



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**OPTIMAL AND NON-COMMITTAL GUIDANCE
OF AN AIR-TO-AIR MISSILE**

by

William B. Thames III

December 2023

Thesis Advisor:
Second Reader:

Mark Karpenko
Sean P. Kragelund

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 2023	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE OPTIMAL AND NON-COMMITTAL GUIDANCE OF AN AIR-TO-AIR MISSILE			5. FUNDING NUMBERS	
6. AUTHOR(S) William B. Thames III				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) When military units need new capability, hardware is often more than 5–7 years out. While a new weapon development may take many years, often software improving performance of current hardware can be developed, tested, and fielded in significantly less time and at less cost. The goal of this thesis is to increase Air-to-Air (A-A) weapon range and lethality by a two-prong approach. Optimal trajectories were generated and applied to the guidance of an A-A weapon in a 6 Degree of Freedom (DOF) simulation. Second, a concept for non-committal guidance (NCG) was developed by mixing traditional pursuit guidance and proportional navigation to allow for lethal endgames, while minimizing missile maneuvers in response to target maneuvers. NCG is logic that minimizes maneuvering in response to target maneuvering for as long as possible. This increases energy available at endgame and minimizes the chance of an overshoot. These two concepts were analyzed for a short range, medium range, and long-range air breathing missiles via simulation. Trajectory solutions were evaluated using the 6 DOF simulations to ensure their reliability, repeatability, and optimality. Compared to proportional navigation with a constant lofting angle, the proposed NCG scheme with an optimized lofting profile was shown to increase weapon range and weapon lethality against a maneuvering target.				
14. SUBJECT TERMS non-committal guidance, optimal lofting, air-to-air, DIDO, modeling and simulation, Pontryagin's principle, missile			15. NUMBER OF PAGES 139	
			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	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

OPTIMAL AND NON-COMMITTAL GUIDANCE OF AN AIR-TO-AIR MISSILE

William B. Thames III
Civilian, Department of the Navy
BS, Mississippi State University, 2019

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
December 2023**

Approved by: Mark Karpenko
Advisor

Sean P. Kragelund
Second Reader

Brian S. Bingham
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

When military units need new capability, hardware is often more than 5–7 years out. While a new weapon development may take many years, often software improving performance of current hardware can be developed, tested, and fielded in significantly less time and at less cost. The goal of this thesis is to increase Air-to-Air (A-A) weapon range and lethality by a two-prong approach. Optimal trajectories were generated and applied to the guidance of an A-A weapon in a 6 Degree of Freedom (DOF) simulation. Second, a concept for non-committal guidance (NCG) was developed by mixing traditional pursuit guidance and proportional navigation to allow for lethal endgames, while minimizing missile maneuvers in response to target maneuvers. NCG is logic that minimizes maneuvering in response to target maneuvering for as long as possible. This increases energy available at endgame and minimizes the chance of an overshoot. These two concepts were analyzed for a short range, medium range, and long-range air breathing missiles via simulation. Trajectory solutions were evaluated using the 6 DOF simulations to ensure their reliability, repeatability, and optimality. Compared to proportional navigation with a constant lofting angle, the proposed NCG scheme with an optimized lofting profile was shown to increase weapon range and weapon lethality against a maneuvering target.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	OVERVIEW	2
1.	Terminology and Jargon	2
2.	A Brief History of Air Combat Technology	7
3.	A Brief Overview of Missile Guidance.....	7
4.	Lofted Trajectories	11
5.	Desired Engagement Geometry	11
B.	BACKGROUND	12
1.	Non-Committal Guidance	12
2.	Optimal Control Theory.....	13
C.	MOTIVATIONS	14
D.	CONTRIBUTIONS.....	14
E.	THESIS ORGANIZATION.....	15
II.	MATHEMATICAL BACKGROUND ON OPTIMIZATION AND SIMULATION	17
A.	PONTRYAGIN'S PRINCIPLE.....	17
1.	Terminology and Variables.....	17
2.	States and Costates	18
3.	Paths and Control Constraints	19
4.	Hamiltonian and Endpoint Lagrangian	19
5.	Evaluating Optimality Conditions	21
6.	Example: Constrained Brachistochrone Problem	22
B.	MACHINE LEARNING METHODS.....	28
1.	Neural Networks	28
2.	Training Methods.....	30
C.	3DOF AND 6DOF DYNAMICAL ENVIRONMENT.....	32
1.	Coordinate Systems	32
2.	Quaternions vs. Euler Angles.....	34
3.	Vertical 3DOF Dynamics Environment.....	35
4.	Horizontal 3DOF Kinematics Environment.....	36
5.	6DOF Dynamics Environment.....	36
D.	SUMMARY	38
III.	METHODOLOGY	39
A.	MISSILE DESIGNS	39
1.	Propulsion.....	40
2.	Aerodynamic Forces	40

3.	Native GNC.....	42
B.	M&S ENVIRONMENTS	43
1.	3DOF Vertical Plane.....	44
2.	3DOF Horizontal Plane	45
3.	6DOF M&S Environment	46
C.	TRAJECTORY OPTIMIZATION	51
1.	Cost Functional	51
2.	Application of Pontryagin’s Principle.....	52
D.	TARGET GENERATION	56
1.	Control Set Engagements	57
2.	Complex Set Engagements	57
3.	Long Range Missile Target Set.....	61
E.	SIMULATION INTEGRATION	62
1.	Implementation of Optimal Vertical 3DOF Trajectories into 6DOF	62
2.	Implementation of Optimal Horizontal 3DOF Trajectories into 6DOF.....	65
F.	NON-COMMITTAL GUIDANCE OF A LONG-RANGE MISSILE	67
G.	SUMMARY	69
IV.	RESULTS	71
A.	SHORT-RANGE MISSILE RESULTS.....	71
1.	Short-Range Missile Control Set Results.....	71
2.	Short-Range Missile Complex Set Results.....	73
B.	MEDIUM-RANGE MISSILE RESULTS	77
1.	Medium-Range Missile Control Set Results.....	77
2.	Medium-Range Missile Complex Set Results.....	78
C.	LONG RANGE MISSILE SCENARIO.....	83
1.	Long Range Target 1	84
2.	Long Range Target 2	85
3.	Long Range Target 3	86
D.	SUPPLEMENTAL ANALYSIS	88
1.	Comparative Analysis.....	88
2.	Run Time and Reliability	89
3.	Interpolants vs. Individual Optimized Profiles	92
E.	SUMMARY	95
V.	CONCLUSIONS	97
A.	ISSUES ENCOUNTERED AND POSSIBLE SOLUTIONS	97

1.	Optimizing Trajectory of 6DOF Dynamics	97
2.	Limitations of MATLAB ODE Solvers.....	98
B.	RECOMMENDATIONS FOR FUTURE WORK.....	99
1.	Optimization Improvements	99
2.	Endgame Improvements	100
3.	Other Areas of Interest.....	100
APPENDIX A. TEST CASE DEFINITIONS.....		101
APPENDIX B. SHORT RANGE MISSILE RESULTS.....		105
APPENDIX C. MEDIUM RANGE MISSILE RESULTS		109
LIST OF REFERENCES.....		113
INITIAL DISTRIBUTION LIST		117

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1.	Component Locations on Air-to-Air Missiles: (a) Python 5. Source: [5], (b) Air Intercept Missile (AIM)-120. Source: [6], (c) AIM-7. Source: [7].....	4
Figure 2.	A-Pole and F-Pole Illustration.	6
Figure 3.	Illustration of Pursuit Guidance 3DOF Engagement: (a) Missile and Target Positions, (b) LOS Angle.	8
Figure 4.	Proportional Navigation 3DOF Engagement: (a) Missile & Target Positions, (b) LOS Angle Rate.....	9
Figure 5.	Non-Committal Guidance 3DOF Engagement: (a) Missile and Target Positions, (b) LOS Angle.	10
Figure 6.	Lofted Trajectory vs. Non-Lofted Trajectory: (a) Missile Velocity, (b) Missile Altitude.	11
Figure 7.	Aspect Angle Definition.	12
Figure 8.	Constrained Brachistochrone “Keep-Out Zone” Constraint.....	23
Figure 9.	DIDO Solution of Constrained Brachistochrone Problem: (a) Optimal Position Trajectory, (b) Optimal Control Trajectory, (c) Hamiltonian vs. Time, (d) Velocity Costate vs. Time.	26
Figure 10.	Constrained Brachistochrone Feasibility Check: (a) Propagated Position, (b) Interpolated Control Input.....	27
Figure 11.	Single Input Neuron Layout.....	29
Figure 12.	Four-Input, Four-Neuron, Single-Layer Neural Network.....	29
Figure 13.	Sample Objective Function Surface.....	31
Figure 14.	Vehicle Reference Frames: (a) Body-Fixed Reference Frame, (b) Wind Reference Frame.	33
Figure 15.	Missile Drag Curve Comparison: (a) Short/Medium Range Weapon Drag Curve, (b) Long Range Weapon Drag Curve.	41
Figure 16.	Fin Configuration Comparison: (a) Cruciform Fin Configuration, (b) X Fin Configuration.	41

Figure 17.	Long Range Missile Flyout Trajectory: (a) Range vs. Altitude, (b) Time vs. Mach, (c) Body Acceleration.	42
Figure 18.	Long Range Missile Cruise Trajectory: (a) Range vs. Altitude, (b) Time vs. Mach, (c) Body Acceleration.	43
Figure 19.	Density vs. Altitude.	44
Figure 20.	Error vs. Altitude.	45
Figure 21.	FlyOut6 and ODESET Flowchart.	46
Figure 22.	Drag Coefficient vs. Angle of Attack.	49
Figure 23.	Lift Coefficient vs. Angle of Attack Using a Neural Network Fit.	50
Figure 24.	Pitching Moment Coefficient vs. Angle of Attack Using a Neural Network Fit.	50
Figure 25.	Cost Functional 1: Without Control Effort Function (left) vs. Cost Functional 2: With Control Effort Function (right).	52
Figure 26.	Complex Target 1: High Aspect Maneuvering Target.	58
Figure 27.	Complex Target 2: Constantly Maneuvering Target.	59
Figure 28.	Complex Target 3: Run Down Short-Range Missile.	60
Figure 29.	Complex Target 3: Run Down Medium-Range Missile.	60
Figure 30.	Complex Target 4: High, Fast Flier.	61
Figure 31.	3DOF Trajectory Optimization Integration to 6DOF Flyout.	63
Figure 32.	Commanded vs. Achieved Altitude.	64
Figure 33.	Commanded vs. Achieved Pitch Angles.	65
Figure 34.	3DOF Trajectory Optimization Integration into 6DOF.	66
Figure 35.	Optimal Trajectory vs. 6DOF FlyOut.	67
Figure 36.	Long Range Missile Flyout Concept.	68
Figure 37.	Short-Range Missile Flyout vs. Control Target.	73
Figure 38.	Short Range Missile vs. Complex Set Target 1 Trajectory.	75

Figure 39.	SRM Complex Target 2 Altitude and Velocity: (a) Missile Altitude, (b) Missile Velocity.	77
Figure 40.	Complex Target 1 – Scenario 1 Medium Range Missile Trajectory Comparison.	80
Figure 41.	Complex Target 1 – Scenario 1 Medium Range Missile Velocity Comparison.	80
Figure 42.	Complex Target Set – Scenario 1 Medium Range Missile Altitude Comparison.	81
Figure 43.	Long Range Target 1 Results at Handover to Terminal Guidance.	85
Figure 44.	Long Range Target 2 Results at Handover to Terminal Guidance.	86
Figure 45.	Long Range Target 3 Results at Handover to Terminal Guidance.	87
Figure 46.	ProNav vs. Non-Committal Guidance Monte Carlo Results.	89
Figure 47.	Short-Range Missile Runtime Analysis.	91
Figure 48.	Medium-Range Missile Runtime Analysis.	91
Figure 49.	Short Range Missile Interpolant vs. DIDO Errors.	93
Figure 50.	Medium Range Missile Interpolant vs. DIDO Errors.	94
Figure 51.	Three Loop Missile Autopilot.	98

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Symbology of Pontryagin’s Principle	18
Table 2.	Constrained Brachistochrone Optimality Conditions	25
Table 3.	Constrained Brachistochrone Results	26
Table 4.	Constrained Brachistochrone Problem Feasibility Test.....	27
Table 5.	Common Aerospace Coordinate Systems Adapted from [15].....	32
Table 6.	Vehicle Reference Frames (adapted from [15]).....	33
Table 7.	Missile Configurations.....	39
Table 8.	3DOF Vertical Plane Optimality Conditions.....	54
Table 9.	3DOF Horizontal Plane Optimality Conditions.....	56
Table 10.	Long Range Missile Target Set.....	62
Table 11.	Error and Drift Rates for Horizontal 3DOF	67
Table 12.	Short-Range Missile Results for the Control Set.....	71
Table 13.	Short Range Missile Complex Set Target 1 Results.....	74
Table 14.	Short-Range Missile Complex Target Set 2	76
Table 15.	Medium-Range Missile Results vs. Control Set.....	78
Table 16.	Medium Range Missile Complex Set Target 1 Results	79
Table 17.	Medium Range Missile Complex Set Target 2 Results	82
Table 18.	Medium Range Missile Complex Set Target 3 Results	82
Table 19.	Medium Range Missile Complex Set Target 4 Results	83
Table 20.	Long Range Target 1 Results.....	84
Table 21.	Long Range Target 2 Results.....	86
Table 22.	Long Range Target 3 Results.....	87
Table 23.	Specifications on Laptop Used to Generate Data	90

Table 24.	MRM Optimized vs. Non-Optimized Percent Difference in Launch and A-Pole Ranges.....	95
-----------	--	----

LIST OF ACRONYMS AND ABBREVIATIONS

A-A	Air-to-Air
A-G	Air-to-Ground
AI	Artificial Intelligence
AIM	Air Intercept Missile
AMRAAM	Advanced Medium Range Air-to-Air Missile
CG	Center of Gravity
DCM	Direction Cosine Matrix
DOF	Degree of Freedom
ECEF	Earth Centered Earth Fixed
FOV	Field of View
GNC	Guidance, Navigation, and Control
IFS	Integrated Flight Simulation
IR	Infrared
ISA	International Standard Atmosphere
Kg	Guidance Gain
KKT	Karush-Kuhn-Tucker
LOS	Line of Sight
M&S	Modeling and Simulation
nm	Nautical Mile
PG	Pursuit Guidance
PN	Proportional Navigation
T_i	Time to Intercept
V&V	Verification and Validation
V_i	Velocity at Intercept

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I am grateful to all who have helped me along the way with this process, especially to Dr. Karpenko as without his technical guidance, none of this would have happened. Dr. Kragelund for his help in the preparation of this document. Tyler Trombetta as a co-developer of the FlyOut6 code, and a significant technical help in the groundwork of this thesis. All of the professors involved in the 299-certificate process for having the patience to teach me while I was working full time. Christian Proctor for being a relentless advocate for my career and arranging the opportunity to attend NPS. The rest of the members of my team, both locally at China Lake and at Patuxent River.

Last but certainly not least, my wife, Megan, for having exceptional patience during many long nights, and stressful days.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In military applications, getting approval to field new hardware is a long and difficult process, while software can be fielded much faster. Thus, achieving maximum performance out of current-generation hardware is often a critical area of focus. This is, in a way, an optimization problem: how does one “squeeze every drop” of performance out of a fielded system?

This thesis focuses on how optimal control theory and a new concept of non-committal guidance (NCG) can be implemented into Air-to-Air (A-A) weaponry, in a practical manner. Optimal control theory is the field of mathematics and engineering that deals with minimizing a cost function for controlling a defined dynamical system. Non-committal guidance will be defined as any guidance method that delays committal guidance decisions, to preserve the total energy available to a flight vehicle. The goal of this thesis is to present a way that optimal control theory can be applied to create near-optimal trajectories (referred to as optimal trajectory planning), and that use non-committal concepts to improve missile performance.

Unlike traditional guidance methods, most (but not all) of the solution methods for optimal control problems utilize nonlinear programming to search for the minima [1]. Tedrake describes nonlinear programming as follows: “Nonlinear programming involves minimizing or maximizing a nonlinear objective function subject to bound constraints, linear, or nonlinear constraints, where the constraints can be inequalities or equalities” [2]. No piece of software is 100% reliable, and optimal control solvers are no exception. With traditional guidance methods, such as proportional navigation, a continuous solution can be calculated for all rational inputs, while with any nonlinear programming solution any failure to converge could lead to a non-existent guidance solution, without which it is not possible to control the vehicle. In addition, nonlinear programming methods often take tactically significant amounts of time to converge (5 to 30 seconds). During the convergence process, the weapon may not be able to be launched at a target. With a rate of closure in excess of 1,000 knots, the lower end of the bound of this time of flight may not be problematic, but the upper end of this time bound is highly likely to cause problems.

Both of these challenges will be explored later in the thesis, along with ways to mitigate them.

The goal of this thesis is to create an optimal and non-committal trajectory planning algorithm in an effort to extend the effective range of an A-A missile, using a software only change. A conventional proportional navigation (PN) guidance [3] utilizing a lofted trajectory is presented as a reasonable standard to compare the new approach to. The variable to be optimized is Time to Intercept (t_i) and/or control effort (u). The stated goal is achieved through the application of Pontryagin's Principle.

Pontryagin's Principle is a mathematical principle that provides necessary conditions that any optimal trajectory must satisfy. Using an optimal control toolbox, such as DIDO [3], optimal trajectories may rapidly be formulated for a variety of problems. Pontryagin's Principle will be covered in detail in Chapter II but will be summarized here. Pontryagin's Principle states that optimal control of a dynamic system can be achieved by what Ross states is the "instantaneous minimization of the Hamiltonian function" [3], which is another mathematical construct that will be discussed in following chapters.

A. OVERVIEW

This section is intended to introduce the reader to some key concepts. Air combat is littered with complex and strange terminology. In addition, missile guidance is a complex and generally unfamiliar subject for most readers. This section will cover common missile guidance methods and some methods that are commonly used to increase kinematic range.

1. Terminology and Jargon

Historically, air combat has been plagued by an exceptionally large amount of specific terminology that is entirely unfamiliar to an outsider. The purpose of this section is to describe a significant amount of terminology (and jargon) that this thesis uses extensively.

a. Common Components of an Air-to-Air Missile

While A-A missiles vary wildly in hardware configuration, several components can be found on most missiles. Some of the applicable components relevant to this thesis are the seeker, warhead, propulsion, guidance computer, and control system. Definitions for each of the above components are included below (adapted from [4]):

- Control System – The control system is defined as the section of the missile that executes guidance commands and provides the forces and moments that steer the missile. This is accomplished by using an autopilot loop to drive the physical control surfaces. Most often these control surfaces are fins, however thrust vectoring or even more exotic methods are occasionally seen.
- Guidance Computer – The guidance computer is the section of the missile that takes seeker input and generates a trajectory towards the target. It also will be used to cover the control section that executes these commands.
- Propulsion – All missiles have a propulsion system, whether it be a solid rocket motor, liquid rocket motor, or air-breathing motor. This differentiates a missile from a gravity weapon that is not forward firing.
- Seeker – The seeker is typically mounted to the nose of the missile and searches a region in space for an appropriate target. Once the target is located, the seeker will provide the target data to the guidance computer. Depending on the type of seeker, the seeker may provide an elevation, azimuth, and range or just elevation and azimuth.
- Warhead – The warhead is an explosive charge, typically with a specially designed fragmentation pattern that is triggered either by proximity or impact with the target.

With few exceptions, most if not all of these components are seen on every A-A missile. Some of these components' locations are obvious, such as the seeker, propulsion, and control surface locations. Components such as guidance computer and warhead, are

not always readily apparent. A few missiles with the relevant components are shown in Figure 1.

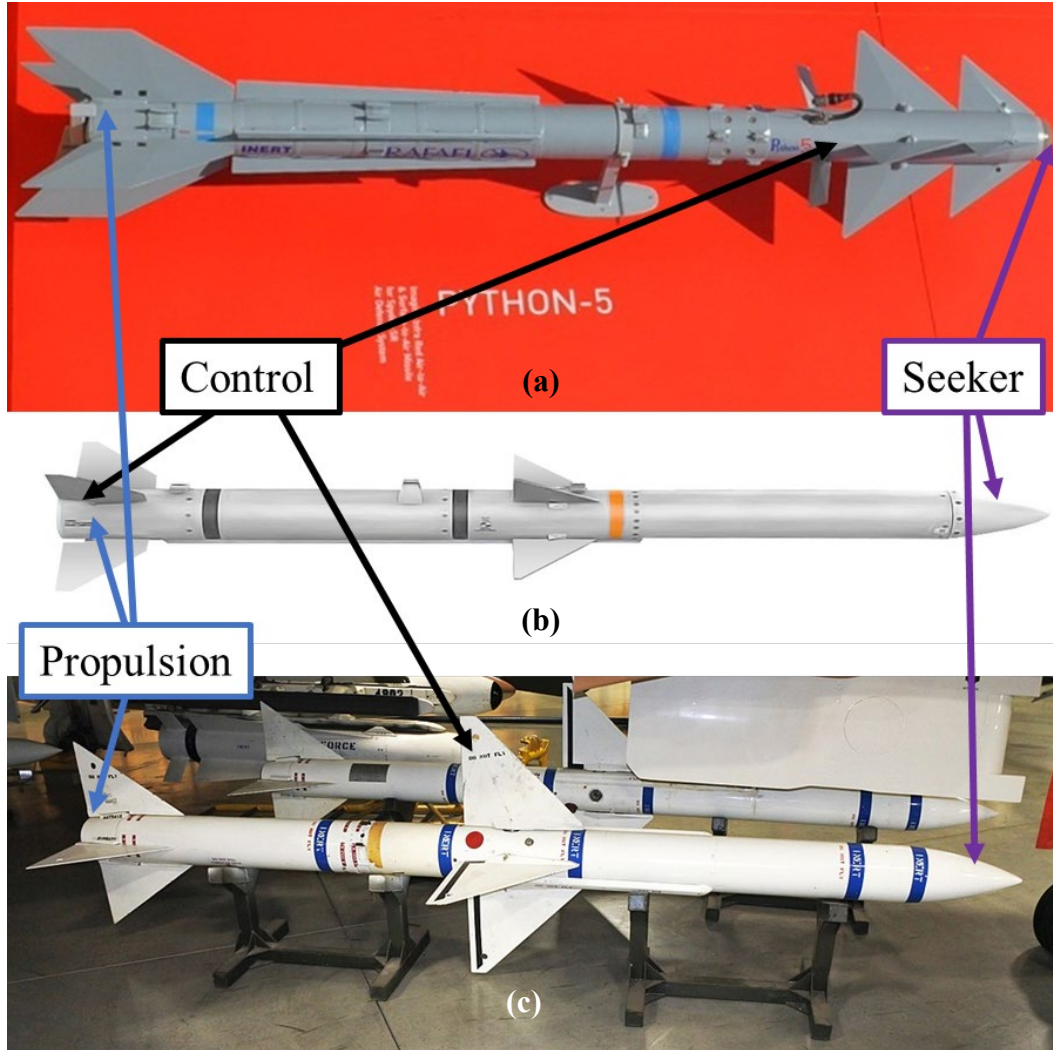


Figure 1. Component Locations on Air-to-Air Missiles: (a) Python 5. Source: [5], (b) Air Intercept Missile (AIM)-120. Source: [6], (c) AIM-7. Source: [7].

b. Missile Flight Terminology

There are several specific terms that will be used to describe the stages of missile flyouts in this thesis. The overall flight from beginning to end will be referred to as the “flyout.” During a flyout there are several stages. The period before the missile has been

launched is referred to as “pre-launch.” During this stage, the launching platform may pass targeting information to the weapon. When the launch command is sent, a motor, typically a solid rocket, begins producing thrust. This stage is referred to as “boosting.” Some weapons may have a second, weaker motor that maintains airspeed. If present, this secondary thrust stage is referred to as “sustain.”

Should the target be outside of seeker detection range, then the missile will enter a “searching” state. During this mode, the missile will fly towards the last known target location and search for the target. Depending on the design of the missile, this may or may not happen simultaneously with the sustain part of flight. Once the target is detected, the missile enters a “track” state. During this part of the flight, the weapon will approach the target utilizing real time information provided by the seeker. Finally, when the missile is very close to the target, it will enter a “terminal” or “endgame” phase, where it will attempt to get close enough to the target to consummate an intercept.

During the course of the engagement, the aircraft that is targeted will make attempts to defeat the missile. There are two main types of defeats: kinematic defeat and overshoot. Kinematic defeat occurs when the missile is unable to reach the target. It typically occurs when the missile is launched near to or outside its maximum range. Overshoot occurs when the target aircraft makes a maneuver during the missile’s terminal stage of flight, and the missile is unable to match it. The missile then flies by the target, which remains outside of the warhead range.

In addition to the terms presented above, there are different flight paths that trajectory designers may elect to use. The two considered in this thesis are lofted and non-lofted trajectories. Lofting is a term that refers to altering the standard flight path of a vehicle to fly at higher altitudes in order to reduce the effect of aerodynamic drag. For some configurations of weapons and targets this is advantageous; however, these guidance decisions should be addressed (and are addressed in this thesis) on a case-by-case basis.

c. Missile Engagement Terminology

In addition to terminology that describes the stages of missile flight, there is terminology that pilots and engineers use to describe stages of the engagement. The first is

shooter, the aircraft that is launching an A-A missile at the target. In addition, there are two major concepts that are derived from the range between the shooter and target. They are known as A-Pole and F-Pole, illustrated in Figure 2.

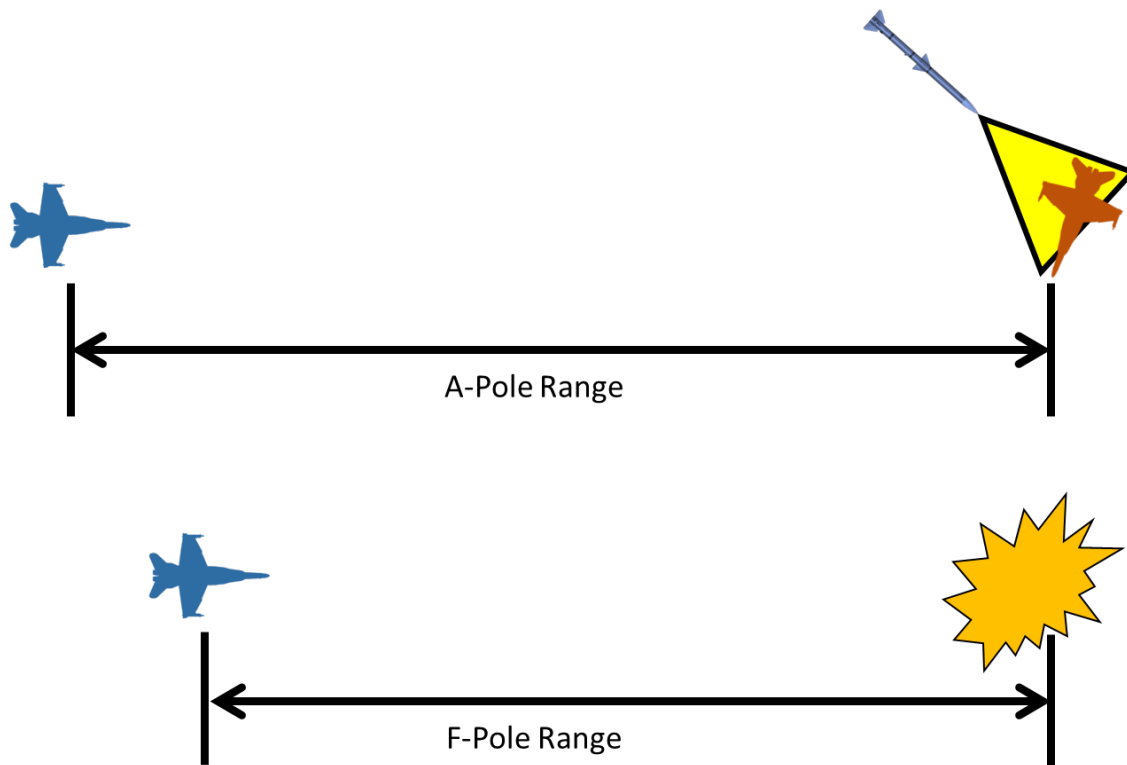


Figure 2. A-Pole and F-Pole Illustration.

A-Pole is the range between the shooter and target aircraft after which an A-A weapon no longer needs support from the launching aircraft. For this thesis, A-Pole will be achieved either at the point where the missile is 1 nautical mile (nm) from the target, or when it can be assured that the weapon will get to 1 nm from the target. F-Pole is a similar concept, except that it refers to the range between the shooter and target when the missile actually intercepts the target. A-Pole and F-Pole are important terms because they describe how close the shooter has to get to the target aircraft in order to eliminate them. For this reason, larger A-Pole and F-Pole numbers are highly desired and correlate with better aircrew survivability.

2. A Brief History of Air Combat Technology

Air-to-Air warfare has existed since the early 20th century. An early example was World War I pilots “hurtling easily-carried items like bricks, chains, nets, darts, and other projectiles, hoping to strike the propeller, wing, or pilot of an enemy machine” [8]. Over the course of the World Wars, technology quickly matured to unguided rockets and machine guns. Shortly after World War II ended, development started for the AIM-4 Falcon. This early weapon was produced in semi-active and infrared (IR) seeking variants. This weapon was quickly proven highly ineffective in live combat, scoring only five kills in 54 launches, and was soon abandoned for the AIM-9 Sidewinder [9].

The AIM-9 is arguably the most successful A-A weapon of all time, as it has been in service continuously since 1956 [10]. It is a short-range, IR seeking A-A missile. The AIM-9 was eventually complemented by the AIM-7 sparrow, a semi-active A-A missile. The AIM-7 was eventually retired in favor of the AIM-120 Advanced Medium Range Air-to-Air Missile (AMRAAM). AMRAAM remains in service with the U.S. Armed Forces to this day [11].

3. A Brief Overview of Missile Guidance

Missile guidance is the method of generating commands for the control system autopilot that guide the missile closer to the target. These autopilots generate fin commands using an error signal to fulfill a given criteria. Three types of flight control systems were considered for this thesis: acceleration control, attitude control, and flight-path angle control [12]. The least complex missile guidance algorithm considered in this thesis is pursuit guidance (PG). The easiest way to describe PG guidance is that “the missile should always head for the target’s current position” [13]. The guidance algorithm attempts to point the missile directly at the target. The angle from the center of the seeker’s Field of Regard (FOR) is referred to as the Line of Sight (LOS) angle. PG tends to overshoot the target and is highly likely to miss in most engagements excluding a shot from directly or significantly behind the target. Equation (1) is the guidance equation for an acceleration autopilot utilizing pursuit guidance, and Figure 3 is a depiction of a 3 Degree of Freedom (DOF) flyout of an A-A missile utilizing PG. In this equation, K_G is the guidance gain, LOS

is the LOS angle to the target, and A_D is the desired acceleration command that is fed to the autopilot to generate fin commands.

$$A_D = K_G(0 - \text{LOS}) \quad (1)$$

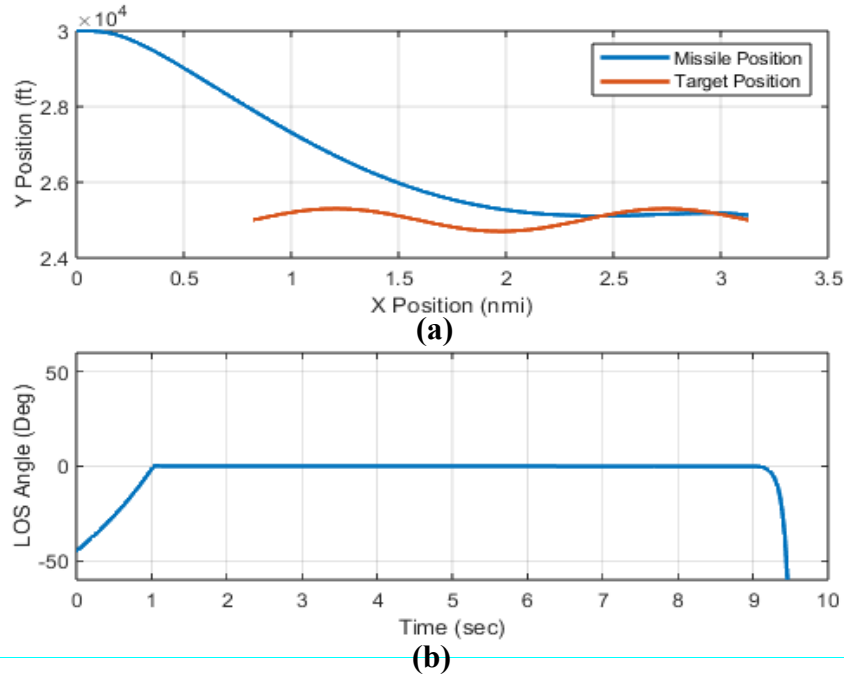


Figure 3. Illustration of Pursuit Guidance 3DOF Engagement: (a) Missile and Target Positions, (b) LOS Angle.

Due to limitations of pursuit guidance, a new guidance law was needed to allow for improved weapon capability. PN was created to solve this problem. Rather than controlling by the LOS angle, PN controls on the LOS angle rate. This allows for more efficient engagements, as PN flies directly to an impact point, rather than chasing down a target. Equation (2) is the guidance equation for an acceleration command autopilot flying a PN trajectory and Figure 4 is an illustration of a 3DOF flyout of an A-A missile using PN guidance. Mathematically, Equation (2) is similar to Equation (1), but the LOS angle in Equation 1 is replaced by the LOS rate in Equation (2). PN seeks to maintain a constant LOS angle rate to the target, thus stabilizing the target's position within the seeker FOR, while PG points the body of the missile directly at the target.

$$A_D = K_G \left(0 - \frac{dLOS}{dt} \right) \quad (2)$$

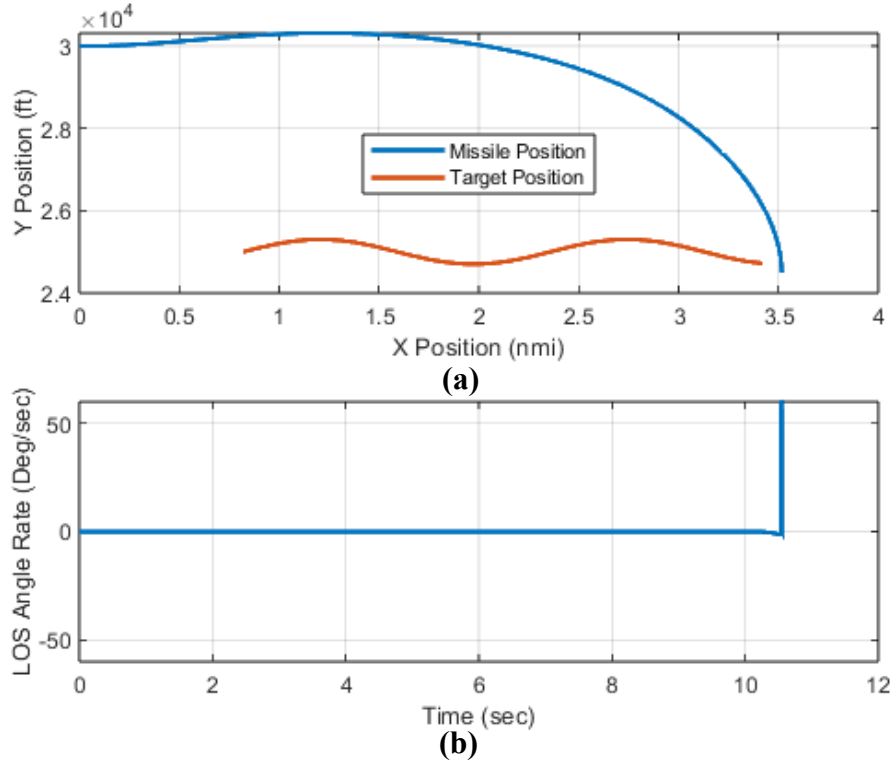


Figure 4. Proportional Navigation 3DOF Engagement: (a) Missile & Target Positions, (b) LOS Angle Rate.

PN also has its limitations. While it is a highly efficient flight path against a non-maneuvering target, it can be defeated fairly easily against a maneuvering target at range. The thrust of the non-committal guidance in this thesis combines PG and PN to exploit the benefits of both methods, while avoiding the negatives. The NCG method proposed in this thesis flies either a lofted or pure pursuit trajectory until close to the target, then switches to a PN endgame. Figure 5 depicts a 3DOF flyout using NCG, which blends PG and PN guidance laws. The implementation used throughout this thesis is listed in Equation (3). The 15-degree offset listed in the lower half of the equation is a tunable design parameter, which should be optimized for every individual engagement. This induced 15-degree angle is used to mimic a PN endgame trajectory.

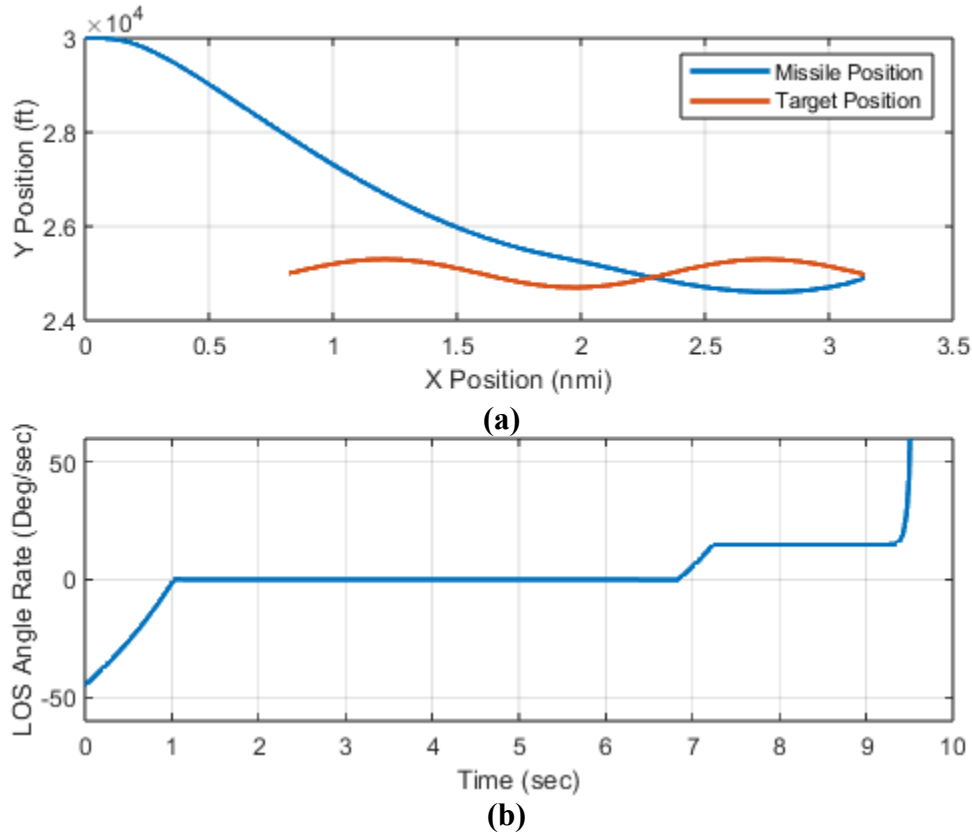


Figure 5. Non-Committal Guidance 3DOF Engagement: (a) Missile and Target Positions, (b) LOS Angle.

$$A_D = \begin{cases} K_A (0 - \text{LOS}) & \text{if Range} > 2 \text{ nmi} \\ K_A (15^\circ - \text{LOS}) & \text{if Range} \leq 2 \text{ nmi} \end{cases} \quad (3)$$

Given a varying set of maneuvering targets and varying launch conditions, there are scenarios where any of the previously discussed algorithms exhibit stellar performance. Generally, NCG will outperform the other two algorithms, because it exploits the strengths of both algorithms and minimizes their weaknesses. PG is highly susceptible to overshoot in the terminal stage of flight. PN is highly susceptible to overly committal guidance decisions. These decisions cause the weapon to perform large maneuvers, which induces large amounts of drag and can lead to kinematic defeat (see the trajectory at the end of the flyout in Figure 4). NCG is highly susceptible to overshoot in the terminal phase if care is not taken to select an appropriate endgame LOS angle. A successful endgame LOS angle should be selected to increase terminal target aspect. Increased terminal target aspect

allows the weapon to approach from the tail of the target, minimizing chance of overshoot. Overall, there are always corner cases where one of the algorithms outperforms the other two, but NCG typically outperforms the other two against real-world maneuvering target sets.

4. Lofted Trajectories

Another common capability used in A-A weapons is the concept of lofting. Lofting is a concept that missiles may use to increase kinematic performance. Lofted profiles are observed by intentionally inducing a missile to climb to higher altitudes to minimize air density traveled through. An example of lofted profiles is shown in Figure 6.

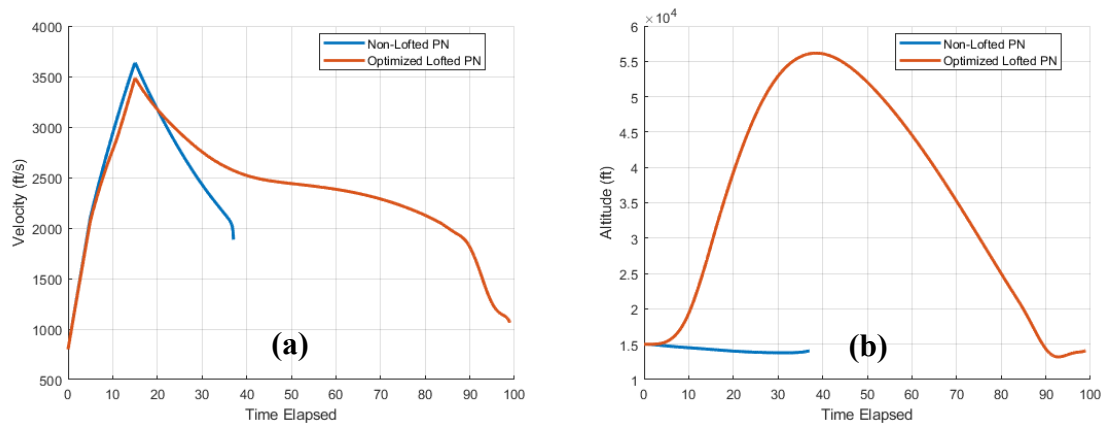


Figure 6. Lofted Trajectory vs. Non-Lofted Trajectory: (a) Missile Velocity, (b) Missile Altitude.

As can be seen in Figure 6, the lofted trajectory takes an initial decrease in velocity, but rapidly regains the velocity advantage.

5. Desired Engagement Geometry

There is a plethora of missiles used in combat, and all have different desired geometry to maximize the chance of mission success. For A-A missiles, the capabilities of the GNC system and the seeker are going to determine the best angle of approach. For the system used for the short and medium range weapons in this thesis, an approach angle

equal to or greater than 45° improves P_k . For the long-range weapon, increasing aspect angle generally increases P_k . Generally, approach angles near 135° will maximize P_k . Figure 7 shows the aspect angle definition used in this thesis.

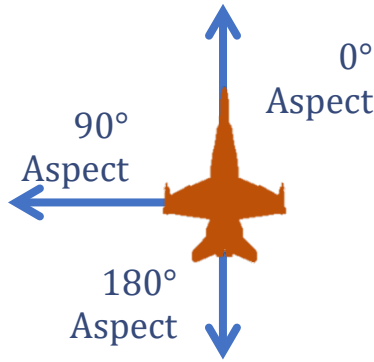


Figure 7. Aspect Angle Definition.

B. BACKGROUND

The non-committal guidance concept has not been well explored in the public literature. Optimal control theory, meanwhile, is an established method for optimizing a cost functional in a dynamical system. Both concepts are described in detail below.

1. Non-Committal Guidance

A common issue with PN is that target maneuvers can cause dramatic over-corrections, which can cause either an overshoot or kinematically defeat the missile. A common issue with PG is that the missile will fly directly towards the target at the target's altitude. If the target is at a lower altitude than the shooter, this causes the missile to experience increased aerodynamic drag which bleeds energy significantly faster. These issues are examples of committal guidance decisions. For the purposes of this thesis, non-committal guidance decisions are defined as logic that minimizes maneuvering in response to target maneuvering for as long as possible. This does not mean that target maneuvers should be ignored, but instead that excess maneuvering of the engaging missile should be minimized. This concept yields two identified benefits over a PN trajectory. First, the

algorithm presented in this thesis improves Probability of Kill (P_K) by being less reactive to target maneuvering. Second, flying a dynamically optimized trajectory maximizes the energy available at endgame. These two components increase an A-A weapon's maximum range and improve P_K within its maximum range.

2. Optimal Control Theory

Optimal control theory is the mathematical field that deals with optimization of a cost functional with respect to dynamic systems. Problems are formulated with a cost functional that is to be optimized, initial and final states, dynamics, and control variables. Generally, such problems can be presented in a standard format proposed by Ross [3], shown in Equation (4).

$$\left\{ \begin{array}{ll} \text{Minimize:} & J[x(\cdot), u(\cdot)] = E(x(t_f), t_f) + \int_{t_0}^{t_f} F(x(t), u(t), t) dt \\ \text{Subject to:} & \begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ x(t_0) &= x^0 \\ t_0 &= t^0 \\ e(x_f, t_f) &= 0 \\ u^L &\leq u(t) \leq u^U \\ h^L &\leq h(t) \leq h^U \end{aligned} \end{array} \right. \quad (4)$$

This standard formulation can address virtually any optimal control problem and is used extensively throughout this thesis. This formulation is normally used in conjunction with Pontryagin's Principle and provides the framework to facilitate the application of optimization used in this thesis. Pontryagin's Principle states that an optimal trajectory can be found by "the instantaneous minimization of the Hamiltonian function" [3]. It is "based on the idea of transforming the [cost function] to computable conditions in the dual (covector) space" [3]. Costates allow all variables to be related to the cost in terms of a common unit. In Pontryagin's Principle, costates are the building blocks for the Hamiltonian and Endpoint Lagrangian functions. Differentiation of these two functions allow determination of the necessary conditions for optimality. These methods can then be used to validate optimality for various solver outputs.

C. MOTIVATIONS

Change is difficult to accomplish within a design space. In 2006, DIDO was utilized to execute propulsion-less practical maneuvers on the ISS [14], proving that trajectory optimization can be implemented on flight hardware. Since then, to the author's knowledge, there has been no realistic push to integrate an optimal trajectory planning solver of this form in an air-to-air weapon. At one point in time, this was due to technological immaturity [15]. Another hurdle to DIDO's widespread implementation is that its ease of use also enables misuse. Specifically, because the software is easy to use, even though the underlying theory is complex, it is easy to misuse the software to generate inaccurate and non-feasible solutions. This is mentioned as a warning in DIDO's documentation [16]. Fortunately, since the publishing of *A Primer on Pontryagin's Principle* [3] and multiple courses teaching this material [17] [18] [19], this issue is starting to recede. Another reason for immaturity is the lack of realistic implementation methods for flight hardware. Pre-planned trajectories are relatively easy to compute with a known launch time and a known target point (e.g., launching a rocket to intercept a space station). On-demand trajectory planning (e.g., an A-A missile) that is highly reliable, repeatable, and optimal is much more complex. To the author's knowledge, this has not been acceptably addressed. This thesis examines the feasibility of on-demand trajectory planning for an A-A missile during the rocket motor burn, as well as the feasibility of an alternative that generates near optimal trajectories.

D. CONTRIBUTIONS

The primary contribution of this thesis is a proposed method to improve the maximum range of an A-A weapon and its performance within maximum range. In addition, these methods are assessed to be feasible for implementation on current generation hardware, even with hardware's high reliability requirements. Fighter-launched weapons require a guaranteed, on-demand trajectory solution. Non-linear programming (NLP) and/or optimal control solvers are often not positioned well to meet these demands. An alternate method involves using interpolation/look-up tables to rapidly generate sub-optimal trajectories. In addition, the author has direct experience with senior engineers who

are hesitant to embrace optimal problem solvers. The author believes that proposing reliable, repeatable solutions is the best way to address this barrier to their adoption. Most issues encountered with previously tested solvers stem from either failed projects or unreliable solutions. DIDO is a powerful tool to solve optimal control problems; however, DIDO makes it possible for users to generate running optimal control code without any knowledge of optimal control theory; thus, DIDO is not infallible. To properly use this tool, care must be taken to verify the generated optimality conditions and validate that its solution propagates in an external simulation. This thesis presents examples on how to validate and implement DIDO code. The secondary contribution is an attempt to design a non-committal guidance system. By delaying committal guidance decisions as late as possible, the “no escape” range of a given weapon can be extended, minimizing chances of defeat.

E. THESIS ORGANIZATION

Chapter II provides an in-depth discussion of the mathematical framework utilized in this research. Chapter III presents the description of the methodology used to solve the problem. Chapter IV describes the results. Overall, the methods proposed are promising, if not outright successful. This research found that: 1) optimization would definitely improve missile performance, but more refinement is needed; and 2) the proposed non-committal guidance method improved performance in most engagements. Chapter V presents conclusions and recommendations.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MATHEMATICAL BACKGROUND ON OPTIMIZATION AND SIMULATION

This thesis makes extensive use of several mathematical concepts that the reader may not be familiar with. This chapter will discuss in detail the mathematical framework for dynamic optimization, machine learning, and modeling and simulation (M&S).

A. PONTRYAGIN'S PRINCIPLE

The goal of this section is to give the reader a brief introduction to Pontryagin's Principle, how it relates to optimal control theory, and its use in this thesis. Optimal control theory is "a branch of mathematical optimization that deals with finding a control for a dynamical system over a period of time such that an objective function is optimized" [3]. It is used to generate optimal trajectories (Optimal Trajectory Planning) that the flight vehicle can follow to maximize the energy available to the weapon at endgame. This is accomplished by using a lower fidelity simulation model to generate optimal trajectories that can be fed to a higher-fidelity simulation model of the flight vehicle to guide an air-to-air missile to the target.

1. Terminology and Variables

Pontryagin's Principle is described using a significant amount of highly specific terminology and variables. Some of this terminology is covered in subsequent sections where it is appropriate to do so, but some foundational terms are covered here. These terms are the cost functional and the concept of common units, as they relate to the optimal control problem. The cost functional is the function that is to be optimized. It is used to model mathematically the decision variables that the system designer wants to minimize (or maximize). It is comprised of two parts, the endpoint cost and the running cost. The endpoint cost is computed at the final time, while the running cost is continuously accumulated via an integral. Table 1 is a list of symbology to be presented in the following sections.

Table 1. Symbology of Pontryagin's Principle

Symbol	Description
$\bar{E}$	Endpoint Lagrangian
$E(x(t_f))$	Endpoint Cost as a function of final states
$e(x(t_f))$	Endpoint Conditions as a function of final states
$F(x,u)$	Running Cost as a function of states and controls
$f(x,u)$	Dynamics as a function of states and controls
$\bar{H}$	Lagrangian of the Hamiltonian
H	Hamiltonian
h	Path Constraint
J	Cost Function
λ	Covector used in the Hamiltonian
v	Covector used in the Endpoint Lagrangian
μ	Covector used in the Path Constraints

2. States and Costates

States are the basic dynamic building blocks of control problems. Any variable whose evolution is described by a derivative function is a state, and any variable that can be “shaped” in time is a control variable. For the purposes of this problem, the dynamics need to be in the form of first-order differential equations.

Covectors are a way to compare states under a common unit. For optimal control problems, the common unit is the cost unit [3]. When a covector is used to convert state values to cost unit values, it is referred to as the costate.

An analogy for the units of covectors is presented in *A Primer on Pontryagin's Principle in Optimal Control* [3]:

Suppose we describe a sandwich that comprises of the variables x_1, x_2, x_3 ... where x_1 is the bread, x_2 is the mayonnaise, x_3 is the lettuce, x_4 is the meat, and so on. Then it is clear that we can describe the state of the sandwich in terms of a vector x of N_x ingredients. Each term of this vector can be measured in appropriate units. For instance, x_1 can be measured in terms of the number of slices of bread, x_2 in terms of ounces of mayonnaise, x_3 in terms of the number of lettuce leaves, and so on.

In this analogy, while all of the ingredients of a sandwich are defined via entirely different units, they can all be compared in a common unit, calories. Ross continues,

“Suppose that each slice of bread is 80 *calories* and each ounce of (lean) meat is 55 *calories*” [3]. If one were to have a sandwich comprised of two slices of bread and five slices of meat, the total calorie count would be 435 calories in the sandwich. In much the same way, covectors relate the units comprising the dynamics by a common unit.

Controls are the name given to variables that can be controlled and can affect states in the problem statement. Common control variables in A-A missile problems are fin positions, pitch rate, and/or yaw rate. In addition to control variables, there is a concept where the control variable “drives” a state variable. This can occur when the time derivative of a state is equal to the control variable itself. These types of variables are often referred to as pseudo controls [15].

3. Paths and Control Constraints

Often in optimization problems, there are constraints that place limitations on the trajectory. Common constraints in aerospace problems are in the form of G-force, terrain, or control surface limits. The first two listed are path constraints, while the latter is a control constraint. Loosely speaking, a control constraint is any constraint that directly limits the control variable, while path constraints are any constraint that either directly limits state values or a combination of state values. Both types of constraints must be enforced over the entire time of the problem. Equation (5) shows the condition any constraints must satisfy.

$$h^L \leq h(x(t), u(t), t) \leq h^U \quad (5)$$

4. Hamiltonian and Endpoint Lagrangian

The Hamiltonian and Endpoint Lagrangian functions are two key concepts of Pontryagin’s Principle. The Hamiltonian is defined by Equation (6):

$$H(\lambda, x, u) := F(x, u) + \lambda^T f(x, u) \quad (6)$$

The purpose of the Hamiltonian is to provide “a ‘linear measure’ for the system dynamics” [20]. Utilizing Pontryagin’s Minimization Principle, the Hamiltonian can be

manipulated in a variety of ways that can provide the necessary conditions of optimality. The Endpoint Lagrangian is a similar concept, and is described in Equation (7):

$$\bar{E}(\nu, x(t_f)) := E(x(t_f)) + \nu^T e(x(t_f)) \quad (7)$$

“The Endpoint Lagrangian provides a ‘linear measure’ for endpoint conditions” [20]. Similarly to the Hamiltonian, the Endpoint Lagrangian can be manipulated to provide optimality conditions.

Naturally, the addition of constraints affects the Hamiltonian. This requires an extension of the Hamiltonian called the Lagrangian of the Hamiltonian described in Equation (8):

$$\bar{H}(\mu, \lambda, x, u, t) := H(\lambda, x, u, t) + \mu^T h(x(t), u(t), t) \quad (8)$$

The Lagrangian of the Hamiltonian incorporates constraints into the Hamiltonian function. The h in Equation (8) is a path constraint function, while μ is the path constraint covector. The behavior of μ is identical for both types of constraints, as shown in Equation (9).

$$\mu_i \begin{cases} \leq 0 & \text{if } h_i(x(t), u(t), t) = h_i^L \\ = 0 & \text{if } h_i^L < h_i(x(t), u(t), t) < h_i^U \\ \geq 0 & \text{if } h_i(x(t), u(t), t) = h_i^U \\ \text{unrestricted} & \text{if } h_i^L = h_i^U \end{cases} \quad (9)$$

These conditions are referred to as Karush-Kuhn-Tucker (KKT) or complementarity conditions, and they exist to provide the mathematics needed to maintain optimality while also satisfying the required constraints, such as G limiters or keep out zones (locations where a weapon is not allowed to fly through).

5. Evaluating Optimality Conditions

The methodology of establishing optimality is performed via partial differentiation of the Hamiltonian and the Endpoint Lagrangian. Ross provides a clear and concise description of these conditions in A Primer on Pontryagin's Principle [3]:

Theorem 2.1 (Pontryagin's Principle). Given an optimal solution to Problem B, there exists an absolutely continuous covector function $\lambda(\cdot)$ and a covector $v(\cdot)$ that satisfy:

- The three Hamiltonian Conditions:
 1. Hamiltonian Minimization Condition
 2. Hamiltonian Value Condition
 3. Hamiltonian Evolution Condition
- The Adjoint Equations
- The Transversality Condition

These conditions provide a mathematical framework to solve any optimization problem. They are described in Equations (10-14):

Hamiltonian Minimization Condition:

$$\frac{\partial \bar{H}}{\partial u} = 0 \quad (10)$$

Hamiltonian Value Condition:

$$H(t_f) = - \frac{\partial \bar{E}}{\partial t_f} \quad (11)$$

Hamiltonian Evolution Condition:

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \bar{H}}{\partial t} \quad (12)$$

The Adjoint Equations:

$$-\dot{\lambda} = \frac{\partial \bar{H}}{\partial x} \quad (13)$$

The Transversality Condition:

$$\lambda(t_f) = \frac{\partial \bar{E}}{\partial x_f}; \lambda(t_0) = - \frac{\partial \bar{E}}{\partial x_0} \quad (14)$$

6. Example: Constrained Brachistochrone Problem

To clarify the concepts presented in Sections 1–5, a constrained brachistochrone problem is presented and solved using DIDO. DIDO is a toolbox for solving optimal control problems published by Elissar Global. DIDO is used extensively throughout the thesis to generate optimal trajectories. The problem statement of the brachistochrone problem was initially published by Johann Bernoulli in a letter to Acta Eruditorum [21]. This problem is an early optimization problem that can easily be described in the standard formulation as:

$$\left\{ \begin{array}{l} \text{Minimize } J[x(\cdot), u(\cdot), t_f] := t_f \\ \text{States: } x, y, v \\ \text{Control: } \theta \\ \text{Subject to } \dot{x} = v \sin \theta \\ \dot{y} = v \cos \theta \\ \dot{v} = g \cos \theta \\ t_0 = 0 \\ (x_0, y_0, v_0) = (0, 0, 0) \\ (x_f - x^f, y_f - y^f) = (0, 0) \end{array} \right. \quad (15)$$

This problem is referred to as the brachistochrone problem, and it is a fairly standard problem that is often used to introduce students to trajectory optimization, as it is easy to work out the necessary conditions by hand. Often more complex and more useful problems cannot be solved by hand; thus, numerical solvers are used. For this example, the final x and y values shall be both set to 10. In order to properly present all of the material, a path constraint is included, creating what will be referred to as the constrained brachistochrone problem (adapted from [3]), as shown in Equation (16).

$$\left\{ \begin{array}{l} \text{Minimize } J[x(\cdot), u(\cdot), t_f] := t_f \\ \text{States: } x, y, v \\ \text{Control: } \theta \\ \text{Subject to } \dot{x} = v \sin \theta \\ \dot{y} = v \cos \theta \\ \dot{v} = g \cos \theta \\ \dot{\theta} = u \\ t_0 = 0 \\ (x_0, y_0, v_0) = (0, 0, 0) \\ (x_f - x^f, y_f - y^f) = (0, 0) \\ 36 \leq x^2 + (y - 10)^2 \end{array} \right. \quad (16)$$

This path constraint creates a “keep out” zone in the bottom left corner of the trajectory space, shown in Figure 8.

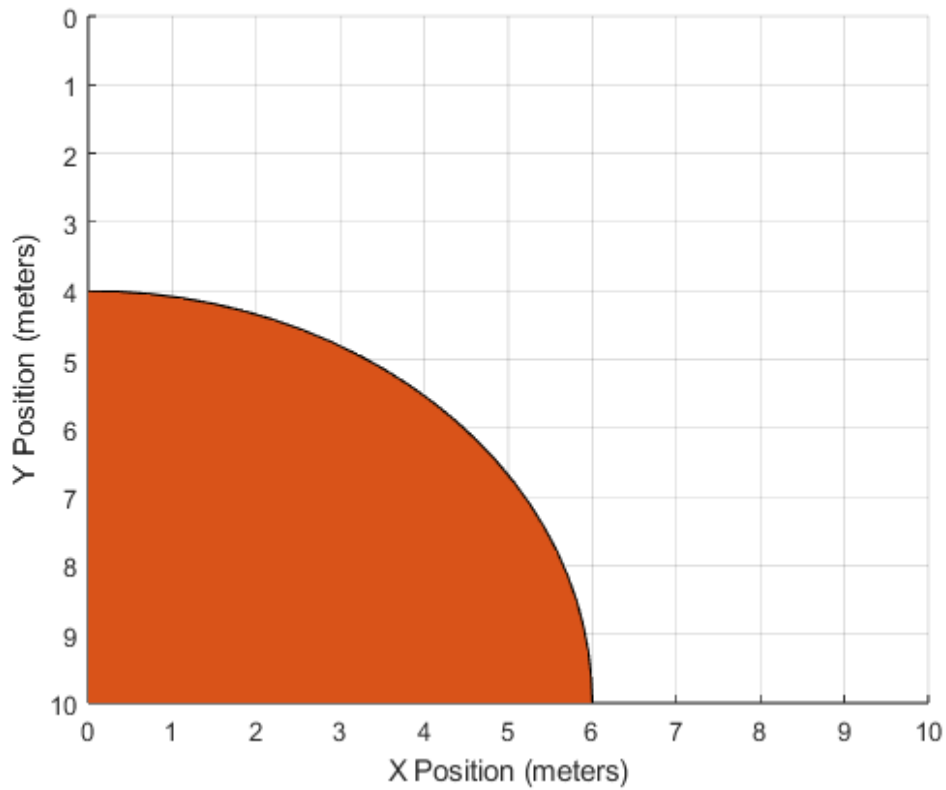


Figure 8. Constrained Brachistochrone “Keep-Out Zone” Constraint.

This modified equation generates the following Hamiltonian, Lagrangian of the Hamiltonian, and Endpoint Lagrangian.

$$H = \lambda_x (v \cos \theta) + \lambda_y (v \sin \theta) + \lambda_v (g \cos \theta) + \lambda_\theta (u) \quad (17)$$

$$\bar{H} = \lambda_x (v \cos \theta) + \lambda_y (v \sin \theta) + \lambda_v (g \cos \theta) + \lambda_\theta (u) + \mu (x^2 + (y - 10)^2) \quad (18)$$

$$\bar{E} = t_f + \nu_1 (x_f - x^f) + \nu_2 (y_f - y^f) \quad (19)$$

Applying the optimality conditions yields the following results.

Hamiltonian Minimization Condition:

$$\frac{\partial \bar{H}}{\partial \theta} = 0 = -\lambda_x v \sin(\theta) + \lambda_y v \cos(\theta) - \lambda_v g \sin(\theta) \quad (20)$$

Hamiltonian Value Condition:

$$H(t_f) = -\frac{\partial \bar{E}}{\partial t_f} = -1 \quad (21)$$

Hamiltonian Evolution Condition:

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \bar{H}}{\partial t} = 0 \quad (22)$$

Adjoint Equations:

$$\dot{\lambda}_x = -\frac{\partial \bar{H}}{\partial x} = -2\mu x \quad (23)$$

$$\dot{\lambda}_y = -\frac{\partial \bar{H}}{\partial y} = 2\mu y \quad (24)$$

$$\dot{\lambda}_v = -\frac{\partial \bar{H}}{\partial v} = -\lambda_x \cos \theta - \lambda_y \sin \theta \quad (25)$$

Transversality Condition:

$$\lambda_x(t_f) = \frac{\partial \bar{E}}{\partial x_f} = \nu_1 \quad (26)$$

$$\lambda_y(t_f) = \frac{\partial \bar{E}}{\partial y_f} = \nu_2 \quad (27)$$

$$\lambda_v(t_f) = \frac{\partial \bar{E}}{\partial v_f} = 0 \quad (28)$$

An optimal solution must satisfy all of the above conditions. For the purpose of determining that an optimal solution has been achieved, often only easy to verify conditions are selected. For the above problem, several of the conditions, especially Equations (26) and (27), are very difficult to verify, as ν_1 and ν_2 are unknown constants. Equations (21), (22), and (28) are all easy to evaluate conditions, thus they were selected to verify optimality. Table 2 presents these selected conditions and the required values to verify optimality.

Table 2. Constrained Brachistochrone Optimality Conditions

<u>Condition</u>	<u>Method of Verification</u>	<u>Required Accuracy</u>
$H(t_f) = -1$	Mean Value	Within 0.01 of desired value
$\frac{\partial H}{\partial t} = 0$	Standard Deviation	Standard Deviation less than 0.001
$\lambda_v(t_f) = 0$	Direct Comparison	Within 0.001 of desired value

The brachistochrone problem was then coded in DIDO and solved. The output is presented in Figure 9 and Table 3.

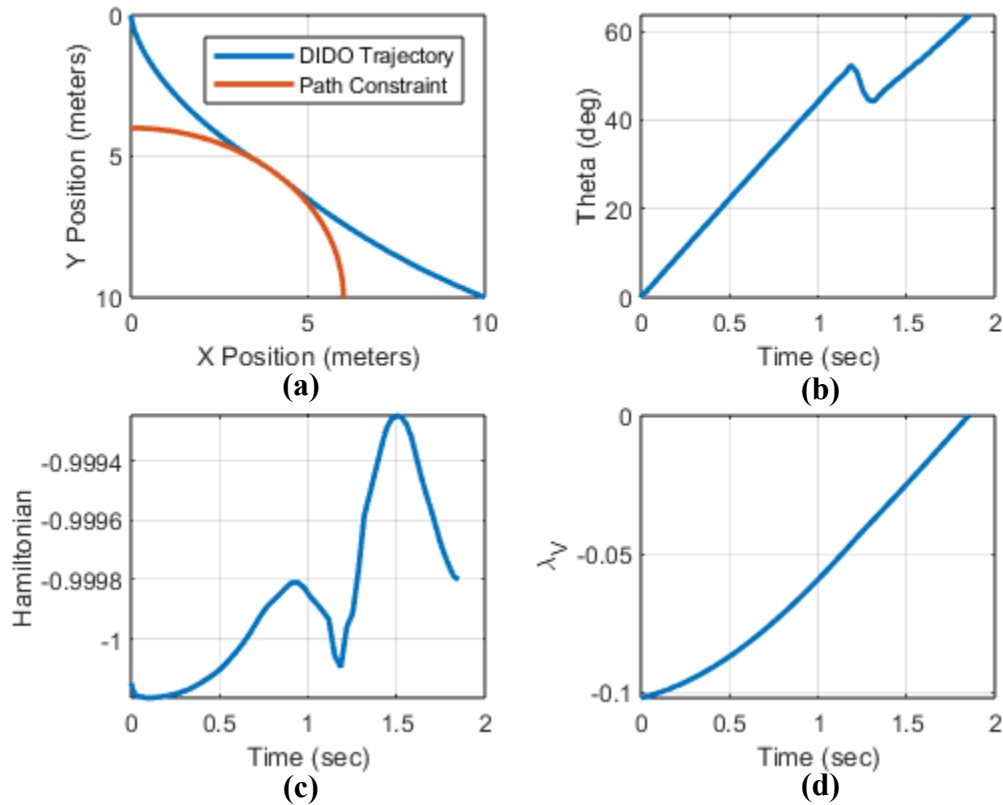


Figure 9. DIDO Solution of Constrained Brachistochrone Problem: (a) Optimal Position Trajectory, (b) Optimal Control Trajectory, (c) Hamiltonian vs. Time, (d) Velocity Costate vs. Time.

Table 3. Constrained Brachistochrone Results

<u>Condition</u>	<u>Value</u>	<u>Pass/Fail</u>
$H(t_f) = -1$	-0.9999	Pass
$\frac{\partial H}{\partial t} = 0$	2.9177e-04	Pass
$\lambda_v(t_f) = 0$	-2.8378e-05	Pass

As Table 3 illustrates, small errors in the values are generally acceptable (for a numerical solution). There is no hard and fast rule for how much error can be tolerated; however, all of the monitored verification and validation (V&V) conditions should be satisfied. Per the requirements in Table 2, the presented conditions have been met, which verifies an optimal solution. In addition to optimality, feasibility must always be tested.

This step validates that the optimal solution presented actually works in the real world. This process is accomplished by fitting the selected DIDO values with an interpolation and inserting this interpolant into the simulated dynamic equations. The exact values may vary depending on application (i.e., in this instance the control value is used, however in later solutions certain state values are used). To match the DIDO generated solution, “X” and “Y” should be 10 at the final time, while “V” should be 14. Results are shown in Figure 10 and Table 4.

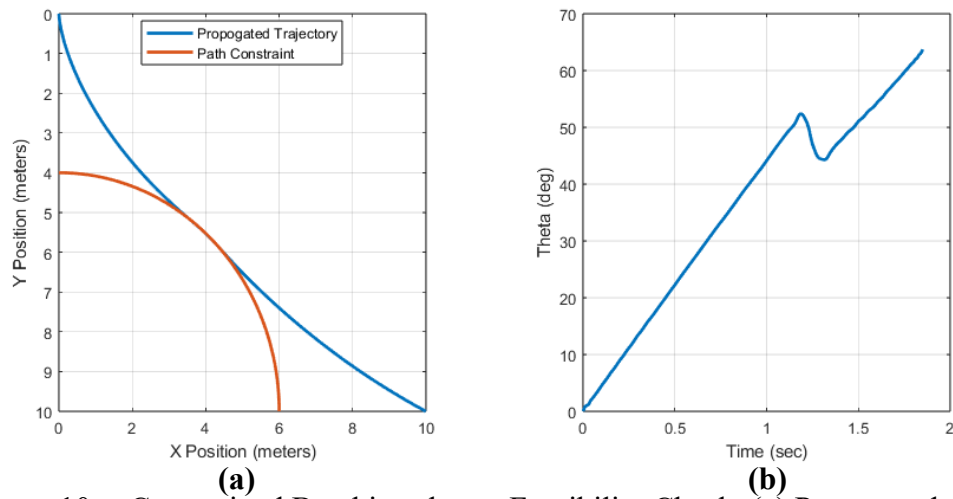


Figure 10. Constrained Brachistochrone Feasibility Check: (a) Propagated Position, (b) Interpolated Control Input.

Table 4. Constrained Brachistochrone Problem Feasibility Test

<u>State</u>	<u>Final Value</u>	<u>Error</u>	<u>Pass/Fail</u>
X	9.9942	0.058%	Pass
Y	10.0022	0.022%	Pass
V	14.0087	0.0621%	Pass

In this section, an optimal trajectory was generated to guide a point to a target in the minimum amount of time while satisfying constraints. This is exactly how optimal trajectory generation works throughout the thesis. In the subsequent sections, the point

mass in a frictionless world will be replaced by an A-A missile in a world with lift, drag, and other forces and moments. The conceptual problem-solving method remains unchanged, however.

B. MACHINE LEARNING METHODS

Alpaydin defines machine learning as follows: “Machine learning is programming computers to optimize a performance criterion using example data or past experience” [22]. Machine learning models use a data set containing input values to predict the output values. A variety of training algorithms exist to determine how to best update the learnable parameters. Depending on the application, the inputs, outputs, and training methods can vary wildly. Some common applications of machine learning include model fitting, classification, and vehicle control (e.g., self-driving cars and stabilizing quadcopters). In this thesis, neural networks are utilized to calculate the aerodynamic polars on the missile while it is in flight, and Bayesian optimization is used to determine maximum launch ranges and lofting range used in the DIDO code.

1. Neural Networks

Neural networks have existed in their modern conceptual form for nearly 80 years. Early applications started around the 1950s [23]. The general consensus was that the computers of that time were not powerful enough to train meaningfully large networks. Making a neural network is a four-step process. These steps are data preparation, modeling, simulation and test, and finally deployment [24]. Feed forward neural networks use a supervised learning method, which means that they require labeled data sets to be trained on.

Once a training dataset has been created, an appropriate network architecture can be created. Neural networks are much more simplistic than the name would suggest. They are formed by attaching a weight and a bias to each input function. In addition, there is an activation function that allows for the neuron to be activated and deactivated. Figure 11 shows the basic design of a single input neuron, and Equation (29) shows its mathematical formulation.

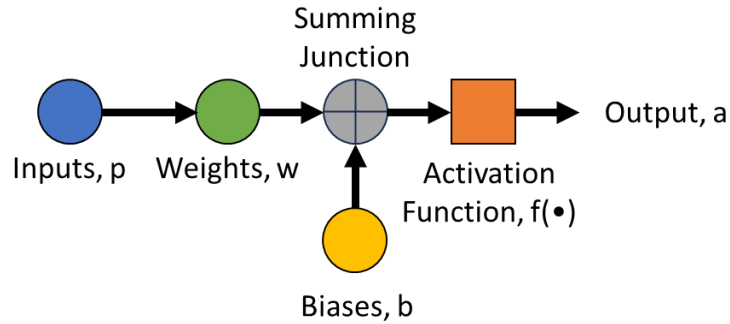


Figure 11. Single Input Neuron Layout.

$$a = f(w p + b) \quad (29)$$

In Equation (29), a is the output value and f is the activation function with the following arguments, w is the weight, b is the bias, and p is the input. A single neuron may not be effective at accurately modeling phenomena, nor at capturing non-linear behavior. For this reason, neural networks are often comprised of multiple neurons in multiple layers. This allows for accurately modeling complex behavior and capturing of non-linear behavior. As an example, a four-input, four-node, single-layer network is shown in Figure 12.

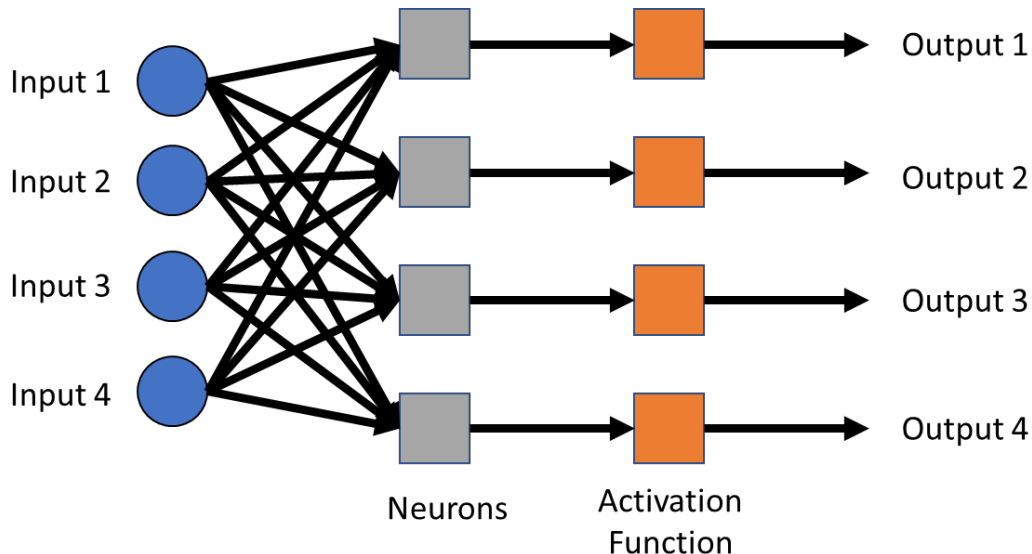


Figure 12. Four-Input, Four-Neuron, Single-Layer Neural Network.

The above structure can be combined in series to form multi-layer networks. All the networks used in this thesis are two layer networks as “it has been shown that two-layer networks, with sigmoid transfer functions in the hidden layer and linear transfer functions in the output layer, can approximate virtually any function of interest to any degree of accuracy, provided many hidden units are available” [23]. Sigmoid transfer functions are another name for activation functions. Neural networks’ near instantaneous calculation of the value modeled make them a natural choice for modeling many phenomena. Their flexibility and rapid runtime, combined with MATLAB’s built in apps and functions for easy network design and deployment, made neural networks a natural choice for modeling aerodynamic polars in this thesis.

2. Training Methods

The final step of creating a useful neural network is training the network. The previous step created a network; however, it did not model anything at all. Training tunes the weights and biases that drive the activation functions in a given network. A variety of training algorithms have been created to improve training duration and performance. This thesis utilizes Levenberg-Marquardt. This thesis presents gradient descent as it is foundational to Levenberg Marquardt and is used in a variety of NLP solvers.

a. Gradient Descent

Gradient descent for neural network training is a simple algorithm that considers each input individually, and iteratively attempts to satisfy Equation (30) by selecting the next query point using an approximated gradient using Equation (31). Note that this is only valid for a specified learning rate.

$$F(x_{k+1}) < F(x_k) \quad (30)$$

$$F(x_{k+1}) = F(x_k) + \alpha_k g_k \quad (31)$$

In Equation (31), α is the learning rate, g is the gradient of the objective function surface, and F is the gradient of the loss function at the previous point. An example function is shown in Figure 13.

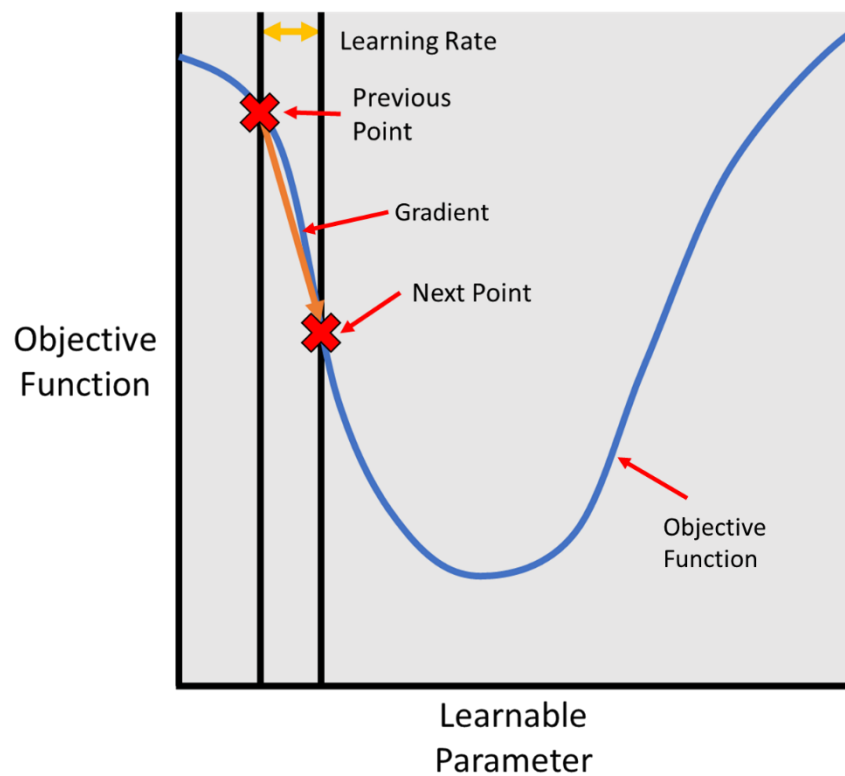


Figure 13. Sample Objective Function Surface.

As can be seen in Equation (31) and Figure 13, step size is a function of the previous steps gradient (not pictured in Figure 13) and the learning rate. Balancing learning rates is a delicate art and should be done for each problem. “If you pick a learning rate that is too small, learning will take too long” [25]. Conversely, a learning rate that is too large will cause the guesses to overshoot the minimum. For this reason, care should be taken when selecting a learning rate. More advanced algorithms using a gradient descent method will use various methodologies to vary the learning rate to attempt to arrive at a minima much faster. One of these methods is the Levenberg-Marquardt method.

b. Levenberg-Marquardt

Levenberg-Marquardt is an algorithm that improves upon gradient descent. While gradient descent uses only the first order derivative of the objective function (gradient), Levenberg Marquardt also utilizes the second order derivative (curvature) to estimate the

direction and location of the local minima. From MATLAB's documentation on Levenberg-Marquardt, "A property of the matrix $Q(k)$ is that when the residual $\|F(k)\|$ tends towards zero as x_k approaches the solution, then $Q(k)$ also tends towards zero" [26], where $Q(k)$ is the matrix of the curvature of an objective function surface, and $F(k)$ is the gradient of the objective function surface. This method is highly successful for medium complexity systems, and it is MathWorks' default solver for Neural Net fitting [27]. As such, it was extensively used in this thesis as the training algorithm for learning the aerodynamic coefficients.

C. 3DOF AND 6DOF DYNAMICAL ENVIRONMENT

Dynamical simulation environments are the backbone of this thesis; therefore, they are used to determine and test the generated trajectories. They comprise the kinematics, dynamics, and implementation of the prior two components into the chosen language. Without a dynamical simulation environment, there is no way to ensure that the design is capable of achieving its goals. To this end, three dynamical environments are utilized in this research. They are a 3DOF Vertical Wind Angle, 3DOF Horizontal Wind Angle, and 6DOF.

1. Coordinate Systems

Aerospace coordinate systems are an exceedingly complex subject, as there are dozens of different systems for various different applications. Some of the more common ones are summarized in Table 5.

Table 5. Common Aerospace Coordinate Systems Adapted from [15]

Reference Frame	Coordinate System	Description
Local-Tangent	Local Area	Tangent-plane at a point on the Earth's Surface
Earth Centered	Earth Centered Earth Fixed (ECEF)	Centered on Earth's Center of Gravity (CG) and rotates with the Earth
Inertial	Earth Centered Inertial	Centered on Earth's CG, but does not rotate with the earth

In addition, there are reference frames for the vehicle itself and the forces applied to the vehicle. Generally, in representing angles and movements for flight vehicles, there are two different conventions. These are referred to as body and wind angles. These two systems are fully equivalent to each other via an appropriate transformation; if used properly they will give the same answer. The difference between them is thus: “the wind axes...are aligned with walls of the wind tunnel (or environmental coordinate system), and the body axes... are aligned to the geometry of the aircraft model” [28]. The two systems are summarized in Table 6 and Figure 14.

Table 6. Vehicle Reference Frames. Adapted from [15].

Reference Frame	Description
Body-Fixed	Origin at the Vehicle’s CG, and rotates with the vehicle
Velocity/Wind	Origin at the Vehicle’s CG and rotates with the velocity vector

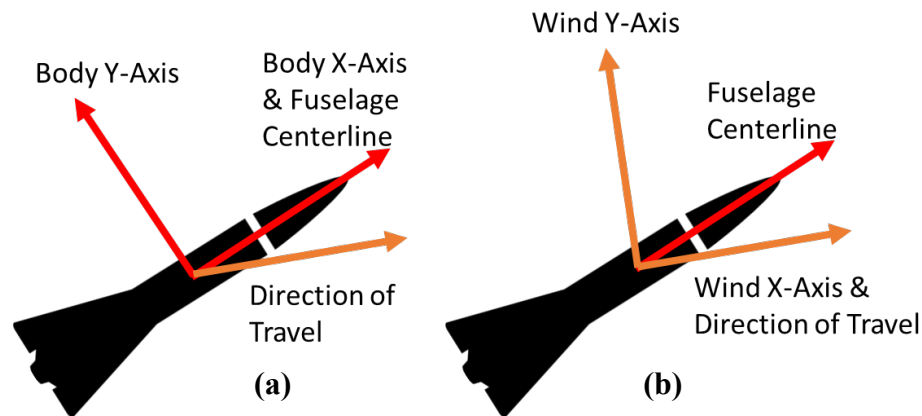


Figure 14. Vehicle Reference Frames: (a) Body-Fixed Reference Frame, (b) Wind Reference Frame.

Typically, wind reference frames are used for 3DOF while 6DOF typically employs body-fixed reference frames. This is the convention MATLAB follows in the default

implementations of the native 3DOF and 6DOF systems, and this thesis also follows that method.

2. Quaternions vs. Euler Angles

Before moving into the coordinate systems, the issue of quaternions vs. Euler angles should be addressed. In the 3DOF dynamics environments, pitch (θ) is all that needs to be considered. The kinetics of a dynamical environment can derive a pitch rate (q). These rates are then integrated and used to propagate the wind angles for the calculation of movement. This is an easily understood method of providing angles and works well for 3DOF simulation. Unfortunately, normal angles do not work well for 6DOF simulation due to singularities found in Euler angles. In the Euler angle rate equations, a singularity occurs for pitch angles of ± 90 degrees. This singularity is not present in 3DOF models. The Euler angle rate equation is:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (32)$$

This equation relates the Euler angle rates (ϕ is roll, θ is pitch, ψ is yaw) to the body-fixed angular velocities, p , q , and r . As can be seen in this equation, when θ is equal to ± 90 degrees the cosine functions in the denominator go to zero and cause a singularity. This is obviously not ideal in an A-A missile simulation. It is highly likely that at some point during the flight of an A-A weapon, one of these singularities would be encountered. This is where quaternions come in. Quaternions have been described by William Thomson, 1st Baron Kelvin [29]: “Quaternions came from Hamilton after his really good work had been done; and, though beautifully ingenious, have been an unmixed evil to those who have touched them in any way.” They are a “System of Imaginaries” [30] that can be used to describe angles on a sphere. A quaternion is comprised of a scalar and an imaginary vector. The general composition of a quaternion is shown in Equation (33) [30]. Following this equation, the four components will be referred to as q_0 , q_1 , q_2 , and q_3 .

$$q = w + ix + jy + kz \quad (33)$$

Using the four components of the quaternion, any orientation and angular rate can be described with no intrinsic singularities (see eqn 34). The quaternions themselves are continuous at all angles, and the equations used to convert from quaternions to Euler angles is significantly more resilient to singularities. No singularities were encountered during any of the work preparing this thesis that could be traced back to an issue with the quaternion implementation. The equation to convert quaternions to Euler angles is shown in Equation (34).

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \operatorname{atan}\left(\frac{2(q_0 q_1 + q_2 q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \\ \operatorname{asin}\left(2(q_0 q_2 - q_1 q_3)\right) \\ \operatorname{atan}\left(\frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \end{bmatrix} \quad (34)$$

3. Vertical 3DOF Dynamics Environment

The first dynamical environment presented is a vertical plane 3DOF. This system is pulled directly from reference [15], and is used for trajectory optimization for the short- and medium-range weapons. This dynamical environment uses a wind reference frame and local tangent plane coordinates. The wind reference frame was judged to be more applicable for this specific application. In addition, the inaccuracies caused by the curvature of the earth were judged to be insignificant. The governing equations are shown in Equation (35) [15]. Forces modeling is shown in Equation (36). The relationship between Flight Path Angle (γ) and Angle of Attack (α) is shown in Equation (37).

$$\left\{ \begin{array}{l} \dot{x} = V \cos \gamma \\ \dot{z} = V \sin \gamma \\ \dot{V} = \frac{F_x}{m} - g \sin \gamma \\ \dot{\gamma} = \frac{F_z}{mV} - \frac{g}{V} \cos \gamma \\ \dot{\alpha} = u_\alpha \end{array} \right. \quad (35)$$

$$\left\{ \begin{array}{l} F_x = \frac{1}{2} \rho V^2 S C_D(\alpha, Mach) \\ F_z = \frac{1}{2} \rho V^2 S C_L(\alpha, Mach) \end{array} \right. \quad (36)$$

$$\alpha = \theta - \gamma \quad (37)$$

4. Horizontal 3DOF Kinematics Environment

The horizontal plane 3DOF is a kinematics only environment. The Guidance, Navigation, and Control (GNC) of the long-range weapon maintains a specified cruise altitude and airspeed, thus kinetics does not need to be considered in the trajectory optimization. Much like the vertical 3DOF, it uses a wind reference frame and local tangent plane coordinates. This is not as valid an assumption for a weapon that cruises farther, however the increased complication to the environment vs. the fidelity benefit was judged to be undesirable. Equation (38) shows the dynamical environment.

$$\left\{ \begin{array}{l} \dot{x} = V \cos \zeta \\ \dot{y} = V \sin \zeta \\ \dot{\zeta} = u_\zeta \end{array} \right. \quad (38)$$

5. 6DOF Dynamics Environment

The 6DOF dynamical environment is the highest fidelity environment used in this thesis. The dynamical environment created from this dynamical environment for this thesis is called FlyOut6. FlyOut6 uses a body-fixed reference frame and local tangent plane coordinates. It is used in all weapon configurations, and all versions of the top-level

simulation. All trajectories generated are verified to be plausible by execution inside of this environment. For this reason, the environment needs to be consistent and execute the problem with a high degree of reliability. The 6DOF quaternion-based implementation used in SIMULINK was selected and is shown in Equation (39). In this formulation, the controls are contained within the forces and moments vectors. These are shown in Equations (40) and (41). The Euler angles in Equation (40) are calculated at the beginning of each step using Equation (34).

$$\begin{cases} \dot{\omega}_b = I^{-1} (M_b - \omega_b \times (I \cdot \omega_b)) \\ \dot{q}_b = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} + (1 - q_0^2 + q_1^2 + q_2^2 + q_3^2) \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \\ \dot{V}_b = \frac{F_b}{m} - \omega_b \times V_b \\ \dot{x}_b = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_0 q_3 + q_1 q_2) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_0 q_1 + q_2 q_3) \\ 2(q_0 q_2 + q_1 q_3) & 2(q_2 q_3 - q_0 q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \cdot V_b \end{cases} \quad (39)$$

$$F_b = \begin{bmatrix} \frac{1}{2} \rho V^2 SC_D(\alpha, \beta, Mach) - mg \sin \theta + Thr \\ \frac{1}{2} \rho V^2 SC_Y(\beta, Mach) - mg \sin \phi \cdot \cos \theta \\ \frac{1}{2} \rho V^2 SC_L(\alpha, Mach) + mg \cos \theta \cdot \cos \phi \end{bmatrix} \quad (40)$$

$$M_b = \begin{bmatrix} 800 \cdot (0 - \phi) + 400 \cdot (0 - \dot{\phi}) \\ \frac{1}{2} \rho V^2 SdC_m(\alpha, Mach) \\ \frac{1}{2} \rho V^2 SdC_n(\beta, Mach) \end{bmatrix} \quad (41)$$

$$\alpha = \text{atan}\left(\frac{w}{u}\right) \quad (42)$$

$$\beta = \text{asin}\left(\frac{v}{V}\right) \quad (43)$$

$$Mach = \frac{V}{a} \quad (44)$$

D. SUMMARY

The mathematical framework for this research is designed with careful consideration of the requirements and how each individual piece interacts with the other pieces. Pontryagin's Principle is the framework used for trajectory optimization. A dynamic optimization problem is solved using the dynamics from the 3DOF environment, and the results are used to generate a general use code, programmed in DIDO. This DIDO code generates solutions that the 6DOF environment executes in order to verify the results. The launch range of the weapon to the target and the final range in the DIDO code is varied using a Monte Carlo approach to determine the region of best performance.

Machine learning is also used in an additional location within the thesis. All coefficients (lift, sideforce, drag, and moments) are modeled using either wind tunnel or CFD data. The MATLAB Neural Network Training app was used, and for most of the curves the default settings sufficed. All curves were checked for evidence of over and under fitting. These fits are extremely easy to implement in MATLAB and allow for rapid simulation of a variety of weapon systems.

III. METHODOLOGY

The previous chapter covered Pontryagin's Principle as well as the underlying physics and dynamics of this thesis, while this one will cover the methodology used to solve the problem. It can largely be split into two parts: 1) the implementation of the missile's physical parameters and simulation environment, and 2) the methodology of trajectory optimization and implementing generated trajectories into the high-fidelity simulation environment.

A. MISSILE DESIGNS

This thesis considers three different configurations for the missiles. They are a short-range solid rocket motor missile physically similar to the AIM-9, a medium range solid rocket motor missile physically similar to the AIM-120, and a long-range air breathing missile. The purpose of the different weapon configurations is to explore the benefits of trajectory optimization on each type of weapon, and to determine whether one type of weapon benefits more or less from applied trajectory optimization. The physical configuration of each weapon is shown below in Table 7.

Table 7. Missile Configurations

<u>Missile</u>	<u>Weight</u>	<u>Propulsion Method</u>	<u>Thrust/Duration</u>
Short-Range	200 lbs.	Solid Rocket Motor	3,600 lbs. for 5 seconds
Medium-Range	600 lbs.	Solid Rocket Motor	5,000 lbs. for 5 seconds 3,700 lbs. for 10 seconds
Long-Range	1,000 lbs.	Solid Rocket Motor, then Airbreather	12,178.1 lbs. for 4.5 seconds Max 1,238.8 lbs. throttleable
<u>Missile</u>	<u>Weight</u>	<u>Reference Area</u>	<u>Reference Length</u>
Short-Range	200 lbs.	0.44 ft ²	1 ft
Medium-Range	600 lbs.	0.6 ft ²	1 ft
Long-Range	1,000 lbs.	1 ft ²	1 ft

1. Propulsion

The missile propulsion selections were based on tuning to meet desired parameters. The small and medium range weapons were designed to achieve Mach 3.5 at burnout, while the tuning of the long-range weapon was a bit different. The long-range weapon has a native GNC algorithm that pursues a constant dynamic pressure. This GNC algorithm will be discussed further in Section III.A.3. This leads to a climbing and accelerating flight profile. While this profile is desirable for maximizing range, it was determined to be too complicated for a study of trajectory optimization on a long-range A-A air breathing cruise missile. For this purpose, a constant cruise condition was inserted into the weapon. The cruise condition is approximately 100,000 feet altitude at a speed of Mach 3.0.

2. Aerodynamic Forces

The aerodynamic properties of each weapon are drawn from a common source [31], however the long-range weapon has had limited CFD iteration done, thus the aerodynamic polars are adjusted. As the base drag, lift/side forces (symmetry is assumed), and moment curves are the same for all weapons (except the long-range weapon where limited CFD runs were completed), some form of adjustment was needed to model a reasonable drag profile for each weapon. This was accomplished via modifying the reference area for each weapon. Beyond that, limited CFD runs were completed for the long-range weapon, which were used to fit force and moment curves for the weapon. Drag curves for the short/medium-range weapons and the long-range weapon are shown in Figure 15.

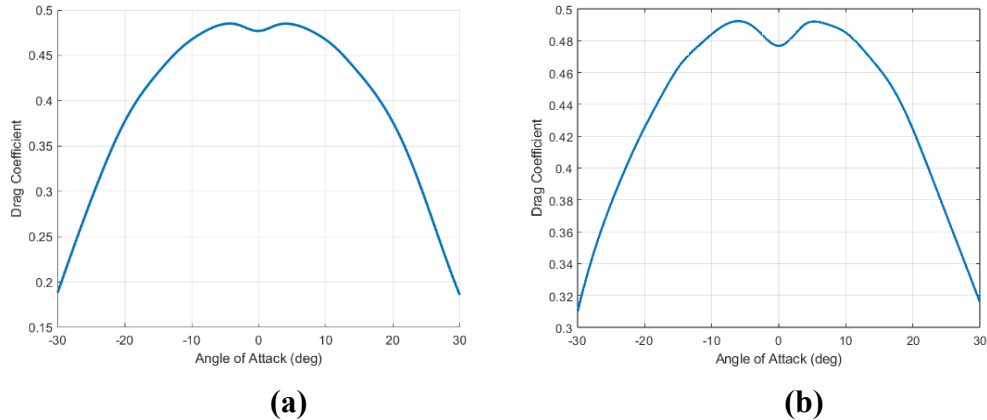


Figure 15. Missile Drag Curve Comparison: (a) Short/Medium Range Weapon Drag Curve, (b) Long Range Weapon Drag Curve.

In addition, while it is assumed that the short- and medium-range weapons have cruciform fins, the long-range weapon has X configuration fins due to external design considerations, with a simple control allocation. The allocation allows for the vehicle to calculate guidance and autopilot commands as if it has a cruciform fin configuration, and then these commands are transformed to X fin commands by rotating the commanded angles around the body X-axis by 45-degrees. The differences between these two configurations are illustrated in Figure 16.

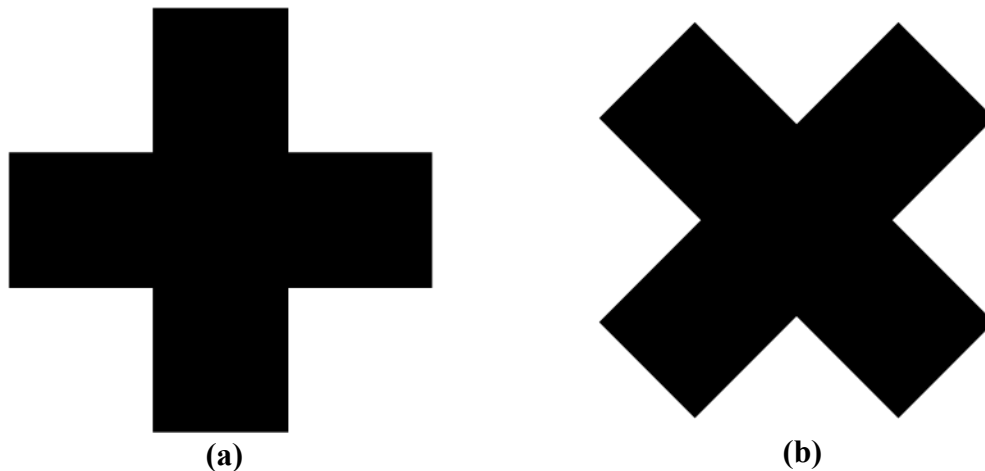


Figure 16. Fin Configuration Comparison: (a) Cruciform Fin Configuration, (b) X Fin Configuration.

3. Native GNC

Both the short- and medium-range weapons are natively capable of utilizing pursuit, PN, or MN guidance. The long-range weapon is not quite as simple. Natively it is an air-to-ground (A-G) weapon. To maximize the range of the weapon, a constant dynamic pressure condition is implemented in the weapon GNC. This leads to a highly efficient climbing and accelerating flight condition but is not ideal for the trajectory optimization work being done in this thesis. The goal of the long-range weapon study in this thesis is to study how a constant cruise condition weapon can improve its approach to the target. The complexity added by the climbing and accelerating flight condition is interesting but adds significant complexity to the study. This trajectory is shown in the flyout chart in Figure 17.

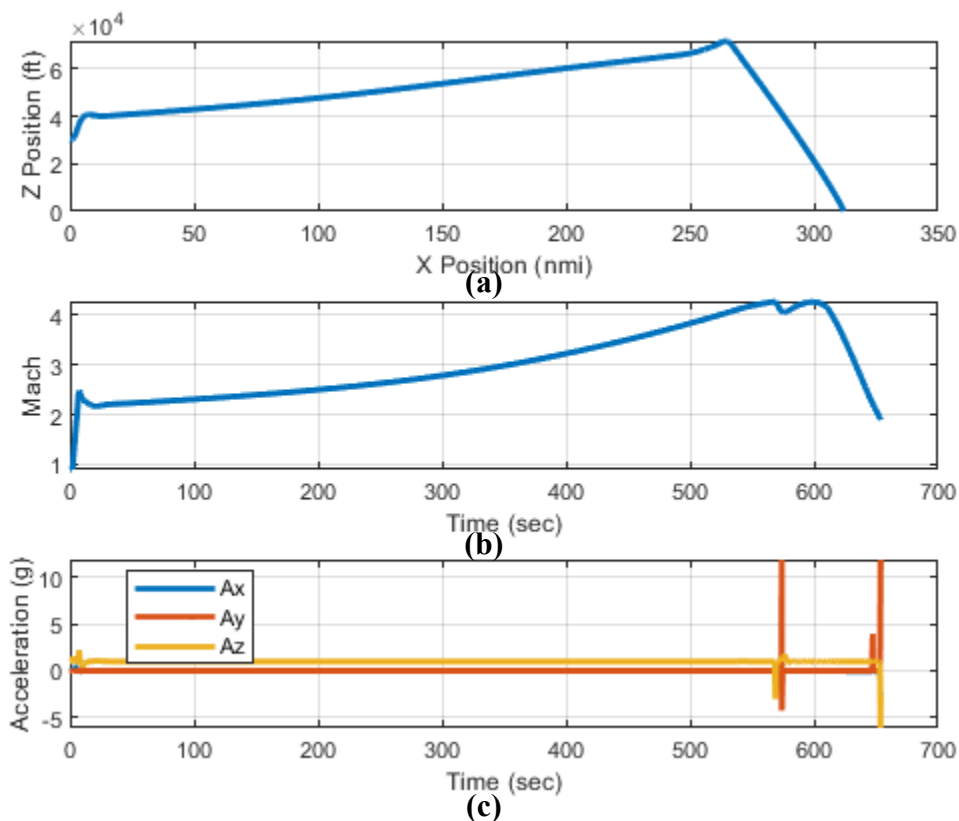


Figure 17. Long Range Missile Flyout Trajectory: (a) Range vs. Altitude, (b) Time vs. Mach, (c) Body Acceleration.

For ease of trajectory optimization, the onboard GNC has been replaced with a constant cruise condition GNC system targeting 100,000 ft and Mach 3.0. This decreases the overall range of the weapon; however, it is still able to complete a 100 nm flyout with sufficient energy to intercept the target. See Figure 18 for the new flyout trajectory used in this thesis.

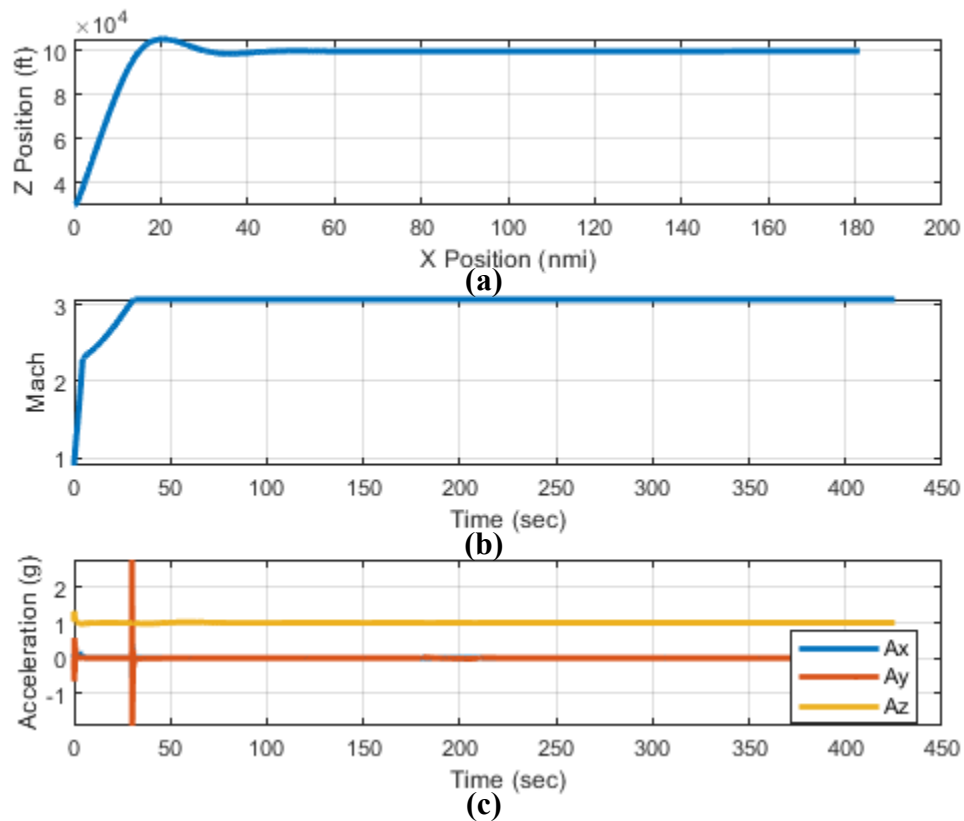


Figure 18. Long Range Missile Cruise Trajectory: (a) Range vs. Altitude, (b) Time vs. Mach, (c) Body Acceleration.

B. M&S ENVIRONMENTS

The cornerstones of this thesis are the M&S environments. Coordinate systems are always local tangent planes and reference frames are consistent between each type of environment (e.g., 3DOF uses Wing and 6DOF uses Body). Transitioning to ECEF coordinates was considered for the long-range weapon, however the error was judged to

be approximately 1.45% over a 100-nm cruise. In addition, this error could easily be accounted for in the real world, and thus was judged acceptable for this application. Each of the environments is slightly different, so they will be described separately in subsequent paragraphs.

1. 3DOF Vertical Plane

The vertical plane 3DOF from a dynamics point of view is taken directly from Bollino [15]. To create a continuous equation for optimization, the International Standard Atmosphere (ISA) model was not used in this step, and instead an exponential atmosphere was derived for use in this model. The differences between the two are shown in Figures 19 and 20. The equation used to create Figure 20 is shown in Equation (45).

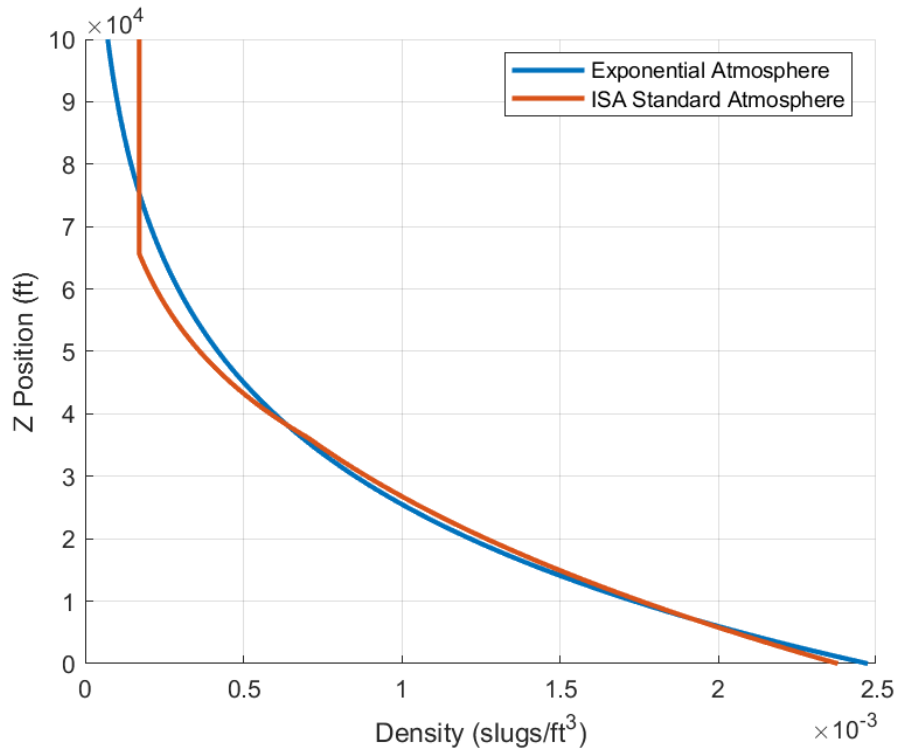


Figure 19. Density vs. Altitude.

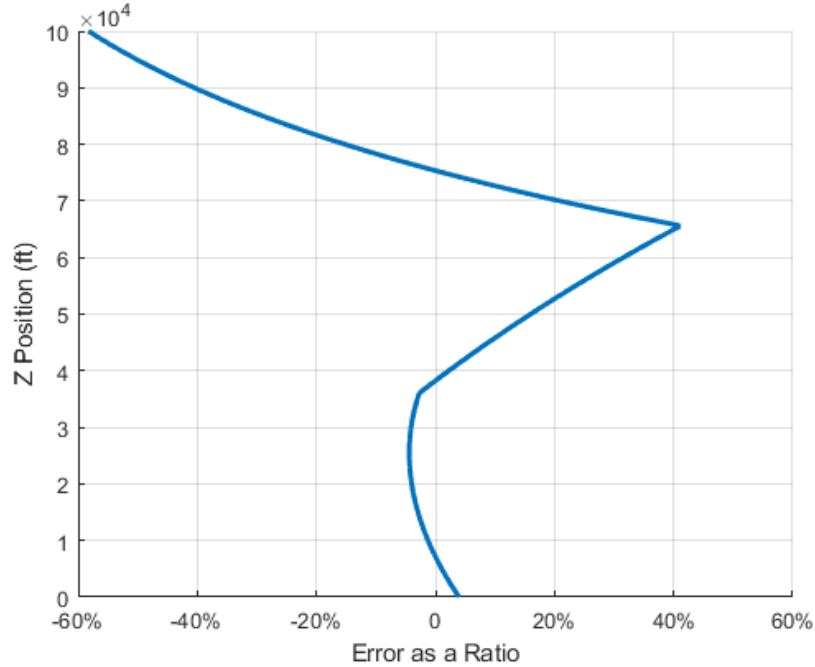


Figure 20. Error vs. Altitude.

$$\%_{error} = \frac{\rho_{exp} - \rho_{ISA}}{\rho_{ISA}} \cdot 100 \quad (45)$$

The exponential fit is within 5% error until 41,000 ft. After that, the fit rapidly departs from the model, reaching 10% error at 44,000 ft. For the purposes of this thesis, the above model is presented as valid for use under 41,000 ft, which is generally sufficient for this work. There may be cases where the weapon dips into a region of the fit that is not valid, however the flyouts will be validated in the 6DOF simulation that uses the ISA model for these cases, so the error is acceptable.

2. 3DOF Horizontal Plane

The 3DOF horizontal plane is intended for use with the long-range weapon only. Due to a constant cruise condition, a heavily simplified dynamical environment consisting of only kinematics was used. This environment is acceptable for use as long as the trajectory demanded does not exceed the engine's capability to provide sufficient thrust. Stable velocity has been observed to be 2,975 ft/s, which relates to Mach 3.07 at 100,000 ft. This is within the theoretical capabilities of the weapon this system is based on, although

it is not an ideal cruise condition. Within that scenario, trajectory optimization is utilized to generate an optimized point-to-point trajectory. No atmospheric effects are modeled as the long-range weapon is capable of maintaining constant cruise while making light maneuvers.

3. 6DOF M&S Environment

The 6DOF M&S environment is the overarching environment of this thesis. All other environments are validated against it. The FlyOut6 code executes individual steps to compute dynamics in a structured way. The general order is unpacking variables from MATLAB's ODESET, calculating external factors such as atmospheric effects, setting the throttle position, seeker detection position, GNC, aerodynamic polars, forces and moments, and mass flow, and then calculating the rates to be fed back into ODESET. Figure 21 details how information is exchanged between these two scripts.

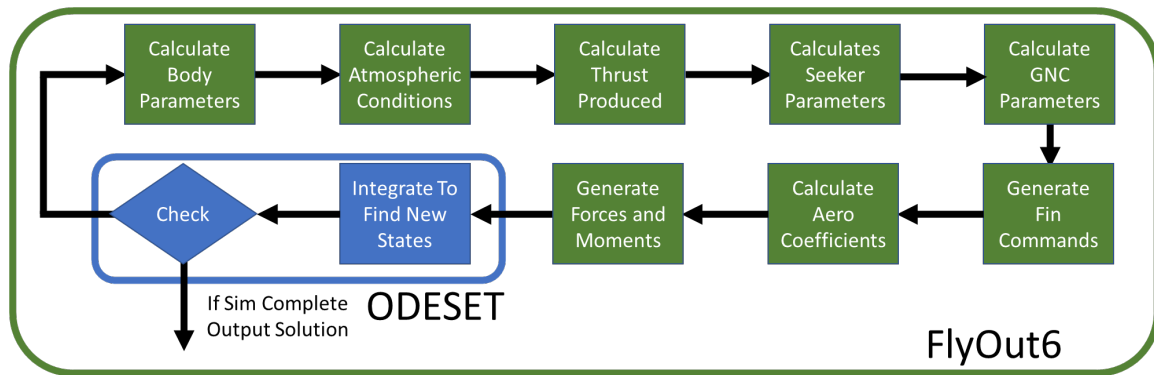


Figure 21. FlyOut6 and ODESET Flowchart.

Since the code uses quaternions for angular positions and rates, conversion to Euler angles is required for some of the seeker and GNC parameters. The equations used to convert quaternions to Euler angles are summarized in Equations (46-48). The axis rotation order used to generate these equations is 3–2–1 or Z–Y–X.

$$\phi = \text{atan} \left(\frac{2 \cdot (q_0 \cdot q_1 + q_2 \cdot q_3)}{1 - 2 \cdot (q_1^2 + q_2^2)} \right) \quad (46)$$

$$\theta = a \sin \left(2 \cdot (q_0 \cdot q_2 - q_1 \cdot q_3) \right) \quad (47)$$

$$\varphi = a \tan \left(\frac{2 \cdot (q_0 \cdot q_3 + q_1 \cdot q_2)}{1 - 2 \cdot (q_2^2 + q_3^2)} \right) \quad (48)$$

Since the environment uses the body reference system; angle of attack (α) and sideslip (β) can be calculated using the body velocities. These relations are presented in Equations (49) and (50).

$$\alpha = a \tan \left(\frac{w}{u} \right) \quad (49)$$

$$\beta = a \sin \left(\frac{v}{V} \right) \quad (50)$$

In addition to the above equations, velocity magnitude is required for some calculations with aerodynamic forces. The equation is shown below in Equation (51).

$$V = \sqrt{u^2 + v^2 + w^2} \quad (51)$$

The 6DOF M&S environment uses the ISA standard atmosphere to determine density and the speed of sound at the current flight altitude. This is an increase in fidelity over the 3DOF environments which use an exponential model. Throttle position (if applicable) follows immediately after the atmospheric calculations. The throttle controller is simplified and uses the formula presented in Equation (52). This equation does require careful tuning and should have the velocity controller gain (K_P) tuned to a scalar value that will ensure a cruise condition when the velocity term is zero.

$$T = 1 + K_P (V_{Des} - V) \quad (52)$$

For these models, the seeker is not accurately simulated. Instead, target position is provided to the guidance section with the options to add position error of a variety of forms (scalar bias, random distribution, normal distribution, etc.). Seeker error was not studied to any degree during the preparation of this thesis. Prior to seeker acquisition (set at 1 nautical

mile), the weapon requires target position passed over a data link from the launching aircraft. The error on this cue is significantly larger than the error on the seeker positioning. GNC for these weapons is done by utilizing an attitude angle autopilot. These autopilots are significantly less accurate in terminal maneuvers than the more traditional acceleration autopilots. This caused a noticeable decrease in performance, particularly in nose aspect engagements. These are utilized because of limitations to the MATLAB ODESET that this environment is built in. In this environment, while states are passed to the solver, unfortunately rates are not. Beyond that, only one individual step is passed to the dynamics file at a time, so a numerical derivative cannot be calculated. As most implementations of acceleration autopilots require numerical derivatives somewhere in the formula, they were precluded from this study. The effect is that terminal accuracy greatly suffers, and true pro-nav guidance is not possible. While this is not ideal, the primary purpose of this thesis is to explore midcourse guidance, thus a degradation in terminal performance is acceptable. Equation (53) presents the mathematical formulation for the autopilots.

$$\delta = K_f \left(K_i \cdot (\theta_c - \theta) - q \right) - q \quad (53)$$

In Equation (48), K_f and K_i are autopilot gains, θ is the current pitch angle, θ_c is the commanded pitch angle, and q is the body pitch rate. The autopilot shown in Equation (53) is a “poor man’s” version of the 3-loop autopilot in [12] and shown in Equation (54).

$$\delta = K_r \left(K_i \int K_a (\theta_c - \theta) - q \, dt - q \right) \quad (54)$$

Due to the previously mentioned restrictions with numerical derivatives and integration, the typical integrator prior to the K_i gain had to be removed. With tuning, satisfactory results were achieved even without this integrator, however it is highly likely that more efficient maneuvering could be achieved with a different system.

Aerodynamic polars are calculated in FlyOut6 via neural networks. This is done for a few reasons. The main one is ease of implementation. A training database can be formed and/or updated from CFD or flight test data, and then a network can be rapidly trained and deployed to the code with ease. The number of data points in the training set

can be increased utilizing interpolants to approximate smooth curves. In this thesis, points were derived from [31], and a modified Akima interpolant was applied to the determined data. Points were sampled from the interpolant, and a training database was formed. Using this methodology, neural networks are a natural choice for modeling the coefficients. Neural networks execute rapidly, and activation functions can create continuous approximations of non-linear behavior, such as the sharp corners often found at zero degrees α . MATLAB's built in neural network fitter was used to generate the networks, and default parameters were found to be sufficient to model the curves.

Figures 22, 23, and 24 show the aerodynamic polars generated by the Neural Networks at Mach 1.5, over a sweep of α values. Equations (55) and (56) show the MATLAB implementation of these in the code.

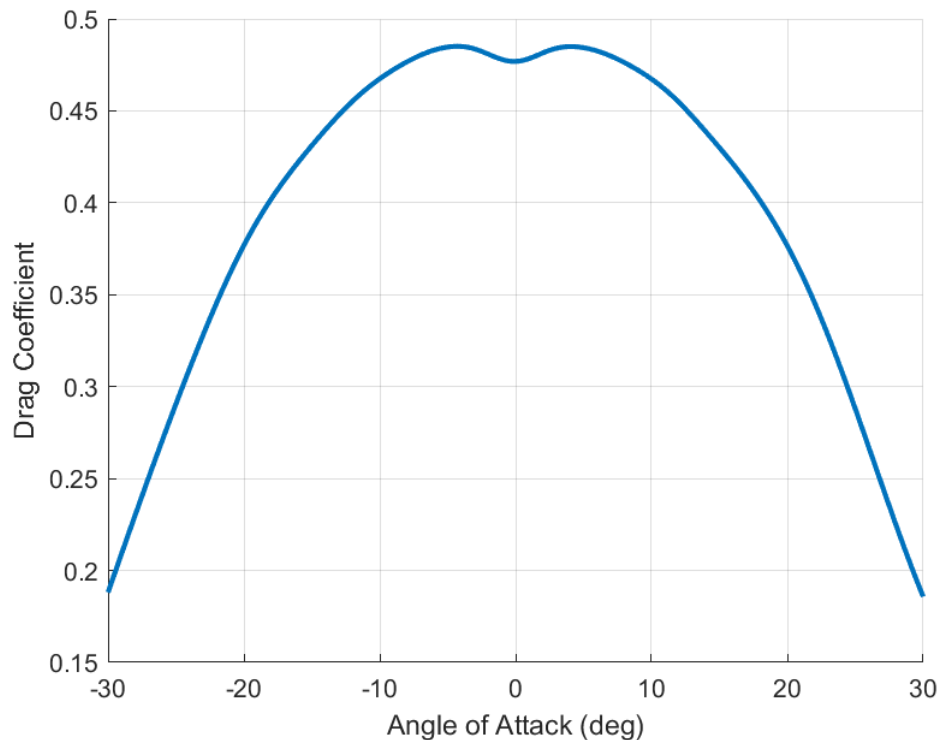


Figure 22. Drag Coefficient vs. Angle of Attack.

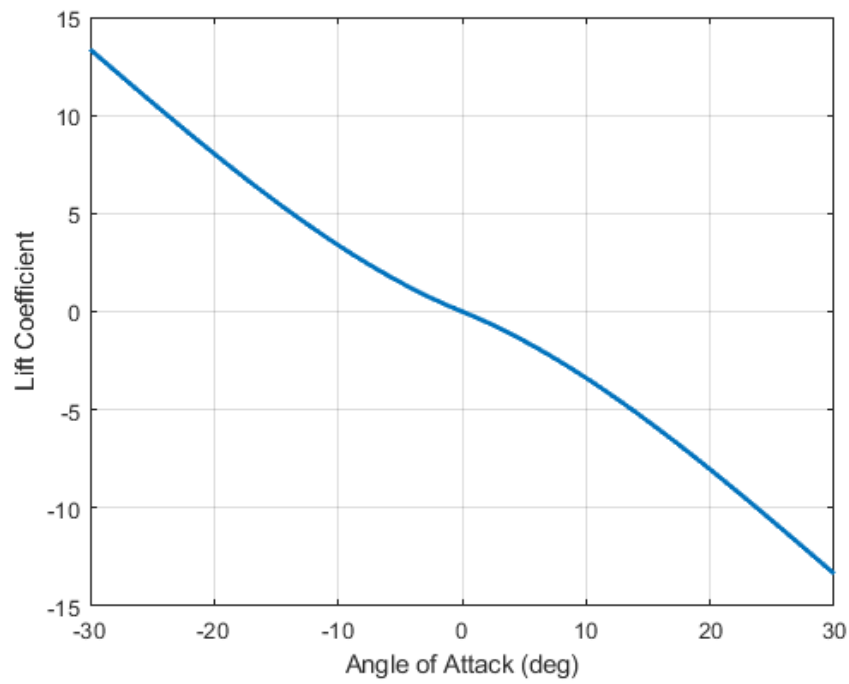


Figure 23. Lift Coefficient vs. Angle of Attack Using a Neural Network Fit.

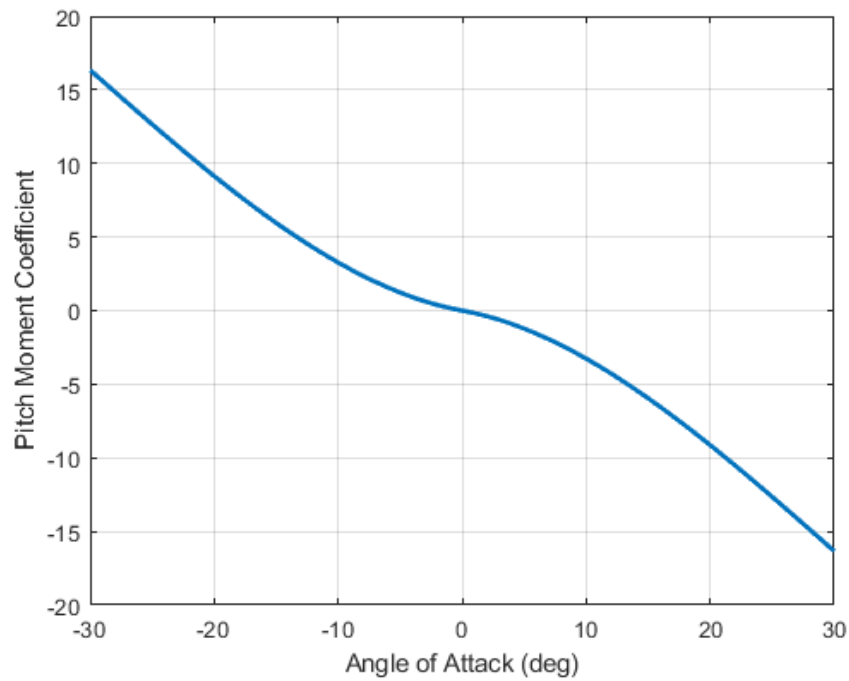


Figure 24. Pitching Moment Coefficient vs. Angle of Attack Using a Neural Network Fit.

$$C_x = -NNCA(Incidence; Mach) \quad (55)$$

$$C_y = -NNCZ(-\beta, Mach) \quad (56)$$

C. TRAJECTORY OPTIMIZATION

Thanks to DIDO, the implementation of trajectory optimization is exceptionally simple. DIDO uses a general format for the description of an optimization problem. The entry is split into five distinct files: *problem file*, *cost file*, *dynamics file*, *events file*, and *preamble file*. These each cover a different part of the problem, and all but the problem file are written as subroutines. The *problem file* is the master file and contains definitions and limits that condition the problem for DIDO, as well as any post processing steps to be executed upon run completion.

1. Cost Functional

The cost functional has the largest individual influence over the trajectory of any individual component. For this reason, care was taken in choosing the cost functional. There were two candidate cost functionals, both stemming from a desire to minimize complexity. They are shown in Equations (57) and (58):

$$C = t_f \quad (57)$$

$$C = t_f + \int \frac{1}{2} u^2 dt \quad (58)$$

The sole difference between these two is the presence of a quadratic running cost in the latter one. This quadratic running cost removes drag induced steering that often occurs if it is not present. Equation (58) was selected for the final version of the cost function because it produced minimal difference in flyout time, but better-preserved available energy (both physical and electrical). The cost function in Equation (57) was observed to cause several additional maneuvers during flyout, and while these maneuvers optimized the time of arrival, they were judged to be non-optimal for an end-to-end engagement due to increased drag. Trajectories produced by each cost functional are shown in Figure 25.

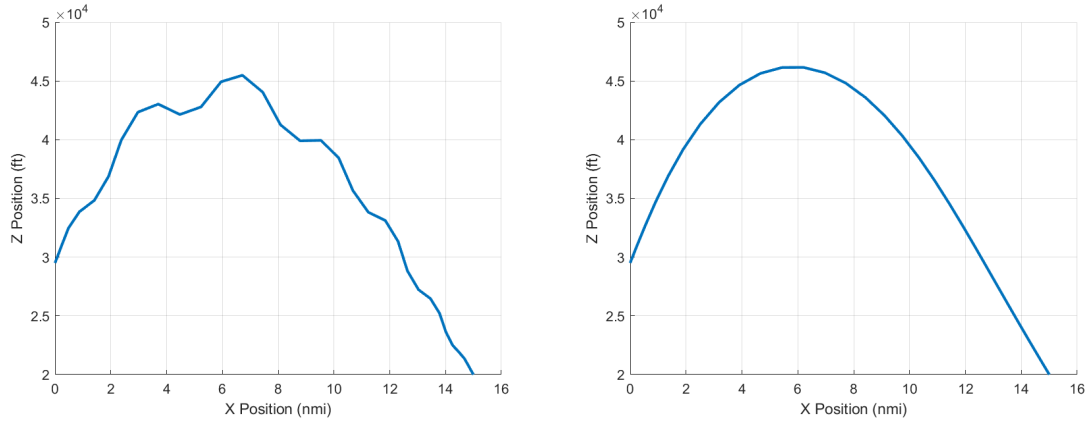


Figure 25. Cost Functional 1: Without Control Effort Function (left) vs. Cost Functional 2: With Control Effort Function (right).

2. Application of Pontryagin's Principle

With the cost functional selected, the optimality conditions can be evaluated. These conditions will be evaluated for each DIDO run to ensure optimality in generated trajectories.

a. 3DOF Vertical Plane

Equation (59) presents the problem formulation in standard form, while Equations (60-75) show the evaluation of Pontryagin's Principle for the 3DOF Vertical plane.

$$\left\{ \begin{array}{l}
 \text{Minimize } J[x(\cdot), u(\cdot), t_f] := t_f + \int \frac{1}{2} u_a^2 dt \\
 \text{Subject to: } \dot{x} = V \cos \gamma \\
 \dot{z} = V \sin \gamma \\
 \dot{V} = \frac{F_x(\text{Mach}, \alpha)}{m} - g \sin \gamma \\
 \dot{\gamma} = \frac{F_z(\text{Mach}, \alpha)}{mV} - \frac{g}{V} \cos \gamma \\
 \dot{\alpha} = u_a \\
 (x_0, y_0, v_0, \gamma_0, \alpha_0) = (0, 0, 0, 0, 0) \\
 (x_f - x^f, y_f - y^f) = (0, 0) \\
 \text{Constrained by: } -300 \frac{\text{deg}}{\text{sec}} \leq u_a \leq 300 \frac{\text{deg}}{\text{sec}} \\
 -40g's \leq A_z \leq 40g's
 \end{array} \right. \quad (59)$$

Hamiltonian, and Endpoint Lagrangian.

$$H = \frac{1}{2} u_\alpha^2 + \lambda_x (V \cos \gamma) + \lambda_y (V \sin \gamma) + \lambda_v \left(\frac{F_x(Mach, \alpha)}{m} - g \sin \gamma \right) + \lambda_z \left(\frac{F_z(Mach, \alpha)}{mV} - \frac{g}{V} \cos \gamma \right) + \lambda_\alpha (u_\alpha) \quad (60)$$

$$\bar{H} = -\frac{1}{2} u_\alpha^2 + \lambda_x (V \cos \gamma) + \lambda_y (V \sin \gamma) + \lambda_v \left(\frac{F_x(Mach, \alpha)}{m} - g \sin \gamma \right) + \lambda_z \left(\frac{F_z(Mach, \alpha)}{mV} - \frac{g}{V} \cos \gamma \right) + \lambda_\alpha (u_\alpha) + \mu_1 (u_\alpha) + \mu_2 \left(\frac{F_z(Mach, \alpha)}{m} \cos \alpha + \frac{F_x(Mach, \alpha)}{m} \sin \alpha \right) \quad (61)$$

$$\bar{E} = t_f + \nu_1 (x_f - x^f) + \nu_2 (y_f - y^f) \quad (62)$$

Applying the optimality conditions yields the following results.

Hamiltonian Minimization Condition:

$$\frac{\partial H}{\partial u_\alpha} = 0 = u_\alpha + \lambda_\alpha + \mu_1 \Rightarrow u_\alpha = -\lambda_\alpha - \mu_1 \quad (63)$$

Hamiltonian Value Condition:

$$H(t_f) = -\frac{\partial \bar{E}}{\partial t_f} = -1 \quad (64)$$

Hamiltonian Evolution Condition:

$$\frac{\partial \mathcal{H}}{\partial t} = \frac{\partial H}{\partial t} = 0 \quad (65)$$

Adjoint Equations:

$$\dot{\lambda}_x = -\frac{\partial H}{\partial x} = 0 \quad (66)$$

$$\dot{\lambda}_z = -\frac{\partial H}{\partial z} = 0 \quad (67)$$

$$\dot{\lambda}_v = -\frac{\partial H}{\partial V} = \lambda_v \frac{\partial}{\partial V} \left(\frac{F_x(Mach, \alpha)}{m} \right) + \mu_2 \frac{\partial}{\partial V} \left(\frac{F_z(Mach, \alpha)}{m} \cos \alpha + \frac{F_x(Mach, \alpha)}{m} \sin \alpha \right) \quad (68)$$

$$\dot{\lambda}_\gamma = -\frac{\partial H}{\partial \gamma} = -\lambda_x (V \sin \gamma) + \lambda_y (V \cos \gamma) - \lambda_v (g \cos \gamma) \quad (69)$$

$$\dot{\lambda}_\alpha = -\frac{\partial H}{\partial \alpha} = 0 \quad (70)$$

Transversality Condition:

$$\lambda_x(t_f) = \frac{\partial \bar{E}}{\partial x_f} = \nu_1 \quad (71)$$

$$\lambda_y(t_f) = \frac{\partial \bar{E}}{\partial y_f} = \nu_2 \quad (72)$$

$$\lambda_v(t_f) = \frac{\partial \bar{E}}{\partial v_f} = 0 \quad (73)$$

$$\lambda_\gamma(t_f) = \frac{\partial \bar{E}}{\partial \gamma_f} = 0 \quad (74)$$

$$\lambda_\alpha(t_f) = \frac{\partial \bar{E}}{\partial \alpha_f} = 0 \quad (75)$$

The above equations give the following optimality conditions for validation. Conditions that are too complex to evaluate have been ignored. Table 8 lists the chosen conditions, and lists the methodology selected to validate them.

Table 8. 3DOF Vertical Plane Optimality Conditions

Condition	Method of Verification
$H(t_f) = -1$	Mean Value
$\frac{\partial H}{\partial t} = 0$	Standard Deviation
$\dot{\lambda}_x = 0$	Standard Deviation
$\dot{\lambda}_y = 0$	Standard Deviation
$\dot{\lambda}_\alpha = 0$	Standard Deviation
$\lambda_v(t_f) = 0$	Direct Comparison
$\lambda_\gamma(t_f) = 0$	Direct Comparison
$\lambda_\alpha(t_f) = 0$	Direct Comparison

b. 3DOF Horizontal Plane

Equation (76) presents the problem formulation in standard form, while Equations (77-87) show the evaluation of Pontryagin's Principle for the 3DOF Horizontal plane.

$$\left\{ \begin{array}{l} \text{Minimize } J[x(\cdot), u(\cdot), t_f] := t_f + \int \frac{1}{2} u_\zeta^2 dt \\ \text{Subject to: } \dot{x} = V \cos \zeta \\ \dot{y} = V \sin \zeta \\ \dot{\zeta} = u_\zeta \\ (x_0, y_0, \zeta_0) = (0, 0, 0) \\ (x_f - x^f, y_f - y^f) = (0, 0) \end{array} \right. \quad (76)$$

Hamiltonian, and Endpoint Lagrangian.

$$H = \frac{1}{2} u_\zeta^2 + \lambda_x (V \cos \zeta) + \lambda_y (V \sin \zeta) + \lambda_\zeta (u_\zeta) \quad (77)$$

$$\bar{E} = t_f + \nu_1 (x_f - x^f) + \nu_2 (y_f - y^f) \quad (78)$$

Applying the optimality conditions yields the following results.

Hamiltonian Minimization Condition:

$$\frac{\partial H}{\partial u_\zeta} = 0 = u_\zeta + \lambda_\zeta \Rightarrow u_\zeta = -\lambda_\zeta \quad (79)$$

Hamiltonian Value Condition:

$$H(t_f) = -\frac{\partial \bar{E}}{\partial t_f} = -1 \quad (80)$$

Hamiltonian Evolution Condition:

$$\frac{\partial H}{\partial t} = 0 \quad (81)$$

Adjoint Equations:

$$\dot{\lambda}_x = -\frac{\partial H}{\partial x} = 0 \quad (82)$$

$$\dot{\lambda}_y = -\frac{\partial H}{\partial y} = 0 \quad (83)$$

$$\dot{\lambda}_\zeta = -\frac{\partial H}{\partial \zeta} = -\lambda_x (V \sin \zeta) + \lambda_y (V \cos \zeta) \quad (84)$$

Transversality Condition:

$$\lambda_x(t_f) = \frac{\partial \bar{E}}{\partial x_f} = \nu_1 \quad (85)$$

$$\lambda_y(t_f) = \frac{\partial \bar{E}}{\partial y_f} = \nu_2 \quad (86)$$

$$\lambda_\zeta(t_f) = \frac{\partial \bar{E}}{\partial \zeta_f} = 0 \quad (87)$$

Conditions that are too complex to evaluate have been ignored. Table 9 shows the optimality conditions selected for evaluation, and the methodology of evaluating them.

Table 9. 3DOF Horizontal Plane Optimality Conditions

Condition	Method of Verification
$H(t_f) = -1$	Mean Value
$\frac{\partial H}{\partial t} = 0$	Standard Deviation
$\dot{\lambda}_x = 0$	Standard Deviation
$\dot{\lambda}_y = 0$	Standard Deviation
$\lambda_\zeta = 0$	Mean Value
$\lambda_\zeta(t_f) = 0$	Direct Comparison

D. TARGET GENERATION

Target generation is accomplished via two methods. The first uses a set target that maneuvers in a preplanned manner to facilitate comparison between methods. The other

method uses actual engagements from a commercial flight simulator to generate more “realistic” target maneuvers. These two target sets are referred to as the control set and the complex set.

1. Control Set Engagements

The control set engagements were structured to be similar in every method to allow for direct comparison. The set was broken into three sub-groups. These were the unaware targets that make no maneuver at any point post launch, the semi-aware targets that maneuver 30 seconds post launch, and the aware targets that maneuver at launch. Inside these sets, the turns were varied from 45–180 degrees in 45-degree increments. A-Pole and F-Pole data was collected from each run. The shooter and target were heading directly towards each other with 1 nautical mile lateral offset. The starting range was determined by regression to determine the maximum range possible. Due to limitations of the terminal guidance, an actual intercept was not required to collect A-Pole and F-Pole data. This made the A-Pole and F-Pole data less useful in the control set engagements. The primary data that should be considered for this data set are the ranges and differences between each method.

2. Complex Set Engagements

The complex set is intended to include additional factors not captured in the control set. This set has varying target parameters, and no pre-planned maneuvers are executed. Instead, the target trajectory is drawn from the flight simulator trajectories. For this set, A-Pole and F-Pole will not be assessed if the weapon is unable to meet the specified criteria for each. The criterion to assess A-Pole is to guide the weapon within 1 nautical mile of the target. The criterion to assess F-Pole is to guide the weapon within 100 ft of the target. Each of the 4 targets considered is covered in the subsequent sections.

a. Complex Target 1: High Aspect Maneuvering Target

This scenario is comprised of a target that begins with approximately 90 degrees aspect to the shooter. The target is consistently maneuvering as the shooter aircraft closes on the target. For the medium range missile, the scenario will neglect shooter position, as

it is assumed the weapon is fired well outside the scenario ranges. For the short-range missile, the missile will be fired from a point on the shooter aircraft's track. Figure 26 illustrates the scenario.

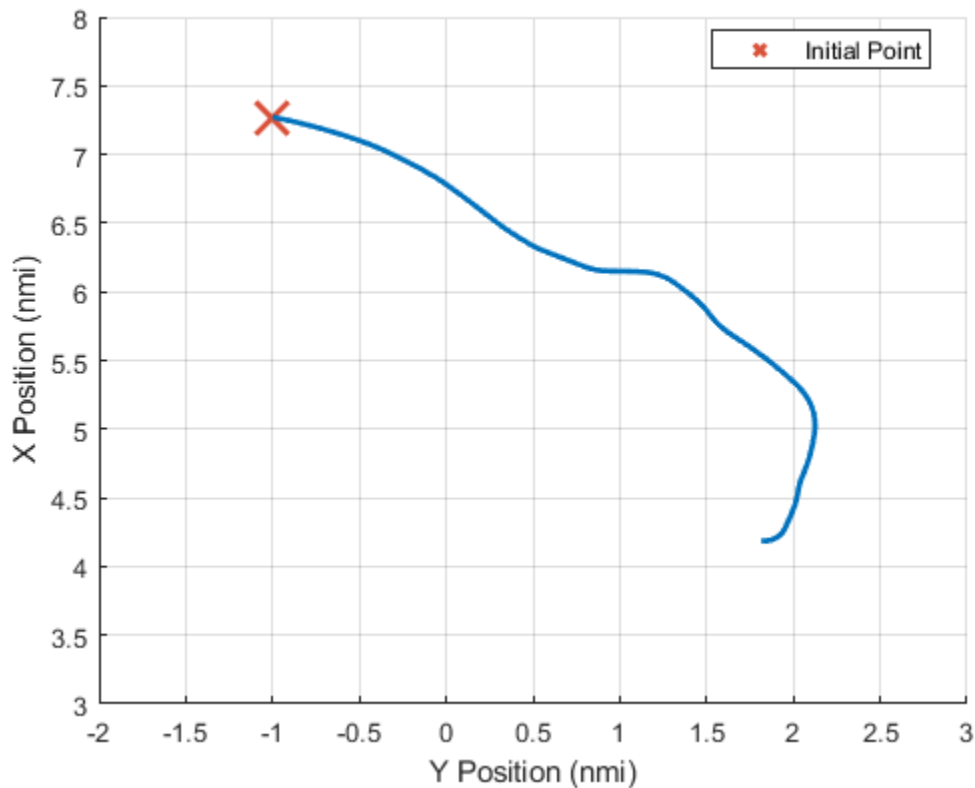


Figure 26. Complex Target 1: High Aspect Maneuvering Target.

b. Complex Target 2: Constantly Maneuvering Target

This scenario is intended to capture algorithm behavior against a constantly maneuvering target. In this scenario, the shooter is positioned behind the target as the target is maneuvering back and forth to attempt to flee. Figure 27 shows the flyout trajectory.

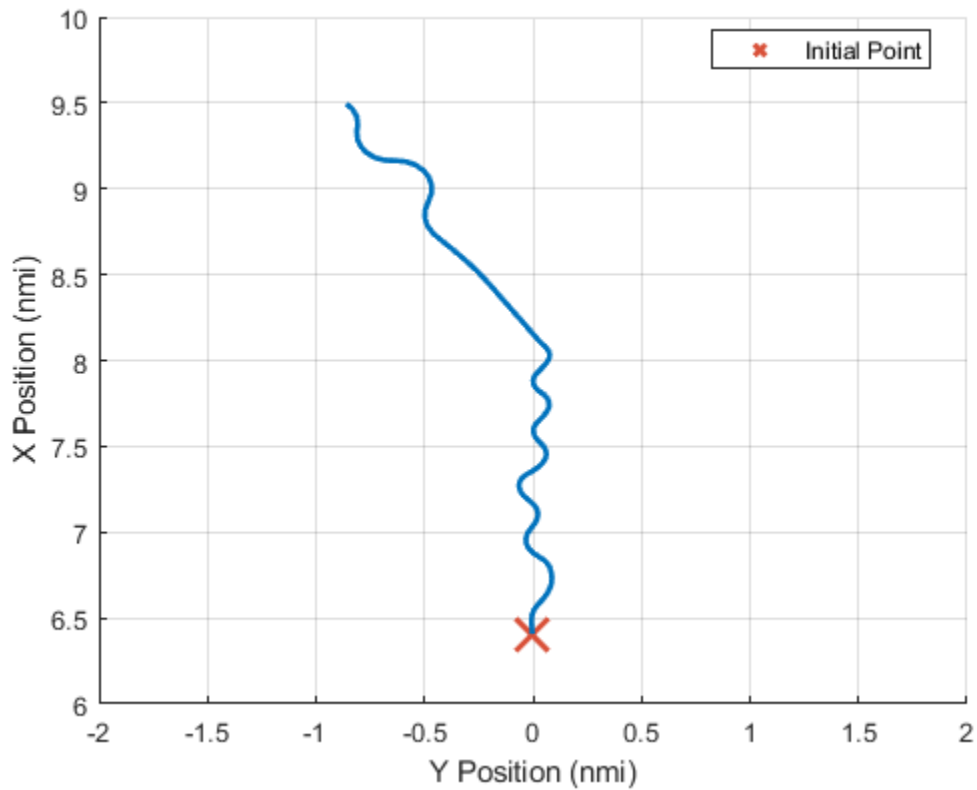


Figure 27. Complex Target 2: Constantly Maneuvering Target.

c. Complex Target 3: Run Down

This scenario is intended to cover the possibility of a target being chased down, and at launch the target turns back into the shooter. Due to the length of the runtime of this scenario, there are two different files used for short and medium range missiles. Visual representations of these scenarios are shown in Figures 28 and 29.

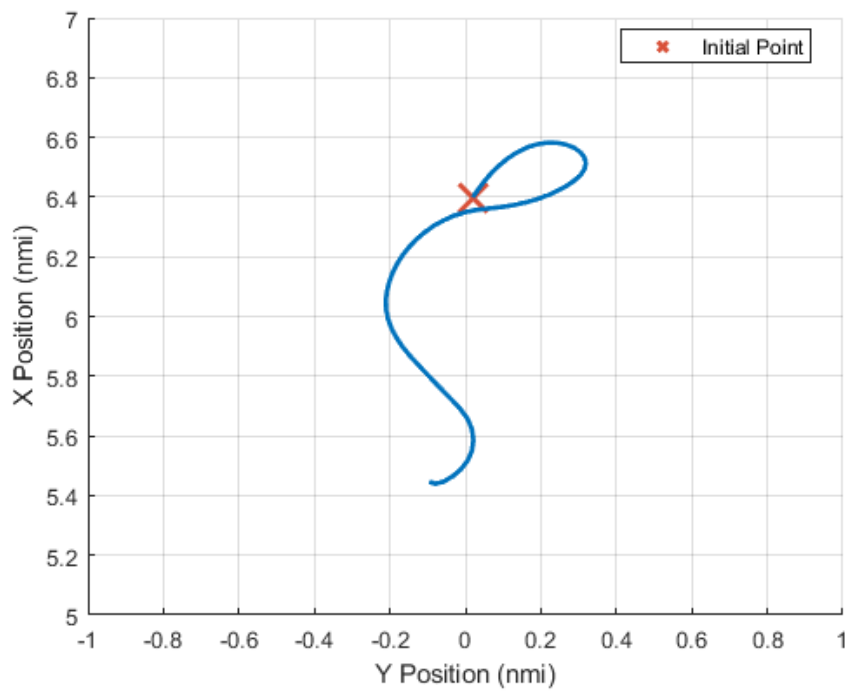


Figure 28. Complex Target 3: Run Down Short-Range Missile.

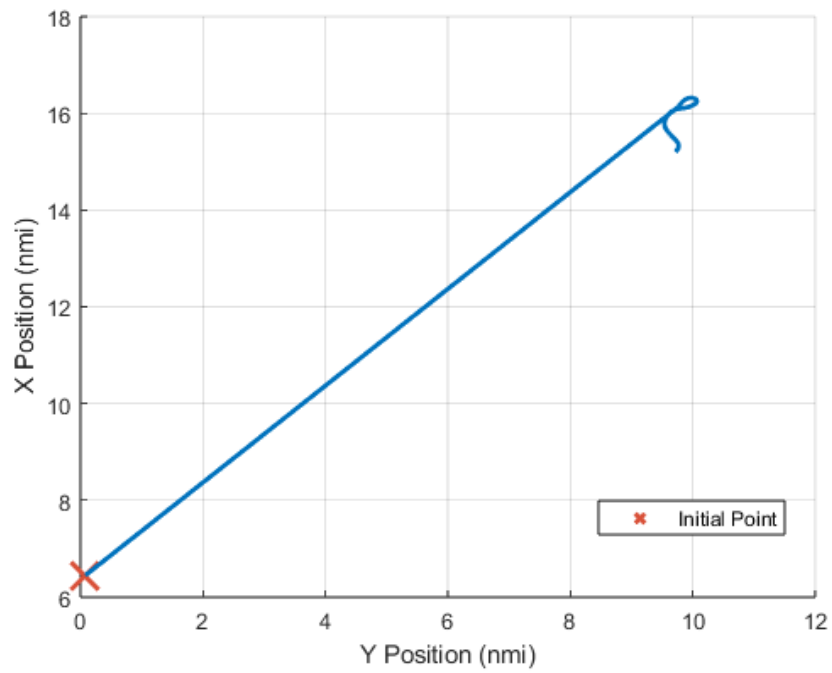


Figure 29. Complex Target 3: Run Down Medium-Range Missile.

d. Complex Target 4: High, Fast Flier

This scenario is intended to cover a scenario where the shooter aircraft is targeting a high-speed, high-altitude aircraft. In this scenario, the shooter pitches up to point towards the target and fires the missile which will have to pitch over to chase down the target. Figure 30 shows a depiction of this engagement.

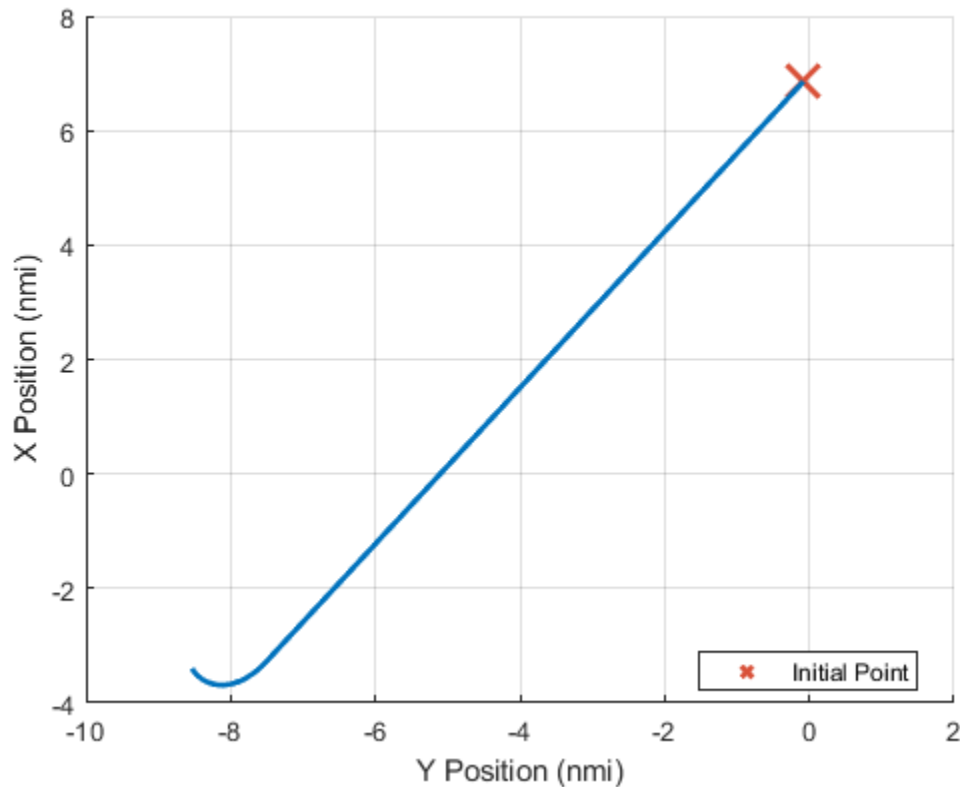


Figure 30. Complex Target 4: High, Fast Flier.

3. Long Range Missile Target Set

The long-range missile target set varies significantly from the prior ones. The goal of this test is to explore the ramifications of ultra long-range A-A missile GNC, and how to minimize the effect of target maneuvering on P_k . To that end, this experiment includes only the control set. Starting approximately 150 nm away, the targets begin executing a series of random maneuvers. All of the GNC algorithms are tested against this target set.

The results of each engagement and a plot of the various trajectories generated to approach the target are developed. For this section, the terminal maneuvering is not considered. Target behaviors are shown in Table 10.

Table 10. Long Range Missile Target Set

Target #	Maneuver 1 Time (sec)	Maneuver 1 Final Heading (deg)	Maneuver 2 Time (sec)	Maneuver 2 Final Heading (deg)
1	44	165	171	150
2	77	210	125	50
3	40	240	132	300

E. SIMULATION INTEGRATION

The final part of this chapter describes the integration of the various different simulation components into one end-to-end simulation. There are three different dynamical environments, multiple different autopilots, multiple GNC methodologies, and a target generator that all have to coexist. The easiest part to integrate is the target generator. The target generator in `FLYOUT6` is incredibly simple; it can accept input as a fit of the target's position vs. time or as an explicit equation. The other integrations are significantly more complicated.

1. Implementation of Optimal Vertical 3DOF Trajectories into 6DOF

In this thesis, DIDO runs are assumed to start after the rocket motor burnout, which is considered acceptable, as a tactical system using this methodology can launch immediately on trigger squeeze and trajectory optimization can be run during the rocket motor burn. The generated optimal trajectory is then executed throughout the flight, and the missile is controlled in the directions not optimized using the selected guidance method. For the short and medium range missiles, typically only one solution is used per flight, but multiple can be run. For the long-range weapon, the trajectory is recalculated every 30 seconds coinciding with receipt of a target update. At seeker acquisition (assumed to be 1

naautical mile) optimal guidance is deactivated, and the missile reverts to PN guidance. This process is shown in Figure 31.

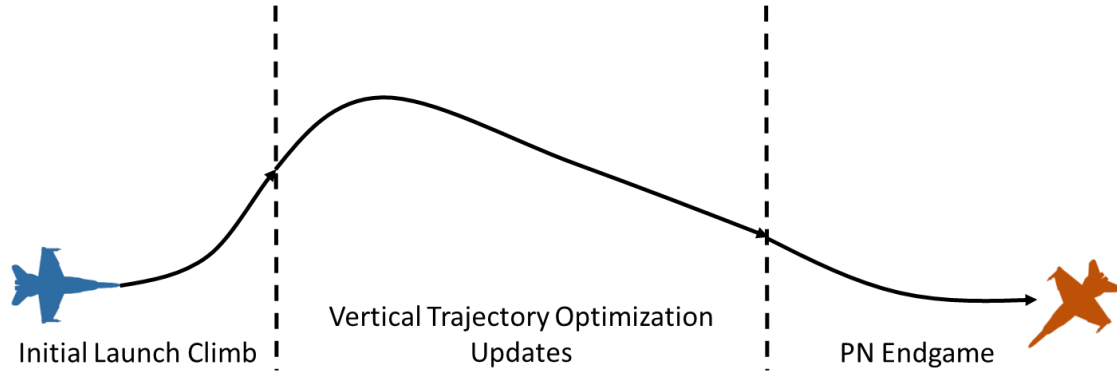


Figure 31. 3DOF Trajectory Optimization Integration to 6DOF Flyout.

The DIDO 3DOF results are fed into an autopilot whose inputs include down-range travel, altitude, and vehicle pitch angle. This generates a “maneuvering guide” for the 6DOF to use. Equation (88) presents the native altitude command guidance, and then Equation (89) presents the modified altitude command guidance used for DIDO generated trajectories that are used in the 6DOF simulation. Equation (84) modifies the default loop by replacing the static desired altitude with a variable one and adding a variable pitch angle command signal. In Equation (88) both z_{des} and θ_{des} are functions of distance traveled (D) and time elapsed (t).

$$\theta_D = K_P(z_{des} - z) + K_D(V_{z,des} - V_z) \quad (88)$$

$$\theta_D = K_P(z_{comm}(D, t) - z) + K_D(0 - V_z) + \theta_{comm}(D, t) \quad (89)$$

The validation of the 3DOF guidance is accomplished by the comparison of final states. Due to a difference in the simulation values, a “scaling factor” needed to be applied to modify the effect of aerodynamics and other forces. This scaling factor was applied to the mass of the missiles and was varied to find a value that caused the 3DOF to match the 6DOF on the majority of runs. This was required because testing revealed that the 3DOF was not accurately modeling aerodynamic drag. After experimentation with scaling

different variables, mass was determined to be the best value to scale. Using the selected scaling values achieved a positional error of 0.75% at final time (same timestep in 3DOF and 6DOF), with this error comprising a position error of 0.41%, a velocity error of 1.49%, and a negligible error in flight path angle. The optimized trajectory and the corresponding simulation-propagated trajectory are depicted in Figures 32 and 33.

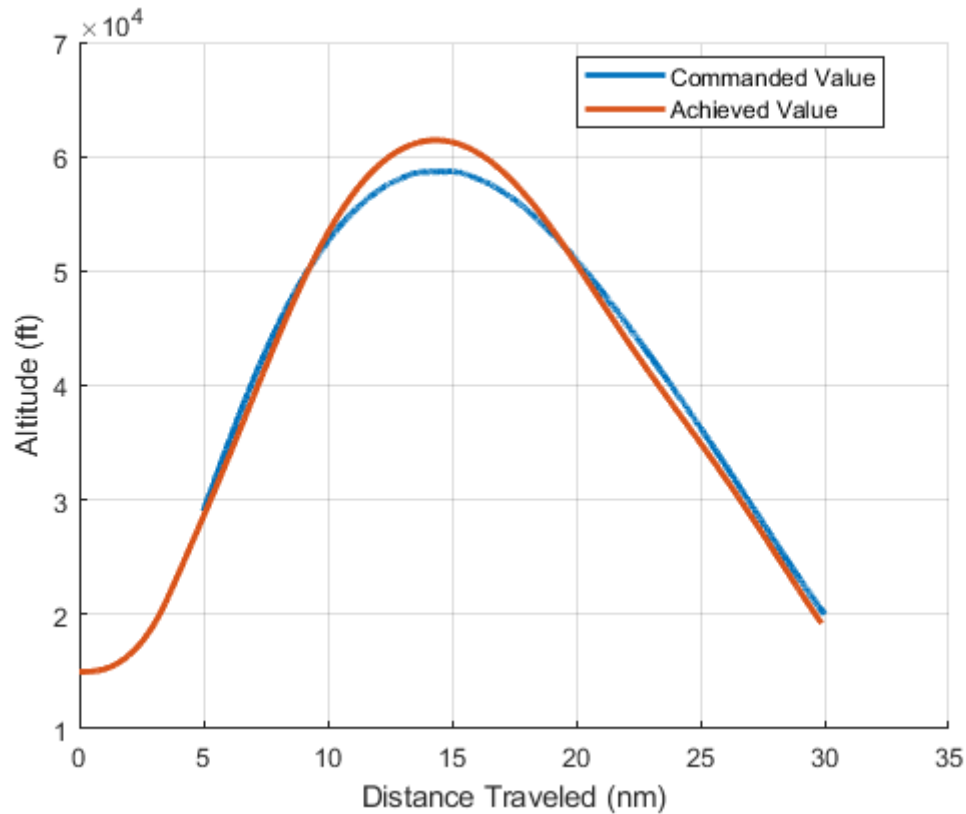


Figure 32. Commanded vs. Achieved Altitude.

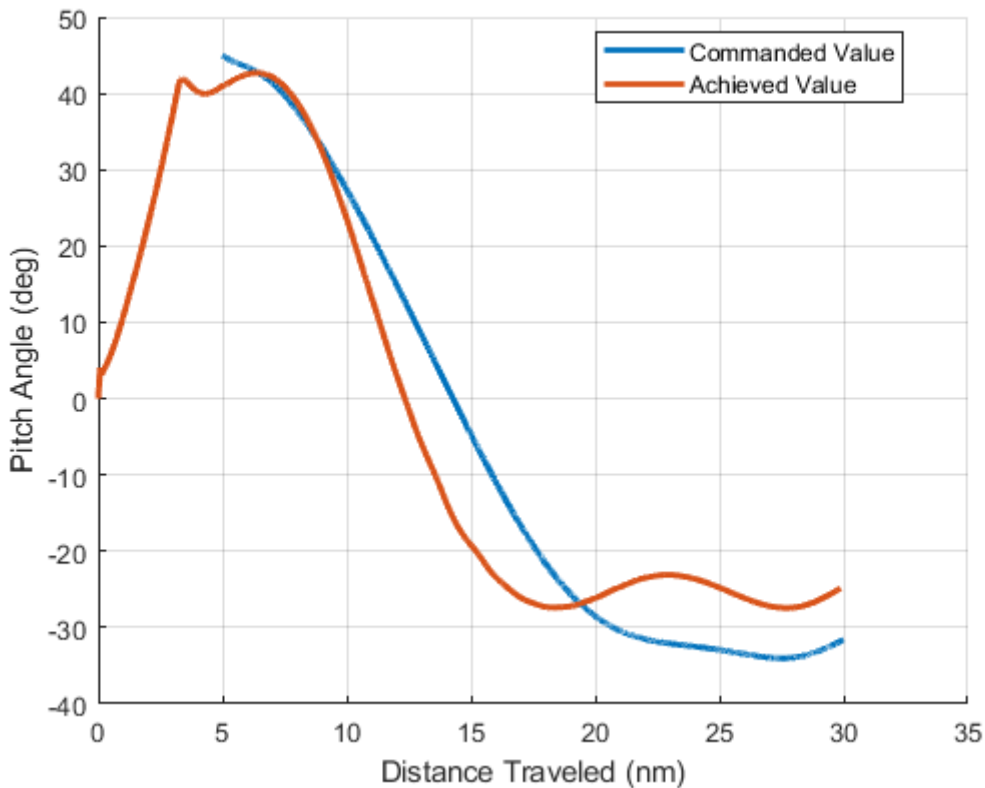


Figure 33. Commanded vs. Achieved Pitch Angles.

2. Implementation of Optimal Horizontal 3DOF Trajectories into 6DOF

The Horizontal 3DOF solution is only used for the long-range cruise weapon. This weapon has a long initial climb out to get to cruise condition, then horizontal trajectory optimization starts. This optimization runs until the weapon elects to start its descent, at which point the vertical plane 3DOF takes over until the weapon reaches seeker range. Figure 34 details the integrated flyout process.

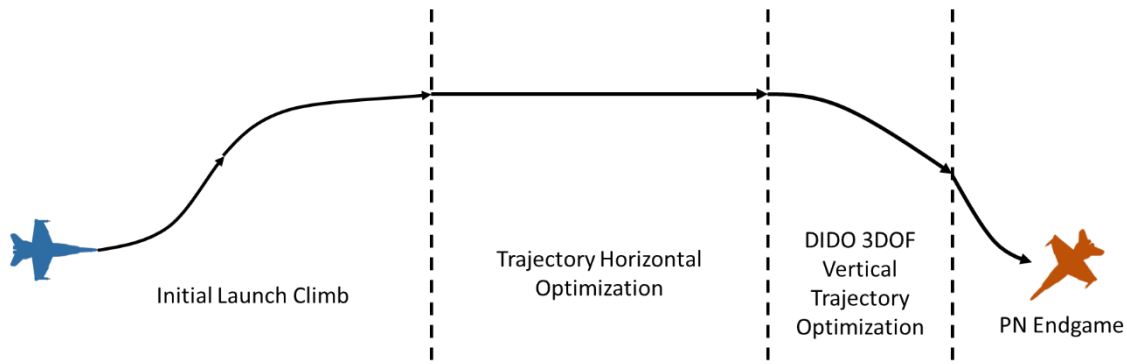


Figure 34. 3DOF Trajectory Optimization Integration into 6DOF.

The DIDO Horizontal 3DOF results are fed into a guidance algorithm via an interpolant of the optimal heading solution using time elapsed as the sole input (i.e., feedforward) as is shown in Equation (90).

$$\psi_D = \psi_{DIDO}(t) \quad (90)$$

This has been found to be sufficiently accurate for use in the cruise weapon scenario. While significant error could be propagated over the course of the flight, recalculation to accommodate target maneuvering should be able to minimize the impact. The total error was evaluated on a 75 nautical mile, 165 second flyout. 2.2% range error and 2.12% cross-range error were observed. In reality, two re-calculations would take place during this time, which would reset the error back to zero as optimization is re-calculated from the new initial condition. Once again, the angular error is negligible. Figure 35 shows the compared flyouts and Table 11 shows the errors and drift rates observed.

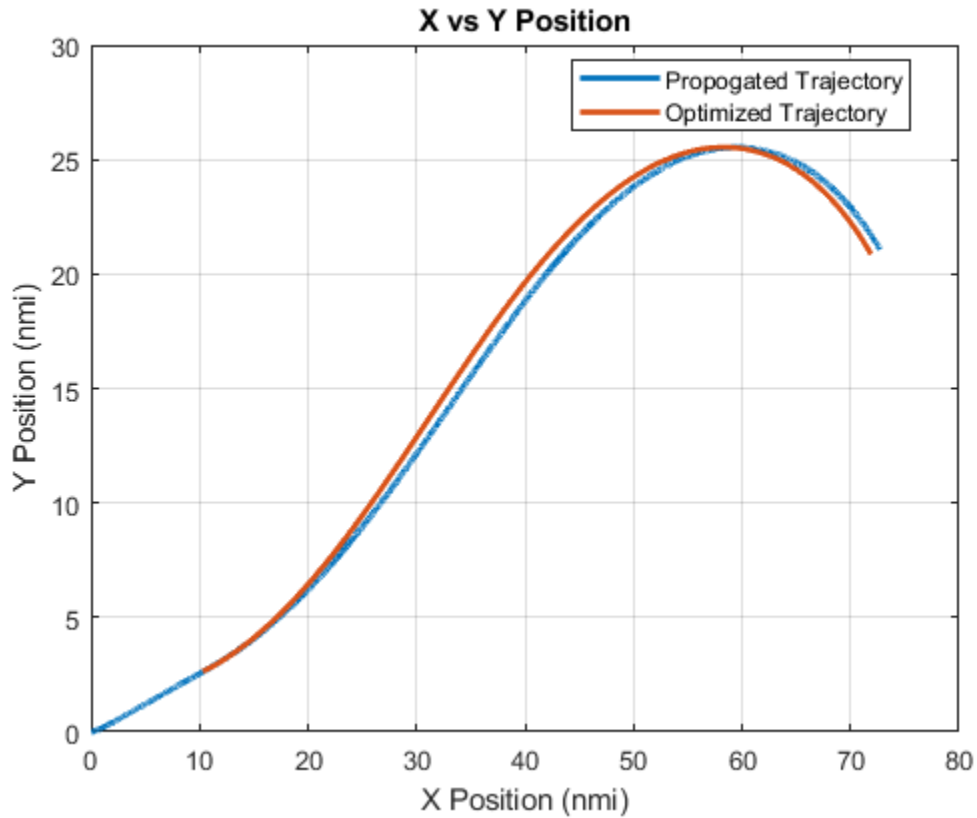


Figure 35. Optimal Trajectory vs. 6DOF FlyOut.

Table 11. Error and Drift Rates for Horizontal 3DOF

Value	Error	Drift Rate
X-Position Error	-1.15%	0.0078 % / sec
Y-Position Error	-0.95%	0.0065 % / sec

F. NON-COMMITTAL GUIDANCE OF A LONG-RANGE MISSILE

Long range air breathing weapons are an emerging area of current research. Air-breathing A-A weapons like the Meteor are entering a fairly new and unexplored design space. Theoretically, large air-breathing A-A missiles could achieve maximum ranges around 300 nautical miles, and >10-minute time of flight. Excessively long time of flight causes many more complications when even a small target maneuver can move the intercept point by 50 miles. Unlike short and medium range weapons, how these engagements are to be designed is unknown.

A proposed methodology for long range flyouts is to fly to a series of selected points that ensure that the target is unable to escape. A zone of uncertainty will be generated based on the target's current position and velocity. Once this intercept point is reached, another point will be generated based on the target's new position, until the weapon reaches sensor range. Unlike the previous two examples, a 1 nm sensor range will not be sufficient in this instance. For this application, the seeker will be assumed to gain positional information against the target at 25 nm. At that point, the weapon will proceed into a terminal mode. PG, PN, and optimal trajectory generation results will be compared to determine which is best for this application. Figure 36 shows the flyout process. Note that uncertainties are not considered in this work. Instead, the seeker acquisition is assumed at a set range. In reality, the seeker acquisition capabilities would drive most of the acceptable uncertainty volume.

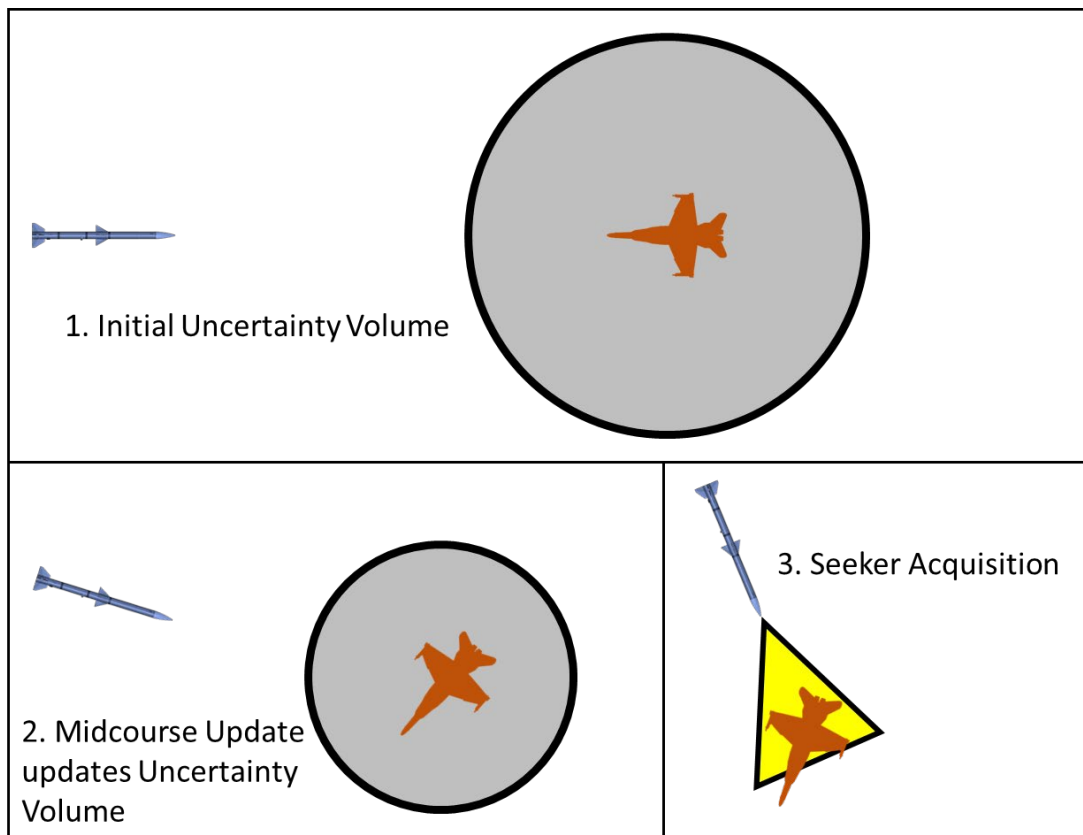


Figure 36. Long Range Missile Flyout Concept.

G. SUMMARY

The overall architecture presented in this chapter is intended to be flexible, rapidly developed, and applied to a variety of different scenarios. All of the designs use the same M&S environment (excluding trajectory optimization), autopilots, and inputs to the solver. On addition, all solutions for the presented environments follow the same process: 1) train neural networks that accurately model the flight vehicle parameters, 2) model the weapon in the appropriate 3DOF environment, and 3) deploy the 3DOF trajectory solutions to the 6DOF simulation. Care must always be taken to validate the resulting solutions.

There is certainly room for improvement to the methodology presented here, especially in the seeker modeling area. Seeker modeling is currently based on assumptions, but a seeker model could drive GNC changes to improve performance based upon a certain acquisition range or optimal acquisition aspect. In addition, if one could run an optimized 6DOF in conjunction with the 6DOF M&S environment, it is expected that additional performance gains could be realized.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS

This chapter details the results and data drawn from the integrated flight sim (IFS). All runs utilized the same code, using Bayesian optimization to characterize the response surface of each GNC algorithm. The Bayesian optimization scheme used initial launch range and optimal lofting distance (if applicable) as inputs, and the cost functional was the minimum distance to the target observed during the IFS run.

A. SHORT-RANGE MISSILE RESULTS

Overall, the short-range missile does not significantly benefit from any of the algorithms presented in this thesis. In general, it appears that the short-range missile does not have the energy required to benefit from lofting within the engagement ranges assessed in this thesis. This could also explain why the Bayesian algorithm rejected lofting in 15 of the 18 optimized lofting runs.

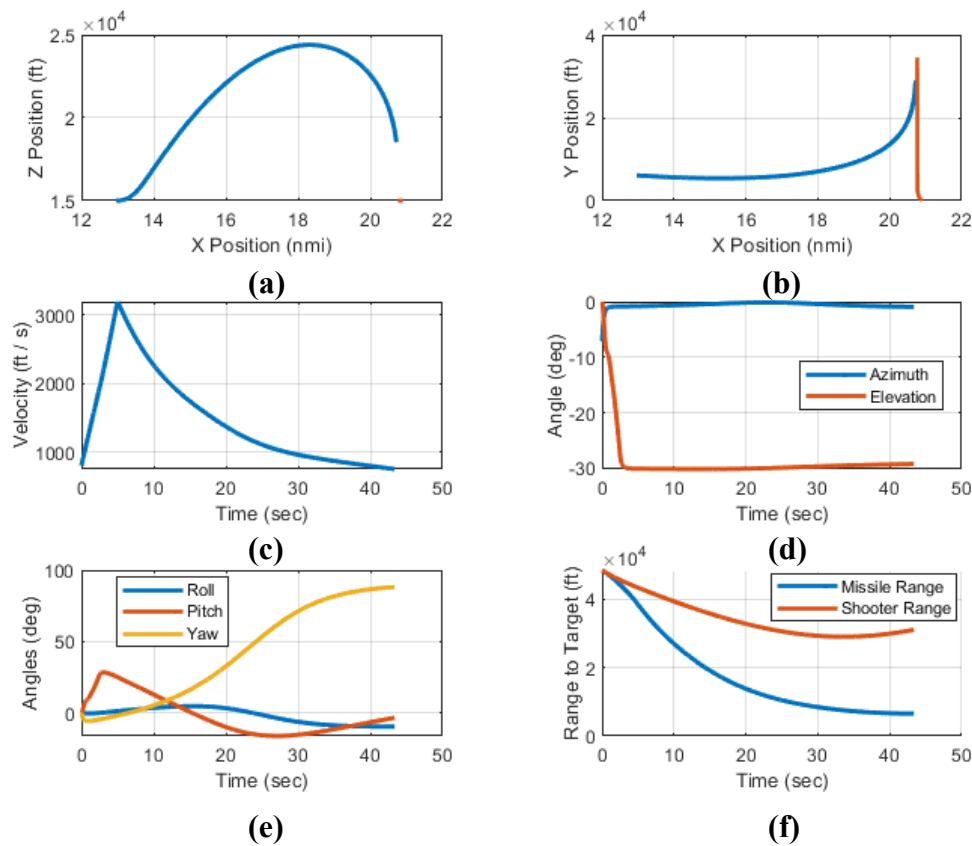
1. Short-Range Missile Control Set Results

In the control set, lofting improves the short-range missile's performance in most scenarios. Table 12 presents the observed results from the control set.

Table 12. Short-Range Missile Results for the Control Set

Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Pursuit Guidance	None	11.21	6.74	21.19	6.25	24.03
Pro-Nav	None	4.71	4.32	9.25	3.13	18.45
Non-Committal Guidance	None	4.71	3.09	11.61	3.14	23.23
Pro-Nav	Non-Optimal	10.16	6.18	19.21	5.85	21.52
Non-Committal Guidance	Non-Optimal	4.30	2.92	10.61	3.27	23.82
Pro-Nav	Optimal	11.38	5.59	25.23	4.82	31.65
Non-Committal Guidance	Optimal	9.80	5.38	20.40	4.57	27.94

In this scenario, guidance methods which used lofting generally outperformed those that did not, with the exception of pursuit. The most likely reason for pursuit's performance is that at the quoted launch ranges, pursuit has plenty of energy to reach the target. With no terminal maneuvers being performed, the most direct route to the target is often the shortest. Although P_k data is not a reliable value to draw conclusions from this data set, some of the values help to explain the observed results. Pursuit guidance, as well as the lofted trajectories considered, were able to score kills only against a subset of the targets ($P_k = 0.333$), while PN and NCG achieved much better results (PN $P_k = 1.000$; NCG $P_k = 0.889$). This occurred mainly because the Bayesian code found a minima at extremely close-range shots for these guidance methods. These minima did not exist when the same target was flown against pursuit and lofted trajectories. Pursuit was unable to close within a lethal radius against any targets with low aspect angles to the weapon. Meanwhile, all lofted trajectories did not have sufficient energy upon descent to consummate an intercept, shown in Figure 37.



(a) Missile Altitude vs. Downrange Travel, (b) Missile Downrange Travel vs. Cross Range Travel, (c) Missile Mach vs. Time, (d) Seeker Azimuth and Elevation vs. Time, (e) Missile Body Angles vs. Time, (f) Shooter and Missile Range to Target vs. Time.

Figure 37. Short-Range Missile Flyout vs. Control Target.

In this example, it can be clearly seen that the only reason the weapon was unable to consummate an intercept was due to a lack of energy at endgame. The drag observed during flyout caused the velocity to decrease significantly faster than expected. This allowed the target to outrun the weapon.

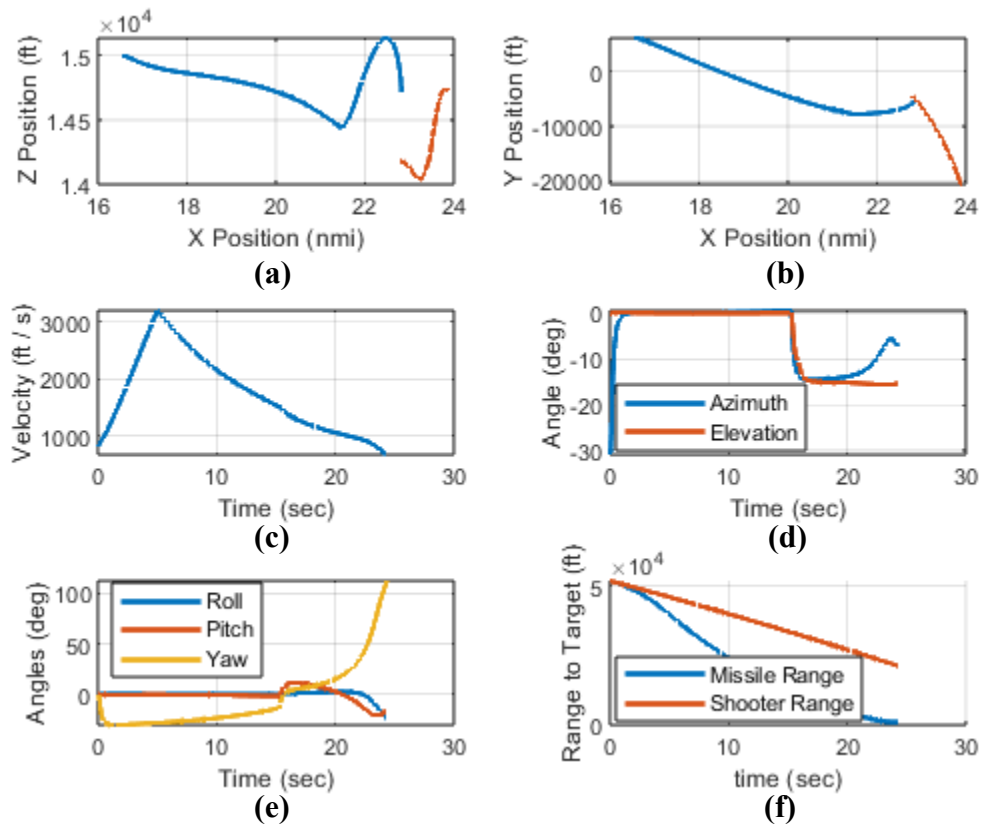
2. Short-Range Missile Complex Set Results

Performance data against Target 1 is heavily limited as most guidance methods perform poorly against this target. Table 13 shows the results from this target.

Table 13. Short Range Missile Complex Set Target 1 Results

Target Number	Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Complex Set Target 1	Pursuit Guidance	None					
	Pro-Nav	None	9.34	4.98	38.27	3.85	43.40
	Pro-Nav	Non-Optimal					
	Non-Committal Guidance	None	4.47	4.69	18.97	3.50	24.35
	Non-Committal Guidance	Non-Optimal					
	Pro-Nav	Optimal					
	Non-Committal Guidance	Optimal					

Overall, NCG performs very poorly against Target 1 due mainly to its poor performance in low aspect intercepts. Figure 38 shows the trajectory of the NCG method.



(a) Missile Altitude vs. Downrange Travel, (b) Missile Downrange Travel vs. Cross Range Travel, (c) Missile Velocity vs. Time, (d) Seeker Azimuth and Elevation vs. Time, (e) Missile Body Angles vs. Time, (f) Shooter and Missile Range to Target vs. Time.

Figure 38. Short Range Missile vs. Complex Set Target 1 Trajectory.

At seeker acquisition, the weapon executes a maneuver to achieve the desired induced LOS angles. This is a drastic maneuver that would not normally be an issue; however, due to the low mass and high drag of the airframe, significant airspeed is lost during the maneuver. Alternate approaches to the target were explored, but none yielded better performance than this one. Less drastic induced angles do not generate enough aspect to create the requisite geometry, and more drastic induced angles bleed too much energy in the turn.

The weapon performs better against Target 2. Lofting and NCG show significant performance improvements. Table 14 shows the Target 2 results.

Table 14. Short-Range Missile Complex Target Set 2

Target Number	Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Complex Set Target 2	Pursuit Guidance	None	5.06	4.98	14.94	4.23	33.36
	Pro-Nav	None					
	Pro-Nav	Non-Optimal	7.92	6.09	45.73	5.18	60.00
	Non-Committal Guidance	None	5.14	5.00	17.21	4.18	34.86
	Non-Committal Guidance	Non-Optimal	7.38	6.07	37.28	5.05	53.87
	Pro-Nav	Optimal					
	Non-Committal Guidance	Optimal					

NCG performs significantly in this scenario better than it did in the previous scenario, but not much better than pursuit guidance. In this scenario, non-optimized lofting outperforms non-lofted trajectories, but optimized lofting could not successfully consummate an intercept. The reasons for this appear to be similar to the previous scenario. The target for this scenario is at a lower altitude than the other scenarios, but the architecture did not adequately account for this variable. As a result, the non-optimized lofting flies a significantly lower trajectory before diving towards the target. On the other hand, optimized lofting flies a significantly higher trajectory and loses significant amounts of time. Figure 39 shows the trajectory and velocity from each guidance method.

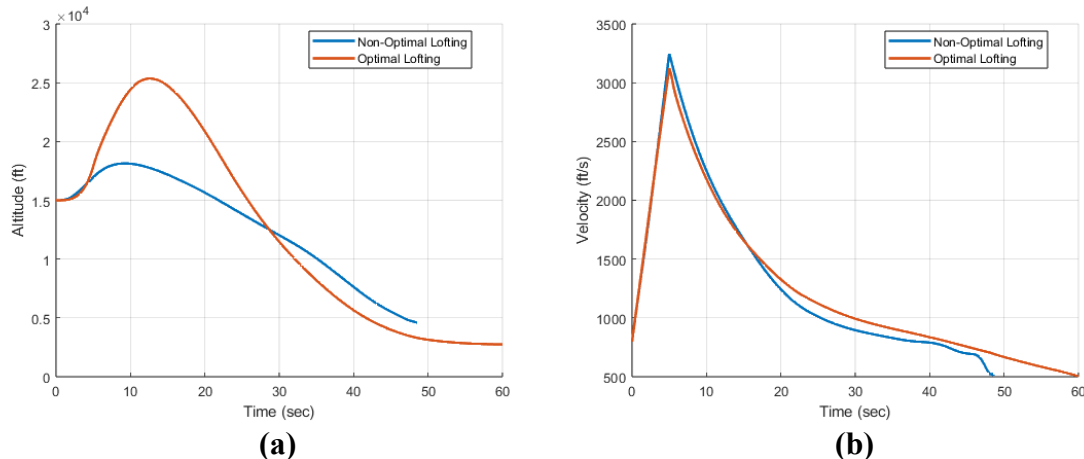


Figure 39. SRM Complex Target 2 Altitude and Velocity: (a) Missile Altitude, (b) Missile Velocity.

No other shots for the short-range missile are covered in the body of the thesis. A full data set is available in Appendix A.

B. MEDIUM-RANGE MISSILE RESULTS

Overall, the medium range missile results were improved by using the methods presented in this thesis. This is likely due to this missile's longer burn rocket motor and increased mass, which minimized the effects of aerodynamic drag. In this data set, optimal lofting was not rejected in any of the scenarios analyzed.

1. Medium-Range Missile Control Set Results

In contrast to the short-range missile, the medium-range missile saw significantly better results against the control set. Tremendous increases in launch range, A-Pole, and F-Pole are all observable in the data set as listed in Table 15.

Table 15. Medium-Range Missile Results vs. Control Set

Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Pursuit Guidance	None	22.03	14.03	53.97	13.18	75.58
Pro-Nav	None	21.07	13.97	52.47	13.60	70.88
Pro-Nav	Non-Optimal	45.59	30.69	154.71	30.64	161.21
Non-Committal Guidance	None	19.17	12.50	47.24	11.95	68.21
Non-Committal Guidance	Non-Optimal	45.00	30.21	138.12	30.15	161.89
Pro-Nav	Optimal	45.01	28.66	126.58	27.66	144.37
Non-Committal Guidance	Optimal	46.28	29.99	136.71	29.25	151.13

NCG with optimal guidance shows improvement against all other GNC methods. In Table 15, launch range and time to target are both significantly improved over alternate guidance methods. Optimal lofting likewise shows good performance compared to non-lofted and non-optimal lofting methods. Even greater performance gains were realized in the complex results.

2. Medium-Range Missile Complex Set Results

Against the complex target set, both NCG and optimal lofting showed significant improvement over the conventional GNC methods. Optimal lofting showed a 14.2 nmi increase in launch range over the nearest GNC method, and non-lofted NCG showed a 35.4% increase in launch range compared to non-lofted Pro-Nav. Table 16 shows the medium-range missile results against the complex target 1.

Table 16. Medium Range Missile Complex Set Target 1 Results

Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Pursuit Guidance	None	19.22	13.02	46.81	10.45	63.72
Pro-Nav	None	15.09	11.50	35.00	10.91	37.81
Pro-Nav	Non-Optimal					
Non-Committal Guidance	None	20.44	13.33	51.52	10.84	69.68
Non-Committal Guidance	Non-Optimal					
Pro-Nav	Optimal	34.80	20.86	89.52	19.12	98.93
Non-Committal Guidance	Optimal	34.66	20.44	91.04	18.91	99.33

Against a constantly maneuvering target non-committal guidance generally outperforms PN. Considering that non-committal guidance is explicitly designed to counter a maneuvering target, this is not surprising. In addition, the two non-optimized lofting profiles were unable to secure an intercept, while the optimized lofting profiles were the only algorithms that were able to secure a kill. This is largely because these trajectories resulted in increased kinematic energy available at endgame. Figure 40 shows the lofting profiles during flyout, Figure 41 shows the differences in velocities, and Figure 42 shows the lofting profiles of the successful guidance methods.

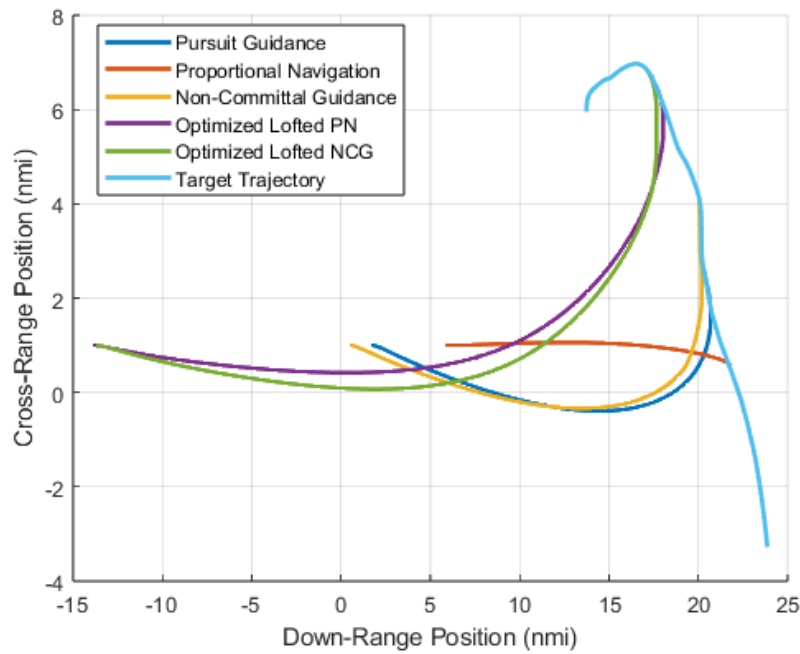


Figure 40. Complex Target 1 – Scenario 1 Medium Range Missile Trajectory Comparison.

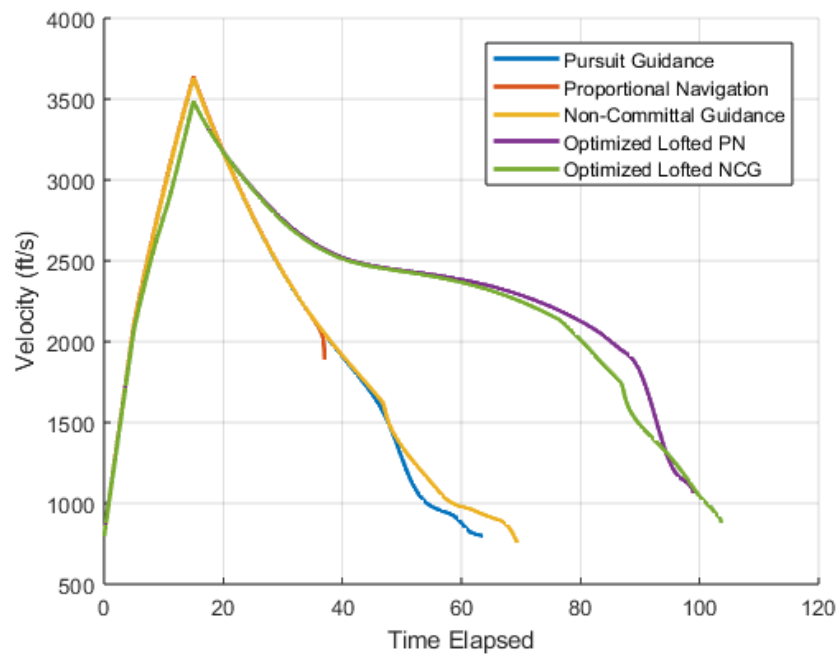


Figure 41. Complex Target 1 – Scenario 1 Medium Range Missile Velocity Comparison.

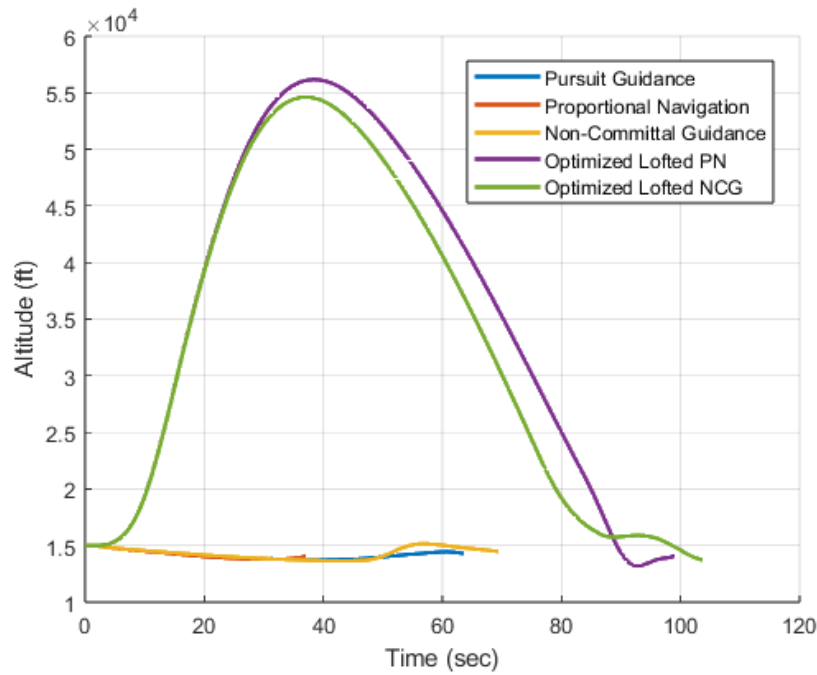


Figure 42. Complex Target Set – Scenario 1 Medium Range Missile Altitude Comparison.

As in the previous scenario, the medium range weapon is exceptionally lethal in a tail aspect engagement regardless of algorithm. Like the short-range missile, the non-optimized lofting outperforms the optimized lofting due to the lower target altitude. Table 17 shows the results of the experiment.

Table 17. Medium Range Missile Complex Set Target 2 Results

Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Pursuit Guidance	None	17.17	14.17	59.81	13.43	72.55
Pro-Nav	None	17.24	14.19	60.56	13.47	71.22
Pro-Nav	Non-Optimal	27.25	22.28	94.12	22.33	98.91
Non-Committal Guidance	None	17.20	14.09	61.43	12.93	78.71
Non-Committal Guidance	Non-Optimal	26.89	21.96	92.25	21.98	99.67
Pro-Nav	Optimal	24.73	19.79	96.31	19.88	96.31
Non-Committal Guidance	Optimal	24.77	19.77	104.61	19.86	104.61

Scenario 3 showed the exceptional performance of optimal trajectory generation against a target that is dissimilar to the previous targets. Table 18 shows the results of the experiment.

Table 18. Medium Range Missile Complex Set Target 3 Results

Target Number	Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Complex Set Target 3	Pursuit Guidance	None	19.23	15.76	74.47	14.85	100.15
	Pro-Nav	None	19.25	15.77	74.39	14.86	100.37
	Pro-Nav	Non-Optimal					
	Non-Committal Guidance	None	19.15	15.60	76.26	14.34	112.84
	Non-Committal Guidance	Non-Optimal	35.31	29.64	122.05	29.65	125.97
	Pro-Nav	Optimal	47.30	34.86	162.10	34.25	165.52
	Non-Committal Guidance	Optimal	46.61	33.52	164.95	32.79	169.04

Scenario 4 shows an interesting result. For a supersonic target flying above the shooter non-lofted profiles consistently outperformed the lofted profiles. Three of the four lofted profiles considered failed to even create an intercept. Table 19 shows the results.

Table 19. Medium Range Missile Complex Set Target 4 Results

Target Number	Guidance Method	Lofting Method	Launch Range (nmi)	A-Pole (nmi)	A-Pole Time (sec)	F-Pole (nmi)	F-Pole Time (sec)
Complex Set Target 4	Pursuit Guidance	None	5.11	15.19	33.69	29.26	67.18
	Pro-Nav	None	2.85	15.67	31.83	24.43	51.76
	Pro-Nav	Non-Optimal	1.95	34.89	75.97	47.11	108.25
	Non-Committal Guidance	None	4.01	17.74	38.22	35.45	80.98
	Non-Committal Guidance	Non-Optimal					
	Pro-Nav	Optimal					
	Non-Committal Guidance	Optimal					

Based on this engagement, pursuit is the most successful profile, while NCG was unable to achieve an intercept. Pro-nav significantly underperformed compared to pursuit. This is unsurprising against a supersonic target with head-on geometry given the autopilot limitations in that flight regime. Due to limitations with MATLAB ODESET, the GNC system struggles intercept low-aspect targets.

C. LONG RANGE MISSILE SCENARIO

The goal of this section is to the ability of optimized trajectories to improve target aspect at intercept while still maintaining a reasonably quick time to intercept. For these targets, the values of interest are Time to A-Pole, number of updates needed, and target aspect at the terminal stage of flight. Time to A-Pole has been covered previously, and the number of target position updates is merely the time of flight divided by 30 seconds. This assumes that the target position updates arrive at a constant rate, but this is a realistic assumption for weapons in this battlespace. Target aspect at terminal refers to the azimuth

LOS vector from the target aircraft to the missile. Zero degrees aspect is a head-on approach, while 180 degrees aspect is a tail-on approach. The previous sections demonstrated that a higher target aspect increases P_k against a given target.

1. Long Range Target 1

Target 1 made two turns early into its flyout at 44 and 171 seconds respectively. The first turn changed the target heading from 180 to 165 degrees; the second turn changed the target heading from 165 to 150 degrees. Pursuit reached the target first, but at a higher aspect angle than the optimal trajectory. PN was likewise faster than the optimal trajectory but required the same number of updates to reach the target. Due to the flight path of the PN engagement, the aspect angle at intercept was significantly lower than either of the other methods. Table 20 shows the results, and Figure 43 shows the flyout comparison plot.

Table 20. Long Range Target 1 Results

Method	Time to A-Pole (sec)	Updates Needed	Target Aspect at Terminal (deg)
Pursuit	299.22	9	50.9
Pro-Nav	304.37	10	7.0
Optimized Trajectory	308.05	10	66.7

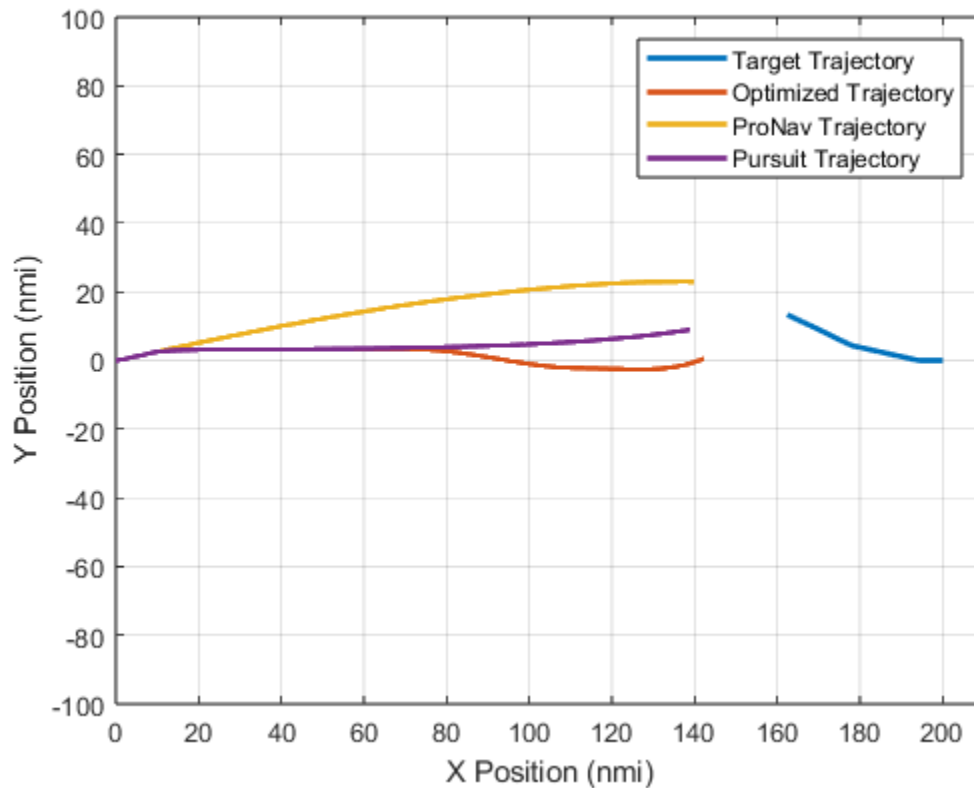


Figure 43. Long Range Target 1 Results at Handover to Terminal Guidance.

2. Long Range Target 2

Target 2 executed later and more drastic maneuvers than target 1. Two maneuvers took place at 77 and 125 seconds into the flight. These maneuvers first changed the target heading from 180 to 210 degrees, then changed the target heading from 210 to 050 degrees. This flight path resembles a target initially maneuvering to potentially defeat a weapon, then turning away from the engagement in an attempt to escape. This increased the overall flight time of the weapon by approximately 90 seconds. Due to the specific geometry of the engagement PN was the fastest GNC method to the target, and the target aspect at terminal was certainly acceptable for ensuring a kill. However, if the target had turned to re-engage the shooter, then the weapon would have had a lower P_k due to the potential for an overshoot. Table 21 shows the data gathered from the M&S simulations and Figure 44 shows all of the weapon trajectories.

Table 21. Long Range Target 2 Results

Method	Time to A-Pole (sec)	Updates Needed	Target Aspect at Terminal (deg)
Pursuit	393.38	14	152.7
Pro-Nav	391.86	14	126.1
Optimized Trajectory	399.74	14	161.4

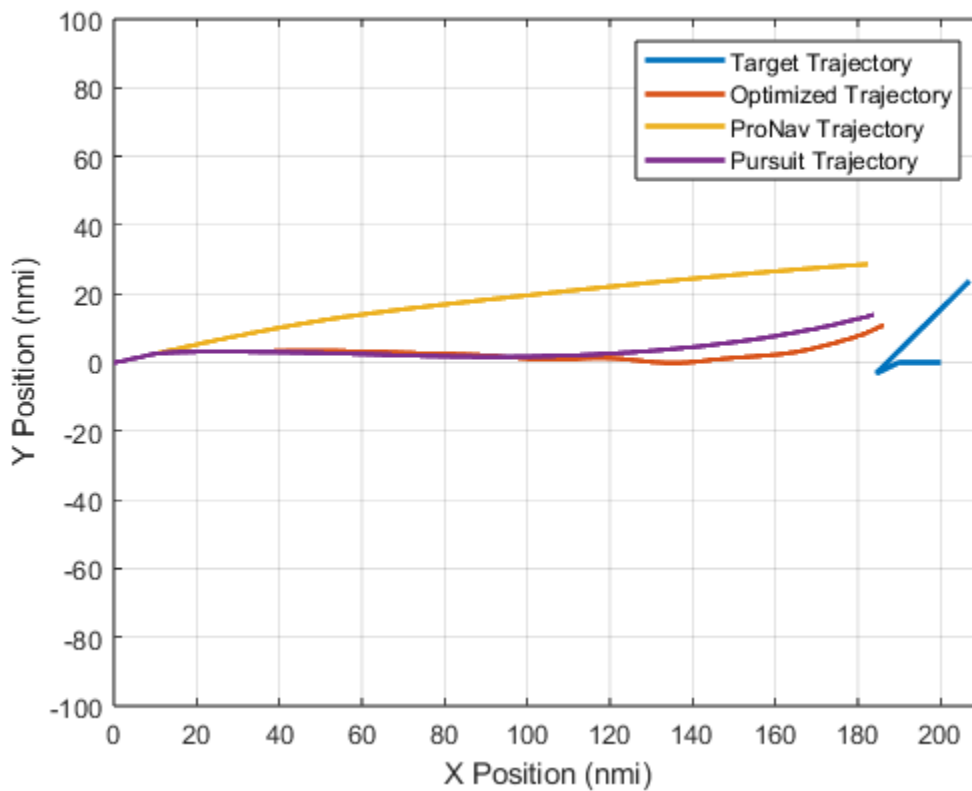


Figure 44. Long Range Target 2 Results at Handover to Terminal Guidance.

3. Long Range Target 3

Target 3 is similar to target 2, however target 3 turns to withdraw in the opposite direction. Interestingly, this significantly increased its time of flight and the relative time of flight among different guidance methods. The optimized trajectory performed very well,

therefore, significantly increasing the target aspect at A-Pole while arriving significantly sooner than PN guidance. Table 22 shows the data collected from the M&S simulations while Figure 45 shows the trajectories of the engagement.

Table 22. Long Range Target 3 Results

Method	Time to A-Pole (sec)	Updates Needed	Target Aspect at Terminal (deg)
Pursuit	401.99	14	152.3
Pro-Nav	446.57	15	170.4
Optimized Trajectory	429.3	15	173.7

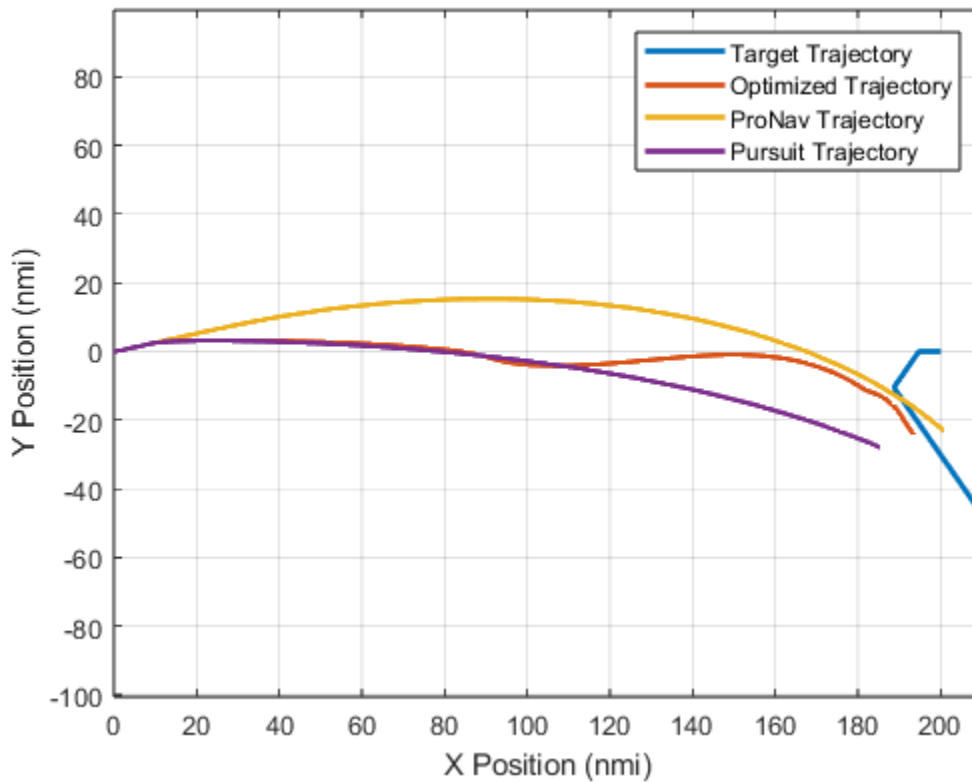


Figure 45. Long Range Target 3 Results at Handover to Terminal Guidance.

D. SUPPLEMENTAL ANALYSIS

A variety of special studies and statistics were generated to support the claims made in this thesis. Separate analyses were conducted to compare 1) PN guidance vs. NCG against a maneuvering target; 2) the runtime and reliability of the trajectory optimization scheme used in optimal lofting; and 3) the impact of using interpolants vs. discrete optimization. Run time and reliability studies are critically important as they are directly the ability to implement this methodology in actual flight hardware.

1. Comparative Analysis

The goal of this section is to compare traditional guidance methods to the presented non-committal guidance method. Previous chapters analyzed trajectories which used different guidance and lofting methods. This section analyzes PN and non-committal guidance trajectories that used the same lofting method for direct comparison of guidance methods. A Monte Carlo analysis was conducted for scenario 2 of the complex set and the medium range missile. Launch range was varied from a minimum range of 15 nautical miles to 40 nautical miles. The lofted profiles were run for both PN and NCG. Figure 46 displays each simulated engagement's miss distance as a function of launch range.

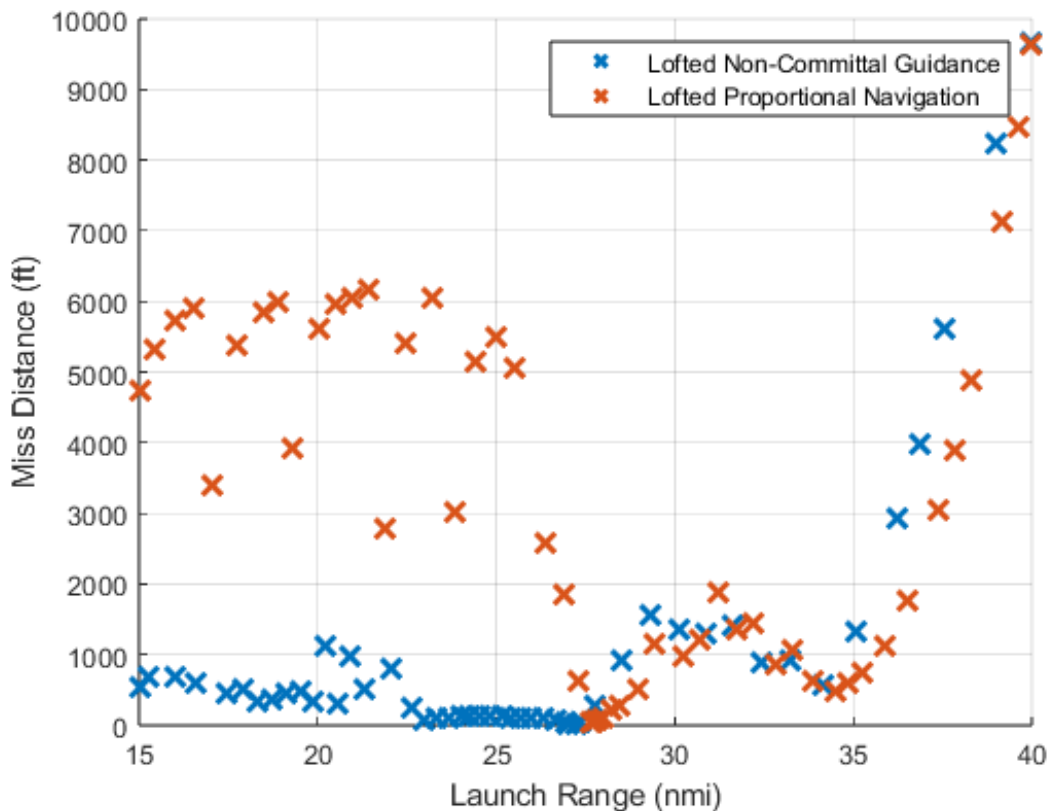


Figure 46. ProNav vs. Non-Committal Guidance Monte Carlo Results.

NCG produced excellent results against scenario 2, and pro nav was highly ineffective at countering target maneuvers. Although PN exhibits slightly greater maximum range, its heavy P_k penalty at shorter ranges does not justify this minimal increase in range. All of the complex set engagements showed similar tradeoffs between the two algorithms.

2. Run Time and Reliability

If one were to implement this method on real time flight hardware, the algorithms must be highly reliable and able to run faster than the rocket motor burn time. While the trajectories could be pre-computed before launch, in a dogfight every second counts. It is not feasible to require a pilot to maintain a nose on trajectory to the target for 5–15 seconds while the weapon calculates an optimal trajectory. The laptop used to generate the data in

this thesis is hardly a dedicated high-performance computer. Its specifications are presented in Table 23.

Table 23. Specifications on Laptop Used to Generate Data

Component	Spec
Processor	2.40 GHz
RAM	16 GB
Graphics	2 GB

Given the specs in Table 23, 100 Monte Carlo simulations were completed using random ranges for both missile configurations. Automatic V&V was performed and graded. There were five V&V conditions considered for each weapon, and each was assigned a weighting factor of 0.2. If all conditions were satisfied, failure rate was 0; for each failed condition, a 0.2 failure rate penalty was assigned. Figures 47 and 48 display the results of this test.

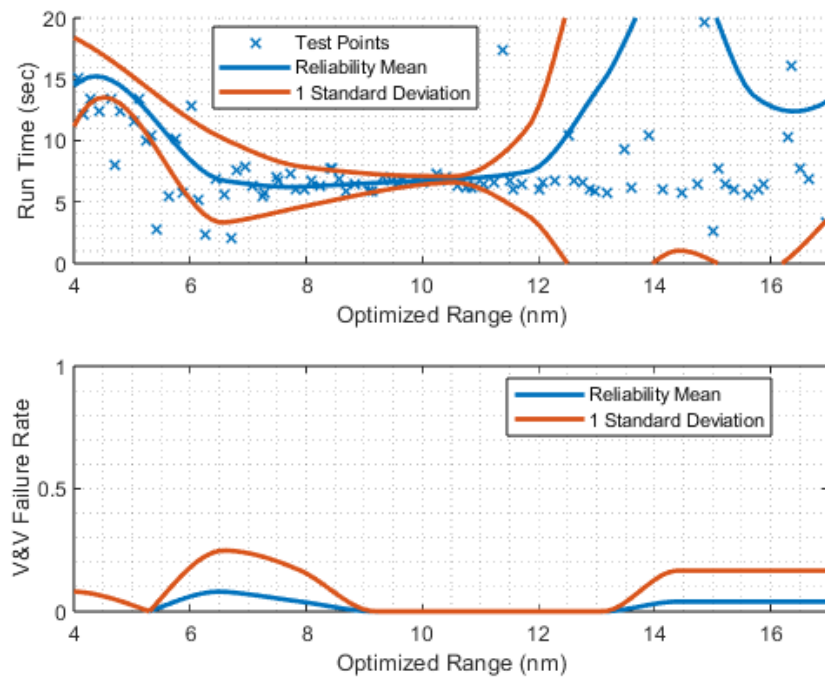


Figure 47. Short-Range Missile Runtime Analysis.

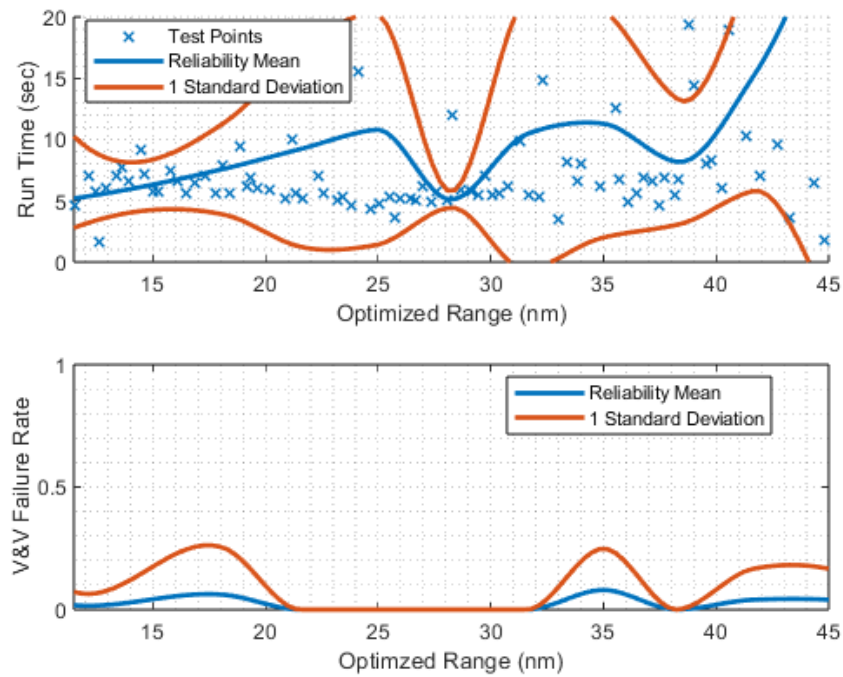


Figure 48. Medium-Range Missile Runtime Analysis.

Overall, run time and reliability was judged to be inconclusive, but likely acceptable for this application. In this implementation, optimized trajectory files take control of missile guidance after motor burnout. This occurs for the short-range missile at 5 seconds, while the medium range missile burns out at 15 seconds. The expected runtime of the medium range missile stays below 15 seconds, with a very large uncertainty, for optimized ranges below 40 nm. It is highly likely that with dedicated hardware this runtime would stay below 15 seconds for all ranges. The short-range missile experimental results are less encouraging. At no point during the entire range envelope does the expected runtime drop beneath the 5 second burnout time. While dedicated hardware should improve these runtimes, guaranteeing this level of performance requires further study.

3. Interpolants vs. Individual Optimized Profiles

Another method explored involves interpolation between optimized profiles to create near-optimal profiles virtually instantaneously. These interpolants are continuous across the optimization envelope and have no possibility of failure. For this experiment, another 100 Monte Carlo simulations were performed. These optimization runs were compared to interpolants created from a database of previously generated optimal trajectories. These optimal trajectories varied from the minimum to the maximum observed optimization ranges. The minimum range is typically determined by the range at which problem constraints make it unlikely to find an optimal solution. Optimal trajectory generation can be performed within the minimum observed range; however, within this range the weapon has enough kinematic energy that lofting is not required to intercept the target. The maximum range is defined by the kinematic capabilities of the weapon. At the maximum range, optimal trajectories cannot be generated that meet the minimum velocity requirements.

The solution database comprised optimization runs with ranges separated by no less than 0.5 nm and no greater than 1.5 nm. The collected data included time, altitude, downrange travel, and pitch angle. These values were upsampled from the number of DIDO nodes to 5000 points using a modified Akima [32] interpolant fit. The author has used these interpolants in MATLAB extensively, with significant success for these

applications. The modified Akima fit was chosen over other interpolation fits due to its ease of use and better performance against the overfitting observed when using piecewise cubic Hermite interpolation polynomial (PCHIP) [33] fits.

Once the training database was completed, a script was created to generate a maneuvering guide for a specified range. The outputs of this script were compared to the optimized trajectories generated using the general-purpose DIDO code. There was no commonality between the training set and the test sets used for this experiment. In addition, any optimization runs that failed V&V were neglected. The results are presented in Figures 49 and 50.

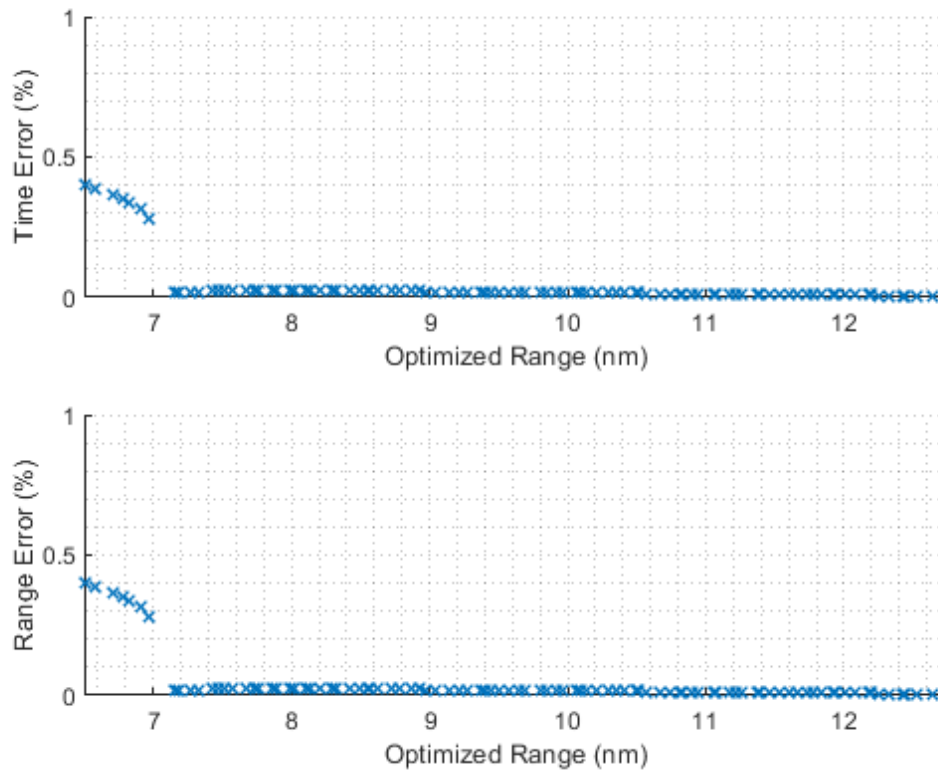


Figure 49. Short Range Missile Interpolant vs. DIDO Errors.

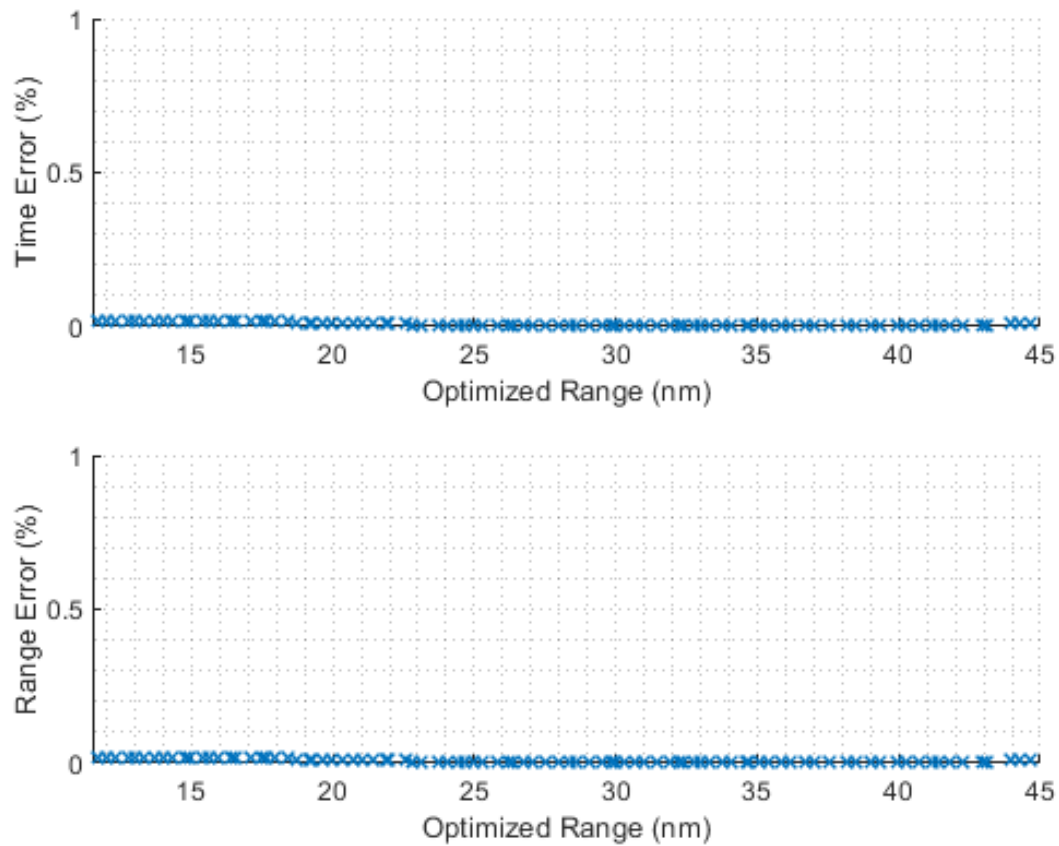


Figure 50. Medium Range Missile Interpolant vs. DIDO Errors.

Overall, the interpolants performed very well at generating sub-optimal trajectories that strongly resembled trajectories generated in DIDO. The short-range weapon interpolant did not generate accurate trajectories below 7.25 nm range, however additional investigation is required to determine the cause of these numerical discrepancies. Nevertheless, based on the above data, interpolated solutions appear suitable for time sensitive or vital systems, although a properly constructed problem should always have better accuracy. For the application of an A-A missile, the best implementation would likely be an in flight generated trajectory with an interpolant backup should the solver fail to converge in time. Implementing this methodology could be an interesting area of further study.

E. SUMMARY

Overall, the methods presented in this thesis are highly successful in reducing an air-to-air weapon's sensitivity to target maneuvers. Increasing range via optimized lofting profiles shows promise, but still requires more improvement before it can be called fully successful. Table 24 shows the percent improvement from optimized lofting vs. non-optimized lofting for the medium-range missile.

Table 24. MRM Optimized vs. Non-Optimized Percent Difference in Launch and A-Pole Ranges

Target Set	MRM Launch Range	MRM A-Pole Range
Control	2.77%	5.83%
Complex	7.83%	0.23%

In addition to the above table, optimized lofting shows significant benefits in P_k . Four of the eight non-optimized lofting scenarios resulted in a kill, while six of the eight optimized lofting scenarios resulted in a kill. In addition, one of the four non-optimized lofting scenarios that resulted in a kill was shot at a range of 1.95 nm, well within the range of the non-lofted scenarios. Furthermore, in the two scenarios where optimized lofting was unable to score a kill, close passes within 2,000 ft were executed. Those close passes were outside the grading criteria but are still worth mentioning in the author's opinion.

For the long-range weapon case, the optimized trajectory was able to reliably generate better approach angles, while not requiring significantly more updates from the shooter aircraft. Additional research should be conducted once better information is known on how these types of A-A weapons will impact the modern battlespace, but this method should reliably increase P_k while reducing a target's ability to escape.

Finally, non-committal guidance is successful at increasing P_k within the maximum range as compared against a comparable PN trajectory. Moreover, it consistently preforms better against a maneuvering target and shows significantly less sensitivity to maneuvers at any stage of flight.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

Overall, with improvements to technique and simulation it is probable that significant performance increases could be gained in current generation weapon hardware by applying the concepts described in this thesis. While some older, higher drag airframes such as the AIM-9M may not see benefits, more modern airframes such as the AIM-9X and AIM-120 could see a 5–15% increase in range. In addition, the non-committal concepts presented significantly improve performance against target maneuvers compared to a traditional PN approach. This compounds with the 5–15% range increase to improve P_k , which ultimately allows for better aircrew survivability.

The long-range weapon explores a potential future for ultra-long-range A-A weapons, but it could easily be applied to other forms of weapon guidance (such as air-to-surface) as well. The runtime and reliability studies demonstrate that with further development, it is likely that these algorithms can be executed in real time onboard these weapons. However, if on-board computation is not feasible, then interpolation from a database of points can be utilized instead.

A. ISSUES ENCOUNTERED AND POSSIBLE SOLUTIONS

During the preparation of this thesis, a number of issues were encountered; some were successfully addressed, while others were not. This section describes the issues encountered that could not be entirely overcome. Due to time constraints, not all potential solutions were explored. The subsequent sections will propose some methods for solving the encountered issues.

1. Optimizing Trajectory of 6DOF Dynamics

There are many limitations of adopting on-demand trajectory optimization for implementation in A-A missiles for a wide range of target scenarios. One of the major ones is its dependence on scaling and balancing of the relevant equations. The methodologies presented only work reliably and quickly when the problem variables are consistent from

case to case. This is because in general, each state and costate pair should have a similar magnitude [34].

The issue is exacerbated when Cartesian coordinates and rotation angles are combined. Often rotation rates are very, very small compared to Cartesian accelerations after scaling. In a situation like this, each component should be individually scaled. The scaling and balancing problem can be difficult, and the author was unable to generate working code to solve a 6DOF problem while preparing this thesis. Two possible solutions were considered, but only one of them was explored in detail. The first option involves independently scaling the cartesian and rotational parts of the equations. Due to the difference between quaternion values and Euler values, this is difficult but not impossible. However, during a brief exploration of this concept, no options proved satisfactory. The second option involves solving the translational and rotational parts of the problem using separate solvers that are appropriately configured. This option was quickly deemed to be infeasible for this project and was thus abandoned.

2. Limitations of MATLAB ODE Solvers

MATLAB ordinary differential equation solvers can solve a wide variety of problems efficiently; however, they have some limitations for use in this application. Chief among them is that the only information available to the dynamics subroutine is the current values of states and constants. Without having calculated rates or the ability to generate derivatives and integrals, several autopilot and GNC options are unavailable. The most notable drawback is the inability to create a body acceleration autopilot. Figure 51 shows the block diagram of a three-loop acceleration autopilot from an Overview of Missile Flight Control Systems [12].

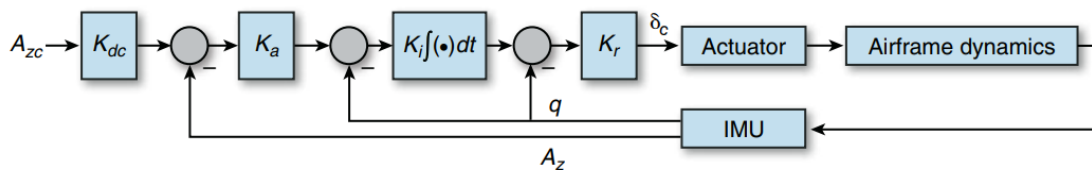


Figure 51. Three Loop Missile Autopilot.

Without the ability to perform the integral in the Figure 51, this autopilot was not available for use in this thesis. In addition, the previous step's acceleration value is not accessible in the calculation of fin inputs since acceleration is not a state, and thus not passed to the dynamics function. Furthermore, this limitation prevents utilization of a true PN guidance system.

Recreating the environment in SIMULINK and using a custom solver are both potential options, and both were explored. SIMULINK is an excellent tool; however, it can be difficult to use and has been observed (in work preceding this thesis) to run significantly slower than MATLAB. Custom solvers may offer a better option, however matching MATLAB ODE solvers' accuracy and runtime is a difficult task to accomplish.

B. RECOMMENDATIONS FOR FUTURE WORK

This thesis uncovered several interesting concepts that can be explored in future work. They are broken down into improvements to the optimization methods and endgame, as well as other recommendations gathered during the preparation of this thesis.

1. Optimization Improvements

One recommendation for future work is 6DOF trajectory optimization, specifically developing a new autopilot using Real Time Optimal Control [3] or generating 6DOF trajectories for execution by current or improved autopilots. Several of the limitations of the current methodology could be addressed with a different trajectory optimization approach or a better scaled problem. Both could potentially enable 6DOF trajectory optimization.

There are also potential ways to improve the 3DOF optimization presented in this thesis. The current methodology does not utilize re-calculation in response to target maneuvers. This is due to the difficulties in bringing solvers into the 6DOF environment. There is no feasible way to implement maneuver detection and re-calculating an optimal trajectory at every time step (or at certain intervals) would incur too much computational cost. There are two ways the author sees that the 3DOF optimization could be improved. The first method involves creating a large lookup table for virtually any conceivable

combination of missile states, target states, and engagement geometry. This would be a monumental undertaking, but it would permit practical re-calculation of optimal trajectories at every timestep of the simulation. The other option is to pursue a real time optimal control based GNC solution, which could generate optimal solutions based off the 3DOF environments in real time.

2. Endgame Improvements

All of the methods presented in this thesis struggle to consummate a forward quarter intercept. Improvements to this methodology could be achieved by considering range as well as angles to the target, or any other number of potential solutions. Additionally, midcourse guidance improvements could potentially increase P_k by optimizing the terminal target aspect.

3. Other Areas of Interest

Based on experience with this project, the author recommends that A-G and anti-ship weapons be considered as candidates for employing the methodologies presented in this thesis. These methods are expected to be highly successful against a slow or stationary target, and long-range missile scenarios against a ship target would be very interesting to explore. Additionally, guidance decisions with seeker considerations would also be an interesting area of future study.

APPENDIX A. TEST CASE DEFINITIONS

Case #	Set	Guidance Method	Maneuver (deg)	Time (sec)
1	Control	Pursuit	0	0
2	Control	Pursuit	45	30
3	Control	Pursuit	90	30
4	Control	Pursuit	135	30
5	Control	Pursuit	180	30
6	Control	Pursuit	45	0
7	Control	Pursuit	90	0
8	Control	Pursuit	135	0
9	Control	Pursuit	180	0
10	Control	Pro-Nav	0	0
11	Control	Pro-Nav	45	30
12	Control	Pro-Nav	90	30
13	Control	Pro-Nav	135	30
14	Control	Pro-Nav	180	30
15	Control	Pro-Nav	45	0
16	Control	Pro-Nav	90	0
17	Control	Pro-Nav	135	0
18	Control	Pro-Nav	180	0
19	Control	Lofted Pro-Nav	0	0
20	Control	Lofted Pro-Nav	45	30
21	Control	Lofted Pro-Nav	90	30
22	Control	Lofted Pro-Nav	135	30
23	Control	Lofted Pro-Nav	180	30
24	Control	Lofted Pro-Nav	45	0
25	Control	Lofted Pro-Nav	90	0
26	Control	Lofted Pro-Nav	135	0
27	Control	Lofted Pro-Nav	180	0
28	Control	Modified Nav	0	0
29	Control	Modified Nav	45	30
30	Control	Modified Nav	90	30
31	Control	Modified Nav	135	30
32	Control	Modified Nav	180	30
33	Control	Modified Nav	45	0
34	Control	Modified Nav	90	0
35	Control	Modified Nav	135	0
36	Control	Modified Nav	180	0

Case #	Set	Guidance Method	Maneuver (deg)	Time (sec)
37	Control	Lofted Modified Nav	0	0
38	Control	Lofted Modified Nav	45	30
39	Control	Lofted Modified Nav	90	30
40	Control	Lofted Modified Nav	135	30
41	Control	Lofted Modified Nav	180	30
42	Control	Lofted Modified Nav	45	0
43	Control	Lofted Modified Nav	90	0
44	Control	Lofted Modified Nav	135	0
45	Control	Lofted Modified Nav	180	0
46	Control	DIDO Pro Nav	0	0
47	Control	DIDO Pro Nav	45	30
48	Control	DIDO Pro Nav	90	30
49	Control	DIDO Pro Nav	135	30
50	Control	DIDO Pro Nav	180	30
51	Control	DIDO Pro Nav	45	0
52	Control	DIDO Pro Nav	90	0
53	Control	DIDO Pro Nav	135	0
54	Control	DIDO Pro Nav	180	0
55	Control	DIDO Mod Nav	0	0
56	Control	DIDO Mod Nav	45	30
57	Control	DIDO Mod Nav	90	30
58	Control	DIDO Mod Nav	135	30
59	Control	DIDO Mod Nav	180	30
60	Control	DIDO Mod Nav	45	0
61	Control	DIDO Mod Nav	90	0
62	Control	DIDO Mod Nav	135	0
63	Control	DIDO Mod Nav	180	0
64	Complex	Pursuit	Situation 1	N/A
65	Complex	Pro-Nav	Situation 1	N/A
66	Complex	Lofted Pro-Nav	Situation 1	N/A
67	Complex	Modified Nav	Situation 1	N/A
68	Complex	Lofted Modified Nav	Situation 1	N/A
69	Complex	DIDO Pro Nav	Situation 1	N/A
70	Complex	DIDO Mod Nav	Situation 1	N/A
71	Complex	Pursuit	Situation 2	N/A
72	Complex	Pro-Nav	Situation 2	N/A
73	Complex	Lofted Pro-Nav	Situation 2	N/A
74	Complex	Modified Nav	Situation 2	N/A
75	Complex	Lofted Modified Nav	Situation 2	N/A

Case #	Set	Guidance Method	Maneuver (deg)	Time (sec)
76	Complex	DIDO Pro Nav	Situation 2	N/A
77	Complex	DIDO Mod Nav	Situation 2	N/A
78	Complex	Pursuit	Situation 3	N/A
79	Complex	Pro-Nav	Situation 3	N/A
80	Complex	Lofted Pro-Nav	Situation 3	N/A
81	Complex	Modified Nav	Situation 3	N/A
82	Complex	Lofted Modified Nav	Situation 3	N/A
83	Complex	DIDO Pro Nav	Situation 3	N/A
84	Complex	DIDO Mod Nav	Situation 3	N/A
85	Complex	Pursuit	Situation 4	N/A
86	Complex	Pro-Nav	Situation 4	N/A
87	Complex	Lofted Pro-Nav	Situation 4	N/A
88	Complex	Modified Nav	Situation 4	N/A
89	Complex	Lofted Modified Nav	Situation 4	N/A
90	Complex	DIDO Pro Nav	Situation 4	N/A
91	Complex	DIDO Mod Nav	Situation 4	N/A

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. SHORT RANGE MISSILE RESULTS

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
1	18.7848		10.0854	33.124	9.5571	35.307	Yes	No
2	13.8485		7.6852	23.516	7.2416	25.348	Yes	No
3	13.8485		7.6852	23.516	7.2416	25.348	Yes	No
4	13.8485		7.6852	23.516	7.2416	25.348	Yes	No
5	13.8485		7.6852	23.516	7.2416	25.348	Yes	No
6	8.7811		5.0644	16.172	3.2538	25.446	Yes	No
7	6.8926		4.744	15.559	4.4716	18.771	Yes	Yes
8	5.7513		4.9015	15.788	4.8542	16.622	Yes	Yes
9	5.2479		5.0961	16.018	5.1205	18.708	Yes	Yes
10	3.534		1.8756	6.846	1.9996	19.406	Yes	Yes
11	3.534		1.8756	6.846	1.9996	19.406	Yes	Yes
12	3.534		1.8756	6.846	1.9996	19.406	Yes	Yes
13	3.534		1.8756	6.846	1.9996	19.406	Yes	Yes
14	3.534		1.8756	6.846	1.9996	19.406	Yes	Yes
15	6.329		3.5232	11.933	3.1658	13.62	Yes	Yes
16	7.2318		15.884	5.0461	4.8523	18.161	Yes	Yes
17	5.9679		5.1126	16.053	5.0742	18.525	Yes	Yes
18	5.1694		5.017	16.013	5.041	18.738	Yes	Yes
19	4.12673		2.1064	8.164	2.434	23.584	Yes	Yes
20	4.12673		2.1064	8.164	2.434	23.584	Yes	Yes
21	4.12673		2.1064	8.164	2.434	23.584	Yes	Yes
22	4.12673		2.1064	8.164	2.434	23.584	Yes	Yes

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
23	4.12673		2.1064	8.164	2.434	23.584	Yes	Yes
24	2.0001		1.1976	3.338	0.6224	5.913	Yes	No
25	7.8504		5.2341	20.371	4.7342	30.027	Yes	Yes
26	6.40461		5.4517	20.072	5.3668	27.589	Yes	Yes
27	5.53376		5.3839	19.857	5.4097	27.656	Yes	Yes
28	12.9514		7.2037	21.945	6.7582	23.778	Yes	No
29	12.9514		7.2037	21.945	6.7582	23.778	Yes	No
30	12.9514		7.2037	21.945	6.7582	23.778	Yes	No
31	12.9514		7.2037	21.945	6.7582	23.778	Yes	No
32	12.9514		7.2037	21.945	6.7582	23.778	Yes	No
33	9.1997		5.3313	16.881	4.8903	19.074	Yes	No
34	6.8765		4.7314	15.526	4.4681	18.627	Yes	Yes
35	5.56576		4.7265	15.347	4.6725	18.4	Yes	Yes
36	4.99642		4.8426	15.38	4.8659	18.683	Yes	Yes
37	2.2645		1.3886	4.057	2.4873	16.315	Yes	No
38	2.2645		1.3886	4.057	2.4873	16.315	Yes	No
39	2.2645		1.3886	4.057	2.4873	16.315	Yes	No
40	2.2645		1.3886	4.057	2.4873	16.315	Yes	No
41	2.2645		1.3886	4.057	2.4873	16.315	Yes	No
42	8.9421		4.4695	19.759	2.5787	31.588	Yes	No
43	7.1801		4.7203	18.838	4.2847	34.302	Yes	Yes
44	5.9173		5.0011	18.447	4.9182	34.719	Yes	Yes
45	5.31		5.1586	18.132	5.1833	32.215	Yes	Yes
46	20.1539	10.17570411	6.3093	52.784	5.2188	57.086	Yes	No
47	14.2437	Reject Flag	6.0505	31.321	5.0732	35.539	Yes	No

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
48	12.2982	Reject Flag	5.9058	24.444	4.9117	28.358	Yes	No
49	12.8953	Reject Flag	5.9639	26.49	5.0683	30.223	Yes	No
50	12.4651	6.58531233	5.4064	26.996	4.6542	30.541	Yes	No
51	11.9111	Reject Flag	5.8584	23.152	4.8899	26.974	Yes	No
52	7.3254	Reject Flag	5.1935	15.53	4.3277	30.296	Yes	Yes
53	5.9887	Reject Flag	5.1637	15	4.959	29.697	Yes	Yes
54	5.1571	Reject Flag	4.4385	11.348	4.3141	16.103	Yes	Yes
55	12.6119	Reject Flag	6.0007	25.272	5.0425	29.095	Yes	No
56	12.1033	Reject Flag	5.9429	23.559	5.0013	27.284	Yes	No
57	13.0726	Reject Flag	6.0391	26.876	5.0766	30.826	Yes	No
58	11.259	Reject Flag	5.8056	20.87	4.9254	24.363	Yes	No
59	12.7041	Reject Flag	6.0094	25.589	5.0605	29.433	Yes	No
60	9.3771	6.5003	4.434	22.062	2.6454	34.655	Yes	No
61	6.4867	Reject Flag	4.6249	12.87	3.9356	20.921	Yes	Yes
62	5.627	Reject Flag	4.8372	13.709	4.625	30.981	Yes	Yes
63	4.9169	Reject Flag	4.7625	12.804	4.7853	23.897	Yes	Yes
64							No	No
65	9.3369		4.983	38.274	3.847	43.403	Yes	No
66							No	No
67	4.472		4.6899	18.965	3.5026	24.353	Yes	No
68							No	No
69							No	No
70							No	No
71	5.06		4.9844	14.94	4.229	33.355	Yes	Yes
72							No	No

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
73	7.9223		6.0867	45.727	5.176	59.999	Yes	Yes
74	5.137		5.0043	17.205	4.1759	34.863	Yes	Yes
75	7.3792		6.0699	37.281	5.052	53.868	Yes	Yes
76							No	No
77							No	No
78	15.1125		5.5889	54.994	4.5138	59.999	Yes	No
79	14.5965		5.5445	53.151	4.3988	58.066	Yes	No
80	7.9388		4.3517	28.35	3.1279	36.441	Yes	No
81	7.524		5.3599	18.794	3.2083	33.715	Yes	No
82							No	No
83							No	No
84							No	No
85								
86								
87								
88								
89								
90								
91								

APPENDIX C. MEDIUM RANGE MISSILE RESULTS

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
1	33.6384		15.6549	68.358	14.8193	71.807	Yes	No
2	19.621		11.1001	32.617	10.5969	35.008	Yes	No
3	24.4214		13.5299	53.048	11.2608	83.102	Yes	Yes
4	23.3733		14.4117	56.734	14.0205	81.875	Yes	Yes
5	22.9793		15.0664	58.266	15.1386	87.261	Yes	Yes
6	22.9993		13.04677	45.763	9.6428	65.772	Yes	No
7	19.593		13.6866	54.878	13.1008	82.294	Yes	Yes
8	16.4586		14.7092	57.905	14.8502	88.722	Yes	Yes
9	15.2205		15.0944	58.182	15.1667	84.347	Yes	Yes
10	32.1932		15.3974	63.847	14.5768	67.235	Yes	No
11	21.6692		12.1073	37.147	11.5539	39.836	Yes	No
12	20.049		14.0756	55.205	13.4371	84.978	Yes	Yes
13	23.5414		14.5739	56.914	14.1753	82.596	Yes	Yes
14	22.9353		15.0224	57.586	15.0944	80.609	Yes	Yes
15	17.3427		10.5314	30.666	10.0248	33.394	Yes	Yes
16	20.109		14.1031	55.648	13.4859	85.628	Yes	Yes
17	16.6467		14.8921	57.532	14.9551	82.954	Yes	Yes
18	15.1745		15.0483	57.727	15.1205	80.695	Yes	Yes
19	4.1474		2.1289	8.148	2.0543	22.057	Yes	No
20	67.2673		29.622	174.359	29.016	178.948	Yes	Yes
21	54.5809		33.0219	173.21	32.9746	178.897	Yes	Yes
22	47.4455		36.1697	172.363	36.3252	178.188	Yes	Yes

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
23	45.1419		37.2343	172.421	37.4129	178.23	Yes	Yes
24	64.8911		29.9988	173.025	29.4609	177.713	Yes	Yes
25	49.2679		34.1596	172.973	34.2342	179.013	Yes	Yes
26	40.182		36.6124	172.127	36.8127	178.079	Yes	Yes
27	37.4017		37.2836	173.753	37.4623	179.721	Yes	Yes
28	4.417		2.23658	8.196	2.7017	25.816	Yes	No
29	24.2094		13.1067	44.011	12.286	47.993	Yes	No
30	24.3834		13.4852	53.114	11.4165	78.913	Yes	Yes
31	23.2433		14.2765	56.962	13.8534	88.006	Yes	Yes
32	22.6373		14.7243	56.663	14.7949	80.75	Yes	Yes
33	23.1013		11.818	37.689	10.122	46.776	Yes	No
34	19.4589		13.58	54.651	12.9911	80.408	Yes	Yes
35	16.1846		14.457	56.801	14.5446	83.1	Yes	Yes
36	14.9245		14.7981	57.037	14.869	82.083	Yes	Yes
37	3.809		2.04	7.205	2.4323	22.329	Yes	No
38	66.4956		28.9798	174.014	28.1585	179.999	Yes	Yes
39	53.7204		32.3681	171.458	32.2351	179.996	Yes	Yes
40	47.0237		35.777	35.9275	35.9275	179.999	Yes	Yes
41	44.7735		36.8658	169.768	37.0426	177.925	Yes	Yes
42	64.5575		29.6127	173.695	28.8578	179.999	Yes	Yes
43	47.9898		33.2368	169.435	33.2858	178.457	Yes	Yes
44	39.4449		35.9322	169.525	36.1458	178.308	Yes	Yes
45	37.1707		37.0525	172.062	37.2302	179.999	Yes	Yes
46	82.0909	59.4964	36.8092	171.987	36.1307	175.22	Yes	No
47	53.1248	39.5255	26.3108	118.48	22.2607	141.798	Yes	Yes

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
48	43.8585	40.0532	26.2753	121.537	25.4242	140.284	Yes	Yes
49	36.3908	38.3589	26.1518	110.245	26.1373	123.26	Yes	Yes
50	38.9269	46.4051	31.0183	130.571	31.167	156.782	Yes	Yes
51	52.3247	40.8778	26.8215	122.458	23.2103	146.324	Yes	Yes
52	35.0258	35.7172	24.5113	108.436	24.1767	124.027	Yes	Yes
53	34.4639	46.3403	31.3089	135.959	31.5243	158.636	Yes	Yes
54	28.8688	41.9526	28.7491	119.529	28.8869	133.008	Yes	Yes
55	73.9697	47.4749	33.3347	154.344	32.561	157.874	Yes	No
56	54.1173	40.586	26.3977	123.053	23.3648	140.627	Yes	Yes
57	44.2354	40.5771	26.3099	126.538	25.6425	142.825	Yes	Yes
58	42.3465	48.8535	31.5385	141.021	31.6117	153.203	Yes	Yes
59	41.5494	51.3208	33.6413	146.542	33.8026	159.286	Yes	Yes
60	52.2286	40.8346	26.4045	124.388	23.7921	141.661	Yes	Yes
61	40.4929	42.6936	28.1561	133.073	27.9472	152.054	Yes	Yes
62	37.2442	51.7168	33.8825	154.816	34.156	176.796	Yes	Yes
63	30.3407	44.8694	30.2213	126.614	30.3663	135.845	Yes	Yes
64	19.2229		13.017	46.807	10.4487	63.716	Yes	No
65	15.0945		11.4952	35.003	10.9121	37.807	Yes	No
66								
67	20.441		13.3349	51.523	10.8376	69.683	Yes	No
68								
69	34.7967	30.8723	20.8634	89.522	19.1247	98.933	Yes	Yes
70	34.6606	28.5145	20.4431	91.043	18.9115	99.326	Yes	Yes
71	17.1667		14.1743	59.809	13.4303	72.5519	Yes	Yes
72	17.2427		14.1932	60.556	13.473	71.22	Yes	Yes

Case	Launch Range (nmi)	Optimized Range (nmi)	A-Pole Range (nmi)	A-Pole Time (sec)	F-Pole Range (nmi)	F-Pole Time (sec)	Intercept Achieved	Kill Achieved
73	27.2517		22.2793	94.115	22.3316	98.914	Yes	Yes
74	17.1973		14.0862	61.427	12.9326	78.711	Yes	Yes
75	26.8922		21.9594	92.25	21.9826	99.672	Yes	Yes
76	24.7289	15.3317	19.7875	96.308	19.8824	96.308	Yes	Yes
77	24.7676	20.7789	19.7661	104.61	19.8609	104.61	Yes	Yes
78	19.2326		15.7557	74.468	14.8524	100.149	Yes	Yes
79	19.2479		15.7741	74.387	14.8592	100.374	Yes	Yes
80								
81	19.1454		15.5977	76.263	14.3404	112.839	Yes	Yes
82	35.307		29.6439	122.045	29.6457	125.97	Yes	Yes
83	47.301	52.445	34.8593	162.098	34.2459	165.515	Yes	Yes
84	46.6071	47.0009	33.5188	164.954	32.7947	169.038	Yes	No
85	5.1122		15.1852	33.692	29.2605	67.184	Yes	Yes
86	2.8458		15.6701	31.831	24.4337	51.758	Yes	Yes
87	1.9507		34.8948	75.968	47.105	108.245	Yes	Yes
88	4.0138		17.7376	38.219	35.4483	80.977	Yes	No
89								
90								
91								

LIST OF REFERENCES

- [1] R. Tedrake, “Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation,” class notes for MIT 6.832, Massachusetts Institute of Technology, Cambridge, MA, USA, February 2023. Available: <https://underactuated.csail.mit.edu/>
- [2] MathWorks, “Nonlinear Programming,” Accessed Feb. 25, 2023 Available: <https://www.mathworks.com/discovery/nonlinear-programming.html>
- [3] R. Yunashevsky, *Modern Missile Guidance*. Boca Raton, FL, USA: CRC Press, 2008.
- [4] I. M. Ross, *A Primer on Pontryagin’s Principle*, edition 2. San Francisco, CA, USA: Collegiate Publishers, 2015.
- [5] T. Harris, “How Sidewinder Missiles Work,” HowStuffWorks, Accessed Oct. 3, 2022. Available: <https://science.howstuffworks.com/sidewinder3.html>
- [6] Tiraden, *Python5_missile.jpg*. Accessed Dec. 16 2023. Available: https://commons.wikimedia.org/wiki/File:Python5_missile.jpg
- [7] M. Burns, *AIM120AMRAAM*. Accessed Dec. 16 2023. Available: <https://commons.wikimedia.org/wiki/File:AIM120AMRAAM.jpg>
- [8] Wilson44691, *AIM 7 HAFB Museum.jpg*. Accessed Dec. 16 2023. Available: https://commons.wikimedia.org/wiki/File:AIM_7_HAFB_Museum.jpg
- [9] G. C. Warne, “Unusual aerial weapons used in World War One,” Warnepieces, March 16, 2016. Available: <http://warnepieces.blogspot.com/2016/03/unusual-aerial-weapons-used-in-world.html>
- [10] R. Olds, C. Olds and E. Rasimus, *Fighter Pilot: The Memoirs of Legendary Ace Robin Olds*, New York City, NY, USA: St. Martin’s Griffin, 2011
- [11] NAVAIR, “AIM-9M Sidewinder,” Accessed Jan. 7, 2023. Available: <https://www.navair.navy.mil/product/AIM-9M-Sidewinder>
- [12] NAVAIR, “AMRAAM,.” Accessed Jan. 7, 2023. Available: <https://www.navair.navy.mil/product/AMRAAM>
- [13] P. B. Jackson, “Overview of Missile Flight Control System,” *John Hopkins APL Technical Digest*, vol. 29, no. 1, pp. 15–17, 2010. Available: <https://secwww.jhuapl.edu/techdigest/Content/techdigest/pdf/V29-N01/29-01-Jackson.pdf>

- [14] E. Berglund, “Guidance and control technology,” Swedish Defence Research Agency, Stockholm, 2001. Available: <https://apps.dtic.mil/sti/citations/ADP010953>
- [15] N. Bedrossian, S. Bhatt, M. Lammers and L. Nguyen, “Zero propellant maneuver flight results for 180 deg ISS rotation,” presented at *International Symposium on Space Flight Dynamics*, Annapolis, MD, USA, 2007.
- [16] K. P. Bollino, “High-fidelity real-time trajectory optimization for reusable launch vehicles,” Ph.D. dissertation, MAE, NPS, Monterey, CA, USA, 2006. Available: <https://apps.dtic.mil/sti/tr/pdf/ADA460473.pdf>
- [17] Elissar Global, LLC, “The DIDO Beginner’s Guide,” unpublished.
- [18] Naval Postgraduate School, “AE3830 Aerospace Guidance and Control.” Accessed July 4, 2023. Available: <https://nps.smartcatalogiq.com/current/academic-catalog/courses/ae/3000/ae3830/>
- [19] Naval Postgraduate School, “AE4850 Dynamic Optimization.” Accessed July 4, 2023. Available: <https://nps.smartcatalogiq.com/en/current/academic-catalog/courses/ae/4000/ae4850/>
- [20] Naval Postgraduate School, “AE4881 Aerospace Trajectory Planning and Guidance.” Accessed July 4, 2023. Available: <https://nps.smartcatalogiq.com/en/current/academic-catalog/courses/ae/4000/ae4881/>
- [21] “Chapter 3, System Theory,” class notes for AE3830 Aerospace Guidance and Control, MAE, Naval Postgraduate School, Monterey, Fall 2021.
- [22] J. Bernoulli, “A new problem to whose solution mathematicians are invited,” *Acta Eruditorum*, vol. 18, pp. 269, June 1696.
- [23] E. Alpaydin, *Introduction to Machine Learning*, Cambridge, UK: The MIT Press, 2013.
- [24] M. T. Hagan, H. B. Demuth, M. H. Beale and O. De Jesus, *Neural Network Design*, Stillwater, OK, USA: Hagan and Demuth. Available: <https://hagan.okstate.edu/NNDesign.pdf>
- [25] MathWorks, “Neural Networks,” Accessed Feb. 25, 2023. Available: <https://www.mathworks.com/discovery/neural-network.html>
- [26] Google “Reducing Loss: Learning Rate,” Accessed July 23, 2023. Available: <https://developers.google.com/machine-learning/crash-course/reducing-loss/learning-rate>

- [27] MathWorks, “Least Squares Model Fitting Algorithms,” Accessed Feb. 25, 2023. Available: <https://www.mathworks.com/help/optim/ug/least-squares-model-fitting-algorithms.html>
- [28] MathWorks, “Fit Data with a Shallow Neural Network,” Accessed Oct. 22, 2023. Available: <https://www.mathworks.com/help/deeplearning/gs/fit-data-with-a-neural-network.html>
- [29] Glenn Research Center, “Force Balance Coordinates,” Accessed Dec. 25, 2022. Available: <https://www.grc.nasa.gov/www/k-12/airplane/tunbalaxes.html>.
- [30] S. P. Thompson, “Life of Lord Kelvin,” London, UK: Macmillan and Co. Limited, 1910, p. 1138.
- [31] W. R. Hamiltonian, “On Quaternions; or on a new System of Imaginaries,” *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, vol. 25, pp. 1–2.
- [32] V. Dobrokhodov, “Missile Lab User Guide,” presented at ME4703 Missile Lab Workshop, Dec. 7, 2021. Available: https://cle.nps.edu/access/content/attachment/40b4e5c6-f657-43d7-8098-2527972371ee/Assignments/16a32134-5864-475b-9ca3-4ff155a30e8b/Missile_Lab_ADcoeffs.pdf
- [33] MathWorks, “makima,” Sep. 17, 2019. Available: <https://www.mathworks.com/company/newsroom/mathworks-announces-release-r2019b-of-matlab-and-simulink.html>
- [34] MathWorks, “pchip.” Accessed Nov. 28, 2023. Available: <https://www.mathworks.com/help/matlab/ref/pchip.html>
- [35] I. M. Ross, Q. Gong, M. Karpenko and R. Proulx, “Scaling and Balancing for High-Performance Computation of Optimal Controls,” *Journal of Guidance, Control and Dynamics*, vol. 41, no. 10, pp. 2086–2097, 2018.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California



DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

WWW.NPS.EDU

WHERE SCIENCE MEETS THE ART OF WARFARE