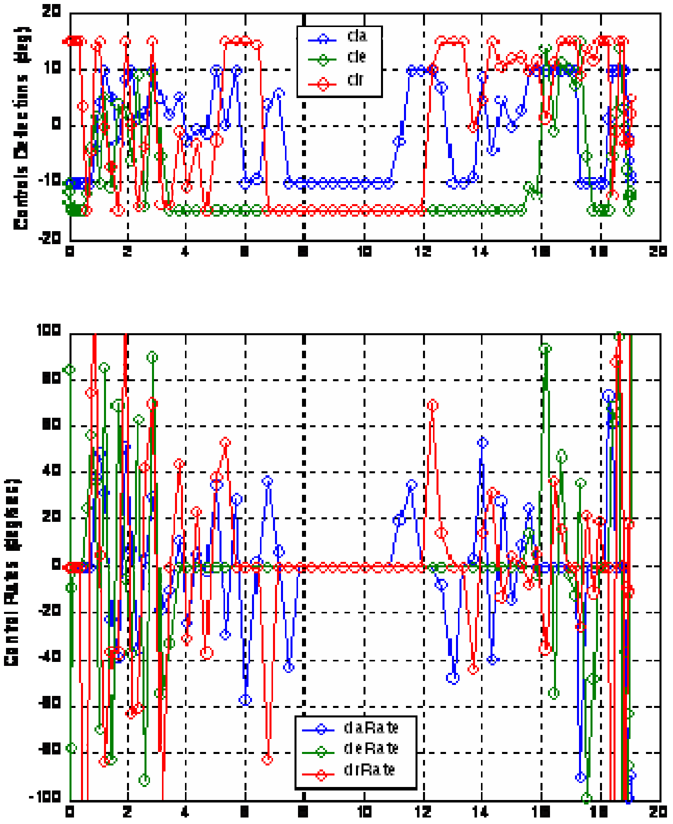


Figure 16. Control History (No Rate Limits)

(*NOTE:* Thrust is not included in the plots above because of the discrepancy in units and order of magnitude with the other aircraft controls. In *Analysis.m* it is actually plotted on a separate chart. Also, while there *is* a rate limit for how quickly thrust can be increased or decreased, it was neither researched nor included in the aircraft model for the following reason: 99% of all time-optimal maneuvers produced by DIDO used maximum thrust for the duration of the maneuver.)

Fortunately, the numerical issue of unrealistic control rates is one that was possible to address directly, unlike the previous issue of state variable constraints. All that was needed were deflection rate limits for every control, and since the Navion was used for this experiment, that meant only three rate limits. Two different options were studied for limiting control rates: the first used path constraints; the second assigned control *rates* to be the aircraft controls, while control *deflections* became state variables.

The path constraint method was very straight forward. Control rates, like load factor, were defined in the *AircraftPathConstraints.m* file with the use of previously defined variables (namely the control deflections and time). Then a limit was assigned to each control rate in the appropriate section of *Main.m*. The limits used for the Navion were based on those of the HARV, but reduced by about 10°/sec to somewhat account for the difference in technology levels between the two aircraft. That translated to the following rate limits: 90°/sec for the aileron, 30°/sec for the elevator and 70°/sec for the rudder. When the Navion flew the Reversal with those three path constraints in place, DIDO produced the following results:

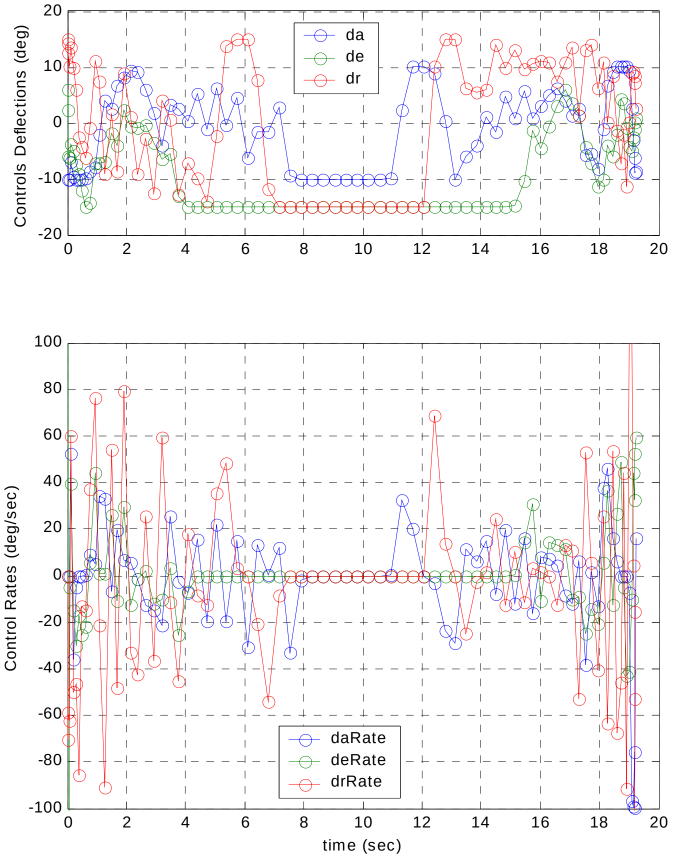


Figure 17. Control History (Path Constraint Rate Limits)

Obviously there is a marked improvement between the plots in Figure 17 and those in Figure 16. However, the control rates *still* exceed their max values, and there is still a lot more “switching” than would realistically occur in a maneuver. Therefore the second option was studied.

By assigning the control rates to be the aircraft's "controls," the limits were guaranteed to be followed, and much more realistic control behavior was expected. Unfortunately, by making the control deflections new state variables, almost every single file in the aircraft code had to be overhauled, and what resulted was a new set of files for a Navion with control rate limits. (A similar process was performed to develop a set of files for the HARV and for the UCAV-X.) When that task was finally completed, the Navion flew the Reversal again, with the following results:

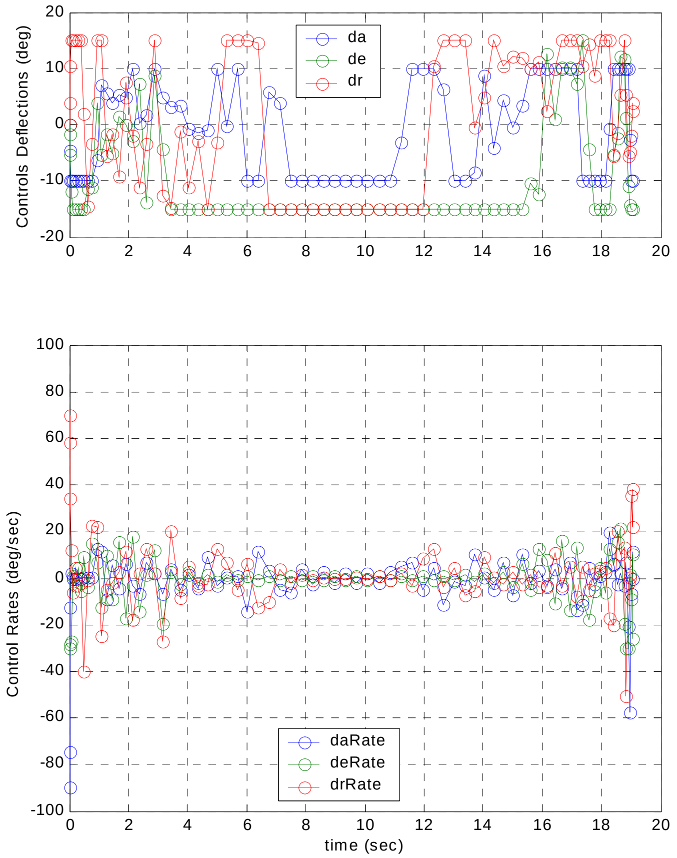


Figure 18. Control History (State Variable Rate Limits)

This time the control history and control rate history were right on track. Switches occurred frequently but not at an impossible rate, and the controls generally went from maximum positive to negative values, exhibiting the "bang-bang" pattern characteristic in optimal control theory. Based on those results, the "rate limit model" was adopted for the HARV and UCAV-X.

One last study was done on the Navion's rate limit model, and that was comparing run times using that model with run times using the standard model. Each maneuver was performed once by each model, and the DIDO run time and total number of iterations were recorded in "DIDO Run History" (see page two of Appendix C). The major points of interest from this study were the two trends noted in "DIDO Run History": one, that all four "standard" maneuvers increased in total iterations and run time when the rate limit model was used, and two, that three out of four "optimal" maneuvers decreased in total iterations and run time when the rate limit model was used. Unfortunately there's no obvious reason why this was the case, and neither was there a correlation between the change in the number of iterations and the change in run time (i.e., a greater decrease in iterations corresponding to a greater time savings). On top of that, one maneuver had an increased number of iterations when using the rate limit model, but had a *decreased* run time.

Despite all that, these results were taken as a good sign for the use of the rate limit model on the HARV and UCAV-X. For one, the model reduced run times more often (and by more time) than it increased them – the average time savings being 11%. Also, very importantly, the model's effect on the optimal maneuvers was that it always decreased run times, and almost always decreased iterations. Since studying those four families of maneuvers is the principal goal of this paper, anything that generally improves or expedites their analysis is a good thing. So in conclusion, the rate limit model (as demonstrated by the Navion) not only improves the fidelity of the aircraft being studied, but generally decreases DIDO run time.

3. Node Selection

One of the best things about DIDO is how few parameters need to be set in order to use the program. Of those that are needed though, choosing the correct number of nodes and a good initial guess (to be discussed in the next section) are both extremely important. The difficulty when choosing the number of nodes for a DIDO run is trying to balance run time with result quality. (In this case quality would be better described as "resolution," because the number of nodes defines the density of analysis points along the flight path, and therefore determines how detailed the results are.) This balancing act between run time and resolution was done in two steps. First, for each of the standard maneuvers (again performed by the Navion), an initial number of nodes was selected based on past experience running the code. Then, each maneuver was run several more times with 10 or 20 nodes more or less than the initial value.

The results of four runs for each maneuver were recorded in "DIDO Run History" (see page 3 of Appendix C). For each run the number of nodes, major iterations and total iterations, as well as run time, were recorded. The comments on the far right side of each entry were short-hand notes to describe how good the results looked. As it says in Appendix C, the best result was based on "a smooth flight path, propagated results that match DIDO's results, and a flat Hamiltonian." While this identified the node value for each maneuver that produced the best result quality (or resolution), there was still the issue of run time to be factored in.

As expected, the results in Appendix C show that in (almost) every case, increasing the number of nodes increases run time. However, it doesn't always increase result quality. Increasing the nodes for the Wingover from 60 to 70, for instance, added another two minutes onto the run time, but had no appreciable effect on the results. Going from 60 to 70 nodes for the Straight Climb had a similar time penalty, but actually decreased the result quality. For those two maneuvers, it was easy to pick 60 nodes as the optimum choice for both run time and result quality.

The node values that gave the best results for the Level Turn and Climbing Turn were 90 and 110, respectively. However, both maneuvers had prohibitively long run times at those node values. Looking at other options for the Level Turn that had "ok" results, 70 nodes was the obvious choice as it reduced the run time by 38% from the 90 node solution. Picking an alternate node value for the Climbing Turn was quite easy, because only the 90 node solution didn't have any problems with it. This solution also resulted in a significant 58% time savings from the 110 node solution, without making too great a sacrifice in result quality.

Obviously the method described in the preceding paragraphs is one of trial and error; however, if done correctly, the optimum node value can be found very efficiently.

The priority is to examine the results of a maneuver performed over a wide range of node values (perhaps 60 nodes at intervals of 10 nodes). Once the "best" solution is found, compare it to any solutions that were nearly as good to see if any significant time savings can be found. After that's done, and the best balance of run time and result quality is found, that optimum node value should be used for all future work to be done with that maneuver. (Keeping in mind that a higher node value can always be used at a later date, when analysis time is not a pressing issue and the best possible result quality is desired.)

Some final, general comments on nodes and node selection: for the aircraft code, 20 nodes is the absolute minimum resolution needed to get a rough "sketch" of a maneuver (this will be extremely important in the next section). Along those same lines, node values closer to 100 (as seen in Appendix C) are much more common. Lastly, 10 nodes was chosen to be the smallest "increment" of analysis for two reasons. First, with such large node values needed to achieve acceptable resolution, node deviations of one or two were not expected to make a significant difference. The second, and overriding reason for using 10 nodes, was that there was simply not the time to compare results for every maneuver at forty or sixty different node values. Hopefully in the future a more comprehensive study of node selection (as it pertains to the aircraft code) can be made. (See the section on improvements at the conclusion of this report.)

4. Bootstrapping

Bootstrapping is directly related to providing an initial guess of the solution to DIDO. This initial guess of what the time-optimal maneuver will be is created using the structural array **guess**, which is described as follows. The structure contains the following three elements: **guess.states**, **guess.controls**, and **guess.time**. The first two of these are matrices with rows corresponding to the number of state variables and aircraft controls (just like **primal.states** and **primal.controls**). The third is a vector that is very similar to **primal.nodes** – it stores the times during the "guessed" maneuver that correspond to the values in **guess.states** and **guess.controls**. Just as using more nodes for an actual DIDO run increases the resolution, using more values for the **guess** structure can have a positive impact on result quality and (especially) run time. This is where bootstrapping comes in.

The default guess for the aircraft code is a *straight line trajectory* from the aircraft's initial to final conditions. The two points in this trajectory are defined explicitly by the aircraft code's global variables **INIT** and **FINAL**. In terms of maneuver resolution, this would be like running DIDO with only two nodes. However, since the trajectory is feasible (if not practical), this is a very efficient way to generate a guess and start analyzing a maneuver. The difficulty with the aircraft code is that there is a huge disparity in resolution between a guess (two points) and the final time-optimal maneuver (roughly 100 points). Bootstrapping is used to bridge that gap.

Now that the **guess** structure has been explained, bootstrapping can be defined as "using a previous solution as a guess for a subsequent, higher resolution run." For instance, if 60 nodes is not enough to get a good solution for a maneuver, the maneuver can be run again with 80 nodes *and* with the 60 node solution as the initial guess. Since the 60 node solution is very close to what the 80 node solution will be, it will take fewer iterations and (most importantly) less run time for DIDO to arrive at that solution than if it had used the default two point guess. The difficult part (for this study) was determining what the optimum starting node value was; in other words, what node value saved the most time when its result was bootstrapped to a higher resolution run. A relatively short study was done on this topic, the results of which were saved in "DIDO Run History."

The Navion was again used as the test aircraft for this "bootstrapping time" study, and the Reversal was chosen as the test maneuver. The maneuver was run at 20 node intervals from 20 nodes to 100 nodes, as can be seen in the table below. The original run times are listed in the far left column, and the run times from bootstrapping with different values of nodes fill the rest of the table. The percentages listed in the table are a measure of how much faster the runs were completed by bootstrapping. For example, 20 nodes bootstrapped to 60 nodes takes 2.5 min + 2.7 min, whereas 60 nodes by itself takes 9.6 min. So the "bootstrapping time" is equal to 5.2/9.6 or 54% of the "original time." The time-improvement ratio, or TIR (a shorthand term for "how much faster" that will be used in the next two sections) is the inverse of that (9.6/5.2), or 1.85.

Table 8. Effect of Bootstrapping on Run Time

		Bootstrapped # of Nodes					
		0	*20*	*40*	*60*	*80*	*100*
Original # of Nodes	*20*	2.5		2.9 60%	2.7 54%	9.5 47%	16.5 15%
	40	9.0	0.5 381%		2.8 123%	4.7 54%	8.8 14%
	60	9.6	0.8 416%	1.8 127%		119.0 508%	79.3 72%
	80	25.3	0.6 1036%	2.6 310%	3.7 302%		12.9 31%
	100	123.0	0.5 4941%	2.8 1398%	1.7 1299%	3.6 500%	

There were three important facts gleaned out of this experiment. First, never, never bootstrap to a lower node value. This was actually known *before* the experiment, which is why the definition for bootstrapping included the two key words "higher resolution." However, it was included as part of the experiment for completeness. Second: a greater difference between the original node value and the bootstrapped node value results in a greater time savings. For instance, bootstrapping the 20 node solution to 60 nodes takes 54% of the original run time, while bootstrapping it to 80 nodes takes only 47% of the original time, and bootstrapping all the way to 100 nodes takes a meager 15% of the original time. Thirdly, and lastly, smaller original node values like 20 or 40 show the best time improvement (at least for the range of nodes tested in this experiment).

As mentioned in the last section, 20 nodes is the absolute minimum resolution needed to get a rough "sketch" of a maneuver. Based on this section's experiment, it is also the optimum starting node value for the aircraft code: for final node values ranging from 60 to 100, bootstrapping from 20 nodes reduced DIDO run times by an average

56%. However, it is interesting to note that as the final node values increase, 40 nodes appears to become a better starting value. In point of fact: bootstrapping from 20 nodes instead of 40 nodes saves 6.6 min when the final value is 60 nodes, but only 1.7 min when the final value is 80 nodes, and it takes 1.2 min *longer* when the final value is 100 nodes. Expanding this experiment to 120 nodes might show an even better improvement from using 40 nodes, and going beyond that to 180 or 200 nodes might show that 60 or 80 nodes would be a better starting node value at those levels. As with the previous section, hopefully a more comprehensive study of node selection can be made in the future. In the meantime, bootstrapping all results from an initial run of 20 nodes was an excellent way to save a lot of run time.

5. Data Table Look-Up

Just as a reminder, the function *SCTables.m* was added to the aircraft aerodynamic models in order to update the S&C derivatives during a maneuver. A very thorough description of how the tables work can be found in Chapter IV’s section on the HARV under “Aerodynamic Characteristics” (on page 47), and under the same heading in Chapter IV’s section on the UCAV-X (on page 51). (A summary of the function can also be found in the “File Descriptions” subsection earlier in this chapter on page 60.) Since the format and purpose of the table look-up functions have already been explained, this section will discuss the following topics: developing and improving the tables, creating S&C derivative plots from the tables, and comparing run times between different aircraft models and tables. These run time comparisons are the most important part of this section, as they were critical in determining the need to make further improvements to the S&C derivative models of the HARV and UCAV-X.

The Navion, with its constant S&C derivative values, had no need for a look-up table, but one was made for it in order to test the theory and format of *SCTables.m*. The Navion’s constant values were entered at 10° AOA increments, with MATLAB’s cubic “interp1” and “extrap” commands used to calculate the S&C derivatives at AOA values between those data points. Also, because AOA is a state variable, and therefore not a scalar but a vector, *SCTables.m* had to analyze a set of S&C derivatives for every value of AOA. This was originally done using loops, which turned out to be very inefficient due to MATLAB's current inability to quickly manipulate them. (See the note on page 56.) Two improvements were then tested to reduce the table's run time: the first was preallocating a matrix to store the S&C derivative values (a good MATLAB habit), and the second was replacing the table's loops with vectors. These improvements were tested separately and together, and compared against the original version of *SCTables.m* for different numbers of nodes and iterations. Results are shown and discussed below:

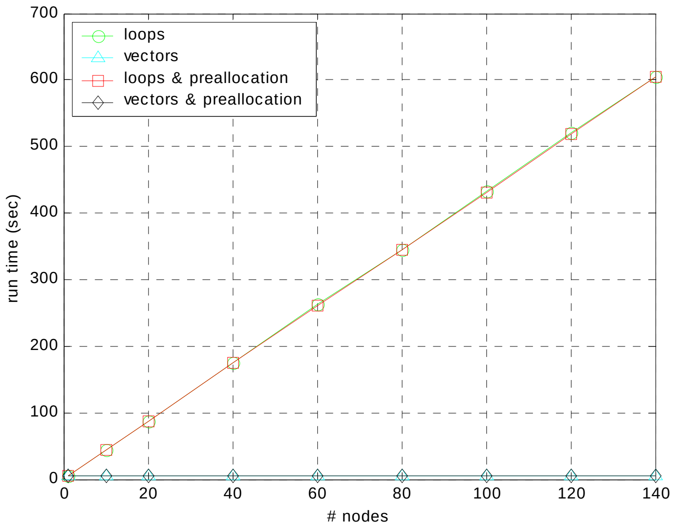


Figure 19. Table Look-Up Run Time vs. Number of Nodes

The plot above shows how the four different versions of *SCTables.m* were affected by different node values (for a constant 365 iterations): both versions of the function that used vectors had practically the same run time with 1 node as with 140 nodes (4.7 sec versus 5.9 sec). On the other hand, the two versions that used loops had run times that increased dramatically as the number of nodes increased (going from 4.7 sec to 604 sec). Also, looking more closely at the run times of the versions that used preallocation, it became apparent that they were consistently a fraction of a second faster than their conventional loop and vector counterparts. In conclusion, the resulting TIR between the original and improved version of *SCTables.m* increased linearly from 1.0 to 104; for the common node values used with the aircraft code, the TIR varied between 50 and 80 (a huge time savings).

The next plot shows how the four different versions of *SCTables.m* were affected by different numbers of iterations (this time for a constant 7 nodes). Note that the iterations in this plot are not the same as the iterations counted by DIDO. For instance, one iteration of DIDO could access the *SCTables.m* function twenty times; the iterations in this plot represent that second number. As with the previous plot, the two versions that used loops had nearly identical run times, as did the two versions that used vectors. Also, the loop versions again had noticeably longer run times. However, *unlike* the previous plot, the TIR between the original and improved version of *SCTables.m* stayed constant as iterations increased, which is pictured very well in the logarithmic plot. Also pictured well in the plot is the fact that run times for all versions increased proportionally with the increased number of iterations: the original version took 8.4 sec for 100 iterations, 84 sec for 1000 iterations and 852 sec for 10,000 iterations; likewise, the improved version took 1.3 sec, 13 sec and 130 sec for those same number of iterations. In conclusion, the resulting TIR of 6.5 was solely a function of choosing to use 7 nodes, and high numbers of iterations are going to have an inescapable, significant impact on run times.

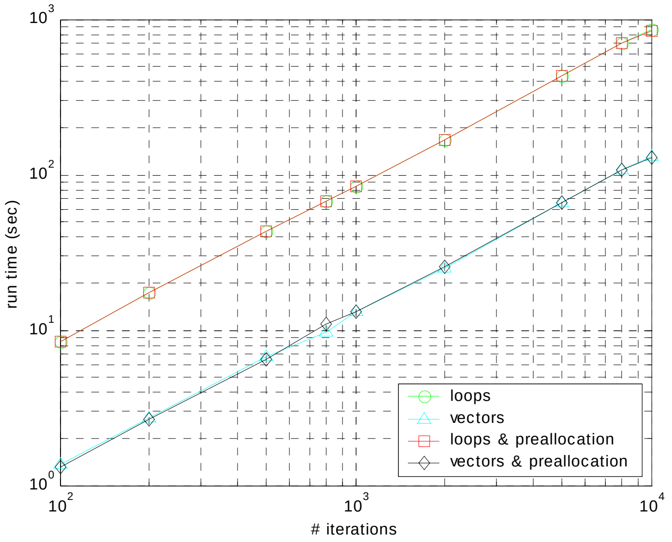


Figure 20. Table Look-Up Run Time vs. Number of Iterations

With the format of *SCTables.m* decided upon, the HARV and UCAV-X versions of the function were modified to include the same improvements. The functions *SCPlots2.m* and *SCPlots3.m* were then coded to print out plots of all the S&C derivatives for the HARV and UCAV-X, respectively. These plots were used in the case of the HARV to determine when zero AOA corrections were needed, to compare the interpolated values to the Iliff and Wang fairings [Ref. 19-20], and to compare the relative strength of each derivative on the six force and moment coefficients. They are included in Appendix D for reference.

a. Analytic vs. Table Look-Up Navion Models

This first comparison was recorded in "DIDO Run History," and compares the run times of actual maneuvers as flown by the Navion with three different aerodynamic models: the original, analytic model, the loop-based table look-up model, and the vector-based table look-up model. (These results are listed in the table below.) Two important ratios were developed from this comparison study: one was the ratio of run times with the loop-based model to run times with the vector-based model; this ratio (for each maneuver) was compared to the predicted TIR derived from Figure 19. Another important ratio was the run times with the vector-based model over the run times with the analytic model; these values for each maneuver gave an impression of how much longer it takes DIDO to converge on a solution when using a table look-up model.

Table 9. Analytic vs. Table Look-Up Run Times (Navion)

Maneuver	*1. Analytic*	*2. Table Look-up*	*Ratio (#2 / #1)*	*3. Table w/ Vectors*	*Ratio (#3 / #1)*	*Ratio (#2 / #3)*
Straight Climb (60N, 2531 iter.)	4.826 min	564.3 min	116.9	16.56 min	3.43	34.1
Wingover (60N, 4375 iter.)	4.586 min	415.8 min	90.7	13.22 min	2.88	31.5
Level Turn (80N, 2450 iter.)	21.89 min	2185 min	99.9	57.77 min	2.65	37.8
Climbing Turn (90N, 5126 iter.)	14.86 min	1250 min	84.1	32.52 min	2.19	38.4

As can be seen, the first ratio (Time 2/3) ranged from 31.5 for the 60 node Wingover to 38.4 for the 90 node Climbing Turn. The TIR's predicted for those particular node values were 49 and 71; however, it was not surprising that they were somewhat less when used in the actual aircraft code. The second ratio (Time 3/1) averaged out to 2.8, implying that it takes DIDO approximately three times as long to converge on a solution when using a table look-up model.

b. Navion vs. HARV vs. UCAV-X Table Look-Up Models

This second run time comparison was for all three test aircraft. The goal of the experiment was to determine if it took longer to access the look-up tables for the HARV and UCAV-X than it did to access the Navion's look-up table, and if so, by how much. Run times to access all three look-up tables are listed in the table below for a wide range of nodes and iterations.

Table 10. Navion vs. HARV vs. UCAV-X Table Look-Up Run Times

# Nodes	*# Iterations*	*1. Navion*	*2. HARV*	*Ratio (#2 / #1)*	*3. UCAV-X*	*Ratio (#3 / #1)*
20	**500**	6.33 sec	15.11 sec	2.39	13.06 sec	2.06
20	**1,000**	13.09 sec	30.14 sec	2.30	25.99 sec	1.99
20	**2,000**	25.19 sec	59.80 sec	2.37	51.74 sec	2.05
20	**4,000**	50.67 sec	120.3 sec	2.38	104.0 sec	2.05
20	**8,000**	101.7 sec	141.6 sec	2.38	208.6 sec	2.05
20	**10,000**	204.2 sec	482.8 sec	2.36	417.2 sec	2.04
60	**1,000**	14.47 sec	32.98 sec	2.28	28.64 sec	1.98
60	**2,000**	28.68 sec	66.36 sec	2.31	57.31 sec	2.00
60	**4,000**	56.68 sec	132.1 sec	2.33	114.0 sec	2.01

The more complex look-up tables for the HARV and UCAV-X did, as expected, take longer to access than the Navion's look-up table. This is not surprising due to both the larger number of S&C derivatives for those models, and the fact that the derivatives are not all the same value (as they are for the Navion). It was also not surprising that the HARV table took longer to access than the UCAV-X table, because for one, the HARV data is a lot more sparse and therefore requires more interpolation. Also, the HARV data is provided from multiple sources at different ranges of AOA, complicating the look-up table. In conclusion, the HARV and UCAV-X look-up tables take approximately 2.3 and 2.0 times as long to access as the Navion's. Based on the results of this and the previous run time comparison, it was estimated that maneuvers flown by the HARV and UCAV-X (with look-up tables) would take about six times as long to converge as maneuvers flown by the Navion (without look-up tables).

In conclusion, look-up tables were a good method of accessing more complex S&C data. For the HARV and UCAV-X, vectors stored data points of S&C derivatives at different angles of attack, and interpolation was used to determine S&C derivative values between those data points. This not only guaranteed that certain angles of attack would yield exactly the correct S&C values, but for models with a high density of data points (like the UCAV-X), this guaranteed extremely accurate results throughout the AOA regime. Unfortunately, the table look-up models also incurred run time penalties. While the factor of six discussed in the previous paragraph would be acceptable, that number was based on certain conditions remaining constant that, for the HARV and UCAV-X, did not. The two most significant of those conditions were the number of state variables and the number of iterations. While the number of state variables for the Navion was twelve, the number for the HARV and UCAV-X (which both have eight control deflection states) was twenty. Also, for an unknown reason, the number of iterations for a typical maneuver with the HARV were five to ten *times* greater than for the Navion. These effects combined to make run times unacceptably high, and led for the need to "speed up" the aerodynamic model.

6. Curve-Fitting

Curve-fitting was the technique used to reduce run times for the HARV and UCAV-X to an acceptable level (i.e. an hour or less for a fully converged solution). The specific goal of curve-fitting was to eliminate the need for both the large data tables of the UCAV-X, and the all the interpolation required by the data tables of the HARV. This was done by replacing the function *SCTables.m* with simple polynomials for each S&C derivative. (See the description of this process in Chapter IV's section on the HARV under "Aerodynamic Characteristics" [on page 47], and under the same heading in Chapter IV's section on the UCAV-X [on page 51].) This section will discuss the results of curve-fitting the HARV and UCAV-X aerodynamic models, and compare the resulting S&C derivatives with those obtained from the table look-up models. As with the last section, though, the focus of this section will be on comparing run times between different aircraft aerodynamic models. The results of these comparisons will demonstrate whether or not curve-fitting adequately reduced run times for the HARV and UCAV-X.

Some initial testing was done to determine what order polynomials were needed to best reproduce the S&C derivatives. After experimenting with a few derivatives for the UCAV-X, it was obvious that higher order polynomials were better able to capture the sometimes irregular patterns of the derivatives. That being established, each S&C derivative for the HARV was curve-fitted with different order polynomials to determine the best fit before triggering a "badly conditioned" warning in MATLAB. (This resulted in 4th order polynomials for 24 derivatives, and 5th order polynomials for the remaining 19 derivatives.) The resulting curve-fitted plots are included in Appendix E for reference.

The plots are an excellent indication of the accuracy of curve-fitting, because the actual data points are included on the plots for comparison. Also, the accuracy of the HARV curve-fitting was further verified by comparing the plots to the fairings produced by Iliff and Wang. (The HARV curve-fitted plots can also be compared to the table look-up plots in Appendix D.) The UCAV-X had too many data points for them to all be captured by curve-fitting, but the trends matched. (Also, matching the exact values of the X-31 S&C derivatives with the UCAV-X curve-fittings was not a priority since the aircraft being modeled is a theoretical one.) One final note on the curve-fittings for both the HARV and UCAV-X: as mentioned earlier, even higher order polynomials were unable to imitate the "double-peak" exhibited by many of the S&C derivatives. In most cases the curve split the two peaks, as demonstrated in the figure below. Overall, though, the curve-fitted derivatives very accurately reproduced the S&C data for both the HARV and UCAV-X.

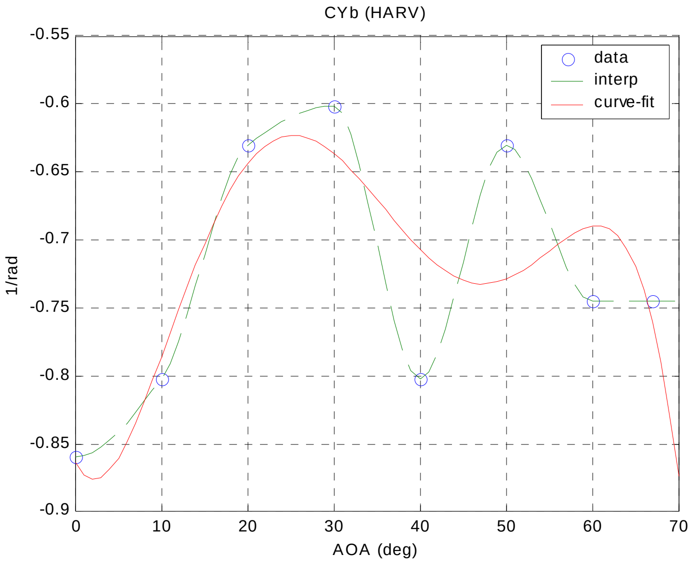


Figure 21. Example of Curve-Fitting a "Double Peak"

a. Table Look-Up vs. Curve-Fit HARV Models

Continuing from the previous section on the "Data Table Look-Up" models, this is actually the third run time comparison test. Like the first test, this one compares run times using different aerodynamic models of the same aircraft (in this case the HARV). The models being compared were the improved table look-up file from the previous section (*SCTables2.m*) and the curve-fitted S&C derivative equations (as coded directly into *LoadAircraft2c.m*). The times needed to run these models for a variety of nodes and iterations are listed in the table below. *NOTE:* The times listed in this and every other table in this section will be noticeably greater than run times listed in the previous section or "DIDO Run History." The reason is that these tests were done on a slower computer – a 1998 Pentium II (versus a 2002 Pentium IV). However, the results are still valid because the *ratios* are of primary importance, *not the run times*.

Table 11. Table Look-Up vs. Curve-Fit Run Times (HARV)

# Nodes	*# Iterations*	*1. Table Look-Up*	*2. Curve Fitting*	*Ratio (#2 / #1)*
1	**200**	36.97 sec	0.58 sec	63.75
10	**200**	38.83 sec	1.67 sec	23.21
20	**200**	39.52 sec	1.84 sec	21.51
40	**200**	41.19 sec	2.67 sec	15.44
60	**200**	41.85 sec	2.72 sec	15.40
100	**200**	43.64 sec	3.01 sec	14.48
150	**200**	53.65 sec	5.81 sec	9.24
20	**100**	19.25 sec	0.91 sec	21.23
20	**1,000**	1079 sec	49.31 sec	21.90
20	**2,000**	2185 sec	100.8 sec	21.68
20	**4,000**	4371 sec	201.8 sec	21.66
20	**10,000**	13573 sec	607.9 sec	22.33

Looking at the bottom half of the table, it's apparent that increased iterations proportionally increased run times for both the table look-up and curve-fit models. However, increasing the number of nodes from one to 150 only increased run time for the table look-up model by 45%, while run time for the curve-fit model went up by almost 1000%. (Most likely due to the time consuming calculations of large vectors in polynomial equations.) Unfortunately this meant that the greatest TIR's were at node values that were too small to be used in this study. However, even at the node values most commonly used in this study (40 to 100), the TIR of using curve-fitting over table look-up was still about 15 (and as high as 23 when using 20 nodes).

b. Table Look-Up vs. Curve-Fit UCAV-X Models

This fourth run time comparison repeated the procedure of the previous test for the UCAV-X. Run times for the table look-up model (*SCTables3.m*) and the curve-fit derivative equations (coded directly into *LoadAircraft3c*.m) are listed in the table below (continuing on to the following page) for a range of node values and iterations. (*NOTE:* Ideally, both this and the previous test would have compared not only the run times of the two different models, but the run times of actual DIDO maneuvers flown *using* the two models [as in the first run time comparison]. However, it would have taken far too much time to perform a comparison of actual maneuvers. The theoretical TIR's calculated in this and the previous table should closely match those that would be found with actual DIDO results.)

Table 12. Table Look-Up vs. Curve-Fit Run Times (UCAV-X)

# Nodes	*# Iterations*	*1. Table Look-Up*	*2. Curve Fitting*	*Ratio (#2 / #1)*
1	**200**	32.66 sec	0.64 sec	51.28
10	**200**	34.26 sec	1.63 sec	21.02
20	**200**	34.25 sec	1.89 sec	18.12
40	**200**	36.05 sec	2.67 sec	13.49
60	**200**	36.96 sec	2.65 sec	13.93
100	**200**	38.29 sec	3.05 sec	12.54
150	**200**	47.18 sec	6.67 sec	7.07
20	**100**	17.39 sec	0.98 sec	17.80
20	**1,000**	1173 sec	63.24 sec	18.54
20	**2,000**	1953 sec	109.2 sec	17.87

20	**4,000**	3877 sec	211.7 sec	18.32
20	**10,000**	9786 sec	535.9 sec	18.26

As with the previous test, the TIR from using curve-fitting over table look-up stayed constant with increasing iterations, but dropped quickly as the number of nodes increased. For the UCAV-X, the TIR stabilized out at 13 for node values of 40 to 100, and went up to 18 when using 20 nodes. As an aside, the increased TIR for both the HARV and UCAV-X at 20 nodes proved very beneficial since most maneuvers were run multiple times at 20 nodes in order to fine tune the solution before going to a higher resolution run.

c. Navion vs. HARV & UCAV-X Curve-Fit Models

This final run time comparison was for all three test aircraft. (The title is misleading because there was no curve-fitted model for the Navion – the title is referring to the original, analytic version of *LoadAircraft1.m*.) As stated at the beginning of this section, the goal of curve-fitting the S&C derivatives for the HARV and UCAV-X was to reduce their run times. The previous two sections have shown this was accomplished by calculating the TIR's of the curve-fit models over the table look-up models. This section will show just how successful curve-fitting was by comparing run times of the curve-fit models with run times of the Navion's original aerodynamic model (the simplest and fastest model used in this study).

Table 13. Navion vs. HARV & UCAV-X Curve-Fit Run Times

# Nodes	*# Iterations*	*1. Navion*	*2. HARV*	*Ratio (#2 / #1)*	*3. UCAV-X*	*Ratio (#3 / #1)*
1	**2,000**	1.50 sec	5.78 sec	3.85	5.91 sec	3.94
10	**2,000**	1.55 sec	18.03 sec	11.66	16.84 sec	10.89
20	**2,000**	1.45 sec	17.72 sec	12.14	18.12 sec	12.41

40	**2,000**	1.56 sec	24.58 sec	15.79	26.03 sec	16.72
60	**2,000**	1.90 sec	26.11 sec	13.76	28.13 sec	14.83
100	**2,000**	1.47 sec	30.04 sec	20.39	31.92 sec	21.67
150	**2,000**	2.31 sec	57.54 sec	24.94	62.58 sec	27.12
20	**1,000**	0.76 sec	8.87 sec	11.67	8.91 sec	11.72
20	**4,000**	2.86 sec	34.91 sec	12.19	36.49 sec	12.74
20	**10,000**	7.04 sec	88.46 sec	12.56	90.48 sec	12.85

At a node value of one, the HARV and UCAV-X curve-fit models performed extremely well compared to the Navion's analytic model – both took less than four times as long as the Navion model to run. Unfortunately, by 20 nodes that ratio had increased to 12, and for node values between 40 and 100 that ratio was 15 to 20. On the plus side, a ratio of 12 for a 20 node solution with curve-fitting is certainly much better than a ratio of 352 (12 x 21) or 216 (12 x 18) for a 20 node solution with the HARV or UCAV-X look-up tables.

In conclusion, the curve-fitted S&C derivatives for the HARV and UCAV-X significantly reduced run times for those aerodynamic models, without sacrificing accuracy. While access times for those aerodynamic models weren't reduced to the level of the original Navion model, they *were* dramatically improved – only taking 10 to 20 times as long to access versus 200 to 400 times as long to access the look-up tables. (There's an additional time savings to using the curve-fitting over the look-up tables that wasn't calculated in this section: the curve-fitted derivatives don't require calling the function *SCTables.m* because they are written directly into *LoadAircraft.m*. While a function call probably only takes a fraction of a second, that fraction multiplied by 1000 or 10,000 iterations can add up to a lot of time.) Since nearly all Navion maneuvers took between 5 and 15 minutes for a fully converged solution (when bootstrapping and running on a fast computer), it's possible (based on these results) that under similar conditions a fully converged solution for the HARV or UCAV-X could take the desired time of approximately one hour. (Actual run times will be discussed in the next chapter.)

D. OPTIMALITY

1. Definition

For the aircraft code, optimality is simply the minimum time for an aircraft to get from its initial to final condition, as determined by DIDO. DIDO arrives at this time by determining the "state-control function pair {x(˙), u(˙)}, design parameters p, and the 'clock times' τ_0 and τ_f that minimize the Bolza cost functional

$$J[x(\cdot),u(\cdot),\tau_0,\tau_f;p] = E(x(\tau_0),x(\tau_f),\tau_0,\tau_f;p) + \int_{\tau_0}^{\tau_f} F(x(\tau),u(\tau),\tau;p)d\tau \text{ .''}$$

[Ref. 29] However, even a maneuver that is returned by DIDO as "optimal" is still verified by several analysis tools in the aircraft code before being accepted as a genuine time-optimal maneuver. This section discusses the criteria that were used to determine whether or not a maneuver was time-optimal, as well as how different analysis tools were used to verify optimality.

2. Conditions for a Valid Maneuver

Six different criteria were used to test both the optimality and feasibility of maneuvers generated by DIDO. They are listed and discussed below in rough order of precedence (the first three relating to optimality and the next three to feasibility). For instance, if the DIDO output from a maneuver stated that the result was *not* "locally optimal," it would have been dropped right then without any further analysis, and tried again with different BC's. However, for various reasons, certain maneuvers that did *not* meet all six criteria were retained and included in this report; such maneuvers were simply annotated as "non-optimal" solutions. In conclusion, those maneuvers that pass the rigorous set of criteria described below are referred to in this report as "optimal solutions" with a great deal of confidence.

a. DIDO Output

Naturally, the first condition for optimality is that the output from a DIDO run states that the solution is optimal. The most common DIDO outputs (and some of the associated comments) are listed below:

1. "Locally Optimal" (However, check the following: Hamiltonian, dual variables, states and controls, and cost.)
2. "No Solution" (While a result was calculated, there are most likely singularities in the functions. Number of nodes could also be too small.)
3. "Infeasible" (Usually due to incorrect problem formulation or incorrect coding. Also check for over-specified BC's, redundant constraints, etc.)

The desired output, of course, for this and any other study, was the first one on the list – "locally optimal." This indicated that in the solution space defined by the user (an important qualification), the maneuver generated by DIDO was the optimal way to achieve the final flight conditions specified. On the other hand, a result labeled by DIDO as "infeasible" had to be reworked in order to get any kind of state and control history for the maneuver. Results between those two extremes, as previously mentioned, were generally discarded: the maneuver would be run again with different final state variables or different final BC's until a "locally optimal" result was achieved, and analysis could continue. (For more information on how, specifically, DIDO defines a maneuver as "locally optimal" versus "infeasible," etc., please refer to the author's user manual [Ref. 29].)

b. Hamiltonian

Optimal control theory states that the Hamiltonian should be constant for the type of maneuvers being analyzed in this study. Also, according to the original author of the aircraft code, "since this minimum time problem was formulated as a Lagrange cost (and thus the Hamiltonian is not an explicit function of time) the expected value of the Hamiltonian is zero." [Ref. 30] Fortunately, both of these conditions were easy to verify using *Analysis.m*. Take, for example, the following figure produced by *Analysis.m* for a Level Turn flown by the Navion:

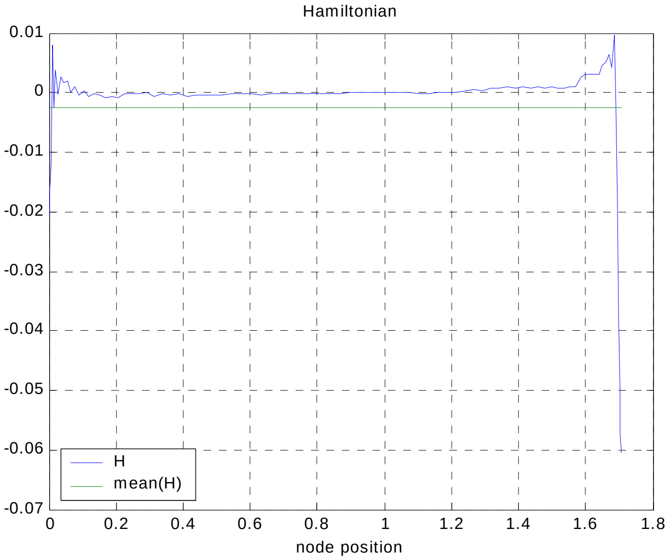


Figure 22. Sample Hamiltonian

It is immediately obvious by inspecting the time history (and mean value) of the Hamiltonian that the optimality criteria of a flat, near zero Hamiltonian is satisfied. The only inconsistency is in the spikes that occur at the endpoints of the maneuver; while no good explanation for the spikes can be offered, they are characteristic of almost every maneuver that was run with the aircraft code. Fortunately, in most cases the peak value of the spikes are of an order <<1. With the exception of the spikes, then, any other noticeably jagged appearance of the Hamiltonian was associated with non-optimality. (This was usually supported by several other criteria not being met, as well.)

c. Dual Variables

The dual variables (or costates) are a very useful set of values that DIDO stores the time history of in the array **dual.states**. While these variables have no physical meaning, they were used to test another criteria for optimization. That criteria states that the partial derivative of the Hamiltonian with respect to each of the controls must be zero. For the Navion, in other words,

$$\frac{\partial H}{\partial \delta_a} = \frac{\partial H}{\partial \delta_e} = \frac{\partial H}{\partial \delta_r} = \frac{\partial H}{\partial \delta_T} = 0 \,.$$

The partial derivatives of the Hamiltonian were developed implicitly by using aircraft data, state variables and dual variables. A sample printout of these partial derivatives, created by *Validation.m* for a Level Turn flown by the Navion, is included below:

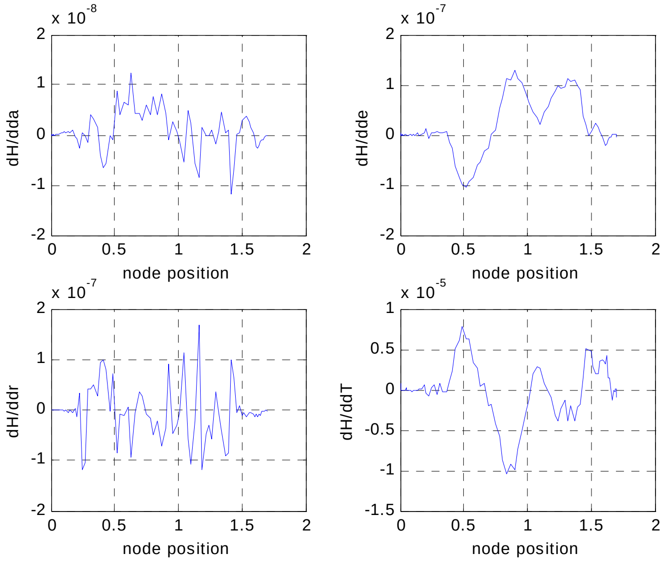


Figure 23. Sample ‘Validation.m’ Plot

Obviously this maneuver satisfied the “partial derivative” condition, as the largest value of the partial derivatives pictured above is on the order of 10^{-5}. For a maneuver that passed this criteria *and* the previous two (like the Level Turn), first order optimality is well established. The remaining three criteria will establish the feasibility of the maneuver.

d. Propagated Solution

Propagator.m is a great tool for testing whether a maneuver produced by DIDO can really be performed. In general terms, it takes the control history from a DIDO maneuver, “flies” the test aircraft with those control inputs, and stores the resulting maneuver. If that “propagated” maneuver matches the original DIDO maneuver, then the maneuver is feasible. (For a more detailed discussion of how *Propagator.m* works, refer back to the initial description of the file on page 62.) In order to verify that the maneuvers matched, the propagated results were plotted alongside the original results on the first four *Analysis.m* figures that correspond to the aircraft's position, velocity, Euler angle and body rate terms. The body rates were usually the most difficult to match, so that figure is pictured below as an example (again for a Level Turn flown by the Navion):

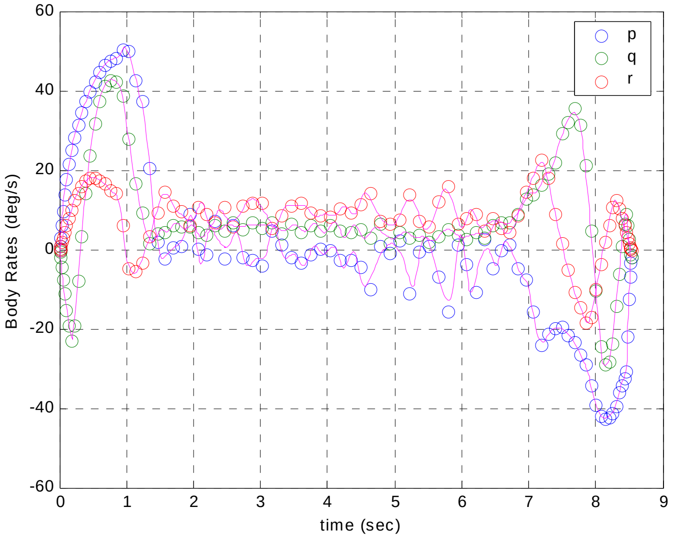


Figure 24. Sample 'Propagator.m' Plot

In the figure, the solid lines are the propagated values and the circles are the original values. As can be seen, the propagated values were an excellent match at the beginning and end of the maneuver, running right through the center of just about all the original data points. However, there were a few roll rate terms between 4.5 and 6 sec that weren't captured by the propagated results, and a few yaw rate terms between 1.5 and 3 sec. On the whole though, the propagated results did a very good job of matching the relatively high frequency body rates – they also perfectly matched the position, velocity and Euler angle terms. Based on all that, the first and most important criteria for the feasibility of this particular maneuver was considered to be met.

e. States & Controls

This criteria is much more subjective than the others that have been discussed. Essentially it is just a common sense check to make sure that the state variable history and control history are reasonable. This check, for instance, brought to light the need for control rate limits to be added to the aircraft models. This was also a good point to check for things like path constraints not being followed, BC's being violated, etc. While most maneuvers passed this criteria, occasionally one would come up that didn't behave correctly in some area, and the *LoadManeuver.m* file would have to be manipulated until the problem went away.

f. Result & Cost

As with the previous criteria, this one is also rather subjective. The purpose of it is to examine the DIDO solution for accuracy – to make sure that the maneuver finished at the correct position, speed, and orientation, and that it finished in a reasonable time. For instance, if an aircraft was supposed to travel 1000 ft cross range, ±200 ft, and it in fact traveled 1400 ft, then DIDO's result did not solve the specified maneuver. (The usual problem in such a situation was that another BC made it impossible to solve the particular BC that was violated.) Maneuver time was a more difficult matter to check, but if the test aircraft averaged about 200 fps during a maneuver, then it was possible to compare distance with time and make sure the ratio matched that average velocity. So, assuming a maneuver finished as it was designed to and in a reasonable time, this final condition for feasibility was met, and the maneuver could be declared "optimal."

As a conclusion to this section on optimality, the six different criteria that were used to test optimality and feasibility of maneuvers generated by DIDO are listed below. Maneuvers that passed these criteria will be referred to in the results of the study as optimal.

1. DIDO output of "locally optimal"
2. Flat, near zero Hamiltonian

3. Partial derivative of Hamiltonian with respect to controls equals zero
4. Propagated results match DIDO results
5. State variable and control histories are reasonable
6. Result and maneuver time are accurate

VI. RESULTS

A. NAVION

The main purpose of the Navion was to prove the validity of the aircraft code, which it did quite well (as these next few subsections will demonstrate). The Navion model was also a test bed for changes that had to be made to the aircraft code for it to be compatible with the HARV and UCAV-X. A number of these changes (control rate limits, bootstrapping, table look-up, etc.) were discussed in the previous chapter. The third and final goal for the Navion was to demonstrate conventional air combat maneuvers. As discussed at the end of Chapter II, optimal maneuvers performed by the Navion were expected to closely resemble typical air combat maneuvers like the Immelman, Split-S, and low yo-yo. After analyzing the eight Navion maneuvers included for this study, they will be compared to maneuvers like those just mentioned to verify that prediction.

NOTE: MATLAB plots summarizing the results of each maneuver are included in Appendix F, along with other pertinent data such as time to complete the maneuver, number of nodes used, etc. The maneuvers are listed in Appendix F in the order they will be discussed in this section; the appendix should be followed closely while reading this section as the figures were not reproduced here. (For reference, the order of the figures is as follows: 3D trajectory, position, velocity, body rates, Euler angles, control deflections, load factor, Hamiltonian and the derivative of the Hamiltonian with respect to controls.)

1. "Standard" Maneuvers

The first four Navion maneuvers were dubbed "standard" maneuvers because the delta expressions and BC's were designed to produce four very specific, simple, non-dynamic maneuvers. (As opposed to the normal procedure of simply picking a delta expression and seeing what the optimal maneuver looks like.) However, while the maneuvers are very conventional, they are all optimal, minimum-time maneuvers.

The first of these maneuvers, "Straight Climb," was a minimum-time climb from 1000 ft to 2000 ft, with the added constraint that the final velocity be equal to the initial velocity. (See Appendix F, page 183.) As expected, the maneuver was characterized by a rapid pitch-up, a straight climb, and a push-over at the end of the maneuver to regain airspeed. The pitch-up was accomplished by a max deflection of the elevator (15° TED), which resulted in a momentary pitch rate of 25°/sec. The elevator was then relaxed to maintain a 4° AOA climb. Since the max AOA for the Navion is 20°, the climb was probably thrust-limited. (Note the drop in airspeed from 176 fps to 132 fps.) The zero values for aileron and rudder deflection, roll and yaw, beta and lateral displacement (Y) all match the expected appearance of this maneuver. The initial spike in load factor is of course from the very rapid pitch-up, and the spike at the end of the maneuver is from pulling up out of the 0.8g push-over.

This maneuver was an excellent example of optimality. The DIDO result was “locally optimal,” the Hamiltonian was a perfect flat line at zero (except for the peaks of 0.75 and 0.21), and the partial derivatives of the Hamiltonian ranged from 10^{-10} to 10^{-5}. The feasibility was excellent as well: the propagated values on the position, velocity, body rate and Euler angle plots (marked by a solid line) almost perfectly matched the DIDO output. In addition, the two subjective criteria of reasonable state variable and control histories and accurate results were both met.

Some final thoughts: while the maneuver took just over two and half minutes to complete, the Navion actually hit the target altitude of 2000 ft 123 sec into the maneuver. Therefore if the final velocity was unconstrained, the maneuver time should be reduced by about 20%. Also, it could be verified that the climb was thrust-limited by arbitrarily increasing the Navion’s max thrust and re-running the maneuver with all other conditions the same. The result should be a higher AOA climb.

The second standard maneuver was a “Level Turn,” which was designed to change the Navion’s heading angle by 90° with little to no change in altitude during the maneuver. (See Appendix F, page 185.) In order to do this, the final displacement of the Navion was set to approximately 1000 ft in the X and Y directions, and the energy constraint was used ($\Delta H = 0$ and $\Delta V = 0$). While the controls chatter quite a bit, it’s obvious the maneuver starts with max right roll aileron, max rudder (15° TER) and max elevator (15° TEU). The ailerons deflect long enough to get about 60° angle of bank, then reset to zero until deflecting the opposite way at the end of the maneuver to return to wings level. For this particular maneuver, the Euler angle plot is a good picture of what's happening: a steady increase in heading angle from 0 to 90°, a bank angle of 60° for most of the maneuver, and a pitch angle that switches from positive to negative at the midpoint of the maneuver. (Due to the bank angle, elevator deflection during this maneuver mostly helps to change the Navion's heading.) The AOA peaks at 1.0 and 7.8 sec correspond to the load factor peaks, and are preceded by peaks in beta (probably due to starting a climb/descent while steeply banked).

The DIDO output for this maneuver was "locally optimal," and not only was the Hamiltonian flat and zero, but the peaks at the beginning and end of the maneuver only reached values of –0.02 and –0.06. Hamiltonian partials were all near zero, with the largest being 10^{-5}. Propagated values matched perfectly except in the case of the body rates, where a few points weren't captured due to the rapid oscillations of the DIDO-produced rates. In spite of that, the feasibility criteria was considered to be met. Lastly, while the controls switch from max to min values at an unrealistic rate, that was a known limitation of the Navion model, so this control history was considered reasonable.

Some final thoughts: the 80 ft climb at the midpoint of the Level Turn was not considered a significant change in altitude. However, if a perfectly level turn was desired, a path constraint could be entered to hold the altitude throughout the maneuver to 1000 ft. This would most likely result in a reduced bank angle, larger turn radius, and longer maneuver time. Also, as with the last maneuver, the Navion actually hit the target heading of 90° 7.6 sec into the 8.5 sec maneuver. In this case, the last 0.9 sec was spent rolling to wings level and pulling the nose up to zero AOA.

The next maneuver, a "Climbing Turn," was quite simply a combination of the previous two: climbing 1000 ft and turning 90°. Different values for X and Y displacement were experimented with to produce the smoothest trajectory. (See Appendix F, page 187.) As expected, the maneuver strongly resembled the previous two, but predominately the Straight Climb. In fact, the only noticeable differences between the two sets of results are the plot of ψ (which increases to 90° for the Climbing Turn) and the plots of Y and the ground track (which reflect traveling in the X *and* Y directions). The ailerons barely deflect (less than 2°), and the bank angle through the entire maneuver is less than 3°. Also, the climb AOA is 4° and velocity drops to 132 fps, just like the Straight Climb.

The maneuver definitely satisfies the criteria for optimality: "locally optimal" DIDO output, Hamiltonian flat and near zero (with small peaks on the ends), partial derivatives of the Hamiltonian also near zero. Propagated values are an excellent match, and the state variable histories are reasonable.

The last standard maneuver was a "Wingover," which was a heading reversal with an energy constraint and a lateral displacement of 1000 ft (to make the maneuver look like a classic Wingover…see Appendix F, page 189.) The maneuver turned out to be very smooth and symmetric, inscribing a neat semicircle over the ground. It was started with a quick bank and climb, and the progression of the maneuver can be seen very clearly from the control history: max aileron for right roll at the beginning of the maneuver, and max aileron for left roll at the end (to right the aircraft); there was also some left roll aileron applied between six and twelve seconds into the maneuver that started to reduce the bank angle. The rudder followed the same pattern of max TER at the beginning and max TEL at the end (to straighten the aircraft on its new heading). Elevator started at max TEU to climb the aircraft, but after that, elevator deflection was slightly unintuitive. Pitch angle decreased linearly from three to eleven seconds, but elevator deflection (TEU) increased and then decreased in a parabola during that same period. The reason for using the elevator in the middle of the maneuver was to continue to change heading while in a highly banked position with little rudder authority (note the lack of rudder use from seven to twelve seconds). Finally, max rudder at the end of the maneuver brought the airplane from its nose down position back to level flight.

The Wingover was "locally optimal," had a perfectly flat Hamiltonian except for a peak at the end of 0.1, and had near zero partial derivatives of the Hamiltonian. Propagated values matched very well, and the state and control histories were reasonable – even the controls only chattered slightly during the maneuver, and for the most part went from max to min.

2. Optimal Maneuvers

The maneuvers that will be discussed in this subsection include one example of each of the four families of maneuvers defined at the end of Chapter II: the reversal, turning, pointing and braking maneuvers. A description of the BC's chosen for each maneuver will be included in the discussion, as well as listed in the appropriate section of Appendix F. (Predominately the "energy" and "displacement" constraints were used.)

The first maneuver was a reversal, which was very similar to the Wingover except that no lateral displacement was included. This particular example of a reversal included both the energy constraint and the displacement constraint. With these BC's, the maneuver met all the same requirements as the Herbst Reversal, and was expected to look very similar (which it did…see Appendix F, page 191). The maneuver was much more dynamic than the Wingover was, even with the performance limitations of the Navion: AOA was maxed out at 20° for over half the maneuver, β values reached 30°, velocity dropped to 34 fps, and both pitch and bank angles reached values as high as 82°.

The maneuver started out with a slightly banked left turn and a short 12° AOA climb (note the 3+ g's between zero and two seconds into the maneuver). The Navion then actually leveled off before starting an incredibly steep climb and turn that totally reversed its direction. During this part of the maneuver the Navion ended up nearly standing on its tail, and then on its right wing (see the trajectory view in Appendix F). The Euler angles and the middle part of the control history are a good picture of what happened during this maneuver…note in particular the max deflections of the ailerons, rudder and elevator from 8-11 sec, 7-12 sec and 3-16 sec, respectively. (The elevator, for instance, remains deflected even after the Navion starts descending because it's acting like a rudder when the airplane's on its side. In the same way, the rudder helps pitch the nose down when deflected TER in this orientation.)

Optimality is not as obvious with this maneuver; certain criteria are easily met (like a "locally optimal" output from DIDO), but others (like the Hamiltonian and control history) are not. The plot of the Hamiltonian is very jagged, but the mean value (and peak values) are so small that it's safe to claim the maneuver is optimal. Supporting that claim are the values of the partial derivatives of the Hamiltonian, which range from 10^{-7}

to 10^{-3}. Propagated results for the reversal actually matched very well, with the exception of the plot for the body rates, which "chattered" quite a bit at the beginning of the maneuver and a little again at the end. This chatter matches the high frequency control deflections at the beginning and end of the maneuver, while the body rates and control deflections in the middle are well-behaved. All in all the results satisfactorily demonstrate feasibility.

Some final thoughts: while this maneuver actually took longer to complete than the Wingover (19 sec vs. 14 sec), the heading reversal was actually accomplished 13 sec into the maneuver; the remainder of the maneuver was spent diving to regain speed, and in the process returning to the initial altitude and position.

The second maneuver was the turning maneuver, which was a 90° heading change limited by the displacement and energy constraints. (See Appendix F, page 193.) The maneuver turned out to be a near-perfect figure eight when viewed from above, with each loop characterized by a banking climb, a turn, and then a descent to recover airspeed. Notice the two peaks in altitude, dips in airspeed and dips in load factor (<1 g) that occur at 6 and 17 sec into the maneuver, upon reaching the "top" of each loop. The peaks in load factor are another good indication of what's happening during the maneuver: they occur at approximately 1, 11 and 23 sec, when the Navion is climbing into its first and second loops and pulling up out of its final descent. Unfortunately, the body rates and control deflections chatter far too much to derive anything of use from them (in fact, the incredibly high frequency control deflections at the end of the turning maneuver are probably the cause for the rapid oscillations in load factor during the last 5 sec of the turn).

The optimality of this maneuver is extremely similar to that of the reversal: "locally optimal" with near zero (but jagged) Hamiltonian and near zero partial derivatives of the Hamiltonian. Feasibility is also similar, because the results are excellent but the body rates (and control deflections) chatter *a lot*. As with the reversal, the turning maneuver sufficiently met the criteria to be considered an optimal maneuver.

The next maneuver was the pointing maneuver, which for the Navion was defined as $\Delta\theta = 30^\circ$ and $\Delta\psi = 30^\circ$. (For the HARV and UCAV-X, more aggressive angles of

60° were used.) For this maneuver, the final values of the position variables were all unconstrained by ±1000 ft (effectively defining the allowable maneuvering space as a cube 2000 ft per edge). More restrictive BC's were experimented with, and some very interesting maneuvers were produced, but this set was chosen because it resulted in the kind of rapid reorientation that was the purpose of the pointing maneuver (i.e. to rapidly point weapons at an opponent during ACM without taking the time to gain a typical positional advantage). The result of this maneuver was a straight forward push-over followed by a rapid pull-up and turn to the right. (See Appendix F, page 195.) The Navion reached almost 200 fps in its dive, and flew at maximum positive and negative AOA on the way down and the way up (respectively). Perfectly matching the AOA was the load factor, which was a sustained –1 g for the push-over and a sustained 4.5 g for the pull-up. (This would be pretty hard on a pilot, in particular going from –1 g to 4.5 g in half a second.)

Due to the simple nature of the maneuver, the *Analysis.m* results were excellent. All three optimality criteria were met, and the state variable histories and propagated values were perfect matches all around – even the body rate history matched was easy to read. The control deflection history was a great example of the "bang-bang" appearance of optimal controls, showing very distinct switching from max to min. In conclusion, a very well-behaved optimal maneuver.

Some final thoughts: although unrelated to the maneuver itself, it's interesting to note that this four second maneuver took over 80,000 iterations and two hours (without bootstrapping) for DIDO to solve. (The pointing and braking maneuvers were similarly time-consuming for DIDO.)

The last of the Navion maneuvers was the braking maneuver, which was a minimum time maneuver to reduce velocity by 50%. Like the pointing maneuver, this one had the final values of all position variables unconstrained by ±1000 ft. The maneuver resulted in a simple 20° AOA, 4 g climb of a little over 300 ft (see Appendix F, page 197). AOA was maxed out (as expected), and was accompanied by a steady drop in airspeed. This maneuver is also somewhat unique in that it is the only maneuver where thrust was not set to max (it was actually set to min – probably to assist with bleeding off airspeed).

The braking maneuver easily met all the criteria for optimality and feasibility: near zero Hamiltonian and partial derivatives of the Hamiltonian (10^{-11}), and good state variable history that was perfectly matched by the propagated results. Also, just like the pointing maneuver, this maneuver displayed excellent control deflection characteristics.

3. Comparing to Current Air Combat Maneuvers

The reversal can be compared to several different maneuvers, and this section will start with the Wingover. Both maneuvers reverse the aircraft's heading and maintain the aircraft's energy level; the biggest difference between the two is that for an extra 5 sec, the reversal returns the aircraft to its starting position, which, depending on the air combat scenario, may be worthwhile. A maneuver more similar to the Navion's reversal is the "Hammerhead," which involves a steep climb to a vertical attitude, full rudder input to swing from nose up to nose down, and then diving to regain speed. The main difference between these two being that the Navion (possibly due to control power) doesn't reverse direction in the vertical, but inscribes a tight (300 ft diameter), high AOA, highly banked turn to reverse its direction. It's possible that a specific set of BC's could cause the Navion to perform a Hammerhead, at which point the maneuvering time between the two could be compared to see which maneuver was faster. Without that information, it's assumed from this report that the "locally optimal" reversal is the fastest way to return to the same point with the same speed and an opposite heading. Lastly, the Navion's reversal is, by definition, an example of a Herbst Maneuver (although it's not a post-stall maneuver since the Navion retains aerodynamic control authority).

The turning maneuver can be compared to the high yo-yo, or more accurately, two high yo-yo maneuvers back to back. The primary similarity between the two maneuvers is the use of the vertical plane to minimize turn radius. In the high yo-yo, using the vertical plane also enables the attacker to avoid overshooting his target, while retaining energy. The turning maneuver also retained energy, by diving at the end of its two turns to regain speed while returning to its initial altitude. Obviously certain nuances of the high yo-yo (like the attacker keeping his nose ahead of his target) can't be compared to the turning maneuver since it is a single airplane maneuver. However, the general appearance of a highly banked climbing turn followed by a dive is certainly common to both.

The pointing maneuver (partly due to its short duration) is not reminiscent of a particular maneuver. However, the pattern of diving and then pulling the aircraft's nose up and to the left or right is similar to that of the low yo-yo. (The difference being that in the low yo-yo, the attacker uses a *climb* after diving and turning to reduce the forward component of his velocity to get into a good attacking position. The pointing maneuver attempts to simply pull the nose of the aircraft into firing position while the attacker is still below the target.) Lastly, the braking maneuver looks like a tame version of the Cobra that doesn't enter the post-stall region. It's also a simple example of trading airspeed for altitude.

So in conclusion, many similarities could be drawn between the optimal maneuvers performed here by the Navion and conventional air combat maneuvers. There were also some interesting differences, such as the optimal reversal performed by the Navion versus a Wingover or Hammerhead. Each maneuver was informative though, and served as a good baseline for studying the HARV and UCAV-X maneuvers.

B. HARV

The HARV was the most important aircraft for this study for several reasons: first of all, it had an excellent set of aerodynamic data derived from extensive flight testing. Also, since the aircraft is essentially an F/A-18 Hornet with a TV unit installed, the maneuvers in this section are a good indication of the enhanced performance that *current* Navy fighters would be capable of if they could operate in the post-stall regime. Finally, since the HARV has been used in several other time-optimal studies, it makes it possible to compare DIDO and the aircraft code to other optimization tools. With that in mind, the goals of the HARV were as follows: to demonstrate dynamic post-stall maneuvers (which will be discussed in the first two subsections), to prove DIDO's capability to generate unconventional maneuvers, and to show that the HARV model, aircraft code and DIDO compare favorably to the tools used in other studies (to be discussed in the third and final subsection).

NOTE: As before, MATLAB plots summarizing the results of each maneuver are included in Appendix G. For this study, four maneuvers were included, and they are listed in Appendix G in the order they will be discussed in this section; the appendix should be followed closely while reading this section as the figures were not reproduced here. (The figures for the HARV maneuvers are the same as those included for the Navion maneuvers, with two exceptions: a plot of control deflection rates was added, and a plot of the derivative of the Hamiltonian with respect to controls was removed.)

1. Comments on DIDO and the HARV

Before presenting the HARV maneuvers, there are two shortcomings with the HARV and UCAV-X results that need to be explained. The first is the lack of propagated results: for both aircraft, trying to run *Propagator.m* on any DIDO maneuver produced the following warning in MATLAB:

```
Warning: Failure at t=4.242761e-001.  Unable to meet integration
tolerances without reducing the step size below the smallest value allowed
(1.507331e-015) at time t.
```

Although the exact time of the fault was not always the same, it was always at the very beginning of the maneuver; hence, little to no useful propagated data was available on the position, velocity, Euler angle and body rate plots. Since the aircraft code is fundamentally the same for all three test aircraft (see note), there was no obvious reason why the *Propagator.m* file did not work for the HARV and UCAV-X maneuvers. Even for *infeasible* Navion maneuvers, the file had no problem plotting the propagated results: the example on the following page was an early 40 node incarnation of the Straight Climb, and was probably the most blatant example from this entire study of propagated results that did not match DIDO's results.

(*NOTE:* The only structural differences between the Navion code and the HARV or UCAV-X code were the addition of state variables for each control surface, and using deflection rates as the actual controls. Both changes were tested on the Navion [see page 69], and the modified code functioned as well as the original. In particular, no error messages were generated when running *Propagator.m*.)

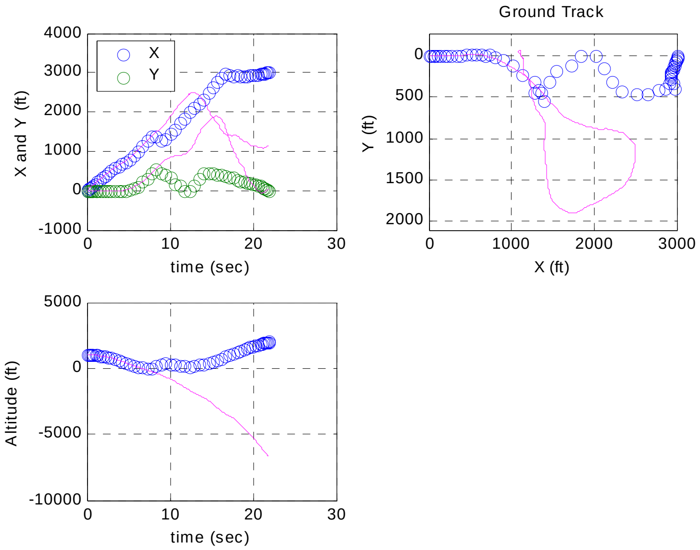


Figure 25. Example of Infeasible Propagated Results

While the above example is an extreme case of infeasibility, in a more typical example the propagated results would closely match DIDO's, and then diverge towards the end of the maneuver. This usually occurred when insufficient node values were used. (In some cases, however, the maneuver was still valid: i.e. if the propagated final state variables did not violate any BC's). In addition to correctly propagating infeasible results, the Navion code also correctly propagated non-optimal results. An excellent example of that was a non-optimal braking maneuver: the Hamiltonian for the maneuver spiked to values on the order of 10^9, but the propagated results matched DIDO's results *exactly*. Since *Propagator.m* worked for both non-optimal and infeasible Navion results, it was unlikely that it failed to work for the HARV and UCAV-X because their results were either non-optimal or infeasible. Also, since it failed to work for *any* HARV or UCAV-X maneuver, it was very unlikely that the problem was related to the quality of the results at all.

That being said, the only difference noted between the Navion code and the HARV or UCAV-X codes was software related… DIDO uses a third-party software called TOMLAB/SOL. According to the company's website (www.tomlab.biz), "the TOMLAB/SOL toolbox efficiently integrates the well-known solvers developed by the Stanford Systems Optimization Laboratory (SOL) with MATLAB and TOMLAB." Included in the toolbox is the sparse general nonlinear solver SNOPT, which is the primary component used by DIDO. For the HARV and UCAV-X maneuver optimization, a new version of TOMLAB/SOL was used (v4.2). (The older version of TOMLAB that was used for the Navion could no longer be used because the license had expired.) Unfortunately, there were some compatibility issues between DIDO and the new version of TOMLAB, including (for lack of other cause) failure of the *Propagator.m* file.

The solution to this problem was finding another way to verify that the control inputs for any given maneuver were realistic and would in fact produce the maneuver described by DIDO. The first part of that was easily taken care of by adding a control rate plot to all HARV and UCAV-X results. In fact, the control rate limits for the HARV and UCAV-X just about guaranteed realistic control inputs. They also dramatically improved the likelihood that the control inputs, if propagated forward, would produce a maneuver that matched DIDO's output. The reason being that every Navion maneuver that *didn't* have good propagated results also had very "choppy" controls. While the controls for the HARV and UCAV-X could still oscillate slightly, they certainly were never as erratic as the controls for an infeasible Navion maneuver. Hence, including the control rate plots should adequately replace the feasibility test of *Propagator.m*.

The second problem was more subjective, but it is fairly obvious (especially for certain maneuvers) that the HARV and UCAV-X trajectories are not as smooth as the Navion trajectories were. This is particularly apparent when watching video clips of a maneuver, but can also be seen by noting sharp angles on a trajectory plot line. Fortunately the cause of the problem was quickly determined by comparing trajectory plots for a number of different Navion, HARV and UCAV-X solutions of varying node values. Take the following trajectory, for example:

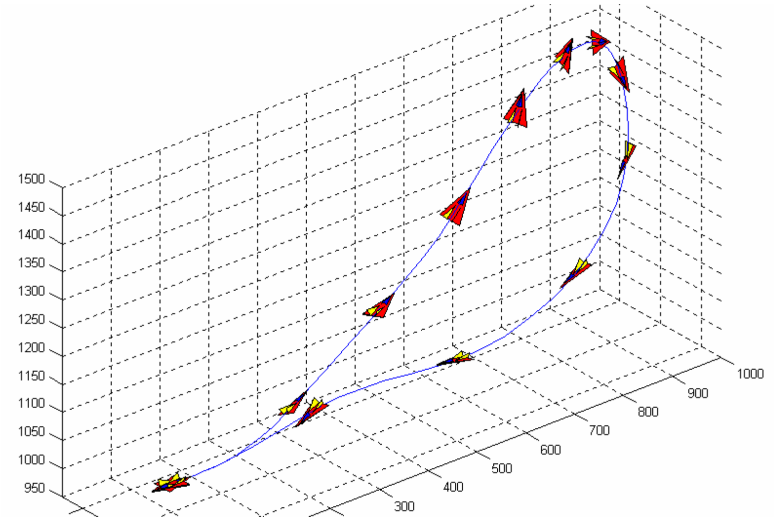


Figure 26. 80 Node Navion Trajectory

First of all, the trajectory is very smooth: the Navion inscribes a nice semi-circle at the top of the maneuver, and the last three snapshots curve together perfectly. Also, there are no two sequential snapshots that look like they are missing a step between them (although that is not surprising for this maneuver as there are no rapid reorientations).

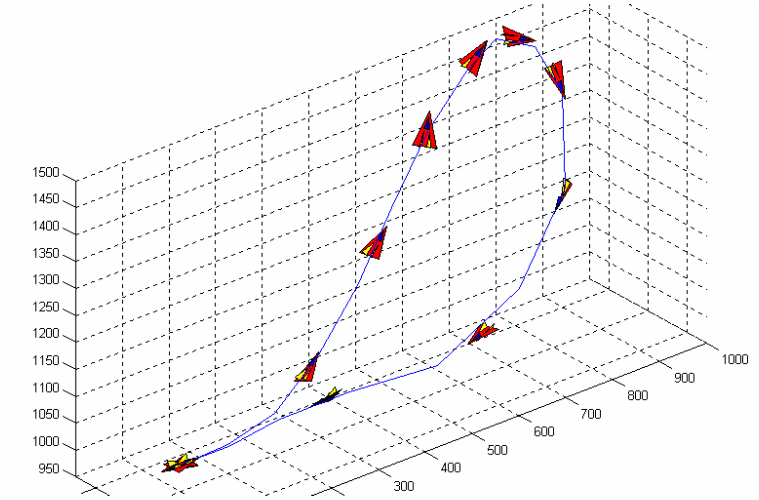


Figure 27. 20 Node Navion Trajectory

The trajectory pictured above is for exactly the same maneuver, but only an initial 20 node solution. Notice that there are several points at the top of the maneuver and in the descent where the trajectory line forms an angle instead of a curve. The reason for this is simply that the maneuver was only analyzed at 20 discrete points, and values between those points were just derived by interpolation.

After making the above comparison, it was easy to see that most of the HARV and UCAV-X solutions had the same “jagged” appearance that most 20 node Navion solutions had. (Although in some cases they were more extreme, because those maneuvers were more dynamic than the Navion maneuvers, and therefore suffered more from not having a sufficient number of nodes.) The reason for this problem was that bootstrapping any HARV or UCAV-X maneuver from a “locally optimal” 20 node solution to any higher node solution (40, 80, 160, etc.) *always* resulted in a DIDO output of “no solution.” Even though the results would look good, and would not violate any BC’s, DIDO would always term it a non-solution. So, all of the HARV and UCAV-X results presented in this study are, in actuality, 20 node solutions. While the results are still valid, they are just not as refined as a full 80-120 node solution would be. For example, the Navion reversal had a slightly different trajectory with 80 nodes than it did with 20 nodes (as previously discussed), and the maneuvering time was 19.0 sec versus 18.2 sec. Besides that, the two solutions are just different resolutions of the same time-optimal maneuver.

Now despite the fact that the HARV and UCAV-X maneuvers would not *correctly* bootstrap to a higher number of nodes, all solutions in this study were bootstrapped to 80 nodes. This was done to increase the density of points in all of the *Analysis.m* plots, and make them easier to read. However, these additional points are just interpolations between the original 20 nodes. Also, since any maneuver that was bootstrapped to 80 nodes was considered a non-solution by DIDO, the Hamiltonian would increase from a peak value on the order of 10^{-1} to a peak value on the order of 10^{2}. (See the example Hamiltonian on the following page, from an 80 node HARV solution.)

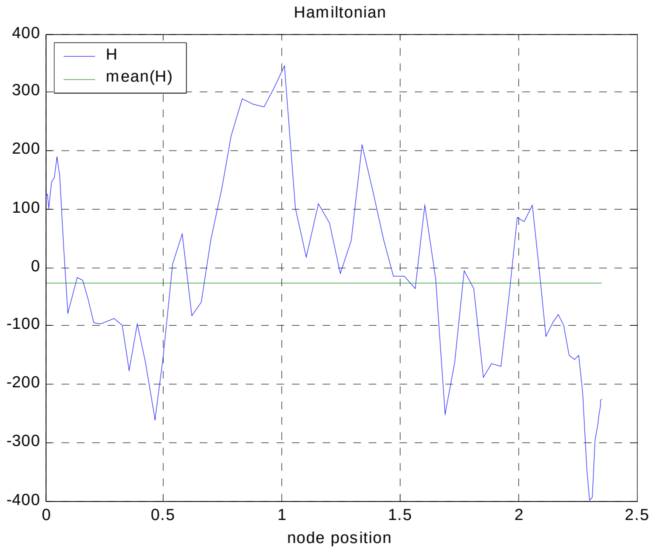


Figure 28. Example of a “No Solution” Hamiltonian

For this reason, the 20 node Hamiltonian will be displayed with the rest of the 80 node results for each HARV and UCAV-X maneuver. Since each result is really a 20 node solution, analyzing that Hamiltonian will give an accurate assessment of the optimality of the maneuver.

In conclusion, the reason that the HARV and UCAV-X maneuvers failed to bootstrap beyond 20 nodes is more than likely the same reason that *Propagator.m* failed to work: i.e. software incompatibility between TOMLAB/SOL v4.2 and DIDO 2003b. While a very unsatisfying reason, there is no other explanation for why these two problems occurred with *every* HARV and UCAV-X maneuver. Had the problems been limited to just one aircraft model or only certain maneuvers, another reason would be more likely, but that was not the case. However, in spite of these problem, the solutions that were presented in this section should allow the HARV and UCAV-X maneuvers to be studied in as much detail as the Navion maneuvers.

2. Optimal Maneuvers

The maneuvers that will be discussed in this subsection include one example of each of the four families of maneuvers defined at the end of Chapter II: the reversal, turning, pointing and braking maneuvers. To simplify discussion, a descriptive name was given to each maneuver. In addition, the BC's chosen for each maneuver will be mentioned in the discussion, as well as listed in the appropriate section of Appendix G. (As with the Navion, the "energy" and "displacement" constraints were the most commonly used.)

The first HARV maneuver was the "Classic Herbst" – a reversal. Both the energy and displacement constraints were used, and the maneuver resembled the Herbst as expected. (See Appendix G, page 199.) AOA maxed out at an impressive 63° during this maneuver, and sideslip values reached 30° (the maximum allowed by the aircraft code's state variable BC's). At 5.3 sec into the maneuver the HARV hit its peak altitude of 30,700 ft, as well as its max AOA and min velocity (344 fps). The min velocity was surprising high – only about 50 fps slower than the entry airspeed for this maneuver. Along the same lines, it was surprising that the HARV only climbed 700 ft while it traveled over 1600 ft down range. Due to the significant AOA values during the maneuver, a more "vertical" appearance was expected. Lateral displacement was limited to less than 300 ft either side of centerline, although the maneuver became somewhat jagged at the points of max lateral deflection. Finally, load factor was another surprise, peaking out at a very mild 3 g's at the beginning of the climb.

The "Classic Herbst" 20 node solution was evaluated as "locally optimal" by DIDO, and the Hamiltonian peaked out at +1 and –3 (with a mean value very near zero). The control rate history was extremely well-behaved, and deflection rates for each control surface stayed at or below their coded limits. Due to the control rate history, there was no chattering in the control deflection history, and most controls oscillated between their max and min deflection angles. The rest of the state variables were also well-behaved, and the final cost was reasonable – enough evidence to consider the maneuver time-optimal.

The second HARV maneuver was the "Falcon Turn" (see Appendix G, page 201). The energy constraint was used for this version of the turning maneuver, but not the displacement constraint: final values of X and Y were allowed to vary from –1000 ft to 1000 ft. The result was a very interesting maneuver that used a slightly canted loop to accomplish the desired 90° turn. Velocity peaked out at 482 fps in the dive preceding the loop, and AOA peaked twice during the maneuver: once in the loop (53°), and once at the end of the maneuver to bleed off airspeed (58°). Body rates were typical, with the exception of p, which spiked to almost 300°/sec at the beginning of the maneuver (note the corresponding switch in phi from –148° to 111°). The extent of the dive preceding the loop was surprising, because the HARV finished the loop with excess speed *and* altitude, necessitating a short braking maneuver. (See the discussion on the "Maverick" braking technique on page 117.) Finally, the "Falcon Turn" was another very low g maneuver, with n values ranging from about –0.5 to 2.5.

The "Falcon Turn" 20 node solution was evaluated as "locally optimal" by DIDO, and the Hamiltonian peaked out at +2.5 (with a mean value very near zero). The control rate history was extremely well-behaved, and there was no chattering in the control deflection history. The rest of the state variables were also well-behaved, and the final cost was reasonable – enough evidence to consider the maneuver time-optimal.

Some final thoughts: adding the displacement constraint to the turning maneuver resulted in the trajectory depicted below. The major differences between it and the "Falcon Turn" are that the latter exhibited slightly more dynamic maneuvering, had a better Hamiltonian, and was 0.7 sec faster. However, the alternate maneuver was an equally valid time-optimal maneuver for its specific BC's.

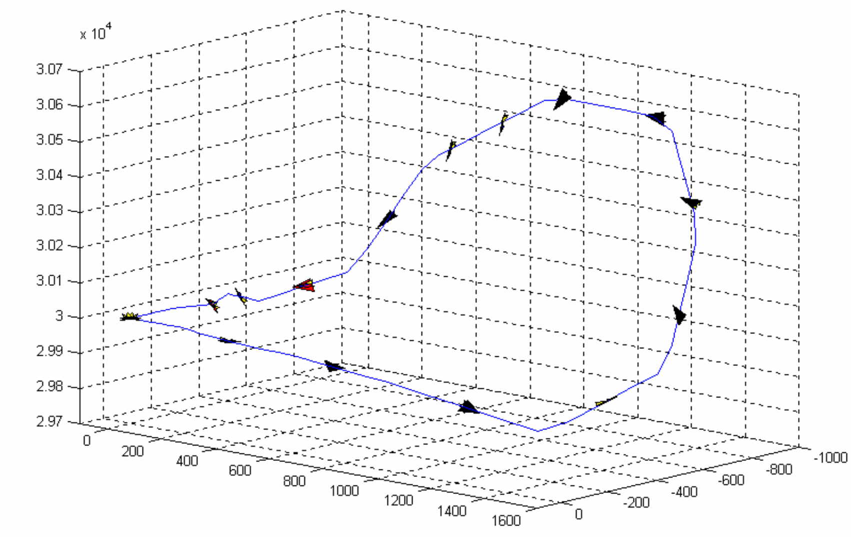


Figure 29. Alternate HARV Turning Maneuver

The third HARV maneuver was simply called the "Point & Shoot B" maneuver, since it followed the same format as the first pointing maneuver in this study (which was performed by the Navion). Position variables X, Y and H were all unconstrained by ±3000 ft, giving the HARV a large volume of space to maneuver in. Velocity was also unconstrained by plus of minus 100 fps, since the goal of this maneuver was to rapidly reorient the aircraft so it could put weapons on target (versus gaining a positional advantage on an opponent, where energy management would be more of a priority). The appearance of the maneuver, as already mentioned, was unsurprising (see Appendix G, page 203): a rapid pitch-up to achieve the commanded θ, and then a bank and turn to the right to achieve the commanded ψ. The control inputs perfectly matched those movements: nearly full deflection of the stabilator, TEF's and pitch TV from 2 sec through the end of the maneuver, and short, max deflections of the rudder, ailerons and yaw TV at the end of the maneuver. In addition, max δ_{sa} is used to help pitch the aircraft at the beginning of the maneuver, and the max δ_{ds} is used to help yaw the aircraft at the end of the maneuver.

Max AOA during this maneuver was 62°, and an AOA greater than 30° was held for almost 2 sec. Also, velocity decreased to the minimum allowed value of 298 fps. Based on these observations, increasing the velocity tolerance to ±200 fps would probably allow the HARV to sustain a high AOA during more of the maneuver, which would take one or two seconds off the maneuvering time. On the other hand, constraining velocity to exactly 398 fps resulted in the maneuver depicted on the following page, which took an additional 9.4 sec to complete. (The HARV had to accelerate to almost 550 fps in the dive in order to complete the maneuver at the required airspeed.)

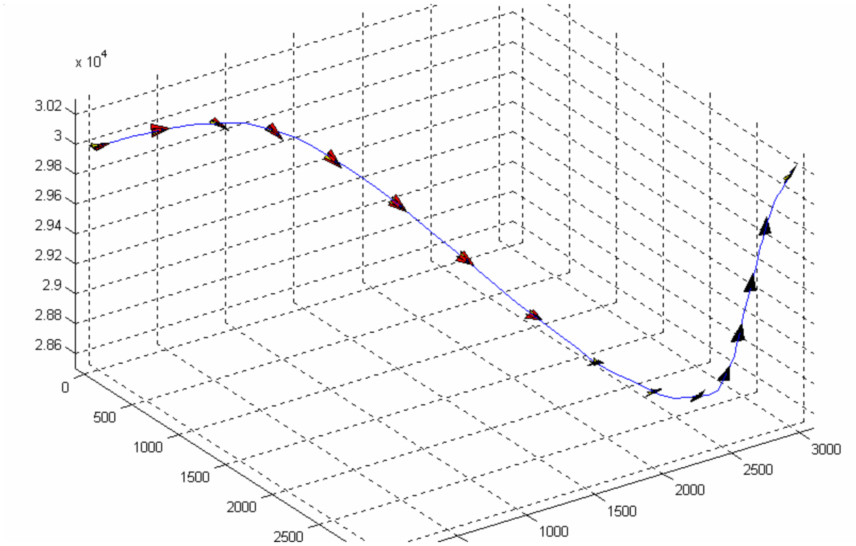


Figure 30. Alternate HARV Pointing Maneuver

The “Point & Shoot B” 20 node solution was evaluated as “locally optimal” by DIDO, and the Hamiltonian peaked out at –1.9 at the end of the maneuver (mean value approximately 0.2). As discussed above, the controls performed very well – in general switching from max to min at their maximum deflection rates. Other state variables were well-behaved, with no high frequency chatter in the Euler angles or body rates (the usual suspects). Final cost was excellent, but believable: the HARV achieved a $\Delta\theta$ and $\Delta\psi$ of 60° in 6.7 sec (the Navion took 3.9 sec for delta’s of only 30°). In conclusion, a good time-optimal maneuver.

Some final thoughts: looking at the Euler angle plot for this maneuver revealed that the HARV actually hit its desired attitude much earlier than 6.7 sec. The pitch angle was greater than 60° after 3.3 sec, and the yaw angle was greater than 60° after 4.9 sec. However, while θ was stabilized at about 80° (until dropping to 60° at the very end of the maneuver), ψ oscillated almost constantly from 4.9 to 6.7 sec. (Checking the body rate values revealed that the average yaw rate was positive or negative 35°/sec during that last 2.8 sec of the maneuver.)

The last HARV maneuver recorded for this study was a braking maneuver (see Appendix G, page 205). The expected outcome of this maneuver was a rapid transition to a high AOA, holding that attitude with the help of TV, and then recovering to straight and level flight after airspeed had dropped by 75% (which, for the HARV, meant slowing from 398 fps to only 100 fps). However, the actual result was not that straight forward, and due to the rather bewildering appearance of the maneuver it was called the "Crazy Straw." There are essentially four parts to the maneuver: two climbs and two descents.

The two climbs decelerate the HARV quite effectively, at rates of approximately –14.3 and –16.7 ft/sec^2. A large part of that deceleration is due to the fact that the HARV climbs at a nearly vertical attitude, and is even angled beyond the vertical plane at the peak of each climb. (In fact, the HARV spends almost 66% of this maneuver at an AOA greater than 30°, and a full 33% of the maneuver is flown at greater than 50°.) The descents, however, are counter intuitive: in each descent the HARV loses almost 1500 ft of altitude and (more importantly) gains almost 100 fps in airspeed. The most likely explanation for them is that the HARV is gaining energy needed to complete the maneuver at the specified final conditions. (A difficult thing to do since the last third of the maneuver is flown below stall speed, and at altitude the HARV doesn't have enough thrust to maintain a steep climb without some kinetic energy stored up.)

All in all an unusual maneuver, and based on how long the maneuver took (49 sec), it was not surprising to find that the "Crazy Straw" was not a time-optimal maneuver. Even though it met DIDO's criteria of "locally optimal," the Hamiltonian was just too large. In fact, its average value was greater than the peak values of any previous maneuver. In spite of that, examining the feasibility of the maneuver revealed that the control and state variable histories were well-behaved. Even though the HARV went through some unusual attitudes, body rates were very small (typically around 20°/sec), and none of the Euler angles exceeded their maximum values. Also, control inputs logically matched the HARV's movement and orientation (note the max deflections of the stabilator and TEF's at the beginning of each climb, for example). In conclusion, while the "Crazy Straw" was not a time-optimal maneuver, the results demonstrate that the HARV could actually perform it.

A good example of what a time-optimal braking maneuver would probably look like is the "Maverick." This technique was used by the HARV at the end of both the "Falcon Turn" and the "Classic Herbst," and is pictured on the following page:

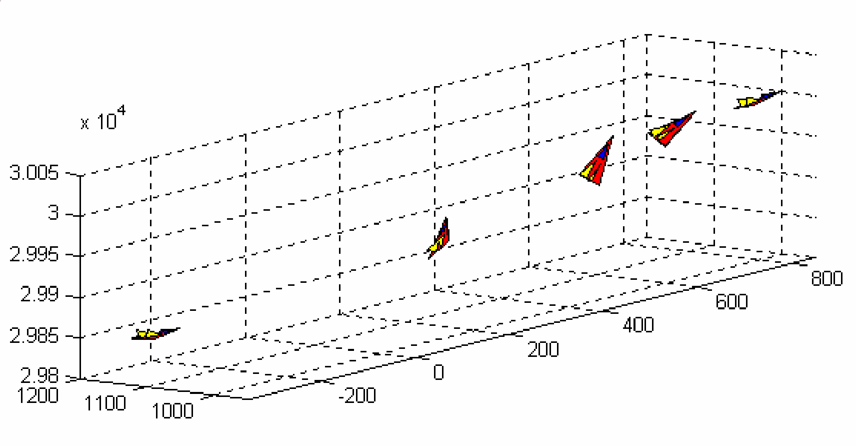


Figure 31. “Maverick” Braking Technique

It is similar in appearance to a mild version of the “Cobra” maneuver. One very interesting difference, though, is that the “Maverick” isn’t a strictly two-dimensional maneuver: the HARV rolls 48° left wing down at the beginning of the maneuver, and then rolls to 28° right wing down about halfway through the maneuver. The primary result of these bank angles is that the HARV traveled slightly to the left, and as a result did not gain as much altitude as it would have otherwise. (This would be important for a ‘defender becoming the attacker’ scenario: if the defender did a braking maneuver but gained too much altitude to immediately take a shot at his attacker, the attacker may have time to maneuver into a better position or bug out.) Some sample data on the Maverick (taken from the “Falcon Turn” results) is listed in the table below:

Table 14. “Maverick” Data

Time (sec)	*H (ft)*	*V (fps)*	*α (deg)*
8.8	29,854	454	2
10.1	29,904	448	43
11.4	29,962	403	58
11.8	29,984	394	32
12.3	29,996	394	6

Trends that are characteristic of both the “Maverick” and “Cobra” include the rapid pitch-up (56° in 2.6 sec), and the rapid deceleration (–20.0 ft/sec^2). Note the improved deceleration of the “Maverick” over the “Crazy Straw” – at this new rate it would take approximately 15 sec to complete a full braking maneuver. Presumably, with the right delta expression and BC’s, DIDO would be able to calculate a time-optimal braking maneuver for the HARV that utilized the “Maverick” technique.

3. Comparing to Navion Maneuvers

The general prediction for the HARV maneuvers was that they would be faster, flown at higher AOA, and demonstrate more unusual attitudes than the Navion maneuvers (thanks largely to the HARV’s TV unit). This section will compare and contrast the two sets of maneuvers in terms of general appearance, maneuvering time, control inputs, etc.

The reversal maneuvers performed by the Navion and HARV looked very similar to the two versions of the Herbst maneuver presented in Chapter II (see Figures 9 and 10, respectively, on page 21). Both aircraft climbed about half the distance they traveled downrange, and hit their peak altitude, minimum velocity and max AOA at the same relative time – halfway through the maneuver. However, while the pattern of the maneuvers was the same, the results were not. For one, the bird’s-eye view of the two maneuvers reveal that the HARV inscribed a much narrower path than the Navion did (as predicted). Also, the HARV did not lose as much airspeed as the Navion: its velocity had only dropped by 14% at the peak of the maneuver, while the Navion’s velocity had decreased by a significant 84%. This last difference is almost certainly due to the fact that the HARV was flying at AOA values up to 63°, which would have eliminated the aircraft’s *forward* component of velocity without necessitating a drastic reduction in total velocity. (Note the extensive use of pitch TV when the HARV is flying at high AOA.) In conclusion, the HARV completes the maneuver in 11.8 sec, a full 7.2 sec faster than the Navion.

Unlike the reversal, the Navion and HARV turning maneuvers were markedly different. (BC’s were slightly different as well: the HARV did *not* use the displacement constraint, while the Navion did.) While both took advantage of maneuvering in the vertical plane to minimize distance traveled (and therefore maneuvering time), they did so to different degrees. The Navion used full elevator deflection for almost the entire climbing portion of the maneuver, flying at or near its max AOA of 20°. However, even with that configuration it took 16.7 sec to reach its peak altitude, and 28.4 sec for the entire maneuver. The HARV, on the other hand, accomplished its direction change in an almost purely vertical loop that only took 7 sec, and finished the maneuver by 12.8 sec. (Note the full deflection of all longitudinal controls between approximately 4 and 6 sec, when entering the loop. In addition, pitch TV was used throughout almost the entire maneuver.) In conclusion, the HARV's post-stall control authority allowed it to rapidly accomplish a 90° heading change while minimizing the effective turning radius.

The pointing maneuvers for the Navion and HARV were nearly identical. Two versions were made of each: one with the constraint that $V_f = V_i$, and another where $V_i - 50 \le V_f \le V_i + 50$ (where 50 fps was used for the Navion and 100 fps was used for the HARV). Only one version of each was recorded in the Appendices, but both versions will be discussed here. For the first version (velocity constrained), both aircraft pushed over into a dive before climbing and turning to the correct pitch and yaw angles (30° for the Navion, 60° for the HARV). The Navion achieved 200 fps in the dive, and the HARV an impressive 550 fps; final times were 3.9 sec and 16.1 sec. Without the velocity constraint, both aircraft immediately climbed – the HARV completed the maneuver in 6.7 sec at its minimum allowable airspeed (298 fps), while the Navion only decelerated to 155 fps and finished in 2.5 sec. While the time differences aren't indicative of performance (due to the different delta expressions), it is interesting to note some other differences: the Navion almost hit its max positive and negative g-limits, while the HARV experienced very few g's (despite its much greater g-tolerance). Also, the Navion only slowed down 21 fps (12% of V_i) to reach its desired attitude, while the HARV slowed by 200 fps (50% of V_i).

The braking maneuvers of the Navion and HARV can not really be compared since the HARV solution was not a time-optimal maneuver. However, there were some similarities that can be noted between the Navion braking maneuver and the climbing portions of the “Crazy Straw” maneuver. Both were performed at max or near max AOA, and both utilized full deflection of an effective control surface (elevator for the Navion, predominately pitch TV for the HARV). Also, the deceleration rates were very similar: –15.4 ft/s^2 for the Navion, and –14.3 to –16.7 ft/s^2 for the HARV. (As already mentioned, for a time-optimal braking maneuver for the HARV, a better rate would be expected.)

4. Comparing to Previous Optimal Maneuver Studies

There have been a number of excellent studies on maneuver optimization published in the last sixteen years. Six of these papers (published between 1989 and 2002) were singled out to be included in this study for comparison with the HARV model and DIDO. These papers represent several different optimization techniques, and utilize three different aircraft (to include the F-18 HARV). Also, due to the span of time covered by these papers, they illustrate the increased complexity of analysis that is now possible, versus what was possible sixteen years ago. This increased complexity is due in large part to increased computational ability: i.e. computers can now handle more complete EOM and much more detailed aircraft aerodynamic data. Based on this, both aircraft models *and* optimization tools have improved dramatically, allowing for more realistic simulation of air combat maneuvers.

The primary goals of this section are as follows: to compare DIDO with the optimization routines used in each paper, to compare the HARV model with the aircraft models developed for each paper, and (where applicable) to compare results. The two most recent papers will be analyzed first in detail. The other four papers will then be analyzed selectively, with specific characteristics about their code or model being highlighted for comparison. Also, additional papers that were considered applicable, but were not included in this study, will be listed for reference. In conclusion, the comparisons made in this section should demonstrate that the performance of DIDO and the aircraft code are on par with other techniques currently being used in the field of maneuver optimization.

a. Komduur & Visser

Komduur and Visser's 2002 paper titled "Optimization of Vertical Plane Cobralike Pitch Reversal Maneuvers" [Ref. 31] was one of the very first to study maneuver optimization using the full 6DOF EOM. That, combined with the fact that the F-18 HARV was used as the test aircraft, made this paper an excellent benchmark for comparing DIDO and the aircraft code against. To start with, Komduur and Visser's HARV model was in most respects very similar to the one developed in this study: it was a rigid body model complete with dimensions and moments of inertia; deflection *and* rate limits were included for every control surface, and the AOA regime was restricted to values for which aerodynamic data was available. Most of this data was found in Iliff and Wang's papers [Ref. 19-20]; however, Komduur and Visser only reference *one* of these papers (surprisingly *not* the one containing longitudinal S&C data, which would be the most relevant for a vertical plane maneuver). While this was most likely just an omission, none of their HARV references contain any drag data. Based on their results, drag certainly was not left out, but without knowing where the data came from, it is questionable how accurately their aerodynamic model simulates the HARV.

There were two other major differences between Komduur and Visser's HARV model and the one developed in this study. The first was simply that they chose to use max thrust with afterburner for their model, vs. max dry thrust. (Afterburner was not used for this study because the data was not available for calculating an accurate wet thrust at altitude.) While this would certainly affect the results (noticeable velocity and maneuvering time), both approaches are legitimate. The other difference between the two HARV models was the use of control surfaces. For one, neither LEF's, TEF's, δ_{ds}, δ_{sa} nor the speedbrake were used in Komduur and Visser's model. Secondly, ailerons, rudder and yaw TV were only used to test the feasibility of the maneuver, which meant that the stabilator and pitch TV were the only controls included in the actual optimization process. Lastly, those two controls were linked during the optimization, so they were in effect a single control (a design that the authors admitted was simplistic, but useful for their purposes).

Moving onto Komduur and Visser's code, one thing in particular that stood out was their problem formulation: first, they chose a specific family of maneuvers to be optimized (in this case vertical plane pitch reversals). Second, they very clearly defined BC's (like the horizontal distance, final velocity and pointing constraints) to describe four distinct maneuvers. This methodology was very similar to the one used for this study: in place of vertical plane pitch reversals, a broader family of braking maneuvers was studied, and the velocity and distance constraints were very similar to this study's energy and displacement constraints. In addition to good problem formulation, Komduur and Visser also used a good program for their optimization: a nonlinear programming and collocation technique called DOVNLPAC. Another strong point of their code was that following optimization, the results were verified by a 6DOF simulator that used a nonlinear inversion closed-loop control technique. However, unlike the aircraft code's *Propagator.m* file (which verifies the feasibility of DIDO's control history), their 6DOF simulator has to develop its own (sub-optimal) control history.

The biggest drawback to Komduur and Visser's code actually had nothing to do with the code itself, but rather with their decision to constrain their pitch reversals to the vertical plane, and simplify their EOM from 6DOF to 3DOF. While a perfectly valid decision for the purpose of their study, their resulting maneuvers are consequently only time-optimal given a very specific set of BC's. Also, because their maneuvers were optimized with only 3DOF, the results of their study give no indication of the ability of their program DOVNLPAC to optimize a 6DOF maneuver. According to the authors, the program can handle "about 1000 variables plus boundary conditions" [Ref. 31: p. 699]. However, it is uncertain how those variables are counted, since the authors also state

> The step size for the optimization was chosen as small as possible…to capture the fast dynamics of the maneuvers. Too large a step size would result in spiky behavior of pitch rate and control variables. Decreasing step size means increasing the time needed for computations and increasing the total number of variables and boundary conditions. [Ref. 31: p. 699]

Decreasing the step size in DOVNLPAC obviously has the same effect as increasing the number of nodes in DIDO: higher definition results but longer run times. Apparently DOVNLPAC has an additional limiting factor in how many variables it can handle with a given step size. Lastly, some other data that would be nice to compare with DIDO is how long run times were for DOVNLPAC and the 6DOF simulator, and what kind of computer the runs were performed on.

Unfortunately, Komduur and Visser's results can not be directly compared to DIDO's results, as the HARV braking maneuver was not a time-optimal solution. Also, their problem formulation was slightly different…their first case, pictured below, used a "horizontal distance constraint" of $X_f = V_0 t_f - 1000$, while the delta expression for the braking maneuver (expressed in similar terms) was $V_f = V_0 \cdot 0.25$.

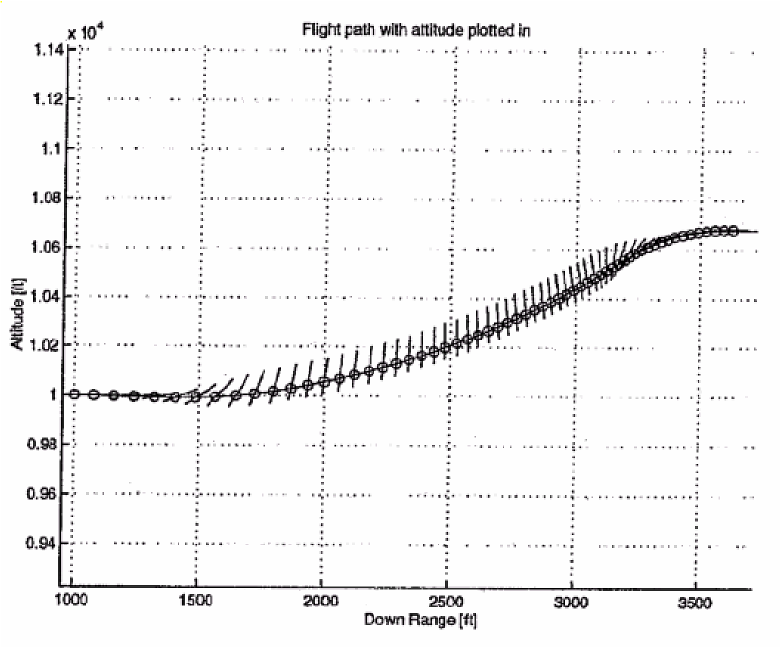


Figure 32. Vertical Plane Pitch Reversal (From: Ref. 31, pp. 695)

However, the maneuver does share some commonalities with the Navion and UCAV-X braking maneuvers: a rapid pitch-up, a gradual velocity decrease, and a generally two-dimensional appearance. Comparing specifically with the Navion, Komduur and Visser's HARV takes 11.5 sec to complete the maneuver vs. the Navion's 5.7 sec…the HARV's final velocity is 47.5% of V_0, practically identical to the Navion's velocity decrease of 50%…and lastly, the HARV's trajectory is not as steep as the Navion's. (No comments on performance should be drawn from the above comparisons as the maneuvers had different goals.) The following figure shows the HARV's control history for the maneuver just discussed.

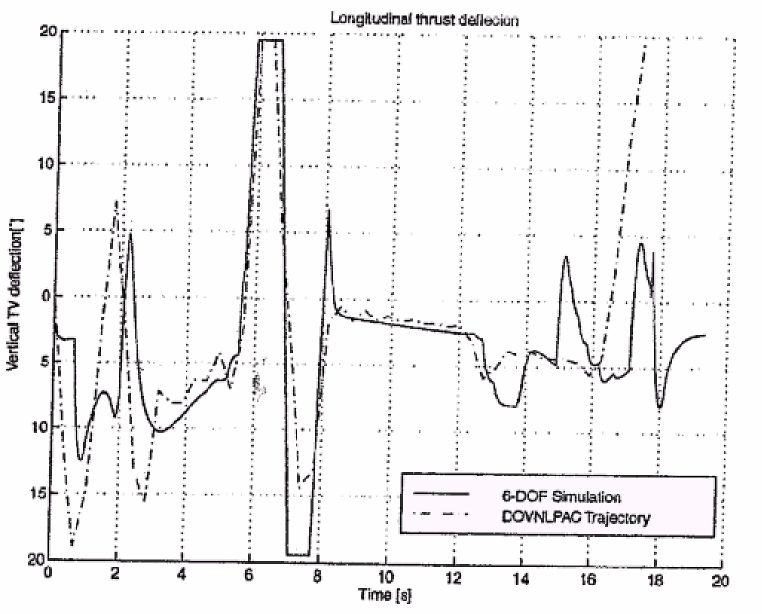


Figure 33. Pitch TV Control History (From: Ref. 31, pp. 701)

Similar to what the results of *Propagator.m* look like for certain maneuvers, the 6DOF simulator does not exactly match the control inputs calculated during optimization. However, since the two plots are relatively close, Komduur and Visser's HARV should be able to fly something very similar to their calculated optimal maneuver

b. Lichtsinder, Kreindler & Gal-Or

In 1998 Lichtsinder, Kreindler and Gal-Or published a paper titled "Minimum-Time Maneuvers of Thrust-Vectored Aircraft" [Ref. 32]. The focus of their study was minimum-time pitch and yaw reversals, as performed by a TV-equipped F-15B. Before presenting their results, though, Lichtsinder et. al. made two very significant observations. The first was noting the relative effectiveness of aerodynamic vs. TV control in different flight regimes (as defined by velocity, altitude and AOA). Specifically, they made an excellent graphical representation of the effectiveness of the elevator and pitch TV on the F-15B, which is pictured on the following page:

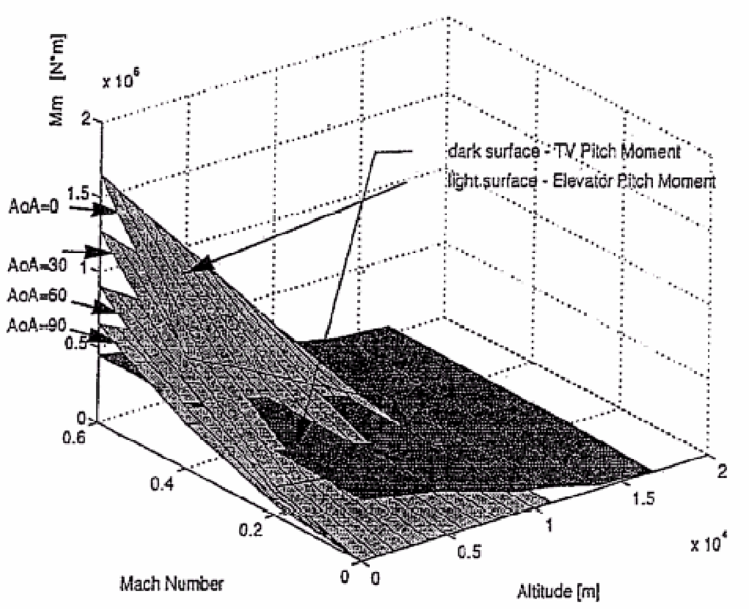


Figure 34. Aerodynamic vs. TV Effectiveness (From: Ref. 32, pp. 245)

It's evident from this figure that TV benefits are maximized during high AOA and low speed maneuvers. While this generality was already known, the graph above provides a much clearer picture of the exact relationship between the afore-mentioned variables. Lichtsinder et. al. also made a very convincing case for the necessity of using a full 6DOF model for aircraft maneuver optimization: at the beginning of their paper they discussed the results of performing a pitch reversal without any lateral-directional controls. During the simulation, the F-15B experienced significant (and undesirable) heading change and roll. They concluded that due to "asymmetrical forebody vortices" [Ref. 32: p. 245] it was imperative that a full set of aerodynamic and TV controls be utilized for realistic maneuver optimization.

The 6DOF F-15B model created for their study was very thorough: ailerons, rudder, stabilator, pitch and yaw TV and even δ_{ds} were all included with both deflection and rate limits. While the physical aspects of the model were not discussed at length, it is assumed that accurate moments of inertia and other important data was included. Aerodynamic data was developed in almost exactly the same manner that the HARV and UCAV-X aerodynamic models were created. According to the authors, the S&C derivatives were

> …nonlinear functions of state and control variables…represented as multidimensional polynomials of up to ninth order. These polynomials were data fitted from empirical results… [Ref. 32: p. 247]

In conclusion, a well-designed and realistic model.

While the F-15B model was very detailed, this caused some problems with the optimization process. In short, Lichtsinder et. al. had to make several simplifications to the problem in order to handle the complexity of the 6DOF aircraft and EOM. Some of these were minor, such as formulating the problem as two steps (a "to target" phase and a "recovery" phase) Other were more significant: for instance, Lichtsinder et. al. replaced their *only* two state constraints (which defined min and max values for θ and ψ) with a "weighted criterion of time and square errors" [Ref. 32: p. 248]. In the end, they settled on a sub-optimal solution to the problem.

Compromising between a detailed aircraft model and high quality optimization is a theme seen in every single paper on maneuver optimization. Typically the twelve coupled, nonlinear EOM are too complicated to have both, so the quality of one has to slightly sacrificed. In the case of Lichtsinder et. al. it was optimization…in the case of Komduur and Visser it was the aircraft model, which was reduced to 3DOF. (These two options are, in fact, the most common workarounds.) One of the goals of this study was to show that it was possible to have both.

Lichtsinder et. al. calculated at the conclusion of the their paper that their TV F-15B could perform an 80° pitch reversal in 5.2 sec. (The trajectory is pictured on the following page.)

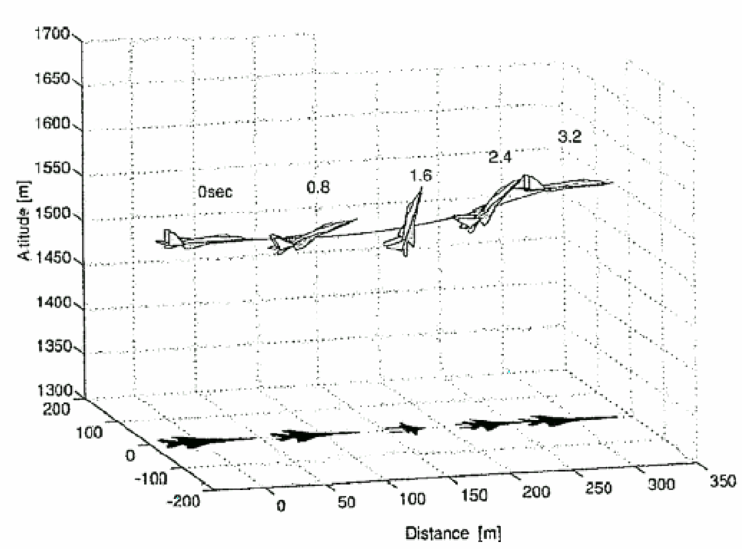


Figure 35. 80° Pitch Reversal (From: Ref. 32, pp. 249)

While the maneuvering time seems unrealistically fast, the problem formulation is slightly different than Komduur and Visser's definition of a pitch reversal, *or* this study's definition of a braking maneuver, which may be a factor. However, the lack of state variable BC's (in particular a max pitch rate) could have noticeably reduced maneuvering time. Also, considering the rapid reorientation around the y-axis, it would be interesting to see what value of I_Y was used for the F-15B model. Finally, as already mentioned this result was a sub-optimal maneuver, and as the authors succinctly conclude, "one never knows [how close the result is to optimum] unless the optimal solution is available." [Ref. 32: p. 250]

c. Other Studies

Komduur and Visser's work was largely based on two other studies: one published in 2000 by Horie and Conway [Ref. 33], and the other published in 1997 by Murayama and Hull [Ref. 34]. Both papers will be discussed briefly, and in light of their similarity in subject matter to each other and Komduur and Visser's paper, comparing the three studies should provide an excellent description of recent progress in the field of maneuver optimization.

To start with, Horie and Conway used a code very similar to Komduur and Visser's, called DCNLP (direct collocation with nonlinear programming). However, while the optimization method was the same, Komduur and Visser's analysis was likely more detailed than the 20 nodes Horie and Conway used. Another similarity between the two was problem formulation, and the use of various terminal constraints to define several different cobra-like maneuvers. The two biggest detractors to Horie and Conway's work were their 3DOF EOM (also used by Komduur and Visser) and their aircraft model. Their model was an F-16 like aircraft equipped with TV…however, it was a very basic: 1) It was a point-mass model with no moments of inertia. 2) Only lift and drag data was included for aerodynamic information. 3) The T/W ratio was arbitrarily set to 1.0. Obviously, replacing this model with a rigid, 6DOF model was Komduur and Visser's most significant contribution to the "minimum-time Cobra" problem.

Murayama and Hull's paper was published only three years prior to Horie and Conway's, so much of their work was unchanged. Their aircraft model, in particular, was exactly the same, and their problem formulation and method of using terminal constraints was also closely followed. The biggest difference between the two was their optimization technique: Murayama and Hull used a 4th order Runge-Kutta method and "sequential quadratic programming" [Ref. 34: p. 340] with 11 nodes and a convergence tolerance of 10^{-4}. Horie and Conway, on the other hand, used 20 nodes and a tolerance of 10^{-8} (among other improvements). So, as Komduur and Visser had improved on Horie and Conway's work with a better aircraft model, Horie and Conway improved on Murayama and Hull's work with better optimization.

To sum up, these three papers illustrate the progression of the "minimum-time Cobra" problem. While the problem has yet to be conclusively solved with a 6DOF model *and* optimization, it is very close. Hopefully in the next year DIDO and the aircraft code can make that last step. (For comparison, times for various cobra-like maneuvers from all three studies are listed in the table on the following page.)

Table 15. Results of Minimum-Time Cobra Studies

Constraints:	**Murayama & Hull (1997)**	**Horie & Conway (2000)**	**Komduur & Visser (2002)**	*(Key to Constraints)*
HD	8.3 sec	8.3 sec	11.5 sec	HD = horizontal distance
HD, γ_f	9.3 sec	9.0 sec	==//==	[Ref. 31: p. 694]
HD, γ_f , V_f	13.7 sec	==//==	==//==	TD = total distance
HD, γ_f , V_f, H_f	16.3 sec	13.6 sec	==//==	[Ref. 31: p. 695]
P	==//==	8.2 sec	==//==	P = pointing
P, V_f	==//==	10.0 sec	==//==	[Ref. 31: p. 695]
P, V_f, γ_f	==//==	12.0 sec	==//==	$\gamma_f = 0$
TD, P	==//==	==//==	13.1 sec	$V_f = V_0$
TD, P, V_f	==//==	==//==	14.9 sec	$H_f = H_0$
TD, P, V_f, γ_f	==//==	==//==	17.3 sec	==//==

The 1993 paper published by Bocvarov, Lutze and Cliff [Ref. 35] is unique in that it is one of few time-optimal maneuver studies that does *not* examine cobra-like pitch reversals. Instead, Bocvarov et. al. studied minimum-time reorientation maneuvers. Interestingly, in their analysis they reduced the EOM to 3DOF by eliminating translational variables (i.e. X, Y and H) entirely. This meant their EOM and aircraft model (which was based on the HARV) only contain roll, pitch and yaw data. Unfortunately, while their study was very well presented, the results would be impossible to compare to reorientation maneuvers performed by DIDO due to the lack of translational data. In spite of that, Bocvarov et. al. perform a very instructional analysis on the control authority of the HARV, as well as calculating what benefits were gained from TV. In their own words,

> The results for two classes of reorientation maneuvers suggest that it is worth enhancing the [F-18] with thrust-vectoring capability. The savings in maneuvering time are about 20-30%. [Ref. 35: p. 239]

Stalford and Hoffman’s 1989 paper [Ref. 36] was probably the first to use the HARV as a test aircraft. In fact, their study was published only two years after NASA had started test flights with the HARV. As with most studies that were published in the next decade, Stalford and Hoffman used a 3DOF model and EOM. For optimization they used Pontryagin’s maximum principle to solve a two-point boundary value problem…this method was replaced four years later by Bocvarov et. al. with a homotopy method and a multipoint boundary value problem (MPBVP). More recently, nonlinear programs like DCNLP [Ref. 33], DOVNLPAC [Ref. 31] and DIDO have become the preferred method of optimization. Nonetheless, with the model and tools that they had, Stalford and Hoffman calculated 9.1 sec for a pitch-up to 90° AOA using just the stabilator, and an impressive 1.8 sec with the stabilator and pitch TV.

In conclusion, hopefully the various papers discussed in these last few pages have illustrated the following points: first, that the study of aircraft maneuver optimization has progressed significantly in the past decade and a half. Second, that the HARV model developed for this study is as realistic as any model that has been used for maneuver optimization to date, and dramatically better than most. Third, that DIDO (in addition to its many other benefits) has proved more capable than any other optimization tool currently in use due to its successful optimization of aircraft maneuvers with 6DOF EOM. Also, while the papers discussed in this section were considered to be the best studies of aircraft maneuver optimization, a number of related papers were very informative as well. This papers are listed in the table on the following page for reference.

Table 16. Additional References on Maneuver Optimization

Year	***Author(s)***	***Title (Publishing Information)***
1997	Ericsson & Martin	“Conceptual Fluid/Motion Coupling in the Herbst Supermaneuver” (JA, Vol. 34, No. 3)
1996	Weiss, Friehmelt, Plaetschke & Rohlf	“X-31A System Identification Using Single-Surface Excitation at High AOA” (JA, Vol. 33. No.3)
1995	Alcorn, Croom & Francis	“The X-31 Experience: Aerodynamic Impediments to Post-Stall Agility” (AIAA 95-0362)
1995	Fan, Lutze & Cliff	“Time-Optimal Lateral Maneuvers of an Aircraft” (JGCD, Vol. 18, No. 5)
1994	Bocvarov, Cliff & Lutze	“Aircraft Time-Optimal Heading Reversal Maneuver” (AIAA 94-3556)
1994	Bocvarov, Cliff & Lutze	“Some Nonintuitive Features in Time-Efficient Attitude Maneuvers of Combat Aircraft” (JGCD, Vol. 17, No. 2)
1993	Dwyer & Lutze	“Quasi-Optimal Steady State and Transient Maneuvers With and Without Thrust Vectoring” (AIAA 93-3778)
1993	Rokhsaz & Steck	“Use of Neural Networks in Control of High-Alpha Maneuvers” (JGCD, Vol. 16, No. 5)
1993	Gal-Or & Baumann	“Mathematical Phenomenology for Thrust-Vectoring-Induced Agility Comparisons” (JA, Vol. 30, No. 2)
1992	Bugajski & Enns	“Nonlinear Control Law with Application to High Angle-of-Attack Flight” (JGCD, Vol. 15, No. 3)
1992	Menon & Duke	“Time-Optimal Aircraft Pursuit Evasion with a Weapon Envelope Constraint” (JGCD, Vol. 15, No. 2)
1991	Hoffman & Stalford	“Classical Turning Performance of a Fighter Aircraft Revisited” (AIAA 91-2667)

JA = Journal of Aircraft, JGCD = Journal of Guidance, Control and Dynamics

C. UCAV-X

The UCAV-X represented an opportunity to selectively choose the physical and aerodynamic characteristics for a test aircraft. In fact, based on the results of this study, further changes will be made to the model in an effort to find what characteristics consistently produce the best post-stall results. For this initial version of the UCAV-X, the most significant differences between it and the HARV are as follows: AOA range expanded by 50°, almost twice the T/W ratio at altitude, smaller moments of inertia, and control surfaces more conducive to high AOA maneuvering (i.e. canards vs. a stabilator). For this study, the goal of the UCAV-X was very clear-cut: to demonstrate dynamic post-stall maneuvers that could be compared to those performed by the HARV.

NOTE: As before, MATLAB plots summarizing the results of each maneuver are included in Appendix H. For this study, four maneuvers were included, and they are listed in Appendix H in the order they will be discussed in this section; the appendix should be followed closely while reading this section as the figures were not reproduced here. (The figures for the UCAV-X maneuvers are the same as those included for the HARV maneuvers; also, the points made about DIDO on page 106 and following apply equally to the UCAV-X.)

1. Optimal Maneuvers

The maneuvers that will be discussed in this subsection include one example of each of the four families of maneuvers defined at the end of Chapter II: the reversal, turning, pointing and braking maneuvers. To simplify discussion, a descriptive name was given to each maneuver. In addition, the BC's chosen for each maneuver will be mentioned in the discussion, as well as listed in the appropriate section of Appendix G. (As with the Navion and HARV, the "energy" and "displacement" constraints were the most commonly used.)

The first UCAV-X maneuver was the "Lazy Eight," which was a reversal that met the same criteria as the Herbst, but was very different in appearance (see Appendix H, page 207). The maneuver started with a 1000 ft descent before going into a steep loop that reversed the aircraft's direction, and even made it fly inverted for several seconds. In point of fact, AOA during the loop was over 50° for almost 4 sec, and peaked out at 89°. The velocity extremes during the maneuver were 524 fps after the initial descent, and 279 fps at the top of the loop. The reason that 518 fps was not used as the initial and final velocity for this maneuver was because DIDO (for whatever reason) could not compute a good, optimal trajectory with that velocity. So, while 518 fps worked as an entry velocity for the other UCAV-X maneuvers, for the "Lazy Eight" that speed was reduced by 25%

to 384 fps. Body rates were minimal during the maneuver, with the exception of roll rate (a situation that proved to be typical of both the HARV *and* UCAV-X). Finally, while no large positive g's were pulled during the maneuver, almost –3 g's were experienced about 7 sec into the maneuver. (Note the abrupt change in Y at that time – the UCAV-X was on its right side and essentially put the "stick" full forward to execute a left turn.)

The 20 node solution for the "Lazy Eight" was "locally optimal," and the mean value of the Hamiltonian was –0.3 (although the Hamiltonian itself was quite jagged, with peaks of 2, –5, 4 and –4). Control history looked very good for this maneuver, in particular the longitudinal controls – almost all controls were at max or min deflection during the entire maneuver, with a few breaks to switch from one to the other. The other state variables were well-behaved, and the final cost (while not as good as had been hoped for) was a reasonable 24.1 sec. In conclusion, both an optimal and feasible maneuver.

Some final thoughts: increasing the entry velocity for this maneuver by even 15 fps dramatically changed the appearance of the maneuver. Not only that, but the alternate trajectory (pictured below) is almost *exactly* what a 90° max AOA Herbst was predicted to look like in Chapter II. However, while the maneuver looks very efficient, it actually takes 3.5 sec longer to complete than the "Lazy Eight."

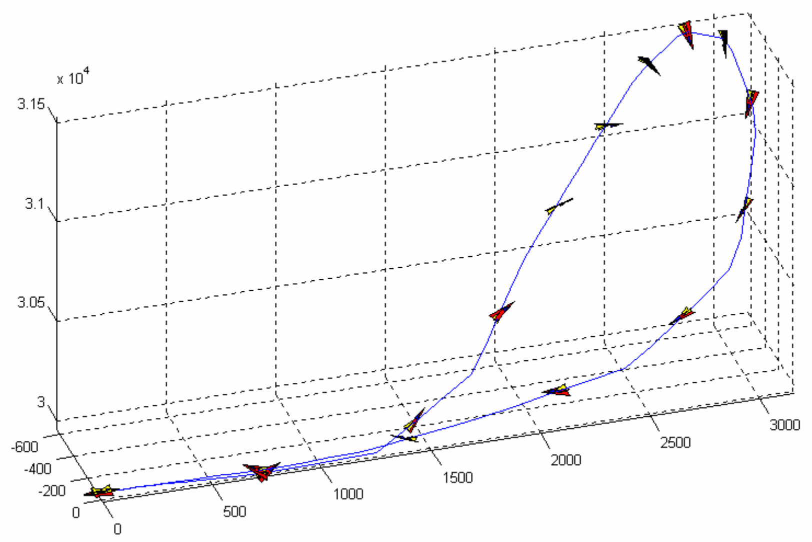


Figure 36. Alternate UCAV-X Reversal Maneuver

The next UCAV-X maneuver was the “Cloverleaf,” a turning maneuver resembling a freeway off-ramp. (The energy constraint was used for this maneuver, but X and Y were unconstrained by ±2000 ft. Results are listed in Appendix H, page 209.) The first part of the maneuver was an 8 sec, 2000 ft climb that slowed the UCAV-X from 512 fps to 425 fps. The aircraft started turning 90° left towards the top of the climb, and began an increasingly steeper descent. Notice at approximately 15 sec (where Y is almost 4000 ft) that the aircraft is up-side down ($\phi = 180^\circ$) and pointing almost straight down ($\theta = -85^\circ$): at this point the aircraft reverses its direction in the vertical, which can be seen by ψ rapidly switching from approximately –100° to +100°. Pulling out of this nose-down attitude was accompanied by the max positive load factor experienced during this maneuver, 3.5 g’s. On a similar note, there were four separate times during this maneuver where load factor was between –2 and –3 g’s, and at every one of those times the UCAV-X was inverted. Positive AOA during the maneuver was typically only 20-30°, with a maximum of 48°, while negative AOA was frequently maxed out at –20°. Lastly, minimum thrust was used for almost the entire maneuver, possibly to keep the UCAV-X within the allowed maneuvering space.

The 20 node solution for the “Cloverleaf” was evaluated as “locally optimal” by DIDO. The Hamiltonian had a very small mean value of –0.06, with max positive and negative values of 1.2 and –2.6 (the latter occurring at the very end of the maneuver). Control deflections alternated between max and min values for the most part, but the oscillations were more frequent. (Note the higher control rate values for this maneuver as compared to the “Lazy Eight.”) However, deflection limits remained within their allowable values, as did the other state variables. Final cost was reasonable (albeit somewhat slow) at 22.2 sec, and based on all of the above the maneuver was considered a time-optimal solution.

The third UCAV-X maneuver was the “Point and Shoot C,” which followed the same general pattern as the Navion and HARV pointing maneuvers (see Appendix H, page 211). Due to the increased velocity of the UCAV-X over the HARV, the maneuvering space was enlarged to a cube of ±5000 ft centered on the aircraft’s starting position. Final velocity was unconstrained by 200 fps, but surprisingly the UCAV-X finished the maneuver at 468 fps (versus the minimum allowable airspeed of 312 fps). Appearance closely matched the previous pointing maneuvers, with the exception of a slight detour to the left at the beginning of the climb. Control deflections were just as expected, with max deflection of nearly all longitudinal controls for the duration of the maneuver, and some rudder inputs at the beginning and end of the maneuver. Load factor was minimal, and AOA values were *extremely* small – the maximum positive and negative values were 6° and –9°, respectively. Body rates were typical, with the exception of high roll rates corresponding to two miniature aileron rolls at 3 sec and 11 sec.

The 20 node solution for the "Point and Shoot C" was "locally optimal," and had a relatively flat Hamiltonian with a peak value of only 0.9. As already mentioned, the control deflections for this maneuver were excellent: max and min deflections of most longitudinal controls, and switching from max to min for the lateral controls. The Euler angle plots were extremely smooth, as were the velocity and other state variable plots. In conclusion, a time optimal and feasible maneuver.

Some final thoughts: for the Navion and HARV pointing maneuvers, constraining final velocity to equal initial velocity significantly changed the appearance of the maneuver. Both aircraft had to descend and gain airspeed before they were able to make the climb to the required pitch and yaw angles. The UCAV-X, on the other hand, accelerated in level flight for about 4 sec and immediately went into the climb. Also, this alternate pointing maneuver (see figure) only took 2 sec longer to complete than the "Point and Shoot C" maneuver.

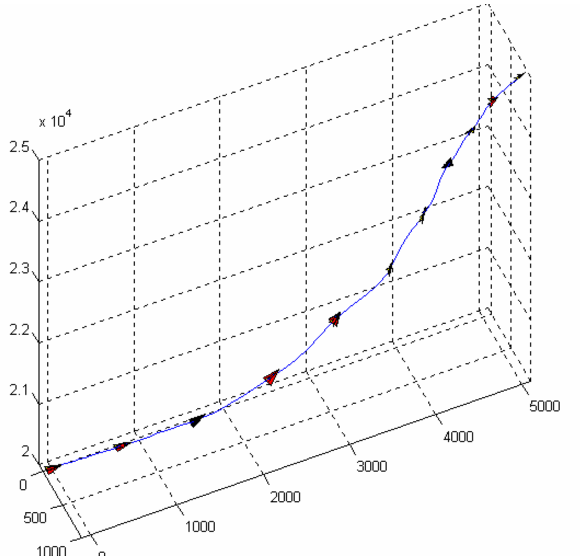


Figure 37. Alternate UCAV-X Pointing Maneuver

The final UCAV-X maneuver was the “Vertical Cobra,” which was an excellent example of a post-stall braking maneuver. The delta for the maneuver was a 75% reduction in velocity, and as with the pointing maneuver, all position variables were unconstrained by ±5000 ft. (See Appendix H, page 213.) The maneuver started with a rapid climb, and within 6 sec the UCAV-X had passed through 22,000 ft and was at its max allowable pitch angle of 85°. The key part of this maneuver started at about 8 sec: first, the aircraft rolls in the vertical to a bank angle of –180° (so the top of the aircraft is now facing “forward”). Then, the UCAV-X sets thrust to minimum and pulls its nose back down to the horizon; the aircraft travels *vertically* in this max AOA (“belly-up”) position for almost 4 sec before righting itself and finishing the maneuver in its normal cruise attitude. (Max load factor of 2 g’s occurred during the transition from a vertical to an inverted attitude.)

There are several impressive characteristics about this maneuver, one of which is symmetry. The “Vertical Cobra” is essentially two 90° pitch-up maneuvers performed back-to-back: the first pitches the aircraft to a vertical attitude, and starts to bleed off airspeed by gravity. The second pitches the aircraft to a level, inverted attitude, and bleeds off airspeed by both gravity and drag. (Both pitch-up maneuvers are similar in appearance to “Cobra” maneuvers that are limited to a max AOA and pitch angle of 90°.)

Secondly, the *rate* at which the UCAV-X decelerates is also impressive: the aircraft slows to its target airspeed in 15.3 sec, a rate of –34 ft/s^2. Lastly, AOA was greater than it had been for any previous maneuver in this study: it was over 50° for 4.0 sec, over 70° for 2.7 sec, and was actually pegged at 90° for 1.3 sec.

The 20 node solution to the "Vertical Cobra" was "locally optimal" with a very flat Hamiltonian. (The average value was –0.07, and the max and min values of 0.9 and –1.4 occurred at the very beginning and end of the maneuver, respectively.) Control history was excellent, with controls deflections either at their max or min values or switching between them. (Also, notice that every single one of the control surfaces are in transit from 8-10 sec when the UCAV-X is moving from a vertical to an inverted attitude.) The other state variable plots were smooth and well-behaved; even the body rates, which had roll rates that peaked out at 350°/sec and –212°/sec. All in all, the maneuver was a fine example of post-stall maneuverability, and certainly met the criteria of a feasible, time-optimal maneuver.

2. Comparing to HARV Maneuvers

The HARV and UCAV-X maneuvers were expected to be similar in appearance since the two aircraft were of comparable performance: high AOA maneuverability, post-stall control authority (TV), good T/W ratios, etc. In actuality, only the pointing maneuvers looked similar; the other maneuvers differed both in appearance *and* maneuvering time. This section will highlight those differences, and make some comments about why the two aircraft performed the way they did.

The HARV reversal was a good (if somewhat jagged) example of a Herbst maneuver, while the UCAV-X reversal was something completely different. Instead of climbing into a tight, high AOA turn like the "Classic Herbst," the "Lazy Eight" has the UCAV-X descend and execute a high AOA loop in order to reverse direction. While the technique was interesting, the result was somewhat disappointing: the maneuver took 24.1 sec to complete, as compared to the 11.7 sec needed for the HARV maneuver. The simplest explanation for the difference in maneuvering time is that the UCAV-X traveled a lot farther: 3450 ft in the x-direction versus 1650 ft, and 1200 ft in the z-direction versus 700 ft. That being said, determining *why* the UCAV-X trajectory was so much longer was a more difficult question.

For one, both aircraft typically used their control surfaces at max deflection (in particular the longitudinal controls for the UCAV-X and the lateral-directional controls and TEF's for the HARV). Also, both aircraft utilized almost their full range of AOA – the HARV peaked at 63° and the UCAV-X at 89°. However, even though the max AOA for the UCAV-X was almost 30° greater than it was for the HARV, the *average* AOA for the UCAV-X was actually 5° less. Looking at the AOA plot for the "Lazy Eight" shows that the UCAV-X flew at approximately zero AOA for almost half the maneuver – even when flying in a negative pitch attitude the UCAV-X did not fly at max *negative* AOA, which would have reduced its forward component of velocity and made its high AOA control devices more effective.

This brings up what was likely the primary detractor to the performance of the UCAV-X: due to an error in the aerodynamic model, the values for rudder and aileron deflection were linked (which is why the control deflection plot only shows one line for those two controls). This significantly limited the lateral-directional control authority of the UCAV-X, because any rudder input had to be tempered so it did not cause unwanted roll (likewise for aileron deflection and yaw). In addition to that, the rudder and ailerons had different max deflection limits, so with the controls linked the smaller of the two limits (±30°) was used for both. Finally, notice that the control deflection plot for the ailerons and rudder oscillates back and forth, and only comes close to max deflection once during the entire maneuver (trying to balance the needed rudder and aileron inputs). However, while the UCAV-X reversal and turning maneuvers were certainly hindered by the coding bug, the results were still well worth investigating, and made for an interesting contrast to the HARV maneuvers.

The HARV and UCAV-X turning maneuvers – the "Falcon Turn" and the "Cloverleaf" – were also very different. Oddly enough, for these two maneuvers it was the HARV that used a loop, and the UCAV-X that executed a high AOA climbing turn. However, the results were similar to those of the reversal maneuvers: 1) As before, the

UCAV-X traveled significantly farther than the HARV during the course of the maneuver (max ΔX, Y & H of 2900 ft, 4000 ft & 2800 ft, versus 1600 ft, 2100 ft & 600 ft). 2) The UCAV-X also had a proportionally slower maneuvering time: 22.2 sec versus the HARV's 12.8.

The problem of linked rudder and ailerons can easily be seen from the control deflection plot, and the constant oscillation of those controls between ±20° is a good indication that the UCAV-X was lacking in control authority for this maneuver. This was expected since a turning maneuver, by definition, will tend to need more lateral-directional control inputs than a reversal. For example, note the almost constant max deflection of the ailerons and rudder for the HARV's "Falcon Turn." Likewise, the yaw TV on the UCAV-X (which is an excellent example of optimal control) is either at, or switching to, max or min deflection for the entire "Cloverleaf" maneuver. As far as AOA goes, the max for the UCAV-X was 48° (versus 58° for the HARV), but it frequently flew at or near its min AOA of –20°. These min AOA values always coincided with phi values of plus or minus approximately 180° (instances of "adverse roll" caused by yawing the aircraft). They are also probably the reason for the low positive AOA values. So, as with the reversal, the UCAV-X turning maneuver was intriguing but not nearly as efficient as the corresponding HARV maneuver.

As already mentioned, the UCAV-X pointing maneuver was essentially the same maneuver that the Navion and HARV both performed. (With the caveat that the required deltas for the UCAV-X turning maneuver were 60° for theta and psi, as they were for the HARV.) For the velocity unconstrained version of this maneuver, both aircraft flew level for a second or two and then went into a climb (notice the slight increase and then steady drop in velocity, and the max or near-max deflection of the longitudinal controls for both aircraft). Final times for the two aircraft were 6.8 sec for the HARV and 13.0 sec for the UCAV-X. The likely cause of the difference between the two was that the HARV pointing maneuver allowed the aircraft to slow by 80 fps (or 20% of V_0), while the UCAV-X only slowed by about 50 fps (or 10% of V_0). On the other hand, for the velocity constrained versions of the pointing maneuver the UCAV-X finished in 15.0 sec – 1.1 sec *faster* than the HARV, and surprisingly only 2.0 sec slower than the velocity unconstrained version. Whether due to a higher T/W ratio or better longitudinal control authority, the UCAV-X certainly handled well in the vertical plane. (In fact, it had a climb rate of 380 fps, or *22,800* fpm. The HARV had an excellent climb rate as well, just not *as* impressive: 225 fps, or 13,500 fpm.) This vertical capability was put to excellent use in the next maneuver.

The UCAV-X braking maneuver, the “Vertical Cobra,” was definitely the aircraft’s best example of post-stall maneuverability (at least from the maneuvers developed for this study). In addition to that, since the HARV reversal was not an optimal solution, this maneuver filled in that blank and demonstrated what a time-optimal braking maneuver could look like for a post-stall fighter. Although already discussed, the maneuver was a nice mix of something instantly recognizable as a post-stall maneuver (the “Cobra” maneuver) with something new (performing a Cobra-like maneuver in the vertical). Also, since the maneuver was performed in a nearly 2D plane, not a lot of lateral-directional control inputs were necessary, allowing the UCAV-X to perform at its max potential. Like the UCAV-X, the HARV flew in a nearly vertical attitude during its braking maneuver, but was unable to transition from that attitude to the ending BC’s for the maneuver (hence the need for a second dive and a second climb). Hopefully in the future a different set of BC’s will produce a time-optimal braking maneuver for the HARV, which could then be compared in detail to the UCAV-X “Vertical Cobra,” as well as the traditional “Cobra.”

VII. CONCLUSIONS

The primary goal of this thesis was to develop several time-optimal air combat maneuvers. The last chapter described eight such maneuvers in detail – four flown by the HARV and four flown by the UCAV-X. Three of those maneuvers in particular demonstrated excellent post-stall characteristics: the Classic Herbst, the Falcon Turn and the Vertical Cobra. Each one took full advantage of a generous AOA range, good T/W ratio, and multi-axis TV to maintain controlled flight through the post-stall region. Common traits among the above maneuvers included roll rates in excess of 300°/sec, AOA pegged near or at limits, 30° of β, and extensive use of TV control surfaces. In general terms, the maneuvers were a nice mix of techniques both new (the Falcon Turn) and familiar (the Herbst and elements of the Cobra). This chapter will briefly review some of the necessary elements that were developed to create these maneuvers: the aircraft models, aircraft code, and DIDO.

After much coding and testing, the aircraft models for the HARV and UCAV-X were impressively realistic. Accurate physical data, complete sets of S&C derivatives, and effective limits on control rates make them arguably two of the most complete models that have been used in maneuver optimization to date. (The F-15B model created by Lichtsinder et. al. was the only model out of 18 papers researched for this thesis that had the same level of detail and accuracy.) Curve-fitting the S&C derivatives for those aircraft proved by far the most efficient means of coding the aerodynamic data – not quite as accurate as a table look-up, but significantly less time consuming. (On average it was about 15 times faster; see Tables 11 and 12 on pages 86-87.) The accuracy of the curve-fitting can be examined by comparing the plotted results in Appendix E to the table look-up plots in Appendix D. (See also Figure 21 on page 85.)

The basic format of the aircraft code went largely unchanged from the way it was originally written by Scott Josselyn. However, the code had to be rewritten for both the HARV and UCAV-X, and some major changes had to be made to incorporate the rate limit model discussed on page 69 and following. As with the original version of the code, though, state variables were scaled to improve numerical stability, and bootstrapping was used to improve the **guess** structure in *Main.m*. Extensive testing found that bootstrapping from an initial 20 node solution reduced run times by an average 56% (see Table 8 on page 77). Minimizing run times was a major challenge during the course of this thesis, but between curve-fitting, bootstrapping, and selectively choosing state variable constraints, run times for a fully converged HARV or UCAV-X maneuver were reduced to less than two hours. (Note that the times listed in Appendices G and H are from the 1998 computer, and are approximately 4-6 times longer than they would be on the primary 2002 computer.)

While the aircraft code was important framework for developing time-optimal maneuvers, the optimization code DIDO was what really made this entire study possible. The numerous advantages of DIDO include its minimum operating requirements (a standard home PC, MATLAB and TOMLAB), its ease of use, its ability to optimize 6DOF maneuvers *and* handle complicated aerodynamic data, and its ability to independently create unique, time-optimal maneuvers. Ease of use was a much appreciated quality of DIDO, as only a basic understanding of optimal control theory was required to take advantage of DIDO's capabilities. The next two advantages were demonstrated in detail last chapter by contrasting DIDO with programs from previous optimal maneuver studies (see the section starting on page 120). However, the greatest strength of DIDO is really summarized in the last advantage listed above: namely, that DIDO's format requires only loose parameters from the user, leaving DIDO considerable flexibility in deriving optimal maneuvers. This attribute allowed for the format utilized in this study of simply defining maneuvers as deltas of aircraft state variables.

Besides successfully developing several time-optimal maneuvers with the HARV and UCAV-X, a major accomplishment of this thesis was developing a methodology for discovering, analyzing and categorizing such maneuvers. In this study, maneuvers were classified into one of four families, and further defined by energy, displacement, and other state variable constraints. (On the latter, see the discussion starting on page 65.) After a minimal outline by the user of initial and final conditions, DIDO then did the work of integrating the complete time history of an optimal maneuver. The tools built into the aircraft code completed the process by thoroughly analyzing DIDO's maneuver for optimality and feasibility, and creating graphics to summarize the results.

All told, the method developed in this thesis can take a maneuver from concept to finished product in as little as four hours. Hopefully this will enable the future development of more maneuvers like the Classic Herbst, Falcon Turn and Vertical Cobra. Even more importantly, it is hoped that this tool will encourage a continued, thorough study of time-optimal air combat maneuvers.

A. FUTURE IMPROVEMENTS

As with any project this size, there are still several areas that could be improved upon. The few that will be looked at here are the aircraft models, aircraft code and software (i.e. DIDO/TOMLAB). This section will talk about those areas, make recommendations, and also discuss potential future work on this project.

To start with, there are a few minor improvements to be made to the aircraft models. Most importantly, the ailerons and rudder need to be unlinked in the UCAV-X model, and the UCAV-X needs to be run through this study's optimal maneuvers again. (In particular the turning and pointing maneuvers, which would reap the greatest improvement from the additional lateral-directional control.) Also, once that correction is made, it would be very informative to study how changes to the UCAV-X (different moments of inertia, control derivatives, etc.) affected the aircraft's performance of the four optimal maneuvers developed in this study. In fact, this technique could allow a set of optimal maneuvers to be used to "optimize" an aircraft. (Obviously a larger set of optimal maneuvers will result in a better airplane.) The HARV model, on the other hand, has no real errors to correct, but could be made less conservative with the following two changes: first, the aircraft could be flown at a lower altitude to give it a greater T/W ratio. Second, a time-limited afterburner model could be added to the aircraft's performance data.

One thing that was missing from the aircraft code was a *Validation.m* file for the HARV and UCAV-X maneuvers. While this optimality test was replaced with another good analysis tool (the control rate plots), it would be very useful to develop a version that would work for the HARV and UCAV-X. This next recommendation is geared more towards simplifying (and improving) the actual operation of the code: removing node selection from the list of user inputs. One way to accomplish this would be to use the results of this study's analysis of node selection to choose optimum initial and bootstrapped node values for the user. (Refer to the Node Selection and Bootstrapping sections starting on page 73, and Table 8 in particular.) Another possibility is that node selection could be integrated into DIDO itself; either method would alleviate the user from the trial and error process of selecting the best node values to balance run times and result quality.

The only software issue that needs to be worked out is the incompatibility between DIDO and TOMLAB, discussed in detail in Chapter VI (see page 106). At a minimum, either a newer version of DIDO or an older version of TOMLAB will be needed to fix the problem. It is also likely that some minor rewriting of the aircraft code will be necessary to properly interact with the new DIDO/TOMLAB. However, when resolved there will be numerous benefits evident in the HARV and UCAV-X results: 1) the maneuvers will correctly bootstrap to their full 80-100 node solutions, 2) propagated results will be available for those solutions, and 3) video files and plotted trajectories will smooth out.

Finally, there are two areas relating to this study that are prime candidates for future research and development: the first is an improved user interface for the aircraft code, and the second is the more ambitious design of a two-aircraft maneuver optimization code. The greatest advantage of the former is that it would make the code much more efficient. Currently, while the code does its job very well, operating it is rather cumbersome: each run requires multiple files to be copied, edited, saved or renamed. Ideally, a very basic GUI construct with a few menus and spaces for data entry could accomplish all of those tasks. That kind of single-point interface would then both save time *and* simplify operation of the aircraft code.

Designing a two-aircraft code would be the logical (albeit somewhat elusive) follow-on to what was accomplished in this thesis. The basic framework of such a code would be very similar to the aircraft code in terms of format, aircraft models, 6DOF EOM, etc. The following key differences, though, would need to be fleshed out:

1. A set of variables describing the geometry between two opposing aircraft (including appropriate limits).
2. A system of equations governing those variables, predominately defined by the state variables of both aircraft.
3. Criteria for a "kill" expressed mathematically as final conditions. (These criteria should be based on realistic weapon engagement envelopes for guns and short range missiles.)

While this short list would just be a beginning, it covers the most important aspect of developing code: problem formulation. Along with that, one of the great things about DIDO is that if your problem formulation is sound, it will translate very quickly and easily into code. In spite of this fact, there will still undoubtedly be challenges to overcome simply due to the complexity of the problem. For instance, developing a file structure that will handle not only two copies of the aircraft code, but the additional two-aircraft EOM. On a practical note, keeping straight the 40-50 variables that would be active in such a code. On a related note, finding a way to limit the solution space so that DIDO can successfully manipulate all of those variables. Then, of course, there is the "physical" consideration of what kind of computing power would be necessary to handle all the required calculations. However, the incredible potential of such a code certainly warrants a concerted attempt to make it a reality.

So in conclusion, this thesis has just begun the task of creating a catalog of time-optimal maneuvers, and more importantly, has developed the methodology and tools needed to accomplish that task. As these last three pages have shown, though, there is still room for improvement…and, of course, plenty of room for more post-stall maneuvers to be discovered.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: SAMPLE AIRCRAFT DATA

This Appendix contains the following MATLAB files: *LoadAircraft1a.m* (analytic version of the Navion), *LoadAircraft2.m* (table look-up version of the HARV), *LoadAircraft2c.m* (curve-fit version of the HARV), and *SCTables1.m* (look-up table for the Navion).

```
%Load Aircraft #1a (original version)
%Contains physical data & operating limits for the Navion
%(One of 3 load files for MAIN.M)
%=========================================================================

global CRAFT

%Sources of Aircraft Data:
%   Physical:  Schmidt, "Into to Flight Dynamics"
%   S&C Derivatives:  Schmidt, "Into to Flight Dynamics"
%   Control Limits:  estimated (ThrustTrim = Drag @ 176fps)
%   Load Factor Limits:  T-34C NATOPS
%   AOA Limits:  estimated

%Physical Data
CRAFT.Ix = 1048;                %MOI (slug-ft^2)
CRAFT.Iy = 3000;
CRAFT.Iz = 3530;
CRAFT.Ixz = 0;
CRAFT.s = 180;                  %wing reference area (ft^2)
CRAFT.b = 33.4;                 %span (ft)
CRAFT.c = 5.7;                  %chord (ft)
CRAFT.AR = CRAFT.b^2/CRAFT.s;   %aspect ratio (for reference)
CRAFT.cg = 0.29*CRAFT.c;        %c.g. location (for reference)
CRAFT.m = 85.5;                 %mass (slugs)

CRAFT.rho = 0.002377;           %density, SSL (slugs/ft^3)
CRAFT.g = 32.2;                 %gravity (ft/sec^2)

%Stability & Control Derivatives
%-------------------------------------------------------------------------
%Longitudinal
CRAFT.CD0 = 0.051;
CRAFT.CDa = 0.330;
CRAFT.CL0 = 0.415;
CRAFT.CLa = 4.44;
CRAFT.CLq = 3.8;
CRAFT.CLde = 0.355;
CRAFT.Cma = -0.683;
CRAFT.Cmadot = -4.46;   %(for reference)
CRAFT.Cmq = -9.96;
CRAFT.Cmde = -0.889;
```

```
%Lateral, Directional
CRAFT.CYb = -0.564;
CRAFT.CYdr = 0.157;
CRAFT.Clb = -0.074;
CRAFT.Clp = -0.410;
CRAFT.Clr = 0.107;
CRAFT.Clda = 0.134;
CRAFT.Cldr = 0.012;
CRAFT.Cnb = 0.070;
CRAFT.Cnp = -0.058;
CRAFT.Cnr = -0.125;
CRAFT.Cnda = -0.003;
CRAFT.Cndr = -0.072;
%-------------------------------------------------------------------------

%Limits on Controls
CRAFT.daMax = 10*DtoR;
CRAFT.deMax = 15*DtoR;
CRAFT.drMax = 15*DtoR;
CRAFT.ThrustTrim = 338;
CRAFT.ThrustMax = CRAFT.ThrustTrim/0.80;   %422 lb
CRAFT.ThrustMin = CRAFT.ThrustMax*0.70;    %295 lb

%Limits on Load Factor
CRAFT.LoadFactorMax = 4.5;
CRAFT.LoadFactorMin = -2.3;

%Limits on Angle of Attack
CRAFT.AlphaMax = 20*DtoR;
CRAFT.AlphaMin = -10*DtoR;
%=========================================================================


%Load Aircraft #2 (data table version)
%Contains physical data & operating limits for the F-18 HARV
%(One of 3 load files for MAIN.M)
%=========================================================================

global CRAFT

%Sources of Aircraft Data:
%   Physical:  NASA TP 206573
%   S&C Derivatives:  NASA TP 206539 & 206573, WVU Report, MDC A7247
%   Control Limits:  NASA TP 206573
%   Load Factor Limits:  estimated
%   AOA Limits:  based on available data range (+10, -10 deg)

%Physical Data
CRAFT.Ix = 22789;               %MOI (slug-ft^2)
CRAFT.Iy = 176809;
CRAFT.Iz = 191744;
CRAFT.Ixz = -2305;
CRAFT.s = 400;                  %wing reference area (ft^2)
CRAFT.b = 37.4;                 %span (ft)
CRAFT.c = 11.52;                %chord (ft)
CRAFT.AR = 3.5;                 %aspect ratio (for reference)
CRAFT.cg = 0.238*CRAFT.c;       %c.g. location (for reference)
CRAFT.m = 36099/32.2;           %mass (slugs)

CRAFT.rho = 0.0008907;          %density, 30k ft (slugs/ft^3)
CRAFT.g = 32.2;                 %gravity (ft/sec^2)

%Stability & Control Derivatives
%-------------------------------------------------------------------------
if exist('alpha') == 1
    SCDeriv = SCTables(alpha);
else
    alpha = 0;
    SCDeriv = SCTables(alpha);
end

%Longitudinal
CRAFT.CD0 =    SCDeriv(1,:);
CRAFT.CDa =    SCDeriv(2,:);
%CRAFT.CDq =   SCDeriv(3,:);
CRAFT.CDds =   SCDeriv(4,:);
CRAFT.CDdtf =  SCDeriv(5,:);
CRAFT.CDdsa =  SCDeriv(6,:);
CRAFT.CDdPV =  SCDeriv(7,:);
CRAFT.CL0 =    SCDeriv(8,:);
CRAFT.CLa =    SCDeriv(9,:);
%CRAFT.CLq =   SCDeriv(10,:);
CRAFT.CLds =   SCDeriv(11,:);
CRAFT.CLdtf =  SCDeriv(12,:);
CRAFT.CLdsa =  SCDeriv(13,:);
CRAFT.CLdPV =  SCDeriv(14,:);
CRAFT.Cm0 =    SCDeriv(15,:);
CRAFT.Cma =    SCDeriv(16,:);
CRAFT.Cmq =    SCDeriv(17,:);
CRAFT.Cmds =   SCDeriv(18,:);
CRAFT.Cmdtf =  SCDeriv(19,:);
CRAFT.Cmdsa =  SCDeriv(20,:);
CRAFT.CmdPV =  SCDeriv(21,:);

%Lateral, Directional
CRAFT.CY0 =    SCDeriv(22,:);
CRAFT.CYb =    SCDeriv(23,:);
CRAFT.CYda =   SCDeriv(24,:);
CRAFT.CYdr =   SCDeriv(25,:);
CRAFT.CYdds =  SCDeriv(26,:);
CRAFT.CYdYV =  SCDeriv(27,:);
CRAFT.Cl0 =    SCDeriv(28,:);
CRAFT.Clb =    SCDeriv(29,:);
CRAFT.Clp =    SCDeriv(30,:);
CRAFT.Clr =    SCDeriv(31,:);
CRAFT.Clda =   SCDeriv(32,:);
CRAFT.Cldr =   SCDeriv(33,:);
CRAFT.Cldds =  SCDeriv(34,:);
CRAFT.CldYV =  SCDeriv(35,:);
CRAFT.Cn0 =    SCDeriv(36,:);
CRAFT.Cnb =    SCDeriv(37,:);
CRAFT.Cnp =    SCDeriv(38,:);
CRAFT.Cnr =    SCDeriv(39,:);
CRAFT.Cnda =   SCDeriv(40,:);
CRAFT.Cndr =   SCDeriv(41,:);
CRAFT.Cndds =  SCDeriv(42,:);
CRAFT.CndYV =  SCDeriv(43,:);
%-------------------------------------------------------------------------

%Limits on Controls
CRAFT.dsMax = 10.5*DtoR;    % Stabilator rate limit: 40 deg/sec
CRAFT.dsMin = -24.0*DtoR;
CRAFT.dtfMax = 45*DtoR;     % TEF rate limit: 18 deg/sec
CRAFT.dtfMin = -8*DtoR;
CRAFT.dsaMax = 45*DtoR;     % Aileron rate limit: 100 deg/sec
CRAFT.dsaMin = -24*DtoR;
CRAFT.dPVMax = 70*DtoR;     % TV vane rate limit: 80 deg/sec
CRAFT.dPVMin = -70*DtoR;
CRAFT.daMax = 69*DtoR;
CRAFT.daMin = -69*DtoR;
CRAFT.drMax = 30*DtoR;      % Rudder rate limit: 82 deg/sec
CRAFT.drMin = -30*DtoR;
CRAFT.ddsMax = 34.5*DtoR;
CRAFT.ddsMin = -34.5*DtoR;
CRAFT.dYVMax = 17.5*DtoR;
CRAFT.dYVMin = -17.5*DtoR;
CRAFT.ThrustMax = 6173;     % Calculated in GASTURB @ 30k ft, M=0.4
CRAFT.ThrustMin = 4939;     % 80% of Max (~ cruise thrust)

%Limits on Load Factor
CRAFT.LoadFactorMax = 9;
CRAFT.LoadFactorMin = -6;

%Limits on Angle of Attack
CRAFT.AlphaMax = 70*DtoR;
CRAFT.AlphaMin = 0;
%=========================================================================


%Load Aircraft #2c (analytic version, curve-fit derivatives)
%Contains physical data & operating limits for the F-18 HARV
%(One of 3 load files for MAIN.M)
%=========================================================================

global CRAFT

%Sources of Aircraft Data:
%   Physical:  NASA TP 206573
%   S&C Derivatives:  NASA TP 206539 & 206573, WVU Report, MDC A7247
%   Control Limits:  NASA TP 206573
%   Load Factor Limits:  estimated
%   AOA Limits:  based on available data range (+10, -10 deg)

%Physical Data
CRAFT.Ix = 22789;               %MOI (slug-ft^2)
CRAFT.Iy = 176809;
CRAFT.Iz = 191744;
CRAFT.Ixz = -2305;
CRAFT.s = 400;                  %wing reference area (ft^2)
CRAFT.b = 37.4;                 %span (ft)
CRAFT.c = 11.52;                %chord (ft)
CRAFT.AR = 3.5;                 %aspect ratio (for reference)
CRAFT.cg = 0.238*CRAFT.c;       %c.g. location (for reference)
CRAFT.m = 36099/32.2;           %mass (slugs)

CRAFT.rho = 0.0008907;          %density, 30k ft (slugs/ft^3)
CRAFT.g = 32.2;                 %gravity (ft/sec^2)

%Stability & Control Derivatives
%-------------------------------------------------------------------------
if exist('alpha') == 1
    a = alpha*180/pi;
else a = 0;
end

%Longitudinal
CRAFT.CD0 =    1.5896e-010*a.^5 + 9.0505e-008*a.^4 - 2.5567e-005*a.^3 +...
               1.6981e-003*a.^2 + 1.5341e-003*a    + 8.8616e-003;
CRAFT.CDa =   -3.9591e-008*a.^5 + 5.5626e-006*a.^4 - 2.9322e-004*a.^3 +...
               8.6964e-003*a.^2 - 1.6503e-001*a    + 2.1904e-003;
CRAFT.CDds =  -5.6342e-007*a.^4 + 8.6895e-005*a.^3 - 4.8912e-003*a.^2 +...
               1.0646e-001*a    - 8.0641e-001;
CRAFT.CDdtf =  1.5182e-007*a.^4 - 2.5310e-005*a.^3 + 1.3188e-003*a.^2 -...
               2.1993e-002*a    + 8.1848e-002;
CRAFT.CDdsa = -1.1356e-008*a.^5 + 1.8033e-006*a.^4 - 1.0530e-004*a.^3 +...
               2.7133e-003*a.^2 - 2.7786e-002*a    - 2.5164e-004;
CRAFT.CDdPV =  1.1311e-006*a.^4 - 1.5051e-004*a.^3 + 7.0816e-003*a.^2 -...
               1.4281e-001*a    + 6.7403e-001;
CRAFT.CL0 =   -2.9961e-010*a.^5 + 1.4665e-007*a.^4 - 1.1840e-005*a.^3 -...
               8.9395e-004*a.^2 + 9.0443e-002*a    - 1.2526e-002;
CRAFT.CLa =   -3.4948e-008*a.^5 + 7.7588e-006*a.^4 - 5.6949e-004*a.^3 +...
               1.7293e-002*a.^2 - 2.8261e-001*a    + 5.1531e+000;
CRAFT.CLds =   1.0239e-007*a.^4 - 8.4156e-006*a.^3 + 2.5567e-004*a.^2 -...
               1.1848e-002*a    + 6.7073e-001;
CRAFT.CLdtf = -1.4052e-007*a.^4 + 2.3214e-005*a.^3 - 9.3371e-004*a.^2 -...
               1.1890e-002*a    + 8.0659e-001;
CRAFT.CLdsa =  1.8612e-008*a.^5 - 2.8741e-006*a.^4 + 1.5824e-004*a.^3 -...
               3.4917e-003*a.^2 + 1.3289e-002*a    + 4.5900e-001;
CRAFT.CLdPV = -1.2158e-006*a.^4 + 1.7069e-004*a.^3 - 8.1980e-003*a.^2 +...
               1.3632e-001*a    + 3.6692e-001;
CRAFT.Cm0 =    1.5777e-009*a.^5 - 3.0875e-007*a.^4 + 1.8428e-005*a.^3 -...
               3.1021e-004*a.^2 - 7.3489e-003*a    + 3.8385e-002;
CRAFT.Cma =    8.5944e-008*a.^5 - 1.3537e-005*a.^4 + 7.4951e-004*a.^3 -...
               1.7324e-002*a.^2 + 1.3970e-001*a    - 2.7842e-001;
CRAFT.Cmq =   -1.8750e-005*a.^4 + 2.9306e-003*a.^3 - 1.4146e-001*a.^2 +...
               2.3308e+000*a    - 1.5000e+001;
CRAFT.Cmds =   2.6857e-007*a.^4 - 4.9404e-005*a.^3 + 3.0389e-003*a.^2 -...
               6.7330e-002*a    - 3.5571e-001;
CRAFT.Cmdtf =  2.0054e-008*a.^5 - 3.0959e-006*a.^4 + 1.6644e-004*a.^3 -...
               3.6260e-003*a.^2 + 2.6870e-002*a    + 2.9789e-002;
CRAFT.Cmdsa = -6.6845e-009*a.^5 + 1.0830e-006*a.^4 - 6.1680e-005*a.^3 +...
               1.4031e-003*a.^2 - 7.2818e-003*a    - 1.4348e-001;
CRAFT.CmdPV =  2.2381e-007*a.^4 - 2.7819e-005*a.^3 + 1.3095e-003*a.^2 -...
               2.3187e-002*a    - 6.0280e-001;

%Lateral, Directional
CRAFT.CY0 =   -4.0135e-007*a.^4 + 5.7639e-005*a.^3 - 2.7649e-003*a.^2 +...
```

```
               5.1019e-002*a    - 2.9235e-001;
CRAFT.CYb =   -1.5357e-008*a.^5 + 2.5985e-006*a.^4 - 1.5072e-004*a.^3 +...
               3.2580e-003*a.^2 - 1.2246e-002*a    - 8.6330e-001;
CRAFT.CYda =   3.3501e-007*a.^4 - 5.7540e-005*a.^3 + 3.2434e-003*a.^2 -...
               6.4101e-002*a    + 3.1716e-001;
CRAFT.CYdr =  -1.1258e-007*a.^4 + 1.5439e-005*a.^3 - 6.1431e-004*a.^2 +...
               3.9319e-003*a    + 2.3621e-001;
CRAFT.CYdds =  3.2044e-007*a.^4 - 4.9740e-005*a.^3 + 2.4692e-003*a.^2 -...
               4.0870e-002*a    + 1.5539e-001;
CRAFT.CYdYV = -8.0260e-007*a.^4 + 1.1623e-004*a.^3 - 5.9102e-003*a.^2 +...
               1.2227e-001*a    - 1.4332e+000;
CRAFT.Cl0 =   -2.5604e-008*a.^4 + 3.9623e-006*a.^3 - 2.0410e-004*a.^2 +...
               4.0433e-003*a    - 2.3970e-002;
CRAFT.Clb =    8.2309e-008*a.^4 - 1.5772e-005*a.^3 + 1.0031e-003*a.^2 -...
               2.3417e-002*a    + 1.1850e-002;
CRAFT.Clp =   -2.5861e-009*a.^5 + 2.4347e-007*a.^4 - 5.0306e-006*a.^3 +...
               4.8376e-005*a.^2 + 8.9702e-003*a    - 5.0675e-001;
CRAFT.Clr =    1.0384e-008*a.^5 - 9.1066e-007*a.^4 - 1.7926e-005*a.^3 +...
               2.9609e-003*a.^2 - 6.5118e-002*a    + 4.1285e-001;
CRAFT.Clda =   6.4740e-008*a.^4 - 1.1321e-005*a.^3 + 7.2251e-004*a.^2 -...
               2.0491e-002*a    + 2.6838e-001;
CRAFT.Cldr =   1.0165e-009*a.^5 - 1.6145e-007*a.^4 + 8.4635e-006*a.^3 -...
               1.5815e-004*a.^2 + 2.0824e-004*a    + 2.3080e-002;
CRAFT.Cldds = -5.5665e-008*a.^5 + 9.7770e-006*a.^3 - 5.7644e-004*a.^2 +...
               1.1935e-002*a    + 6.2956e-003;
CRAFT.CldYV = -7.0767e-006*a.^4 + 9.6398e-004*a.^3 - 4.4453e-002*a.^2 +...
               7.9746e-001*a    - 4.4896e+000;
CRAFT.Cn0 =   -1.0043e-008*a.^4 + 2.5937e-006*a.^3 - 1.6377e-004*a.^2 +...
               3.5237e-003*a    - 2.1952e-002;
CRAFT.Cnb =    5.7139e-009*a.^4 + 2.9619e-006*a.^3 - 4.2322e-004*a.^2 +...
               9.7820e-003*a    + 8.6820e-003;
CRAFT.Cnp =   -2.9866e-008*a.^5 + 4.5629e-006*a.^4 - 2.2470e-004*a.^3 +...
               3.8544e-003*a.^2 - 8.4827e-003*a    - 1.9894e-001;
CRAFT.Cnr =    3.7109e-008*a.^5 - 5.2469e-006*a.^4 + 2.3384e-004*a.^3 -...
               3.1041e-003*a.^2 - 1.8686e-002*a    + 1.8305e-002;
CRAFT.Cnda =  -3.8052e-008*a.^4 + 6.0376e-006*a.^3 - 2.9961e-004*a.^2 +...
               4.7598e-003*a    - 1.5341e-002;
CRAFT.Cndr =   1.0377e-008*a.^4 - 1.2360e-006*a.^3 + 2.2668e-005*a.^2 +...
               1.6279e-003*a    - 8.3614e-002;
CRAFT.Cndds = -1.6548e-008*a.^4 + 2.6548e-006*a.^3 - 1.1075e-004*a.^2 -...
               9.3077e-004*a    + 2.3226e-002;
CRAFT.CndYV =  3.4956e-006*a.^4 - 5.5119e-004*a.^3 + 3.1614e-002*a.^2 -...
               7.6194e-001*a    + 1.7877e+001;
%--------------------------------------------------------------------------

%Limits on Controls
CRAFT.dsMax = 10.5*DtoR;    % Stabilator rate limit: 40 deg/sec
CRAFT.dsMin = -24.0*DtoR;
CRAFT.dtfMax = 45*DtoR;     % TEF rate limit: 18 deg/sec
CRAFT.dtfMin = -8*DtoR;
CRAFT.dsaMax = 45*DtoR;     % Aileron rate limit: 100 deg/sec
CRAFT.dsaMin = -24*DtoR;
CRAFT.dPVMax = 70*DtoR;     % TV vane rate limit: 80 deg/sec
CRAFT.dPVMin = -70*DtoR;
CRAFT.daMax = 69*DtoR;
CRAFT.daMin = -69*DtoR;
CRAFT.drMax = 30*DtoR;      % Rudder rate limit: 82 deg/sec
CRAFT.drMin = -30*DtoR;
CRAFT.ddsMax = 34.5*DtoR;
CRAFT.ddsMin = -34.5*DtoR;
CRAFT.dYVMax = 17.5*DtoR;
CRAFT.dYVMin = -17.5*DtoR;
CRAFT.ThrustMax = 6173;     % Calculated in GASTURB @ 30k ft, M=0.4
CRAFT.ThrustMin = 4939;     % 80% of Max (~ cruise thrust)

%Limits on Load Factor
CRAFT.LoadFactorMax = 9;
CRAFT.LoadFactorMin = -6;

%Limits on Angle of Attack
CRAFT.AlphaMax = 70*DtoR;
CRAFT.AlphaMin = 0;
%=========================================================================


function [SCDeriv] = SCTables(alpha)
%Stability & Control Derivative calculations for LoadAircraft1.m
%Contains look-up tables for the Navion as a proof of concept
%=========================================================================

alpha = alpha * 180/pi;   %(Tables list derivatives vs deg)
x = length(alpha);
SCDeriv = zeros(21,x);

%Longitudinal
%Control Surfaces: elevator(e)
%-------------------------------------------------------------------------
ar1 = 10:10:60;   %Alpha range 1

CD0         =   [ 0.051  0.051  0.051  0.051  0.051  0.051];
CDa         =   [ 0.33   0.33   0.33   0.33   0.33   0.33 ];

CL0         =   [ 0.415  0.415  0.415  0.415  0.415  0.415];
CLa         =   [ 4.44   4.44   4.44   4.44   4.44   4.44 ];
CLq         =   [ 3.8    3.8    3.8    3.8    3.8    3.8  ];
CLde        =   [ 0.355  0.355  0.355  0.355  0.355  0.355];

Cma         =   [-0.683 -0.683 -0.683 -0.683 -0.683 -0.683];
Cmq         =   [-9.96  -9.96  -9.96  -9.96  -9.96  -9.96 ];
Cmde        =   [-0.889 -0.889 -0.889 -0.889 -0.889 -0.889];

SCDeriv(1,1:x)  =   interp1(ar1,CD0,alpha(1:x),'cubic','extrap');
SCDeriv(2,1:x)  =   interp1(ar1,CDa,alpha(1:x),'cubic','extrap');
SCDeriv(3,1:x)  =   interp1(ar1,CL0,alpha(1:x),'cubic','extrap');
SCDeriv(4,1:x)  =   interp1(ar1,CLa,alpha(1:x),'cubic','extrap');
SCDeriv(5,1:x)  =   interp1(ar1,CLq,alpha(1:x),'cubic','extrap');
SCDeriv(6,1:x)  =   interp1(ar1,CLde,alpha(1:x),'cubic','extrap');
SCDeriv(7,1:x)  =   interp1(ar1,Cma,alpha(1:x),'cubic','extrap');
SCDeriv(8,1:x)  =   interp1(ar1,Cmq,alpha(1:x),'cubic','extrap');
SCDeriv(9,1:x)  =   interp1(ar1,Cmde,alpha(1:x),'cubic','extrap');
%-------------------------------------------------------------------------


%Lateral, Directional
```

```
%Control Surfaces: ailerons (a), rudder (r)
%-------------------------------------------------------------------------
ar2 = [0:10:50];   %Alpha range 2

CYb         =   [-0.564 -0.564 -0.564 -0.564 -0.564 -0.564];
CYdr        =   [ 0.157  0.157  0.157  0.157  0.157  0.157];

Clb         =   [-0.074 -0.074 -0.074 -0.074 -0.074 -0.074];
Clp         =   [-0.41  -0.41  -0.41  -0.41  -0.41  -0.41 ];
Clr         =   [ 0.107  0.107  0.107  0.107  0.107  0.107];
Clda        =   [ 0.134  0.134  0.134  0.134  0.134  0.134];
Cldr        =   [ 0.012  0.012  0.012  0.012  0.012  0.012];

Cnb         =   [ 0.07   0.07   0.07   0.07   0.07   0.07 ];
Cnp         =   [-0.058 -0.058 -0.058 -0.058 -0.058 -0.058];
Cnr         =   [-0.125 -0.125 -0.125 -0.125 -0.125 -0.125];
Cnda        =   [-0.003 -0.003 -0.003 -0.003 -0.003 -0.003];
Cndr        =   [-0.072 -0.072 -0.072 -0.072 -0.072 -0.072];

SCDeriv(10,1:x) =   interp1(ar2,CYb,alpha(1:x),'cubic','extrap');
SCDeriv(11,1:x) =   interp1(ar2,CYdr,alpha(1:x),'cubic','extrap');
SCDeriv(12,1:x) =   interp1(ar2,Clb,alpha(1:x),'cubic','extrap');
SCDeriv(13,1:x) =   interp1(ar2,Clp,alpha(1:x),'cubic','extrap');
SCDeriv(14,1:x) =   interp1(ar2,Clr,alpha(1:x),'cubic','extrap');
SCDeriv(15,1:x) =   interp1(ar2,Clda,alpha(1:x),'cubic','extrap');
SCDeriv(16,1:x) =   interp1(ar2,Cldr,alpha(1:x),'cubic','extrap');
SCDeriv(17,1:x) =   interp1(ar2,Cnb,alpha(1:x),'cubic','extrap');
SCDeriv(18,1:x) =   interp1(ar2,Cnp,alpha(1:x),'cubic','extrap');
SCDeriv(19,1:x) =   interp1(ar2,Cnr,alpha(1:x),'cubic','extrap');
SCDeriv(20,1:x) =   interp1(ar2,Cnda,alpha(1:x),'cubic','extrap');
SCDeriv(21,1:x) =   interp1(ar2,Cndr,alpha(1:x),'cubic','extrap');
%-------------------------------------------------------------------------
```

APPENDIX B: SAMPLE MANEUVERS

This Appendix contains the following MATLAB files: *LoadManeuver2.m* (load file for the Wingover – only the Navion version), and *LoadManeuver01.m* (load file for the Reversal – both the Navion and HARV versions).

```
%Load Maneuver #2 (Navion)
%Contains the initial, final & boundary conditions for a "Wingover"
%(Delta: Y[1000 ft], psi[180 deg])
%(One of 3 load files for MAIN.M)
%=========================================================================

global INIT FINAL

%Initial & Final Condition Terms:
%   Position (X & Y-coordinates, altitude)
%   Velocity (true airspeed, angle of attack, sideslip angle)
%   Body rates (roll, pitch & yaw rates)
%   Euler angles (phi, theta, psi)

%Initial Conditions
INIT.X = 0;
INIT.Y = 0;
INIT.H = 1000;
INIT.V = 176;
INIT.alpha = 0;
INIT.beta = 0;
INIT.p = 0;
INIT.q = 0;
INIT.r = 0;
INIT.phi = 0;
INIT.theta = 0;
INIT.psi = 0;

%Final Conditions
FINAL.X = 0;
FINAL.Y = 1000;
FINAL.H = 1000;
FINAL.V = 176;
FINAL.alpha = 0;
FINAL.beta = 0;
FINAL.p = 0;
FINAL.q = 0;
FINAL.r = 0;
FINAL.phi = 0;
FINAL.theta = 0;
FINAL.psi = pi;

tfGuess = 15; %seconds
```

```
%Boundary Conditions
MyEventBounds = [
        0, 0;                                       %X0 bounds
        0, 0;                                       %Y0 bounds
        INIT.H/UNITS.dist, INIT.H/UNITS.dist;       %H0 bounds

        INIT.V/UNITS.vel, INIT.V/UNITS.vel;         %V0 bounds
        0, 0;                                       %alpha0 bounds
        0, 0;                                       %beta0 bounds

        0, 0;                                       %p0 bounds
        0, 0;                                       %q0 bounds
        0, 0;                                       %r0 bounds

        0, 0;                                       %phi0 bounds
        0, 0;                                       %theta0 bounds
        0, 0;                                       %psi0 bounds

        -100/UNITS.dist, 100/UNITS.dist;            %Xf bounds
        900/UNITS.dist, 1100/UNITS.dist;            %Yf bounds
        FINAL.H/UNITS.dist, FINAL.H/UNITS.dist;     %Hf bounds

        FINAL.V/UNITS.vel, FINAL.V/UNITS.vel;       %Vf bounds
        0, 0;                                       %alphaf bounds
        0, 0;                                       %betaf bounds

        (-pi/90)*UNITS.time, (pi/90)*UNITS.time;    %pf bounds
        (-pi/90)*UNITS.time, (pi/90)*UNITS.time;    %qf bounds
        (-pi/90)*UNITS.time, (pi/90)*UNITS.time;    %rf bounds

        0, 0;                                       %phif bounds
        0, 0;                                       %thetaf bounds
        FINAL.psi, FINAL.psi;                       %psif bounds
                                 ];
%==========================================================================


%Load Maneuver #01 (Navion)
%Contains the initial, final & boundary conditions for a "Reversal"
%(Delta: X,Y,H[0 ft], V[0 fps], psi[180 deg])
%(One of 3 load files for MAIN.M)
%==========================================================================

global INIT FINAL

%Initial & Final Condition Terms:
%   Position (X & Y-coordinates, altitude)
%   Velocity (true airspeed, angle of attack, sideslip angle)
%   Body rates (roll, pitch & yaw rates)
%   Euler angles (phi, theta, psi)

%Initial Conditions
INIT.X = 0;
INIT.Y = 0;
INIT.H = 1000;
INIT.V = 176;
INIT.alpha = 0;
INIT.beta = 0;
INIT.p = 0;
INIT.q = 0;
INIT.r = 0;
INIT.phi = 0;
INIT.theta = 0;
INIT.psi = 0;

%Final Conditions
FINAL.X = 0;
FINAL.Y = 0;
FINAL.H = 1000;
FINAL.V = 176;
FINAL.alpha = 0;
FINAL.beta = 0;
FINAL.p = 0;
FINAL.q = 0;
FINAL.r = 0;
FINAL.phi = 0;
FINAL.theta = 0;
FINAL.psi = pi;

tfGuess = 20; %seconds

%Boundary Conditions
MyEventBounds = [
        0, 0;                                    %X0 bounds
        0, 0;                                    %Y0 bounds
        INIT.H/UNITS.dist, INIT.H/UNITS.dist;    %H0 bounds

        INIT.V/UNITS.vel, INIT.V/UNITS.vel;      %V0 bounds
        0, 0;                                    %alpha0 bounds
        0, 0;                                    %beta0 bounds

        0, 0;                                    %p0 bounds
        0, 0;                                    %q0 bounds
        0, 0;                                    %r0 bounds

        0, 0;                                    %phi0 bounds
        0, 0;                                    %theta0 bounds
        0, 0;                                    %psi0 bounds

        0/UNITS.dist, 0/UNITS.dist;              %Xf bounds
        0/UNITS.dist, 0/UNITS.dist;              %Yf bounds
        FINAL.H/UNITS.dist, FINAL.H/UNITS.dist;  %Hf bounds

        FINAL.V/UNITS.vel, FINAL.V/UNITS.vel;    %Vf bounds
        0, 0;                                    %alphaf bounds
        0, 0;                                    %betaf bounds

        0, 0;                                    %pf bounds
        0, 0;                                    %qf bounds
        0, 0;                                    %rf bounds

        0, 0;                                    %phif bounds
        0, 0;                                    %thetaf bounds
        FINAL.psi, FINAL.psi;                    %psif bounds
```

```
                                 ];
%=========================================================================


%Load Maneuver #01 (HARV)
%Contains the initial, final & boundary conditions for a "Reversal"
%(Delta: X,Y,H[0 ft], V[0 fps], psi[180 deg])
%(One of 3 load files for MAIN.M)
%=========================================================================

global INIT FINAL

%Initial & Final Condition Terms:
%   Position (X & Y-coordinates, altitude)
%   Velocity (true airspeed, angle of attack, sideslip angle)
%   Body rates (roll, pitch & yaw rates)
%   Euler angles (phi, theta, psi)
%   Controls (listed in Boundary Conditions)

%Initial Conditions
INIT.X = 0;
INIT.Y = 0;
INIT.H = 30000;
INIT.V = 398;
INIT.alpha = 0;
INIT.beta = 0;
INIT.p = 0;
INIT.q = 0;
INIT.r = 0;
INIT.phi = 0;
INIT.theta = 0;
INIT.psi = 0;

%Final Conditions
FINAL.X = 0;
FINAL.Y = 0;
FINAL.H = 30000;
FINAL.V = 398;
FINAL.alpha = 0;
FINAL.beta = 0;
FINAL.p = 0;
FINAL.q = 0;
FINAL.r = 0;
FINAL.phi = 0;
FINAL.theta = 0;
FINAL.psi = pi;

tfGuess = 20; %seconds

%Boundary Conditions
MyEventBounds = [
        0, 0;                                       %X0 bounds
        0, 0;                                       %Y0 bounds
        INIT.H/UNITS.dist, INIT.H/UNITS.dist;       %H0 bounds

        INIT.V/UNITS.vel, INIT.V/UNITS.vel;         %V0 bounds
        0, 0;                                       %alpha0 bounds
        0, 0;                                       %beta0 bounds

        0, 0;                                       %p0 bounds
        0, 0;                                       %q0 bounds
        0, 0;                                       %r0 bounds

        0, 0;                                       %phi0 bounds
        0, 0;                                       %theta0 bounds
        0, 0;                                       %psi0 bounds

        0, 0;                                       %ds0 bounds
        0, 0;                                       %dtf0 bounds
        0, 0;                                       %dsa0 bounds
        0, 0;                                       %dPV0 bounds
        0, 0;                                       %da0 bounds
        0, 0;                                       %dr0 bounds
        0, 0;                                       %dds0 bounds
        0, 0;                                       %dYV0 bounds

        0/UNITS.dist, 0/UNITS.dist;                 %Xf bounds
        0/UNITS.dist, 0/UNITS.dist;                 %Yf bounds
        FINAL.H/UNITS.dist, FINAL.H/UNITS.dist;     %Hf bounds

        FINAL.V/UNITS.vel, FINAL.V/UNITS.vel;       %Vf bounds
        0, 0;                                       %alphaf bounds
        0, 0;                                       %betaf bounds

        0, 0;                                       %pf bounds
        0, 0;                                       %qf bounds
        0, 0;                                       %rf bounds

        0, 0;                                       %phif bounds
        0, 0;                                       %thetaf bounds
        FINAL.psi, FINAL.psi;                       %psif bounds

        -24*(pi/180), 10.5*(pi/180);                %dsf bounds
        -8*(pi/180), 45*(pi/180);                   %dtff bounds
        -24*(pi/180), 45*(pi/180);                  %dsaf bounds
        -70*(pi/180), 70*(pi/180);                  %dPVf bounds
        -69*(pi/180), 69*(pi/180);                  %daf bounds
        -30*(pi/180), 30*(pi/180);                  %drf bounds
        -34.5*(pi/180), 34.5*(pi/180);              %ddsf bounds
        -17.5*(pi/180), 17.5*(pi/180);              %dYVf bounds
                                         ];
%==========================================================================
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: DIDO RUN HISTORY

NAVION // Standard Maneuvers // Wingover (LoadManeuver2.m)

DELTA ΔY = 1000 ft
$\Delta\psi$ = 180°

1.	Constrained:	$\alpha, \beta, p, q, r, \phi, \theta = 0$	$V = V(0)$	a)	5.7366 min	40 iter	fair
		$\psi = 180°$		b)	5.440 min	40 iter	fair
	Unconstrained:	X, Y (500 ft)	H (100ft)	c)	5.4572 min	40 iter	fair
2.	Constrained:	$\alpha, \beta, p, q, r, \phi, \theta = 0$	$V = V(0)$	a)	4.7843 min	38 iter	good
		$\psi = 180°$		b)	4.8100 min	38 iter	good
	Unconstrained:	X, Y (100 ft)	H (100ft)				
3.	Constrained:	$\alpha, \beta, p, q, r, \phi, \theta = 0$	$V = V(0)$	a)	2.9611 min	23 iter	good
		$\psi = 180°$	$H = H(0)$	b)	3.0125 min	23 iter	good
	Unconstrained:	X, Y (100 ft)					
4.	Constrained:	$\alpha, \beta, p, q, r = 0$	$V = V(0)$	a)	4.0185 min	31 iter	good
		$\psi = 180°$	$H = H(0)$	b)	3.9395 min	31 iter	good
	Unconstrained:	X, Y (100 ft)	ϕ, θ (2°)				
5.	Constrained:	$p, q, r, \phi, \theta = 0$	$V = V(0)$	a)	4.1448 min	32 iter	good
		$\psi = 180°$	$H = H(0)$	b)	4.1044 min	32 iter	good
	Unconstrained:	X, Y (100 ft)	α, β (2°)				
6.	Constrained:	$\alpha, \beta, \phi, \theta = 0$	$V = V(0)$	a)	3.1140 min	23 iter	great
		$\psi = 180°$	$H = H(0)$	b)	3.0871 min	23 iter	great
	Unconstrained:	X, Y (100 ft)	p, q, r (2°/sec)				

FINAL X (100 ft)
Y (900–1000 ft)
p, q, r (2°/sec)

NAVION // Standard & Optimal Maneuvers // **Control Rate Limits**

	Standard Navion model*			Navion w/ rate control limits*		
	(maneuver)	(min)	(#iter)	(maneuver)	(min)	(#iter)
Standard Maneuvers						
	1	0.56	348	1	0.74	353
	2	0.87	879	2	1.35	1176
	3	0.99	901	3	1.14	1818
	4	0.99	636	4	1.27	660
Optimal Maneuvers						
	5	2.40	2427	5	1.56	3053
	6	3.29	4056	6	1.21	3804
	7	5.13	5375	7	1.14	2419
	8	8.08	8656	8	4.58	5617

Trends: 1. 4 out of 4 standard maneuvers increased in run times and total iterations by using control rate limits.

2. 3 out of 4 optimal maneuvers decreased in run times and total iterations by using control rate limits.

Correlation between changes in iterations & run time? Not directly.

	(#iter)	(run time, min)
Maneuver #7:	2419/5375 = **0.45**	1.14/5.13 = **0.22**
Maneuver #8:	5617/8656 = **0.65**	4.58/8.08 = *0.57*
Maneuver #6:	3804/4056 = **0.94**	1.21/3.29 = **0.37**
Maneuver #5:	3053/2427 = **1.26**	1.56/2.40 = **0.65**
Maneuver #1:	353/ 348 = **1.01**	0.74/0.56 = **1.32**
Maneuver #4:	660/ 636 = **1.04**	1.27/0.99 = *1.28*
Maneuver #2:	1176/ 879 = **1.34**	1.35/0.87 = **1.55**
Maneuver #3:	1818/ 901 = **2.02**	1.14/0.99 = *1.15*

* *All of these runs were done with only 20 nodes. The only variable (for each Navion model) was the maneuver being performed.*

NAVION // Standard Maneuvers // **Nodes**

1. Straight Climb

	70 Nodes	45 major iter.	2,877 total iter.	6.3245 min	Good.
--	**60 Nodes**	48 major iter.	2,531 total iter.	4.5325 min	Great.
	50 Nodes	51 major iter.	1,362 total iter.	3.2804 min	High H @ ends.
	40 Nodes	45 major iter.	1,231 total iter.	2.0847 min	Jagged β, q, θ, n.

2. Wingover

	70 Nodes	31 major iter.	4,350 total iter.	6.0229 min	Ok.
--	**60 Nodes**	34 major iter.	4,375 total iter.	4.2166 min	Ok.
	50 Nodes	34 major iter.	5,634 total iter.	3.2874 min	Jagged controls.
	40 Nodes	70 major iter.	5,168 total iter.	3.9821 min	Jagged.

3. Level Turn

	90 Nodes	34 major iter.	9,318 total iter.	17.8278 min	Good.
--	80 Nodes	65 major iter.	2,450 total iter.	21.9902 min	Ok.
	70 Nodes	77 major iter.	3,375 total iter.	11.1024 min	Ok.
	60 Nodes	60 major iter.	3,446 total iter.	6.7112 min	Jagged H.

4. Climbing Turn

	110 Nodes	64 major iter.	10,076 total iter.	36.0996 min	Good.
--	100 Nodes	43 major iter.	4,601 total iter.	18.6036 min	Bad H.
	90 Nodes	47 major iter.	5,126 total iter.	14.9884 min	Ok.
	80 Nodes	67 major iter.	3,358 total iter.	13.3034 min	Prop. depart.

---

* Initial estimates were done with the number of nodes marked by the dash.

* The node values in bold were selected for each maneuver as the best balance of run time and result quality.

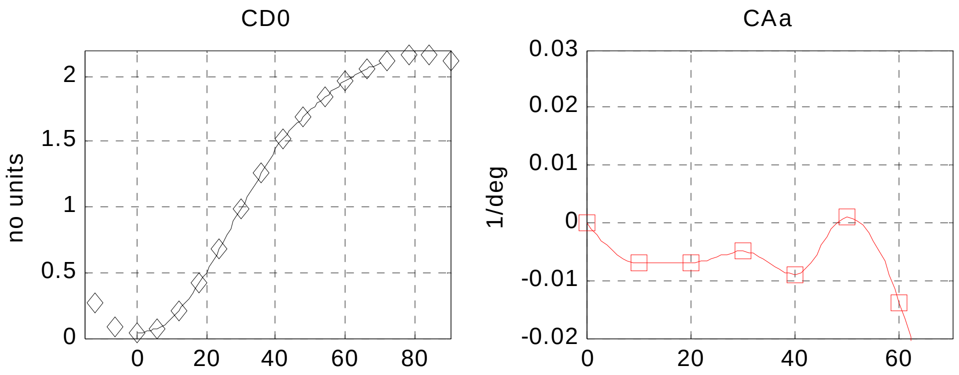
* The underlined node values displayed the best results for each maneuver within the range of node values that were tested. (Best result is based on a smooth flight path, propagated results that match DIDO's results, and a flat Hamiltonian.)

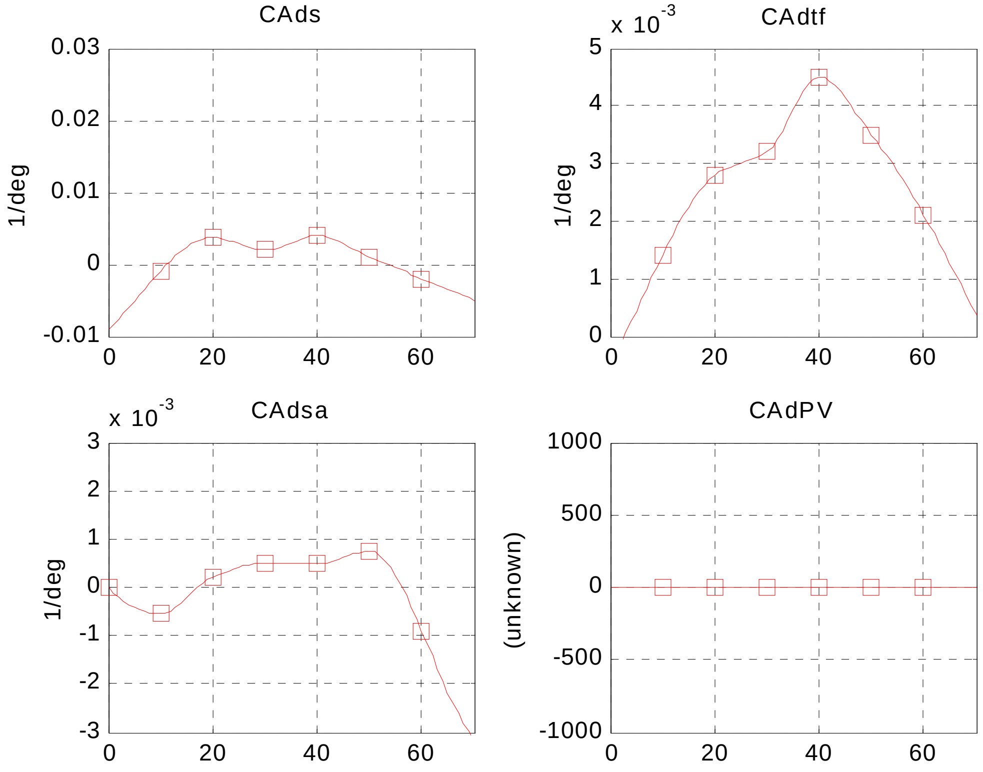
THIS PAGE INTENTIONALLY LEFT BLANK

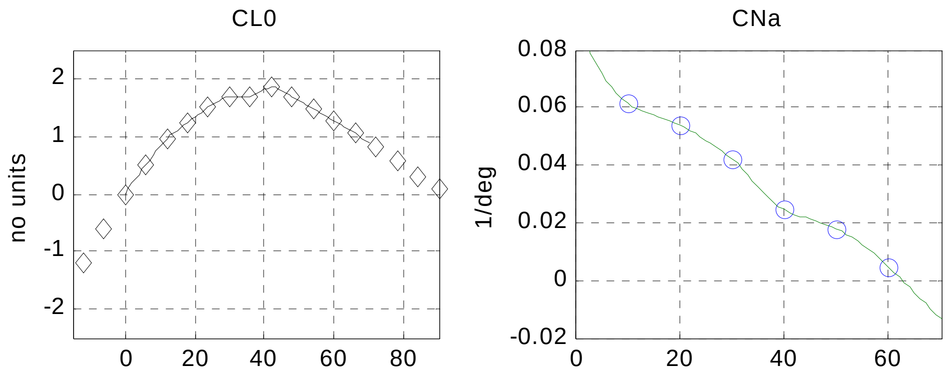
APPENDIX D: TABLE LOOK-UP PLOTS

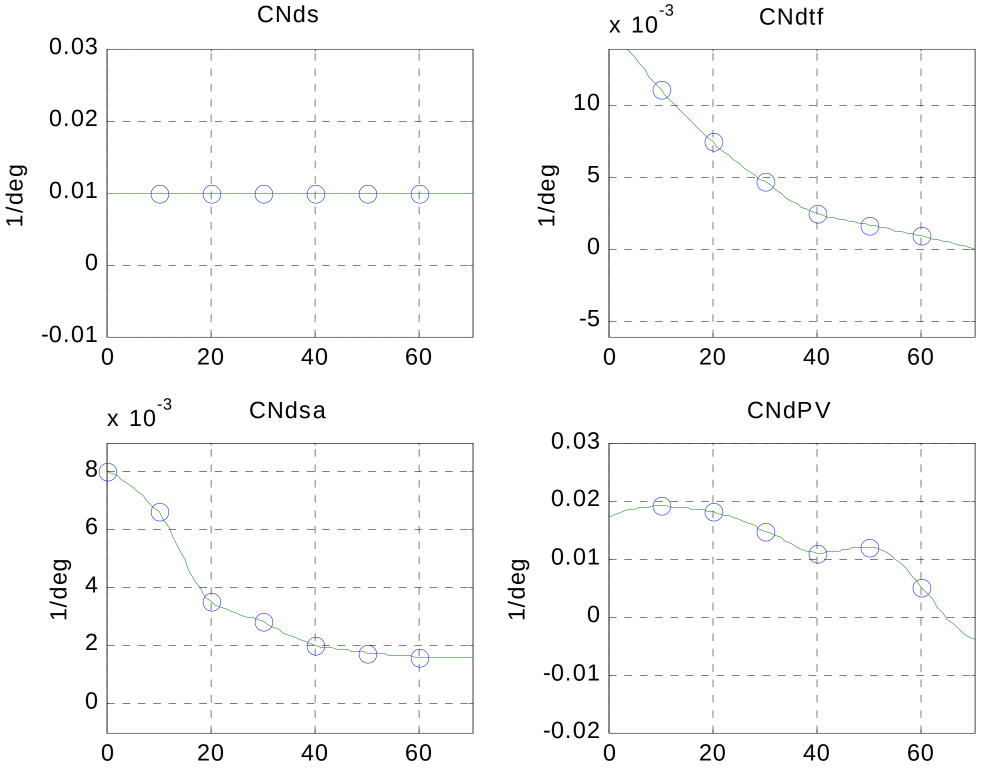
The first part of this Appendix contains plots of the HARV S&C derivatives calculated in *SCTables2.m*. The circles are actual flight test data points, while the lines are the values calculated by MATLAB's "interp1" and "extrap" commands. Each derivative is plotted versus AOA (measured in degrees along the x-axis). S&C derivatives are grouped by force and moment coefficients. (For an explanation of the axial and normal force derivatives, see the discussion of the WVU report on page 47.) Note the units label on the y-axes, as some S&C derivatives are plotted per degree and others are plotted per radian.

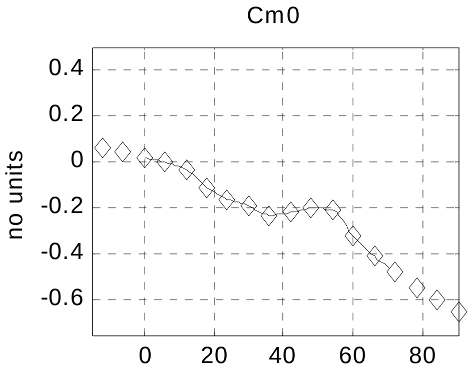
The second part of this Appendix contains summary plots of the HARV S&C derivatives that compare the relative influence that each derivative has on its respective force or moment coefficient. (These plots, as with the previous plots, were created by the script file *SCPlots2.m*.) For these plots, the axial and normal derivatives were converted to drag and lift derivatives, as those were the terms used in the aircraft code. Note that certain derivatives (in particular the baseline values) had to be scaled down to fit on the plots with the other derivatives.

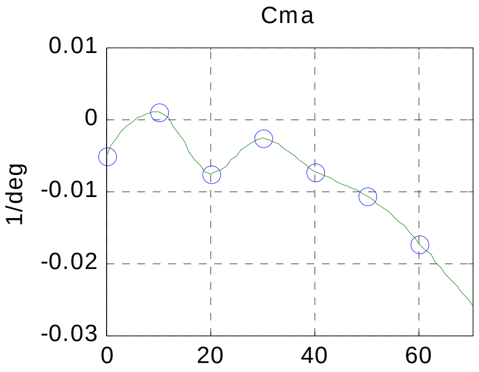

CD0
no units
2
1.5
1
0.5
0
0
20
40
60
80
CAa
1/deg
0.03
0.02
0.01
0
-0.01
-0.02
0
20
40
60

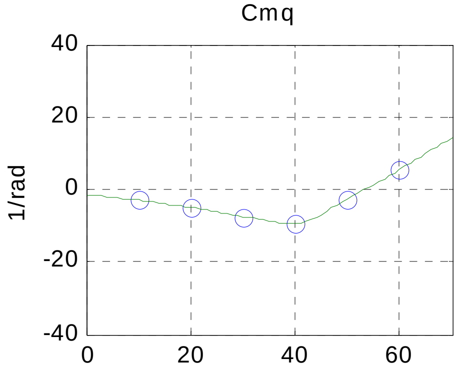

CAds
1/deg
0.03
0.02
0.01
0
-0.01
0
20
40
60
x 10^-3
CAdtf
1/deg
5
4
3
2
1
0
0
20
40
60
x 10^-3
CAdsa
1/deg
3
2
1
0
-1
-2
-3
0
20
40
60
CAdPV
(unknown)
1000
500
0
-500
-1000
0
20
40
60

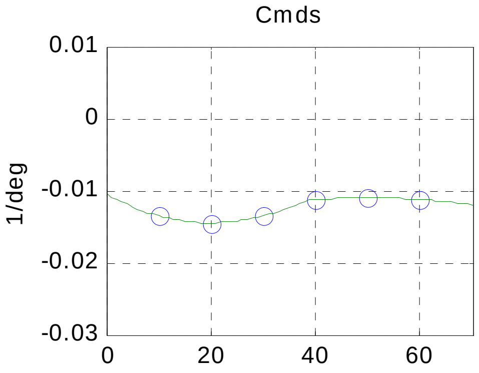

CL0
no units
2
1
0
-1
-2
0
20
40
60
80
CNa
1/deg
0.08
0.06
0.04
0.02
0
-0.02
0
20
40
60

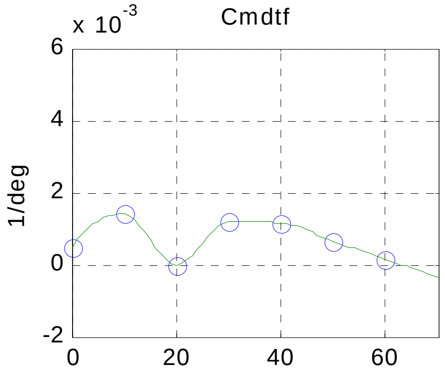

CNds
1/deg
0.03
0.02
0.01
0
-0.01
0
20
40
60
x 10^{-3}
CNdtf
1/deg
10
5
0
-5
0
20
40
60
x 10^{-3}
CNdsa
1/deg
8
6
4
2
0
0
20
40
60
CNdPV
1/deg
0.03
0.02
0.01
0
-0.01
-0.02
0
20
40
60

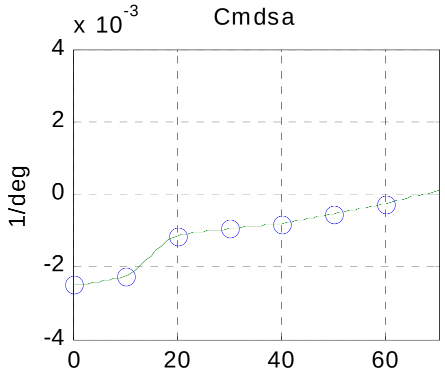

Cm0
no units
0.4
0.2
0
-0.2
-0.4
-0.6
0
20
40
60
80

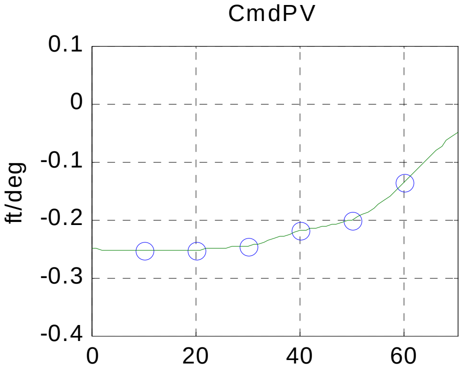

Cma
1/deg
0.01
0
-0.01
-0.02
-0.03
0
20
40
60

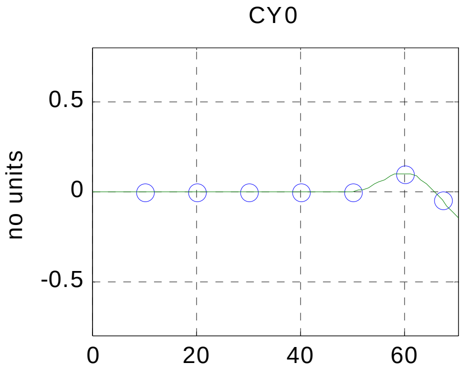

Cmq
1/rad
40
20
0
-20
-40
0
20
40
60

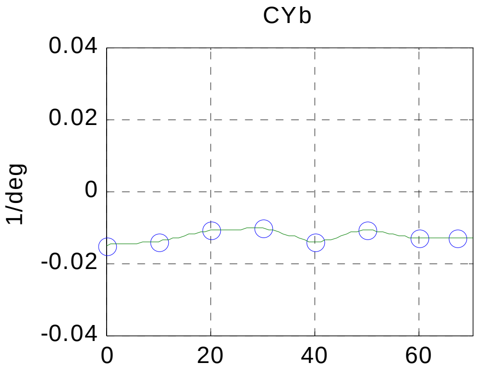

Cmds
1/deg
0.01
0
-0.01
-0.02
-0.03
0
20
40
60

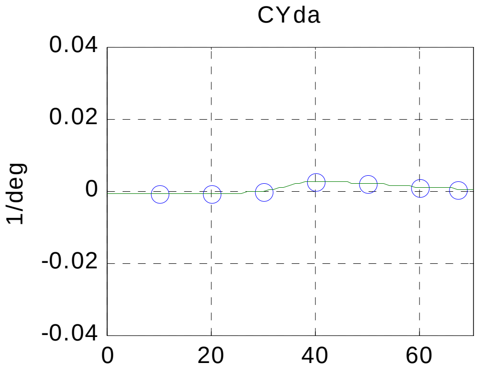

Cmdtf
x 10^-3
1/deg
6
4
2
0
-2
0
20
40
60

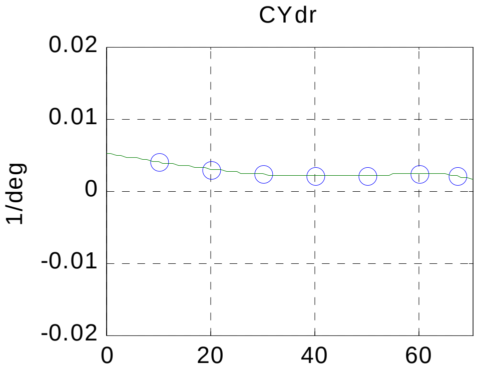

Cmdsa
x 10^-3
1/deg
4
2
0
-2
-4
0
20
40
60

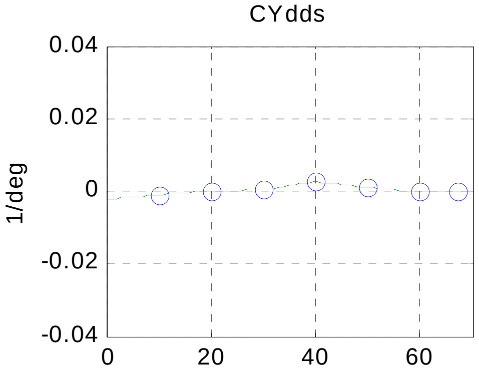

CmdPV
ft/deg
0.1
0
-0.1
-0.2
-0.3
-0.4
0
20
40
60

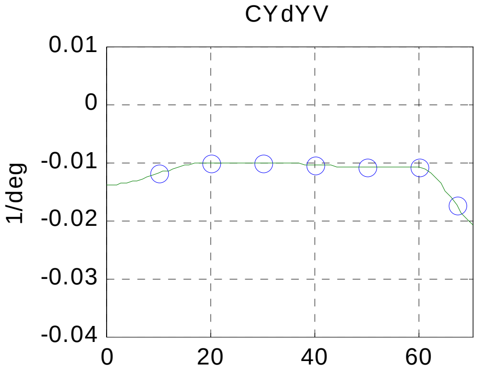

CY0
no units
0.5
0
-0.5
0
20
40
60

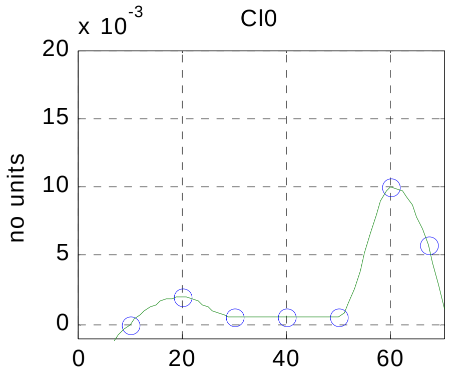

CYb
1/deg
0.04
0.02
0
-0.02
-0.04
0
20
40
60

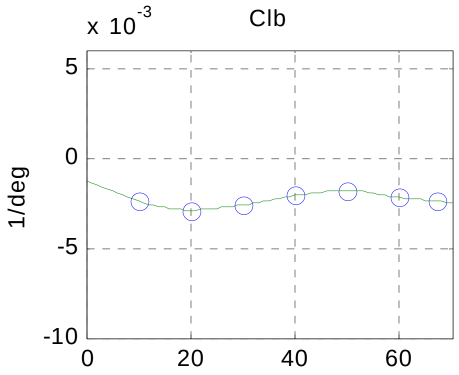

CYda
1/deg
0.04
0.02
0
-0.02
-0.04
0
20
40
60

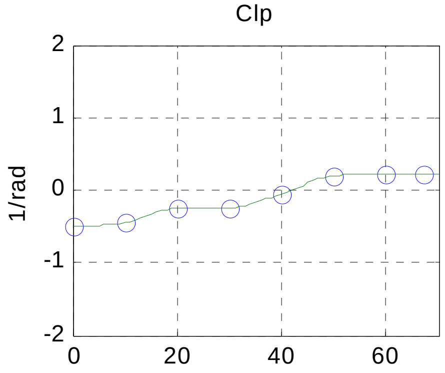

CYdr
1/deg
0.02
0.01
0
-0.01
-0.02
0
20
40
60

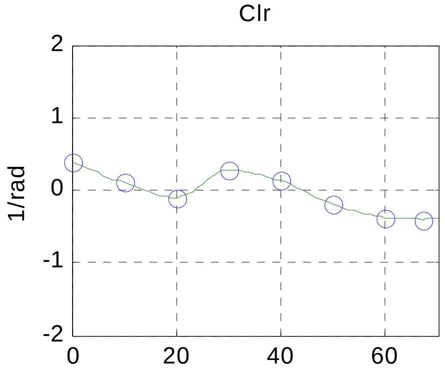

CYdds
1/deg
0.04
0.02
0
-0.02
-0.04
0
20
40
60

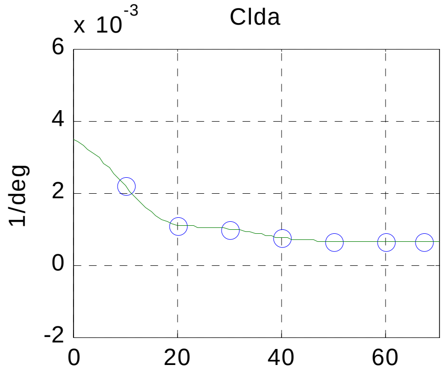

CYdYV
1/deg
0.01
0
-0.01
-0.02
-0.03
-0.04
0
20
40
60

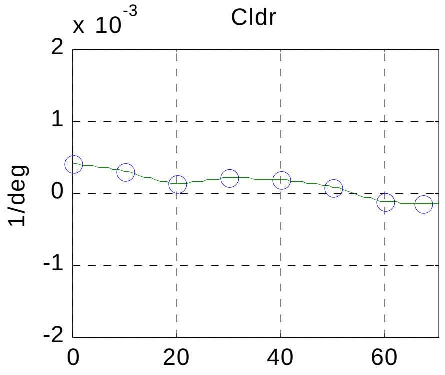

Cl0

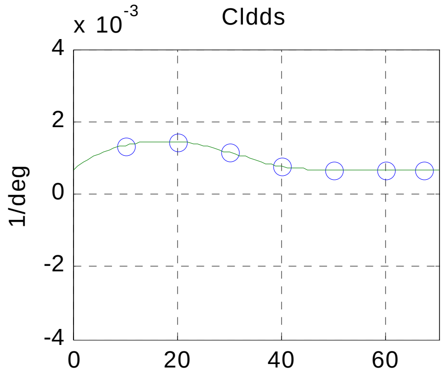

Clb

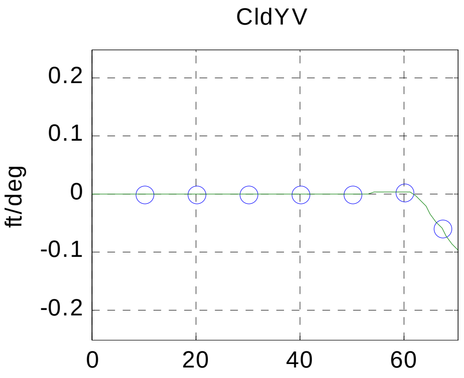

Clp

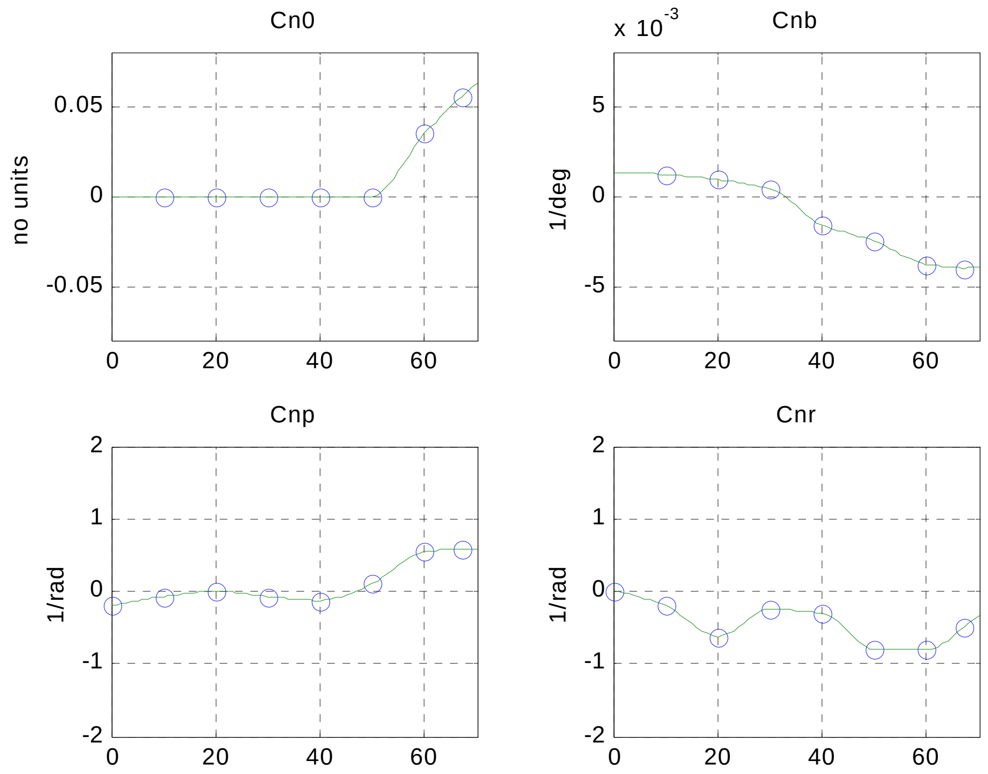

Clr

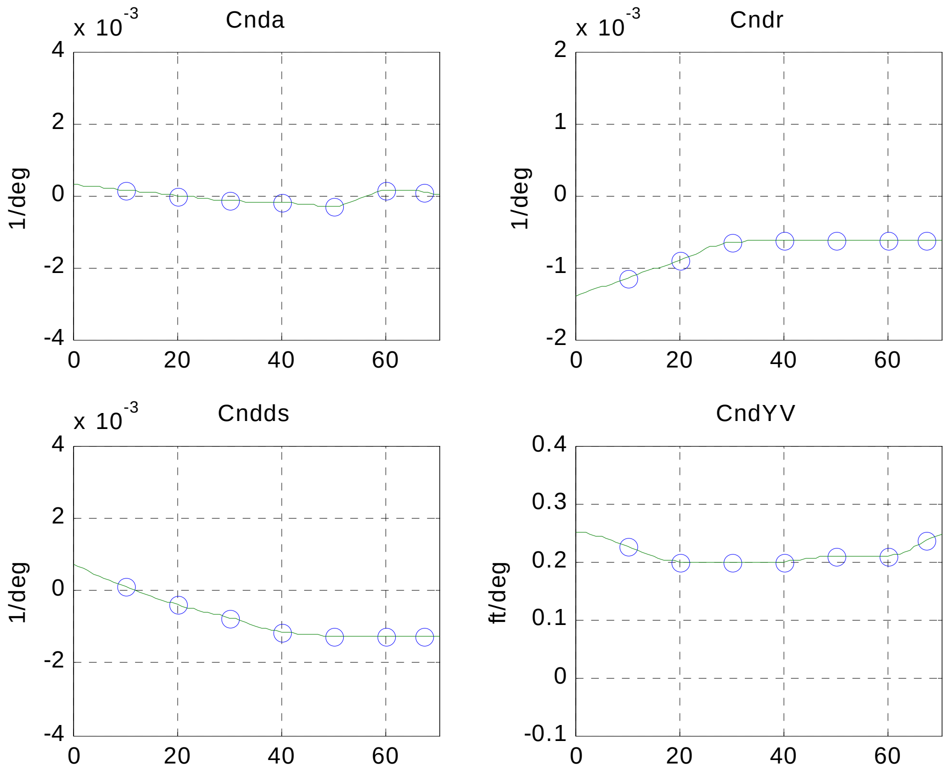

Clda

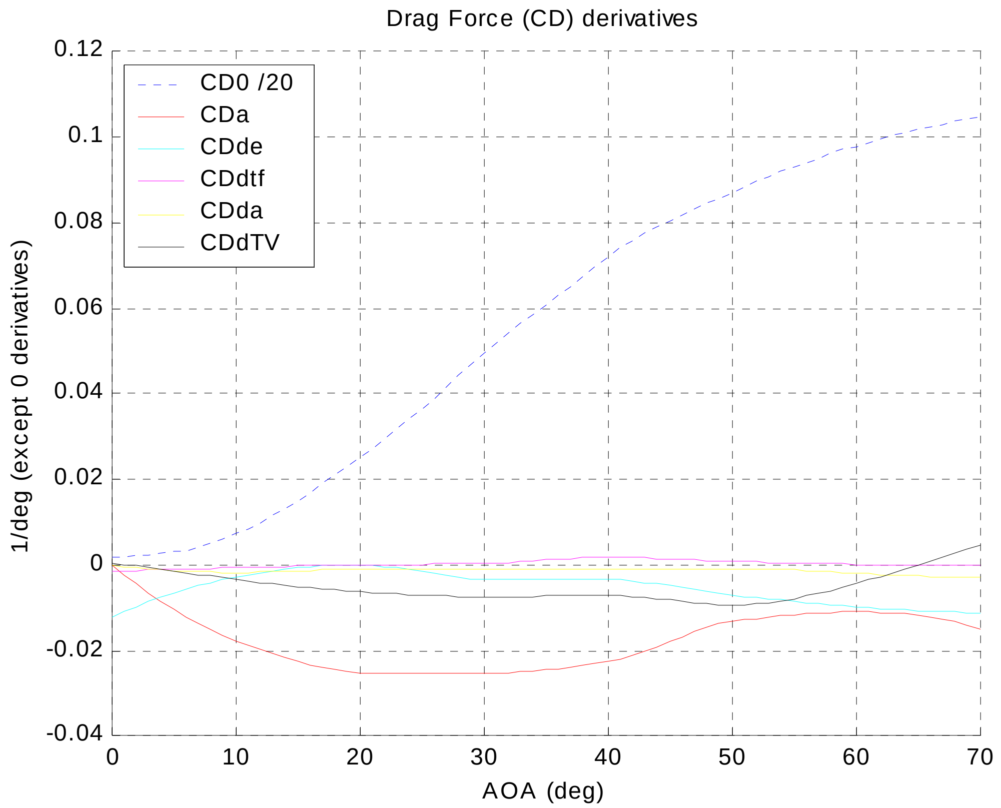

Cldr

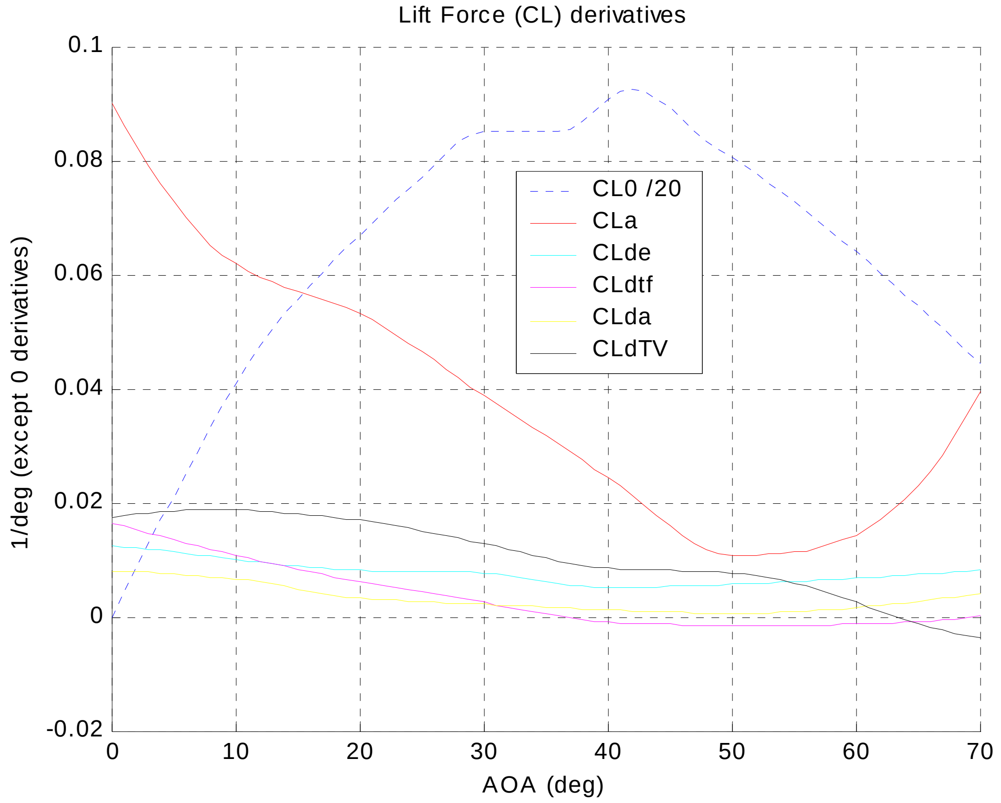

Cldds

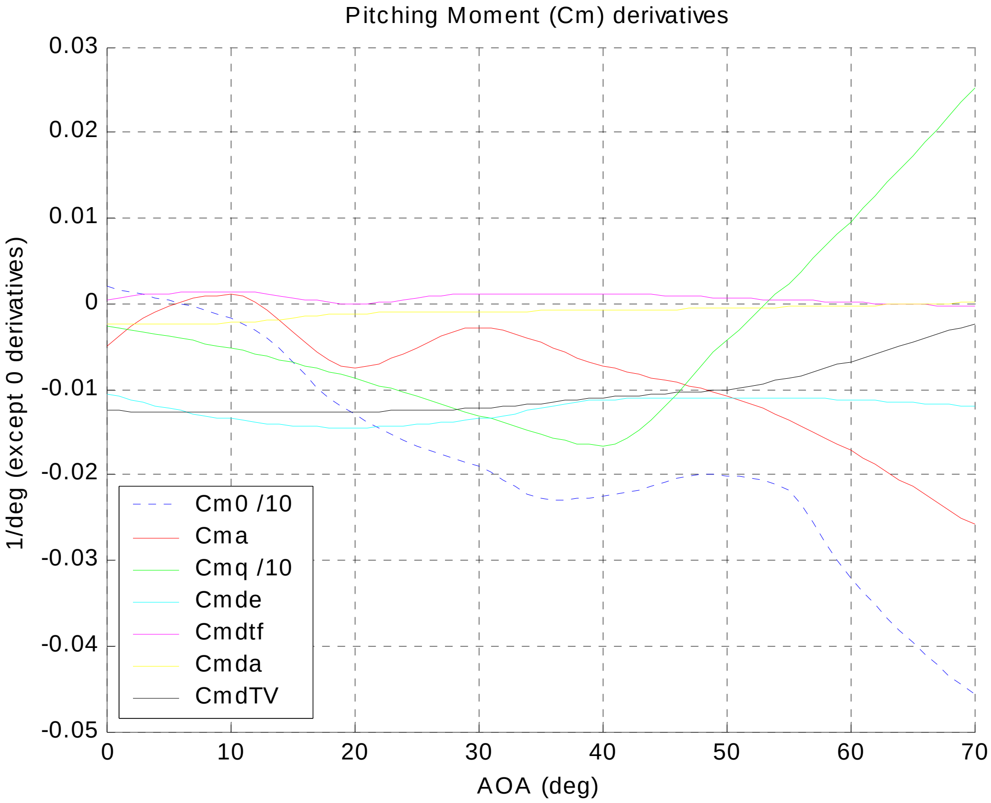

CldYV

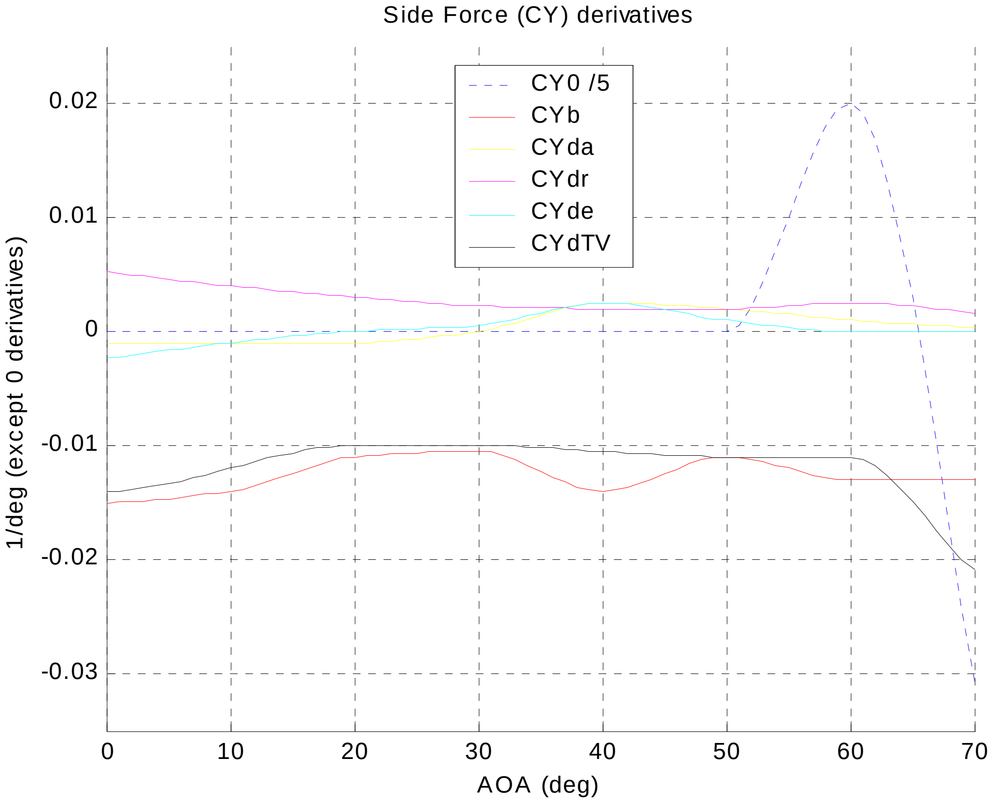

Cn0
no units
0.05
0
-0.05
0
20
40
60
Cnb
x 10^{-3}
1/deg
5
0
-5
0
20
40
60
Cnp
1/rad
2
1
0
-1
-2
0
20
40
60
Cnr
1/rad
2
1
0
-1
-2
0
20
40
60

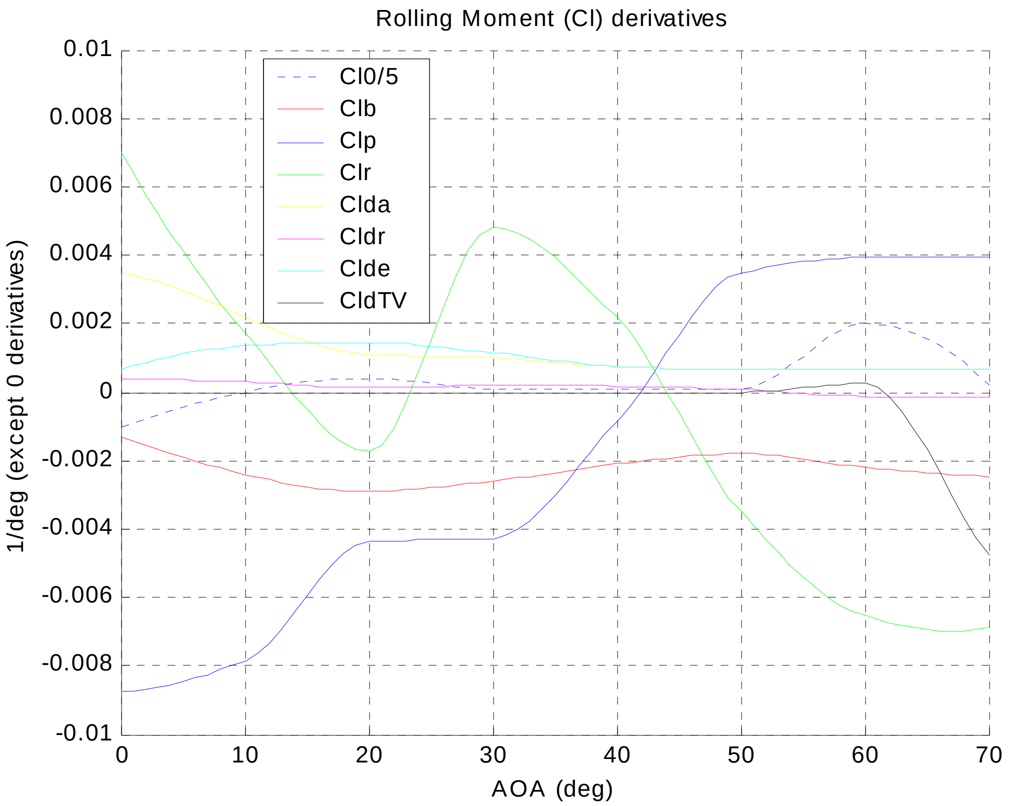

Cnda
x 10^{-3}
1/deg
4
2
0
-2
-4
0
20
40
60
Cndr
x 10^{-3}
1/deg
2
1
0
-1
-2
0
20
40
60
Cndds
x 10^{-3}
1/deg
4
2
0
-2
-4
0
20
40
60
CndYV
ft/deg
0.4
0.3
0.2
0.1
0
-0.1
0
20
40
60

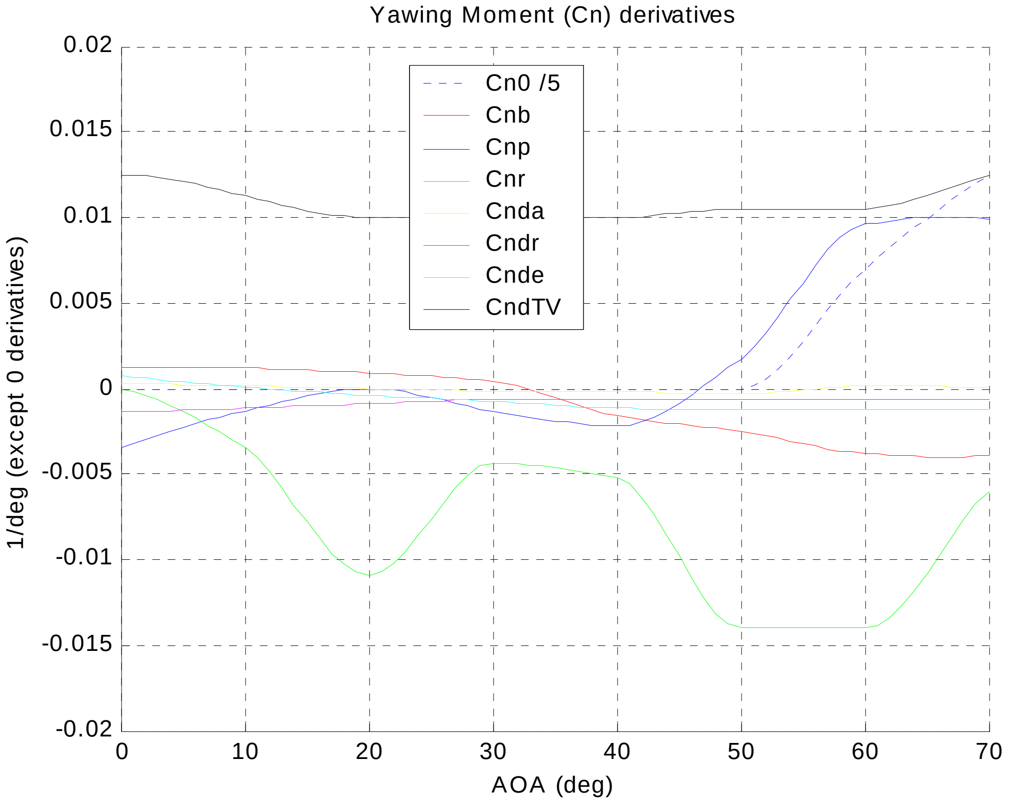

Drag Force (CD) derivatives
1/deg (except 0 derivatives)
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
0
10
20
30
40
50
60
70
AOA (deg)
CD0 /20
CDa
CDde
CDdtf
CDda
CDdTV


Lift Force (CL) derivatives
1/deg (except 0 derivatives)
0.1
0.08
0.06
0.04
0.02
0
-0.02
0
10
20
30
40
50
60
70
AOA (deg)
CL0 /20
CLa
CLde
CLdtf
CLda
CLdTV


Pitching Moment (Cm) derivatives
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
1/deg (except 0 derivatives)
0
10
20
30
40
50
60
70
AOA (deg)
Cm0 /10
Cma
Cmq /10
Cmde
Cmdtf
Cmda
CmdTV


Side Force (CY) derivatives
0.02
0.01
0
-0.01
-0.02
-0.03
1/deg (except 0 derivatives)
0
10
20
30
40
50
60
70
AOA (deg)
CY0 /5
CYb
CYda
CYdr
CYde
CYdTV

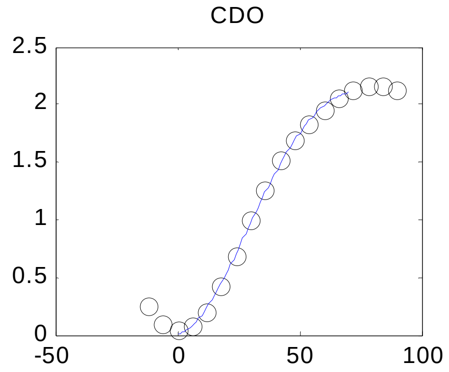

Rolling Moment (Cl) derivatives
0.01
0.008
0.006
0.004
0.002
0
-0.002
-0.004
-0.006
-0.008
-0.01
1/deg (except 0 derivatives)
0
10
20
30
40
50
60
70
AOA (deg)
Cl0/5
Clb
Clp
Clr
Clda
Cldr
Clde
CldTV

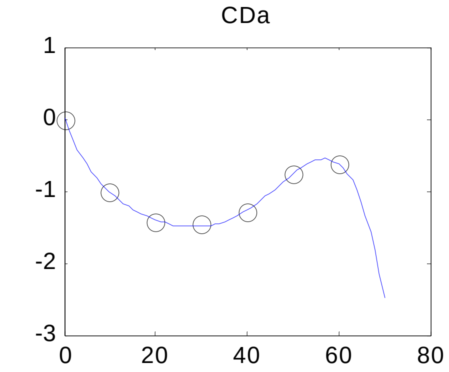

Yawing Moment (Cn) derivatives
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
1/deg (except 0 derivatives)
0
10
20
30
40
50
60
70
AOA (deg)
Cn0 /5
Cnb
Cnp
Cnr
Cnda
Cndr
Cnde
CndTV

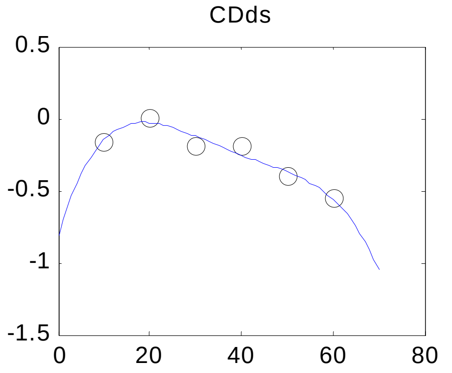
APPENDIX E: CURVE-FITTING PLOTS

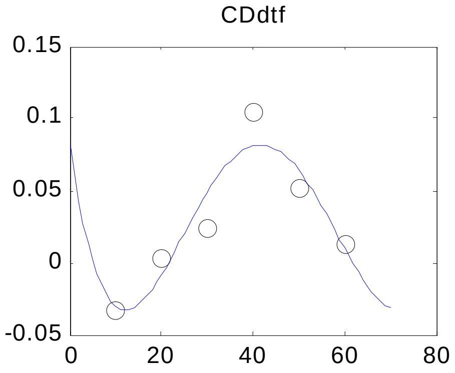
This Appendix contains plots of the HARV S&C derivatives calculated by curve-fitting. The circles are actual flight test data points, while the lines are the values calculated by the fourth and fifth order functions of AOA calculated by MATLAB's "polyfit" command. Each derivative is plotted versus AOA (measured in degrees along the x-axis). S&C derivatives are grouped by force and moment coefficients, and the units for all derivatives are per radian.

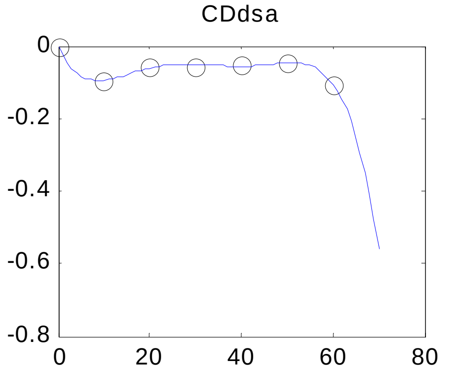
NOTE: Not all S&C derivatives in this Appendix can be compared to the derivatives in Appendix D. For one, this Appendix includes drag and lift derivatives, while Appendix D included axial and normal derivatives (for comparison with fairings done by Iliff and Wang). Also, while all derivatives in this Appendix are in units of per radian, most derivatives in Appendix D are in units of per degree; trends will therefore be the same, but not values.

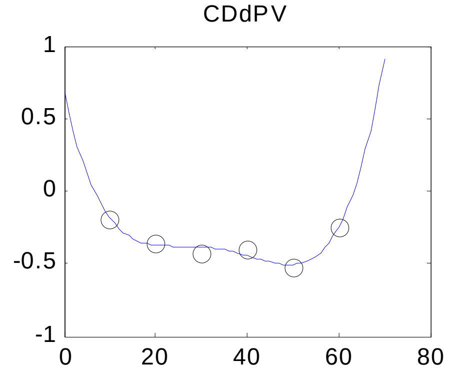

CDO
2.5
2
1.5
1
0.5
0
-50
0
50
100

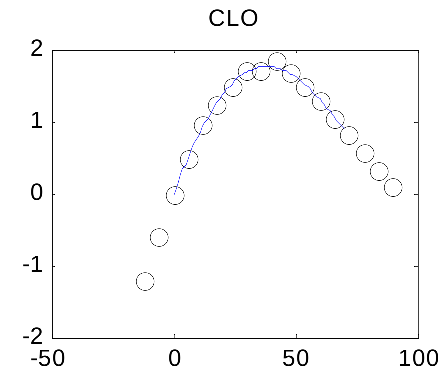

CDa
1
0
-1
-2
-3
0
20
40
60
80

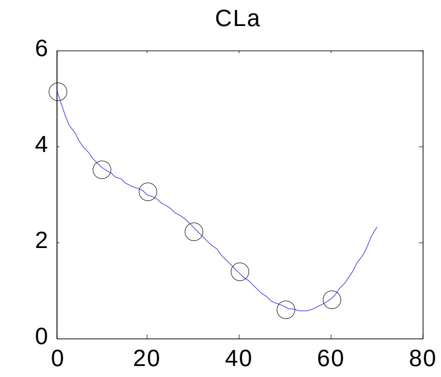

CDds
0.5
0
-0.5
-1
-1.5
0
20
40
60
80

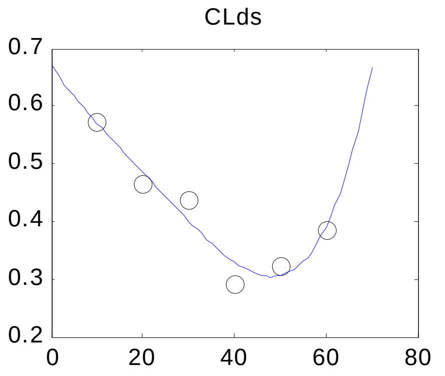

CDdtf
0.15
0.1
0.05
0
-0.05
0
20
40
60
80

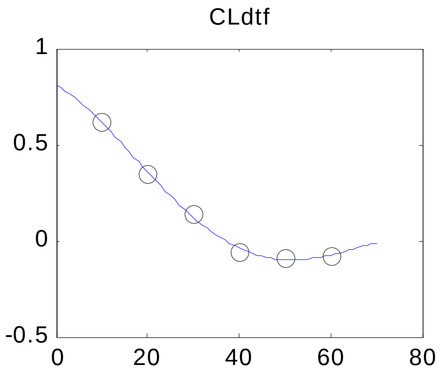

CDdsa
0
-0.2
-0.4
-0.6
-0.8
0
20
40
60
80

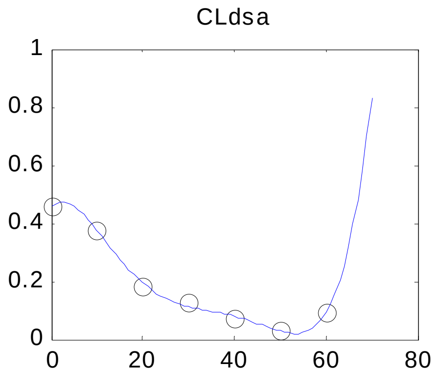

CDdPV
1
0.5
0
-0.5
-1
0
20
40
60
80

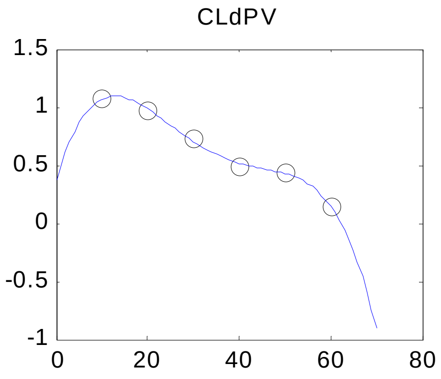

CLO
2
1
0
-1
-2
-50
0
50
100

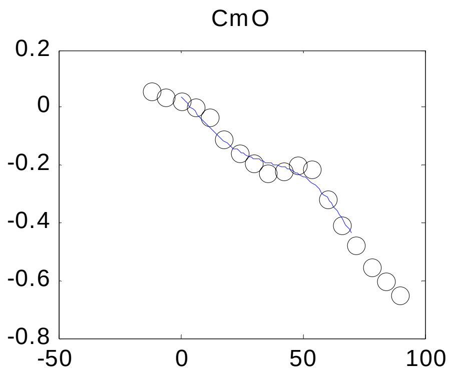

CLa
6
4
2
0
0
20
40
60
80

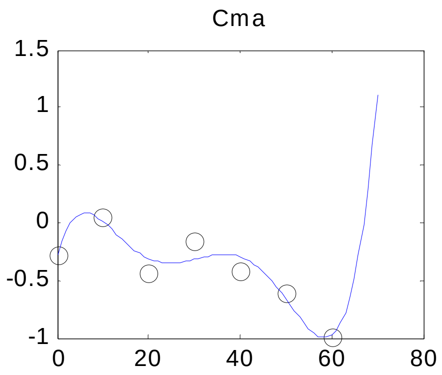

CLds
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80

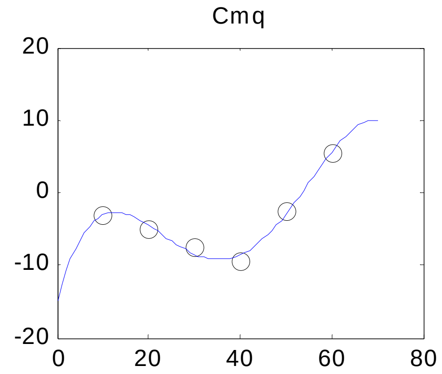

CLdtf
1
0.5
0
-0.5
0
20
40
60
80

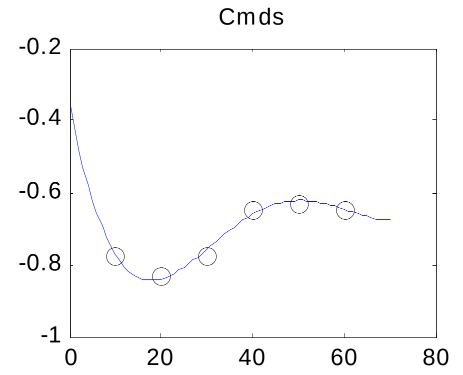

CLdsa
1
0.8
0.6
0.4
0.2
0
0
20
40
60
80

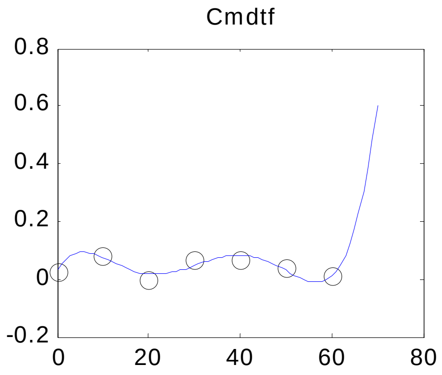

CLdPV
1.5
1
0.5
0
-0.5
-1
0
20
40
60
80

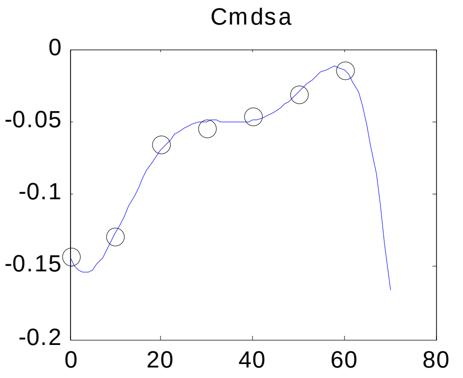

CmO
0.2
0
-0.2
-0.4
-0.6
-0.8
-50
0
50
100

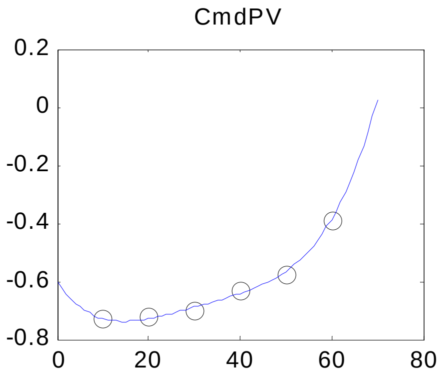

Cma
1.5
1
0.5
0
-0.5
-1
0
20
40
60
80

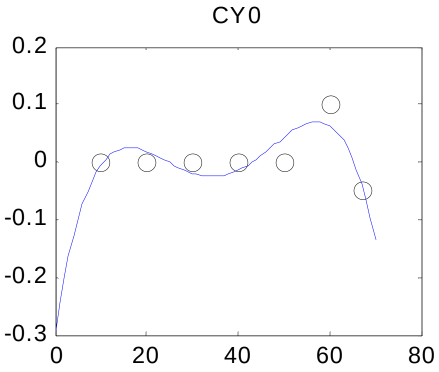

Cmq
20
10
0
-10
-20
0
20
40
60
80

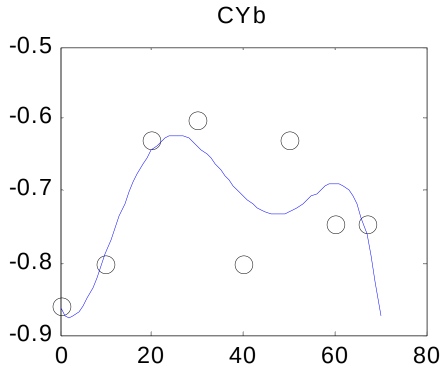

Cmds
-0.2
-0.4
-0.6
-0.8
-1
0
20
40
60
80

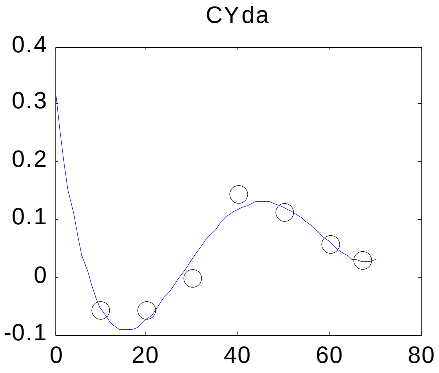

Cmdtf
0.8
0.6
0.4
0.2
0
-0.2
0
20
40
60
80

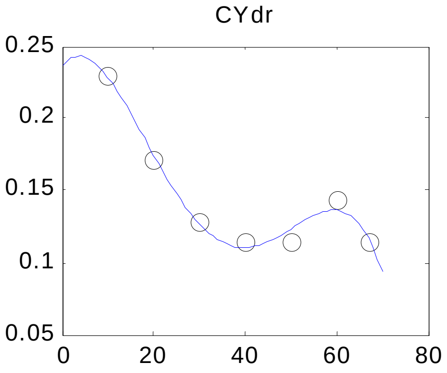

Cmdsa
0
-0.05
-0.1
-0.15
-0.2
0
20
40
60
80

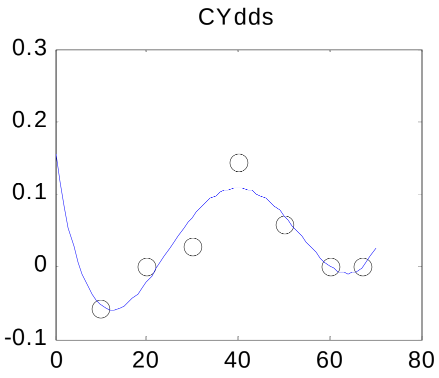

CmdPV
0.2
0
-0.2
-0.4
-0.6
-0.8
0
20
40
60
80

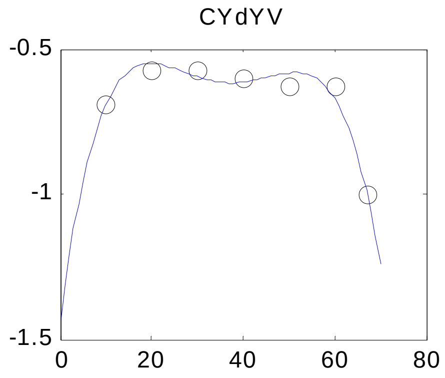

CY0
0.2
0.1
0
-0.1
-0.2
-0.3
0
20
40
60
80

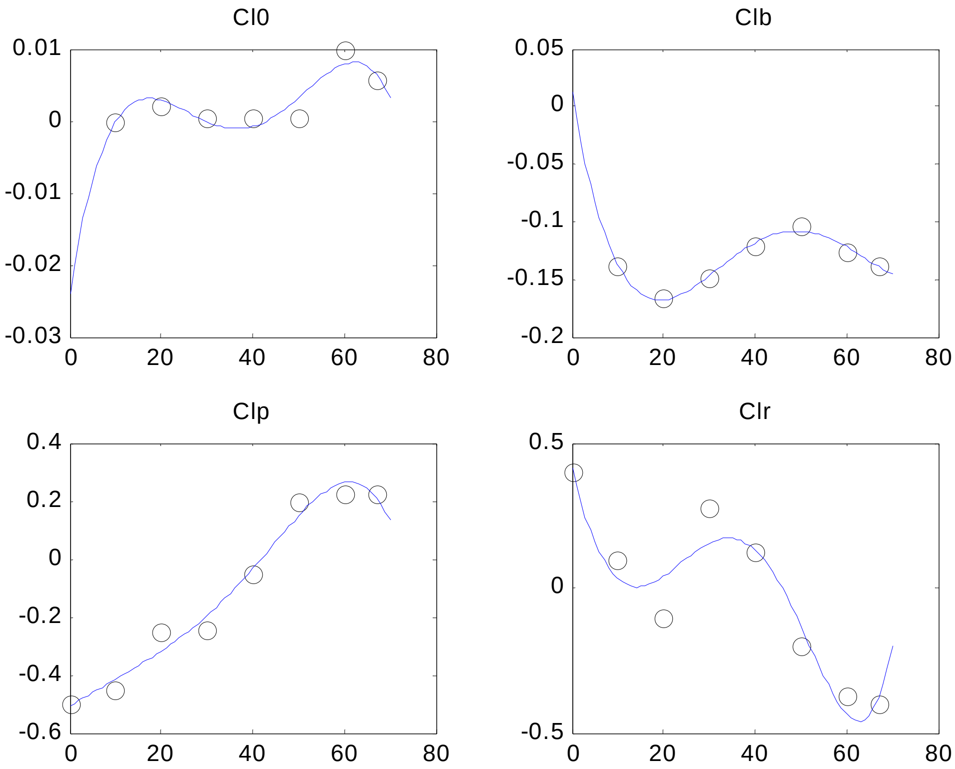

CYb
-0.5
-0.6
-0.7
-0.8
-0.9
0
20
40
60
80

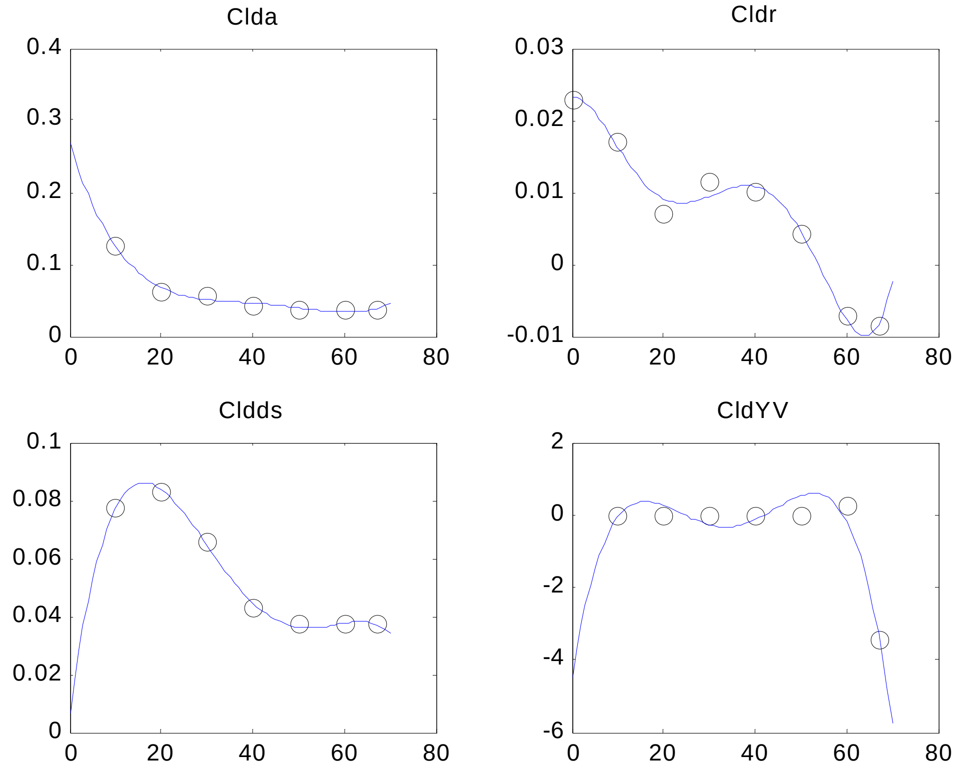

CYda
0.4
0.3
0.2
0.1
0
-0.1
0
20
40
60
80

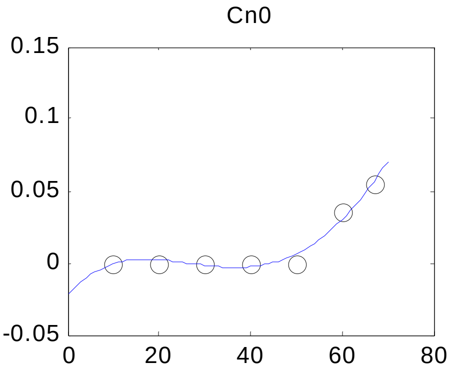

CYdr
0.25
0.2
0.15
0.1
0.05
0
20
40
60
80

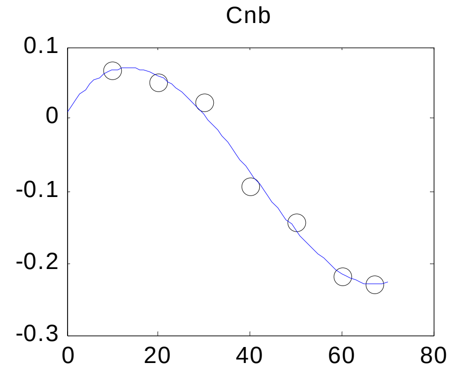

CYdds
0.3
0.2
0.1
0
-0.1
0
20
40
60
80

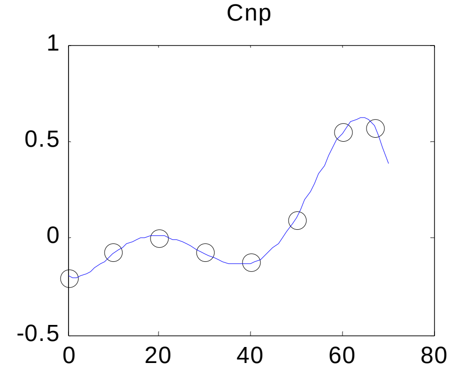

CYdYV
-0.5
-1
-1.5
0
20
40
60
80

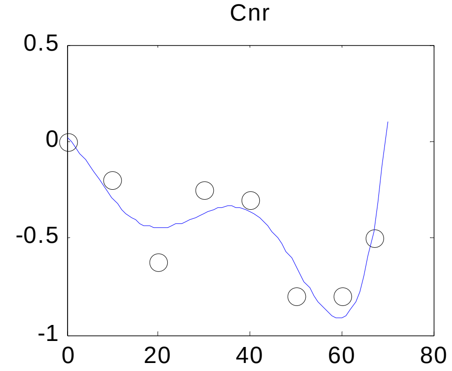

Cl0
0.01
0
-0.01
-0.02
-0.03
0
20
40
60
80
Clb
0.05
0
-0.05
-0.1
-0.15
-0.2
0
20
40
60
80
Clp
0.4
0.2
0
-0.2
-0.4
-0.6
0
20
40
60
80
Clr
0.5
0
-0.5
0
20
40
60
80

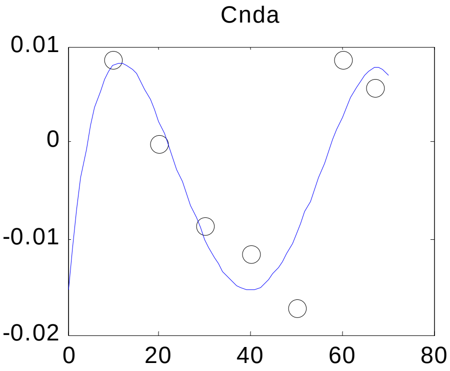

Clda
0.4
0.3
0.2
0.1
0
0
20
40
60
80
Cldr
0.03
0.02
0.01
0
-0.01
0
20
40
60
80
Cldds
0.1
0.08
0.06
0.04
0.02
0
0
20
40
60
80
CldYV
2
0
-2
-4
-6
0
20
40
60
80

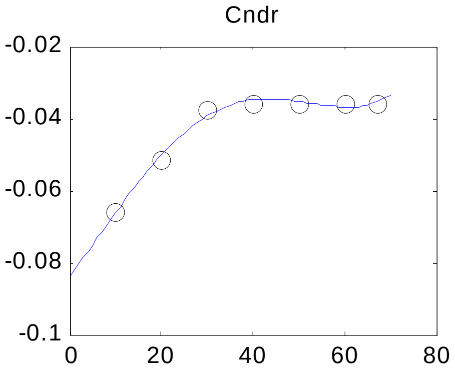

Cn0
0.15
0.1
0.05
0
-0.05
0
20
40
60
80

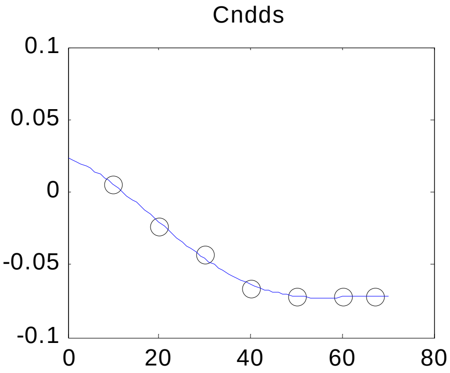

Cnb
0.1
0
-0.1
-0.2
-0.3
0
20
40
60
80

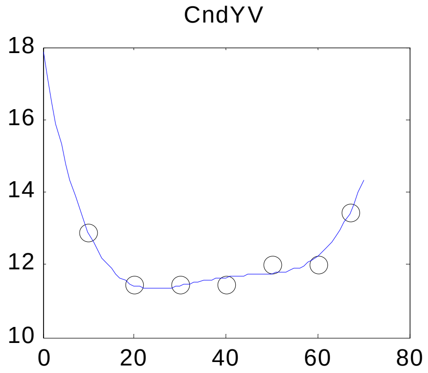

Cnp
1
0.5
0
-0.5
0
20
40
60
80


Cnr
0.5
0
-0.5
-1
0
20
40
60
80


Cnda
0.01
0
-0.01
-0.02
0
20
40
60
80


Cndr
-0.02
-0.04
-0.06
-0.08
-0.1
0
20
40
60
80


Cndds
0.1
0.05
0
-0.05
-0.1
0
20
40
60
80


CndYV
18
16
14
12
10
0
20
40
60
80

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F: NAVION RESULTS

A. STRAIGHT CLIMB

Delta: ΔH = 1000 ft
ΔX = 10000 ft

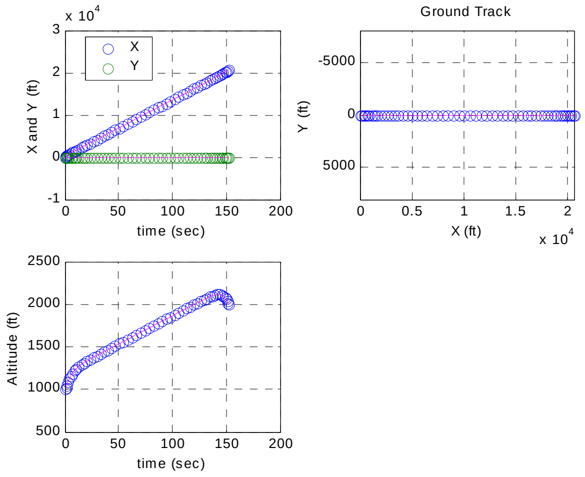
Unconstrained: X (0–50k ft)
p, q, r (0 ± 2°/sec)

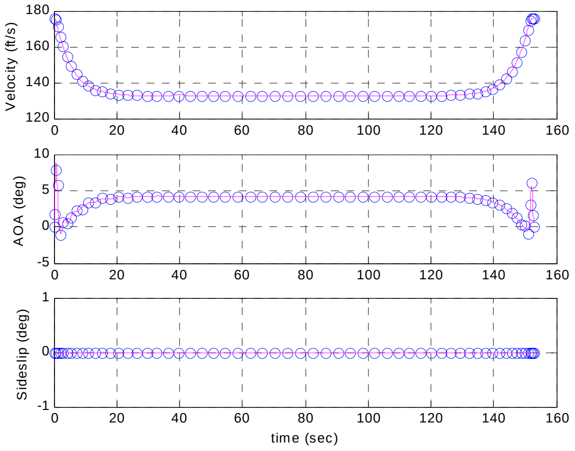
60 Nodes (not bootstrapped)

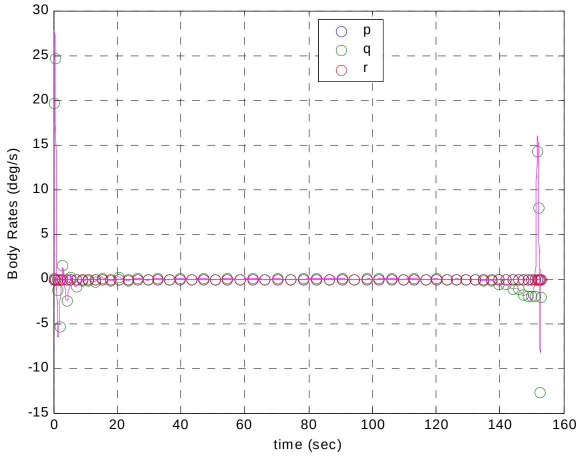
# of Iterations: 2531
Total Run Time: 4.52 min

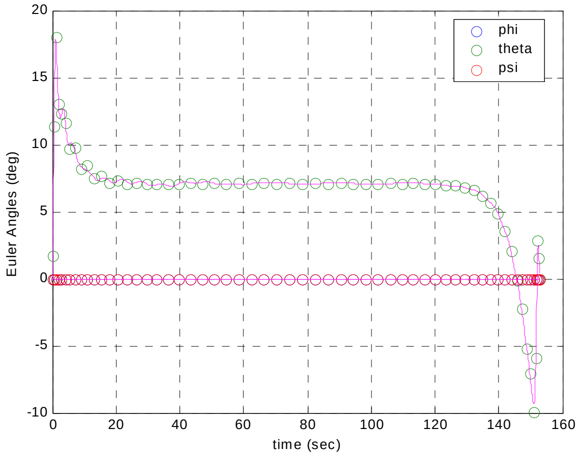
Maneuver Time: 152.8 sec

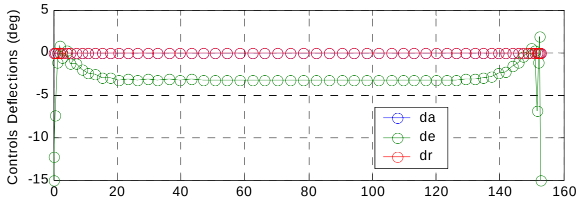
NOTE: Due to the scale of this particular maneuver, a 3D trajectory view was not included. Both the flight path and orientation of the aircraft can easily be determined from the first and fourth DIDO plots, respectively.

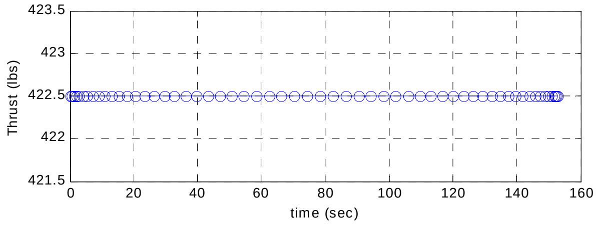


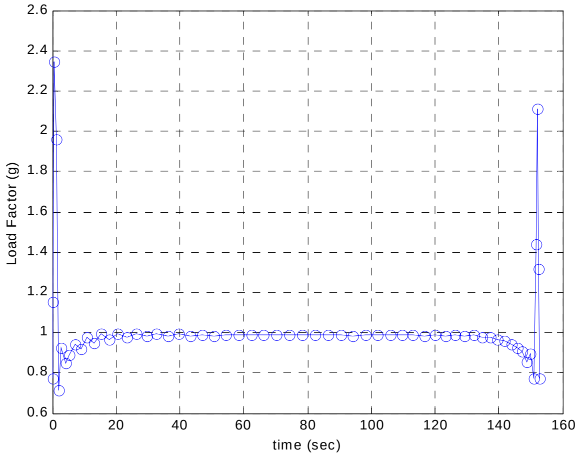


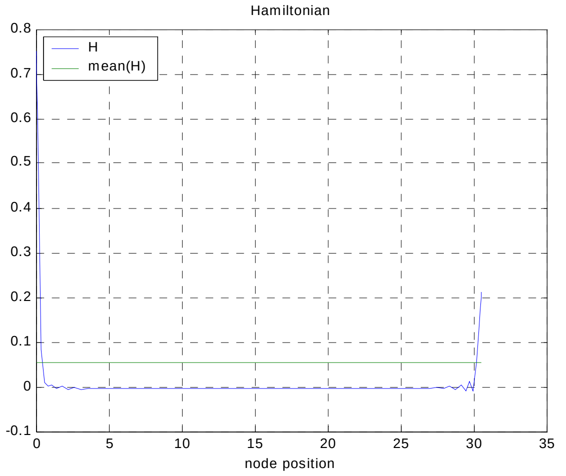

Body Rates (deg/s)
time (sec)
p
q
r
30
25
20
15
10
5
0
-5
-10
-15
0
20
40
60
80
100
120
140
160

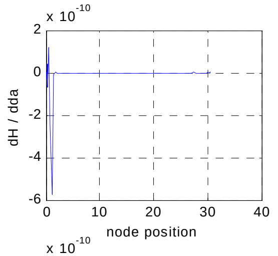

Euler Angles (deg)
time (sec)
phi
theta
psi
20
15
10
5
0
-5
-10
0
20
40
60
80
100
120
140
160

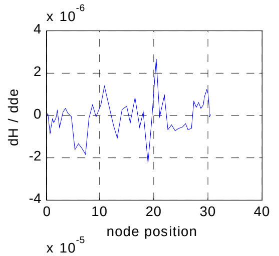

Controls Deflections (deg)
da
de
dr
5
0
-5
-10
-15
0
20
40
60
80
100
120
140
160

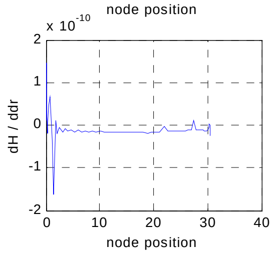

Thrust (lbs)
time (sec)
423.5
423
422.5
422
421.5
0
20
40
60
80
100
120
140
160

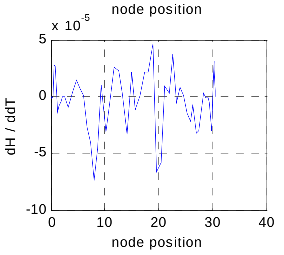

Load Factor (g)
time (sec)
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0
20
40
60
80
100
120
140
160


Hamiltonian
H
mean(H)
node position
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
5
10
15
20
25
30
35


x 10^-10
dH / dda
node position
2
0
-2
-4
-6
0
10
20
30
40


x 10^-6
dH / dde
node position
4
2
0
-2
-4
0
10
20
30
40


x 10^-10
dH / ddr
node position
2
1
0
-1
-2
0
10
20
30
40


x 10^-5
dH / ddT
node position
5
0
-5
-10
0
10
20
30
40

B. LEVEL TURN

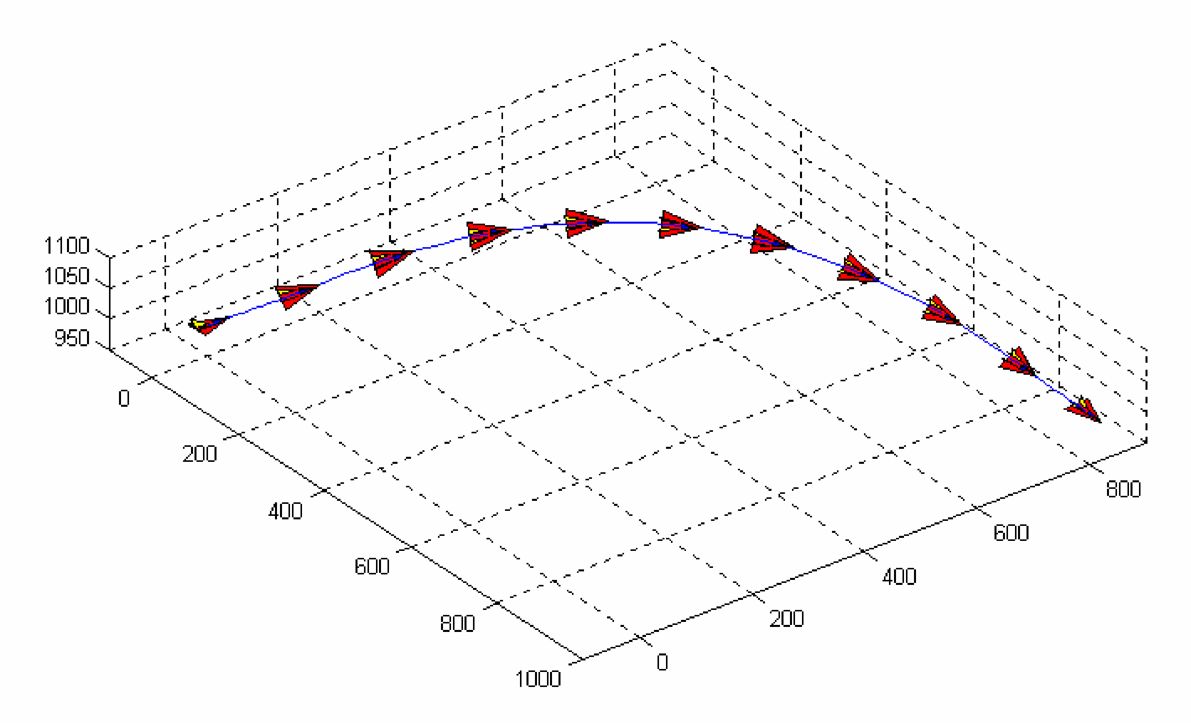
Delta: $\Delta\psi$ = 90°
ΔX = 1000 ft
ΔY = 1000 ft

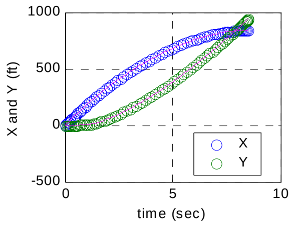
Unconstrained: X, Y (700–1300 ft)
ϕ, θ (0 ± 2°)
p, q, r (0 ± 2°/sec)

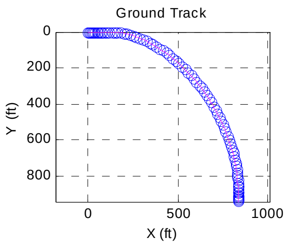
90 Nodes (not bootstrapped)

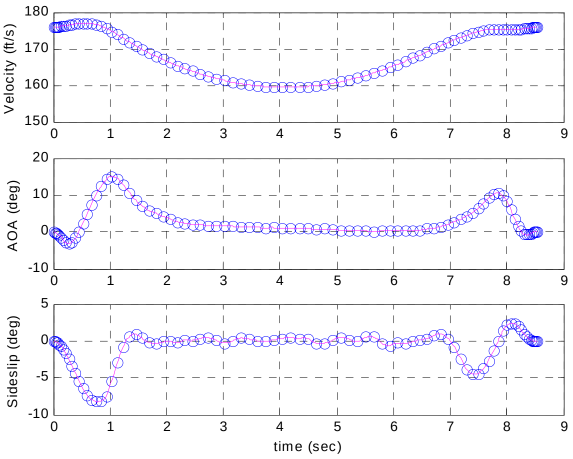
# of Iterations: 9318
Total Run Time: 17.83 min

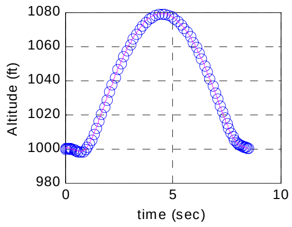
Maneuver Time: 8.52 sec

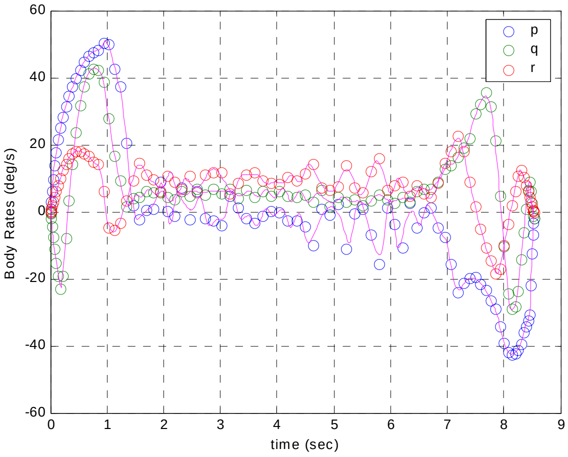


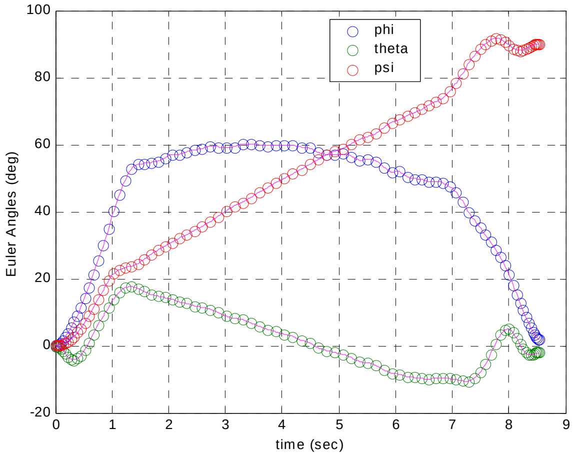


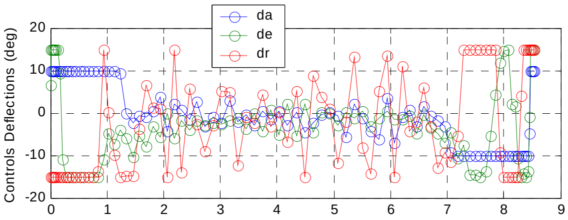


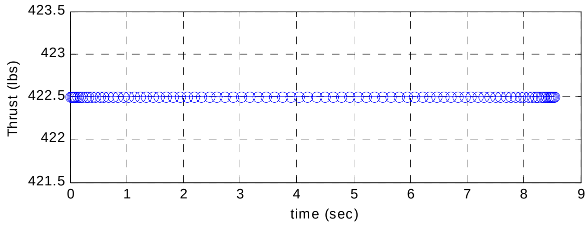


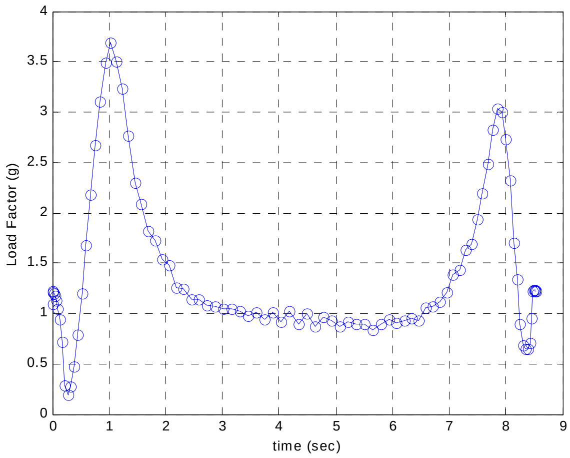


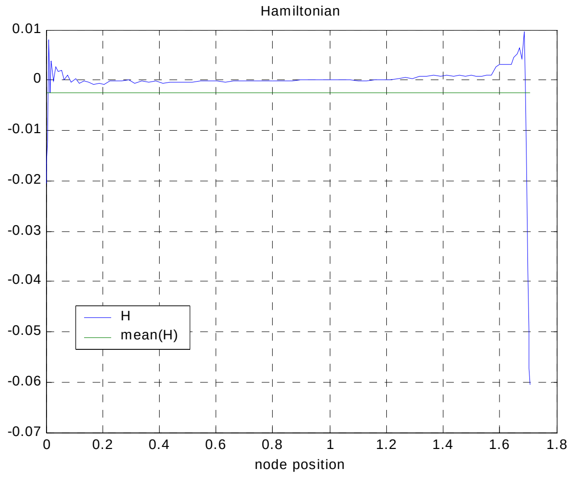

Body Rates (deg/s)
time (sec)
p
q
r
60
40
20
0
-20
-40
-60
0
1
2
3
4
5
6
7
8
9

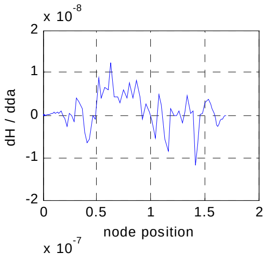

Euler Angles (deg)
time (sec)
phi
theta
psi
100
80
60
40
20
0
-20
0
1
2
3
4
5
6
7
8
9

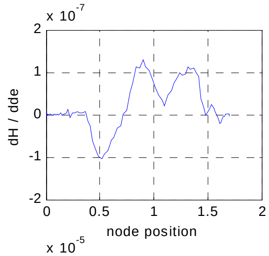

Controls Deflections (deg)
da
de
dr
20
10
0
-10
-20
0
1
2
3
4
5
6
7
8
9

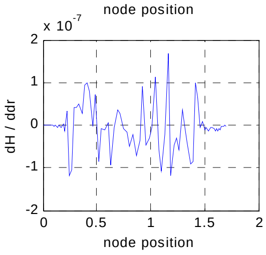

Thrust (lbs)
time (sec)
423.5
423
422.5
422
421.5
0
1
2
3
4
5
6
7
8
9

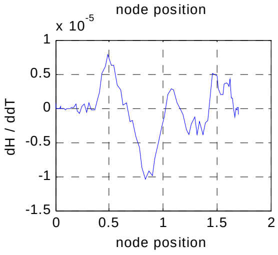

Load Factor (g)
time (sec)
4
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
7
8
9


Hamiltonian
node position
H
mean(H)
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8


x 10-8
dH / dda
node position
2
1
0
-1
-2
0
0.5
1
1.5
2


x 10-7
dH / dde
node position
2
1
0
-1
-2
0
0.5
1
1.5
2


x 10-7
dH / ddr
node position
2
1
0
-1
-2
0
0.5
1
1.5
2


x 10-5
dH / ddT
node position
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5
2

C. CLIMBING TURN

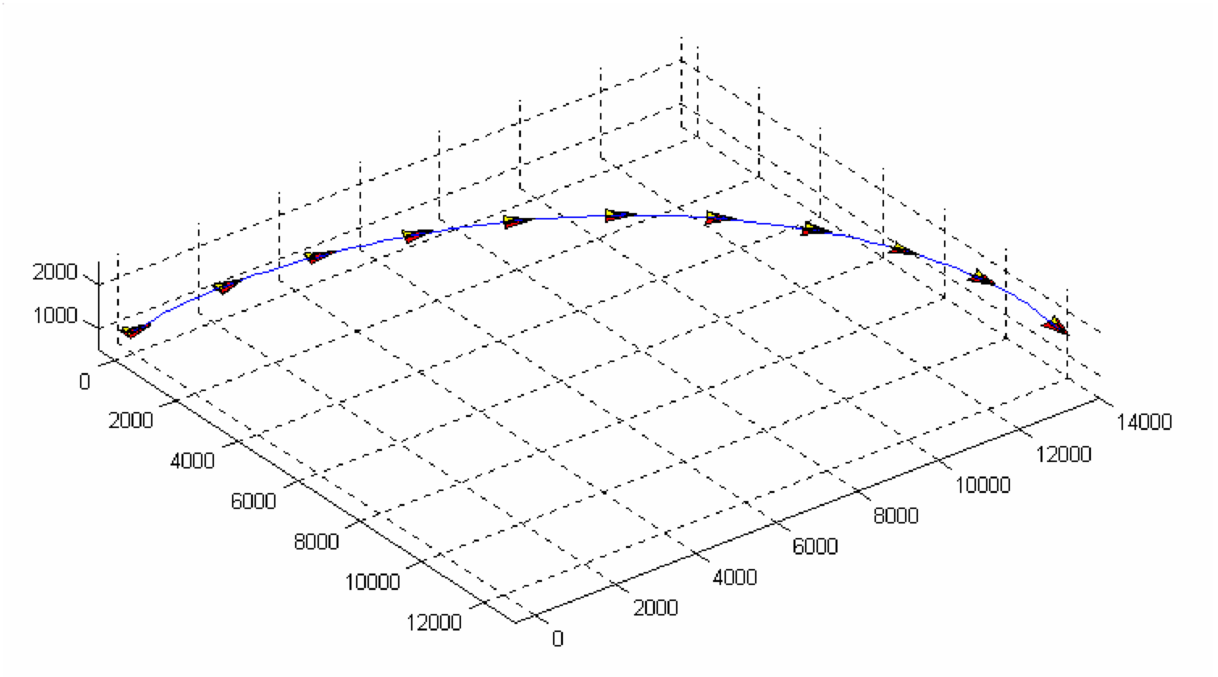
Delta: Δψ = 90°
ΔH = 1000 ft
ΔX = 15000 ft
ΔY = 10000 ft

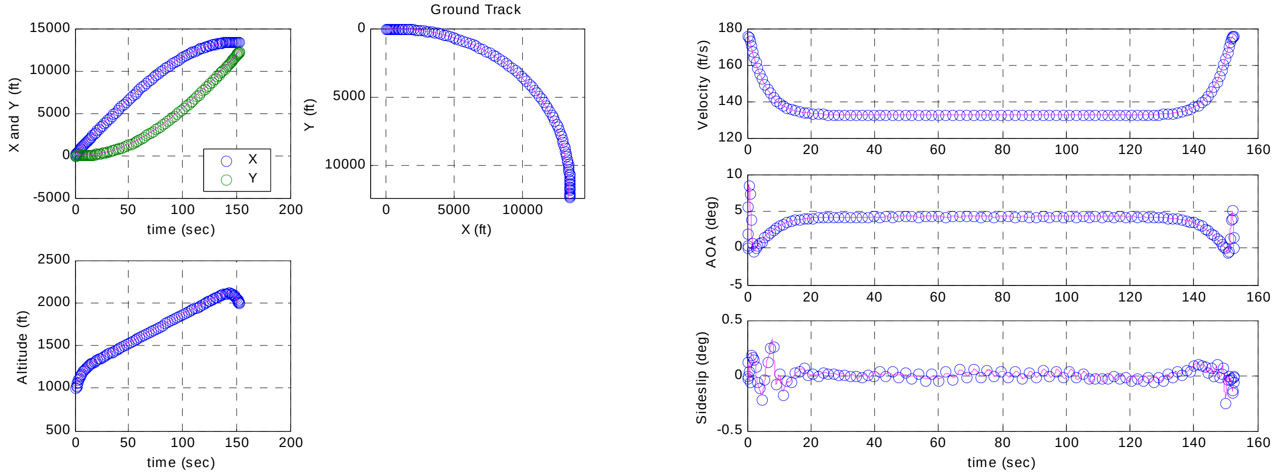
110 Nodes (not bootstrapped)

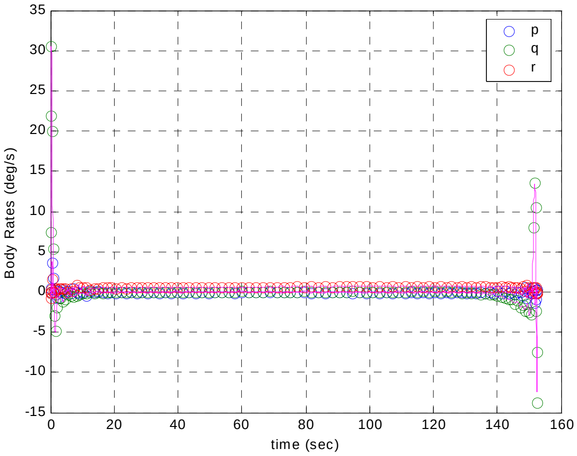
# of Iterations: 10076
Total Run Time: 36.10 min

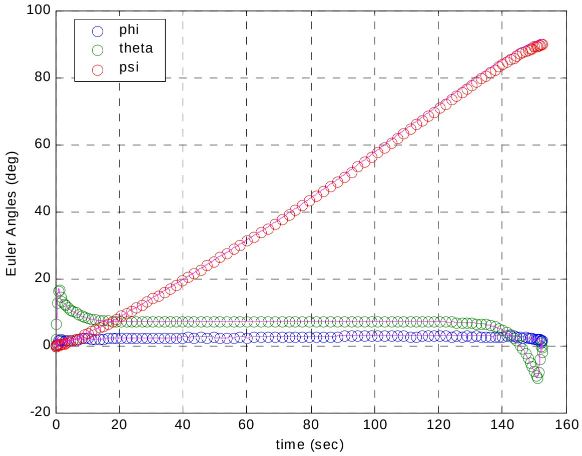
Unconstrained: X, Y (0–30k ft)
ϕ, θ (0 ± 2°)

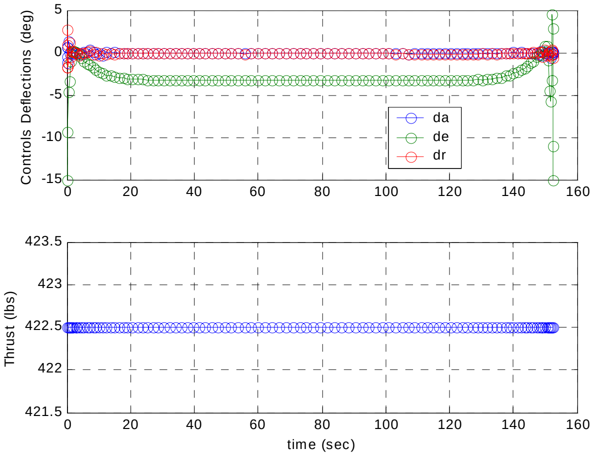
Maneuver Time: 152.6 sec
(Aircraft scale 10:1 in figure below.)

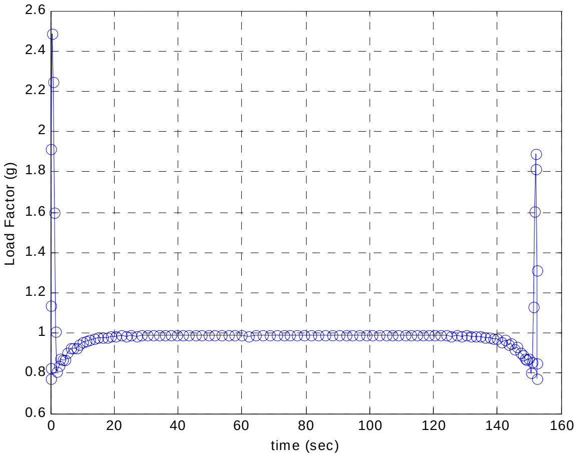


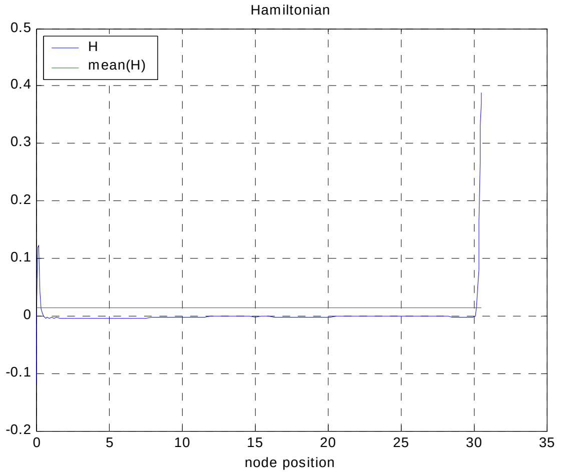


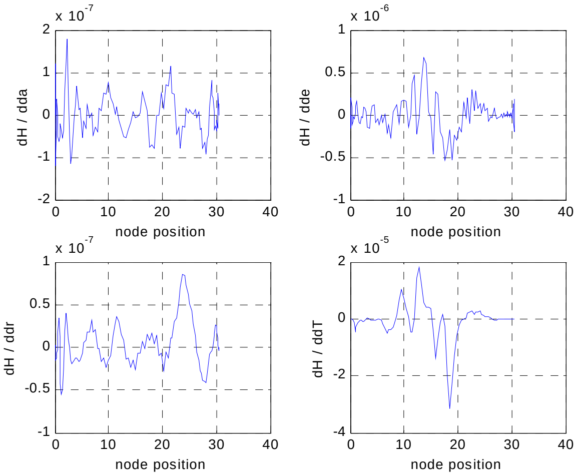

Body Rates (deg/s)
time (sec)
p
q
r
35
30
25
20
15
10
5
0
-5
-10
-15
0
20
40
60
80
100
120
140
160


Euler Angles (deg)
time (sec)
phi
theta
psi
100
80
60
40
20
0
-20
0
20
40
60
80
100
120
140
160


Controls Deflections (deg)
da
de
dr
5
0
-5
-10
-15
0
20
40
60
80
100
120
140
160
Thrust (lbs)
423.5
423
422.5
422
421.5
0
20
40
60
80
100
120
140
160
time (sec)


Load Factor (g)
time (sec)
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0
20
40
60
80
100
120
140
160


Hamiltonian
H
mean(H)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
5
10
15
20
25
30
35
node position


x 10^-7
dH / dda
2
1
0
-1
-2
0
10
20
30
40
node position
x 10^-6
dH / dde
1
0.5
0
-0.5
-1
0
10
20
30
40
node position
x 10^-7
dH / ddr
1
0.5
0
-0.5
-1
0
10
20
30
40
node position
x 10^-5
dH / ddT
2
0
-2
-4
0
10
20
30
40
node position

D. WINGOVER

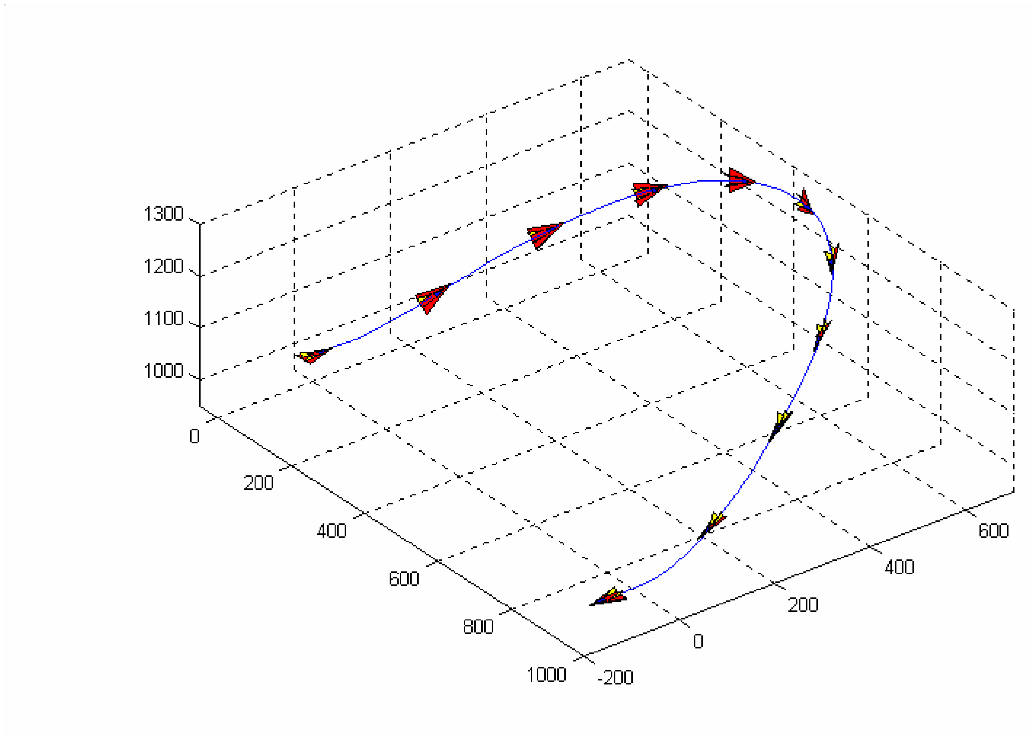
Delta: Δψ = 180°
ΔY = 1000 ft

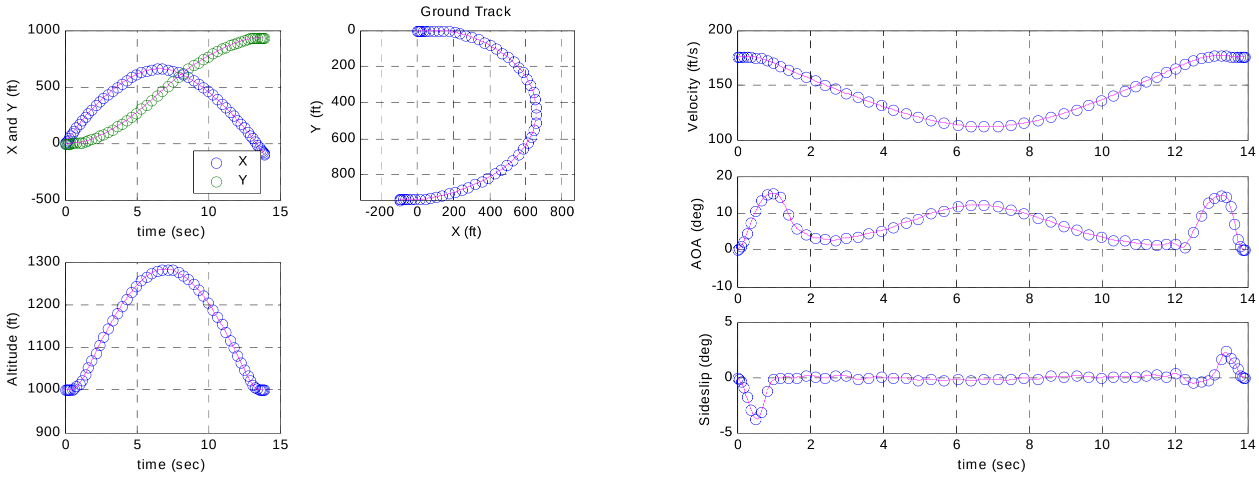
Unconstrained: X (0 ± 100 ft)
Y (900-1100 ft)
p, q, r (0 ± 2°/sec)

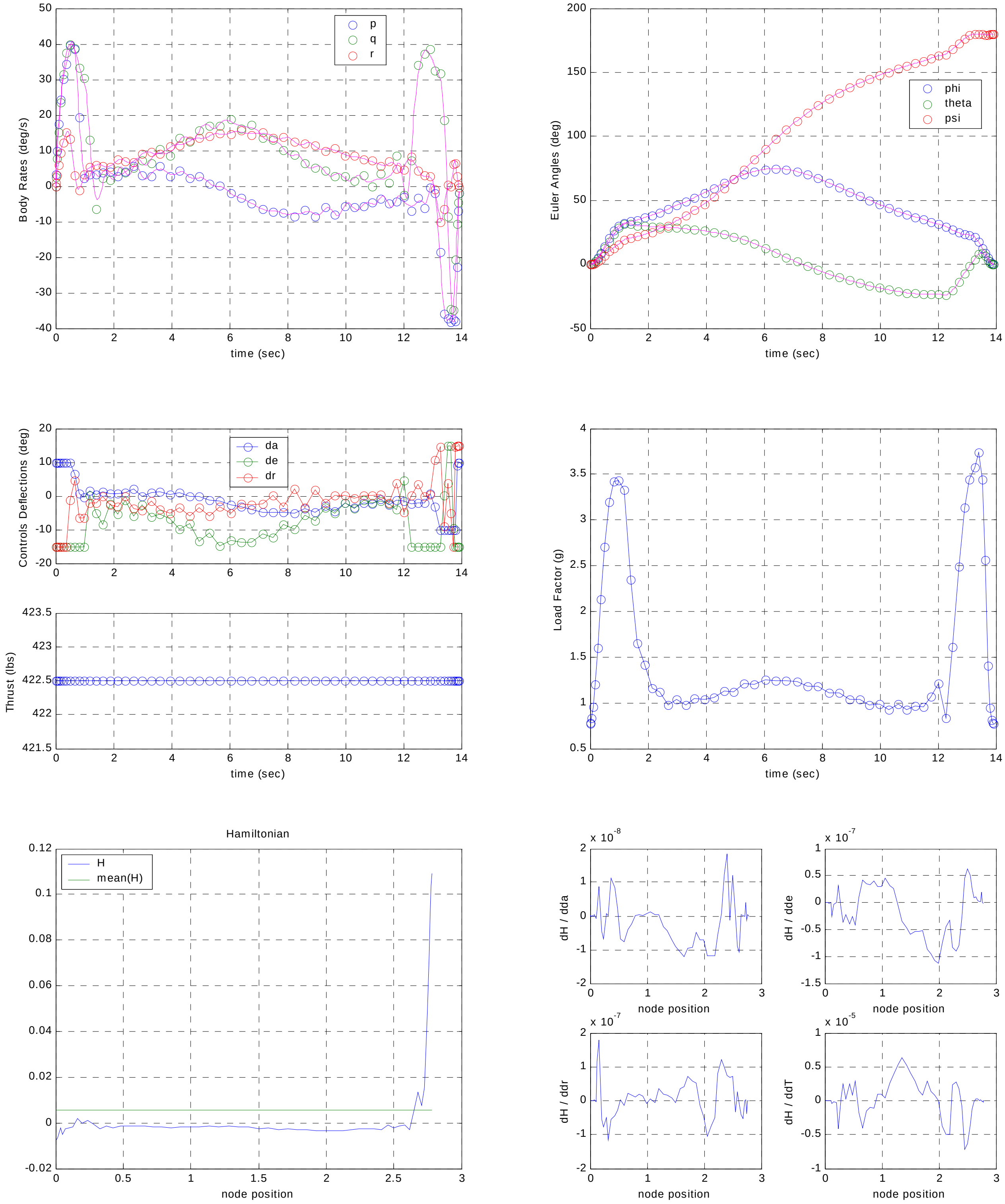
60 Nodes (not bootstrapped)

# of Iterations: 4375
Total Run Time: 4.22 min

Maneuver Time: 13.91 sec






Body Rates (deg/s)
time (sec)
p
q
r
Euler Angles (deg)
time (sec)
phi
theta
psi
Controls Deflections (deg)
da
de
dr
Thrust (lbs)
time (sec)
Load Factor (g)
time (sec)
Hamiltonian
H
mean(H)
node position
x 10^{-8}
dH / dda
node position
x 10^{-7}
dH / dde
node position
x 10^{-7}
dH / ddr
node position
x 10^{-5}
dH / ddT
node position

E. REVERSAL MANEUVER

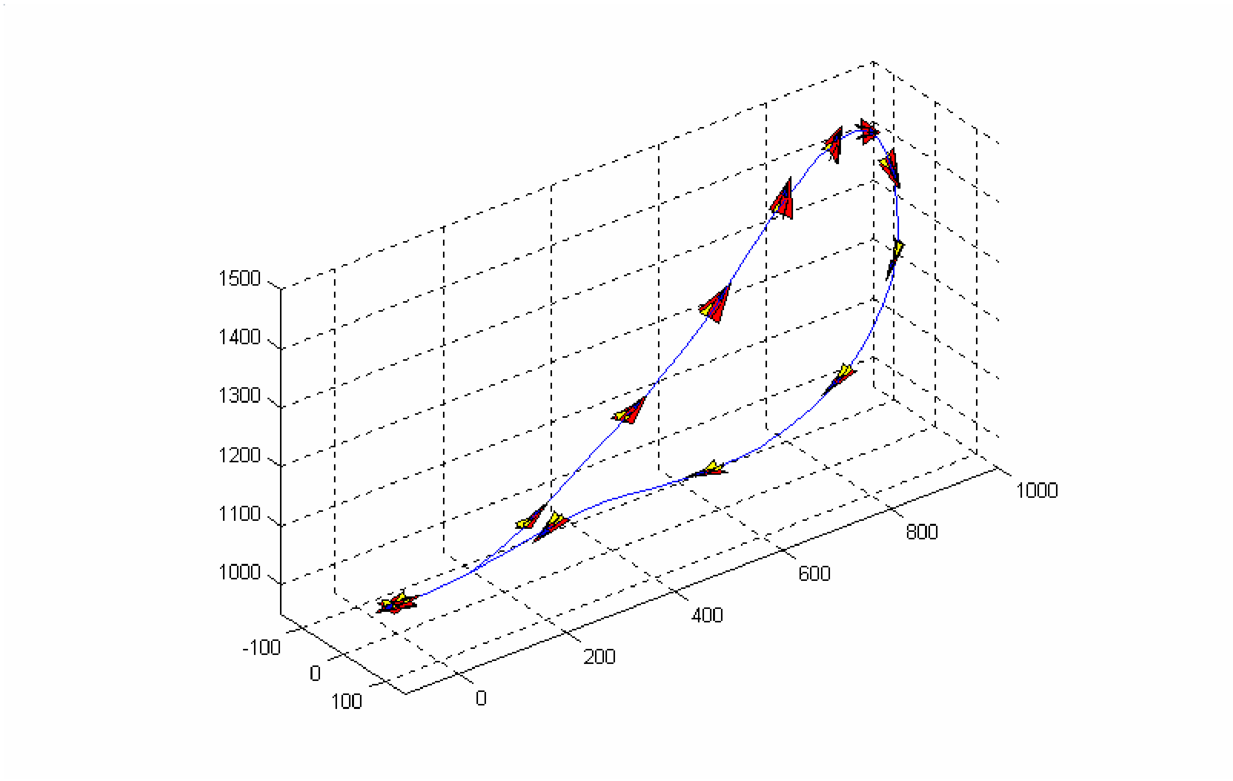
Delta: $\Delta\psi = 180°$
$\Delta H = 0$ ft
$\Delta V = 0$ fps
$\Delta X = 0$ ft
$\Delta Y = 0$ ft

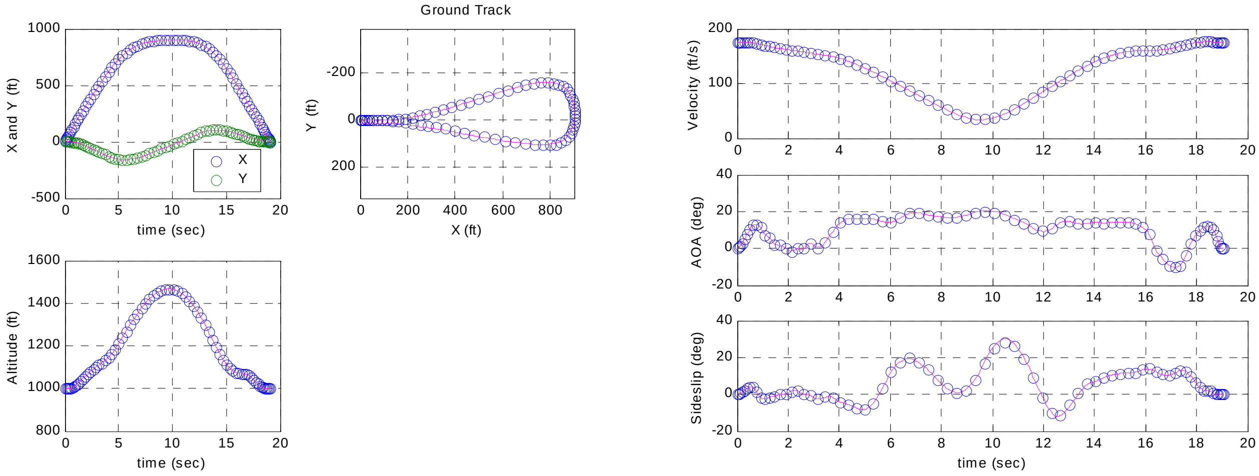
Unconstrained: (None)

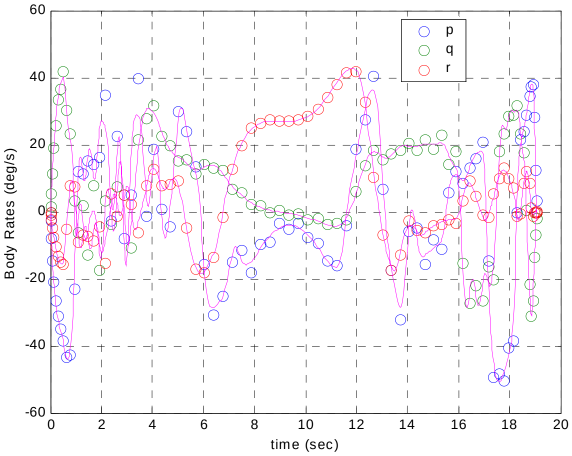
80 Nodes (not bootstrapped)

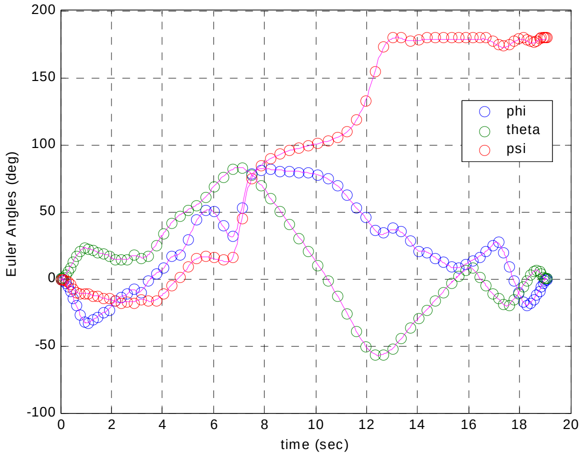
# of Iterations: 6239
Total Run Time: 25.30 min

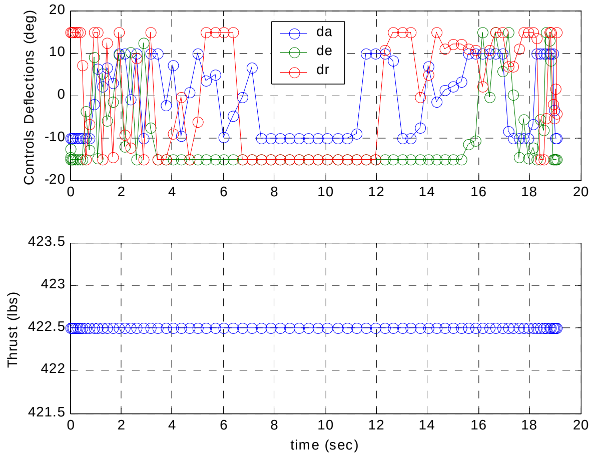
Maneuver Time: 19.05 sec

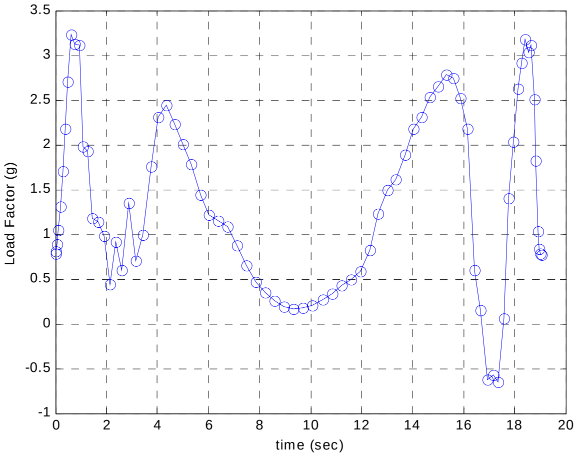


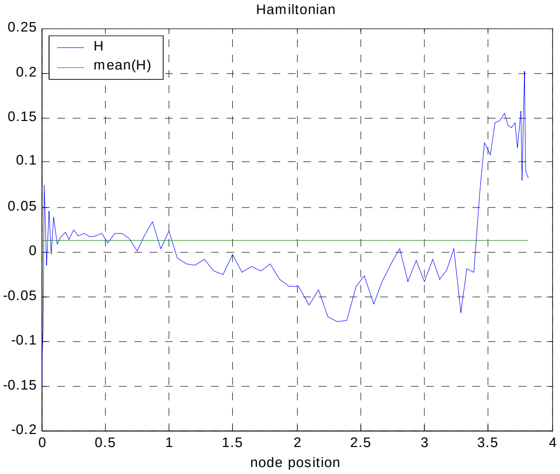


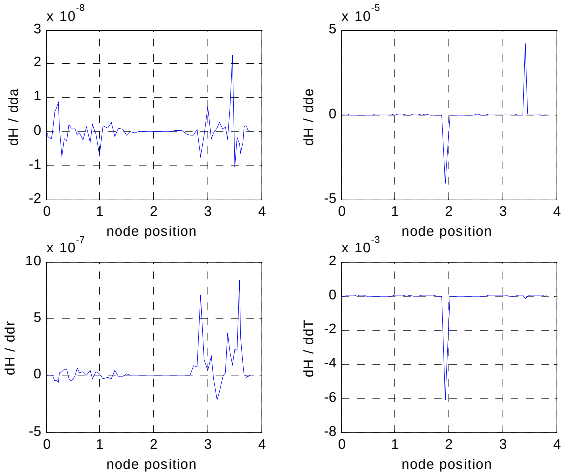

Body Rates (deg/s)
time (sec)
p
q
r
60
40
20
0
-20
-40
-60
0
2
4
6
8
10
12
14
16
18
20


Euler Angles (deg)
time (sec)
phi
theta
psi
200
150
100
50
0
-50
-100
0
2
4
6
8
10
12
14
16
18
20


Controls Deflections (deg)
da
de
dr
20
10
0
-10
-20
0
2
4
6
8
10
12
14
16
18
20
Thrust (lbs)
time (sec)
423.5
423
422.5
422
421.5
0
2
4
6
8
10
12
14
16
18
20


Load Factor (g)
time (sec)
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
18
20


Hamiltonian
H
mean(H)
node position
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
0.5
1
1.5
2
2.5
3
3.5
4


x 10^-8
dH / dda
node position
3
2
1
0
-1
-2
x 10^-5
dH / dde
node position
5
0
-5
x 10^-7
dH / ddr
node position
10
5
0
-5
x 10^-3
dH / ddT
node position
2
0
-2
-4
-6
-8
0
1
2
3
4

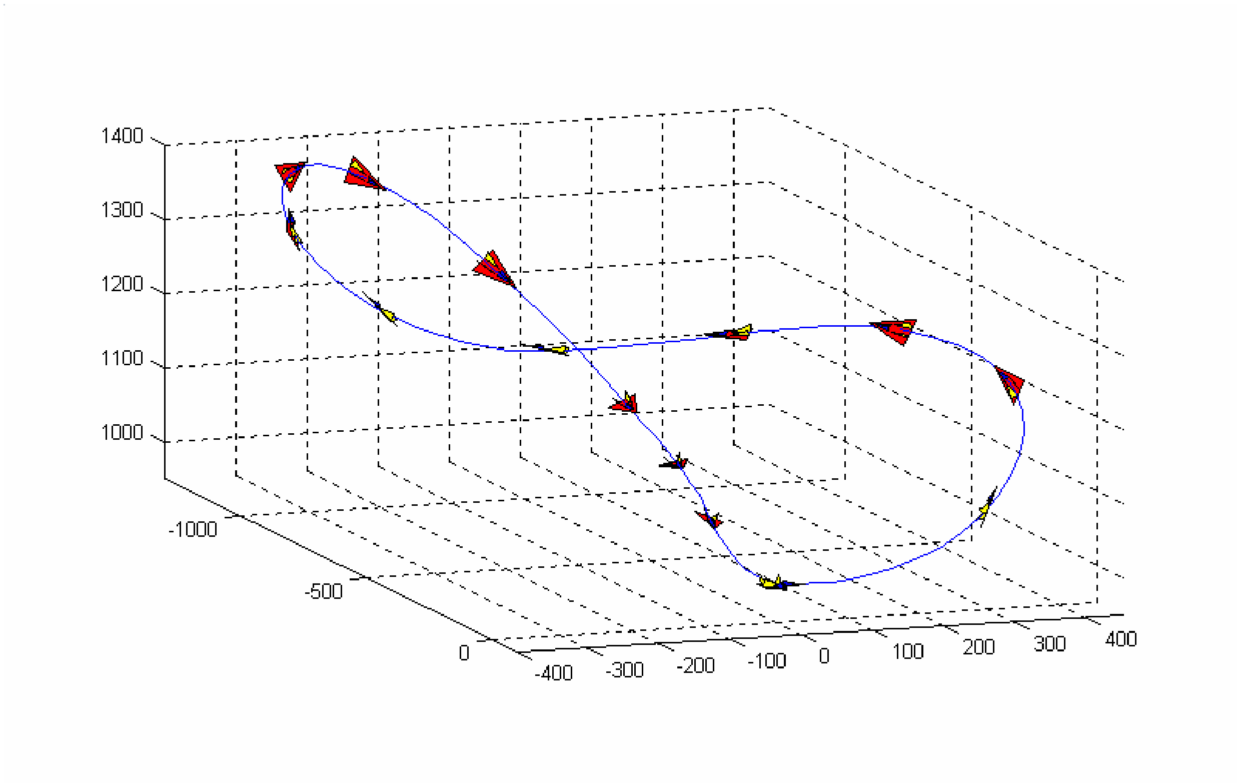
F. TURNING MANEUVER

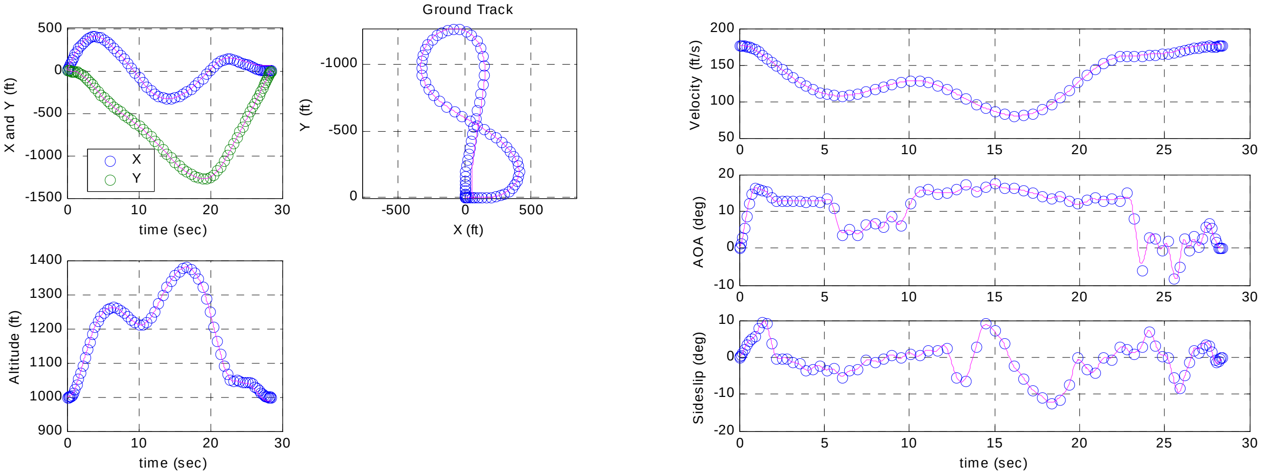
Delta: $\Delta\psi = 90°$
$\Delta H = 0$ ft
$\Delta V = 0$ fps
$\Delta X = 0$ ft
$\Delta Y = 0$ ft

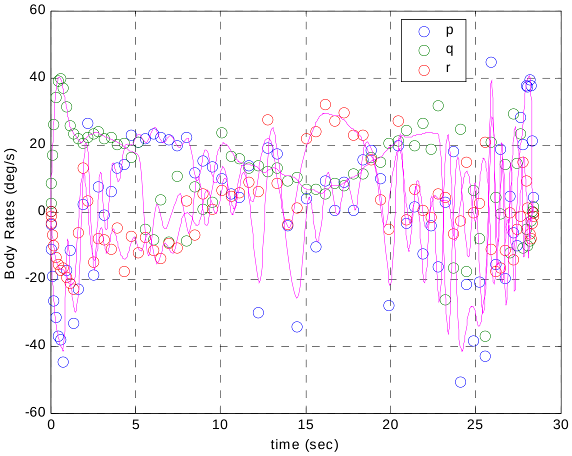
Unconstrained: (None)

80 Nodes (not bootstrapped)

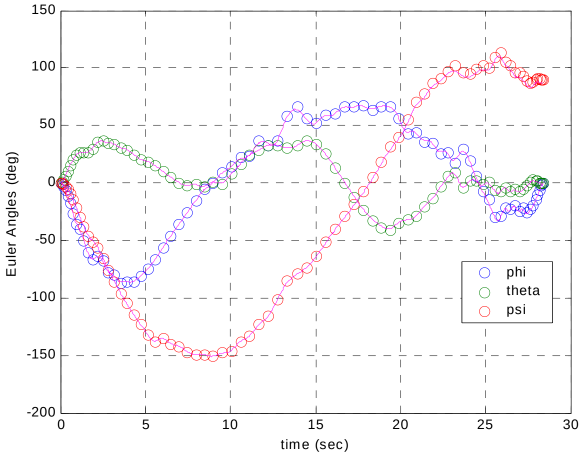
# of Iterations:	22679
Total Run Time:	72.08 min
Maneuver Time:	28.36 sec



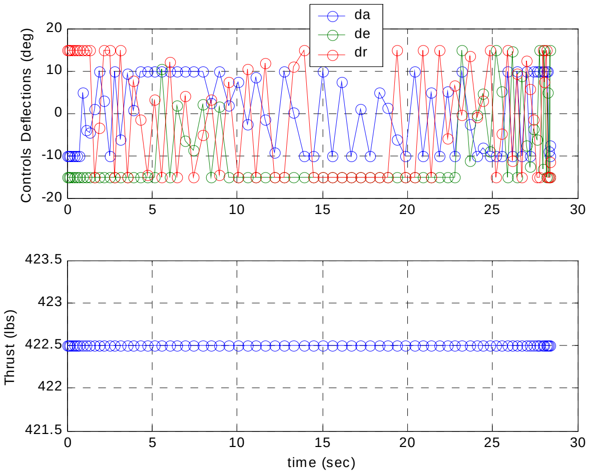




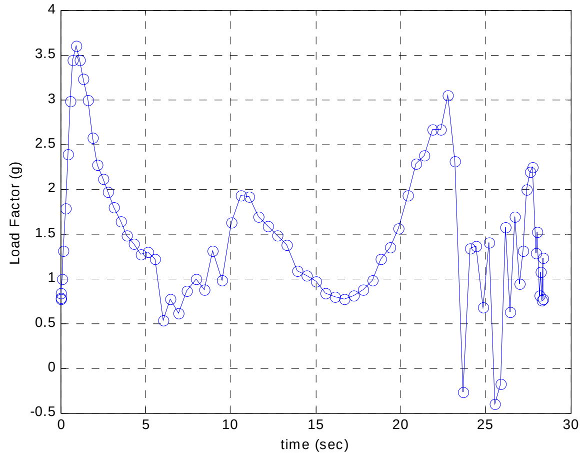
Body Rates (deg/s)
time (sec)
p
q
r
60
40
20
0
-20
-40
-60
0
5
10
15
20
25
30



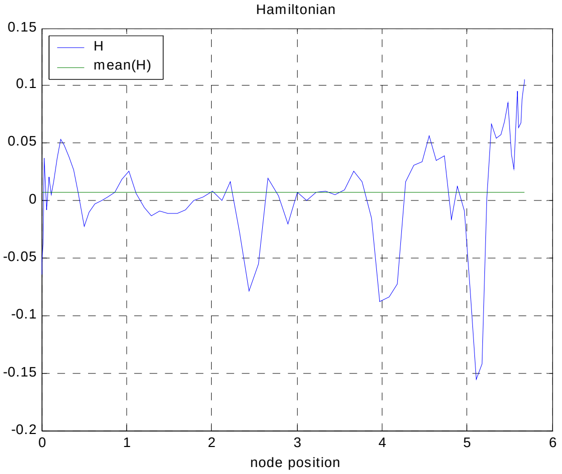
Euler Angles (deg)
time (sec)
phi
theta
psi
150
100
50
0
-50
-100
-150
-200
0
5
10
15
20
25
30



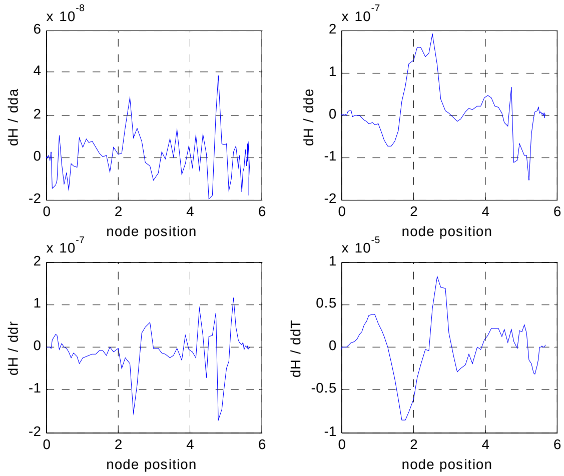
Controls Deflections (deg)
da
de
dr
20
10
0
-10
-20
Thrust (lbs)
423.5
423
422.5
422
421.5
time (sec)
0
5
10
15
20
25
30



Load Factor (g)
time (sec)
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
5
10
15
20
25
30



Hamiltonian
H
mean(H)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
node position
0
1
2
3
4
5
6



x 10^-8
dH / dda
x 10^-7
dH / dde
x 10^-7
dH / ddr
x 10^-5
dH / ddT
node position
0
2
4
6

G. POINTING MANEUVER

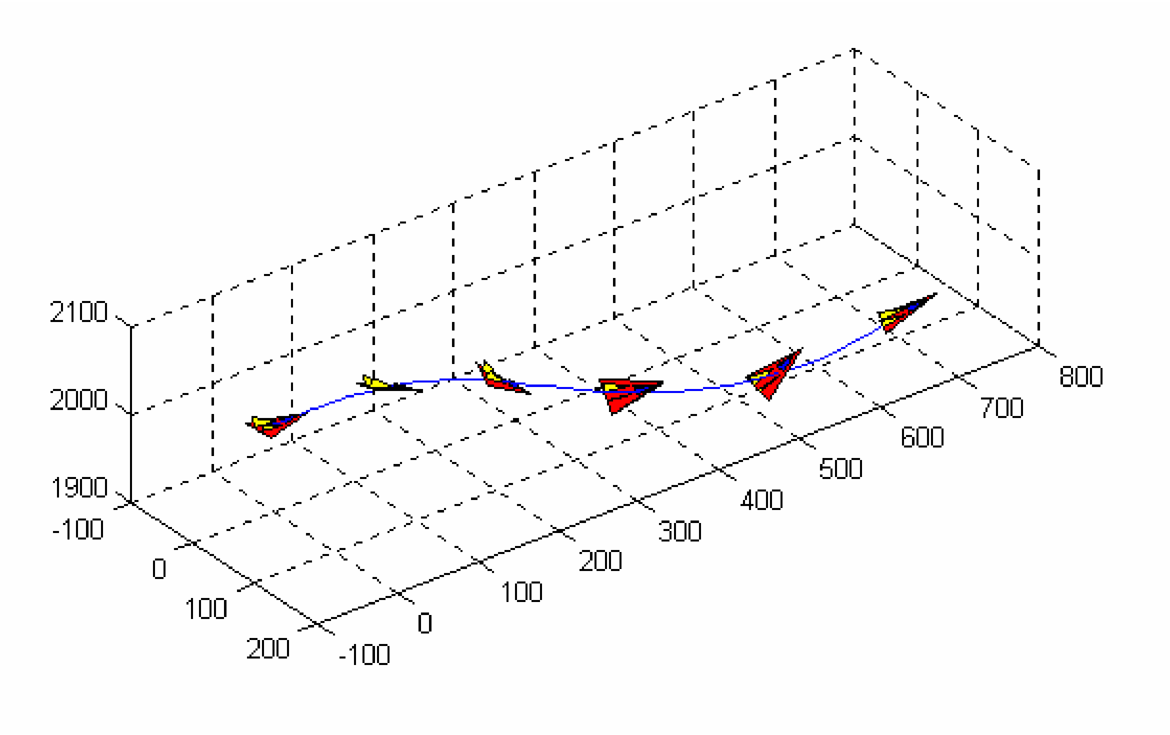
Delta: Δθ = 30°
Δψ = 30°

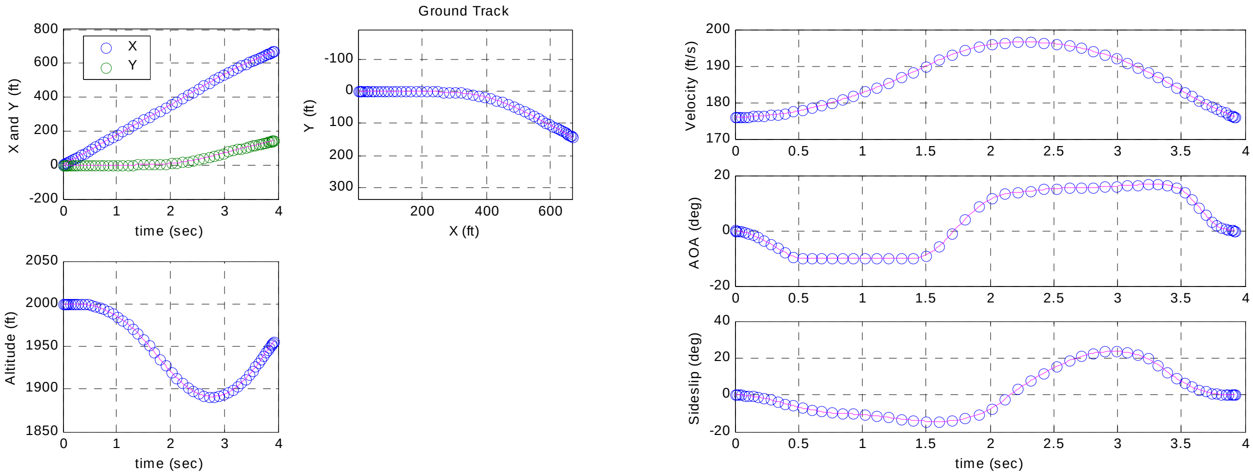
Unconstrained: X, Y (0 ± 1000 ft)
H (1000-3000 ft)

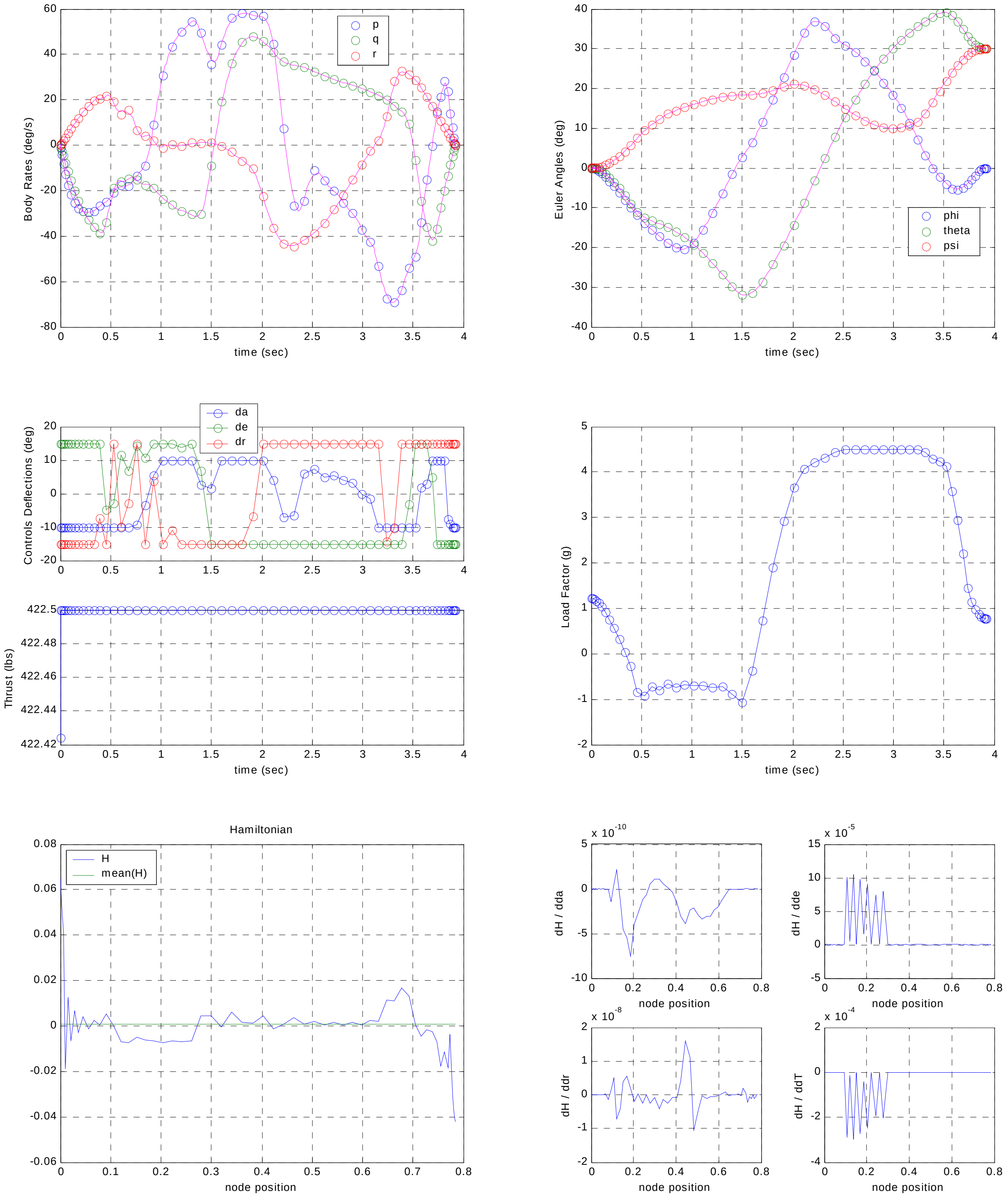
60 Nodes (not bootstrapped)

# of Iterations: 84154
Total Run Time: 122.49 min

Maneuver Time: 3.92 sec







Body Rates (deg/s)
60
40
20
0
-20
-40
-60
-80
p
q
r
time (sec)
0
0.5
1
1.5
2
2.5
3
3.5
4
Euler Angles (deg)
40
30
20
10
0
-10
-20
-30
-40
phi
theta
psi
time (sec)
Controls Deflections (deg)
20
10
0
-10
-20
da
de
dr
Thrust (lbs)
422.5
422.48
422.46
422.44
422.42
time (sec)
Load Factor (g)
5
4
3
2
1
0
-1
-2
time (sec)
Hamiltonian
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
H
mean(H)
node position
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x 10^-10
dH / dda
5
0
-5
-10
node position
x 10^-5
dH / dde
15
10
5
0
-5
node position
x 10^-8
dH / ddr
2
1
0
-1
-2
node position
x 10^-4
dH / ddT
2
0
-2
-4
node position

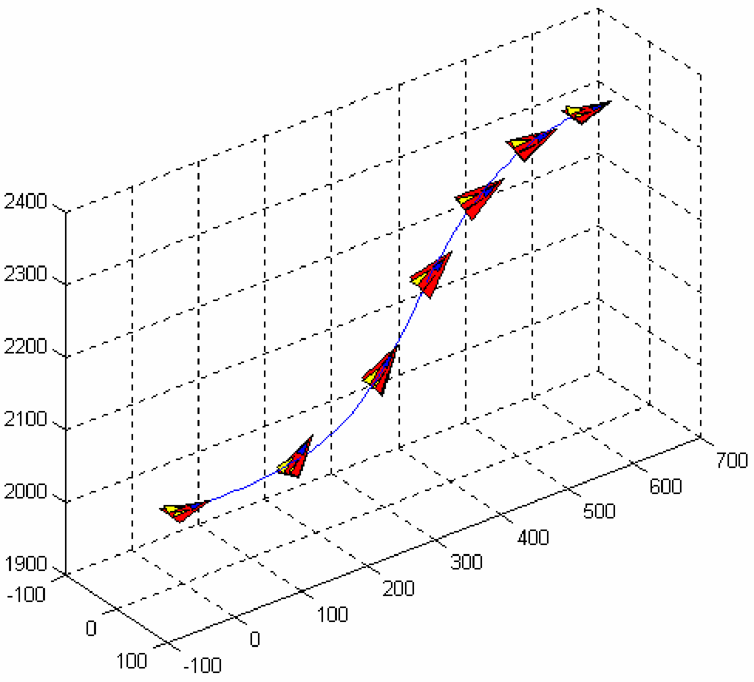
H. BRAKING MANEUVER

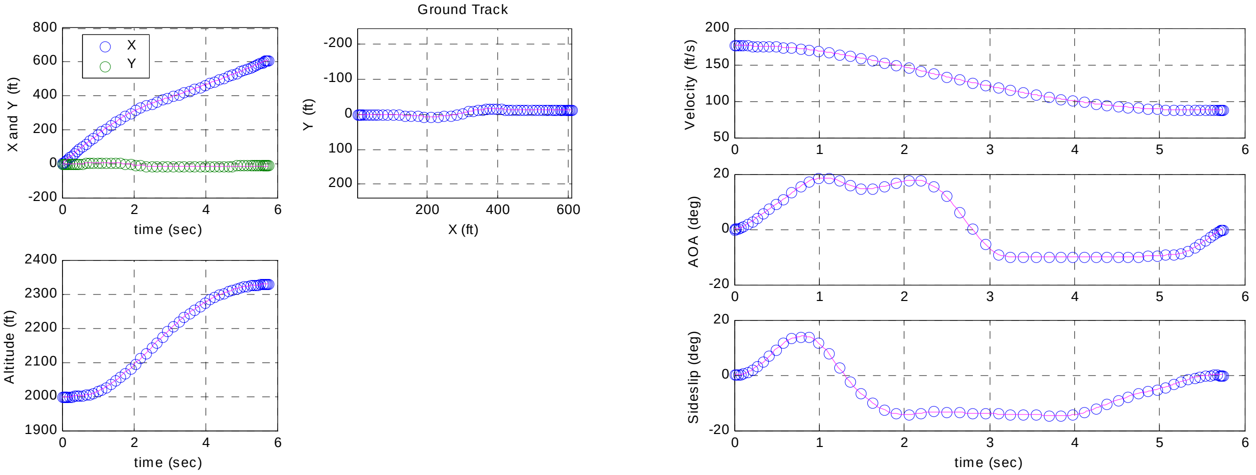
Delta: ΔV = -50% (-88 fps)

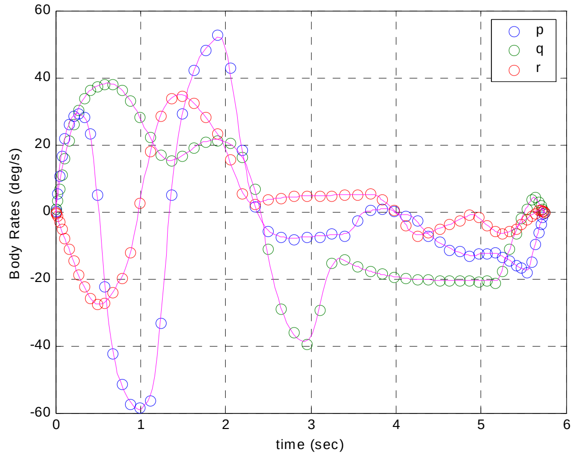
60 Nodes (not bootstrapped)

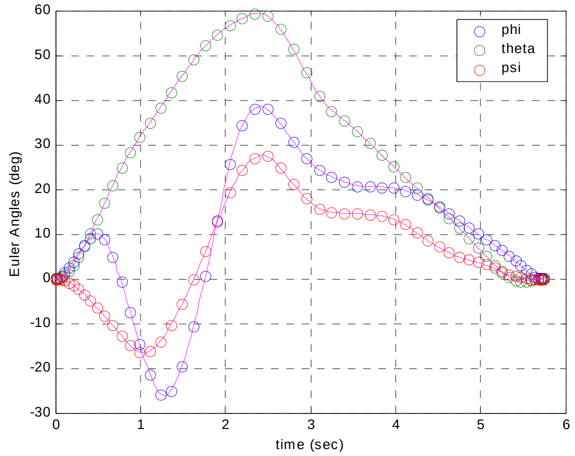
Unconstrained: X, Y (0 ± 1000 ft)
H (1000-3000 ft)

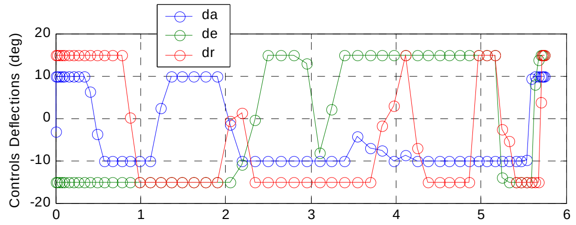
# of Iterations:	29307
Total Run Time:	42.36 min
Maneuver Time:	5.74 sec

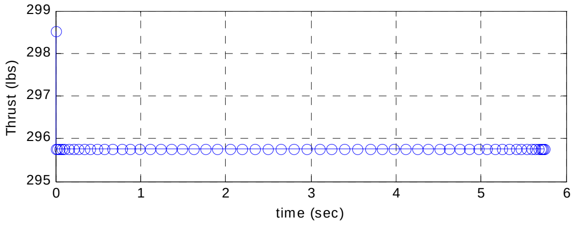


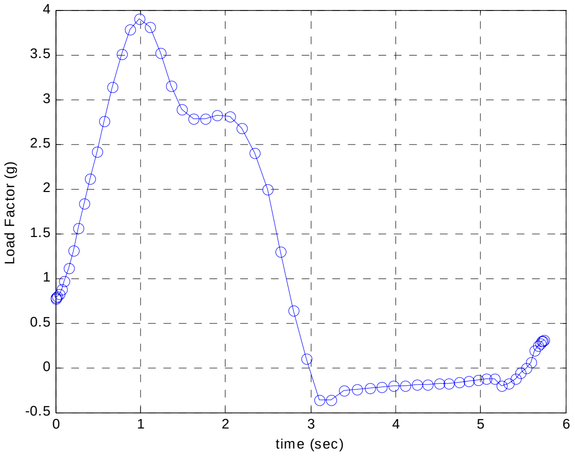


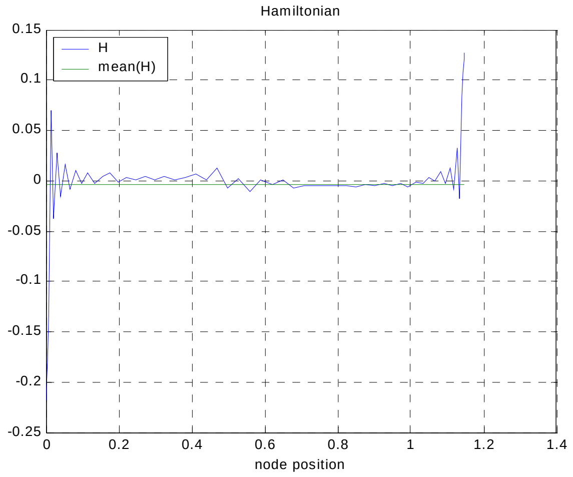

p
q
r
Body Rates (deg/s)
60
40
20
0
-20
-40
-60
0
1
2
3
4
5
6
time (sec)

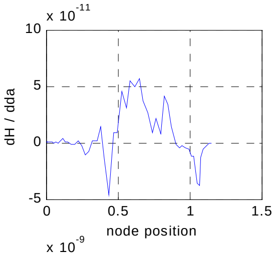

phi
theta
psi
Euler Angles (deg)
60
50
40
30
20
10
0
-10
-20
-30
0
1
2
3
4
5
6
time (sec)

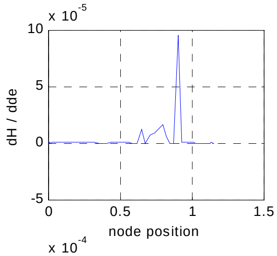

da
de
dr
Controls Deflections (deg)
20
10
0
-10
-20
0
1
2
3
4
5
6

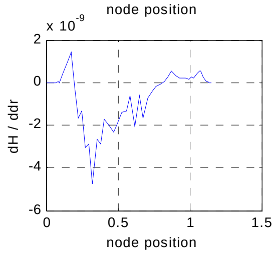

Thrust (lbs)
299
298
297
296
295
0
1
2
3
4
5
6
time (sec)

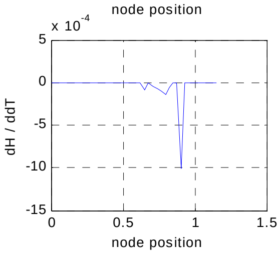

Load Factor (g)
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
1
2
3
4
5
6
time (sec)


Hamiltonian
H
mean(H)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
0.2
0.4
0.6
0.8
1
1.2
1.4
node position


x 10-11
dH / dda
10
5
0
-5
0
0.5
1
1.5
node position


x 10-5
dH / dde
10
5
0
-5
0
0.5
1
1.5
node position


x 10-9
dH / ddr
2
0
-2
-4
-6
0
0.5
1
1.5
node position


x 10-4
dH / ddT
5
0
-5
-10
-15
0
0.5
1
1.5
node position

APPENDIX G: HARV RESULTS

A. "CLASSIC HERBST"

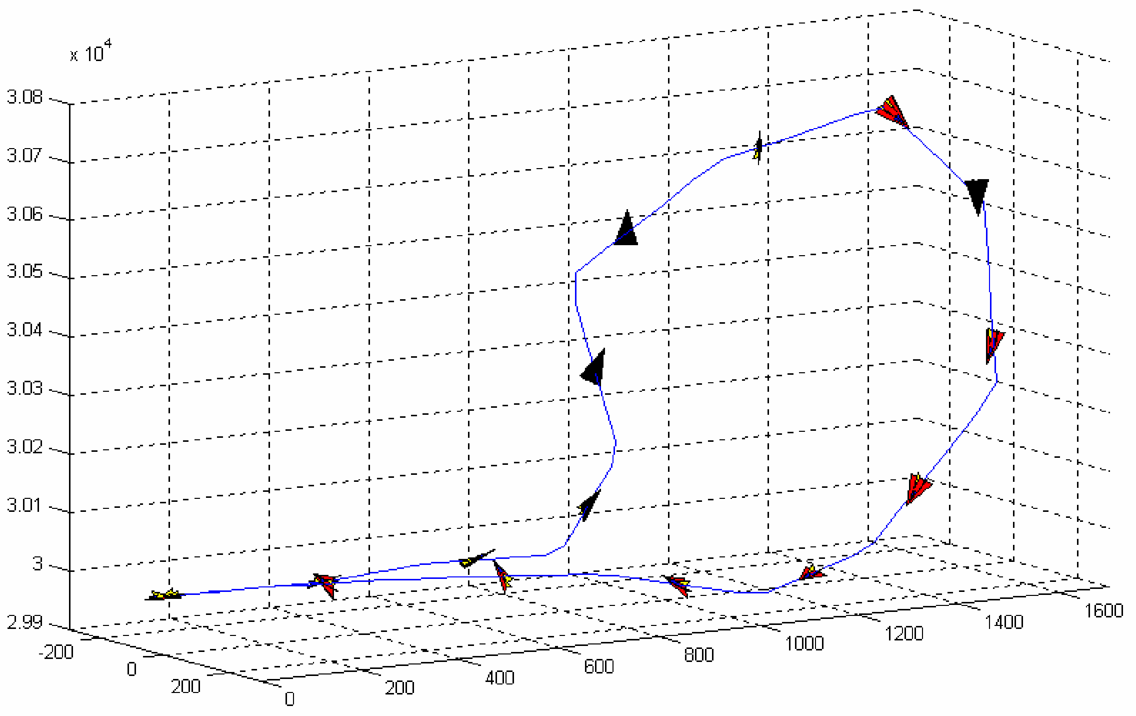
Delta: $\Delta\psi = 180°$
$\Delta H = 0$ ft
$\Delta V = 0$ fps
$\Delta X = 0$ ft
$\Delta Y = 0$ ft

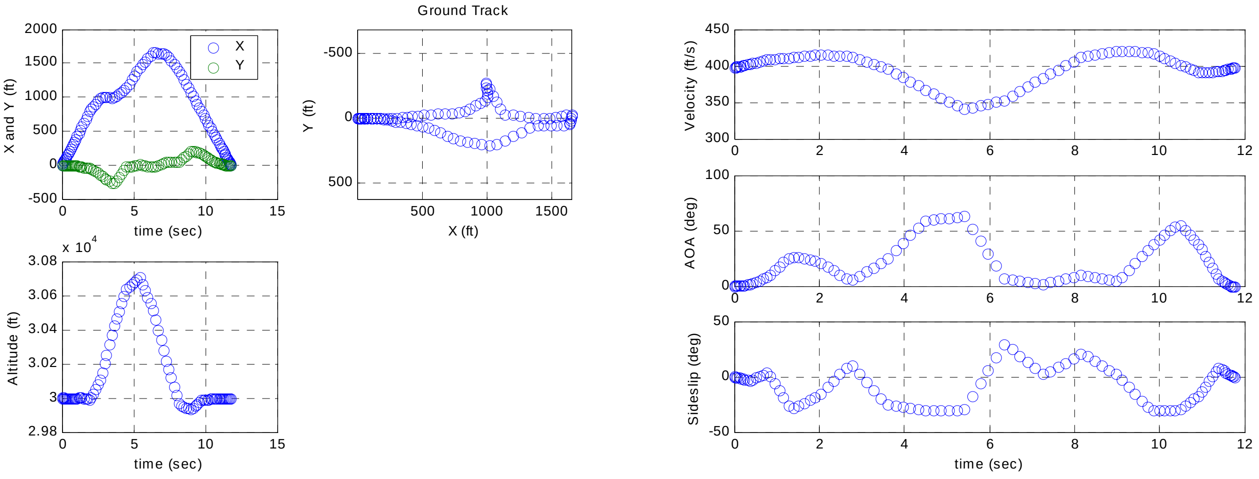
100 Nodes (bootstrapped from 20)

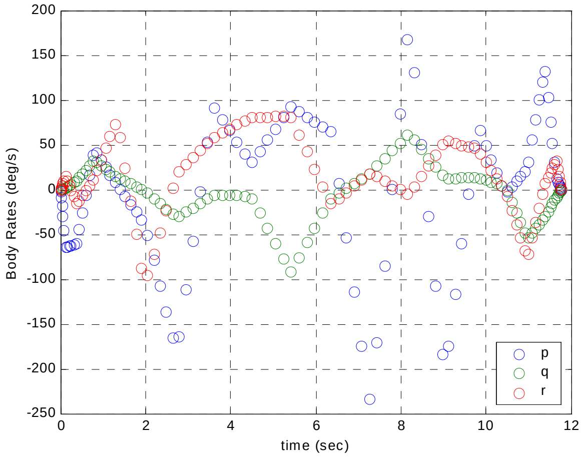
# of Iterations: 32312
Total Run Time: 444.76 min

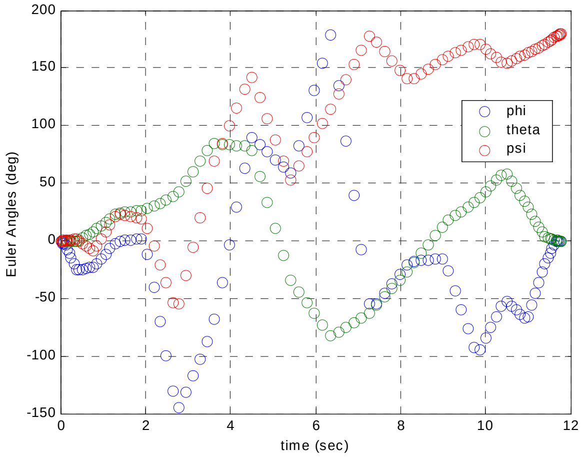
Maneuver Time: 11.75 sec

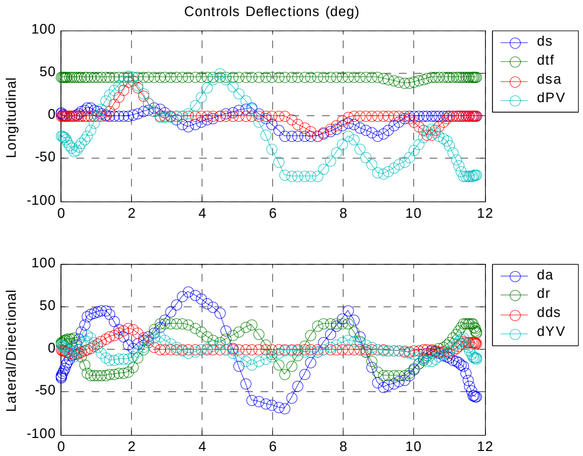
Unconstrained: δ (min-max deflection)

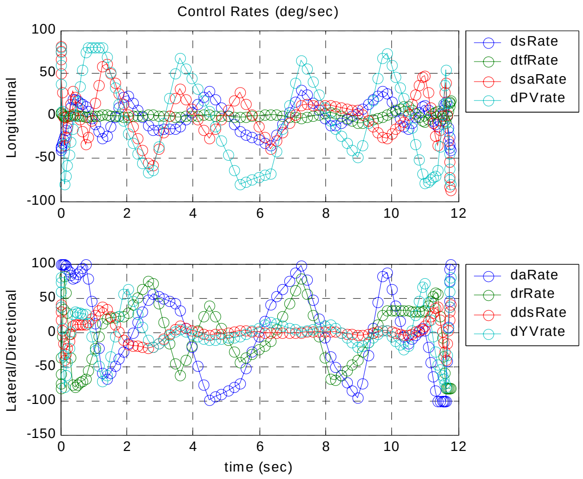


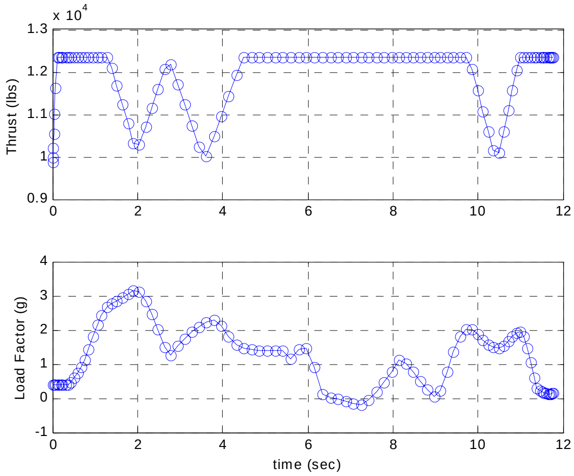


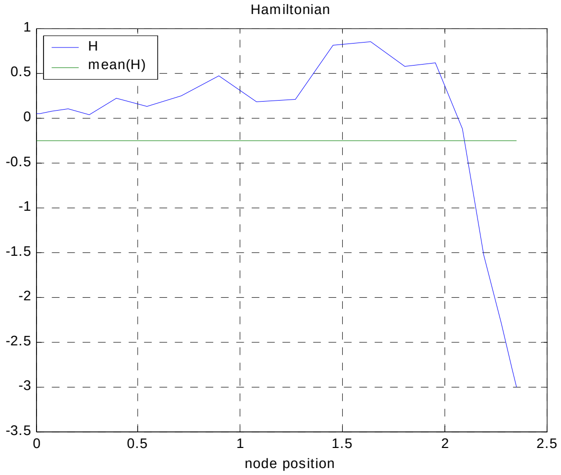

Body Rates (deg/s)
200
150
100
50
0
-50
-100
-150
-200
-250
0
2
4
6
8
10
12
time (sec)
p
q
r


Euler Angles (deg)
200
150
100
50
0
-50
-100
-150
0
2
4
6
8
10
12
time (sec)
phi
theta
psi


Controls Deflections (deg)
Longitudinal
100
50
0
-50
-100
ds
dtf
dsa
dPV
Lateral/Directional
100
50
0
-50
-100
da
dr
dds
dYV
0
2
4
6
8
10
12


Control Rates (deg/sec)
Longitudinal
100
50
0
-50
-100
dsRate
dtfRate
dsaRate
dPVrate
Lateral/Directional
100
50
0
-50
-100
-150
daRate
drRate
ddsRate
dYVrate
0
2
4
6
8
10
12
time (sec)


x 10^4
1.3
1.2
1.1
1
0.9
Thrust (lbs)
Load Factor (g)
4
3
2
1
0
-1
0
2
4
6
8
10
12
time (sec)


Hamiltonian
H
mean(H)
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
0.5
1
1.5
2
2.5
node position

B. “FALCON TURN”

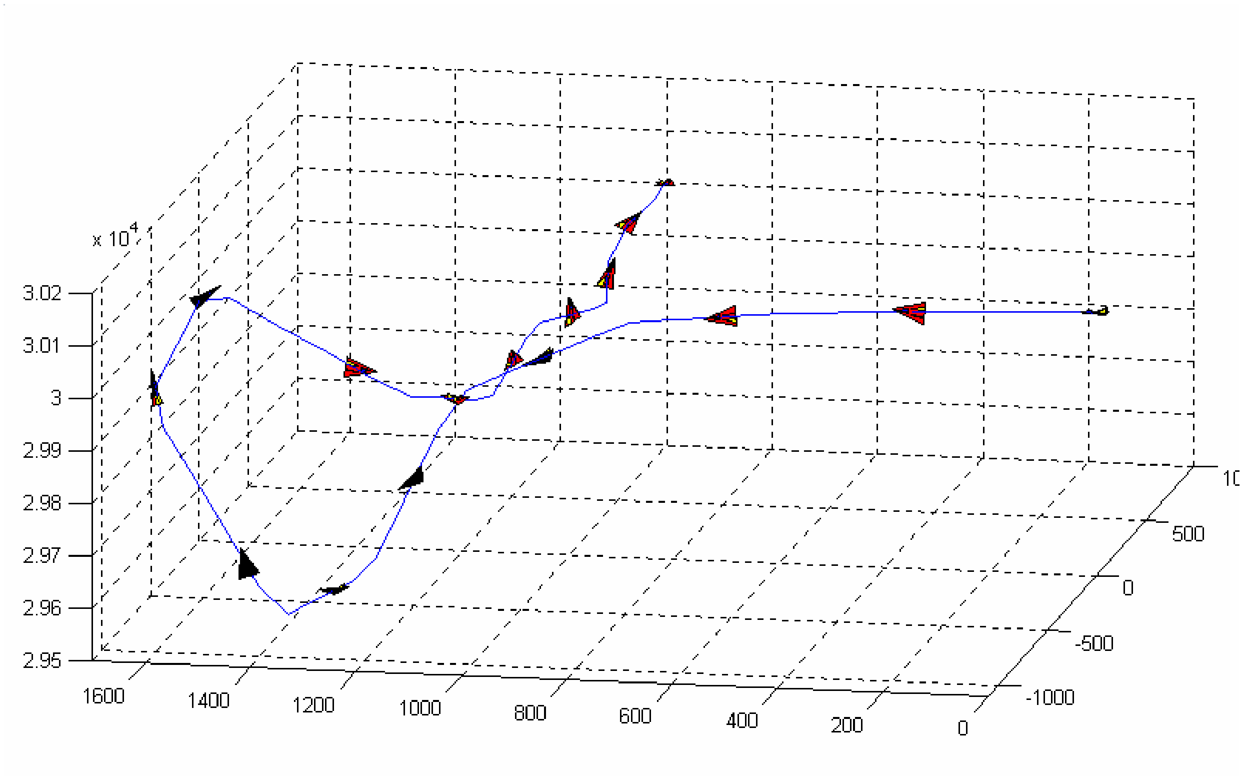
Delta: $\Delta\psi$ = 90°
ΔH = 0 ft
ΔV = 0 fps

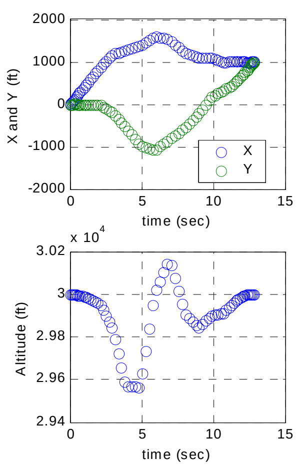
80 Nodes (bootstrapped from 20)

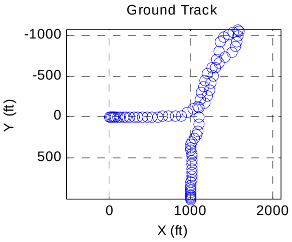
# of Iterations: 24383
Total Run Time: 237.73 min

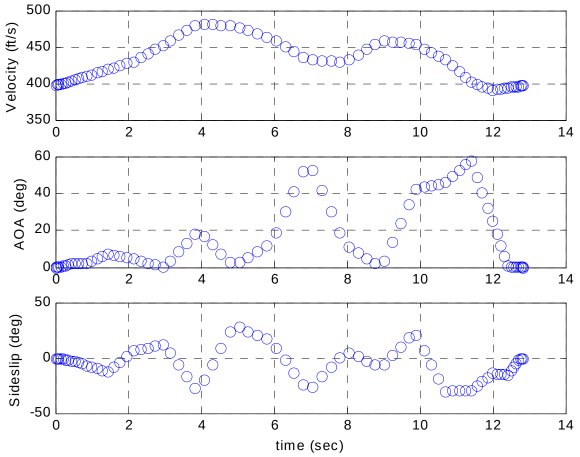
Unconstrained: X, Y (0 ± 1000 ft)
δ (min-max deflection)

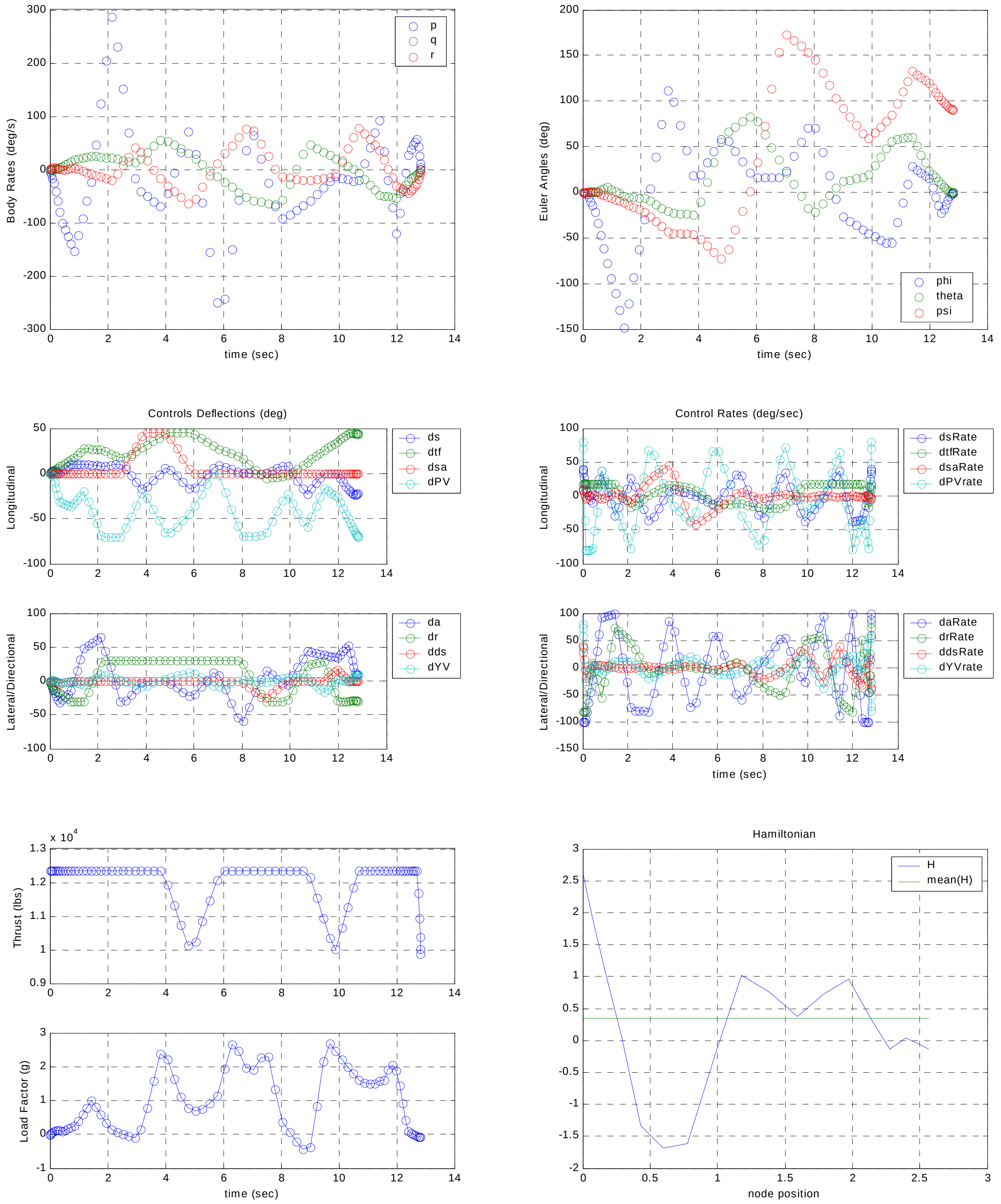
Maneuver Time: 12.84 sec











Body Rates (deg/s)
p
q
r
time (sec)
Euler Angles (deg)
phi
theta
psi
time (sec)
Controls Deflections (deg)
Longitudinal
ds
dtf
dsa
dPV
Lateral/Directional
da
dr
dds
dYV
Control Rates (deg/sec)
Longitudinal
dsRate
dtfRate
dsaRate
dPVrate
Lateral/Directional
daRate
drRate
ddsRate
dYVrate
time (sec)
x 10^4
Thrust (lbs)
Load Factor (g)
time (sec)
Hamiltonian
H
mean(H)
node position

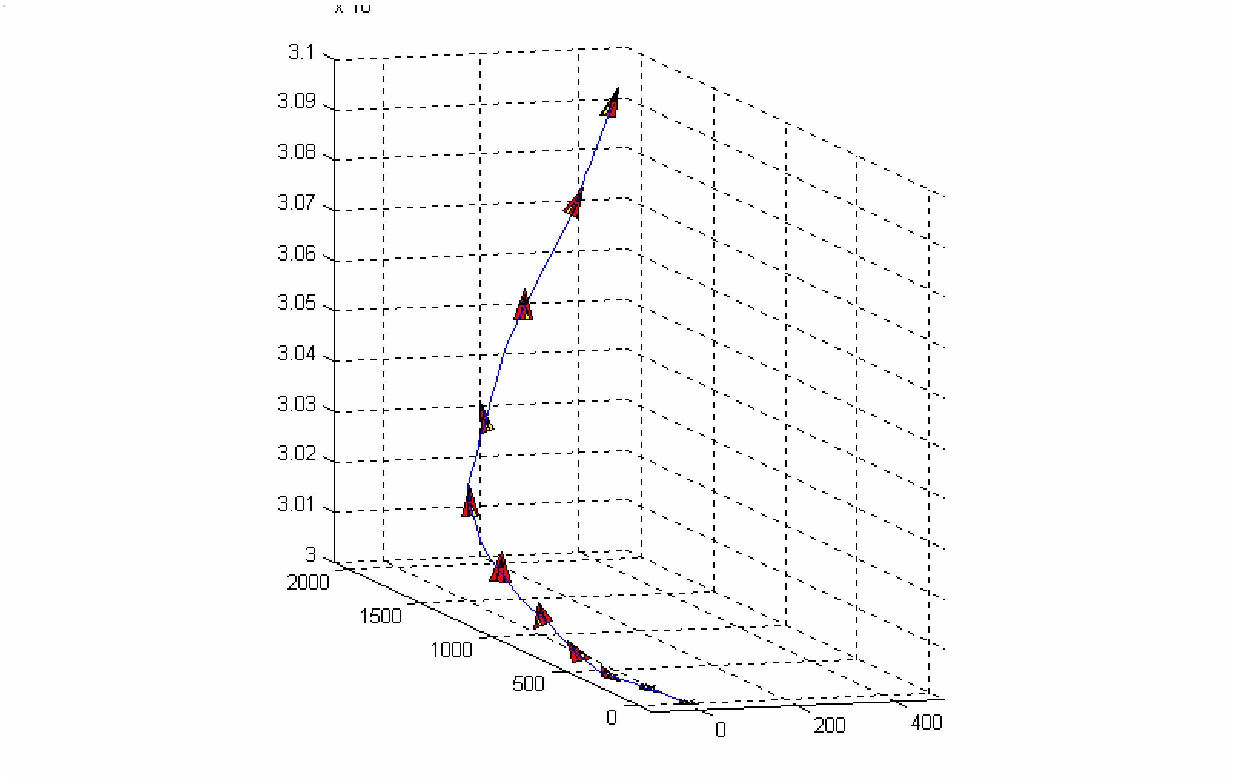
C. “POINT AND SHOOT B”

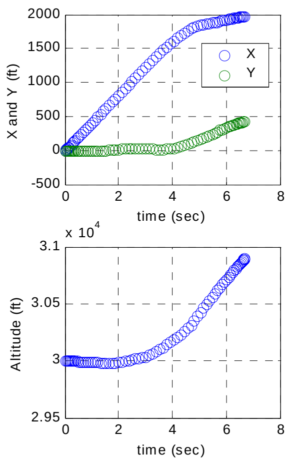
Delta: Δθ = 60°
Δψ = 60°

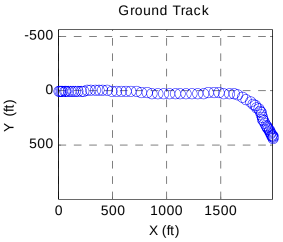
80 Nodes (bootstrapped from 20)

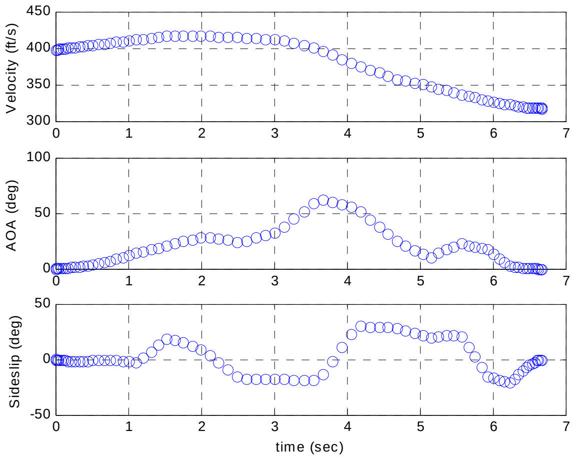
Unconstrained: X, Y (0 ± 3000 ft)
H (27k-33k ft)
V (398 ± 100 fps)
δ (min-max deflection)

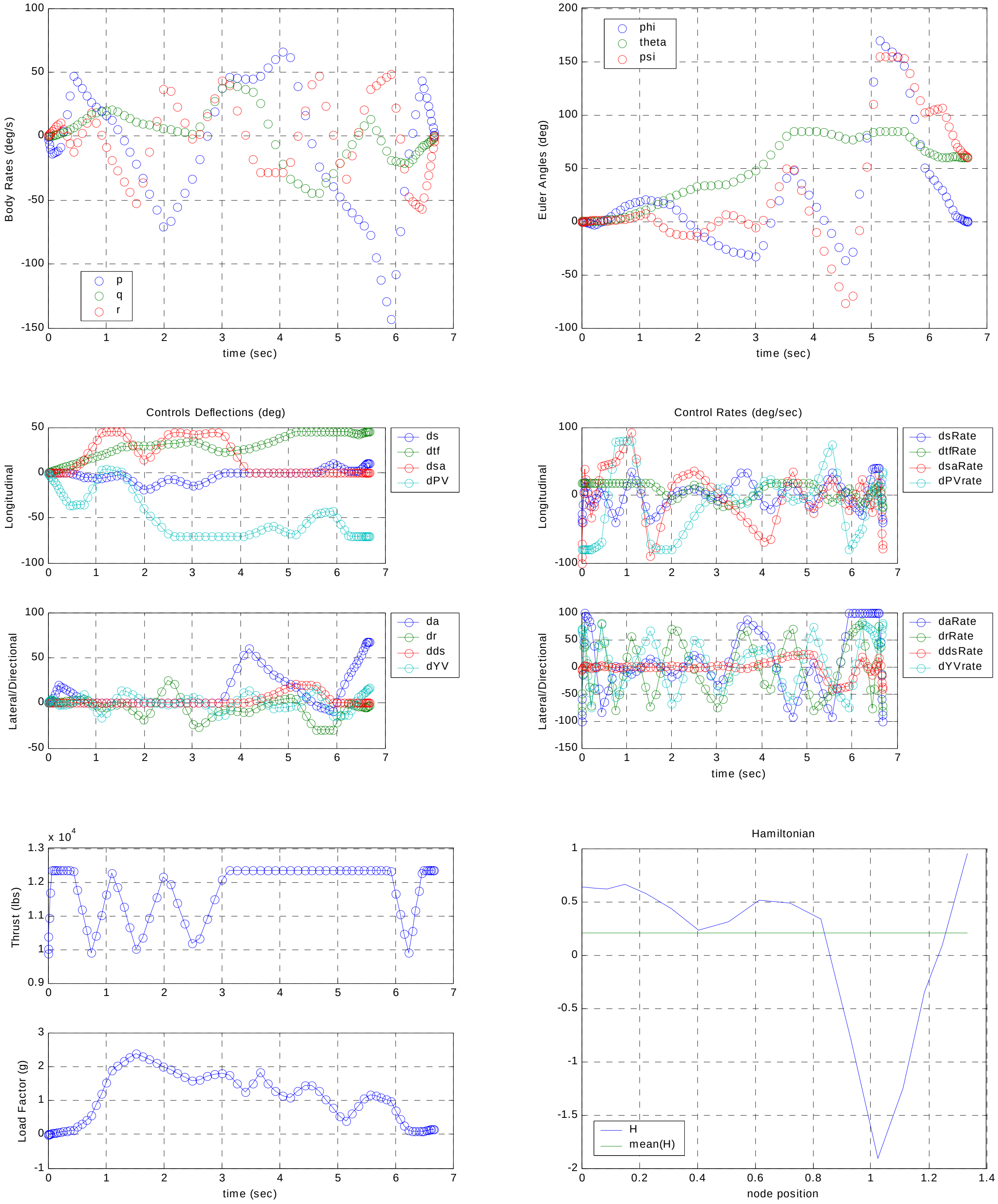
# of Iterations:	14053
Total Run Time:	456.02 min
Maneuver Time:	6.68 sec











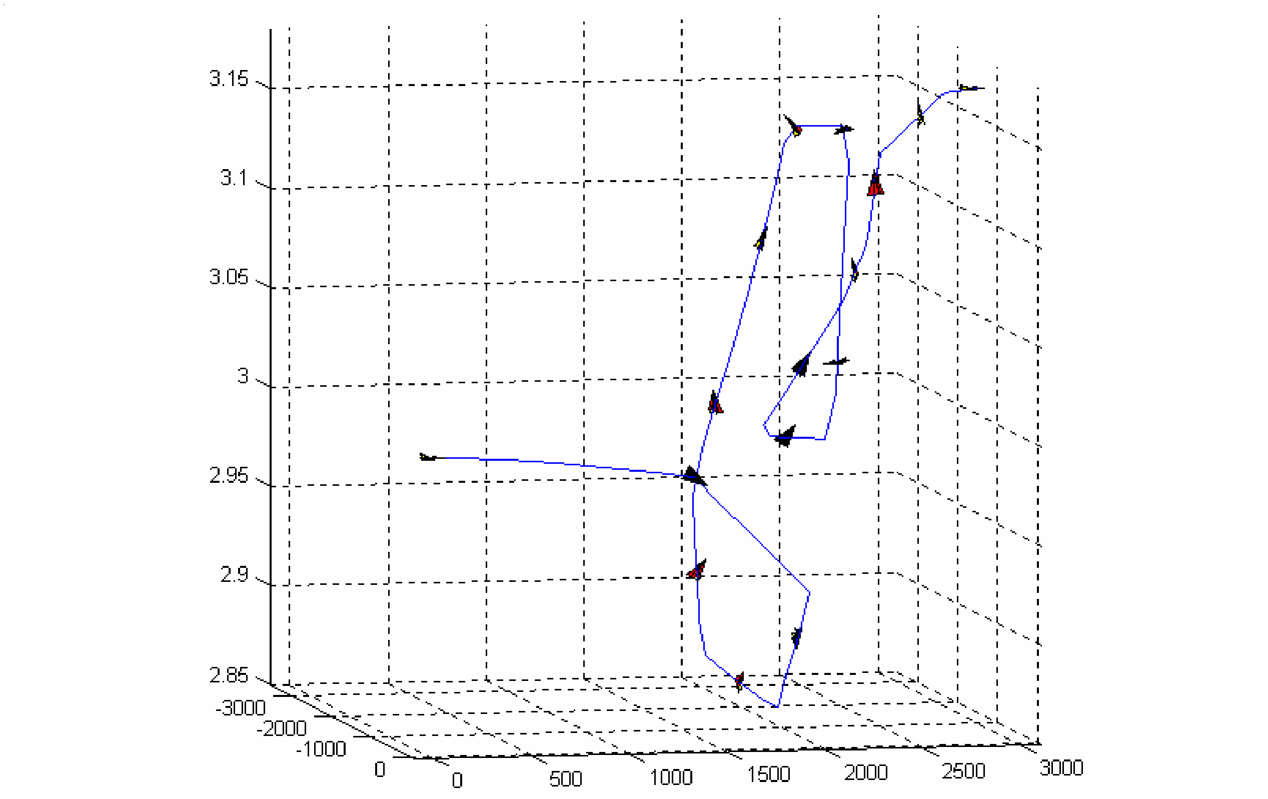
Body Rates (deg/s)
100
50
0
-50
-100
-150
p
q
r
time (sec)
0
1
2
3
4
5
6
7
Euler Angles (deg)
200
150
100
50
0
-50
-100
phi
theta
psi
time (sec)
Controls Deflections (deg)
Longitudinal
ds
dtf
dsa
dPV
Lateral/Directional
da
dr
dds
dYV
Control Rates (deg/sec)
Longitudinal
dsRate
dtfRate
dsaRate
dPVrate
Lateral/Directional
daRate
drRate
ddsRate
dYVrate
time (sec)
x 10^4
1.3
1.2
1.1
1
0.9
Thrust (lbs)
Load Factor (g)
3
2
1
0
-1
time (sec)
Hamiltonian
1
0.5
0
-0.5
-1
-1.5
-2
H
mean(H)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
node position

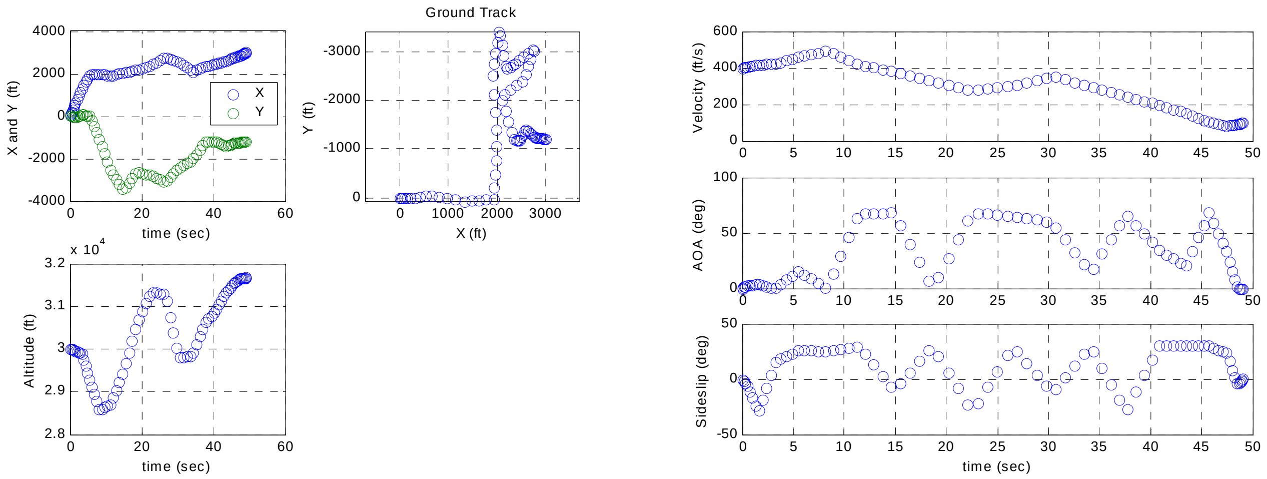
D. “CRAZY STRAW”

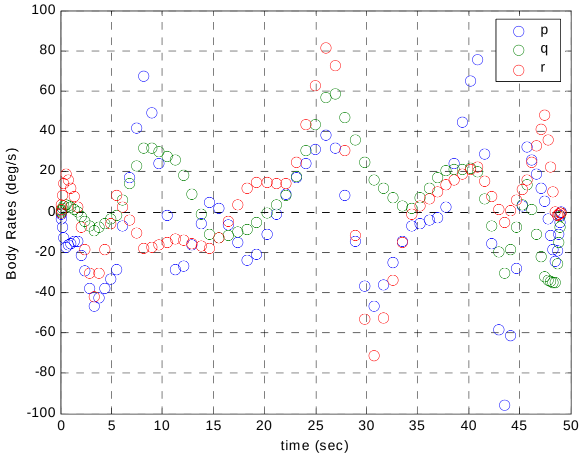
Delta: ΔV = -75% (-298 fps)   80 Nodes (bootstrapped from 20)

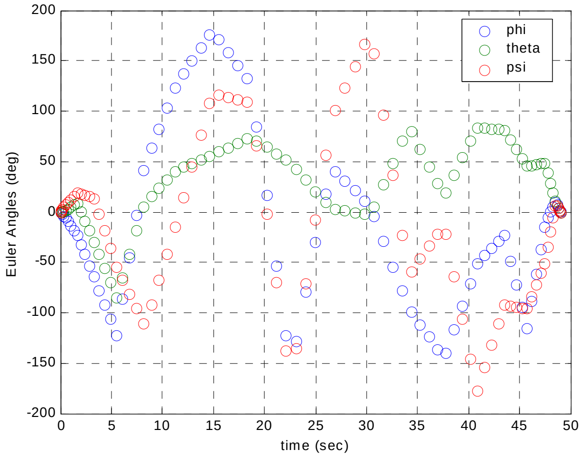
Unconstrained: X, Y (0 ± 3000 ft)   # of Iterations: 22523
H (27k-33k ft)   Total Run Time: 151.53 min
δ (min-max deflection)

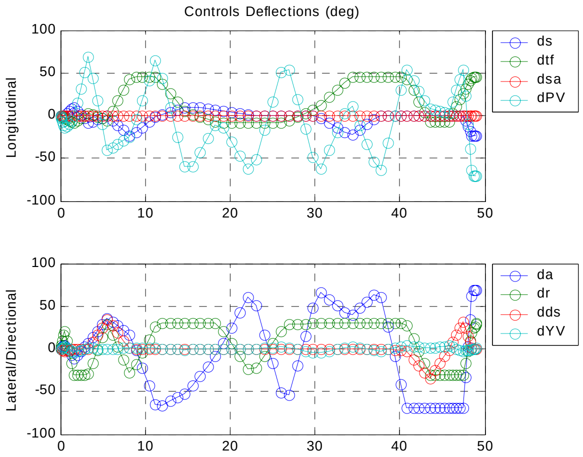
Maneuver Time: 48.99 sec
(Aircraft scale 2:1 in figure below.)

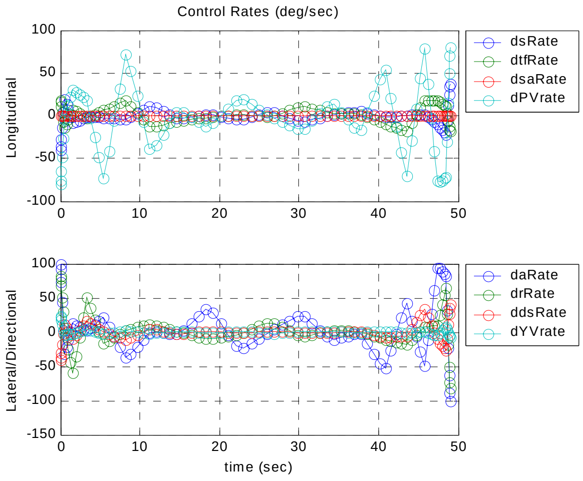


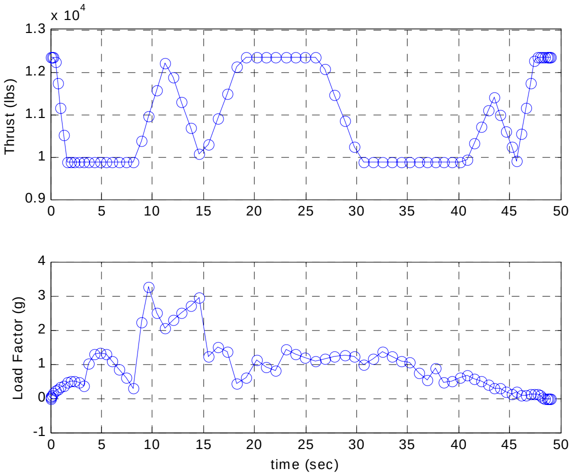


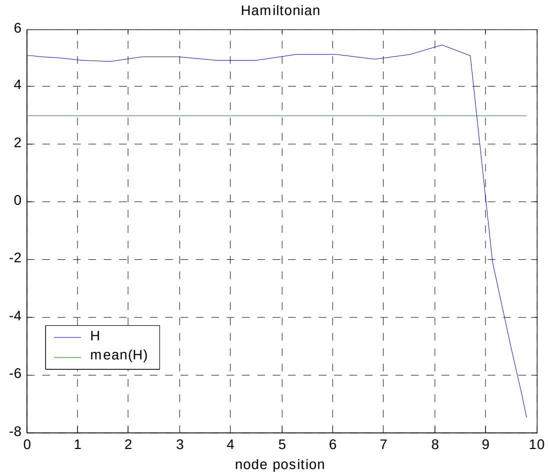

Body Rates (deg/s)
time (sec)
p
q
r
100
80
60
40
20
0
-20
-40
-60
-80
-100
0
5
10
15
20
25
30
35
40
45
50


Euler Angles (deg)
time (sec)
phi
theta
psi
200
150
100
50
0
-50
-100
-150
-200
0
5
10
15
20
25
30
35
40
45
50


Controls Deflections (deg)
Longitudinal
ds
dtf
dsa
dPV
Lateral/Directional
da
dr
dds
dYV
100
50
0
-50
-100
0
10
20
30
40
50


Control Rates (deg/sec)
Longitudinal
dsRate
dtfRate
dsaRate
dPVrate
Lateral/Directional
daRate
drRate
ddsRate
dYVrate
time (sec)
100
50
0
-50
-100
-150
0
10
20
30
40
50


x 10^4
Thrust (lbs)
1.3
1.2
1.1
1
0.9
Load Factor (g)
4
3
2
1
0
-1
time (sec)
0
5
10
15
20
25
30
35
40
45
50


Hamiltonian
H
mean(H)
node position
6
4
2
0
-2
-4
-6
-8
0
1
2
3
4
5
6
7
8
9
10

APPENDIX H: UCAV-X RESULTS

A. "LAZY EIGHT"

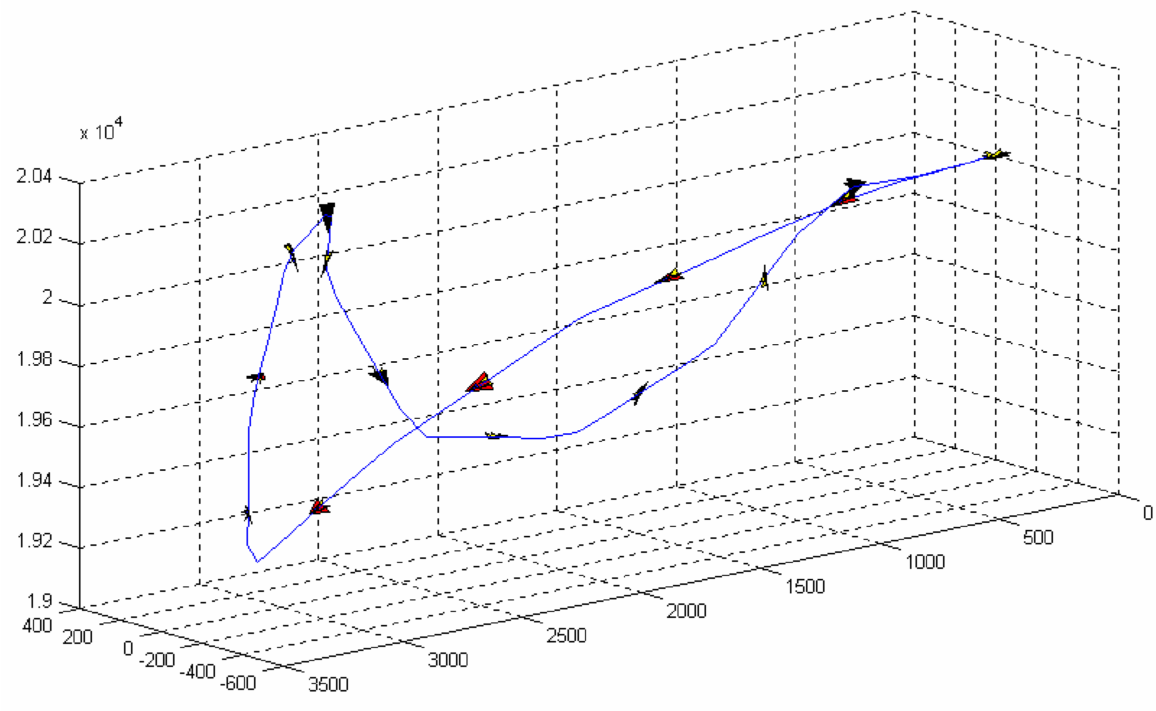
Delta: $\Delta\psi$ = 180°
ΔH = 0 ft
ΔV = 0 fps
ΔX = 0 ft
ΔY = 0 ft

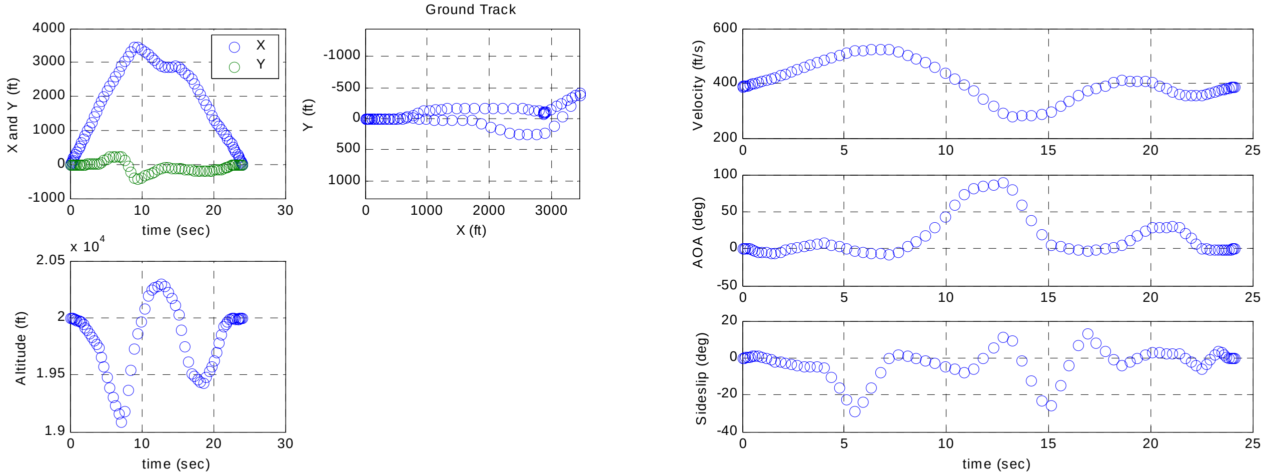
Unconstrained: δ (min-max deflection)

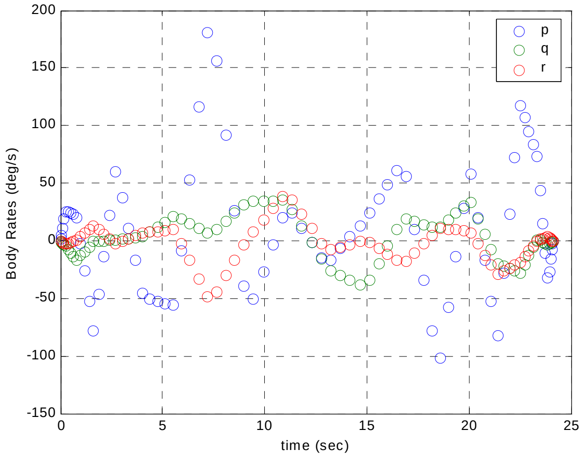
80 Nodes (bootstrapped from 20)

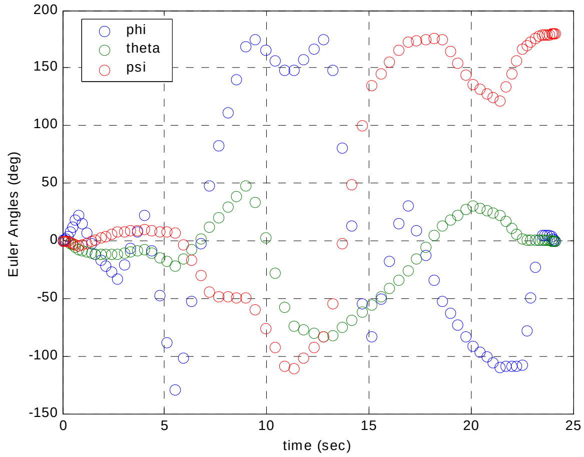
# of Iterations: 11630
Total Run Time: 102.59 min

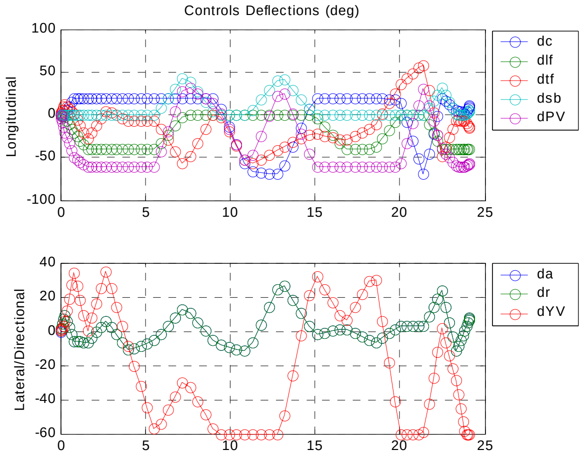
Maneuver Time: 24.08 sec
(Aircraft scale 1.5:1 in figure below.)

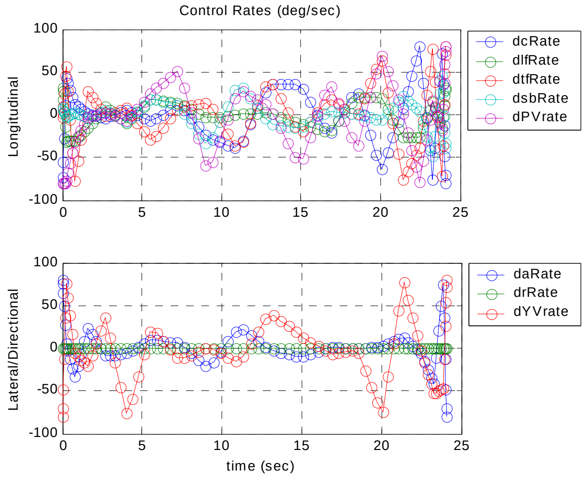


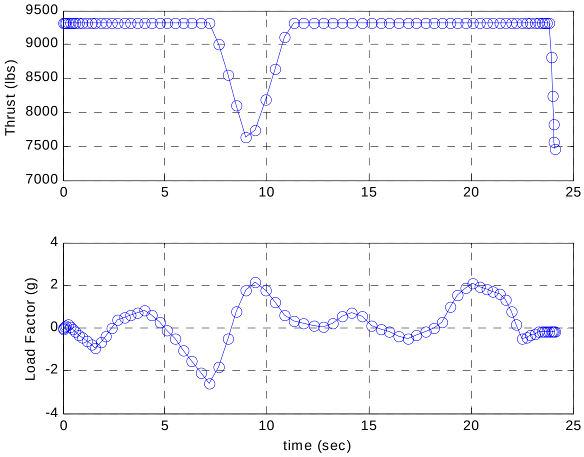


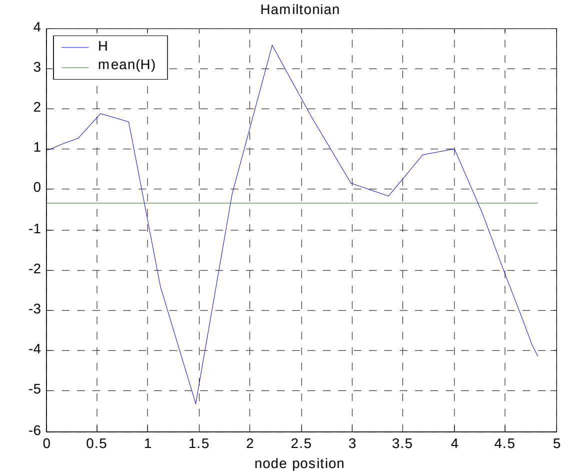

Body Rates (deg/s)
time (sec)
p
q
r
200
150
100
50
0
-50
-100
-150
0
5
10
15
20
25


Euler Angles (deg)
time (sec)
phi
theta
psi
200
150
100
50
0
-50
-100
-150
0
5
10
15
20
25


Controls Deflections (deg)
Longitudinal
dc
dlf
dtf
dsb
dPV
100
50
0
-50
-100
Lateral/Directional
da
dr
dYV
40
20
0
-20
-40
-60
0
5
10
15
20
25


Control Rates (deg/sec)
Longitudinal
dcRate
dlfRate
dtfRate
dsbRate
dPVrate
100
50
0
-50
-100
Lateral/Directional
daRate
drRate
dYVrate
100
50
0
-50
-100
time (sec)
0
5
10
15
20
25


Thrust (lbs)
9500
9000
8500
8000
7500
7000
Load Factor (g)
4
2
0
-2
-4
time (sec)
0
5
10
15
20
25


Hamiltonian
H
mean(H)
4
3
2
1
0
-1
-2
-3
-4
-5
-6
node position
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

B. “CLOVERLEAF”

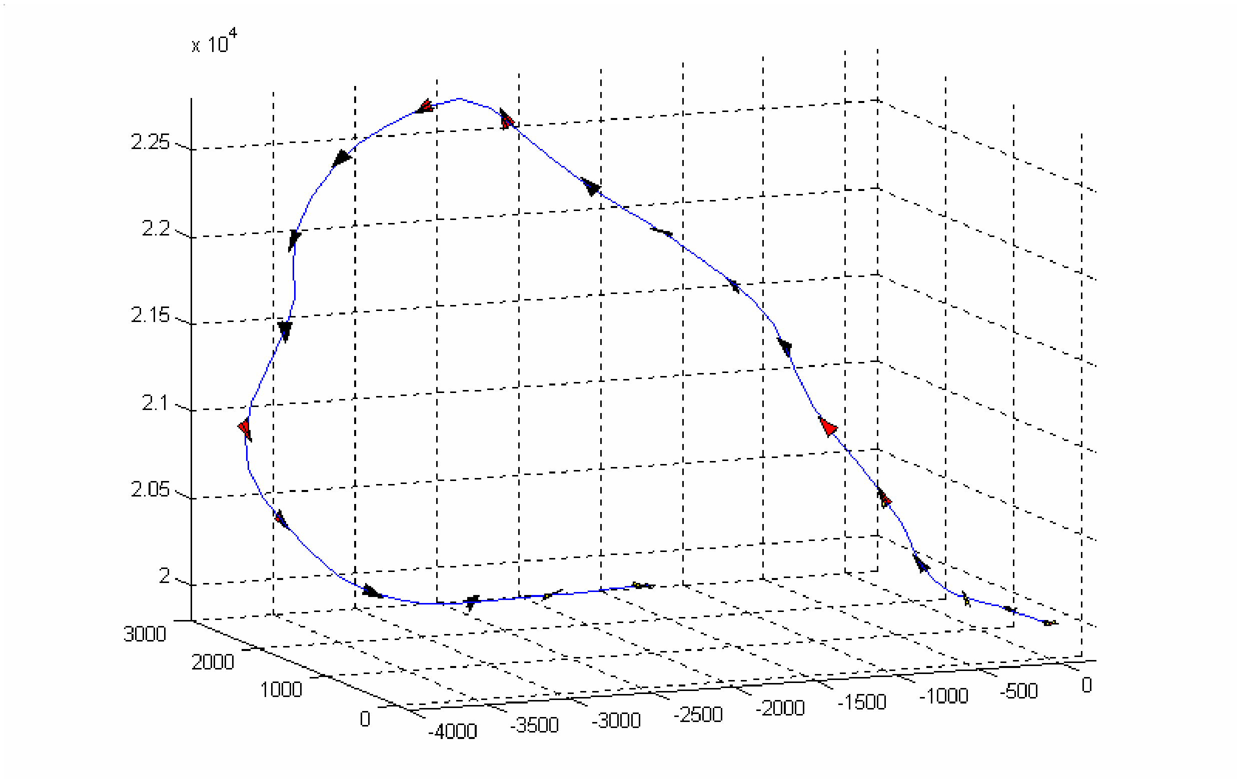
Delta: Δψ = 90°
ΔH = 0 ft
ΔV = 0 fps

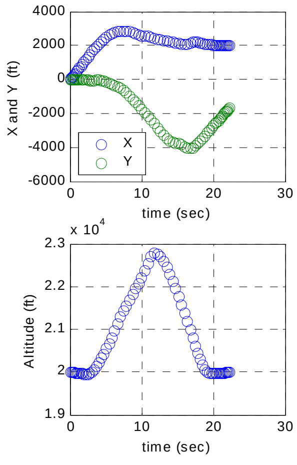
80 Nodes (bootstrapped from 20)

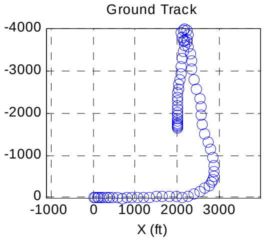
# of Iterations: 34501
Total Run Time: 716.24 min

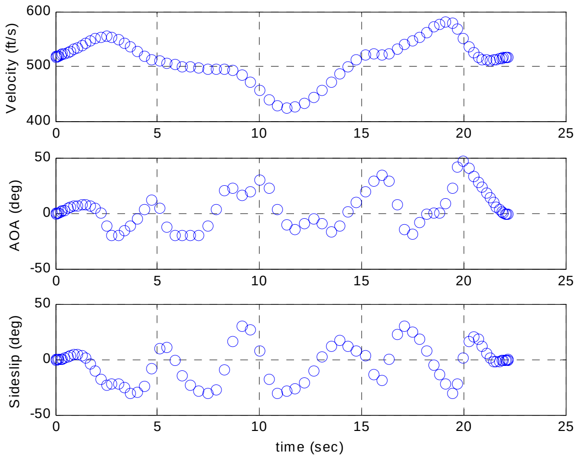
Unconstrained: X, Y (0 ± 2000 ft)
δ (min-max deflection)

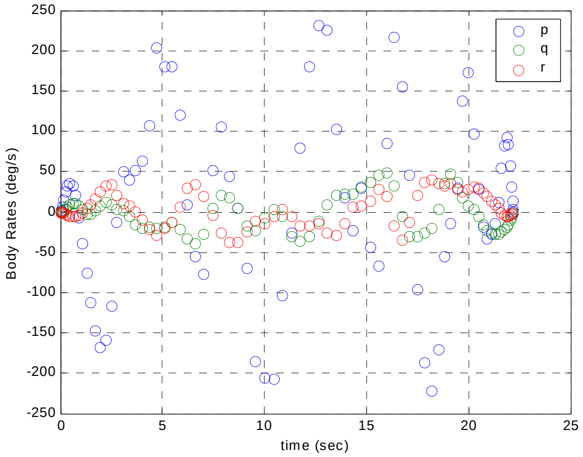
Maneuver Time: 22.18 sec
(Aircraft scale 2:1 in figure below.)

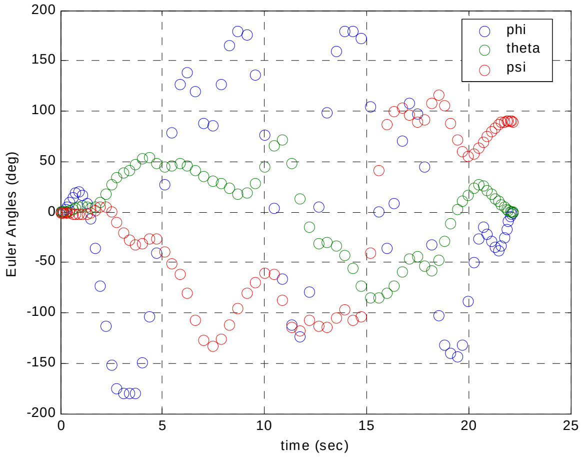




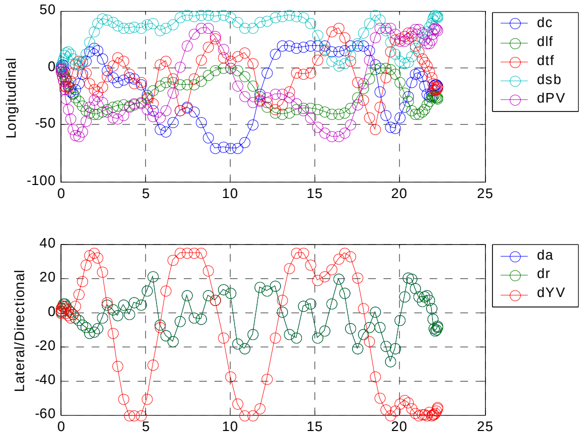




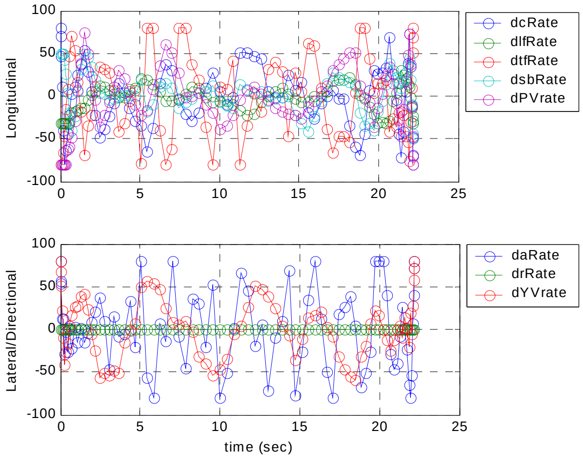


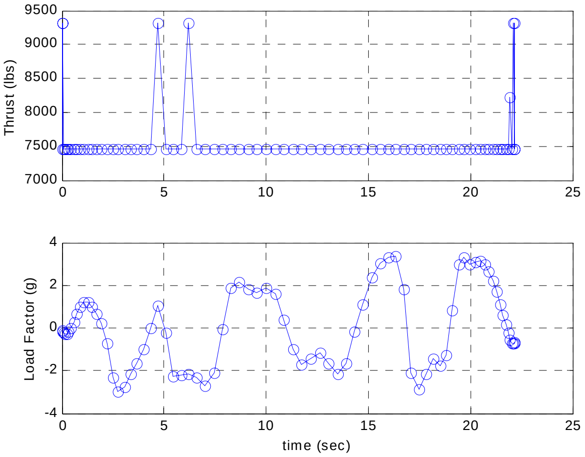


Controls Deflections (deg)

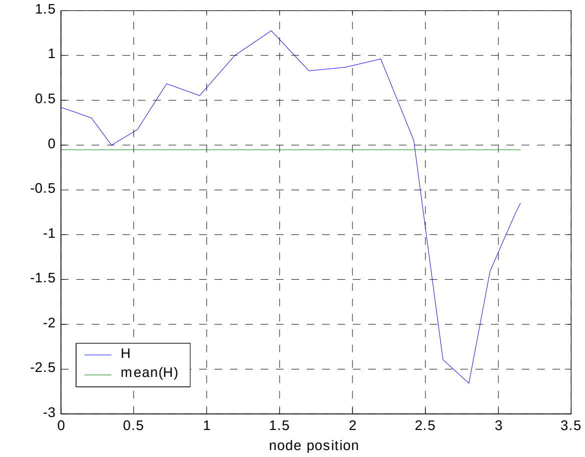


Control Rates (deg/sec)





Hamiltonian



C. “POINT AND SHOOT C”

Delta: Δθ = 60°
Δψ = 60°

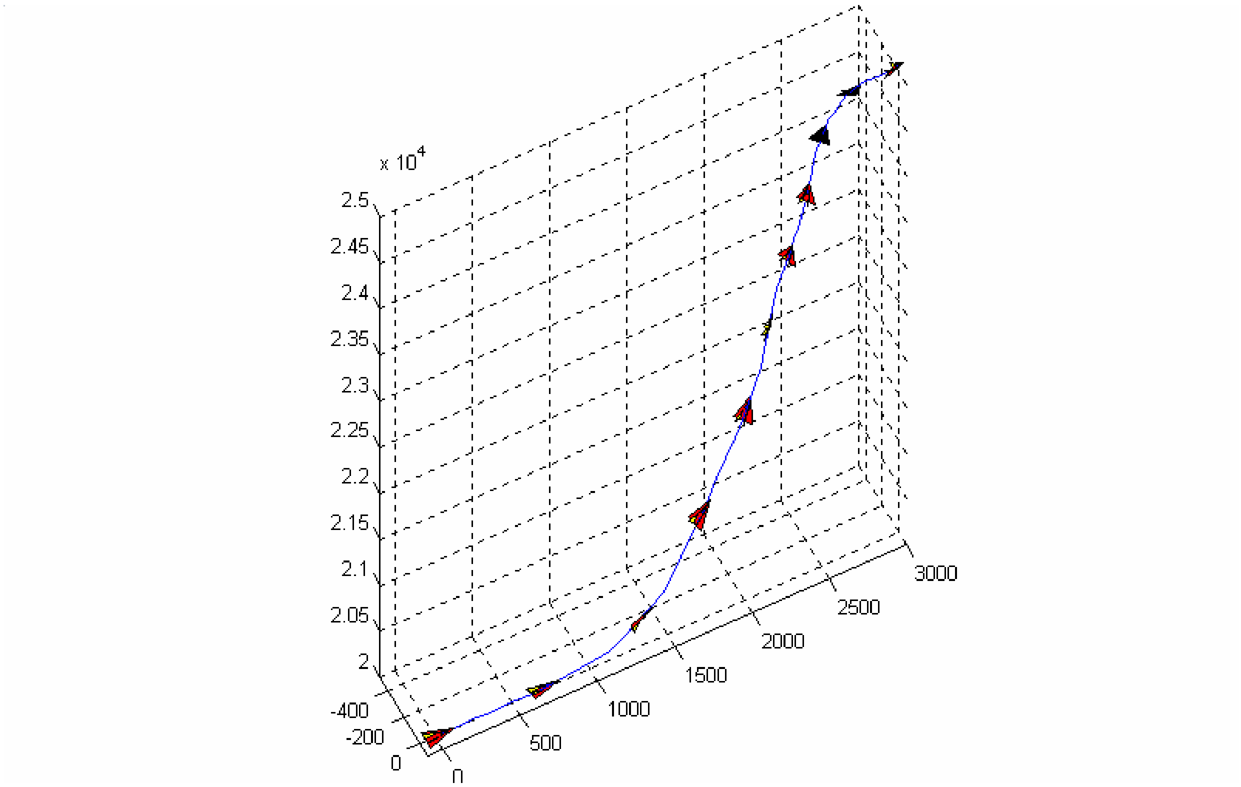
Unconstrained: X, Y (0 ± 5000 ft)
H (25k-35k ft)
V (518 ± 200 fps)
δ (min-max deflection)

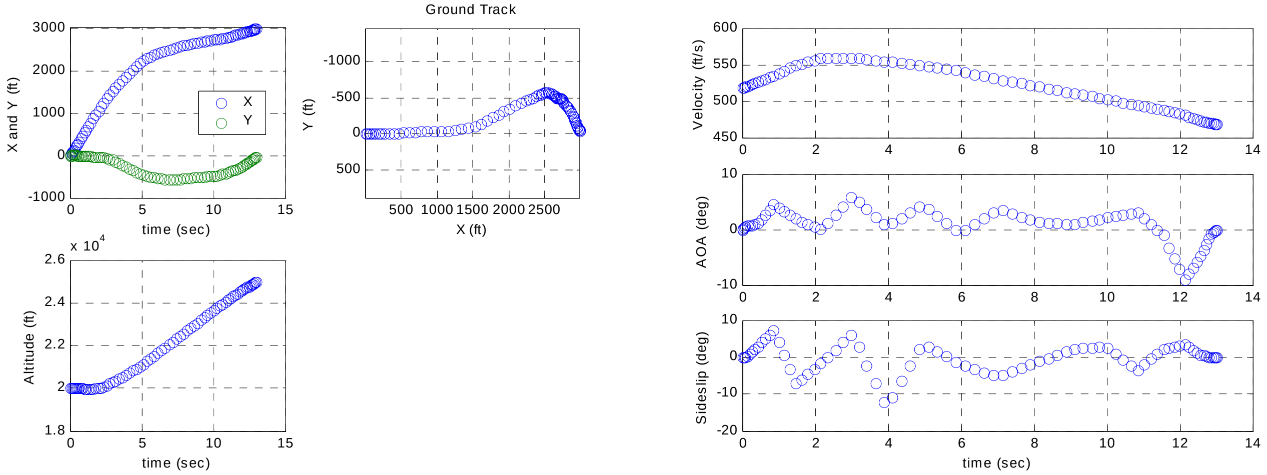
80 Nodes (bootstrapped from 20)

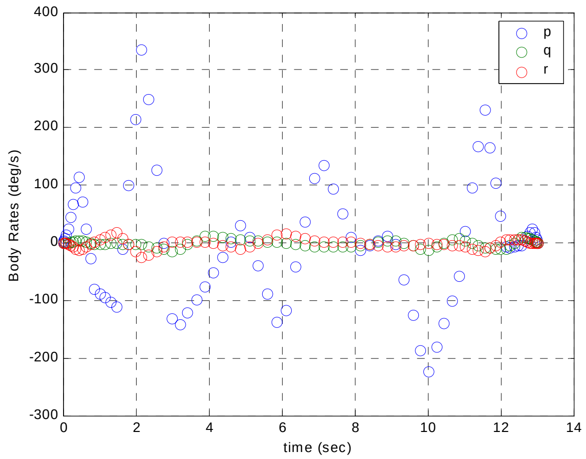
# of Iterations: 8687
Total Run Time: 101.97 min

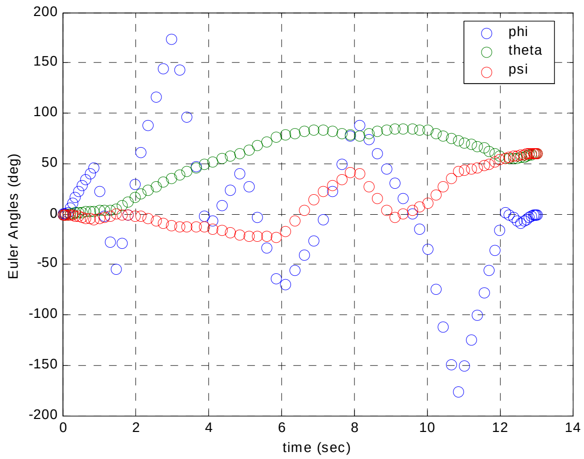
Maneuver Time: 13.00 sec

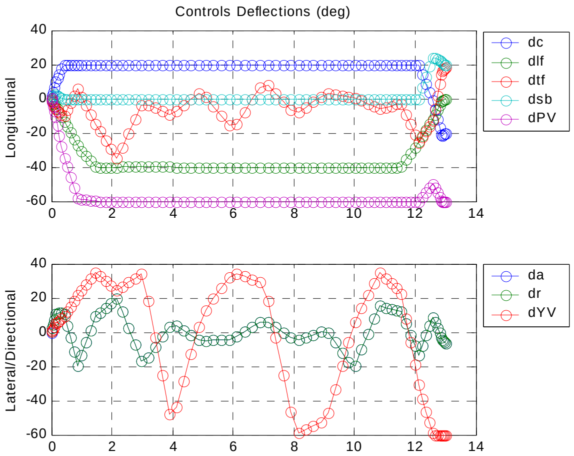
(Aircraft scale 3:1 in figure below.)

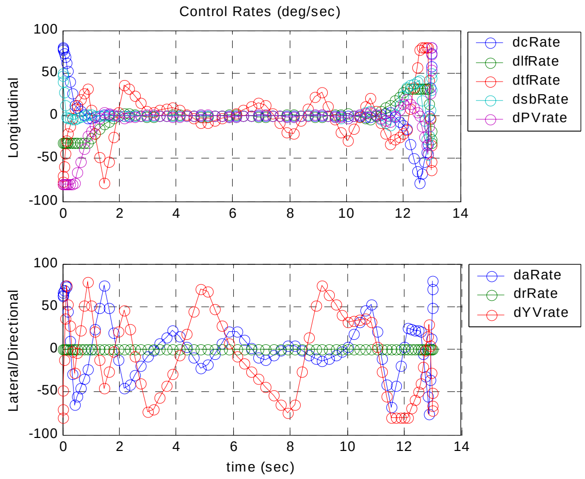


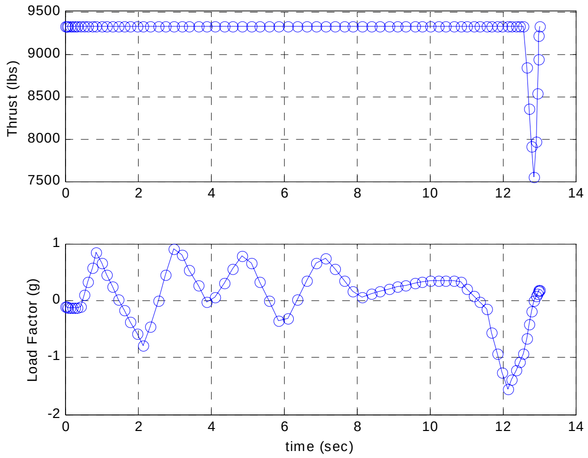


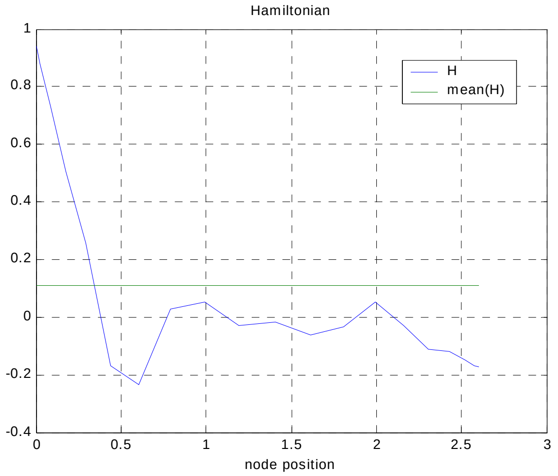

Body Rates (deg/s)
400
300
200
100
0
-100
-200
-300
0
2
4
6
8
10
12
14
time (sec)
p
q
r


Euler Angles (deg)
200
150
100
50
0
-50
-100
-150
-200
0
2
4
6
8
10
12
14
time (sec)
phi
theta
psi


Controls Deflections (deg)
Longitudinal
40
20
0
-20
-40
-60
0
2
4
6
8
10
12
14
dc
dlf
dtf
dsb
dPV
Lateral/Directional
40
20
0
-20
-40
-60
0
2
4
6
8
10
12
14
da
dr
dYV


Control Rates (deg/sec)
Longitudinal
100
50
0
-50
-100
0
2
4
6
8
10
12
14
dcRate
dlfRate
dtfRate
dsbRate
dPVrate
Lateral/Directional
100
50
0
-50
-100
0
2
4
6
8
10
12
14
time (sec)
daRate
drRate
dYVrate


Thrust (lbs)
9500
9000
8500
8000
7500
0
2
4
6
8
10
12
14
Load Factor (g)
1
0
-1
-2
0
2
4
6
8
10
12
14
time (sec)


Hamiltonian
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
0.5
1
1.5
2
2.5
3
node position
H
mean(H)

D. “VERTICAL COBRA”

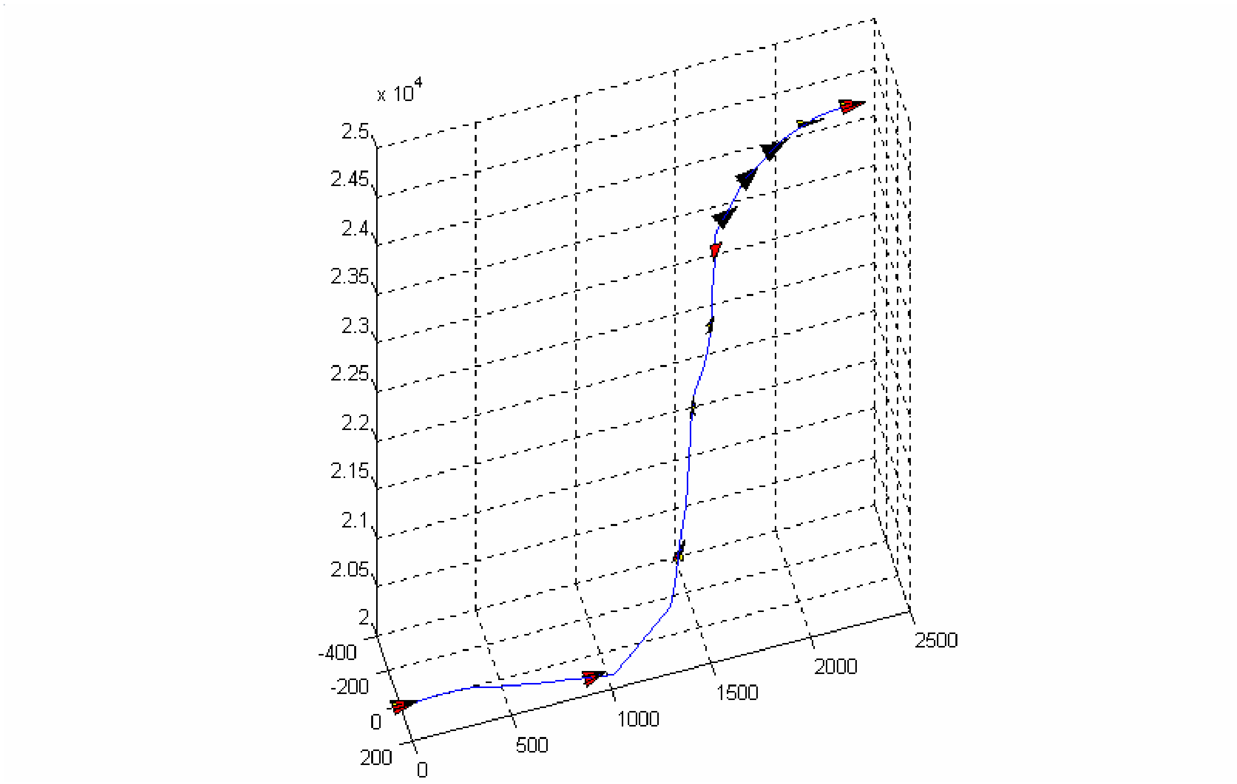
Delta: ΔV = -75% (-388 fps)

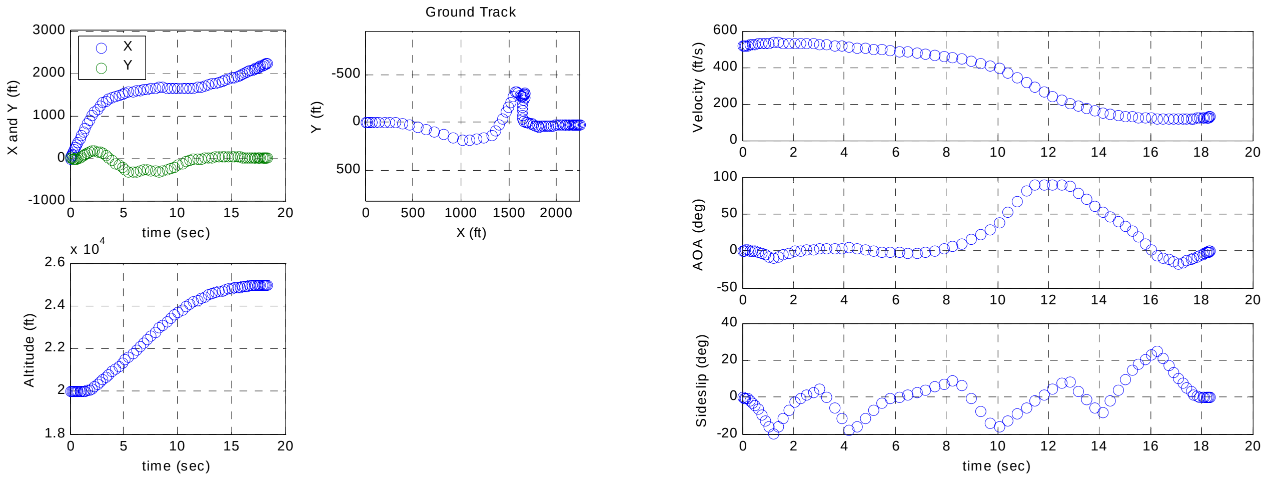
80 Nodes (bootstrapped from 20)

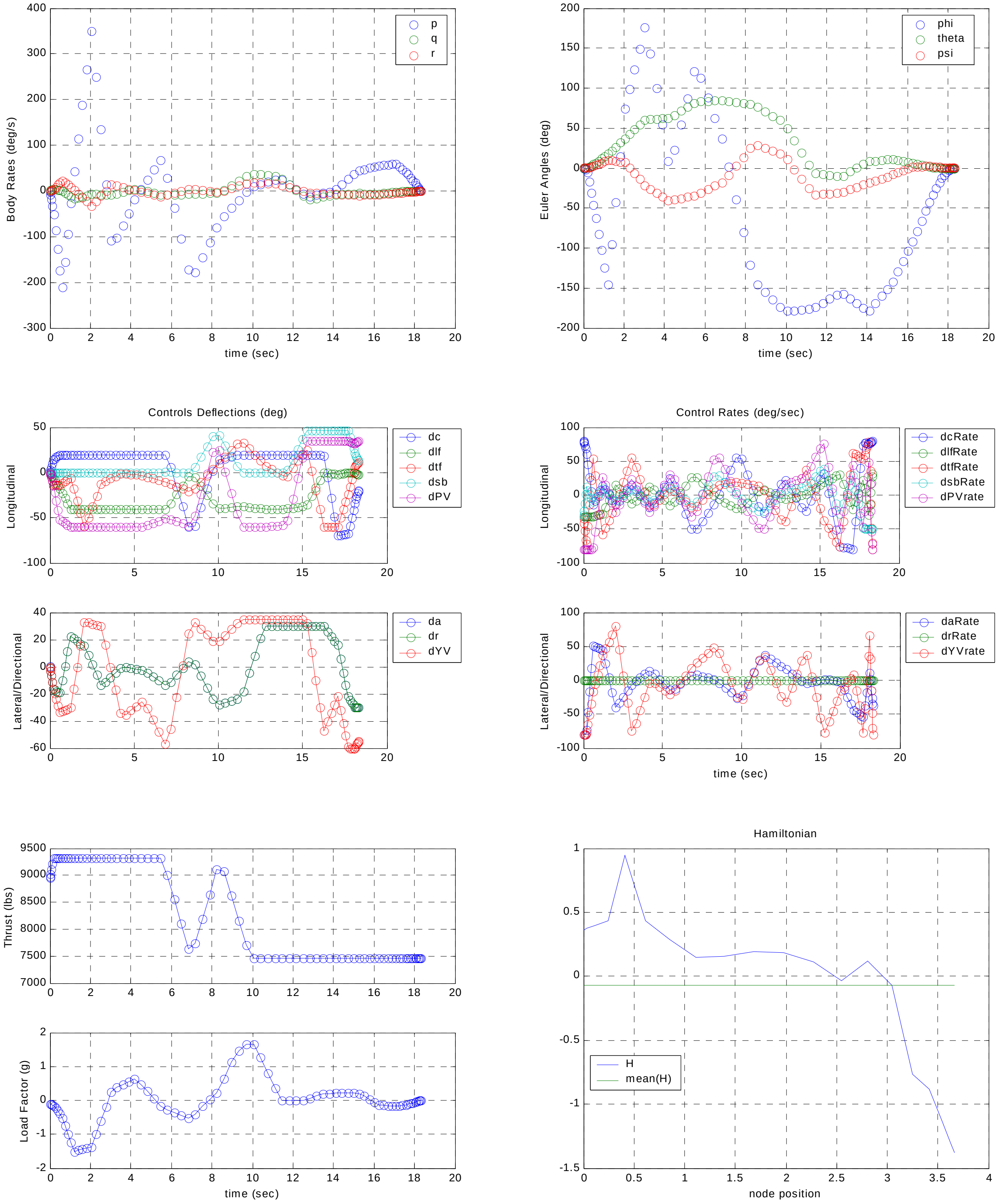
Unconstrained: X, Y (0 ± 5000 ft)
H (25k-35k ft)
δ (min-max deflection)

# of Iterations: 26662
Total Run Time: 274.27 min

Maneuver Time: 18.30 sec
(Aircraft scale 2:1 in figure below.)






Body Rates (deg/s)
400
300
200
100
0
-100
-200
-300
p
q
r
time (sec)
Euler Angles (deg)
200
150
100
50
-50
-150
-200
phi
theta
psi
time (sec)
Controls Deflections (deg)
Longitudinal
dc
dlf
dtf
dsb
dPV
Lateral/Directional
da
dr
dYV
Control Rates (deg/sec)
Longitudinal
dcRate
dlfRate
dtfRate
dsbRate
dPVrate
Lateral/Directional
daRate
drRate
dYVrate
time (sec)
Thrust (lbs)
9500
9000
8500
8000
7500
7000
Load Factor (g)
time (sec)
Hamiltonian
H
mean(H)
node position

LIST OF REFERENCES

1. Shaw, Robert L., *Fighter Combat: Tactics and Maneuvering*, pp. 1-7, 62-97, 99, 387-417. United States Naval Institute, 1985.

2. Herbst, W.B., "Dynamics of Air Combat," *Journal of Aircraft*, v. 20, pp. 594-598, July 1983.

3. Anderson, John D. Jr., *Fundamentals of Aerodynamics*, pp. 282. McGraw-Hill, 2001.

4. Dryden Flight Research Center. "F-18 HARV." [http://www.dfrc.nasa.gov/Research/HARV/] May 2003.

5. Chambers, J.R., William, P.G., and Luat, T.N. (editors), "High Angle of Attack Technology," *NASA CP-3149*, v. 1, May 1992.

6. NASA CP-3207, *High Angle of Attack Projects and Technology Conference*, compiled by N.W. Matheny, v. 1-4, April 1992.

7. NASA CP-10143, *Fourth NASA High Alpha Conference*, compiled by V. Regenie, v. 1-4, July 1994.

8. Herbst, W.B., "Future Fighter Technologies," *Journal of Aircraft*, v. 17, pp. 561-566, August 1980.

9. Zagainov, G.I., "High Maneuverability. Theory and Practice," paper presented at the AIAA Aircraft Design, Systems and Operations Meeting. Monterey, California. 11-13 August 1993. (Also AIAA paper 93-4737.)

10. Gal-Or, Benjamin, *Vectored Propulsion, Supermaneuverability and Robot Aircraft*. Springer-Verlag, 1990.

11. Boeing Phantom Works. "Unmanned Combat Aerial Vehicle (X-45)." [http://www.boeing.com/phantom/ucav.html] June 2003.

12. Howard, R.M., "Flight Mechanics I," course notes from the Navy Postgraduate School. Fall 2002.

13. Well, K.H., Faber, B., and Berger, E., "Optimization of Tactical Maneuvers Utilizing High Angles of Attack," *Journal of Guidance, Dynamics and Control*, v. 5, pp. 131-137, March 1982.

14. Ashley, H., "On the Feasibility of Low-Speed Aircraft Maneuvers Involving Extreme Angles of Attack," *Journal of Fluids and Structures*, v. 1, pp. 319-335, July 1987.

15. Etkin, B., *Dynamics of Flight, Stability and Control, 2nd Ed.*. Wiley, 1982.

16. Schmidt, Louis V., *Introduction to Aircraft Flight Dynamics*. AIAA, 1998.

17. AGARD AG-300, *AGARD Flight Test Techniques Series Volume 3 on Identification of Dynamic Systems – Applications to Aircraft Part 1: The Output Error Approach*, by R.E. Maine and K.W. Iliff, n.d..

18. USAF Museum at Wright-Patterson AFB. "North American L-17A Navion." [http://www.wpafb.af.mil/museum/annex/an29.htm] July 2003.

19. NASA TP-1997-206539, *Flight-Determined, Subsonic, Longitudinal Stability and Control Derivatives of the F-18 HARV with TV*, by K.W. Iliff and K.C. Wang, December 1997.

20. NASA TP-1999-206573, *Flight-Determined, Subsonic, Lateral-Directional Stability and Control Derivatives of the Thrust Vectoring F-18 HARV, and Comparisons to the Basic F-18 and Predicted Derivatives*, by K.W. Iliff and K.C. Wang, January 1999.

21. Napolitano, M.R. and Spagnuolo, J.M, "Determination of the Stability and Control Derivatives of the NASA F-18 HARV Using Flight Data," Contractor Report for NASA Grant # NCC 2-759. December 1993.

22. McDonnell Aircraft Co., McDonnell Douglas Corp. A7247, *F/A-18 Stability and Control Data Report, Revision B, Volume I*, by R.J. Pelikan and R.L. Swingle, November 1982.

23. McDonnell Aircraft Co., McDonnell Douglas Corp. A8575, *F/A-18 Basic Aerodynamic Data*, 1984.

24. NAVAIR. "Naval Unmanned Aerial Vehicles (PMA-263)." [http://uav.navair.navy.mil/home2.htm] June 2003.

25. Dornheim, M.A., "X-31 Flight Tests to Explore Combat Agility to 70 Deg. AOA," *Aviation Week & Space Technology*, pp. 38-41, 11 March 1991.

26. Rockwell Aerospace NA-87-1119L, *System Definition Manual, Revision E – Volume 1: Air Vehicle Summary*, September 1989.

27. Rockwell Aerospace TFD-88-1129L, *X-31 Longitudinal Aerodynamic Dataset for Flight Control Analysis, Revision A*, February 1988.

28. Rockwell Aerospace TFD-88-1268L, *X-31 Lateral/Directional Aerodynamic Dataset for Flight Control Analysis, Revision A, Volumes I & II*, February 1989.

29. Naval Postgraduate School Technical Report AA-02-002, *User's Manual for DIDO 2002: A MATLAB Application Package for Dynamic Optimization*, by I.M. Ross and F. Fahroo, June 2002.

30. Josselyn, Scott, "Optimal Aircraft Maneuvers," paper written for a research class at NPS. Fall 2002.

31. Komduur, H.J., and Visser, H.G., "Optimization of Vertical Plane Cobralike Pitch Reversal Maneuvers," *Journal of Guidance, Control and Dynamics*, v. 25, pp. 693-702, August 2002.

32. Lichtsinder, A., Kreindler, E., and Gal-Or, B., "Minimum-Time Maneuvers of Thrust-Vectored Aircraft," *Journal of Guidance, Control and Dynamics*, v. 21, pp. 244-250, April 1998.

33. Horie, K., and Conway, B.A., "Optimization for Fighter Aircraft Vertical-Plane Maneuvering Using Poststall Flight," *Journal of Aircraft*, v. 37, pp. 1017-1021, December 2000.

34. Murayama, K., and Hull, D.G., "The Cobra Maneuver as a Minimum Time Problem," paper presented at the AIAA Atmospheric Flight Mechanics Conference. Reston, Virginia. 1997. (Also AIAA paper 97-3586.)

35. Bocvarov, S., Lutze, F.H., and Cliff, E.M., "Time-Optimal Reorientation Maneuvers for a Combat Aircraft," *Journal of Guidance, Control and Dynamics*, v. 16, pp. 232-240, April 1993.

36. Stalford, H., and Hoffman, E., "Thrust Vectoring Effect on Time-Optimal 90 Degrees Angle of Attack Pitch Up Maneuvers of a High Alpha Fighter Aircraft," paper presented at the AIAA Guidance, Navigation, and Control Conference. Boston, Massachusetts. 14-16 August 1989. (Also AIAA paper 89-3521.)

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Al Bowers
   NASA Dryden Flight Research Center
   Edwards, California

4. Brent Cobleigh
   NASA Dryden Flight Research Center
   Edwards, California

5. Dave Sorenson
   Boeing Vector Product Team
   NAS Patuxent River, Maryland

6. Kenneth Rauch
   Navy Thrust Vectoring Advanced Development Project Office
   NAS Patuxent River, Maryland

7. Scott Josselyn
   Navy Test Pilot School
   NAS Patuxent River, Maryland