



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**DESIGN AND HARDWARE-IN-THE-LOOP
IMPLEMENTATION OF OPTIMAL CANONICAL
MANEUVERS FOR AN AUTONOMOUS PLANETARY
AERIAL VEHICLE**

by

Marta K. Savage

December 2012

Thesis Co-Advisors:

Mark Karpenko
I. Michael Ross

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
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 December 2012	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE DESIGN AND HARDWARE-IN-THE-LOOP IMPLEMENTATION OF OPTIMAL CANONICAL MANEUVERS FOR AN AUTONOMOUS PLANETARY AERIAL VEHICLE			5. FUNDING NUMBERS	
6. AUTHOR(S) Marta K. Savage				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5-000			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. IRB Protocol number ____N/A____.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) A truly autonomous aerial vehicle is required for conducting aerial missions at distances great enough to cause time lag in communications, such as on other planets. This level of autonomy also reduces the requirement for trained UAV pilots to fly round-the-clock missions. Development of optimal canonical maneuvers is a step towards achieving real-time optimal trajectory generation and more fully autonomous aircraft capable of independent and efficient flight maneuvering. This thesis develops a model of the MONARC aerial vehicle and sets up the optimal control problem for generating canonical maneuver profiles. The DIDO optimal control software is used in order to generate time-optimal trajectories for flight implementation on the MONARC test bed. The ability of the MONARC to fly the optimal trajectories is verified using a 6DOF SIMULINK model. Several canonical maneuvers were developed and optimized to generate trajectories for multiple flight scenarios. One of these cases is analyzed for implementation as part of a Hardware-in-the-Loop (HIL) simulation. This HIL test will verify that the optimization model has sufficient fidelity to be used to generate optimal trajectories that can be physically flown by the MONARC.				
14. SUBJECT TERMS UAV, autonomous, unmanned aerial vehicle, mentor, optimization, trajectory planning, minimum time problem, min time, optimal, DIDO, SLUGS, autopilot			15. NUMBER OF PAGES 135	
			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 UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**DESIGN AND HARDWARE-IN-THE-LOOP IMPLEMENTATION OF
OPTIMAL CANONICAL MANEUVERS FOR AN AUTONOMOUS
PLANETARY AERIAL VEHICLE**

Marta K. Savage
Lieutenant, United States Navy
B.S. United States Naval Academy, 2004
M.S. American Military University, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
December 2012**

Author: Marta K. Savage

Approved by: Mark Karpenko
Thesis Co-Advisor

I. Michael Ross
Thesis Co-Advisor

Knox T. Millsaps
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A truly autonomous aerial vehicle is required for conducting aerial missions at distances great enough to cause time lag in communications, such as on other planets. This level of autonomy also reduces the requirement for trained UAV pilots to fly round-the-clock missions. Development of optimal canonical maneuvers is a step towards achieving real-time optimal trajectory generation and more fully autonomous aircraft capable of independent and efficient flight maneuvering.

This thesis develops a model of the MONARC aerial vehicle and sets up the optimal control problem for generating canonical maneuver profiles. The DIDO optimal control software is used in order to generate time-optimal trajectories for flight implementation on the MONARC test bed. The ability of the MONARC to fly the optimal trajectories is verified using a 6DOF SIMULINK model. Several canonical maneuvers were developed and optimized to generate trajectories for multiple flight scenarios. One of these cases is analyzed for implementation as part of a Hardware-in-the-Loop (HIL) simulation. This HIL test will verify that the optimization model has sufficient fidelity to be used to generate optimal trajectories that can be physically flown by the MONARC.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND	1
	1. Unmanned Vehicles	1
	2. Trajectory Optimization Methods.....	3
B.	MOTIVATION	3
C.	UAV APPLICATIONS.....	5
	1. Military Applications.....	5
	2. Science, Civilian, and Government Applications.....	6
	3. Space Applications	6
	4. Improvements for All Applications	7
D.	RESEARCH OBJECTIVES	7
	1. Feasibility of Execution with Selected Hardware	7
	2. Model Fidelity Level	8
E.	ORGANIZATION, METHODOLOGY AND SCOPE	8
II.	THE MONARC AIRCRAFT.....	11
A.	SELECTION OF TEST AIRCRAFT	11
	1. Selection Criteria	11
	2. Physical and Aerodynamic Characteristics.....	11
	3. Assembly	13
B.	MODELING THE MONARC AIRCRAFT	14
	1. The 6DOF Dynamics Model.....	14
	a. <i>Reference Frame.....</i>	<i>14</i>
	b. <i>Equations of Motion</i>	<i>15</i>
	2. SIMULINK Flight Control Toolbox Model	18
	a. <i>SIMULINK Model</i>	<i>20</i>
	b. <i>Aerodynamic Coefficients.....</i>	<i>21</i>
	c. <i>Control limits.....</i>	<i>22</i>
III.	THE TRAJECTORY OPTIMIZATION PROBLEM FOR A FIXED-WING AIRCRAFT	23
A.	DESCRIPTION OF PROBLEM.....	23
B.	THE OPTIMAL CONTROL PROBLEM.....	23
C.	CREATING A DYNAMIC MODEL OF THE MONARC AIRCRAFT..	25
	1. The Simple 3DOF Point-Mass Model	25
	a. <i>The Minimum Time Problem Formulation</i>	<i>26</i>
	b. <i>Reference Frame.....</i>	<i>26</i>
	c. <i>Disadvantages of the Simplified Point-Mass 3DOF Model ..</i>	<i>27</i>
	2. The 3DOF Dynamic Model	27
	a. <i>Model Assumptions.....</i>	<i>28</i>
	b. <i>3DOF Dynamics.....</i>	<i>28</i>
	c. <i>Aerodynamics and Atmospheric Conditions.....</i>	<i>29</i>
	d. <i>Final 3DOF Dynamic Model.....</i>	<i>30</i>

e.	<i>Minimum Time Problem Formulation.....</i>	<i>31</i>
f.	<i>Analysis of the OCP.....</i>	<i>32</i>
g.	<i>Reference Frame.....</i>	<i>35</i>
D.	INTRODUCTION TO OPTIMAL CONTROL WITH MATLAB AND DIDO	36
1.	MATLAB Optimization Code	36
2.	State Variable Constraints.....	36
3.	Initial and Final Conditions	37
4.	Control Rate Limits	37
5.	Events Function and Limits	37
6.	Time and Node Number Selection.....	38
7.	Maneuver Results, Validation and Verification.....	38
8.	Improvements.....	39
a.	<i>Number of Nodes</i>	<i>39</i>
b.	<i>Scaling.....</i>	<i>39</i>
E.	TESTING THE OPTIMAL CONTROL CODE	40
1.	Code Verification	40
2.	Trim Maneuvers and Additional Model Verification.....	40
IV.	CANONICAL MANEUVERS	41
A.	FORMULATION OF FLIGHT TEST MANEUVERS.....	41
1.	Conditions Common to Each of the Canonical Maneuvers	41
2.	Diagonal Transfer Flight.....	42
a.	<i>Description of Maneuver</i>	<i>42</i>
b.	<i>Initial and Final Conditions.....</i>	<i>43</i>
c.	<i>Box Constraints.....</i>	<i>44</i>
d.	<i>Results.....</i>	<i>45</i>
3.	Wide U-Turn	53
a.	<i>Description of Maneuver</i>	<i>53</i>
b.	<i>Initial and Final Conditions.....</i>	<i>53</i>
c.	<i>Box constraints.....</i>	<i>54</i>
d.	<i>Results.....</i>	<i>54</i>
4.	“Scoot Over”.....	60
a.	<i>Description of Maneuver</i>	<i>60</i>
b.	<i>Initial and Final Conditions.....</i>	<i>61</i>
c.	<i>Results.....</i>	<i>61</i>
5.	Narrow-Space Reversal	66
a.	<i>Description of Maneuver</i>	<i>66</i>
b.	<i>Initial and Final Conditions.....</i>	<i>66</i>
c.	<i>Box constraints.....</i>	<i>67</i>
d.	<i>Results.....</i>	<i>68</i>
B.	DISCUSSION OF THE CANONICAL MANEUVERS.....	72
V.	SIMULATION ENVIRONMENT FOR TRAJECTORY VERIFICATION	73
A.	AUTOPILOT SYSTEM	73
1.	Development and Description.....	74
2.	Components.....	76

3.	6DOF SIMULINK Model.....	77
4.	Integration with MONARC	77
5.	Navigation.....	78
6.	Ground Station Operation	79
B.	HIL SIMULATION.....	79
1.	Background	79
2.	Apparatus	80
C.	TRAJECTORY IMPLEMENTATION.....	81
VI.	MANEUVER IMPLEMENTATION.....	85
A.	EQUILATERAL TRIANGLE MANEUVER	85
1.	Minimum Time to Turn and Intersect Next leg	85
a.	<i>Description of Maneuver</i>	85
b.	<i>Updated Control Limits</i>	86
c.	<i>Initial and Final Conditions for Turn at Top of Triangle</i>	87
d.	<i>Box Constraints</i>	87
e.	<i>Event Bounds</i>	88
f.	<i>Trajectory Comparison with Typical Controller</i>	88
2.	Minimum Time to Turn and Arrive at Next Waypoint	89
a.	<i>Description of Maneuver</i>	89
b.	<i>Initial and Final Conditions for Turn to Next Waypoint</i>	90
c.	<i>Resulting Trajectories</i>	91
3.	Optimal Trajectory Implementation.....	91
a.	<i>Velocity, Altitude and Heading Inputs</i>	91
b.	<i>Roll, Pitch and Velocity Optimal Trajectory Inputs</i>	96
VII.	CONCLUSION AND RECOMMENDATIONS.....	103
A.	CONCLUSIONS	103
B.	FUTURE IMPROVEMENTS.....	104
1.	Iterative Upgrades to the Optimization Model	104
a.	<i>Control Limits</i>	104
b.	<i>Physical Characteristics</i>	104
2.	Optimization of Other Costs	104
C.	FLIGHT IMPLEMENTATION AND TESTING	105
	LIST OF REFERENCES	107
	INITIAL DISTRIBUTION LIST	111

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1.	Map of Countries That Acquired UAVs by Dec. 2011 (From [6])	2
Figure 2.	A Ship-Launched Pioneer UAV and a Predator UAV Armed with a Hellfire Missile (From [1], [14]).....	5
Figure 3.	MONARC Aircraft with Views of Modified Fuselage.....	12
Figure 4.	Body-Axis Coordinate System	14
Figure 5.	6DOF Longitudinal and Directional Flight Angles (After [28])	15
Figure 6.	Procedures for Aircraft Modeling (From [27]).....	19
Figure 7.	MoI Test Apparatus	20
Figure 8.	SIMULINK Model Control Law Structure (From [27]).....	21
Figure 9.	MONARC SIMULINK Model—Highest Level View (From [27]).....	21
Figure 10.	3DOF Simplified Point-mass Model Flight Angles	27
Figure 11.	Relationship between Aerodynamic Coefficients when $\beta=0$ (After [28])	29
Figure 12.	Reference Frame for 3DOF Maneuvers.....	35
Figure 13.	Successful Feasibility Test via Propagation	39
Figure 14.	Overhead view of Diagonal Transfer Maneuver	42
Figure 15.	Diagonal Maneuver 3-D View.....	43
Figure 16.	Overhead View of DIDO Diagonal Transfer Trajectory	46
Figure 17.	Diagonal Trajectory 3-D View	46
Figure 18.	Velocity During the Diagonal Transfer Maneuver	47
Figure 19.	Altitude Profile for Diagonal Maneuver	48
Figure 20.	Thrust Profile for Diagonal Maneuver.....	48
Figure 21.	AoA vs. Time for Diagonal Maneuver	49
Figure 22.	Bank Angle vs. Time for Diagonal Maneuver.....	50
Figure 23.	Flight Path and Heading Angles for Diagonal Maneuver.....	50
Figure 24.	Position Costates vs. Time for Diagonal Maneuver	51
Figure 25.	Control and Costate for Thrust Variable.....	52
Figure 26.	Hamiltonian vs. Time for Diagonal Maneuver	52
Figure 27.	Top View of U-turn Maneuver	53
Figure 28.	Overhead View of Optimized U-turn Trajectory vs. a Semi-Circular Trajectory.....	55
Figure 29.	Wide U-turn Trajectory 3-D View.....	56
Figure 30.	Velocity During the U-turn Maneuver.....	56
Figure 31.	Altitude Profile for U-turn Maneuver	57
Figure 32.	Thrust Profile for U-turn Maneuver.....	57
Figure 33.	AoA for U-turn Maneuver	58
Figure 34.	Bank Angle for U-Turn Maneuver	59
Figure 35.	Flight Path and Heading Angles for U-turn Maneuver.....	59
Figure 36.	Top View of “Scoot Over”	60
Figure 37.	Overhead View of “Scoot Over” Trajectory.....	62
Figure 38.	“Scoot Over” Trajectory 3-D View	62
Figure 39.	Velocity Profile of “Scoot Over” Maneuver.....	63
Figure 40.	Altitude Profile of “Scoot Over” Maneuver	63

Figure 41.	Thrust Profile of “Scoot Over” Maneuver	64
Figure 42.	AoA for “Scoot Over” Maneuver	64
Figure 43.	Bank Angle for “Scoot Over” Maneuver	65
Figure 44.	Flight Path and Heading Angles for “Scoot Over” Maneuver	65
Figure 45.	Top View of Reversal Maneuver	66
Figure 46.	Overhead View of Reversal Trajectory	68
Figure 47.	Reversal Trajectory 3-D View	69
Figure 48.	Velocity Profile of Reversal Trajectory	69
Figure 49.	Altitude vs. Time for Reversal Trajectory	70
Figure 50.	Thrust Profile of Reversal Trajectory	70
Figure 51.	AoA and Bank Angles for Reversal Trajectory	71
Figure 52.	Flight Path and Heading Angles for Reversal Trajectory	72
Figure 53.	Autopilot Board With Size Reference	74
Figure 54.	Control System Rapid Prototyping Process (From [40])	75
Figure 55.	Autopilot Hardware Architecture (From [39])	77
Figure 56.	SIMULINK Diagram of 6DOF Simulator (From [27])	78
Figure 57.	HIL Simulation Apparatus (After [39])	81
Figure 58.	SIMULINK Model of Mentor—Top Level View	82
Figure 59.	SIMULINK Model—Inside Control Laws Block Diagram	82
Figure 60.	SIMULINK Verification Model Using Altitude, Velocity, and Heading Trajectory Inputs (From [29])	84
Figure 61.	SIMULINK Verification Model Using Pitch, Roll, and Velocity Trajectory Inputs (From [29])	84
Figure 62.	Equilateral Triangle Maneuver—Complete Turn and Steady on Next Leg	86
Figure 63.	Load Factors with Varying μ_a Limits for Equilateral Triangle Maneuver	86
Figure 64.	Trajectories for Min Time to Turn and Intercept Next Leg	89
Figure 65.	Equilateral Triangle Maneuver—Complete Turn and Arrive at Next Waypoint	90
Figure 66.	Trajectories for Minimum Time To Arrive at Next Waypoint	91
Figure 67.	Top view of 120-degree Turn to Next Leg with v , z , σ Inputs to SIMULINK Model (From [45])	93
Figure 68.	Altitude profile of 120-degree Turn to Next Leg with v , z , σ Inputs to SIMULINK Model (From [45])	93
Figure 69.	Velocity profile for 120-degree Turn to Next Leg using v , z , σ Simulation Inputs (From [45])	94
Figure 70.	Pitch angle vs. Time for 120-degree Turn to Next Leg using v , z , σ as Simulation Inputs (From [45])	95
Figure 71.	Bank Angle vs. Time for 120-degree Turn to Next Leg (From [45])	96
Figure 72.	Top view of 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])	97
Figure 73.	Altitude profile of 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])	98
Figure 74.	Pitch Angle vs. Time for with v , ϕ , θ Inputs to SIMULINK Model (From [45])	99

Figure 75.	Velocity profile for 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45]).....	99
Figure 76.	BA vs. time for 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45]).....	100
Figure 77.	Rudder Angle used by controller for 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])	101
Figure 78.	Camp Roberts Airspace (From [40])	105
Figure 79.	McMillan Airfield at Camp Roberts (From [40]	106

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Estimated MONARC Aircraft Physical and Performance Parameters	13
Table 2.	6DOF Forces, Moments, and Velocity Components (After [28])	15
Table 3.	6DOF Aerodynamic Forces and Moments	16
Table 4.	6DOF Aircraft States (After [28, 29]).....	16
Table 5.	Preliminary MONARC Aerodynamic Coefficients (From [32]).....	22
Table 6.	Controls for Point-mass 3DOF Model.....	25
Table 7.	States for Point-mass 3DOF Model	26
Table 8.	3DOF Model Assumptions	28
Table 9.	Reduced-Order 3DOF Model States	31
Table 10.	Reduced-order 3DOF Controls	31
Table 11.	Representative Limits for States and Controls.....	37
Table 12.	Initial and Final Conditions for Diagonal Maneuver	43
Table 13.	State Bounds for Diagonal Maneuver	44
Table 14.	Event Bounds for Diagonal Maneuver	45
Table 15.	Control Bounds for All Canonical Maneuvers	45
Table 16.	Initial and Final Conditions for U-turn Maneuver.....	54
Table 17.	State Bounds for U-turn Maneuver	54
Table 18.	Initial and Final Conditions for “Scoot Over” Maneuver	61
Table 19.	Initial and Final Conditions for Reversal Maneuver.....	67
Table 20.	State Bounds for Reversal Maneuver	67
Table 21.	Autopilot Sensors (From [39], [41])	76
Table 22.	Initial and Final Conditions 120-degree Turn to Next Leg.....	87
Table 23.	State Bounds for 120-degree Turn to Next Leg.....	87
Table 24.	Control Bounds for 120-degree Turn to Next Leg.....	88
Table 25.	Initial and Final Conditions 120-degree Turn to Next Waypoint	90

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

3DOF	Three Degree-of-Freedom
6DOF	Six Degree-of-Freedom
AoA	Angle of Attack
AAV	Autonomous Aerial Vehicle
APAV	Autonomous Planetary Aerial Vehicle
BA	Bank Angle
BVP	Boundary Value Problem
CAS	Control Augmentation System
CG	Center of Gravity
CM	Center of Mass
COTS	Commercial Off-the-Shelf
DCM	Direction Cosine Matrix
DOF	Degrees of Freedom
DSC	Digital Signal Controller
DR	Downrange
DU	Distance Unit
EoM	Equations of Motion
FCS	Flight Control System
FDC	Flight Dynamics and Control
FPA	Flight Path Angle
GCS	Ground Control Station
GNC	Guidance, Navigation & Control
GPS	Global Positioning System
HA	Heading Angle
HIL	Hardware-in-the-Loop
HMC	Hamiltonian Minimization Constant
IC	Initial Condition
JPL	Jet Propulsion Laboratory
L/D	Lift-to-Drag Ratio
M	Mach number
MAV	Micro Aerial Vehicle

MoI	Moment of Inertia
MONARC	Mentor Optimal-Trajectory Naval Autonomous Research Craft
MU	Mass Unit
NASA	National Aeronautics and Space Administration
NAVAIR	Naval Air Systems Command
NAVPGSCOLINST	Naval Postgraduate School Instruction
NPS	Naval Postgraduate School
NED	North, East, Down
NL OCP	Nonlinear Optimal Control Problem
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
OPNAVINST	Office of the Chief of Naval Operations Instruction
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PS	Pseudospectral
QGC	QGroundControl
R&D	Research and Development
RC	Remote Controlled
RPV	Remotely Piloted Vehicle
RTB	Return to Base
SAL	Straight-and-Level
SAS	Stability Augmentation System
SLUGS	Santa Cruz Low-cost UAV GNC System
TC	Transversality Condition
TM	Transformation Matrix
TU	Time Unit
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UCAV	Unmanned Combat Aerial Vehicle
UCSC	University of California Santa Cruz
UGV	Unmanned Ground Vehicle
UUV	Unmanned Underwater Vehicle
UV	Unmanned Vehicle

LIST OF SYMBOLS

α	Angle of Attack (rad)
β	Side Slip Angle (rad)
λ	Dual Vector or Costate variable
μ	Bank Angle (rad)
γ	Pitch Angle (rad)
σ	Heading Angle (rad)
ρ	Air density (kg/m ³)
ϕ	Pitch (Euler angle, rad)
θ	Roll (Euler angle, rad)
ψ	Yaw (Euler angle, rad)
C_A	Axial Force Coefficient along body X-axis
C_N	Normal Force Coefficient along body Z-axis
C_D	Drag Coefficient, $C_L = \frac{L}{\frac{1}{2}\rho V^2 S}$
C_L	Lift Coefficient, $C_D = \frac{D}{\frac{1}{2}\rho V^2 S}$
E	Endpoint or event cost function
e	Endpoint or event constraint function
g	Acceleration of Gravity (m/s ²)
H	Control Hamiltonian
I	Moment of Inertia (kg m ²)
J	Scalar Cost Functional
L	Rolling Moment (kg m ²)
M	Pitching Moment (kg m ²)
N	Yawing Moment (kg m ²)
n	Load Factor (g's)
p	Pitch rate (rad/s)
q	Roll rate (rad/s)
q	Dynamic pressure (Pa)

r	Yaw rate (rad/s)
S	Wing Area (m ²)
t	Time (s)
T	Thrust (N)
u	Control variable
v	Velocity (m/s)
X	Axial Force (N)
Y	Side Force (N)
Z	Normal Force (N)
x	Downrange distance (m)
y	Cross-range distance (m)
z	Altitude (m)
u_T	Thrust Control (N/s)
u_α	AoA Control (rad/s)
u_μ	Bank Angle Control (rad/s)

ACKNOWLEDGMENTS

Thank you very much to Dr. Karpenko for absolutely endless patience and willingness to help me work through and around the pile of problems and hang-ups that often seemed insurmountable. Your ability to explain things from other angles whenever needed has been invaluable.

A special thanks to Dr. Ross for offering me a fascinating problem to tackle and some unique tools to attack it with, for putting effort and passion into your teaching, and for always standing up for your students.

To Dr. Lee, thanks for such thorough work and patience with someone who was utterly new to aircraft dynamics, simulation and modeling. Thanks also for your development of the SIMULINK model of the Mentor aircraft, and your interesting ongoing work on algorithms for UAV group coordination and formation flight.

Thank you to Brad Watanabe for talking me through the basics of communications protocols, data storage and processing, and all the other things pertaining to computers that I needed to learn the basics for.

Thank you to Robert Casey, for explaining the “trons” and other computer- and electronics-related things that I needed to know about, how SLUGS and all the peripherals work, for putting up with all of my (often repetitive) questions with good grace and humor, and for always being a good sounding board.

Thank you Travis Bateman for always being generous about helping and sharing what you had learned while overcoming similar problems, for letting me rant, and for letting me spend some time with your very cool family.

Thank you to my parents for instilling in me the importance of the lifelong pursuit of learning and appreciation of the values of higher education, and to keep slogging through, no matter how frustrating school can be. Thank you for your support and your frequent help—despite the long drive—with our moves, juggling of school schedules, childcare, and more.

Thank you to my daughter, for your beyond-your-years understanding of my need to get work done, and for always making me see what is truly important.

Most of all, thanks to my husband. You have been selflessly working your own school and writing schedule around mine, supporting me throughout career paths that have been anything but traditional for either of us, and listening to my gripes—always while being flexible and patient. And to top it all off, you have been doing a wonderful job raising our daughter while all this insanity has been going on.

I. INTRODUCTION

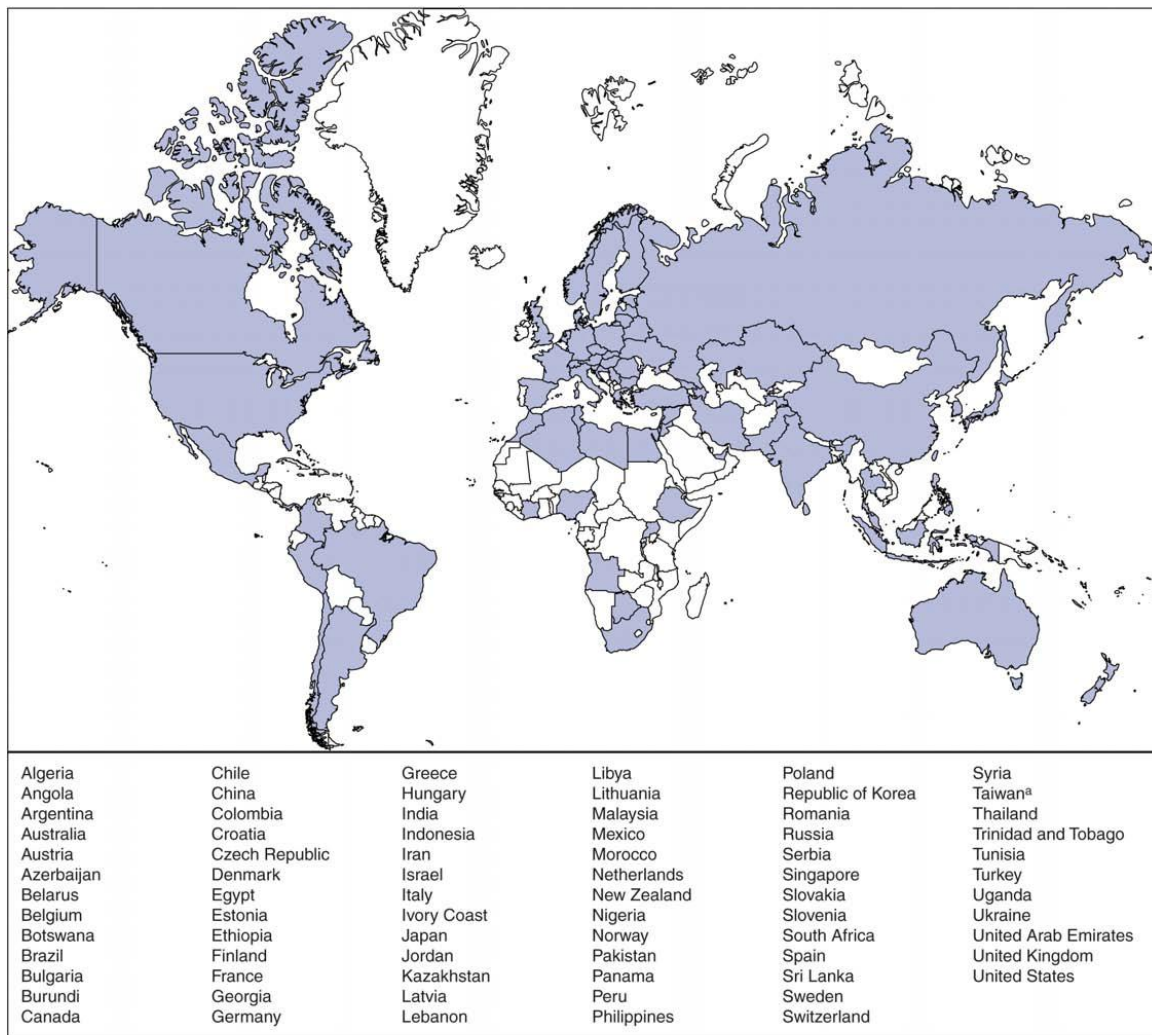
A. BACKGROUND

Unmanned vehicles (UVs), including aircraft, underwater vehicles, ground vehicles, robots and other more exotic autonomous systems, are arguably the cutting edge of technology in multiple fields. Applications for UVs can range from a broad spectrum of military uses to emergency services, law enforcement, weather prediction, surveying, and exploration [1], [2], [3]. The United States had 91 different types of unmanned aerial vehicles (UAVs) in service or development in just the military and commercial sectors in 2003 [4], and that number has continued to grow rapidly.

1. Unmanned Vehicles

In addition to the increase in number of military and government UAVs, there are a number of military unmanned ground vehicles, unmanned underwater vehicles, and vehicles developed at educational and research institutions. The number of unmanned systems in just the United States alone is well into the hundreds, and globally it is into the thousands. UAVs are now prevalent around the world, as shown by the shaded countries in Figure 1. UAVs are making such an impact on how wars are fought today that they have even been categorized by some as a “Revolution in Military Affairs” [5], a distinction reserved for the advent of innovative technology that has sudden and wide-reaching strategic impact, such as artillery and nuclear weapons.

The difference between a Remotely Piloted Vehicle (RPV) and an Autonomous Aerial Vehicle (AAV) is a distinction that is often misunderstood or misrepresented. Unmanned does not necessarily mean unpiloted, but can mean either autonomous or remotely piloted [1], [2]. The unmanned category just means that a human is not onboard, and the category encompasses both AAVs, which are not yet completely autonomous, and RPVs, which are increasingly common and have been in use by the U.S. military since the Vietnam-era Firebee RPV [1], [2], [4].



Sources: GAO analysis of various unclassified sources; Map Resources (map).

Figure 1. Map of Countries That Acquired UAVs by Dec. 2011 (From [6])

The majority of modern UAVs, while being touted as autonomous systems, are still flown by a human pilot, using a handheld remote control or a computer interface that achieves the same end—a person must be piloting at all times. This increases requirement for manning and makes the UAV subject to added accident risk from pilot fatigue. It is also costly to train a pilot and to pay for the additional support manpower needed to keep military UAVs flying [1], [3]. To achieve a true AAV that does not require a pilot presents a new set of challenges, but solves many of the problems with current human-flown UAVs and allows for new and even undiscovered uses.

2. Trajectory Optimization Methods

Finding the most efficient way to get from point A to point B autonomously given a certain set of conditions and boundaries is a problem that has been studied in almost every conceivable field. There are many ways to optimize a trajectory, depending on the goal. This thesis uses the pseudospectral optimal control theory [7], which has been used for over a decade and is now well-established as a method for autonomous motion planning [8], [9], [10], [11], [12].

The cost to be minimized could be time, fuel, or distance; or it could be desirable to maximize an objective such as time-on-station or area coverage. This work addresses only time-optimization problems, but the cost function may be changed at any time to look at other objectives, depending on the goals and mission of the aircraft.

Creating a database of optimal canonical maneuvers for utilization by an autopilot system provides a first step towards enabling operators to give commands to an aircraft, but not be required to actually pilot it—the aircraft can combine the optimal maneuvers in succession to achieve the desired outcome in an autonomous fashion. This can be successfully achieved whether the goal is to fly from one point to another as quickly as possible, or to provide persistent surveillance.

B. MOTIVATION

That a UAV be fully autonomous is crucial for fulfilling any mission that must occur at great distances, such as on a remote planet, where time lags between sending a command and receiving feedback are prohibitively long for a human pilot to control the vehicle. One-way communication times vary depending on the distance between Earth and another planet. For communication to Mars, the one-way lag is in the neighborhood of 14 minutes or longer [13]. Waiting a half hour between manually-commanded maneuvers may work well enough for a ground rover, which can simply stop between commands, but is impossible for an aircraft in flight. In the case of planets, moons or other bodies with atmospheric conditions conducive to powered aerial flight, a vehicle capable of fully autonomous flight would be invaluable for surveying, observation, and other scientific research and exploration purposes.

A ground rover on a planet, moon, or other body such as an asteroid must travel slowly and cannot scale cliffs, cross trenches, ascend mountains, or observe large amounts of terrain at once. A Martian AAV could help fill the large, vacant niche left between ground rovers and orbiting satellites. Such a craft is capable of providing a closer look at the surface than a satellite, and could cover much more area in a short amount of time. An AAV may also go places where a surface rover cannot.

Arguably the most famous unmanned vehicle today is the new Mars Rover (Curiosity), which landed on Mars on 06 August of 2012 [13] and garnered a great deal of public and press interest. Developed by NASA and JPL, Curiosity is to date successfully conducting its exploration mission inside Gale Crater, where it is using a suite of ten scientific instruments to conduct research and return data to Earth [13]. The vehicle has a mission-life driving distance of only 12.4 miles [13], though, a very limited area. Curiosity is semi-autonomous, in that it is mostly driven via direct commands from operators on Earth, but can also operate and navigate on its own. Autonomous operations can be based solely on wheel rotation count, or use a hazard avoidance mode and onboard camera to execute simple maneuvers [13].

An AAV working in conjunction with a ground rover and orbiting satellites could greatly increase knowledge of Mars. While Curiosity is able to make a detailed study of a small area, a Mars AAV could cover more area, and also work cooperatively with the rover. The AAV could send information about safe paths for the rover to take, or find the nearby areas of greatest interest to send the rover to explore. Aerial imagery and sampling combined with that of the rover could greatly expand the scientific data available. Similarly, working jointly with a Mars-orbiting satellite would allow a Mars AAV to investigate areas that appear interesting from orbit, taking a closer look with the AAV. A very advanced AAV on Mars could even be equipped with a small on-board laboratory and sampling equipment, so it could land and take samples and return data to Earth via satellite uplink.

C. UAV APPLICATIONS

The applications possible for an UAV are limited only by imagination. Virtually any task that traditionally uses a camera, a scientific instrument, a vehicle of any kind, a communications or recording device, or even a weapon could feasibly be performed by the right kind of UAV either created for the task or modified to perform the task.

1. Military Applications

The U.S. military increasingly uses UAVs for a growing list of missions, especially those that carry a high risk for personnel. Production of unmanned aircraft has outstripped production of manned aircraft, and unmanned ground and underwater vehicles are also increasing in type and number.

Military UAVs range in size from very small micro-UAVs that can be launched even from the smallest of deployed surface ships, to relatively large and heavy multi-mission UAVs that can carry strike weapons, such as the Predator, which was initially developed for surveillance and reconnaissance but has also been armed since 2001 [14]. Drones are most frequently used as surveillance craft, but the mission applications are broadening as UAVs become more sophisticated and more accepted by the military and government leadership.

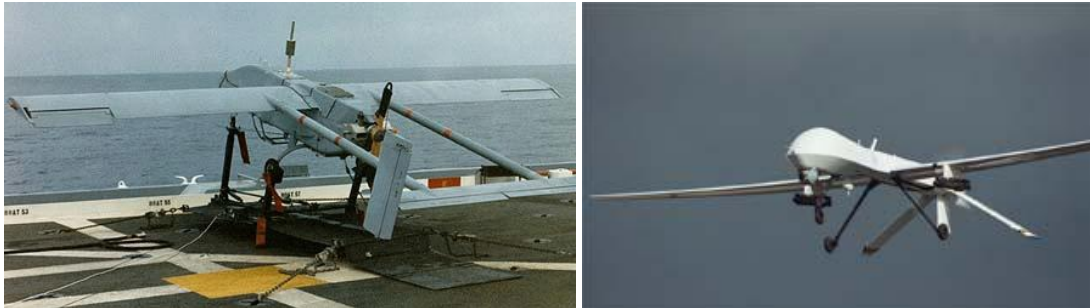


Figure 2. A Ship-Launched Pioneer UAV and a Predator UAV Armed with a Hellfire Missile (From [1], [14])

UAVs are being developed for use as refueling platforms, communications relays, search-and-rescue craft, and targeting platforms. An impressive variety of instruments and equipment may be outfitted, ranging from cameras to RADAR to lasers. A joint, multi-mission AAV platform would be highly desirable in today's budget environment [15]. Having an adaptable, fully autonomous UAV platform would not only allow for the same platform to be used for various missions, but also could allow for one aircraft to be used for all services. This would save money on the development side, as well as cut costs for operations and maintenance.

2. Science, Civilian, and Government Applications

UAVs have entered the mainstream worldwide, especially for civilian commercial uses and scientific research. Applications include taking atmospheric measurements, forest mapping, weather prediction, search and rescue, and traffic control, to name a few. A UAV has even been used by a team from Université de Liège for aerial inventory of elephant populations in Burkina Faso [16]. The market for UAVs for non-military purposes is growing quickly.

3. Space Applications

Demonstration of reliable autonomous flight capability on a robust earth-based platform is essential prior to deployment of the technology in high-risk areas of operation, such as on Mars. Repairs are either extremely difficult or impossible in space, and usually prohibitively expensive, so autonomous craft to be used in space must be extensively tested and demonstrated in a local environment.

The research in this work can be applied to autonomous craft that could eventually operate on the surface of planets, moons, and asteroids, as well as in Earth orbit or deep space. The benefits are not limited to atmospheric aircraft, as a new model can be created or modified from the current UAV model for any environment and vehicle. The model and autopilot are both adaptable, while the trajectory generation process remains the same no matter the environment or mission of the vehicle.

4. Improvements for All Applications

A readily available and easily adaptable AAV platform that can be given commands and independently complete flight maneuvers, such as take-off, landing, trimmed straight-and-level flight, coordinated turns, smooth ascents and descents, and search patterns, and other more exotic maneuvers would remove the need for a pilot with specialized and lengthy training. Whether the objective is to lengthen mission time, reduce fuel expenditure, increase area coverage, or stretch on-station time, the same platform could be utilized for a variety of missions with only minor modifications to the problem formulation to account for the new cost function, as desired.

As supercomputers improve each year, it may not be long before they achieve parity with the human brain, and not long after that before smaller computers can accomplish the same [4]. To be truly autonomous, a vehicle must have intelligent software capable of autonomously determining the best way to complete a high-level task. Solving a motion planning problem using optimal control tools such as DIDO (the implementation of pseudospectral optimal control theory) is one promising approach. Other institutions and organizations are also exploring options such as modified intelligent software agents [17], [18], [19], [20], [21], [22]. These technological advancements will make real-time optimization and intelligent AAV systems achievable in the future, and this will open up a whole range of possibilities for AAV missions and design.

D. RESEARCH OBJECTIVES

This thesis addresses two main questions; first, whether the generated optimal trajectories can be executed with the selected hardware; and second, how the fidelity of the optimal control model influences accomplishing this task.

1. Feasibility of Execution with Selected Hardware

Can a specific fixed-wing, COTS RC aircraft, fitted with a simple control system and low-cost sensors, accurately execute optimal trajectories? This question must be answered first, in order to move on to the next question. The aircraft autopilot must be

capable of taking the generated trajectory commands and successfully completing maneuvers that closely follow the desired trajectory.

2. Model Fidelity Level

What level of model fidelity is required to generate trajectories that can be reliably flown? This second question asks if a lower-fidelity model that makes assumptions to simplify conditions and inputs can create adequate trajectories to fly real-world maneuvers. The 3DOF model could be a simplified point-mass model, or it could include the influences of control surfaces and flight dynamics, i.e. a reduced-order version of the complex 6DOF flight dynamics. If it is possible to fly trajectories generated with this reduced-order model, it will save time, complexity, and computational cost over using the full-order 6DOF model. This aspect is a crucial step towards developing an intelligent autopilot algorithm that can generate optimal trajectories in real time.

E. ORGANIZATION, METHODOLOGY AND SCOPE

This thesis begins with a description of the process used to develop and verify the optimal trajectories to answer the research questions as stated. Chapter II discusses the selection and modeling of the test aircraft. Chapter III discusses the problem statement and setup, and the step-by-step development and troubleshooting of the problem formulation and code in order to produce a representative model and feasible trajectories. Chapter III also provides an introduction to the tools utilized to create the trajectories and solve the minimum time maneuvering problem.

Chapter IV presents a collection of canonical maneuvers created and simulated in MATLAB using DIDO optimal control software, and Chapter V continues on to describe the trajectory implementation environment including the autopilot system, simulator, and verification procedures of the MONARC optimization model.

Chapter VI presents the results of a more complex scenario aimed at demonstrating the application of a typical canonical maneuver for a real aircraft. In particular the results include a 6DOF simulation of the mission scenario as well as a

discussion of maneuver implementation as part of a HIL pre-flight checkout. Finally, Chapter VII discusses future work to include physical flight tests and improvement of the model based on that test data.

This work is aimed at developing technology that can be flight-demonstrated; however due to a number of regulatory issues beyond the control of the author, the scope of this work does not include physical flight test of the maneuvers. Moreover, only the minimum-time maneuvering problem is addressed. Other objective functions are easy to implement in the presented framework but are not studied here. Flight testing was planned and an aircraft platform is flight-ready, but due to an unfortunate accident with an NPS unmanned aerial system at the local airfield during the period of this thesis, all NPS flight testing was put on hold as of 07 March 2012. Pursuant to this accident, a NAVAIR investigation was required, and all flight testing of UAVs involving NPS aircraft became subject to approval, stringent requirements, and numerous prerequisites per OPNAVINST 3750.6, NAVPGSCOLINST 3700.1, and NAVAIR airworthiness standards. The required Naval Safety Center Aviation Safety Survey Checklist has been completed, and a 90-day waiver for flying the MONARC was granted on 27 November 2012. Unfortunately this approval was not granted with enough time to allow for scheduling of facilities and completion of initial checkout and flight testing before completion of this thesis.

This thesis is part of an ongoing effort to eventually produce a variety of vehicles that are capable of truly autonomous operations, using pseudospectral motion planning as a key enabling technology. The autopilot is currently being developed for use with both a Traxxas Summit ground vehicle [23] and a Multiplex Mentor aircraft as described in Chapter II. The autopilot is intended to eventually be readily adaptable for use in multiple environments and vehicles, including but not limited to fixed wing aircraft, quad-rotor aircraft, ground vehicles, earth-orbiting satellites, and planetary aerial vehicles.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THE MONARC AIRCRAFT

A. SELECTION OF TEST AIRCRAFT

1. Selection Criteria

A variety of aircraft were considered as test platforms for demonstrating the implementation of the designed canonical maneuvers. These included a battery-powered quad-rotor craft, gas-powered kit model airplanes, and several other similar fixed-wing, battery-powered, COTS model aircraft. It was considered desirable to choose an aircraft that would be transportable to the airfield inside a personal automobile, and that the aircraft be battery-powered rather than gas-powered for storage and transport reasons.

The Multiplex Mentor aircraft [24] was selected in part due to its low cost, ready availability for purchase from major hobby retail stores such as Hobby Warehouse and RC Planet, and its popularity as a recreational RC aircraft—this meant that parts would be readily available, as well as many forums with solutions to any known issues with the platform. Another reason the Mentor was selected is that it has been used for research at the other universities. Working with the same platform will help to facilitate sharing of data, information, and lessons learned.

Four aircraft were purchased, and three constructed: one by fellow student Robert Casey, one by the author, and one by the safety pilot and NPS UAV expert Dr. Kevin Jones. The remaining kit was utilized for spare parts. The redundancy in test models allows for better and more varied system identification data collection, and in the future will allow for formation flights of multiple autonomous vehicles.

2. Physical and Aerodynamic Characteristics

The Mentor Optimal-trajectory Naval Autonomous Research Craft (MONARC) is built around a modified Mentor airframe, with equipment and airframe changes to support an unmanned avionics (autopilot) system and the associated sensors.

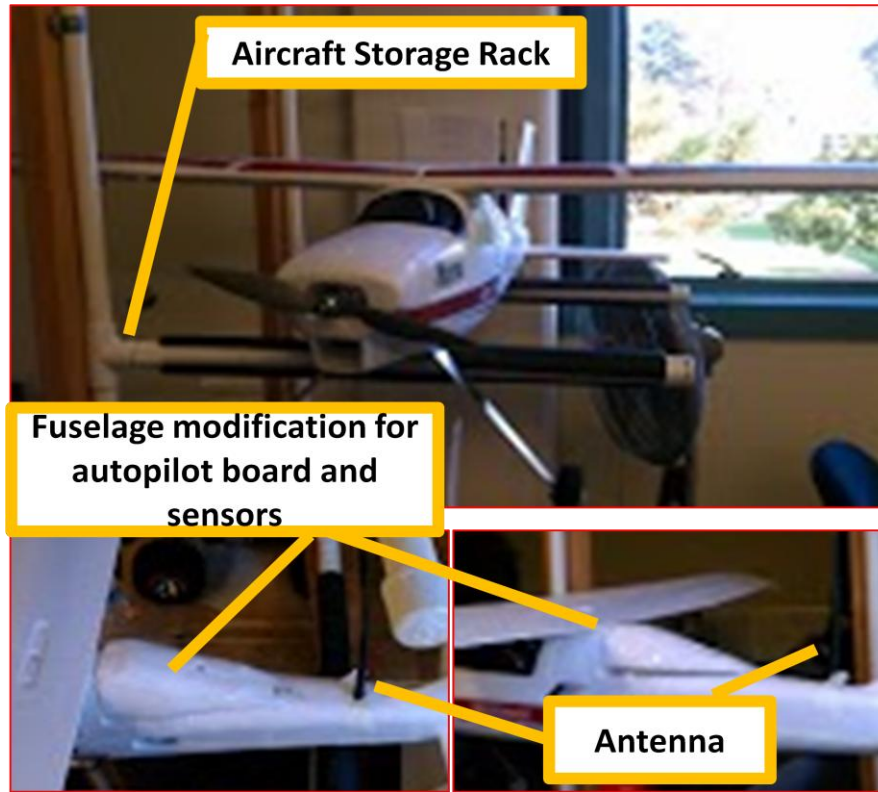


Figure 3. MONARC Aircraft with Views of Modified Fuselage

Figure 3 shows one of the MONARC aircraft in its storage rack, and close-up views of the fuselage modifications to accommodate the autopilot board, sensors, and antenna. Estimates of some aircraft performance parameters and constants are in Table 1. The minimum flight velocity is the stall speed, and maximum flight velocity is the maximum dive velocity of the aircraft, as given by the manufacturer [24], [25] and obtained from a SIMULINK model of the aircraft. Maximum range and endurance are taken from the capability of the motor and propeller used with the MONARC. The aircraft mass is an average value based on the Mentor airframe plus autopilot equipment taken for the three aircraft. The wing area, wingspan, and center of gravity were measured physically. The aerodynamic coefficients were determined from an existing SIMULINK model and scaled down from an aircraft with known coefficients [26], [27]. Maximum and minimum thrust values were determined from simulations

using the SIMULINK model. All of these values will be used for maneuver generation in this thesis. It is anticipated, however, that flight testing will provide more accurate numerical values.

MONARC Aircraft data and constants:

Parameter	Symbol	Value	Units	Notes
Minimum flight velocity	V_{min}	13.9	m/s	Aircraft stall speed
Maximum flight velocity	V_{max}	41.7	m/s	Maximum dive velocity
Median flight velocity	V_{med}	27.5	m/s	
Aircraft mass	m	2	kg	
Wing area	S	0.982	m ²	
Length	-	1.17	m	
Wingspan	-	1.63	m	
Center of Gravity	CG	0.083	m	measured towards tail end from wing's leading edge at fuselage
Minimum Thrust	T_{min}	3	N	
Maximum Thrust	T_{max}	35	N	
Maximum Flight Endurance	-	45	min	
Maximum Range	-	13	km	
Range vs. Altitude (glide capability)	-	500/100	m	500 m horizontal travel per 100 m vertical drop

Table 1. Estimated MONARC Aircraft Physical and Performance Parameters

3. Assembly

Each aircraft was assembled according to the manufacturer's kit directions [25], and then modified in order to integrate the autopilot and supporting hardware. The airframe modifications included removal of small sections of fuselage and wing to make room for the autopilot board and its peripherals such as the antenna, wiring, Pitot tube, GPS, and other attached instruments.

The aircraft can be flown in manual mode, using a conventional RC control, or it can be controlled using the control logic of the autopilot. There is also a HIL mode that allows the autopilot hardware to be driven using simulated sensor inputs derived from a 6DOF aircraft model.

B. MODELING THE MONARC AIRCRAFT

A preliminary model of the MONARC aircraft was created in order to generate trajectories that could be flown by the actual aircraft. The model characteristics were determined from measurement of physical parameters on actual flight hardware, as well as from computer simulation or estimates based on best engineering judgment.

1. The 6DOF Dynamics Model

A full dynamic model of an aircraft considers all six degrees of freedom for the aircraft motion, and has twelve states with a reasonably high level of calculation complexity.

a. Reference Frame

The 6DOF model uses a flat-Earth NED reference frame for position, and body-axes reference frame for all other states, as shown in Figures 4 and 5.

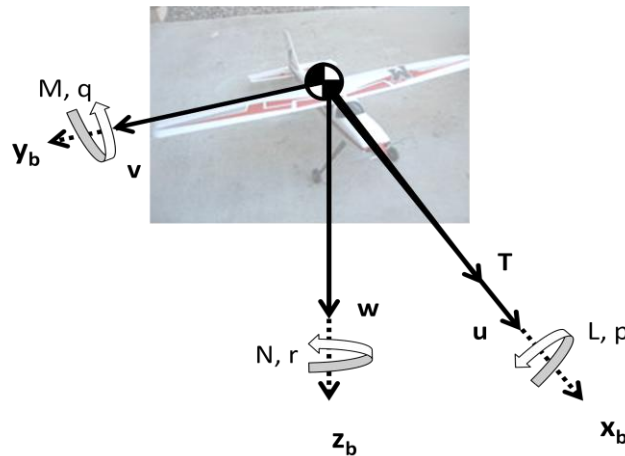


Figure 4. Body-Axis Coordinate System

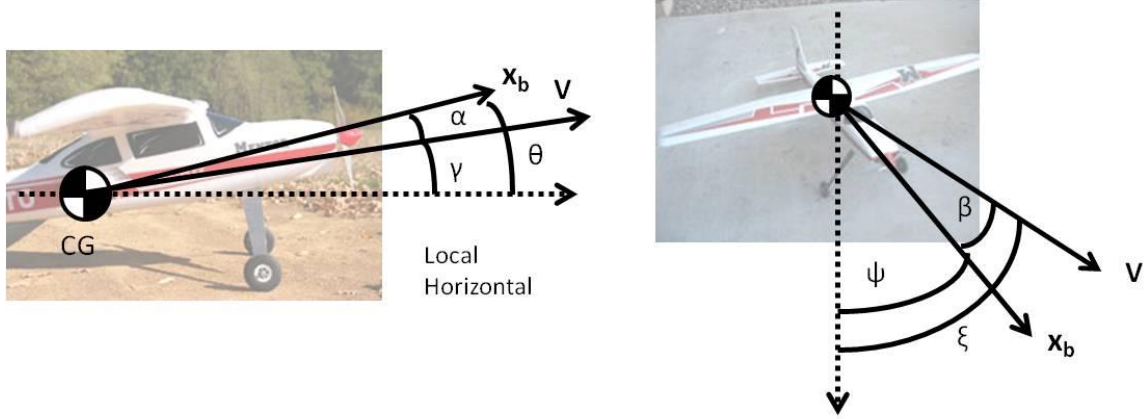


Figure 5. 6DOF Longitudinal and Directional Flight Angles (After [28])

b. Equations of Motion

The 6DOF equations of motion are functions of states, forces, moments, and velocity components as shown in Table 2.

	Roll (x-axis)	Pitch (y-axis)	Yaw (z-axis)
Angular Rates	p	q	r
Velocity Components	u	v	w
Aerodynamic Force Components	X	Y	Z
Aerodynamic Moment Components	L	M	N
MOI about each axis	I_x	I_y	I_z

Table 2. 6DOF Forces, Moments, and Velocity Components (After [28])

The aerodynamic forces and moments on the aircraft in this model are listed in Table 3, and the twelve states for the 6DOF model are listed in Table 4.

The 6DOF model is a nonlinear model in the Flat-Earth, Body-Axes reference frame. The standard 6DOF equations of motion from [28], [30] include the position Equations (2.1), the control rate Equations (2.2), the angular velocity Equations (2.3), the Euler angle rate Equations (2.4), and the force and moment Equations (2.5).

Aerodynamic Forces and Moments	
X	Axial Force
Y	Side Force
Z	Normal Force
L	Rolling Moment
M	Pitching Moment
N	Yawing Moment

Table 3. 6DOF Aerodynamic Forces and Moments

6DOF Aircraft States:

Name	symbol	units
Velocity	V	m/s
Angle of Attack	α	rad
Sideslip Angle	β	rad
Roll rate	p	rad/s
Pitch rate	q	rad/s
Yaw rate	r	rad/s
Roll angle	ψ	rad
Pitch angle	θ	rad
Yaw Angle	ϕ	rad
X-coordinate (Earth Axes)	X _e	m
Y-coordinate (Earth Axes)	Y _e	m
Altitude	Z	m

Table 4. 6DOF Aircraft States (After [28, 29])

Position equations for the nonlinear 6DOF model are given in the earth-reference frame as:

$$\begin{aligned}
\dot{x}_e &= [\cos \theta \cos \psi]V \cos \alpha \cos \beta + [\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi]V \sin \beta \\
&\quad + [\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi]V \sin \alpha \cos \beta \\
\dot{y}_e &= [\cos \theta \sin \psi]V \cos \alpha \cos \beta + [\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi]V \sin \beta \\
&\quad + [\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi]V \sin \alpha \cos \beta \\
\dot{z}_e &= [-\sin \theta]V \cos \alpha \cos \beta + [\sin \phi \cos \theta]V \sin \beta \\
&\quad + [\cos \phi \sin \theta]V \sin \alpha \cos \beta
\end{aligned} \tag{2.1}$$

While the 6DOF control rate, angular velocity, Euler angle rate equations, and force and moment equations are given in the body-reference frame as:

$$\begin{aligned}\dot{V} = & \left[\frac{X}{m} - g \sin \theta \right] \cos \alpha \cos \beta + \left[\frac{Y}{m} + g \sin \phi \cos \theta \right] \sin \beta \\ & + \left[\frac{Z}{m} + g \cos \phi \cos \theta \right] \sin \alpha \cos \beta \\ \dot{\alpha} = & \frac{1}{mV \cos \beta} [Z \cos \alpha - X \sin \alpha] + \frac{g}{V \cos \beta} [\cos \phi \cos \theta \cos \alpha + \sin \theta \sin \alpha] \\ & + q - \tan \beta [p \cos \alpha + r \sin \alpha]\end{aligned}\quad (2.2)$$

$$\begin{aligned}\dot{\beta} = & \frac{\cos \beta}{mV} [Y + mg \sin \phi \cos \theta] + p \sin \alpha - r \cos \alpha \\ & - \frac{\sin \beta}{mV} [Z \sin \alpha + X \cos \alpha] - \frac{g \sin \beta}{V} [\cos \phi \cos \theta \sin \alpha - \sin \theta \cos \alpha]\end{aligned}$$

$$\begin{aligned}\dot{p} = & \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^2} [L - (I_{zz} - I_{yy})qr + I_{xz}qp] \\ & + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} [N + (I_{xx} - I_{yy})qp - I_{xz}qr] \\ \dot{q} = & \frac{1}{I_{yy}} [M - (I_{xx} - I_{zz})pr - I_{xz}(p^2 - r^2)] \\ \dot{r} = & \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} [L + (I_{yy} - I_{zz})qr + I_{xz}qp] \\ & + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} [N + (I_{xx} - I_{yy})qp - I_{xz}qr]\end{aligned}\quad (2.3)$$

$$\begin{aligned}\dot{\phi} = & p + (q \sin \phi + r \cos \phi) \tan \theta \\ \dot{\theta} = & q \cos \phi - r \sin \phi \\ \dot{\psi} = & (q \sin \phi + r \cos \phi) \sec \theta\end{aligned}\quad (2.4)$$

The forces and moments depend on the aircraft configuration and flight requirements are computed from

$$\begin{aligned}
X &= \frac{1}{2} \rho(z) V^2 S_{ref} C_X(\alpha, \beta, M, \underline{\delta}) + T_x \\
Y &= \frac{1}{2} \rho(z) V^2 S_{ref} C_Y(\beta, M, \underline{\delta}) + T_y \\
Z &= \frac{1}{2} \rho(z) V^2 S_{ref} C_Z(\alpha, \beta, M, \underline{\delta}) + T_z \\
L &= \frac{1}{2} \rho(z) V^2 S_{ref} b C_l(\beta, M, \underline{\delta}) \\
M &= \frac{1}{2} \rho(z) V^2 S_{ref} \bar{c} C_m(\alpha, \beta, M, \underline{\delta}) \\
N &= \frac{1}{2} \rho(z) V^2 S_{ref} b C_n(\beta, M, T, \underline{\delta})
\end{aligned} \tag{2.5}$$

The 6DOF model is prohibitively complex for solving the optimization problem in a timely manner [28]. One very recent way that has been used to get around this problem is to use a 3DOF solution as a bootstrap to the 6DOF problem [31]. This thesis takes the approach of creating a model that is of higher fidelity than the simplified 3DOF point mass problem, but that is less complex than the full-order 6DOF model. As will be seen the fidelity of the 3DOF model is sufficient for flight implementation.

2. SIMULINK Flight Control Toolbox Model

The FDC toolbox in SIMULINK contains a full model of a 6DOF De Havilland Beaver aircraft, complete with autopilot functions and flight-verified aerodynamic coefficients, MoI, and performance characteristics [26]. This model was scaled down and modified into a model of the Mentor by Dr. K. Lee [27], utilizing performance criteria and physical characteristics of the Mentor aircraft, either estimated or measured from testing of the MONARC. Actual flight data will be used later to iteratively upgrade the model. The procedure used to develop and test the model is compared with the general procedure for traditional full-scale aircraft testing in Figure 6.

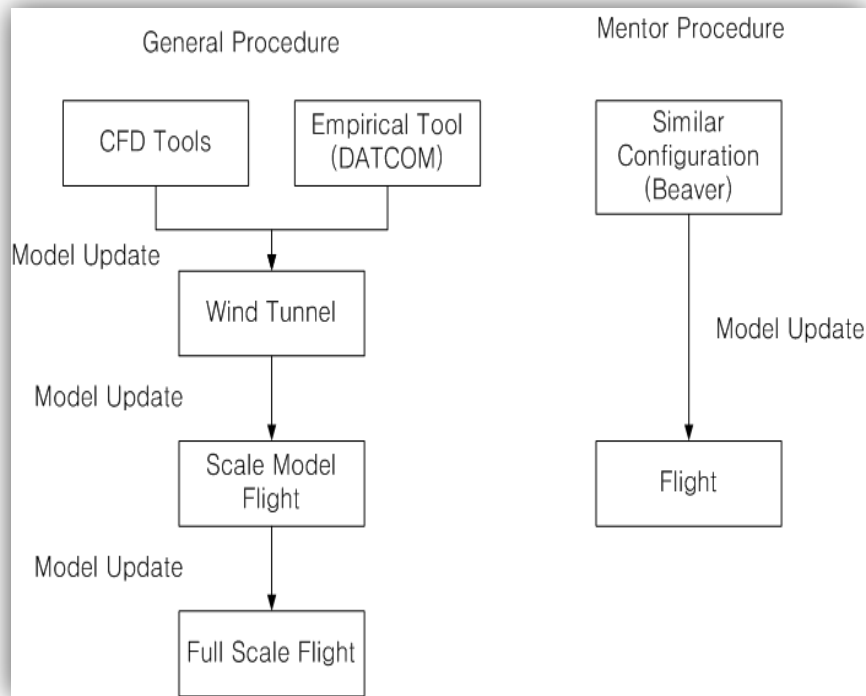


Figure 6. Procedures for Aircraft Modeling (From [27])

The model was scaled down using the SIMULINK FDC Toolbox version 1.4, and the initial modifications to the Beaver model included adjusting to the Mentor's weight and CG, and scaling down the wing characteristics to those of the Mentor [27]. The aerodynamic characteristics were modified using computer simulation test data.

To determine the moments of inertia of the aircraft, a swing test was performed using the MONARC aircraft and the autopilot onboard sensors. The data collected was incorporated with the SIMULINK model. A photo of the pendulum test apparatus in the lab is shown in Figure 7.

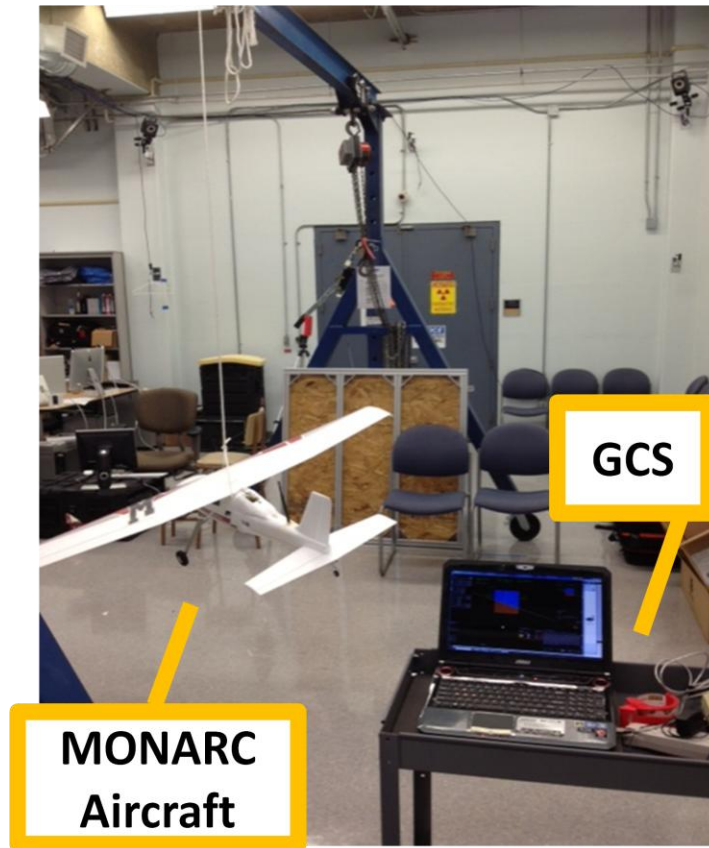


Figure 7. MoI Test Apparatus

a. *SIMULINK Model*

The SIMULINK model serves a computer simulation platform for implementation of the desired maneuver trajectories to show whether the trajectory is flight-feasible for the subject aircraft. The model is also utilized for trajectory comparison of optimized maneuvers versus the trajectory flown by a more traditional autopilot control system. The structure of the SIMULINK model Control Law Structure is shown in Figure 8, and the full model with input and output definitions is shown in Figure 9.

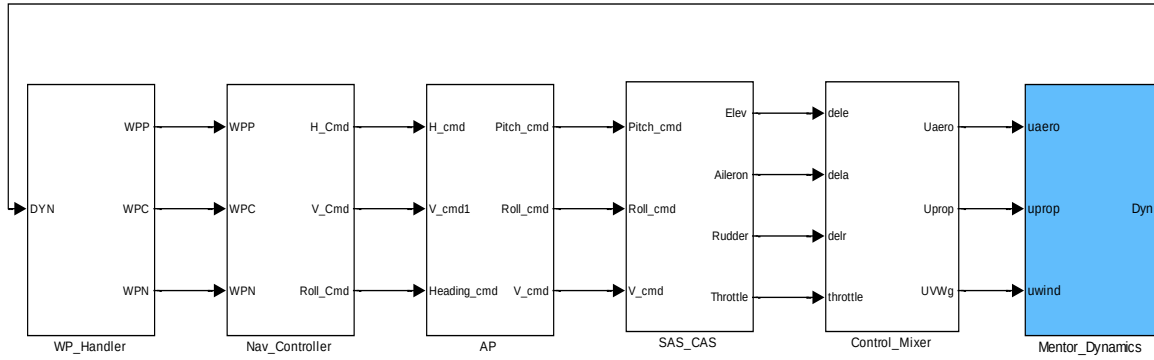


Figure 8. SIMULINK Model Control Law Structure (From [27])

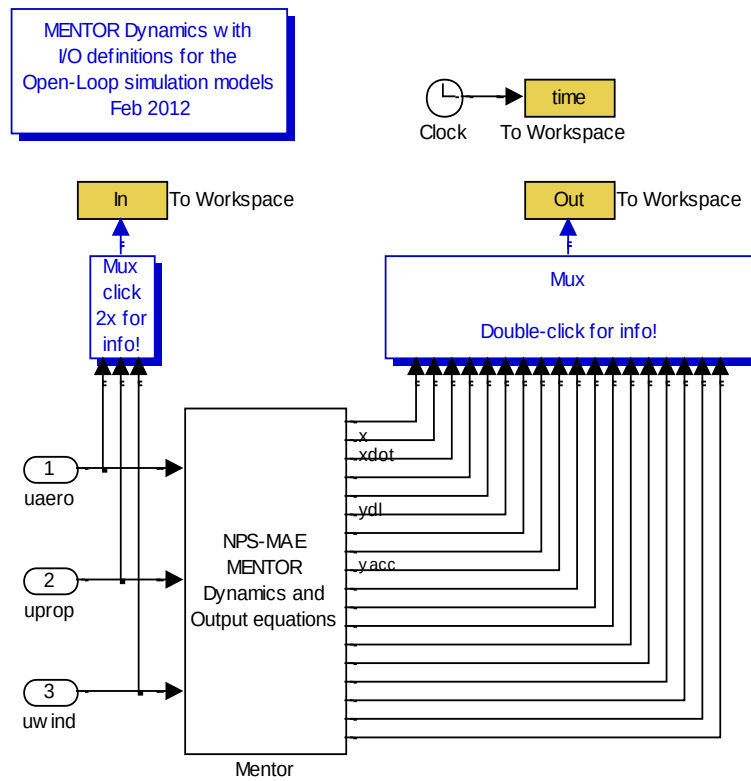


Figure 9. MONARC SIMULINK Model—Highest Level View (From [27])

b. Aerodynamic Coefficients

As mentioned, the aircraft's aerodynamic coefficients were determined by scaling down an existing aircraft model with a similar configuration. The 6DOF aerodynamic coefficients used for all trajectories generated in this thesis are listed in Table 5.

CX0	= -0.03554;	CZ0	= -0.05504;	Cm0	= 0.09448;
CXa	= 0.002920;	CZa	= -5.578;	Cma	= -0.6028;
CXa2	= 5.459;	CZa3	= 3.442;	Cma2	= -2.140;
CXa3	= -5.162;	CZq	= -2.988;	Cmq	= -15.56;
CXq	= -0.6748;	CZde	= -0.3980;	Cmde	= -1.921;
CXdr	= 0.03412;	CZdeb2	= -15.93;	Cmb2	= 0.6921;
CXdf	= -0.09447;	CZdf	= -1.377;	Cmr	= -0.3118;
CXadf	= 1.106;	CZadf	= -1.261;	Cmdf	= 0.4072;
CY0	= -0.002226;	Cl0	= 0.0005910;	Cn0	= -0.003117;
CYb	= -0.7678;	Clb	= -0.06180;	Cnb	= 0.006719;
CYp	= -0.1240;	Clp	= -0.5045;	Cnp	= -0.1585;
CYr	= 0.3666;	Clr	= 0.1695;	Cnr	= -0.1112;
CYda	= -0.02956;	Cl da	= -0.09917;	Cnda	= -0.003872;
CYdr	= 0.1158;	Cl dr	= 0.006934;	Cndr	= -0.08265;
CYdra	= 0.5238;	Cl daa	= -0.08269;	Cnq	= 0.1595;
CYb dot	= -0.1600;			Cnb3	= 0.1373;

Table 5. Preliminary MONARC Aerodynamic Coefficients (From [32])

The pertinent coefficients for use in the 3DOF optimization model are CX0, CZ0, CXa, and CZa, which are used to calculate C_L and C_D for the model. The aerodynamic coefficients will be updated using actual flight data from the system identification flight tests when available, to improve the fidelity of the model.

c. Control limits

The limits on inner loop control system performance can be used for bounds in the optimal control problem to ensure that the designed maneuver can be physically implemented. Their values were determined from the SIMULINK control system model based on the closed-loop response times of the various inner loop control systems. Actual control limits will be determined via flight test, and incorporated into the model. The thrust limits were estimated based on the aircraft size and mass and the motor specifications. The heading rate control limits are from the SIMULINK model. Control limits are listed with the maneuver setup in Chapter IV.

III. THE TRAJECTORY OPTIMIZATION PROBLEM FOR A FIXED-WING AIRCRAFT

A. DESCRIPTION OF PROBLEM

The desired outcome of the trajectory optimizations in this thesis are minimum-time maneuvers that can be performed more quickly than similar conventional maneuvers. The optimal control solution could also be performed for maximizing or minimizing a number of other objectives, including fuel consumption, time on station, and area covered.

In order to solve the underlying optimal control problem and find the best trajectory to perform a given task, an optimal control software tool is used. For all maneuvers in this work, DIDO pseudospectral optimal control software was used.

B. THE OPTIMAL CONTROL PROBLEM

The generic optimization problem P_O from [7], [28], [33], [34] is

$$(P_o) \left\{ \begin{array}{l} \text{Minimize } J[x(\bullet), u(\bullet), t_f] = E(x_f, t_f) + \int_{t_0}^{t_f} F(x(t), u(t)) dt \\ \text{Subject to :} \\ \dot{x}(t) = f(x(t), u(t)) \\ x(t_0) = x^0 \\ t_0 = t^0 \\ t_f = t^f \\ e(x(t_f)) = 0 \end{array} \right. \quad (3.1)$$

Where J is the cost functional, E is the endpoint cost, F is the running cost, x is the set of state variables, $\dot{x} = f(x, u)$ is the set of dynamics equations, u is the set of control variables, and e is the set of endpoint constraints. A set of path constraints, $h(x, u)$, may also be applied in some circumstances. These are not shown in (3.1).

The necessary conditions for optimality are given by [7], [28], [33], [34]

$$\begin{aligned}
\lambda(t) &= \frac{-\partial H}{\partial x} && \text{adjoint equation} \\
\frac{\partial H}{\partial u} &= 0 && \text{Euler-Lagrange equation} \\
\lambda(t_f) &= \frac{-\partial \bar{E}}{\partial x_f} && \text{transversality condition}
\end{aligned} \tag{3.2}$$

where H is the Hamiltonian defined as

$$H(\lambda, x, u) = F(x, u) + \lambda^T f(x, u) \tag{3.3}$$

and the Endpoint Lagrangian is

$$\bar{E}(v, x(t_f)) = E(x(t_f)) + v^T e(x(t_f)) \tag{3.4}$$

For the constrained dynamic optimization problem the Euler-Lagrange equations are not valid, so the optimal control problem P_{OC} must be used instead [33], [34]

$$(P_{OC}) \left\{ \begin{aligned} & \text{Minimize } J[x(\bullet), u(\bullet), t_f] = E(x(t_f)) + \int_{t_0}^{t_f} F(x(t), u(t)) dt \\ & \text{Subject to:} \\ & \dot{x}(t) = f(x(t), u(t)) \\ & x(t_0) = x^0 \\ & t_0 = t^0 \\ & e(x_f, t_f) = 0 \\ & u^L \leq u(t) \leq u^U \end{aligned} \right. \tag{3.5}$$

For the optimal control problem, the control is now bounded, rather than using open sets. The Hamiltonian, adjoint, and transversality conditions remain the same, but instead of the Euler-Lagrange equations, the problem now makes use of the Hamiltonian Minimization Condition [34].

$$(HMC) \begin{cases} \text{Minimize} & H(\lambda, x, u) \\ \text{Subject to} & u^L \leq u \leq u^U \end{cases} \quad (3.6)$$

The value of the minimized Hamiltonian is a constant as a function of time, and in the case of a minimum time problem the value is equal to a constant $H(t) = -1$ [34].

C. CREATING A DYNAMIC MODEL OF THE MONARC AIRCRAFT

The equations of motion, $\dot{x}(t) = f(x(t), u(t))$ that are used in the OCP formulation are derived from standard 6DOF equations of motion for an aircraft, as presented in [30]. The full order 6DOF model comprises 12 equations of motion (see Chapter II) and is reasonably complex. The complexity of the problem is further increased with the addition of approximated atmospheric conditions and other external effects on the aircraft. To arrive at a suitable reduced-order model, several iterations were performed. These are elaborated on next.

The first step in the early development of the model was to use the most simple 3DOF kinematic model possible to gain familiarity with the software, procedures, and maneuvering of the aircraft. Next, the model was upgraded using a more complex 3DOF model that was reduced from the full-order 6DOF model but takes into account values for thrust and the presence of actual aircraft control surfaces.

1. The Simple 3DOF Point-Mass Model

The simple 3DOF model assumed a point-mass aircraft. The minimum time optimal control problem was set up for the simple 3DOF model using the form of Equation (3.5) and the state and control variables listed in Tables 6 and 7.

Control Name	symbol	units
Acceleration	u_a	m/s^2
Pitch rate of change	u_γ	rad/sec
Heading rate of change	u_σ	rad/sec

Table 6. Controls for Point-mass 3DOF Model

State Name	symbol	units
x-position (downrange)	x	Meters, m
y-position (cross-range)	y	Meters, m
z-position (altitude)	z	Meters, m
Velocity (airspeed)	v	Meters per second, m/s
Flight Path Elevation Angle	γ	Radians, rad
Flight Path Heading Angle	σ	Radians, rad

Table 7. States for Point-mass 3DOF Model

a. The Minimum Time Problem Formulation

For the minimum-time problem, the endpoint cost is the final time, t_f , and there is no running cost. Using (3.5) and a point-mass model, the minimum time optimal control problem is

$$\begin{aligned}
 (P_{OC \text{ Min. Time}}) \left\{ \begin{array}{l} \text{Minimize } J[y(\bullet), u(\bullet), t_f] = t_f \\ \text{Subject to:} \\ \dot{x} = v \cos \gamma \cos \sigma \\ \dot{y} = v \cos \gamma \sin \sigma \\ \dot{z} = v \sin \gamma \\ \dot{v} = u_a \\ \dot{\gamma} = u_\gamma \\ \dot{\sigma} = u_\sigma \\ x(t_0) = x^0 \\ t_0 = t^0 \\ e(x(t_f)) = 0 \\ u^L \leq |u| \leq u^U \end{array} \right. \quad (3.7)
 \end{aligned}$$

b. Reference Frame

The reference frame used for the simplified 3DOF point mass model is a body-reference frame for flight angles and a standard Cartesian coordinate system for the position states (see Figure 10).

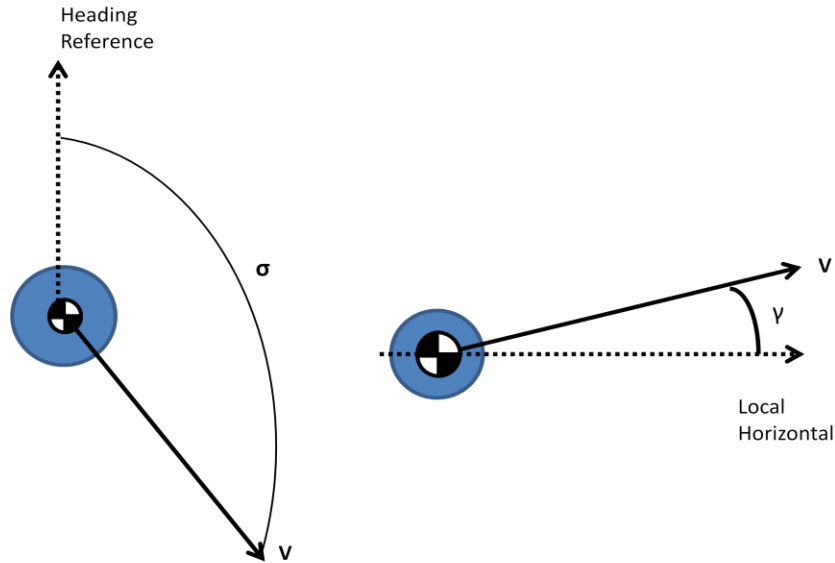


Figure 10. 3DOF Simplified Point-mass Model Flight Angles

c. Disadvantages of the Simplified Point-Mass 3DOF Model

While the 3DOF point-mass model was highly useful for familiarization and development of the basic code, it did not allow for modeling the effects of control surfaces—the simple 3DOF model used a point mass that can only change heading, velocity, and position, without taking into account turn rates, thrust control, or the ability of actuators and control surfaces to make turns. A higher-fidelity model is required to effectively model how an aircraft in flight will actually behave, in order to generate trajectories that may be flown by a real aircraft.

2. The 3DOF Dynamic Model

Using certain assumptions, the full 6DOF model may be reduced to a reasonable 3DOF approximation. This allows for enough fidelity to generate trajectories that an AAV may fly, but reduces the complexity of the computations so that the calculations may be made in a reasonable amount of time, with the eventual goal being to enable an AAV to solve optimal trajectories in real-time.

a. Model Assumptions

The assumptions made in order to simplify the model are listed in Table 8:

Model Assumptions:
Flat, non-rotating Earth
No wind
No gravity variations
No coriolis effect
Rigid body vehicle
Constant mass
Negligible cross-products of inertia
Z=0 is at sea level, standard atmosphere
Steady, coordinated turns with no side-slip

Table 8. 3DOF Model Assumptions

b. 3DOF Dynamics

A 3DOF dynamics model with side-slip and thrust consists of the following dynamic equations [28]:

$$\begin{aligned}
 \dot{x} &= v \cos \gamma \cos \sigma \\
 \dot{y} &= v \cos \gamma \sin \sigma \\
 \dot{z} &= v \sin \gamma \\
 \dot{v} &= \frac{1}{m} (-D \cos \beta + Y \sin \beta + T \cos \beta \cos \alpha) - g \sin \gamma \\
 \dot{\gamma} &= \frac{1}{mv} (-D \sin \beta \sin \mu - Y \sin \mu \cos \beta + L \cos \mu \\
 &\quad + T (\cos \mu \sin \alpha + \sin \mu \sin \beta \cos \alpha)) - \frac{g}{v} \cos \gamma \\
 \dot{\sigma} &= \frac{1}{mv \cos \gamma} (D \sin \beta \cos \mu + Y \cos \mu \cos \beta + L \sin \mu \\
 &\quad + T (\sin \mu \sin \alpha - \cos \mu \sin \beta \cos \alpha))
 \end{aligned} \tag{3.8}$$

This set of equations is clearly of significantly greater complexity than the point mass model (3.7) and allows for simulation of control surfaces. At the same time it

is reduced enough in complexity from the 6DOF model that it could eventually allow for real-time optimization.

c. Aerodynamics and Atmospheric Conditions

The model uses standard atmospheric conditions, which vary by altitude. A basic air density estimation equation is used

$$\rho = 1.21e^{-z/8000} \text{ (kg/m}^3\text{)} \quad (3.9)$$

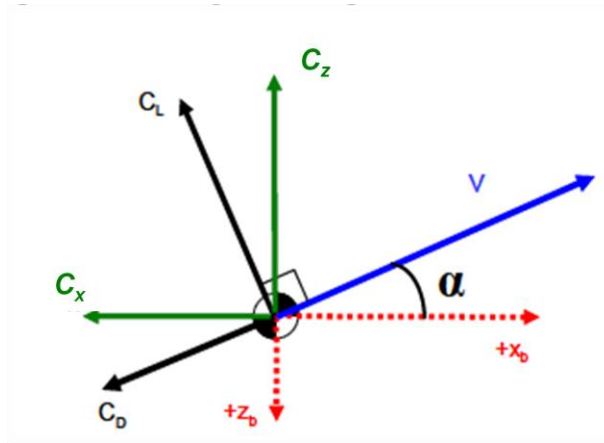


Figure 11. Relationship between Aerodynamic Coefficients when $\beta=0$ (After [28])

The aerodynamic coefficients C_z and C_x are converted to C_L and C_D for use in the lift and drag equations (see Figure 11):

$$C_L = -C_x \sin \alpha + C_z \cos \alpha \quad (3.10)$$

$$C_D = -C_x \cos \alpha + C_z \sin \alpha \quad (3.11)$$

The lift and drag equations are used to calculate L and D values for the dynamics equations

$$L = \frac{1}{2} \rho V^2 S C_L \quad (3.12)$$

$$D = \frac{1}{2} \rho V^2 S C_D \quad (3.13)$$

d. Final 3DOF Dynamic Model

In order to further reduce the complexity of the model, several assumptions are made in addition to those from Table 8. It is also assumed that all turns are perfectly coordinated, with no side-slip. In the absence of side-slip, β and $\dot{\beta}$ are equal to zero, causing all terms containing β to drop out and reducing the equations in (3.8) as follows:

$$\begin{aligned}
 \dot{x} &= v \cos \gamma \cos \sigma \\
 \dot{y} &= v \cos \gamma \sin \sigma \\
 \dot{z} &= v \sin \gamma \\
 \dot{v} &= \frac{1}{m} (-D + T \cos \alpha) - g \sin \gamma \\
 \dot{\gamma} &= \frac{1}{mv} (L \cos \mu + T \cos \mu \sin \alpha) - \frac{g}{v} \cos \gamma \\
 \dot{\sigma} &= \frac{1}{mv \cos \gamma} (L \sin \mu + T \sin \mu \sin \alpha)
 \end{aligned} \tag{3.14}$$

Since the aircraft will be under control of an autopilot, which implements inner loop control logic, the control variables are taken as

$$\begin{aligned}
 \dot{T} &= u_T \\
 \dot{\alpha} &= u_\alpha \\
 \dot{\mu} &= u_\mu
 \end{aligned} \tag{3.15}$$

This enables the optimal control solution to accommodate inner-loop time constants.

Nine states for the aircraft are described by the reduced-order dynamic model (see Table 9), rather than the twelve states of the full-order model or six states of the more simplified 3DOF model. In addition to position and velocity, the states also include Thrust and four angles: flight path elevation angle, which is the angle between the x-y plane and the nose of the aircraft; the flight path heading angle, which is the direction the aircraft is pointed in; the angle of attack, which is the angle between the relative wind and nose of the aircraft, or in other words the pitch of the aircraft; and the bank angle of the aircraft, which combines the elements of yaw and roll.

State	symbol	units
x-position (downrange)	x	Meters, m
y-position (cross-range)	y	Meters, m
z-position (altitude)	z	Meters, m
Velocity (airspeed)	v	Meters per second, m/s
Flight Path Elevation Angle	γ	Radians, rad
Flight Path Heading Angle	σ	Radians, rad
Thrust	T	Newtons, N
Angle of Attack	α	Radians, rad
Flight Path Bank Angle	μ	Radians, rad

Table 9. Reduced-Order 3DOF Model States

The reduced-order model controls are then:

Control Name	symbol	units
Thrust rate of change	u_T	N/sec
AoA rate of change	u_α	rad/sec
BA rate of change	u_μ	rad/sec

Table 10. Reduced-order 3DOF Controls

e. Minimum Time Problem Formulation

Using (3.5), (3.14), and (3.15) along with Tables 9 and 10, the minimum time problem for the reduced-order 3DOF model is

$$\begin{aligned}
& \text{Minimize } J[y(\bullet), u(\bullet), t_f] = t_f \\
& \text{Subject to:} \\
& \dot{x} = v \cos \gamma \cos \sigma \\
& \dot{y} = v \cos \gamma \sin \sigma \\
& \dot{z} = v \sin \gamma \\
& \dot{v} = \frac{1}{m}(-D + T \cos \alpha) - g \sin \gamma \\
& \dot{\gamma} = \frac{1}{mv}(L \cos \mu + T \cos \mu \sin \alpha) - \frac{g}{v} \cos \gamma \\
& \dot{\sigma} = \frac{1}{mv \cos \gamma}(L \sin \mu + T \sin \mu \sin \alpha) \\
& (P_{OC \text{ Min Time}}) \left\{ \begin{aligned}
& \dot{T} = u_T \\
& \dot{\alpha} = u_\alpha \\
& \dot{\mu} = u_\mu \\
& \bar{x}(t_0) = \bar{x}^{-0} \\
& t_0 = t^0 \\
& \bar{e}(x(t_f)) = 0 \\
& u_T^L \leq |u_T| \leq u_T^U \\
& u_\alpha^L \leq |u_\alpha| \leq u_\alpha^U \\
& u_\mu^L \leq |u_\mu| \leq u_\mu^U
\end{aligned} \right. \tag{3.16}
\end{aligned}$$

f. Analysis of the OCP

Applying the procedure from equations (3.2) through (3.7) and (3.10), analysis of the optimal control problem in this case begins with formulation of the necessary conditions for optimality. Using (3.3) to formulate the Hamiltonian for this problem, the running cost term can be eliminated as there is only an end cost in this case, and the Hamiltonian is then

$$\begin{aligned}
H(\lambda, x, u) = & \lambda_x [v \cos \gamma \cos \sigma] + \lambda_y [v \cos \gamma \sin \sigma] + \lambda_z [v \sin \gamma] \\
& + \lambda_v \left[\frac{1}{m} (-D + T \cos \alpha) - g \sin \gamma \right] \\
& + \lambda_\gamma \left[\frac{1}{mv} (L \cos \mu + T \cos \mu \sin \alpha) - \frac{g}{v} \cos \gamma \right] \\
& + \lambda_\sigma \left[\frac{1}{mv \cos \gamma} (L \sin \mu + T \sin \mu \sin \alpha) \right] \\
& + \lambda_T u_T + \lambda_\alpha u_\alpha + \lambda_\mu u_\mu
\end{aligned} \tag{3.17}$$

The Euler-Lagrange equation using this new Hamiltonian and following the form of (3.2) the Euler-Lagrange equation for this problem gives

$$\begin{aligned}
\frac{\partial H}{\partial u_T} &= \lambda_T = 0 \\
\frac{\partial H}{\partial u_\alpha} &= \lambda_\alpha = 0 \\
\frac{\partial H}{\partial u_\mu} &= \lambda_\mu = 0
\end{aligned} \tag{3.18}$$

Since the control variables do not appear explicitly in (3.18), the HMC is applied.

$$(HMC) \begin{cases} \text{Minimize} & H(\lambda, x, u) \\ \text{Subject to} & u^L \leq u \leq u^U \end{cases} \tag{3.19}$$

Thus (3.18) are interpreted as switching functions. For example,

$$\begin{aligned}
S_T = \lambda_T \rightarrow \lambda_T > 0 & \quad u_T = u_{T \min} \\
\lambda_T < 0 & \quad u_T = u_{T \max} \\
\lambda_T = 0 & \quad u_{T \min} \leq u_T \leq u_{T \max}
\end{aligned} \tag{3.20}$$

A similar interpretation can be made regarding the switching function for the other two control variables.

Assuming ρ is constant, the adjoint equation, including (3.10) through (3.13) is formulated as

$$\dot{\lambda}(t) = \frac{-\partial \bar{H}}{\partial \bar{x}}$$

$$\dot{\lambda}_x(t) = \frac{-\partial H}{\partial y} = 0$$

$$\dot{\lambda}_y(t) = \frac{-\partial H}{\partial x} = 0$$

$$\dot{\lambda}_z(t) = \frac{-\partial H}{\partial z} = 0$$

$$\dot{\lambda}_v(t) = \frac{-\partial H}{\partial v} = \lambda_x \cos \gamma \cos \sigma + \lambda_y \cos \gamma \sin \sigma + \lambda_z \sin \gamma$$

$$+ \lambda_v \left[\frac{1}{m} \rho v S C_D \right]$$

$$+ \lambda_\gamma \left[\frac{1}{m} \left(-\frac{1}{2} \rho S C_L \cos \mu + \frac{T}{v^2} \cos \mu \sin \alpha \right) - \frac{g}{v^2} \cos \gamma \right]$$

$$+ \lambda_\sigma \left[\frac{1}{m \cos \gamma} \left(-\frac{1}{2} \rho S C_L \sin \mu + \frac{T}{v^2} \sin \mu \sin \alpha \right) \right]$$

$$\dot{\lambda}_\gamma(t) = \frac{-\partial H}{\partial \gamma} = \lambda_x (v \sin \gamma \cos \sigma) + \lambda_y (v \sin \gamma \sin \sigma) - \lambda_z \left[v \cos \gamma + g \cos \gamma - \frac{g}{v} \sin \gamma \right]$$

$$+ \lambda_\sigma \left[\frac{1}{mv \sin \gamma} (L \sin \mu + T \sin \mu \sin \alpha) \right]$$

$$\dot{\lambda}_\sigma(t) = \frac{-\partial H}{\partial \sigma} = \lambda_x (v \cos \gamma \sin \sigma) - \lambda_y (v \cos \gamma \cos \sigma)$$

$$\dot{\lambda}_T(t) = \frac{-\partial H}{\partial T} = -\lambda_v \left[\frac{1}{m} \cos \alpha \right] - \lambda_\gamma \left[\frac{1}{mv} \cos \mu \sin \alpha \right] - \lambda_\sigma \left[\frac{1}{mv \cos \gamma} \sin \mu \sin \alpha \right]$$

$$\begin{aligned} \dot{\lambda}_\alpha(t) = \frac{-\partial H}{\partial \alpha} = & -\lambda_v \frac{1}{m} \left[-\frac{1}{2} \rho v^2 S (C_x \sin \alpha + C_z \cos \alpha) - T \sin \alpha \right] \\ & - \lambda_\gamma \frac{1}{mv} \left[\frac{1}{2} \rho v^2 S (-C_x \cos \alpha - C_z \sin \alpha) \cos \mu + T \cos \mu \cos \alpha \right] \\ & - \lambda_\sigma \frac{1}{mv \cos \gamma} \left[\frac{1}{2} \rho v^2 S (-C_x \cos \alpha - C_z \sin \alpha) \cos \mu + T \sin \mu \cos \alpha \right] \end{aligned}$$

$$\dot{\lambda}_\mu(t) = \frac{-\partial H}{\partial \mu} = -\lambda_\gamma \left[\frac{1}{mv} (L \sin \mu + T \sin \mu \sin \alpha) \right] - \lambda_\sigma \left[\frac{1}{mv \cos \gamma} (L \cos \mu + T \cos \mu \sin \alpha) \right] \quad (3.21)$$

From (3.21) it can be seen costates for x , y , and z must be constant, and that no information the remaining costates can be predicted. These results can be used to test the optimality of the optimal control solution. Analysis of the transversality condition provides no additional information that can be used to verify optimality of a numerical solution to the OCP.

g. Reference Frame

The optimization model utilizes a flat-Earth reference frame, with position described in x - y - z Cartesian coordinates with respect to an arbitrary point on the flat Earth. Downrange distance is measured in the x -direction, cross-range distance in the y -direction and altitude is measured up in the z -direction (see Figure 12).

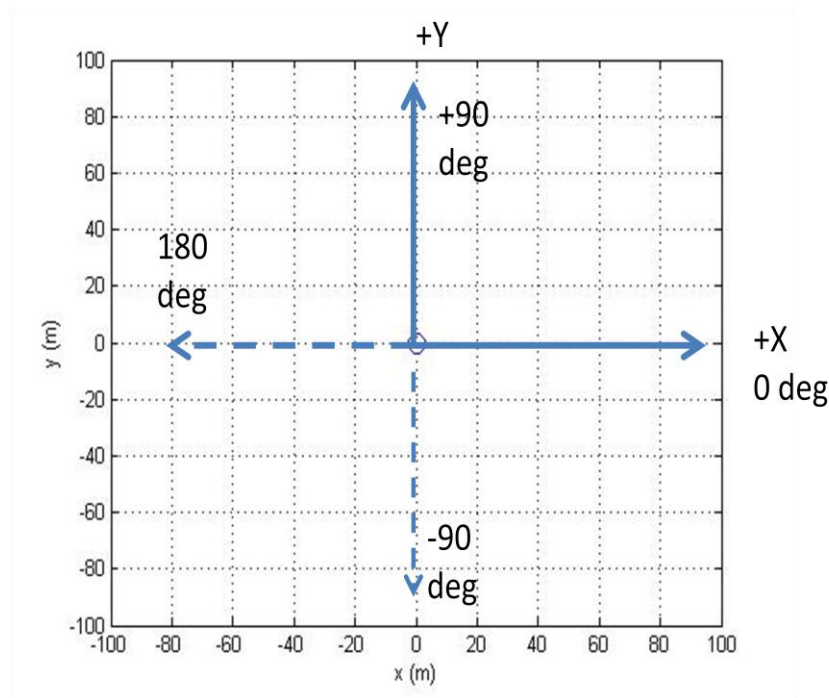


Figure 12. Reference Frame for 3DOF Maneuvers

D. INTRODUCTION TO OPTIMAL CONTROL WITH MATLAB AND DIDO

DIDO is a MATLAB application developed in 1998 as a tool for solving complex optimal control problems [35]. The user must formulate the problem, within a specific format, and DIDO uses a unique pseudospectral optimal control theory approach to find candidate optimal solutions [35]. It is not a direct method, but requires verification and validation by the user to ensure the optimality of the solution found. A more thorough discussion of the mathematics and theory behind DIDO is beyond the scope of this work, but interested readers can find the details in [8] and the references therein.

1. MATLAB Optimization Code

DIDO requires the user to state the problem to be optimized in a specific format. Some of the user-supplied m-files are mandatory for every problem, and some are optional. The mandatory files are the cost function, event function, and dynamics function. The optional file is the path function. DIDO allows for setup of the optimization problem in the form of (3.1) or (3.8) that can be readily adapted or changed as needed.

2. State Variable Constraints

For each state variable, box constraints must be placed. These must be large enough to allow a feasible solution to be found within the limits, but restrictive enough that DIDO will not have to search an unreasonable range for a solution. The state variable limits are part of the main problem script, as the state bounds.

In the case of the time-optimized trajectories, the x- and y-limits are based on leaving enough room for a maneuver to be completed, and the z-limit ranges from sea level to the service ceiling of the aircraft. The velocity limits are set by the stall speed and maximum dive speed of the aircraft, and the heading range is $-\pi$ to π , or 360 degrees. The thrust limit is based on the engine capability, and the maximum and minimum pitch and bank angle limits are based off standard limits for small RPV flight. Representative limits are listed in Table 11.

3. Initial and Final Conditions

The model is provided with initial and final conditions for all states or a subset of states. These may be set at a certain value or within a range of values.

symbol	units	Lower Limit	Upper Limit	Notes
x	m	-1,000	1,000	representative example of box limits
y	m	-1,000	1,000	representative example of box limits
z	m	0	1,000	estimated service ceiling
v	m/s	13	42	stall speed and max dive speed
γ	rad	$-\pi/6$	$\pi/6$	best engineering estimate based on SIMULINK model
σ	rad	$-\pi$	π	full circle
T	N	3	35	from SIMULINK model
α	rad	$-\pi/12$	$-\pi/12$	best engineering estimate based on SIMULINK model
μ	deg	-25	25	constraint from safety pilot
u_T	N/ s	-1	1	best engineering estimate based on SIMULINK model
u_α	rad/ s	-0.05	0.05	best engineering estimate based on SIMULINK model
u_μ	rad/ s	-0.05	0.05	best engineering estimate based on SIMULINK model

Table 11. Representative Limits for States and Controls

4. Control Rate Limits

The limits on controls are set just as the state variable constraints are. For the minimum-time problem for the optimization model, the control limits are set by the time constants of the inner loops for thrust rate and turn rate, and derived from the maximum load factor for the AoA rate. These are coded in the problem formulation as the control bounds and shown in Table 11.

5. Events Function and Limits

The events function is a separate file that describes the equations of the endpoints of the OCP. The event bounds can restrict all or none of the states at the endpoints to a set value or equation, or to a range of values, depending on what the desired endpoint

conditions are. Similarly, if used, the path function may restrict any or all of the states or controls to a value or range of values for the trajectory.

6. Time and Node Number Selection

A starting and ending time range is also required for DIDO. For the optimization model, all maneuvers used a starting time of zero, and a maximum final time of 100 seconds, which was later reduced as much as possible for maneuvers that took significantly less than 100 seconds, to reduce run time.

Node numbers are set low for initial DIDO solution runs, and then increased once the code is debugged and extremal solutions are being generated. Normally it is desirable to increase the node number to improve accuracy as long as this does not make the computation time prohibitively long.

7. Maneuver Results, Validation and Verification

The results generated by DIDO for the OCP include the states, costates, controls, Hamiltonian value, and cost. The controls and states make up the trajectory information that can be exported and used for the aircraft to fly. The cost is the time at completion of the maneuver. For the minimum time problem the numerical value of the Hamiltonian should be very close to negative one [34].

To verify that the solution in each case was both feasible and optimal, the solution is verified via propagation and the costates and Hamiltonian checked against the necessary conditions per (3.18) through (3.21). The aircraft equations of motion were propagated with the controls via an ODE solver in MATLAB. Depending on the maneuver, the *interp1* or *pchip* MATLAB interpolation functions were used to interpolate the optimal control history. Different interpolation functions provided the best interpolation fit and propagated solution, so several interpolation functions were tested for each maneuver to provide the best propagation possible.

The propagated solution is compared with the states from the DIDO solution to determine if the propagated solution is within an acceptable allowance, showing that the solution is feasible. The solution shown in Figure 13 illustrates a

successful feasibility propagation, in which the propagated solution converges with the DIDO-generated solution, as well as an unsuccessful feasibility propagation for comparison.

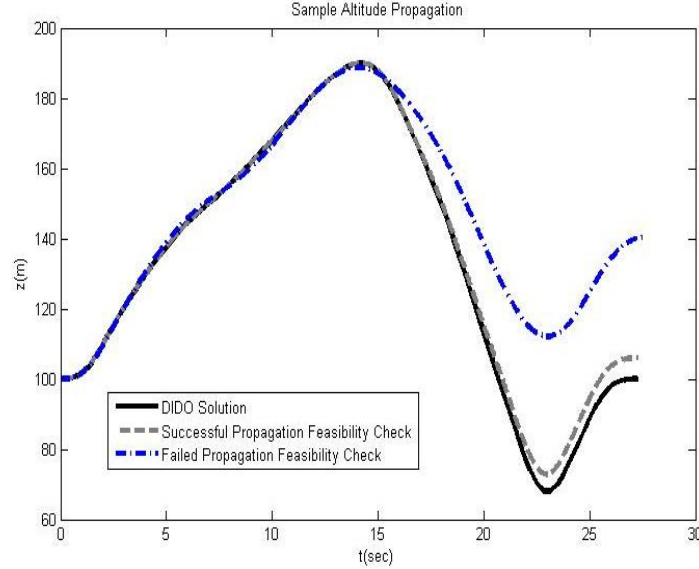


Figure 13. Successful Feasibility Test via Propagation

8. Improvements

a. Number of Nodes

Increasing the number of nodes can increase the fidelity of the results. This can unfortunately also greatly increase the computation time required for DIDO to find a solution. One way to accomplish a node increase more efficiently is to “bootstrap,” or to feed the solution from a lower node count run into DIDO as the “guess” solution for a higher node count run. This can be repeated several times in a row. For example, for most solutions in this work, a 12-node solution was found first. The 12-node solution was used as the initial guess for the 24-node run, and that result was used for the 36-node calculation, and so on.

b. Scaling

Using designer units to scale the problem also reduces the run time and improves the DIDO results [35]. Scaling, so as to make the state variables, control

variables and costates within at least an order of magnitude of each other, or preferably close to one, improves the performance.

E. TESTING THE OPTIMAL CONTROL CODE

The DIDO optimization model combines the dynamics equations, states, and controls from the reduced-order 3DOF model and the physical characteristics of the MONARC aircraft with the time-optimization problem format and basic DIDO codes to form a model for optimizing MONARC trajectories. It is a flexible model that can be iteratively upgraded and as flight data is recorded, and can be optimized for a variety of costs while being used for maneuvers ranging from simple straight-and-level flight to complex patterns.

1. Code Verification

To verify and troubleshoot the dynamics equations, trajectory optimizations were performed using boundary conditions and control limits for a known aircraft configuration, as given in [30]. This allowed for troubleshooting to ensure that early code issues were not being caused by incorrect estimations of the MONARC aircraft's control or state limits.

2. Trim Maneuvers and Additional Model Verification

The first maneuvers obtained using the optimal control formulation were trim maneuvers, including straight-and-level flight, a steady turn, a steady climb, and a turn while climbing. While these are not maneuvers *per se*, they allowed for further troubleshooting, and allowed for implementation of the trimmed flight conditions at the start and end of future maneuvers, simulating a return to straight-and-level that can be executed by the autopilot between optimized maneuvers. In order to verify that the model can be used for other optimization problems, OCPs were also generated and solved for maximizing altitude increase within an area, and minimizing distance to climb, and compared with a problem of minimizing time to climb.

IV. CANONICAL MANEUVERS

A. FORMULATION OF FLIGHT TEST MANEUVERS

Current microcontroller technology has not yet reached the level of computational capacity and speed where real-time optimization is feasible for an onboard autopilot on a small aircraft such as the MONARC. This does not prohibit optimization for autonomous vehicles, but rather suggests the use of an onboard database of already-optimized maneuvers that the autopilot may select from and fit together to perform a commanded task. A series of time-optimized canonical maneuvers was selected and optimized in order to begin such a database as well as to test the model. The maneuvers were chosen to demonstrate a variety of different flight conditions and state changes while highlighting the features of optimal solutions for maneuvers that are commonly performed by aircraft on a broad range of missions. Four typical maneuvers of varied complexity show how optimal trajectories can be generated for a variety of situations.

1. Conditions Common to Each of the Canonical Maneuvers

Each of the canonical maneuvers begins and ends with the aircraft in trimmed, straight-and-level flight, where the AoA and Thrust values are calculated for this trim, and the bank angle rate of change is zero. The aircraft begins each canonical maneuver at the median velocity, and the ending velocity is left free, which allows the maneuver to end back at the straight and level Thrust and AoA level trim values.

The initial and final conditions, state, time, control, and event bounds applied to all of the canonical maneuvers are listed along with each maneuver. Some bounds will vary, such as those on x , y , and z , depending on how much room is allotted to complete the maneuver. Other bounds, such as limits on controls, thrust, and velocity, do not change between canonical maneuvers, as they are based on physical limitations of the aircraft.

A trim function is used to calculate Thrust and AoA for each maneuver in order to ensure that the change in velocity and pitch is equal to zero at the start and end of each maneuver. This allows the aircraft to initiate and terminate each the maneuver from level

flight. Each canonical maneuver optimization is developed by bootstrapping up to 64 nodes. All canonical maneuver result plots are displayed with the solution nodes, each displayed as an 'o' shape, as well as the propagated solution, plotted as a solid line.

2. Diagonal Transfer Flight

a. Description of Maneuver

This canonical maneuver begins and ends on a heading of zero degrees, or directly downrange in the x-direction. It is desired to move the aircraft to a point that is 1000 meters away in each direction—x, y, and z—from the zero starting point. The maneuver is complete when the aircraft is back at straight-and-level flight on the initial heading. The arrows in Figures 14 and 15 indicate the starting point and direction, and the chevrons indicate the ending point and direction. This starting and ending notation is used in the descriptive figures for each maneuver.

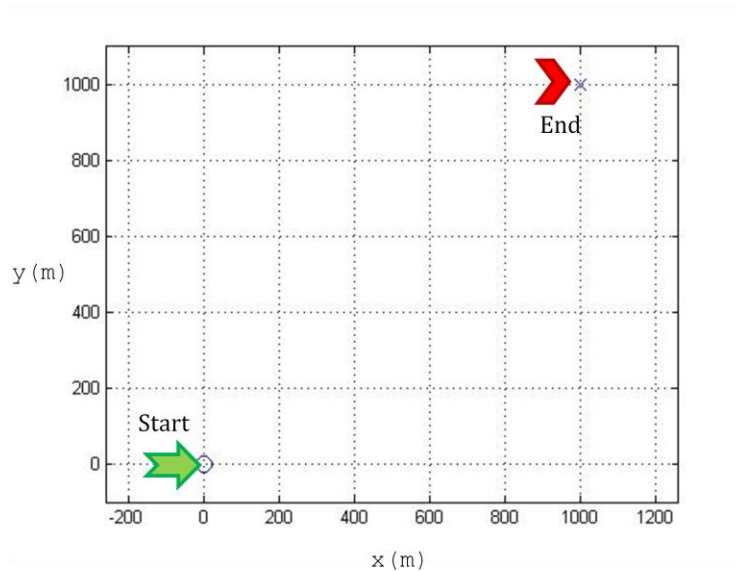


Figure 14. Overhead view of Diagonal Transfer Maneuver

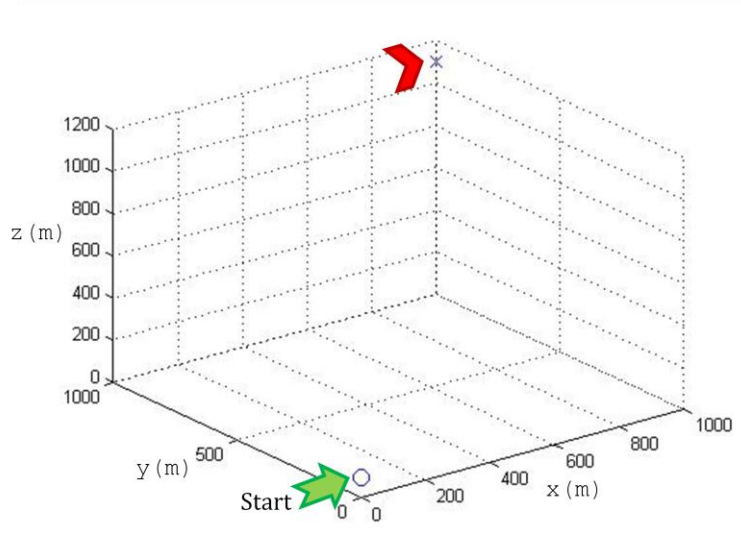


Figure 15. Diagonal Maneuver 3-D View

This diagonal transfer maneuver was selected as the first canonical maneuver to demonstrate in flight because it would show movement in all three directions, accomplishing a diagonal translation, and would also require more than one turn in order to begin and end on the same heading.

b. Initial and Final Conditions

The initial and final conditions for the maneuver describe the starting and ending points for the aircraft's maneuver, as shown in Table 12.

Initial Conditions:

State	Value	units
x_0	0	m
y_0	0	m
z_0	100	m
v_0	27.5	m/s
γ_0	0	rad
σ_0	0	rad
T_0	16.1	N
α_0	-0.0088	rad
μ_0	0	rad

Final Conditions:

State	Value	units
x_f	1000	m
y_f	1000	m
z_f	1100	m
v_f	27.5	m/s
γ_f	0	rad
σ_f	0	rad
T_f	16.1	N
α_f	-0.0088	rad
μ_f	0	Rad

Table 12. Initial and Final Conditions for Diagonal Maneuver

c. Box Constraints

These box constraints describe the limits on the states and controls throughout the maneuver. The x , y , and z bounds shown in Table 13 will be different for some of the canonical maneuvers, but the control bounds and the bounds on all other states are the same for all maneuvers.

Lower State Bounds:

	Value	units
x_L	-2000	m
y_L	-2000	m
z_L	0	m
v_L	13	m/s
γ_L	$-\pi/6$	rad
σ_L	$-\pi$	rad
T_L	3	N
α_L	$-\pi/12$	rad
μ_L	-25	deg

Upper State Bounds:

	Value	units
x_U	2000	m
y_U	2000	m
z_U	1500	m
v_U	42	m/s
γ_U	$\pi/6$	rad
σ_U	π	rad
T_U	35	N
α_U	$\pi/12$	rad
μ_U	25	deg

Table 13. State Bounds for Diagonal Maneuver

The time limit was left long during early test runs, and then reduced to some value comfortably past the end of the maneuver but low enough to not cause DIDO to search for solutions in a very long time window.

The event bounds in Table 14 describe the endpoint conditions for the maneuver. They can constrain each state's beginning and endpoint to either a range of values or to a single value. In this case, the start and end points were constrained to the initial and final condition values, with the exception of the ending velocity, which was left free to be any number within the allowable state bounds.

Event Bounds					
	<i>Lower</i>	<i>Upper</i>		<i>Lower</i>	<i>Upper</i>
start	x_0	x_0	end	x_f	x_f
	y_0	y_0		y_f	y_f
	z_0	z_0		z_f	z_f
	v_0	v_0		v_L	v_U
	γ_0	γ_0		γ_f	γ_f
	σ_0	σ_0		σ_f	σ_f
	T_0	T_0		T_f	T_f
	α_0	α_0		α_f	α_f
	μ_0	μ_0		μ_f	μ_f

Table 14. Event Bounds for Diagonal Maneuver

The control bounds in Table 15 remain the same for all canonical maneuvers conducted. They are estimated from the SIMULINK 6DOF model and will be revised once actual flight tests are conducted to determine the actual safe flight parameters of the MONARC aircraft.

Lower Control Bounds:			Upper Control Bounds:		
	Value	units		Value	units
u_{TL}	-1	N/ s	u_{TU}	1	N/ s
$u_{\alpha L}$	-0.05	rad/ s	$u_{\alpha U}$	0.05	rad/ s
$u_{\mu L}$	-0.05	rad/ s	$u_{\mu U}$	0.05	rad/ s

Table 15. Control Bounds for All Canonical Maneuvers

d. Results

The x-y position view in Figure 16 shows the path taken to complete the move as seen from above, or approximately what the course over ground would look like.

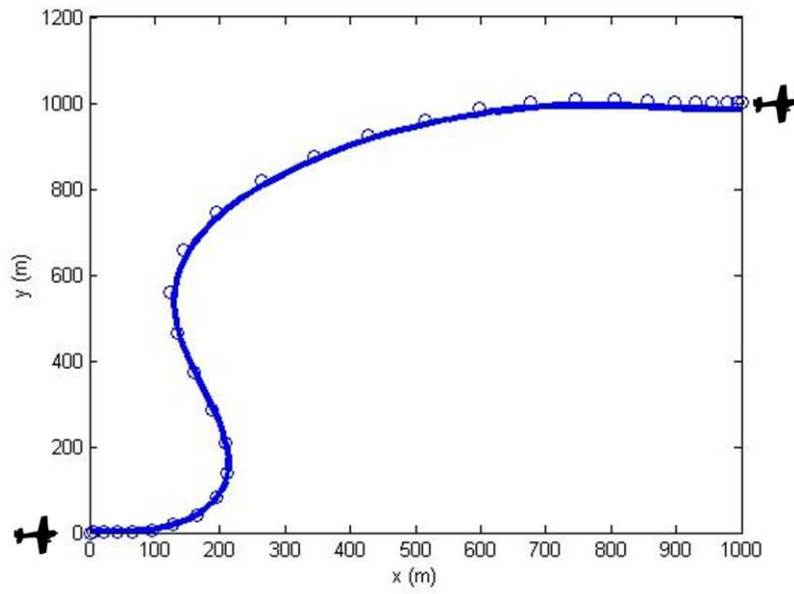


Figure 16. Overhead View of DIDO Diagonal Transfer Trajectory

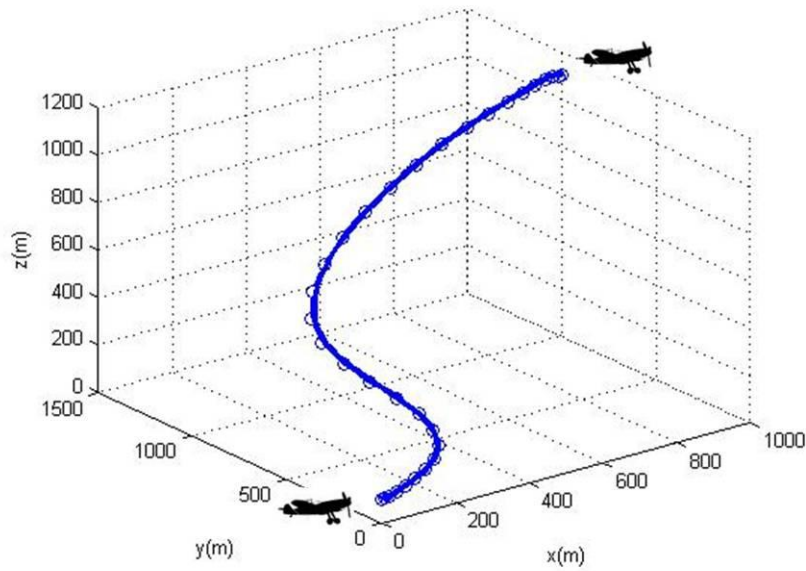


Figure 17. Diagonal Trajectory 3-D View

The three-dimensional view of the maneuver (Figure 17) shows the path taken through 3-D space to arrive at the desired endpoint. A traditional autopilot or human pilot would more likely fly a direct line to the point, then turn and straighten at the last

moment. Because the maneuver is required to begin and end at straight-and-level flight on the original heading, a traditional straight-line maneuver would consist of an immediate turn to a course pointed towards the endpoint. Shortly before reaching the endpoint, the aircraft would need to make large and rapid changes to heading, pitch, and velocity to straighten and steady back on the original velocity. This would most likely result in some overshoot and a failure to exactly intercept the point. The time-optimal result is a smoother, more S-shaped path achieves completion of the maneuver in about 65 seconds. Because of the S-turn the aircraft is able to line up with the endpoint early and approach the final position with the correct heading.

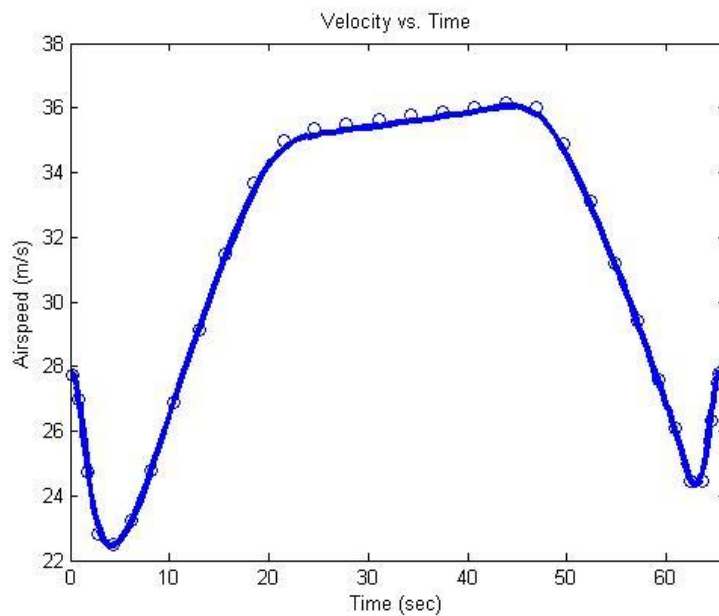


Figure 18. Velocity During the Diagonal Transfer Maneuver

The velocity during the maneuver corresponds with the aircraft slowing to perform the tighter turns (see Figure 18). The optimized climb from 100m to 1000m shown in Figure 19 is quite smooth and direct, as compared with the movement in the lateral and forward directions, which utilizes the wide S-turn.

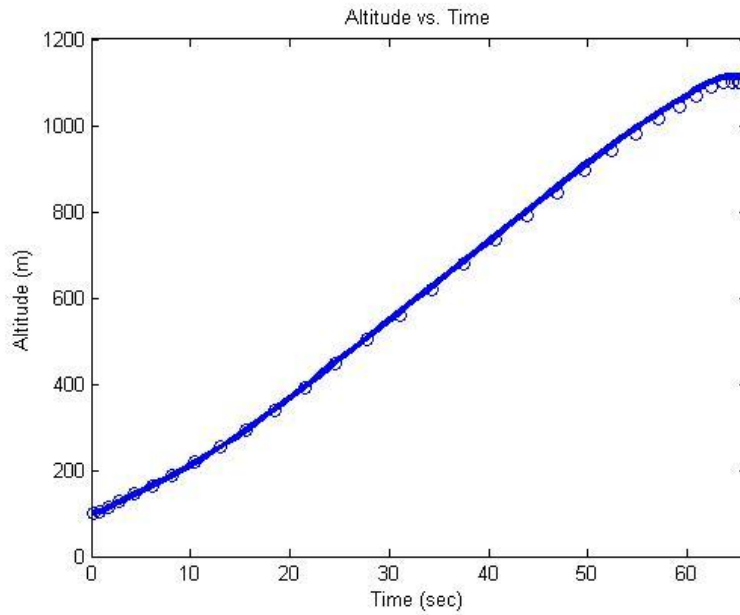


Figure 19. Altitude Profile for Diagonal Maneuver

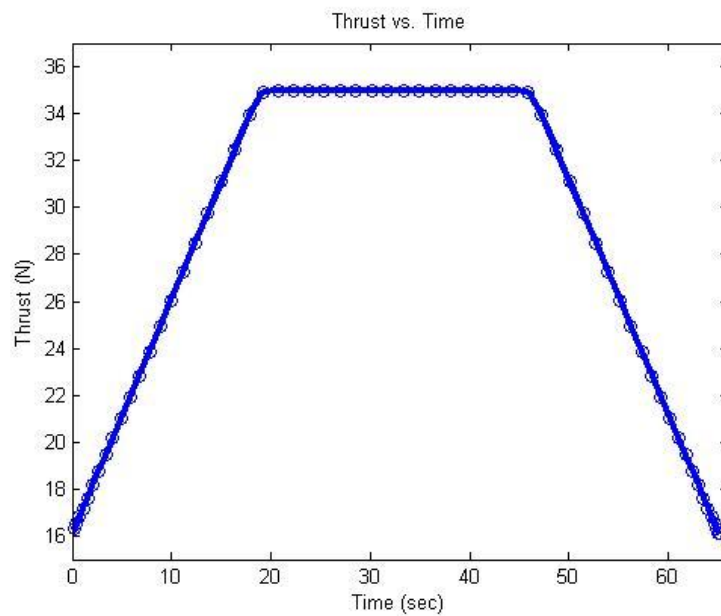


Figure 20. Thrust Profile for Diagonal Maneuver

To achieve the smooth, straight climb as shown in the altitude plot, the maximum thrust available is used for a significant portion of the maneuver, as shown in Figure 20. This highlights that while the maneuver may be time-optimized, it may be

very costly in terms of fuel usage, and it is important to choose carefully which costs are most valuable to minimize for a given problem. However, without comparison with the thrust profile used by a real pilot or other autopilot system, it is hard to judge whether or not there is a significant increase in fuel usage. This aspect will be investigated as part of the planned flight testing.

The maneuver utilizes the full bank angle range available, reaching both the upper and lower limits on bank angle, but minimal angle of attack changes (see Figures 21 and 22). The unique turn shape is highlighted by the heading angle plot (Figure 23) and the flight path angle reaches maximum for the majority of the maneuver.

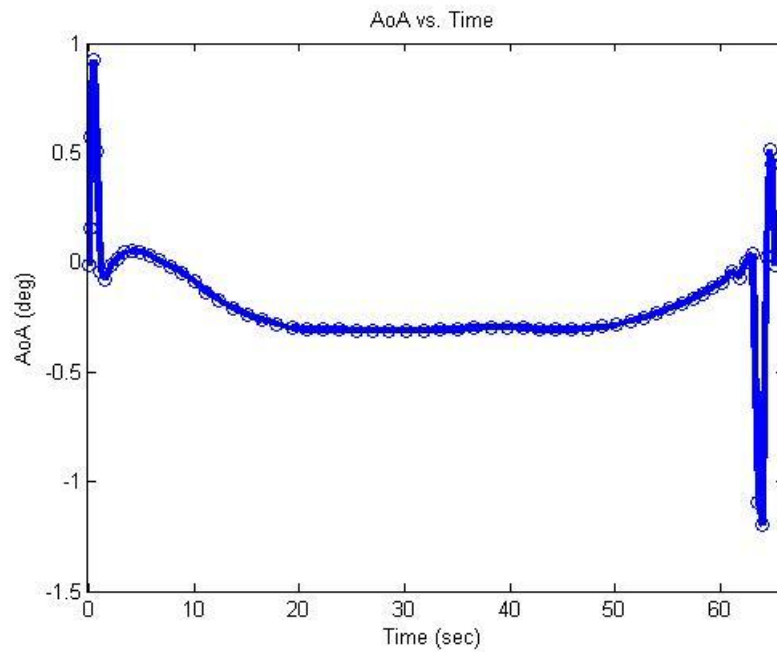


Figure 21. AoA vs. Time for Diagonal Maneuver

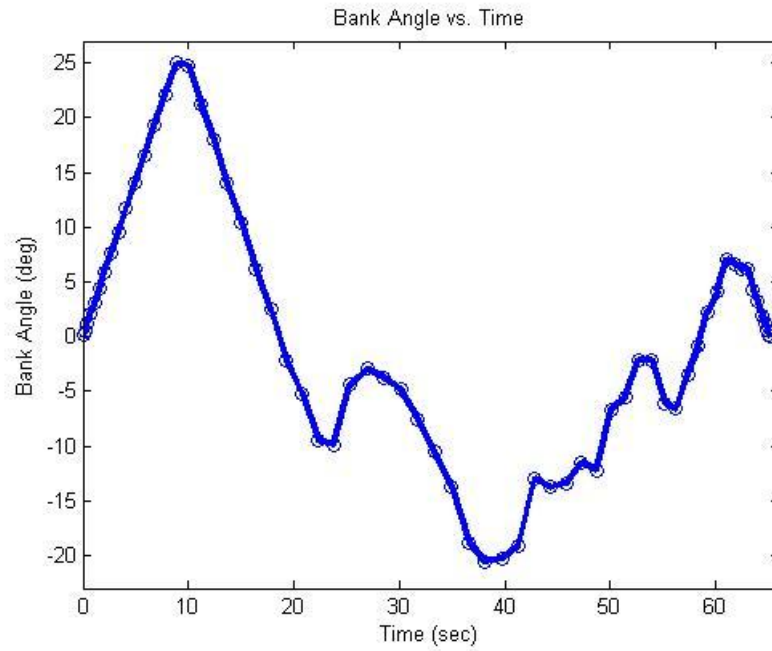


Figure 22. Bank Angle vs. Time for Diagonal Maneuver

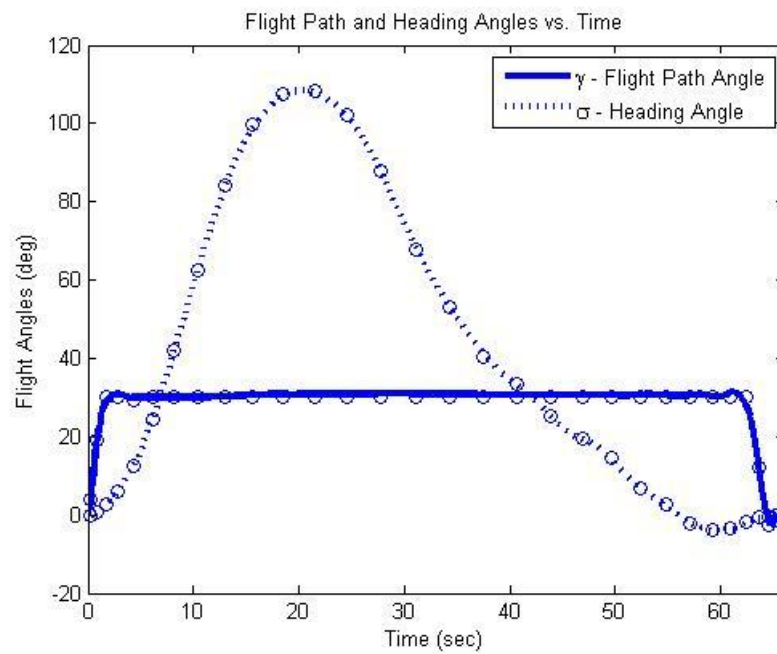


Figure 23. Flight Path and Heading Angles for Diagonal Maneuver

To demonstrate the optimality of the maneuver, the costates shown in Figure 24 are approximately zero for the x, y, and z states, as expected from (3.20) and the corresponding discussion in Chapter III. Figure 25 shows the control and costate comparison for the thrust variable. The curves behave as expected based on the HMC as discussed in Chapter III.

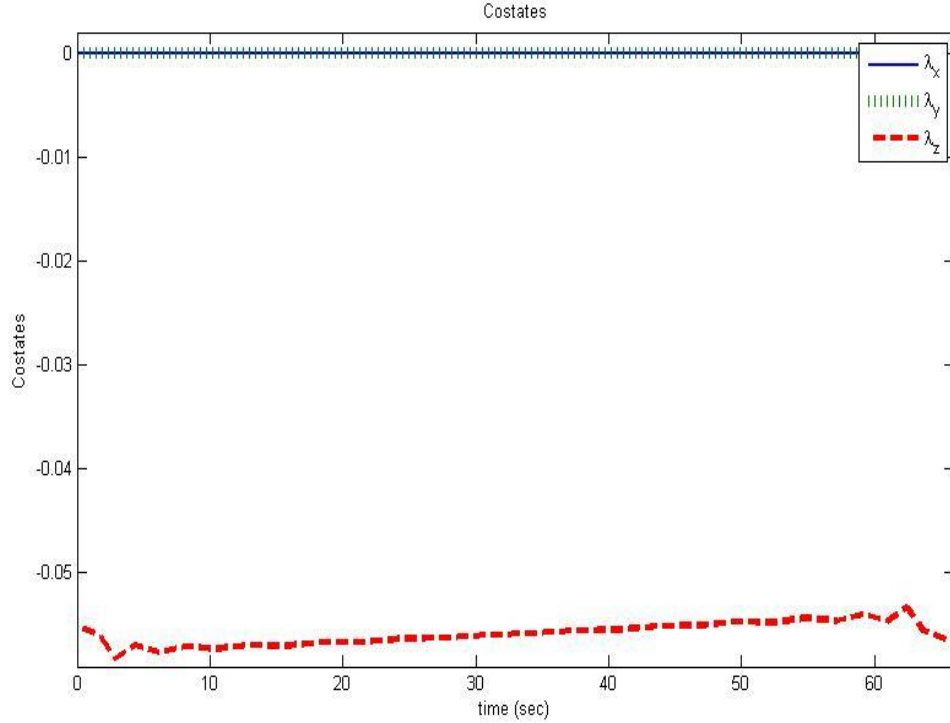


Figure 24. Position Costates vs. Time for Diagonal Maneuver

The Hamiltonian value is very close to -1 for most of the duration of the maneuver, as displayed in Figure 26. This indicates that the maneuver is very close to optimal, as discussed in Chapter III regarding (3.6). The slight deviation from -1, at the two end-points, indicates that the solution could benefit from increasing the number of nodes. However, from an implementation point of view this additional step is not necessary.

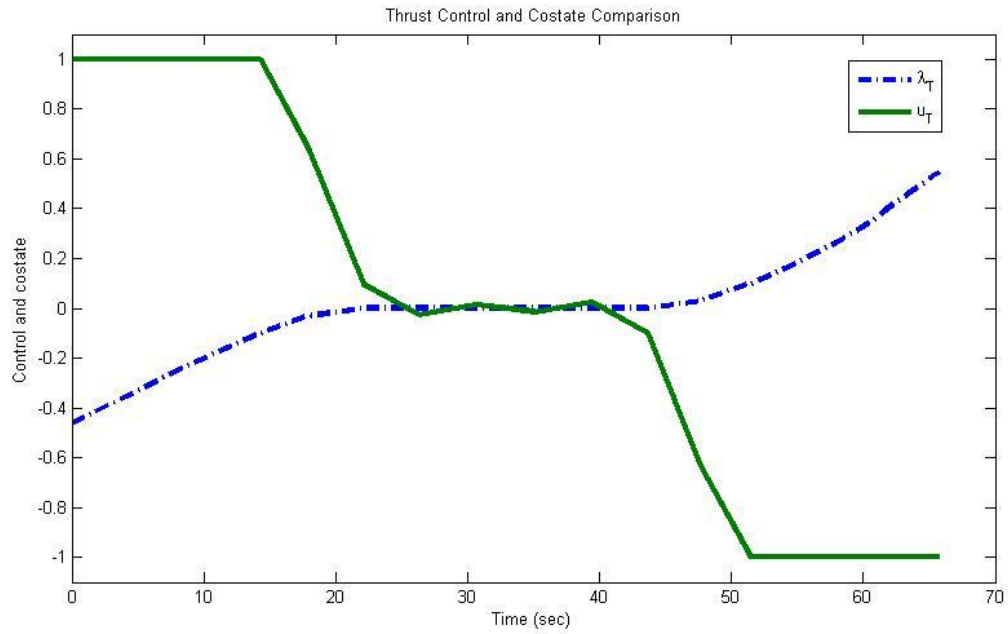


Figure 25. Control and Costate for Thrust Variable

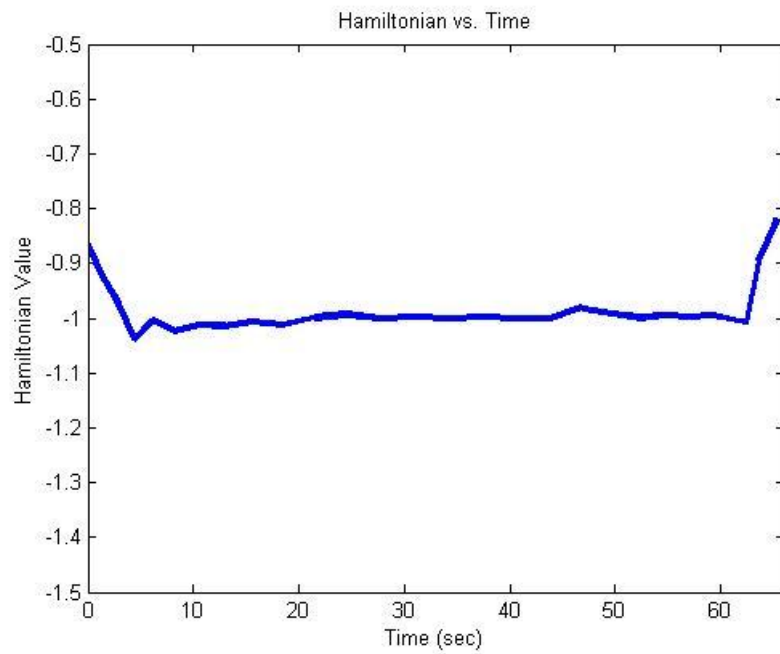


Figure 26. Hamiltonian vs. Time for Diagonal Maneuver

3. Wide U-Turn

a. Description of Maneuver

In this canonical maneuver the aircraft performs a turn in order to finish at a point 1000 meters laterally translated from the starting point, and on the opposite heading. This is a wide U-turn maneuver (see Figure 27), and was chosen as a demonstration maneuver because it is the type of turn commonly flown by aircraft that are performing clearing turns, conducting searches, circling for air control or radar support purposes, or waiting to enter an airport traffic pattern.

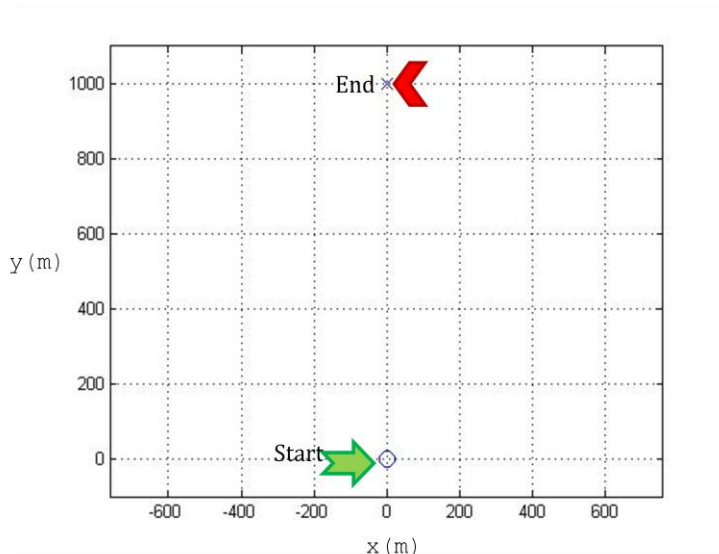


Figure 27. Top View of U-turn Maneuver

b. Initial and Final Conditions

The x , y , z and heading initial and final conditions are changed in this maneuver to describe the desired starting and finishing points, as shown in Table 16.

Initial Conditions:			Final Conditions:		
State	Value	units	State	Value	units
x_0	0	m	x_f	0	m
y_0	0	m	y_f	1000	m
z_0	1000	m	z_f	1000	m
v_0	27.5	m/s	v_f	27.5	m/s
γ_0	0	rad	γ_f	0	rad
σ_0	0	rad	σ_f	π	rad
T_0	16.1	N	T_f	16.1	N
α_0	-0.0088	rad	α_f	-0.0088	rad
μ_0	0	rad	μ_f	0	rad

Table 16. Initial and Final Conditions for U-turn Maneuver

c. Box constraints

In this maneuver, the x, y, and z bounds are varied slightly (see Table 17) to allow more room for the aircraft to conduct the maneuver.

Lower State Bounds:			Upper State Bounds:		
	Value	units		Value	units
x_L	-5000	m	x_U	5000	m
y_L	-5000	m	y_U	5000	m
z_L	0	m	z_U	3000	m
v_L	13	m/s	v_U	42	m/s
γ_L	$-\pi/6$	rad	γ_U	$\pi/6$	rad
σ_L	$-\pi$	rad	σ_U	π	rad
T_L	3	N	T_U	35	N
α_L	$-\pi/12$	rad	α_U	$\pi/12$	rad
μ_L	-25	deg	μ_U	25	deg

Table 17. State Bounds for U-turn Maneuver

d. Results

The optimal trajectory solution found by DIDO within the above conditions is a maneuver that took about 41 seconds to complete. The trajectory as viewed from above

in Figure 28 is presented next to a perfect semi-circle at constant altitude, to show the trajectory that a typical pilot or autopilot might use. This unique trajectory shown in Figures 28 and 29 is not what a typical pilot or autopilot would choose as a path to turn around—changes in altitude are not generally considered as part of the maneuver when making a simple U-turn. An aircraft can make a tighter turn while flying slowly, such as it does during a climb, and can gain speed during a descent to take full advantage of the aircraft’s performance capabilities.

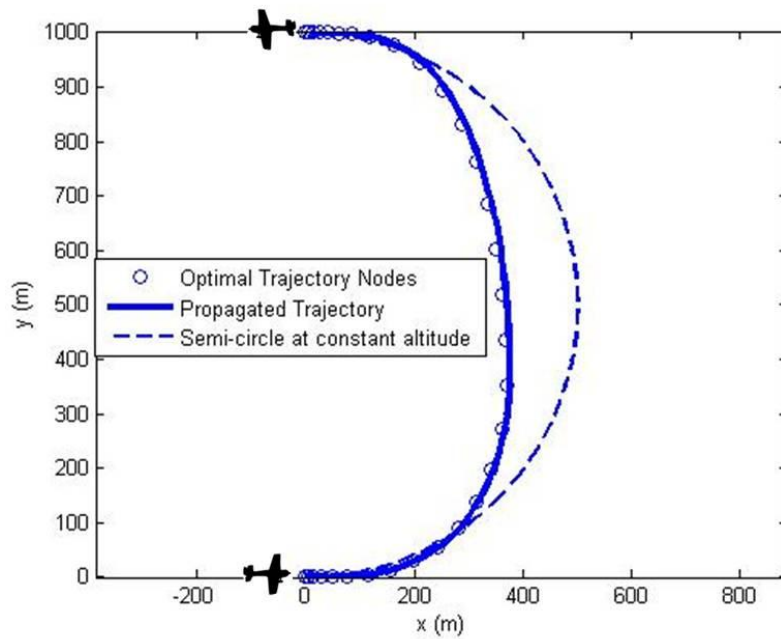


Figure 28. Overhead View of Optimized U-turn Trajectory vs. a Semi-Circular Trajectory

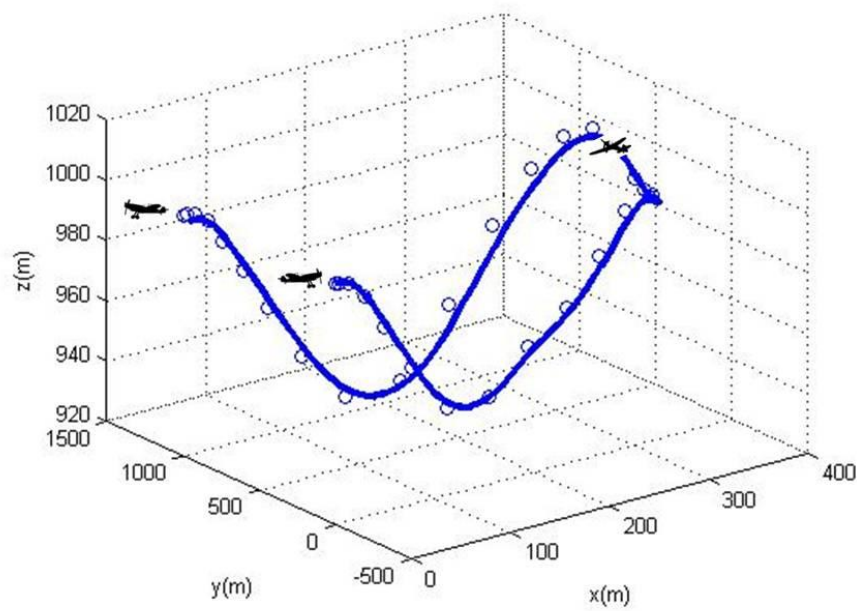


Figure 29. Wide U-turn Trajectory 3-D View

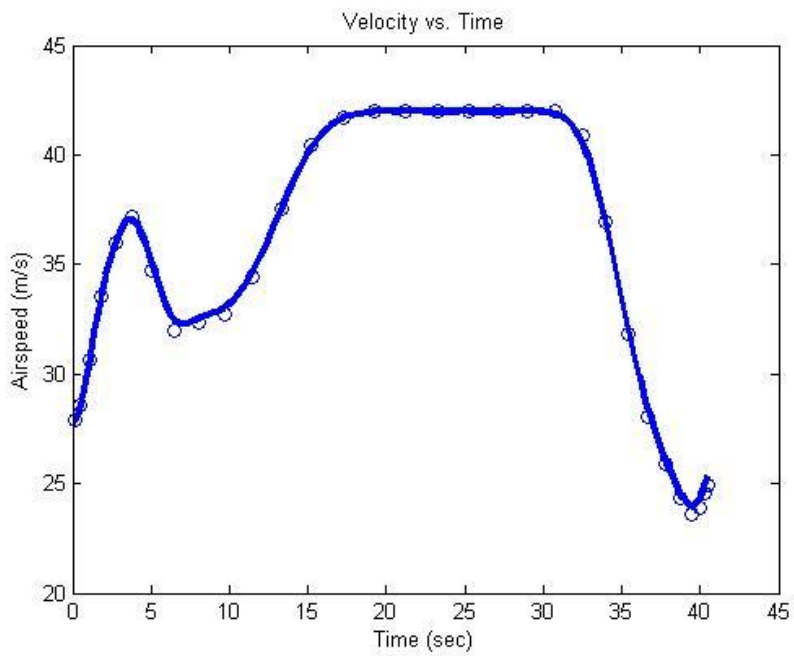


Figure 30. Velocity During the U-turn Maneuver

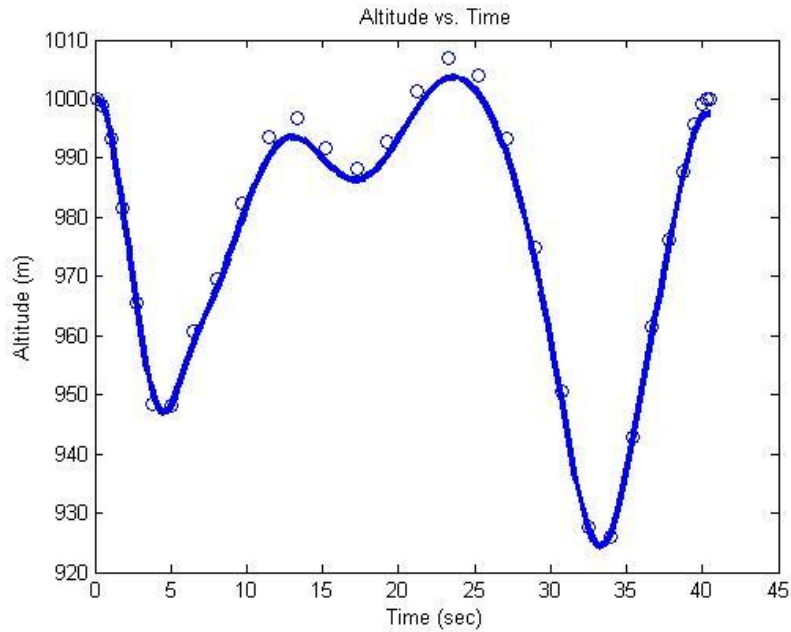


Figure 31. Altitude Profile for U-turn Maneuver

The maximum velocity is reached and sustained during a significant portion of the maneuver (Figure 30). This corresponds to the altitude changes (Figure 31), as the aircraft can be seen to “dive” to increase velocity.

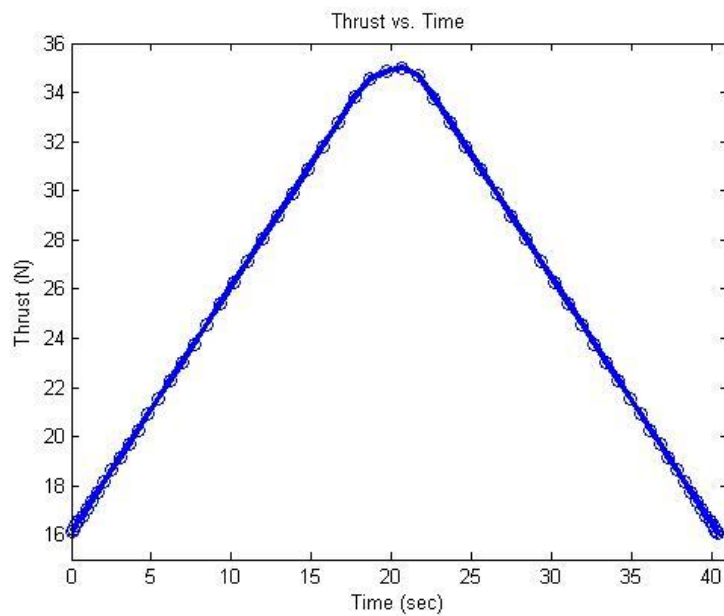


Figure 32. Thrust Profile for U-turn Maneuver

The thrust profile as seen in Figure 32 also highlights the “dive” performed by the aircraft. The maximum thrust is only briefly reached in the middle of the maneuver.

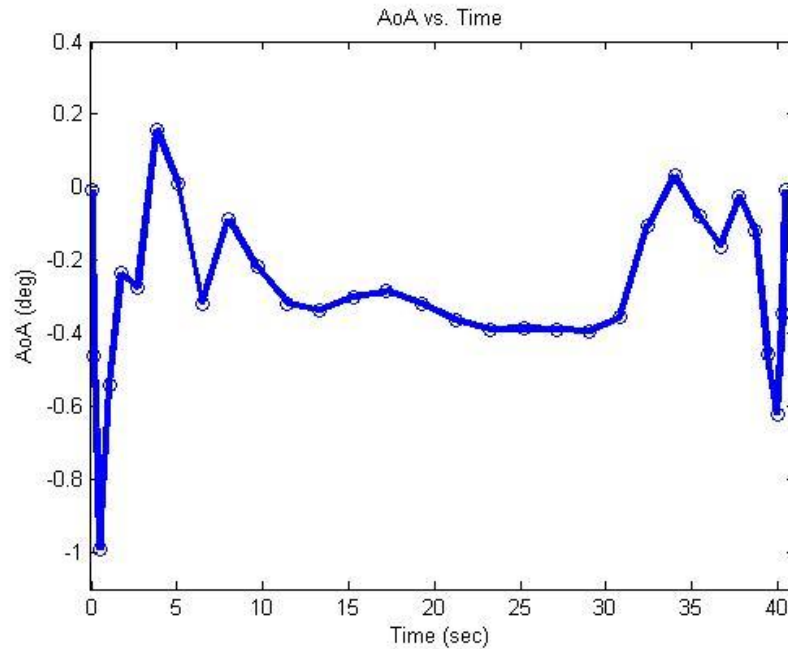


Figure 33. AoA for U-turn Maneuver

Figure 33 shows the AoA over time for the U-turn maneuver, which only changes only over a small range, but does so rapidly.

The maximum allowable bank angle of 25 degrees is reached twice during the maneuver (see Figure 34) and the maximum flight path angle is reached once (see Figure 35). That the aircraft reaches limits for bank angle, velocity, flight path angle and thrust during the maneuver shows that the optimal solution utilizes the full capability of the aircraft.

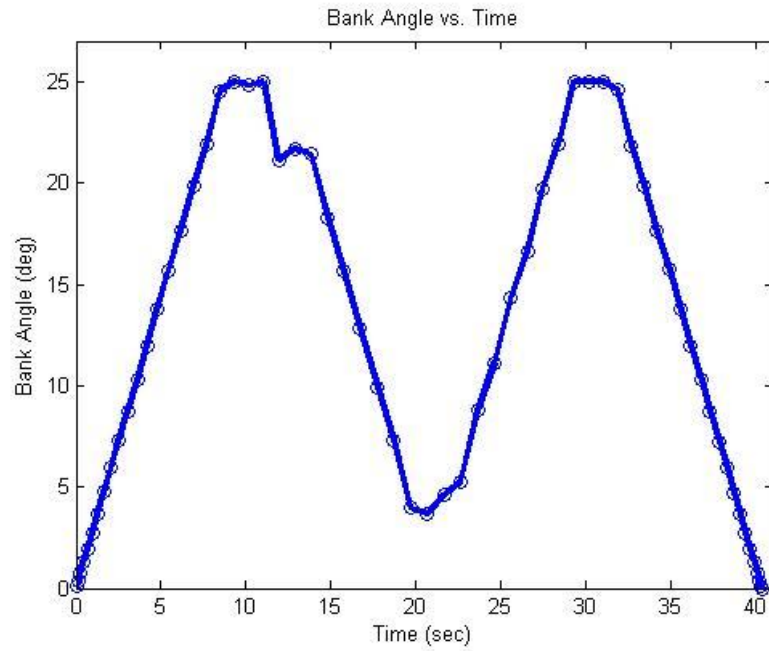


Figure 34. Bank Angle for U-Turn Maneuver

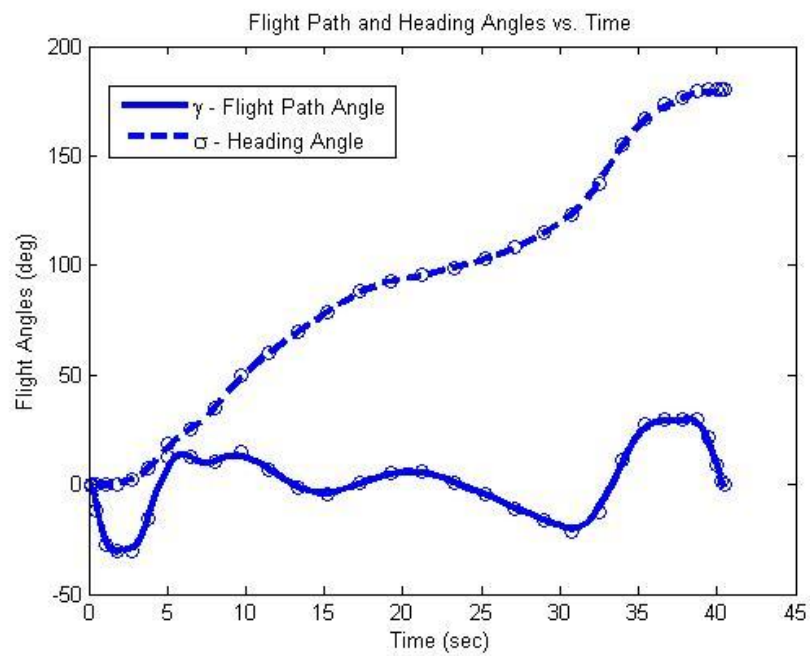


Figure 35. Flight Path and Heading Angles for U-turn Maneuver

4. “Scoot Over”

a. *Description of Maneuver*

This is a similar maneuver to the U-turn in that the aircraft must translate 1000 meters laterally. The difference is that the aircraft must finish on the initial heading, as shown in Figure 36. The initial conditions are the same as for the U-turn, and the only difference in final conditions is the heading. This maneuver was chosen to see what optimal solutions might be found to accomplish something more complex, and it simulates the type of approach that might be used for a strafing run while conducting close air support operations, or when trying to obtain several images of an area from the same angle. This maneuver is useful in situations where the aircraft must come at a nearby area from the same heading. A typical pilot or autopilot might instead use a wide circling maneuver to accomplish the same objective.

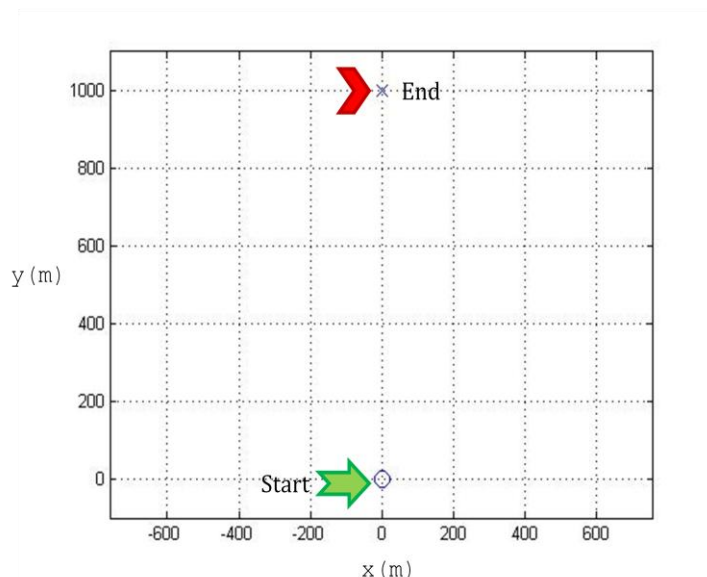


Figure 36. Top View of “Scoot Over”

b. Initial and Final Conditions

The initial and final conditions for this maneuver (Table 18) are based on the desired start and end points as well as the straight-and-level trim conditions. The boundary conditions and event bounds for this problem are the same as for the previous maneuver, the U-turn.

Initial Conditions:			Final Conditions:		
State	Value	units	State	Value	units
x_0	0	m	x_f	0	m
y_0	0	m	y_f	1000	m
z_0	1000	m	z_f	1000	m
v_0	27.5	m/s	v_f	27.5	m/s
γ_0	0	rad	γ_f	0	rad
σ_0	0	rad	σ_f	0	rad
T_0	16.1	N	T_f	16.1	N
α_0	-0.0088	rad	α_f	-0.0088	rad
μ_0	0	rad	μ_f	0	rad

Table 18. Initial and Final Conditions for “Scoot Over” Maneuver

c. Results

This optimal trajectory utilizes a ‘climb and dive’ type maneuver similar to that of the U-turn in terms of altitude maneuvering, while executing a wide S-turn as shown in Figures 37 and 38.

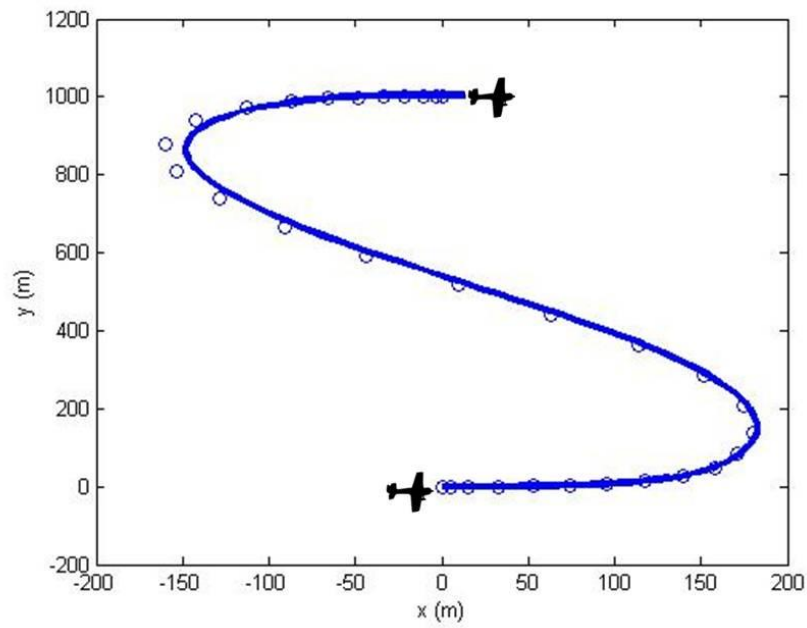


Figure 37. Overhead View of “Scoot Over” Trajectory

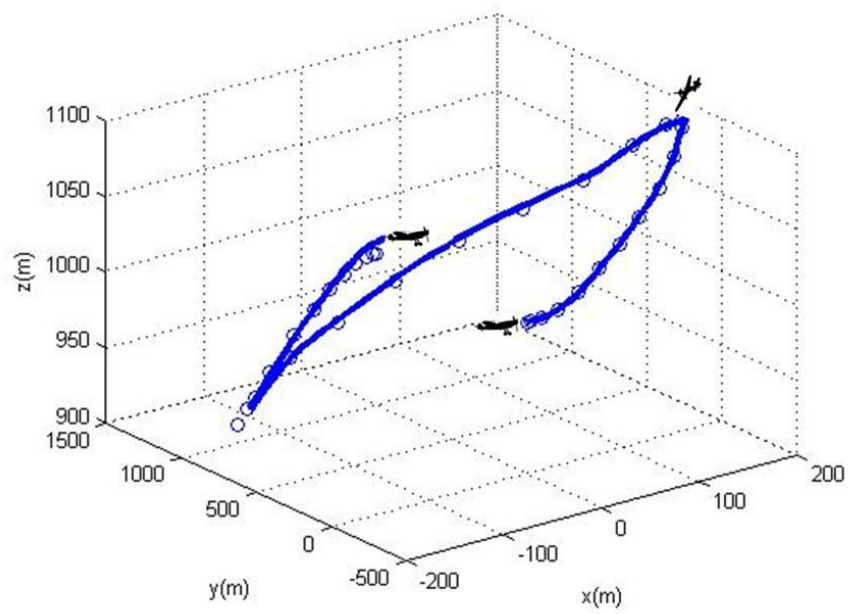


Figure 38. “Scoot Over” Trajectory 3-D View

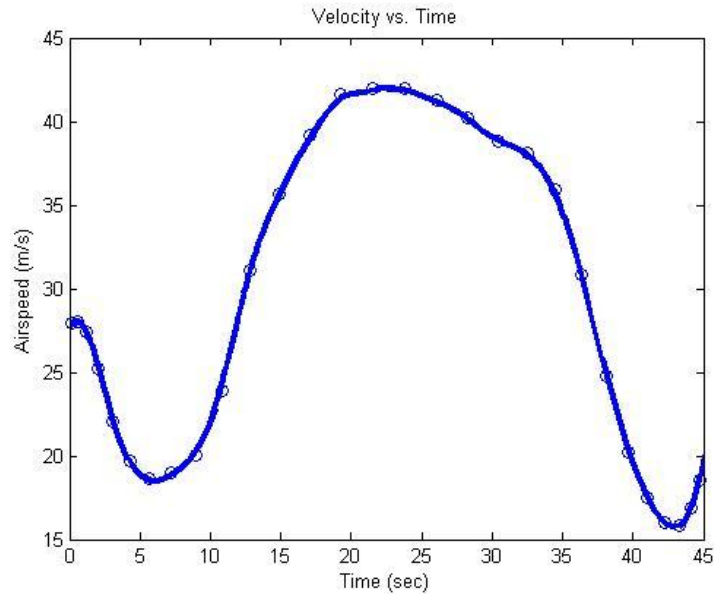


Figure 39. Velocity Profile of "Scoot Over" Maneuver

This maneuver, like the U-turn, utilizes as much of the aircraft's performance range as possible. Figure 39 shows that the maximum velocity is reached during the middle portion of the trajectory, but the aircraft also comes very close to the minimum velocity, remaining just above stall speed.

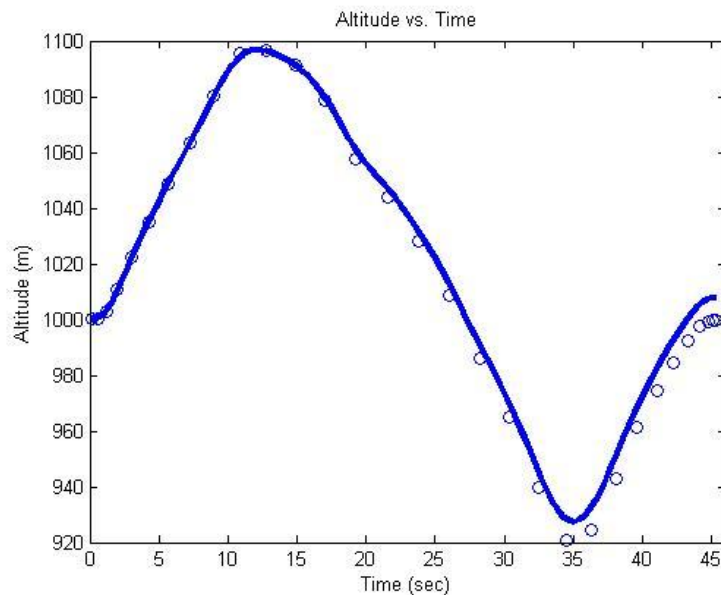


Figure 40. Altitude Profile of "Scoot Over" Maneuver

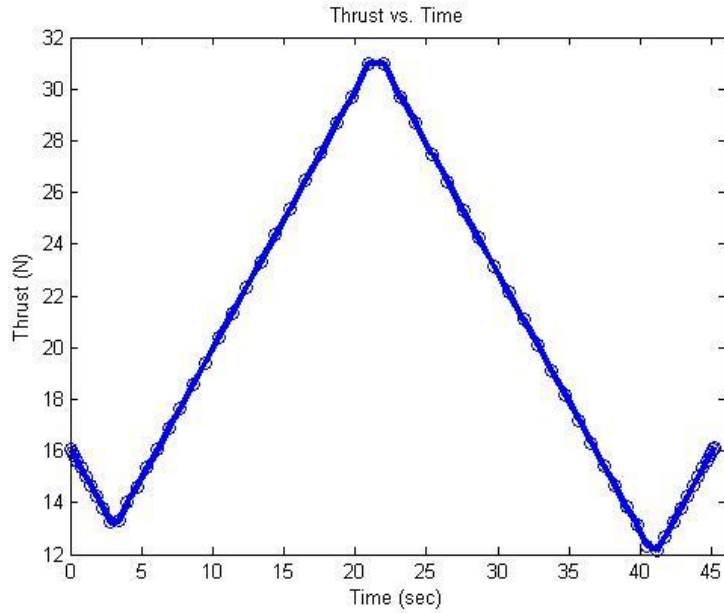


Figure 41. Thrust Profile of “Scoot Over” Maneuver

Maximum thrust and bank angle are once again reached for this maneuver (Figures 41 and 43), as well as maximum flight path angle (Figure 44) while the AoA used in this case is again minimal (Figure 42).

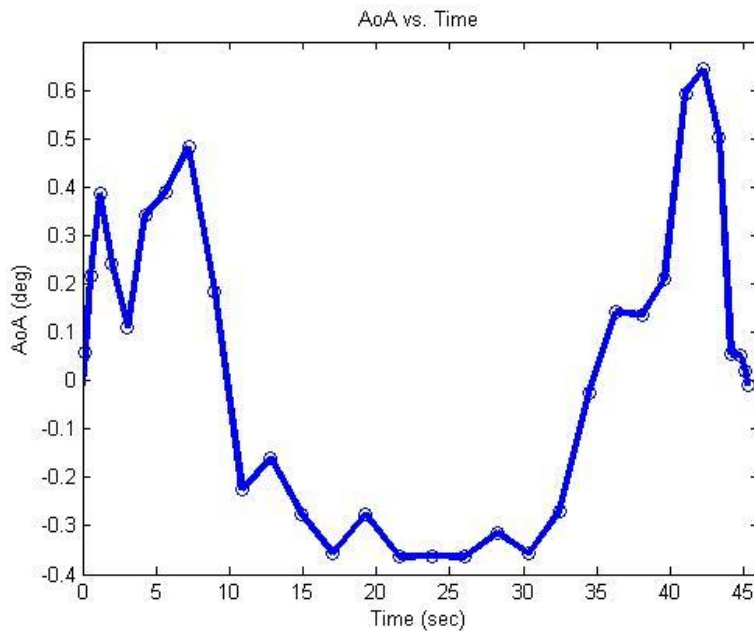


Figure 42. AoA for “Scoot Over” Maneuver

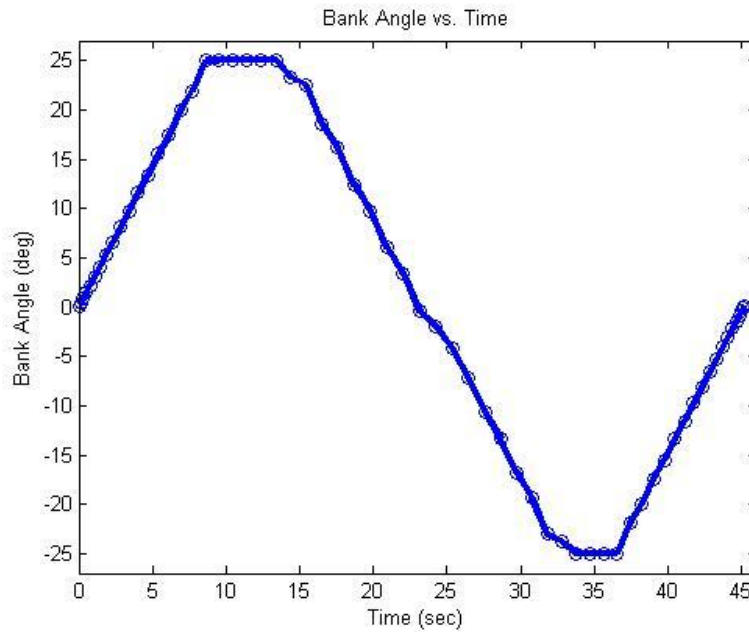


Figure 43. Bank Angle for “Scoot Over” Maneuver

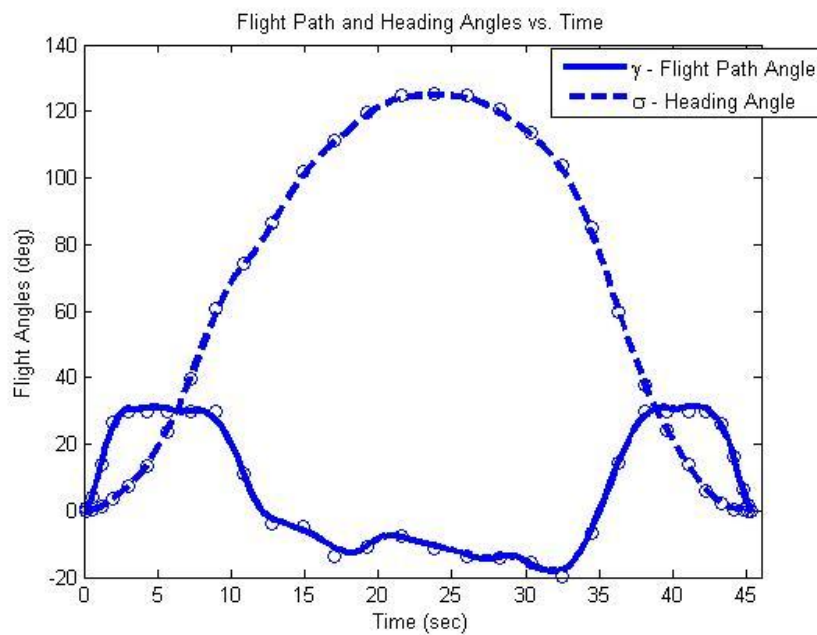


Figure 44. Flight Path and Heading Angles for “Scoot Over” Maneuver

The maneuver is accomplished in 33 seconds. This broad, S-shaped maneuver is an efficient way to bring an aircraft back over to a nearby area on the original heading in the shortest time possible.

5. Narrow-Space Reversal

a. *Description of Maneuver*

This maneuver is based on a post-stall maneuver utilized by military jet pilots as one of a catalogue of primary moves for aerial combat. The objective of the maneuver, named the Herbst Reversal or Herbst Maneuver, is to reverse heading and return through the same point from which the maneuver was started, as quickly as possible and in a small amount of space [36]. This is shown in Figure 45. Normally the move utilizes a very high angle of attack, but in this case the AoA was left free to be any value within the allowable range of the aircraft. The maneuver is performed in a narrow space to simulate the need to turn quickly and in a tight section of airspace, as a pilot engaged in aerial combat might need to do.

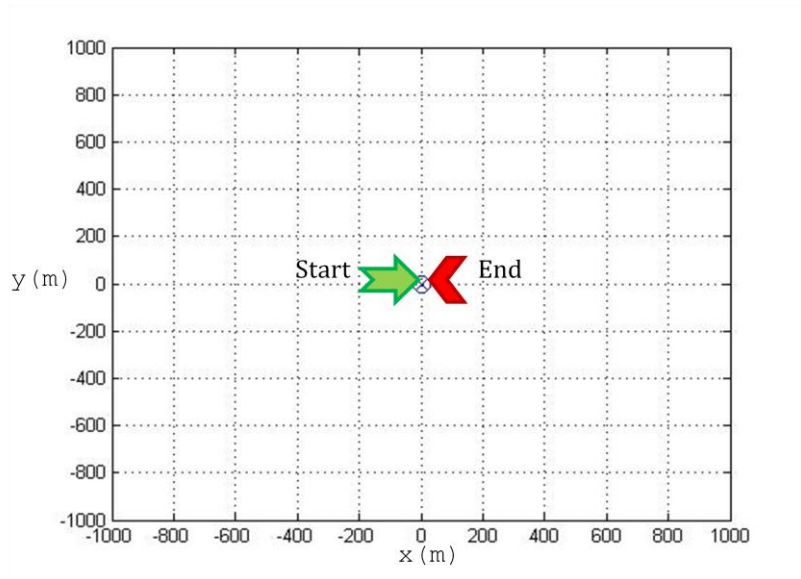


Figure 45. Top View of Reversal Maneuver

b. *Initial and Final Conditions*

This maneuver's initial and final conditions are the same, with the exception of the heading, since the goal is to come straight back through the starting point in the shortest time possible (see Table 19 and Figure 45). The final heading is opposite from the initial heading.

Initial Conditions:			Final Conditions:		
State	Value	units	State	Value	units
x_0	0	m	x_f	0	m
y_0	0	m	y_f	0	m
z_0	1000	m	z_f	1000	m
v_0	27.5	m/s	v_f	27.5	m/s
γ_0	0	rad	γ_f	0	rad
σ_0	0	rad	σ_f	π	rad
T_0	16.1	N	T_f	16.1	N
α_0	-0.0088	rad	α_f	-0.0088	rad
μ_0	0	rad	μ_f	0	rad

Table 19. Initial and Final Conditions for Reversal Maneuver

c. Box constraints

The box constraints in Table 23 are altered to force this maneuver to be executed in a narrower space than previous maneuvers, making it more similar to the Herbst maneuver, although still not at stall-inducing pitch. The x-limits on the box are broadened to ensure enough room that a solution may be found within the narrower box (see Table 20).

Lower State Bounds:			Upper State Bounds:		
	Value	units		Value	units
x_L	-5000	m	x_U	5000	m
y_L	-100	m	y_U	100	m
z_L	0	m	z_U	3000	m
v_L	13	m/s	v_U	42	m/s
γ_L	$-\pi/6$	rad	γ_U	$\pi/6$	rad
σ_L	$-\pi$	rad	σ_U	π	rad
T_L	3	N	T_U	35	N
α_L	$-\pi/12$	rad	α_U	$\pi/12$	rad
μ_L	-25	deg	μ_U	25	deg

Table 20. State Bounds for Reversal Maneuver

d. Results

The narrow-area trajectory shape can be seen in Figures 46 and 47. The optimal trajectory is flown out from the starting point and then in a tight loop. This leaves room for the aircraft to line up early to intercept the starting point on the return leg. To come back through the starting point, a conservative traditional pilot or autopilot maneuver might perform either a very large teardrop shape much wider than the one shown in Figure 48, or a fighter pilot might execute a true Herbst Reversal, which is not possible for the optimal maneuver because it is constrained to keep above the stall speed.

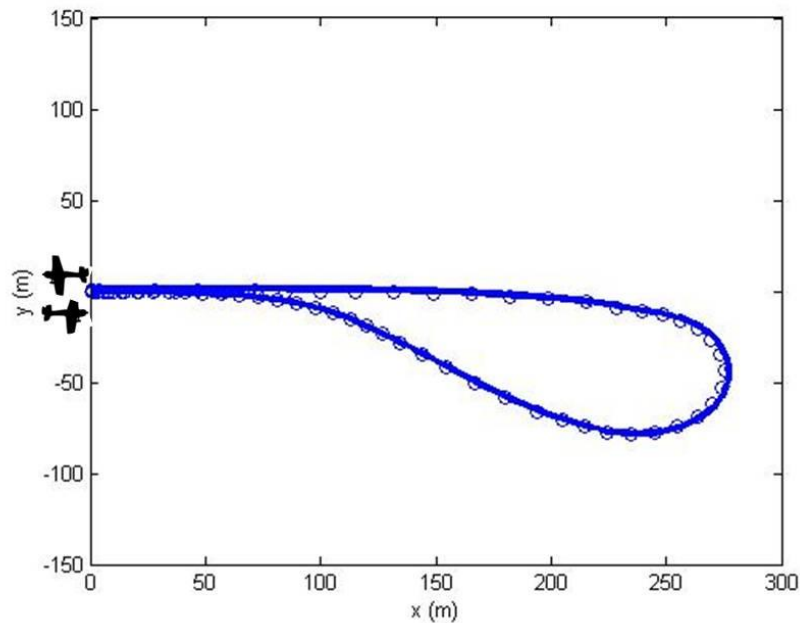


Figure 46. Overhead View of Reversal Trajectory

This maneuver also demonstrates that the optimal trajectories makes full use of limits on velocity, as shown in Figure 48, where the aircraft reaches both maximum and minimum velocity in the same maneuver. Since the minimum velocity is reached for an extended period of time (see Figure 48), it appears that it is desirable to ‘stall’ the aircraft as in the Herbst maneuver.

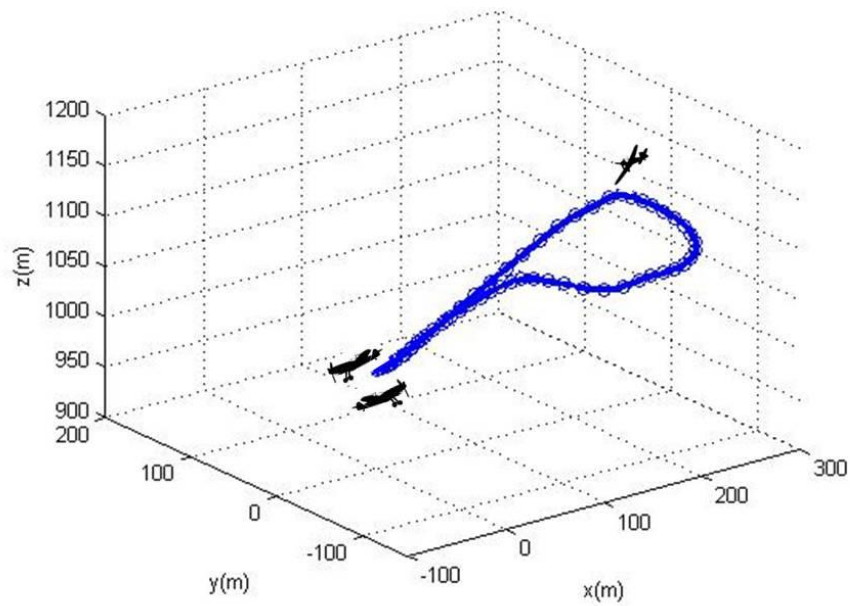


Figure 47. Reversal Trajectory 3-D View

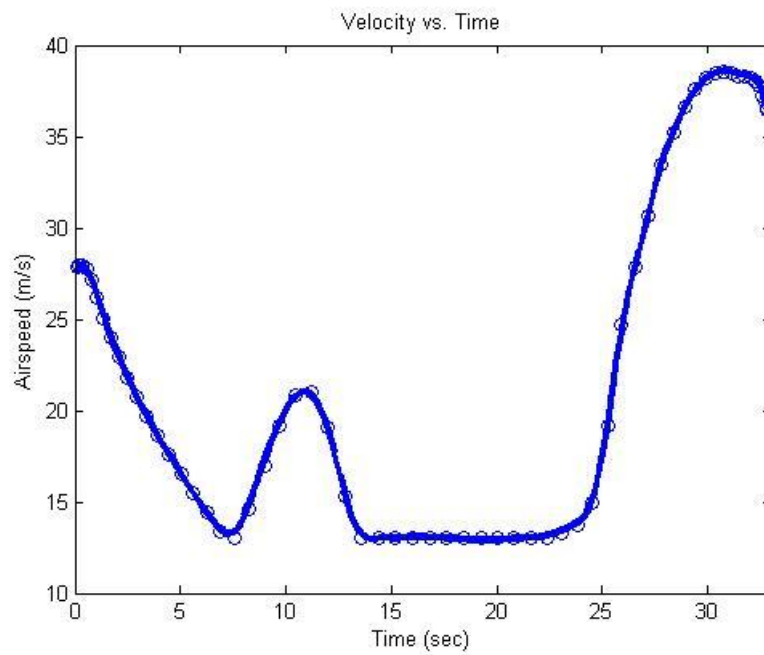


Figure 48. Velocity Profile of Reversal Trajectory

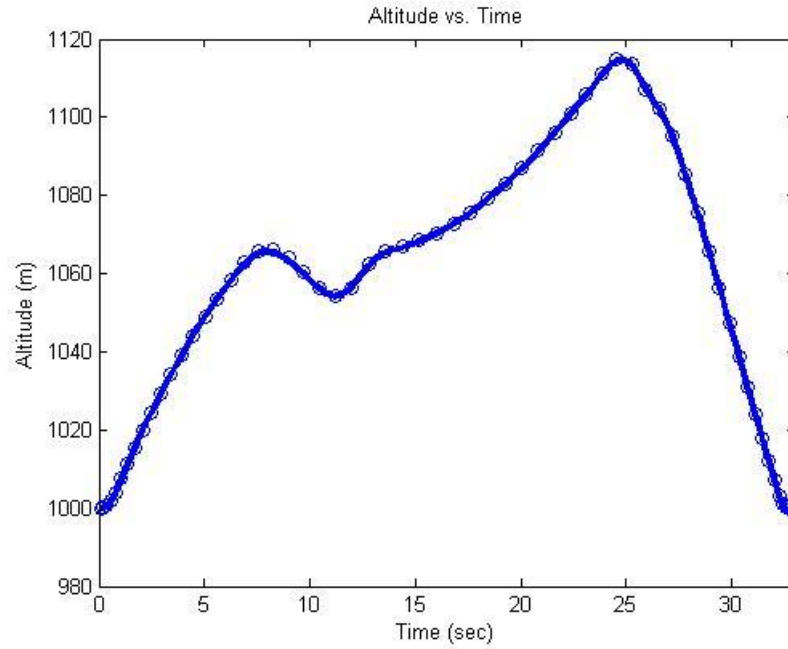


Figure 49. Altitude vs. Time for Reversal Trajectory

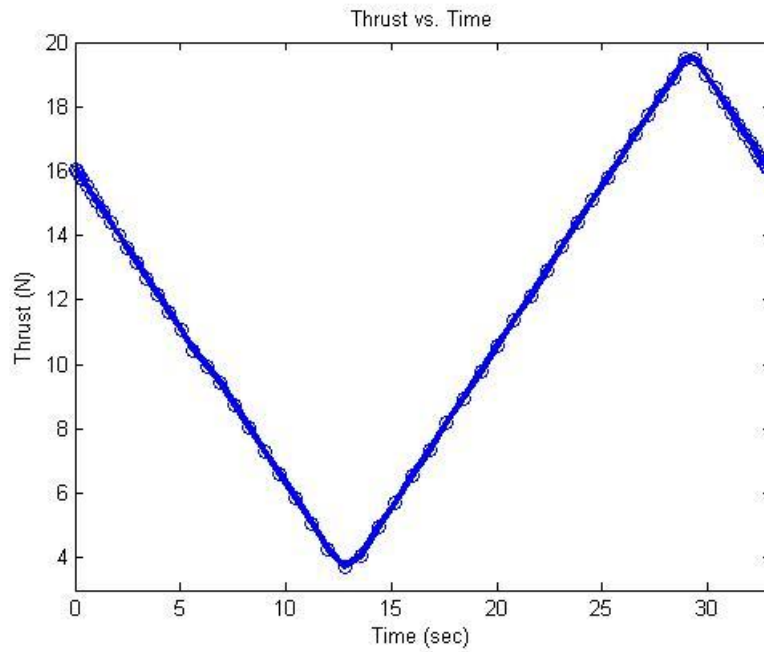


Figure 50. Thrust Profile of Reversal Trajectory

The altitude change during the maneuver is fairly significant, as seen in Figure 49. Since the width of the airspace allowed for this maneuver is reduced, more

vertical motion is required to complete the turn within the airspace allotted. The thrust value nearly reaches the minimum allowable value in the middle of the maneuver (see Figure 50), as the aircraft slows to make the tight turn.

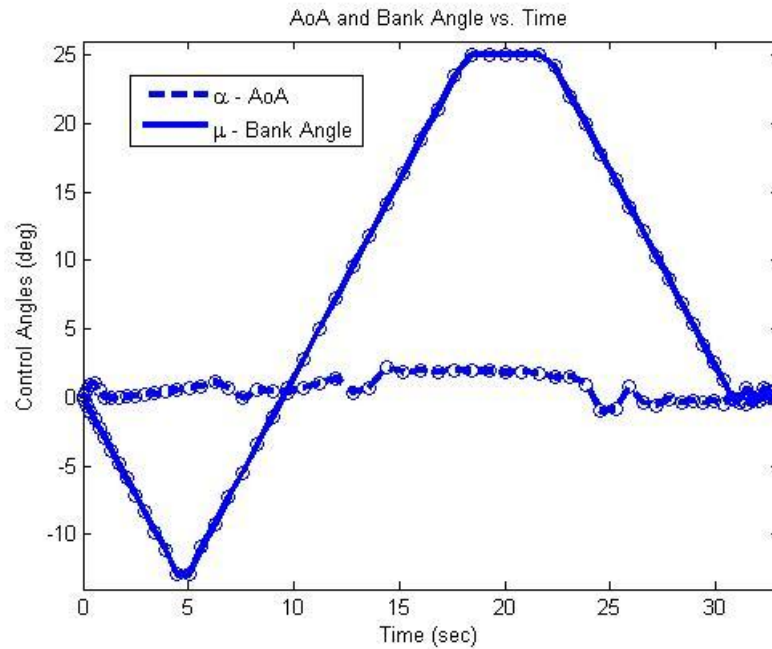


Figure 51. AoA and Bank Angles for Reversal Trajectory

Once again the maneuver uses the maximum bank angle and flight path angle to complete the turns as time-efficiently as possible, as shown in Figures 51 and 52.

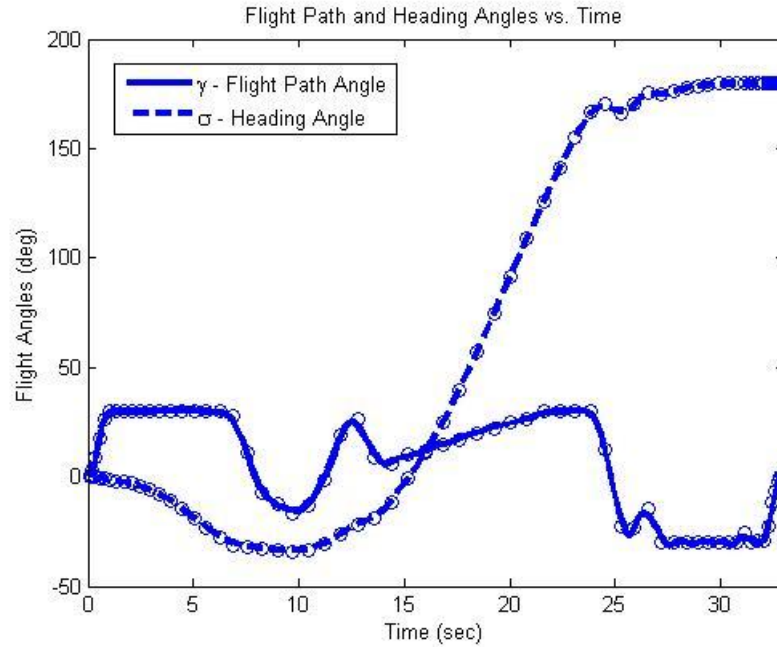


Figure 52. Flight Path and Heading Angles for Reversal Trajectory

B. DISCUSSION OF THE CANONICAL MANEUVERS

The canonical maneuvers covered a range of maneuver types and flight conditions to verify the optimization model and determine that the trajectories being generated are indeed feasible. The resulting trajectories are generally not what a typical autopilot system would use to achieve the same end, but rather are unique solutions to a minimum-time problem.

For most of the maneuvers, maximum or minimum values for several of the states—including velocity, bank angle, flight path angle, and thrust—are reached, and in some cases both the maximum and minimum are reached in the same maneuver. This demonstrates that the optimal trajectories attempt to make full use of the aircraft's performance capabilities as allotted. The maneuvers also show what can be achieved when breaking away from more traditional flight maneuvers as performed by a typical autopilot or human pilot, which for example may utilize only a small portion of the aircraft performance envelope.

V. SIMULATION ENVIRONMENT FOR TRAJECTORY VERIFICATION

The DIDO-generated trajectories should be verified using SIMULINK and HIL simulations before flight. This chapter describes a simulation environment for completing this task. SIMULINK was used to create a 6DOF model of the aircraft to conduct completely computer-simulated flight of the trajectories within the Flight Dynamics and Control Toolbox. SIMULINK was also utilized for the 6DOF model to generate the sensor inputs for the autopilot HIL simulation. The HIL simulation will allow different maneuver implementation strategies to be tested before flight.

A. AUTOPILOT SYSTEM

The autopilot system selected for use (SLUGS) was developed as a collaborative effort between the Autonomous Systems Laboratory at the University of California, Santa Cruz and Dr. V. Dobrokhodov at NPS. The unmanned vehicle guidance system was developed as a rapidly-reconfigurable autopilot system that can be used primarily for small UAVs, but also with other unmanned systems with a wide variety of uses [3]. The autopilot system connects with a ground control station and a HIL simulator. The autopilot board is shown in Figure 53.

The autopilot hardware and software were designed from the beginning to be open-source, in order to encourage further work and research and development of small UAV systems [37] to further the field and to determine just how flexible and reconfigurable the autopilot system can be.



Figure 53. Autopilot Board With Size Reference

1. Development and Description

While several commercial autopilots are readily available, all are based off older waypoint navigation systems, and they are not easy to modify or reconfigure for a specific use [37]. Since it is anticipated that code modifications are necessary to implement the optimal maneuvers, the open-source nature of the MONARC autopilot system is crucial. The autopilot software works with SIMULINK's automatic code generation features to incorporate changes in the flight control system without requiring direct modification of the source code [37]. The iterative prototyping process is shown in Figure 54.

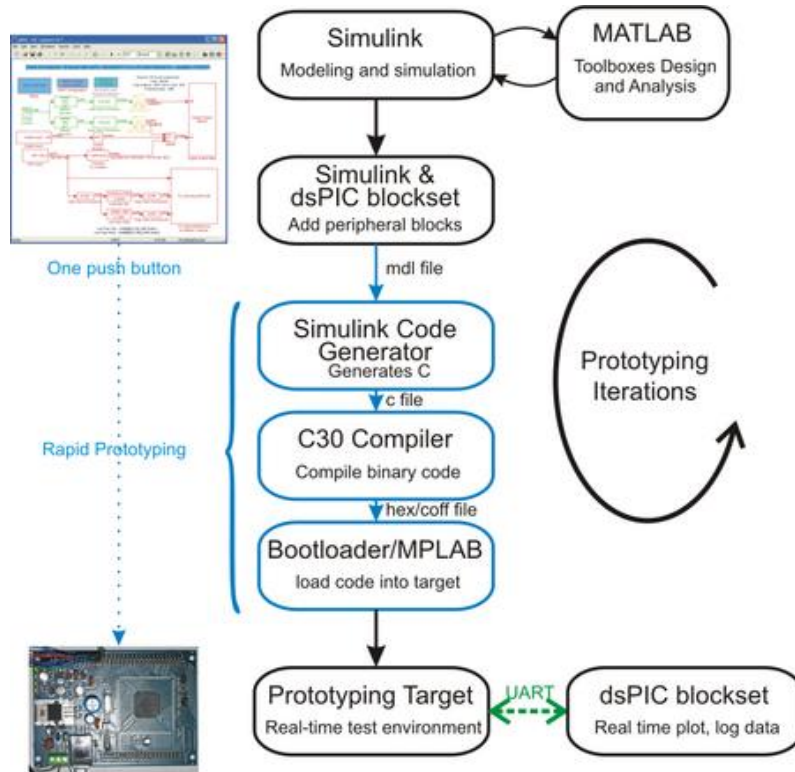


Figure 54. Control System Rapid Prototyping Process (From [40])

One key difference between the selected autopilot system and most commercial autopilots is that the autopilot tasks are split between two processing units, each operating independently. One unit is used for position and attitude estimation, and one for navigation and control. The two DSCs, named sensor DSC and control DSC, allow higher processing power, and independent changes to either unit that will not impact the other, as long as the interface is preserved [38].

The sensor DSC, as the name implies, receives the data from the onboard sensors. It then uses that data to compute the MONARC's attitude and position, via an algorithm that fuses all available sensor data [37]. The computed attitude and position are sent via a communication protocol to the control DSC. The control DSC uses that data and the commands sent from the GCS to generate commands for control surfaces and actuators [37]. The control DSC also sends telemetry reports back to the GCS.

2. Components

The autopilot and its peripheral hardware includes multiple sensors, a radio modem, digital signal controllers for sensors and controls, transceivers, actuators, and regulators in a unique hardware architecture as shown in Figure 55.

The autopilot includes onboard sensors as well as connected sensors external to the autopilot printed circuit board. The outputs of these sensors are fused together for use in accurately determining location and controlling flight. The sensor suite components (see Table 25) include angular rate sensors, accelerometers, a barometer, a differential pressure sensor, a thermometer, a battery monitor, a magnetometer, and a GPS module, [39]. The autopilot hardware works in conjunction with the QGroundControl GCS system [30] to pilot the aircraft. The full list of onboard sensors for the autopilot control function is shown in Table 21.

Sensor	Type(s)	Notes
GPS	Position, heading, altitude	
3-axis IMU	Position, attitude	
Accelerometer	Attitude	Part of IMU
Magnetometer	Attitude	Part of IMU
Rate Gyro	Attitude	Part of IMU; Fused with GPS/accelerometer via complementary filter
Static pressure sensor (barometer)	Altitude	
Dynamic pressure sensor	Airspeed	
Pitot Tube	Airspeed	

Table 21. Autopilot Sensors (From [39], [41])

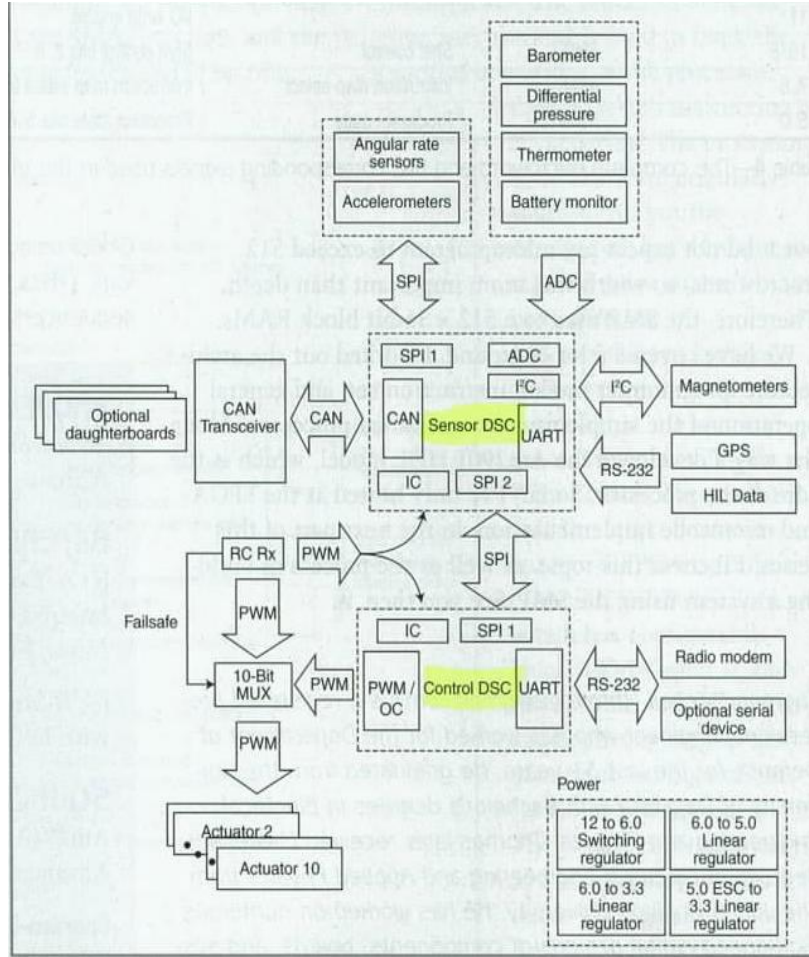


Figure 55. Autopilot Hardware Architecture (From [39])

3. 6DOF SIMULINK Model

The verification setup used for this work includes a 6DOF nonlinear dynamic model of the Mentor aircraft with inner and outer loop simulation capabilities (see Figure 56).

4. Integration with MONARC

The MONARC aircraft can be operated in manual, autonomous, or HIL modes. The manual mode requires a trained safety pilot to fly the aircraft, the autopilot mode utilizes the autopilot control logic for waypoint navigation, and the HIL mode provides sensor input while still using the autopilot hardware to pilot a simulated aircraft [42].

The autopilot board is installed on a modified portion of the Mentor fuselage. The aircraft also has slight modifications to facilitate installation of sensors and antennae.

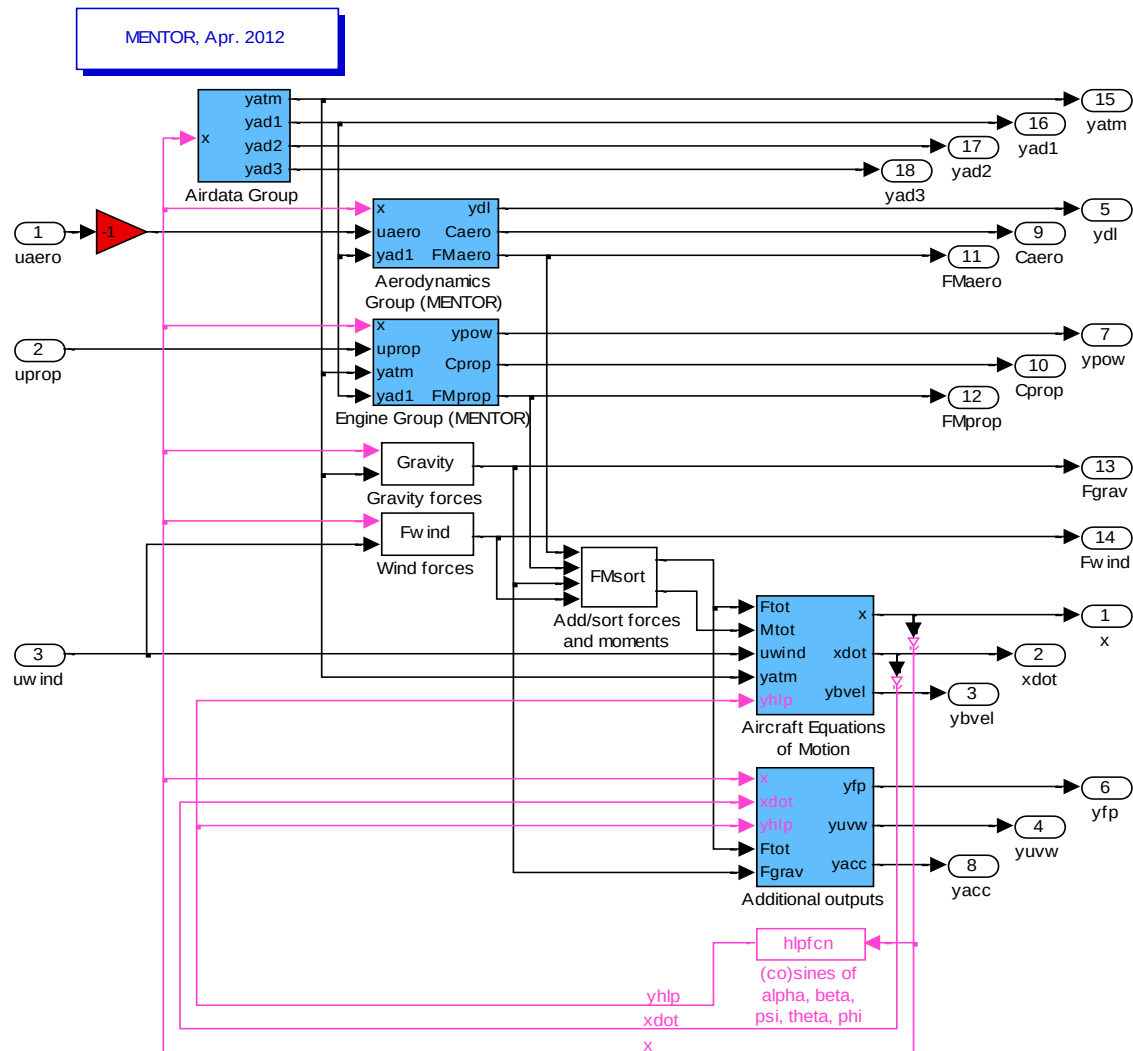


Figure 56. SIMULINK Diagram of 6DOF Simulator (From [27])

5. Navigation

The autopilot performs waypoint navigation using waypoints sent from the user via the ground control software. The inner loop control portion is performed mainly by the control DSC, and this is shared between a lateral navigation channel for sideslip and roll, and a longitudinal navigation channel for altitude and speed [39]. The split

navigation channels allow the algorithms to be less complex. The outer loop navigation portion is performed using the L2+ waypoint following algorithm [39], the details of which are beyond the scope of this work. In order to implement the optimal maneuvers generated in this thesis the outer loop navigation algorithm will be disengaged.

6. Ground Station Operation

The ground control software can be run on a laptop computer, making it portable for flight tests. To conduct HIL testing, the autopilot is put into HIL mode and configured for waypoint navigation or commands may be sent manually via the RC transmitter. The SIMULINK 6DOF model must be running, and generating synthetic sensor data to send to the hardware.

B. HIL SIMULATION

A hardware-in-the-loop simulator can be used to test the autopilot setup by feeding simulated sensor data to the hardware, so that flight may be simulated and the autopilot's functionality can be verified.

1. Background

HIL simulation is invaluable because all actual autopilot systems and components can be checked out prior to flight, and avoid possible damage to the physical aircraft if there is a problem. In the case of this work, the HIL simulation can verify that the model-generated optimized trajectories can be flown by the aircraft, and that the optimization model is in fact viable. Moreover, the HIL setup can be used to determine how the optimal trajectories can best be incorporated into the existing flight control loops.

The HIL simulation operates as the aircraft will in flight, just with simulated inputs. As system identification improves, the HIL simulation fidelity will increase. The HIL simulator closely duplicates the flow of information that would result in physical flight with the autopilot and gives a good indication of how the aircraft would behave in the air [38]. The autopilot will generate commands in response to simulated sensor input received, and the SIMULINK model will respond to the actuator commands [43]. This forms a loop between the simulator and the actual autopilot hardware, where the autopilot

is flying the simulator with actual commands sent to physical actuators, and back to the SIMULINK model to provide feedback to the dynamic control system [35].

HIL simulations save on a variety of costs. Monetary costs are reduced because tests do not require rental of flight facilities, use of actual aircraft, or travel to flight facilities. As the NPS- and Navy-approved flight test facility is a several-hour's drive away, being able to conduct HIL simulations prior to flight testing saves a great deal of time and expense. Risk of damaging hardware in testing is reduced, and more tests may be completed in a shorter period of time. Also, being able to show results of successful HIL testing will aid in the flight approval processes required by NAVAIR. While a HIL simulation is not exactly the same as a physical flight test, the results are such that the number of flight tests may be reduced, and conducted with a greater degree of confidence that there will not be a catastrophic failure because HIL tests have shown any major issues beforehand.

2. Apparatus

The full HIL simulation apparatus consists of one desktop computer running the SIMULINK 6DOF model to respond to commands and generate sensor inputs; a laptop computer equipped with the ground control system; the manual aircraft controller; and the autopilot. HIL mode can also be used with an entire MONARC aircraft connected. In the configuration pictured in Figure 57 there are servo motors connected, but the full airframe is not present. When the autopilot is operating in HIL mode, the simulated sensor data generated by the SIMULINK model is the input [41], but the feedback and commands from the autopilot are real—the actuators and motors move, and if the full MONARC aircraft were connected, the control surfaces and propeller would also respond to the autopilot commands.

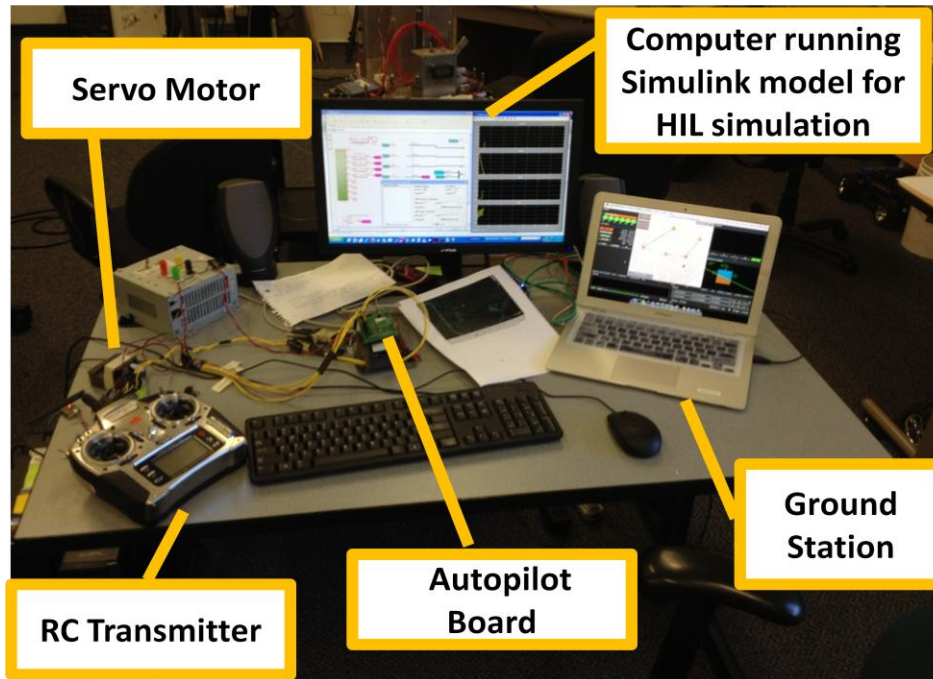


Figure 57. HIL Simulation Apparatus (After [39])

C. TRAJECTORY IMPLEMENTATION

The SIMULINK model (see Figures 58 and 59) uses the generated trajectories as inputs for the model simulations. The navigation control law is designed so that once the aircraft passes a waypoint the aircraft will attempt to steady on the next leg. The control laws within the model include both Stability Augmentation Systems and Control Augmentation Systems [30]. These are shown in Figure 55 as the SCAS for pitch, roll, and yaw. The optimal trajectory values are interpolated at a 100Hz interval for command inputs, as required by the SIMULINK autopilot.

The control law input setup can be adapted by using the appropriate entry points to insert various combinations of the optimal trajectory signals, which are then processed by the various loops to create elevator, rudder, and aileron, and throttle commands to the aircraft.

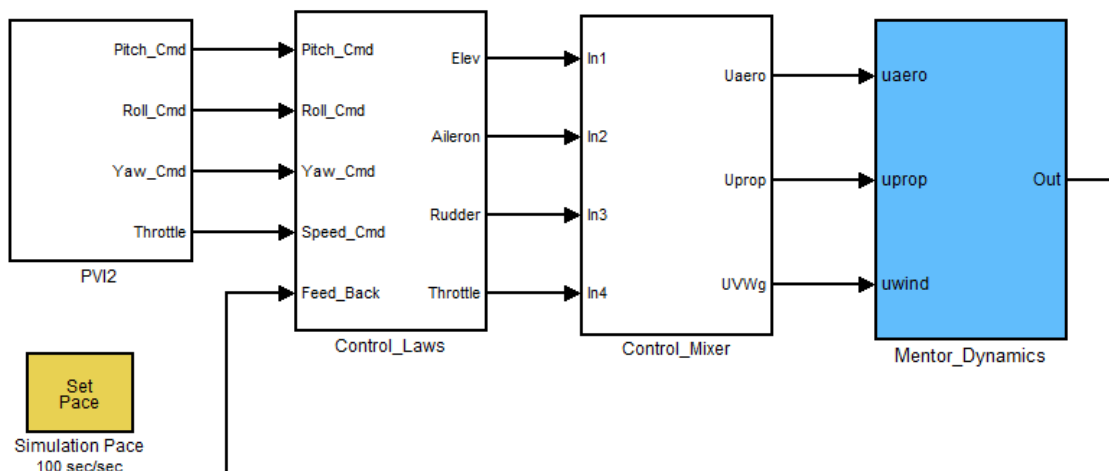


Figure 58. SIMULINK Model of Mentor—Top Level View

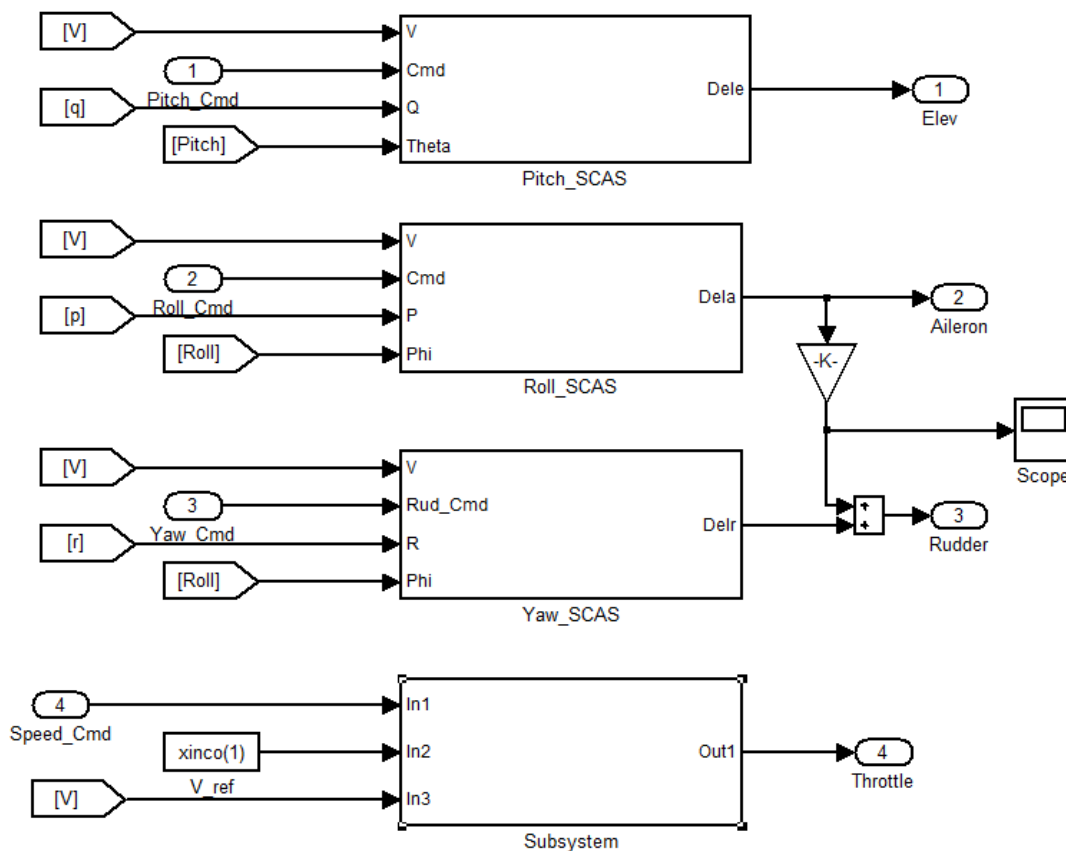


Figure 59. SIMULINK Model—Inside Control Laws Block Diagram

Since the optimization model is 3DOF and is used to send trajectory command information to a 6DOF autopilot, values for the Euler angles and Euler angle rates must first be converted from the available 3DOF states. Relevant equations [44] are:

$$\begin{aligned} p &= \dot{\mu} \cos \alpha - \dot{\sigma} \sin \gamma \cos \alpha - \dot{\sigma} \cos \mu \cos \gamma \sin \alpha + \dot{\gamma} \sin \mu \sin \alpha \\ q &= \dot{\gamma} \cos \mu + \dot{\sigma} \sin \mu \cos \gamma \\ r &= \dot{\mu} \sin \alpha - \dot{\sigma} \sin \gamma \sin \alpha - \dot{\sigma} \cos \mu \cos \gamma \cos \alpha + \dot{\gamma} \sin \mu \cos \alpha \end{aligned} \quad (5.1)$$

$$\begin{aligned} \theta &= \sin^{-1}(\sin \alpha \cos \gamma \cos \mu + \cos \alpha \sin \gamma) \\ \psi &= \sigma + \sin^{-1}\left(\frac{\sin \alpha \sin \mu}{\cos \theta}\right) \\ \phi &= \sin^{-1}\left(\frac{\cos \gamma \sin \mu}{\cos \theta}\right) \end{aligned} \quad (5.2)$$

$$\begin{aligned} \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta} \end{aligned} \quad (5.3)$$

The converted optimal trajectories are used as command inputs, and several combinations of the various available inputs were used to determine which combination worked best in order to reproduce the optimal maneuver. First, the simulation used inputs for altitude, velocity, and heading as shown in Figure 60. Another alternative is to use pitch, roll, and velocity commands to fly the trajectory, as shown in Figure 61. The efficacy of each of these approaches for maneuver implementation is demonstrated in the next chapter.

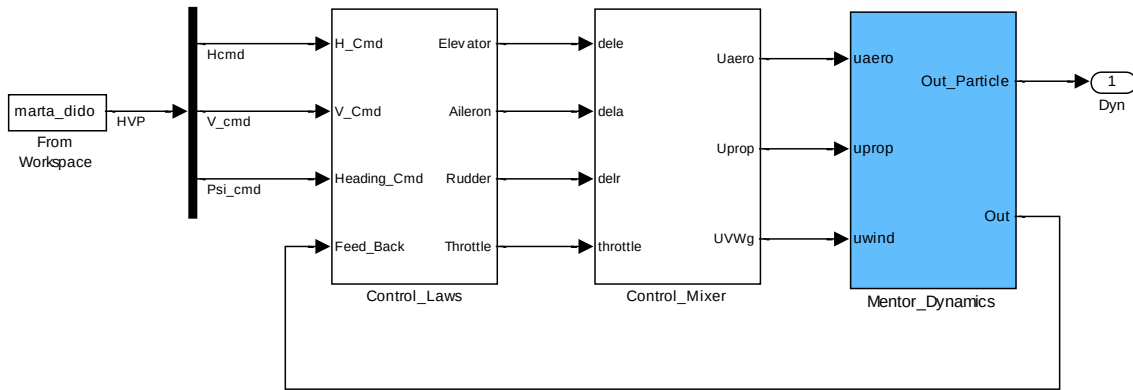


Figure 60. SIMULINK Verification Model Using Altitude, Velocity, and Heading Trajectory Inputs (From [29])

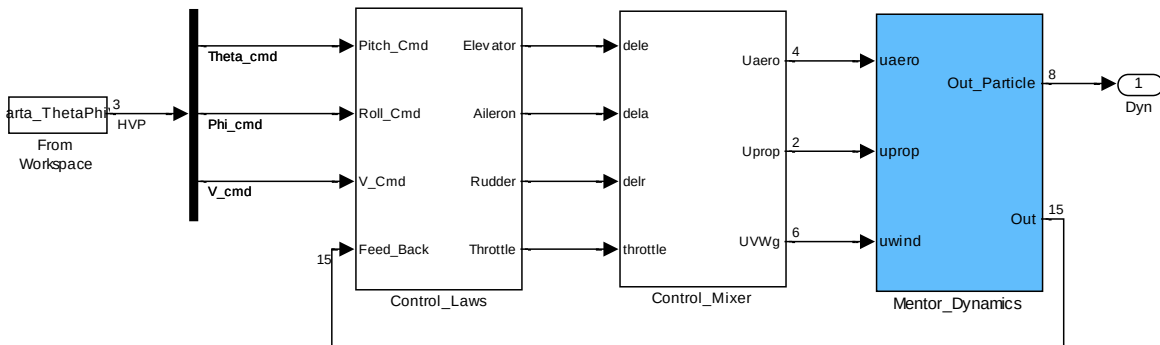


Figure 61. SIMULINK Verification Model Using Pitch, Roll, and Velocity Trajectory Inputs (From [29])

Other variations for inputs to the 6DOF model are possible, taking any of the 3DOF trajectory states or the converted Euler Angles or Euler Rates and using them for the 6DOF inputs.

VI. MANEUVER IMPLEMENTATION

A. EQUILATERAL TRIANGLE MANEUVER

Two variations of an optimal trajectory were created for the main maneuver implementation example, flying three waypoints to complete an equilateral triangle. The turn maneuvers are each the same and can be repeated and utilized by the autopilot in a chain to complete an equilateral triangle, with a return to trimmed SAL flight executed by the autopilot in between turns.

This maneuver is an application of the concept of a library or database of optimal maneuvers that can be completed sequentially to perform a more complex maneuver. In this case, a 120-degree turn is executed three times to complete a trajectory based on a triangle of waypoints. The equilateral triangle maneuver requires a very sharp turn, so it will challenge the controller's abilities to follow around a tight turn more than a square or rectangle, but is also simple to implement repeatedly because the turn and leg are the same each time.

1. Minimum Time to Turn and Intersect Next leg

First, a time-optimal trajectory was generated to start at the top of the triangle, beginning on the heading of the left-hand leg, and complete a 120-degree right turn and return to steady flight in the shortest time possible on the next leg—the line formed between the top of the triangle and the lower right corner of the triangle.

a. Description of Maneuver

This maneuver performs a 120-degree turn at the top of an equilateral triangle, making the turn and returning to steady flight along the leg of the triangle. The triangle is 1000m on each side. The maneuver begins and ends at 100m altitude, and the top of the triangle is considered the point (0,0,100). The maneuver is depicted in Figure 62.

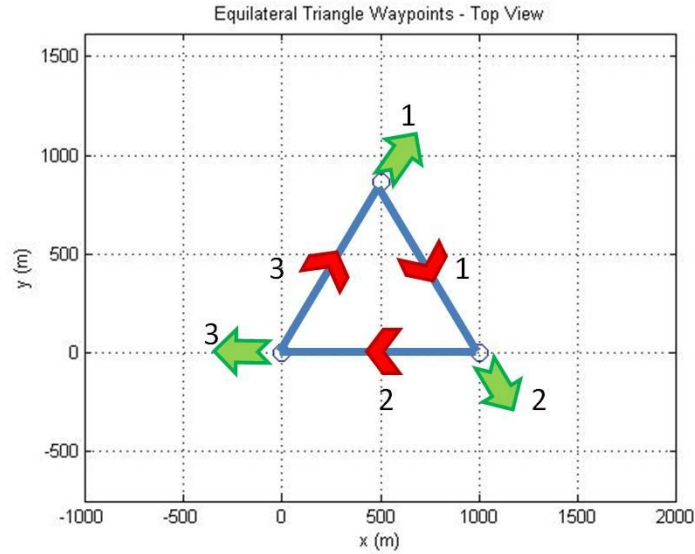


Figure 62. Equilateral Triangle Maneuver—Complete Turn and Steady on Next Leg

b. Updated Control Limits

The pitch rate control limits based on the 6DOF computer model were revised for the chosen test trajectory to include the added constraint of 2g load limits as recommended by the safety pilot. The load factors shown in Figure 63 are for the original u_α limits, the new limit to keep n under 2g, and an in-between value. The u_α limit had to be reduced by nearly an order of magnitude to meet this requirement. The required values are listed in Table 24.

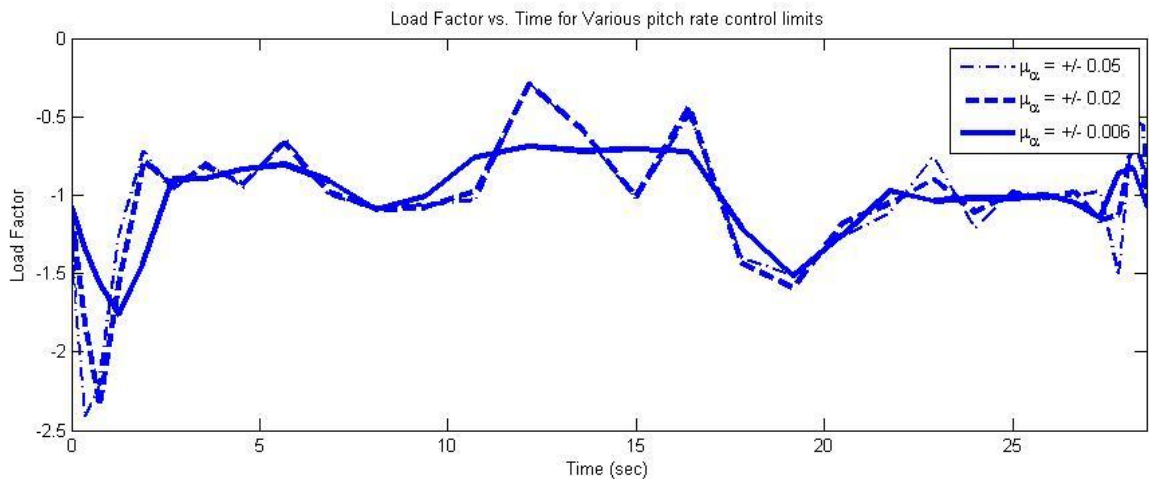


Figure 63. Load Factors with Varying μ_α Limits for Equilateral Triangle Maneuver

c. Initial and Final Conditions for Turn at Top of Triangle

The conditions are set up to correspond with the maneuver depicted in Figure 62, as shown in Tables 22, 23, and 24.

Initial Conditions:			Final Conditions:		
State	Value	units	State	Value	units
x_0	0	m	x_f	x_f	m
y_0	0	m	y_f	$y_f - x_f \tan(-\pi/3)$	m
z_0	100	m	z_f	100	m
v_0	25	m/s	v_f	25	m/s
γ_0	0	rad	γ_f	0	rad
σ_0	$\pi/3$	rad	σ_f	$-\pi/3$	rad
T_0	15.3	N	T_f	15.3	N
α_0	0.0193	rad	α_f	0.0193	rad
μ_0	0	rad	μ_f	0	rad

Table 22. Initial and Final Conditions 120-degree Turn to Next Leg

d. Box Constraints

Lower State Bounds:			Upper State Bounds:		
	Value	units		Value	units
x_L	-1.00E+04	m	x_U	1.00E+04	m
y_L	-1.00E+04	m	y_U	1.00E+04	m
z_L	0	m	z_U	1000	m
v_L	13	m/s	v_U	42	m/s
γ_L	$-\pi/6$	rad	γ_U	$\pi/6$	rad
σ_L	$-\pi$	rad	σ_U	π	rad
T_L	3	N	T_U	35	N
α_L	$-\pi/12$	rad	α_U	$\pi/12$	rad
μ_L	-25	deg	μ_U	25	deg

Table 23. State Bounds for 120-degree Turn to Next Leg

**Lower Control
Bounds:**

	Value	units
u_{TL}	-1	N/ s
$u_{\alpha L}$	-0.006	rad/ s
$u_{\mu L}$	-0.05	rad/ s

**Upper Control
Bounds:**

	Value	units
u_{TU}	1	N/ s
$u_{\alpha U}$	0.005	rad/ s
$u_{\mu U}$	0.05	rad/ s

Table 24. Control Bounds for 120-degree Turn to Next Leg

e. Event Bounds

The events function for this maneuver had to be written so that the final value of x and y was along the line between the first and second waypoints, as shown as the reference line in Figure and the final values in Table 23.

f. Trajectory Comparison with Typical Controller

First, the optimal trajectory was compared with the triangle maneuver as flown using a typical autopilot controller given the same waypoints, conditions, and constraints. As shown in Figure 64, the optimized maneuver intercepts the line far quicker than the standard controller maneuver.

In the SIMULINK Model with the conventional controller, the time to steady within 2% of the distance to the leg using a typical controller was 46 seconds. The optimized maneuver to steady on the next leg took only 28 seconds to complete. This was 18 seconds faster, or a 38% cost improvement.

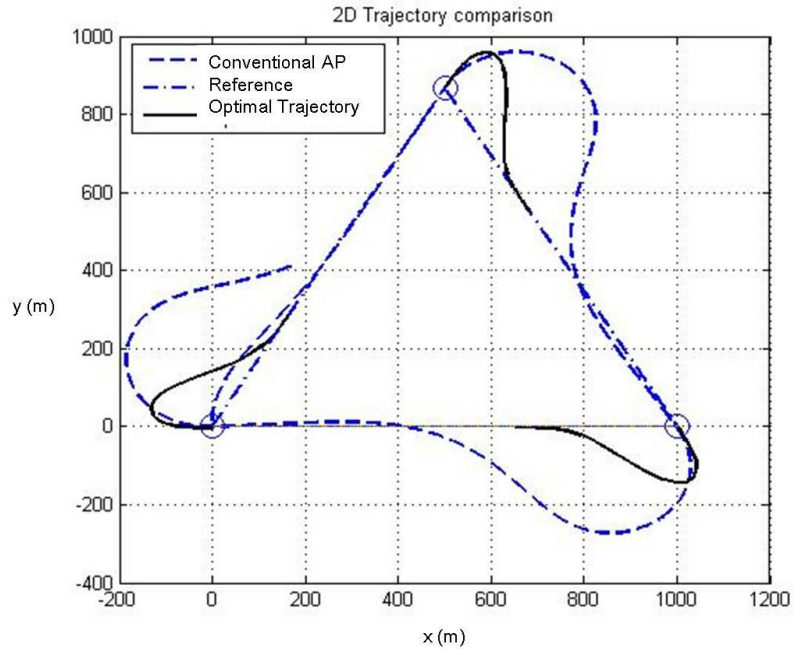


Figure 64. Trajectories for Min Time to Turn and Intercept Next Leg

2. Minimum Time to Turn and Arrive at Next Waypoint

The other variation for the triangle maneuver was to minimize time to turn and reach the next waypoint. This would allow for the maneuver to be conducted three consecutive times by the autopilot to complete the triangle without any other piloting required in between.

a. Description of Maneuver

This maneuver is the same as the first, except rather than steadying on the leg of the triangle, the goal is to arrive in minimum time at the next waypoint, as shown in Figure 65.

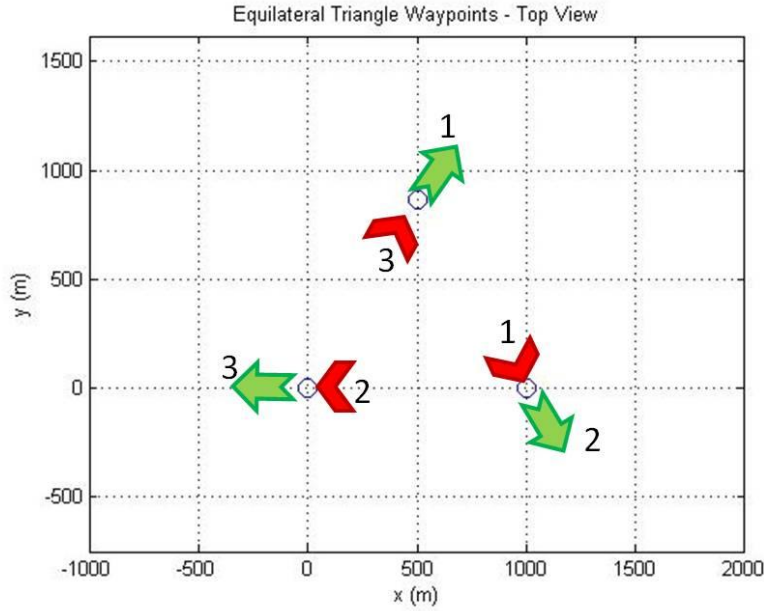


Figure 65. Equilateral Triangle Maneuver—Complete Turn and Arrive at Next Waypoint

b. Initial and Final Conditions for Turn to Next Waypoint

The initial and final conditions for this maneuver are similar to those for the turn to the next waypoint, except for the final x and y conditions, as shown in Table 25. All other bounds remain the same as the previous maneuver.

Initial Conditions:			Final Conditions:		
State	Value	units	State	Value	units
x_0	0	m	x_f	500	m
y_0	0	m	y_f	-866	m
z_0	100	m	z_f	100	m
v_0	25	m/s	v_f	25	m/s
γ_0	0	rad	γ_f	0	rad
σ_0	$\pi/3$	rad	σ_f	$-\pi/3$	rad
T_0	15.296	N	T_f	15.296	N
α_0	0.0193	rad	α_f	0.0193	rad
μ_0	0	rad	μ_f	0	rad

Table 25. Initial and Final Conditions 120-degree Turn to Next Waypoint

c. *Resulting Trajectories*

The optimal trajectory in this case is a much smoother path around the triangle than that of the typical controller (see Figure 66). The advantage of this maneuver over the previous one is that it can be flown as three consecutive 120-degree turns to the next waypoint, without other trajectory commands or autopilot SAL flight required between waypoints. The disadvantage, though, is that it is a much slower time to turn than in the maneuver to intersect the next leg.

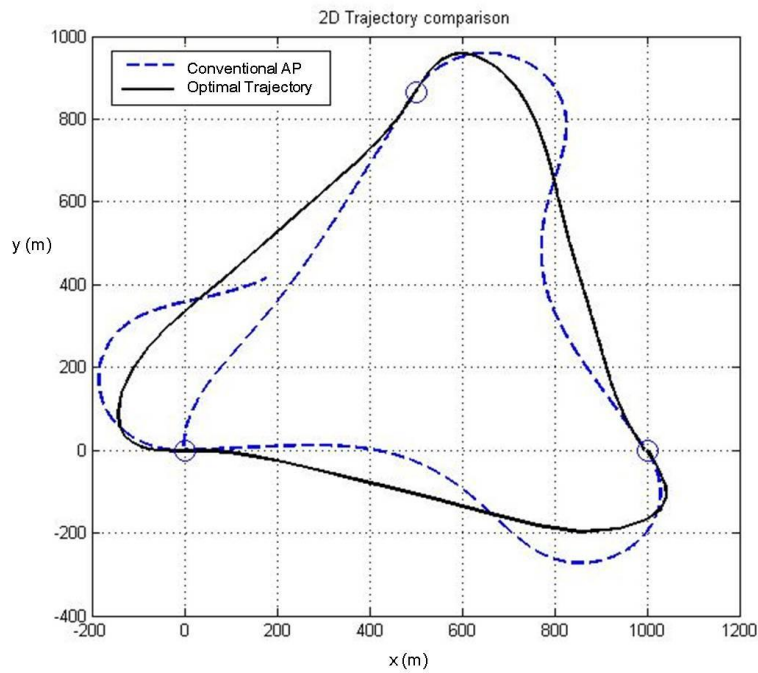


Figure 66. Trajectories for Minimum Time To Arrive at Next Waypoint

3. **Optimal Trajectory Implementation**

a. *Velocity, Altitude and Heading Inputs*

The comparison has been made between the optimal trajectory and a typical controller, so the next step is to verify whether a conventional controller designed specifically for the MONARC can follow the commanded optimal trajectories. The time-

optimal trajectory was sent to the controller for implementation to determine whether the controller could execute the first turn of the optimal trajectory as shown for the triangle maneuver from Figure 64.

In order to verify that the optimal trajectory can be flown by the MONARC controller, the optimal trajectory was used for the command inputs. The control laws for this controller are written for pitch, roll, yaw, and throttle inputs (see Figures 58 and 59). As these are not the states generated for the optimal trajectory, a combination of inputs must be selected and appropriate conversions made to use the 3DOF inputs for the 6DOF model and control laws. The controller sampling rate is 50Hz, so the trajectory states are interpolated at that interval to provide trajectory inputs at the precise frequency required. The first simulation used v , z and σ optimal trajectory values as command inputs, with the altitude and velocity inputs being combined and converted for throttle command inputs and the heading and altitude inputs being used to generate commands for the Euler angles. The scheduled gains used for this simulation were [45]:

$$\begin{array}{ll}
 \text{SAS Control Gain} & \text{CAS Control Gain} \\
 K_p = \begin{pmatrix} 2.0 \\ 0.975 \end{pmatrix} * 1.5 & K_\phi = \begin{pmatrix} 1.32 \\ 1.1 \end{pmatrix} * 1.25 \\
 K_q = \begin{pmatrix} 2.0 \\ 1.0 \end{pmatrix} * 1.5 & K_\theta = \begin{pmatrix} 1.1 \\ 1.53 \end{pmatrix} * 2 \\
 K_r = \begin{pmatrix} 2.0 \\ 1.0 \end{pmatrix} * 1 & K_\psi = \begin{pmatrix} 0.9 \\ 1.0 \end{pmatrix} * 2
 \end{array} \tag{6.1}$$

AP Control Gain

$$\begin{array}{l}
 K_v = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \\
 K_h = \begin{pmatrix} 0.008 \\ 0.005 \end{pmatrix} * 1.2
 \end{array}$$

The trajectory as flown by the model controller compared with the optimal trajectory commanded is shown in Figure 67.

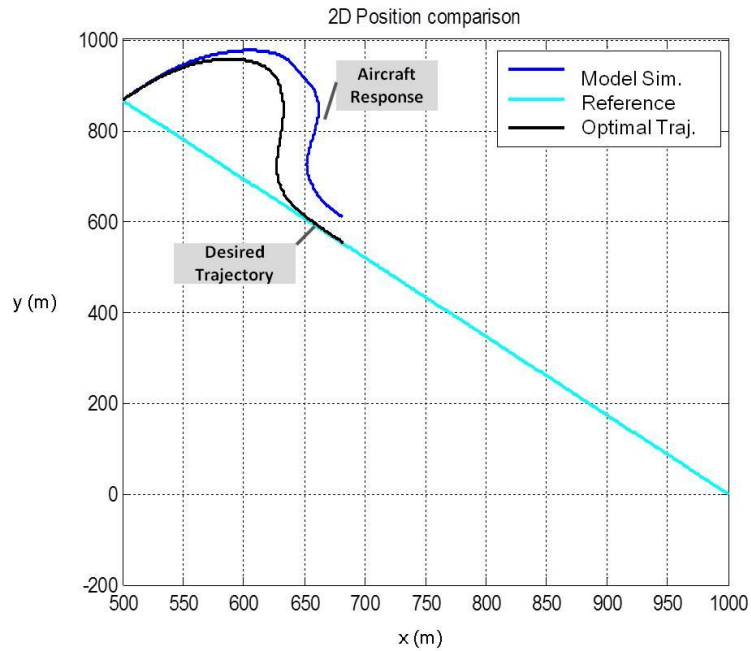


Figure 67. Top view of 120-degree Turn to Next Leg with v , z , σ Inputs to SIMULINK Model (From [45])

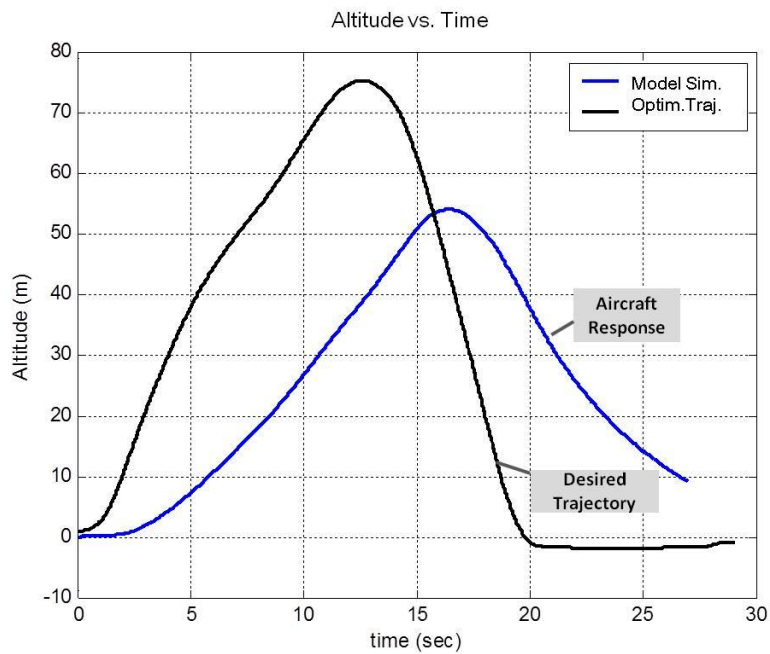


Figure 68. Altitude profile of 120-degree Turn to Next Leg with v , z , σ Inputs to SIMULINK Model (From [45])

In this case the trajectory-following performance is not very good because the aircraft does not closely follow the altitude trajectory or the optimal velocity trajectory. The aircraft does not achieve the desired end-states and does not follow the next leg of the triangle after the turn. The views of particular states over time in Figures 67 through 69 help to illustrate what the possible issues with this maneuver implementation are.

As shown in Figures 68 and 69, the autopilot was not able to closely follow the optimal trajectory using these inputs even with gain tuning. The altitude-following performance was especially poor, likely because the combination of these inputs was not the best for the conventional control laws to follow. This also suggests that new control laws may be needed for this aircraft model.

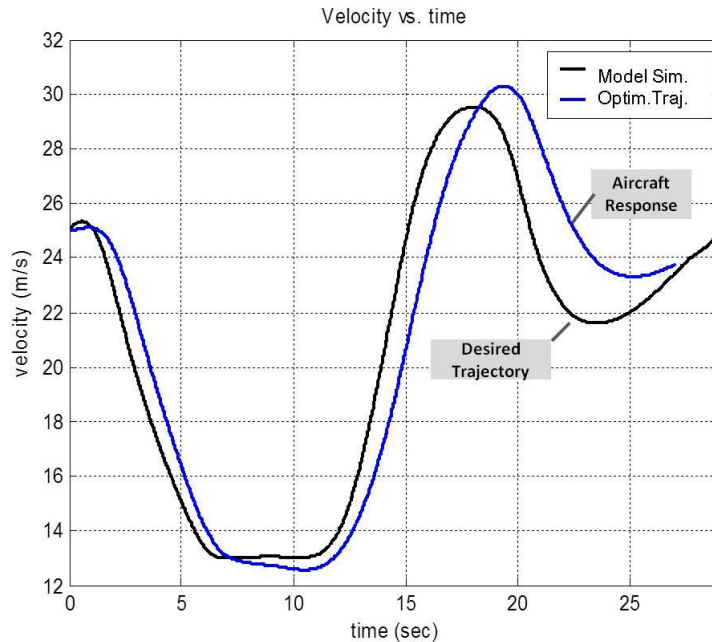


Figure 69. Velocity profile for 120-degree Turn to Next Leg using v , z , σ Simulation Inputs (From [45])

The velocity-following performance was better than the pitch angle following, as indicated by Figure 70. This shows that the velocity input is likely a good choice for inputs from the optimal trajectories to the aircraft control system. The velocity

input translates well for the throttle command control law. However, the optimal altitude inputs cannot be tracked by the pitch control law.

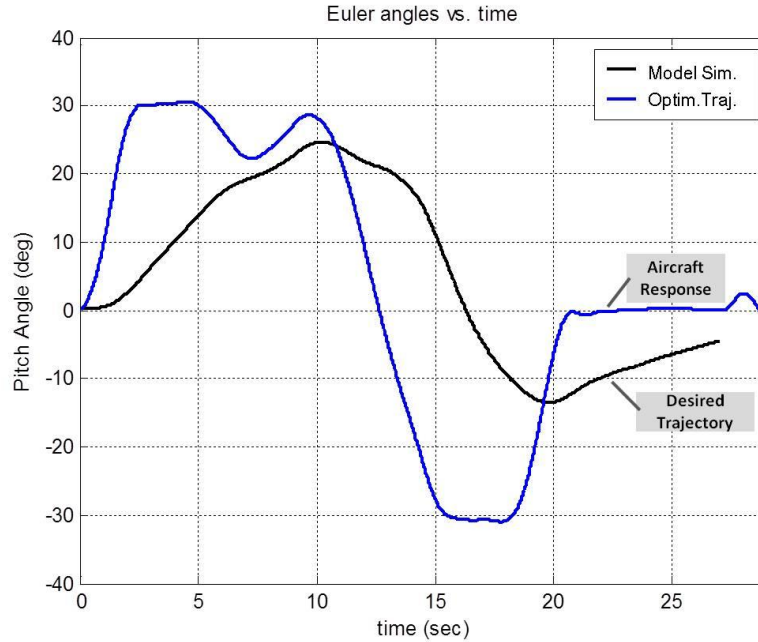


Figure 70. Pitch angle vs. Time for 120-degree Turn to Next Leg using v , z , σ as Simulation Inputs (From [45])

The pitch changes in the optimal trajectory were too steep for the model to follow, as shown in Figure 70. This indicates that improvements may need to be made to the pitch angle rate limits on the optimal control model, or that another input could be tried to improve the performance.

The bank angle-following performance shown in Figure 71 was reasonably accurate, but could also be improved in terms of both following and overshoot amounts. The maximum bank angle of 25 degrees was exceeded twice by the controller during this maneuver. The controller is not able to follow angle changes fast enough, in this case, due to poor damping leading to excessive overshoot. Nonetheless, the heading control loop seems to do a good job at reproducing the desired bank angles.

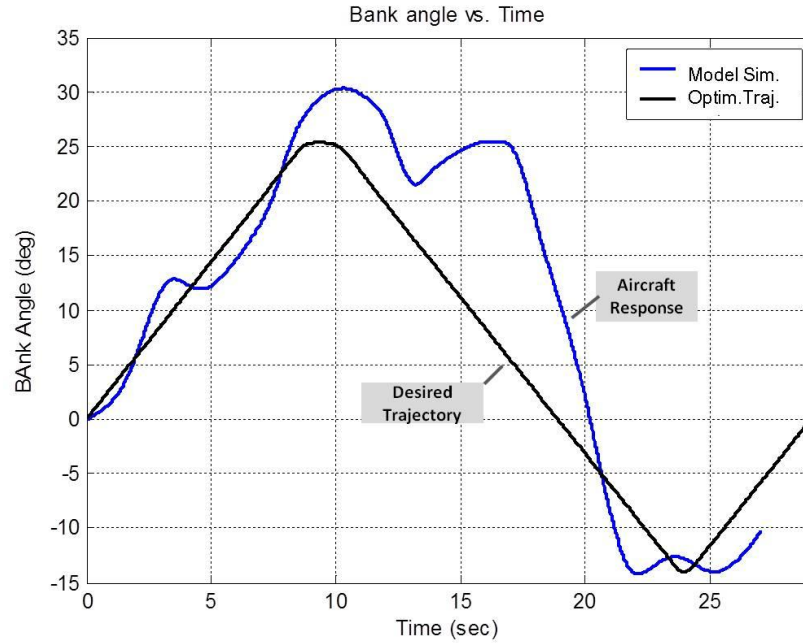


Figure 71. Bank Angle vs. Time for 120-degree Turn to Next Leg (From [45])

Overall, using the v , z , σ trajectory inputs did not produce a good result, as the changes in pitch angle were too fast for the pitch tracking loop to handle, which in turn led to poor altitude tracking [45]. Improvements would need to be made to the pitch tracking loop's speed, either via a re-write of the control laws or by directly using the already-converted pitch angle as an input to the controller.

b. Roll, Pitch and Velocity Optimal Trajectory Inputs

The second implementation used the roll and pitch values converted from the states using (5.2), plus the optimal velocity trajectory as autopilot inputs. The scheduled gains used for this simulation were [45]:

SAS Control Gain	CAS Control Gain	
$K_p = \begin{pmatrix} 2.0 \\ 0.975 \end{pmatrix} * 1.5$	$K_\phi = \begin{pmatrix} 1.32 \\ 1.1 \end{pmatrix} * 1.25$	
$K_q = \begin{pmatrix} 2.0 \\ 1.0 \end{pmatrix} * 1.5$	$K_\theta = \begin{pmatrix} 1.1 \\ 1.53 \end{pmatrix} * 2$	
$K_r = \begin{pmatrix} 2.0 \\ 1.0 \end{pmatrix} * 1$	$K_\psi = \begin{pmatrix} 0.9 \\ 1.0 \end{pmatrix} * 2$	(6.2)

AP Control Gain

$$K_v = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

$$K_h = \begin{pmatrix} 0.008 \\ 0.005 \end{pmatrix} * 1.2$$

As seen in Figure 72, this set of inputs produced a much better trajectory-following result.

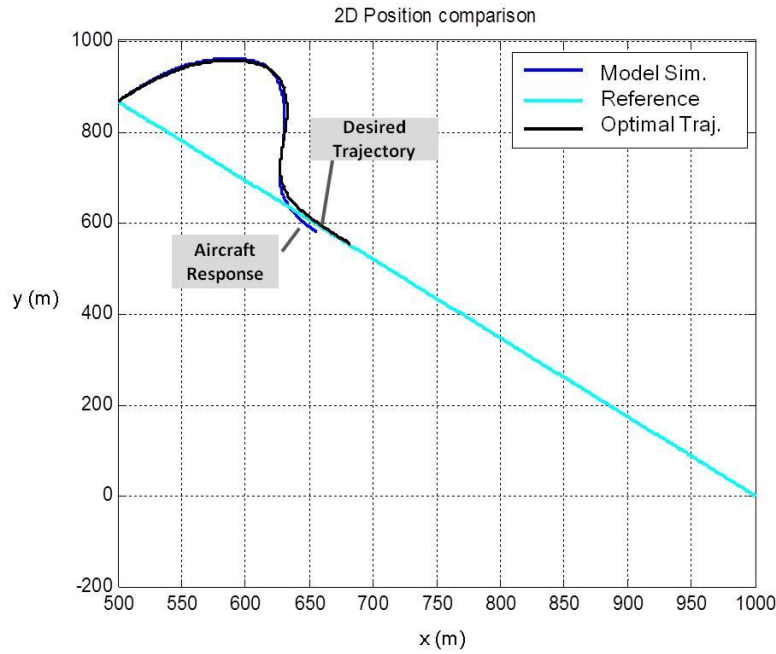


Figure 72. Top view of 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])

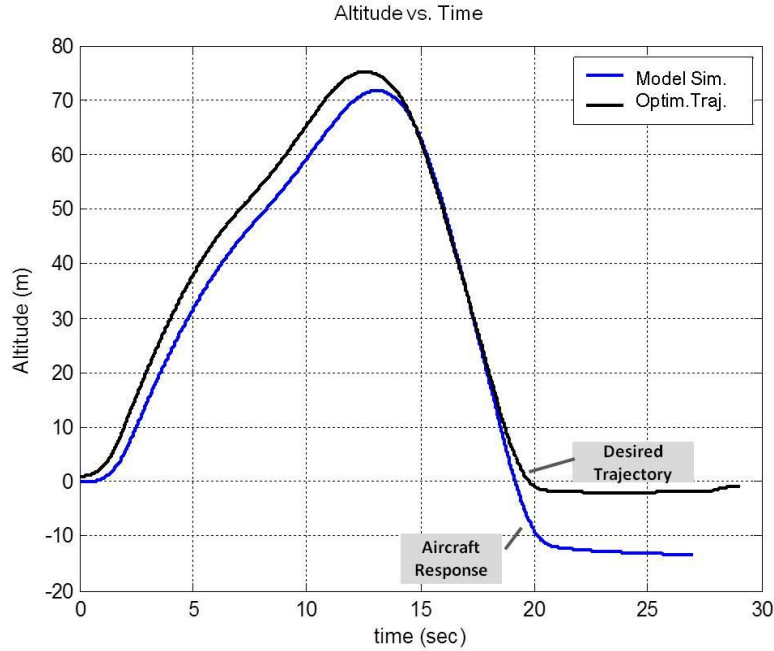


Figure 73. Altitude profile of 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])

The altitude following performance is quite reasonable but still not ideal, as shown in Figure 73. At the end of the maneuver the controller especially has difficulty maintaining the desired altitude. However, considering the range in altitude during the maneuver, the altitude error at the end is only on the order of 10-15%.

The pitch-following performance of the controller in this case is quite good, as compared to the previous case (see Figure 74). The velocity-following performance (Figure 75) is also acceptable, except in the latter half of the maneuver. Due to the issues with maintaining altitude, when the aircraft does not pull up fast enough the velocity also increases beyond the commanded value during this portion of the maneuver, as it is not climbing as steeply as commanded. One way to improve this performance might be to use thrust as the input for the throttle control law rather than velocity. Alternatively, other improvements to the controller could be tried.

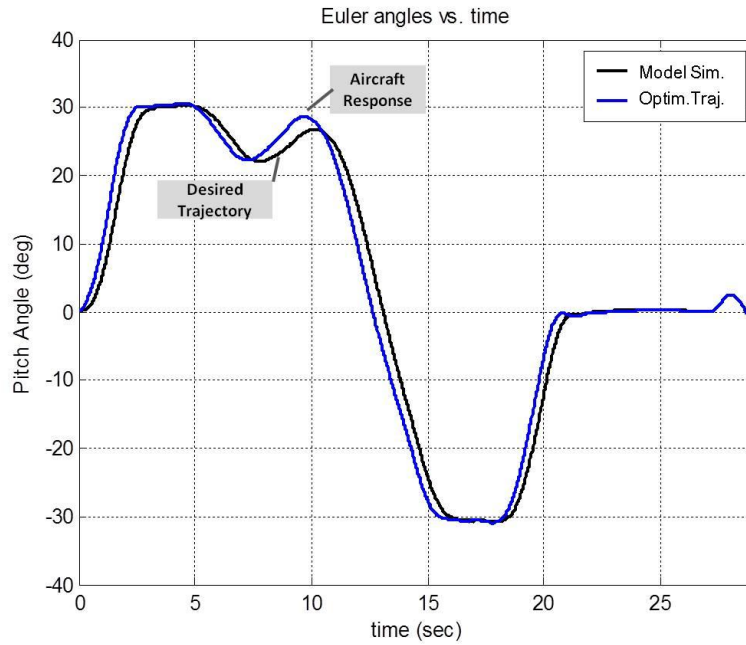


Figure 74. Pitch Angle vs. Time for with v , ϕ , θ Inputs to SIMULINK Model (From [45])

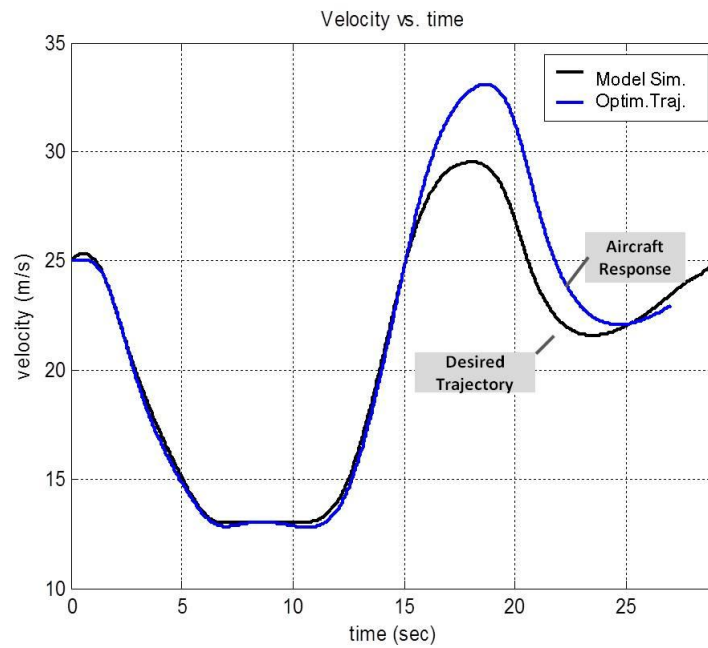


Figure 75. Velocity profile for 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])

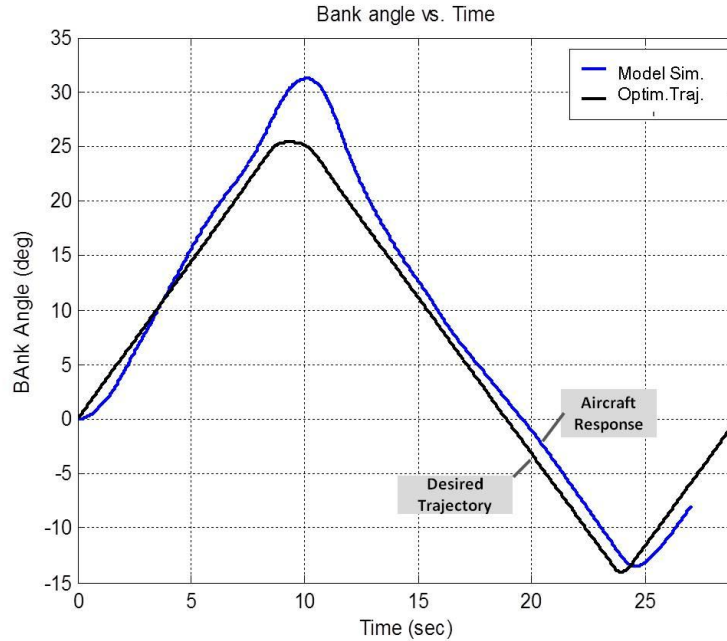


Figure 76. BA vs. time for 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])

The maximum bank angle used for the maneuver by the SIMULINK model exceeds the 25 degree bank angle limit imposed (Figure 76). This is of concern because the bank angle limit is in place for structural and safety concerns. These results indicate that another consideration for improvement of the optimization model and controller is that there may be a need for allowances in case of overshoot of safety constraints in the implementation of the trajectory—the maximum thrust, velocity and bank angle for the airframe must be included as constraints on the controller as well as the optimization model.

The large bank angle values required by the maneuver use comparatively large rudder angle commands to execute the maneuver while maintaining the sideslip suppression [45], as shown in Figure 77. The flight control law, specifically the aileron-rudder interconnects, may need to be modified to alleviate this issue.

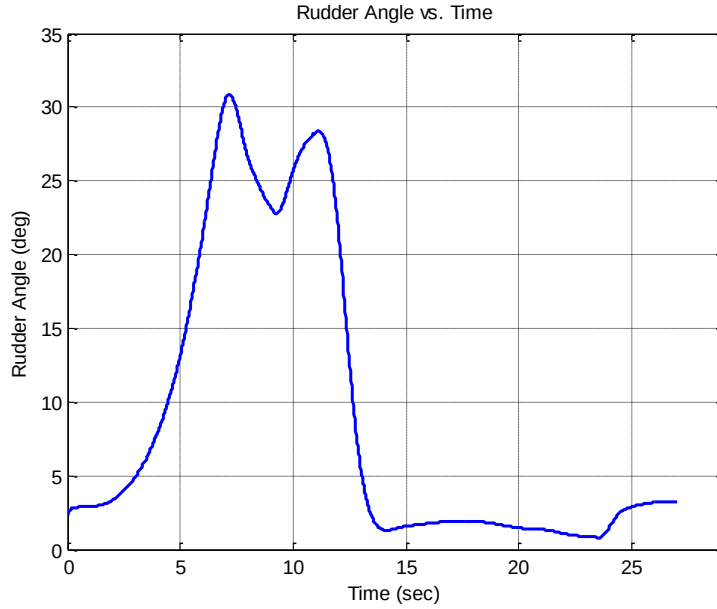


Figure 77. Rudder Angle used by controller for 120-degree Turn to Next Leg with v , ϕ , θ Inputs to SIMULINK Model (From [45])

Based on the results from the various combinations of controller inputs, it seems to be best to use the Euler Angles as inputs to the respective pitch, roll and yaw control laws whenever possible, to remove the need for any conversion within each loop. Accessing these innermost control loops would help to improve speed and following of the optimal trajectory commands. The throttle control input should be tested with thrust and velocity as inputs to determine which provides the best performance.

There are many different ways that the optimal maneuver trajectories developed in this thesis could be interfaced to the autopilot hardware. Although the combination of velocity, roll and pitch inputs seems to be most promising, the next step is to verify the results by implementing the maneuver as a HIL simulation. If the results are consistent with the findings presented in this chapter, the maneuver should next be tested in flight.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSIONS

The intent of this thesis was to determine whether the selected hardware could accurately execute optimal trajectories, and to develop a model shown to be of high enough fidelity to generate optimal trajectories without prohibitively high computation costs. Additionally, this thesis was intended to contribute useful research towards the end goal of a fully autonomous aerial vehicle in order to expand the applications available to unmanned vehicles.

The reduced-order 3DOF model with simplified conditions generates trajectories that are flyable for the MONARC system within a SIMULINK simulation. The complexity level of the model is such that optimal trajectories may be generated in a reasonable amount of time, but not yet to the point where real-time optimization is possible. A database of optimized maneuvers could be accessed in real-time, though, and used by an autonomous craft to execute optimal trajectories as commanded.

The hardware was purchased and assembled, and a 3DOF model of the MONARC system was developed. The model was used to generate a number of canonical, time-optimized trajectories. One trajectory was implemented with a 6DOF controller in SIMULINK, and shown to have a significant time improvement over the trajectory flown by a typical controller.

The kinds of canonical maneuvers designed as part of this thesis can be added together to make longer and more complex trajectories that the MONARC system is capable of flying. Iterative improvements to the optimization model and controller will be needed as flight and HIL testing are conducted. While only a flight test will fully answer the question of whether the optimal trajectories can be translated to flight success, proof of concept has been shown for the ability of the controller to receive and execute commands from the trajectory and for the time improvement of the optimal maneuvers over a typical controller. From the simulation results, the MONARC system seems to be capable of accurately executing the computer-generated optimal maneuvers, but

improvements to both the model and the controller may be needed to consistently and reliably perform the commanded maneuvers in flight.

B. FUTURE IMPROVEMENTS

While valuable, the 6DOF HIL simulations do not completely validate the autopilot or the aircraft models. Flight testing is ultimately required to validate the ability of the MONARC to execute optimal trajectories.

1. Iterative Upgrades to the Optimization Model

The optimization model will require iterative upgrades as flight test data is collected and analyzed. Adjustments will be made to the control limits and aerodynamic coefficients as applicable.

a. Control Limits

The control limits used as bounds in the optimization model need to be upgraded using flight test results to better describe the physical limitations of the aircraft. The thrust, bank angle rate and angle of attack rates will to be updated as needed to reflect the actual aircraft performance in flight tests.

b. Physical Characteristics

The final flight configuration with all hardware installed will likely show slight differences in aircraft mass and MoI than are used in the current model. The physical characteristics for the actual flight configuration will also need be updated in the model in order to generate new trajectories and iteratively improve the model to better reflect the actual aircraft.

2. Optimization of Other Costs

While this work covered only time-optimal trajectories, the optimization model can be modified to minimize or maximize for other outcomes. This may include maximizing on-station time or coverage of a certain assigned area, or minimizing fuel

consumption or distance traveled. This would greatly increase the applications for which the model and the aircraft may be used. The possibilities are endless.

C. FLIGHT IMPLEMENTATION AND TESTING

Flight tests are planned at McMillan Air Field at the Camp Roberts Army National Guard Base near Paso Robles, California. The tests will be conducted in military airspace and meet NAVAIR and NPS requirements for safety and procedures. A qualified safety pilot will fly the MONARC along with an experienced autopilot and ground station operator.



Figure 78. Camp Roberts Airspace (From [40])

The Airspace for Camp Roberts is highlighted in Figure 78. The airstrip area is boxed, and the airspace is outlined.



Figure 79. McMillan Airfield at Camp Roberts (From [40])

Figure 79 shows a closer view of the box marked in Figure 78, with an overhead view of the airstrip.

Early flight tests of the MONARC should consist of collecting data to improve model coefficients of lift and drag, doublet commands to better determine control limits, and confirming the inertial properties. Once several demonstration flights have been conducted, collected data can be used to improve the optimization model, and new optimal trajectories generated. The most important test can then be performed—actual flight of the optimal trajectories by the MONARC aircraft, as piloted autonomously. That test will show with certainty whether the optimization model is viable and how much time improvement can be achieved going from using a generic control algorithm to fly a maneuver to using one of the generated optimized maneuvers.

The canonical maneuvers should be flown to verify the ability of MONARC to perform simple and complex maneuvers, and the equilateral triangle maneuver and its variations can be tested, as well as many similar maneuvers. Some interesting tests might include standard Navy search patterns versus a trajectory optimized for area coverage or minimizing fuel use for planes returning to base. The applications and variety of maneuvers possible for the MONARC are myriad, and the aircraft could be a very useful tool for future development of fully autonomous aircraft capable of maneuvers custom-optimized for a variety of missions.

LIST OF REFERENCES

- [1] E. Bone and C. Bolkcom, “Unmanned Aerial Vehicles: Background and Issues for Congress,” Report to Congress, Order Code RL31872, Apr. 2003.
- [2] N.P. Curtin and P.L. Francis, “Unmanned Aerial Vehicles—Major Management Issues Facing DOD’s Development and Fielding Efforts.” GAO Testimony before the Subcommittee on Tactical Air and Land Forces, Committee on Armed Services, House of Representatives, GAO-04-530T, Mar. 2004.
- [3] *Unmanned Aerial Vehicles Issues Paper*, Civil Aviation Authority of New Zealand, Jan. 2007.
- [4] J. R. Wilson, (2003, Jun.). “UAVS: A Worldwide Roundup—More and More Countries are Developing or Cooperating on UAVs as Their Numbers and Versatility Grow.” *Aerospace America* [Online]. Available: <http://www1.aiaa.org/aerospace/Article.cfm?issuetocid=365&ArchiveIssueID=39>].
- [5] R. Schwing, “Unmanned aerial vehicles—Revolutionary tools in war and peace.” M.A. thesis, Strategic Studies Dept., Army War College, Carlisle Barracks, PA, 2007.
- [6] “Nonproliferation: Agencies Could Improve Information Sharing and End-User Monitoring on Unmanned Aerial Vehicle Exports.” GAO Report to the Ranking Member, Subcommittee on National Security, Homeland Defense, and Foreign Operations, Committee on Oversight and Government Reform, House of Representatives, GAO-12-536, Jul. 2012.
- [7] I.M. Ross and M. Karpenko. (2012, Oct.). A review of pseudospectral optimal control: from theory to flight. *Annual Reviews in Controls*, vol. 36, pp. 182-197.
- [8] K. Bollino *et al*, “A New Approach to Intelligent Path Planning: Principles, Applications and Examples,” paper presented at the *AIAA Infotech@Aerospace Workshop, Conference and Exhibit*, May 2007.
- [9] J. Cascio *et al*, “Smooth Proximity Computation for Collision-Free Optimal Control of Multiple Robotic Manipulators,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, Oct. 2009, pp. 2452-2457.
- [10] M.A. Hurni *et al*, “A Pseudospectral Optimal Motion Planner for Autonomous Unmanned Vehicles,” *Proceedings of the 2010 American Control Conference*, Baltimore, MD, Jun. 2010, pp. 1591-1598.

- [11] M.A. Hurni, P. Sekhavat, and I.M. Ross, "An Info-Centric Trajectory Planner for Unmanned Ground Vehicles," *Dynamics of Information Systems: Theory and Applications*, Springer Optimization and its Applications, 2010, pp. 213-232.
- [12] F. Fahroo and I.M. Ross, "Advances in Pseudospectral Methods for Optimal Control," paper presented at the *AIAA Guidance, Navigation, and Control Conference*, Honolulu, HI, Aug. 2008.
- [13] NASA. (2012, Aug.). "Mars Science Laboratory Landing Press Kit." *Press Kit* [Online]. Available: http://www.jpl.nasa.gov/news/fact_sheets/mars-science-laboratory.pdf.
- [14] J. Nalepka and M. Duquette, "A Multi-Purpose Simulation Environment for UAV Research," presented at the *AIAA Modeling and Sim. Tech. Conf. and Exhibit*, Austin, TX, Aug. 2003.
- [15] F. Roberts and T. Ferrazano, "Multiple Unmanned Aerial Vehicle (UAV) Experimentation: Forward Look, presented at the *AIAA 3rd Unmanned Unlimited Tech. Conf., Workshop and Exhibit*, Sep. 2004.
- [16] J. Lisein *et al.*, "Development of an Original Aerial-Based Inventory Method: First Steps Towards the Use of Mini Unmanned Aerial Vehicles in Elephant Inventory," *Unit of Forest and Nature Mgmt.*, Universite de Liege Gembloux Agro-Bio Tech, Liege, Belgium, 2012.
- [17] A. Lucas *et al.*, "Development & Flight Testing of an Intelligent, Autonomous UAV Capabiltym," presented at the *AIAA 3rd Unmanned Unlimited Tech. Conf., Workshop and Exhibit*, Sep. 2004.
- [18] F. Haro and M. Torres, "A comparison of path planning algorithms for omnidirectional robots in dynamic environments," presented at the *IEEE 3rd Latin American Robotics Symposium*, Santiago, Chile, Oct.2006.
- [19] E. Magid and E. Rivlin, "CautiousBug: A competitive algorithm for sensory based robot navigation," *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep. 28–Oct. 2, 2004.
- [20] Q. Zhu, Y. Yan, and Z. Xing, "Robot path planning based on artificial potential field approach with simulated annealing," *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2, Oct. 2006.
- [21] M. Wzorek and P. Doherty, "Reconfigurable path planning for an autonomous unmanned aerial vehicle," presented at the *2006 International Conference on Hybrid Information Technology*, Cheju Island, Korea, Nov. 2006.

- [22] D. Zhu and J.C. Latombe, "New heuristic algorithms for efficient hierarchical path planning," *IEEE Transactions on Robotics and Automation*, vol. 7, No. 1, Feb.1991.
- [23] T. K. Bateman, "Design and laboratory implementation of autonomous optimal motion planning for non-holonomic planetary rovers," M.S. thesis, Dept. of Mech. and Astronautical Eng., Naval Postgraduate School, Monterey, CA, 2012.
- [24] *Multiplex USA* [Online]. Available: <http://www.multiplexusa.com/>.
- [25] *M213228-Mentor Manual*, MULTIPLEX Modellsport GmbH & Co.KG, Germany, 2008.
- [26] M. O. Rauw (2001, May). *FDC 1.2—A SIMULINK Toolbox for Flight Dynamics and Control Analysis* [Online]. Available: www.dutchroll.com.
- [27] K. H. Lee, "ESEP Research Overview," Nov. 2012, unpublished presentation.
- [28] K. P. Bollino, "High-fidelity real-time trajectory optimization for reusable launch vehicles," PhD. dissertation, Dept. of Mech. and Astronautical Eng., Naval Postgraduate School, Monterey, CA, 2006.
- [29] K. H. Lee, "SAS & CAS Design of Mentor UAV—Mentor Inner Loop Control Law Definitions," unpublished technical report.
- [30] B. Stevens and F. Lewis, *Aircraft Control and Simulation*. Hoboken, NJ: John Wiley & Sons, Inc., 2003.
- [31] M. Bittner *et al.*, "A Multi-Model Gauss Pseudospectral Optimization Method for Aircraft Trajectories," in *AIAA Atmospheric Flight Mechanics Conf.*, Minneapolis, MN, 2012.
- [32] K. H. Lee, "MONARC 6DOF SIMULINK Model and MATLAB Code," unpublished technical report.
- [33] I.M. Ross. *A Primer on Pontryagin's Principle in Optimal Control*. Carmel,CA: Collegiate Publishers, 2009.
- [34] I.M. Ross and M. Karpenko. *AE 4850: Astrodynamic Optimization Course Notes*, Dept. of Mech. And Astronautical Eng., Naval Postgraduate School, Monterey, CA, 2011.
- [35] I.M. Ross, "A Beginner's Guide to DIDO—A MATLAB Application Package for Solving Optimal Control Problems," Elissar, LLC, Tech. Rep. TR-711, Monterey, CA, 2007.

- [36] B. R. Carter, "Time-optimization of high performance combat maneuvers," M.S. thesis, Dept. of Mech. And Astronautical Eng., Naval Postgraduate School, Monterey, CA, 2006.
- [37] M.I. Lizarraga-Fernandez, "Design, implementation and flight verification of a versatile and rapidly reconfigurable research platform," PhD. Dissertation, Dept. of Comp. Eng., Univ. of California, Santa Cruz, 2009.
- [38] R.T. Casey, private communication, Nov, 2012.
- [39] R. T. Casey, private communication, Oct., 2012.
- [40] *QGroundControl—Ground Control Station for Small Air-Land-Water Autonomous Unmanned Systems* [Online]. Available: <http://qgroundcontrol.org/>.
- [41] R.T. Casey, private communication, Sep., 2012.
- [42] R.T. Casey, "Range Safety Review Questions for UAV Projects," unpublished.
- [43] D. Jung and P. Tsiotras, "Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle," Georgia Inst. of Tech., Atlanta, GA, 2006.
- [44] J. Kalviste. (1987). Spherical mapping and analysis of aircraft angles for maneuvering. *Journal of Aircraft*, vol. 24, pp. 523-520.
- [45] K. H. Lee, "Matching Manual Simulation Results," Nov. 2012, unpublished technical report.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Mark Karpenko
Naval Postgraduate School
Monterey, California
4. I. Michael Ross
Naval Postgraduate School
Monterey, California